Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33LC0 系列
低功耗 MCU 芯片

产品说明书

2019. 07

章节列表

表目录

图目录

1 产品综述

1.1 概述

FM33LC0xxx系列低功耗MCU，基于ARM Cortex-M0内核，集成大容量嵌入式闪存，具备丰富的模拟和数字外设，并具有优异的低功耗特性。FM33LC0xxx系列包含两个子系列，分别为支持无晶振USB FS device的FM33LC0xxU系列，和不支持USB的FM33LC0xxN系列。

- 工作电压范围： 1.8~3.6V (USB)；1.8~5.5V (no USB)
- 工作温度范围：-40℃~+85℃
- 处理器内核
  - ARM Cortex-M0
  - 支持用户/特权模式
  - 最高主频64Mhz
  - SWD调试接口
  - 24bit Systick定时器
- 低功耗技术平台
  - 典型运行功耗120uA/MHz@48MHz，98uA/MHz@64MHz (Coremark)
  - 32KHz下LPRUN功耗：typ 30uA
  - Sleep模式：typ 6uA
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：typ 1uA
  - DeepSleep模式，RTC停止+全部RAM保持+CPU内核保持：typ 0.8uA
- 存储器
  - 64/128/256KB Flash空间
  - Flash擦写寿命：100,000次
  - Flash数据保存时间：10年@85℃
  - 用户代码保护
  - 最大24KB RAM空间
- 丰富的模拟外设
  - 高可靠、可配置BOR电路（支持4级可编程下电复位阈值）
  - 超低功耗PDR电路（支持4级可编程下电复位阈值）
  - 可编程电源监测模块（SVD）
  - 2xOPA
  - 2x低功耗模拟比较器
  - 12bit 1Msps SAR-ADC

- 高精度温度传感器，精度+/-2℃

- 通用通信接口
  - UART*4
  - LPUART*2
  - 7816主机*1
  - SPI*2，主从模式
  - I2C*1，1Mbps Fm+，主从模式
  - 7通道外设DMA
  - 可编程CRC校验模块
- USB从机
  - USB2.0 FS device，支持无晶振
  - 支持以下端点：1个双向control，2个可配置IN，2个可配置OUT
  - 512 bytes Packet RAM
  - USB2.0 LPM支持
  - Suspend/Resume
- 定时资源
  - 16bit高级定时器*1，最高PWM分辨率120MHz
  - 16bit通用定时器*2
  - 32bit基本定时器*1
  - 24-bit Systick*1
  - 32-bit低功耗定时器*1
  - 带窗口的CPU看门狗定时器*1
  - 系统看门狗定时器*1
  - 低功耗实时时钟日历（RTCC），带有数字调校功能，调校精度+/-0.476ppm
- LCD显示控制电路
  - 最大支持4COM×32SEG / 6COM×30SEG / 8COM×28SEG
  - 1/3 bias、1/4bias
  - 片内电阻分压
  - 支持休眠显示
- 安全算法
  - AES硬件运算单元，128/192/256-bit
  - AES支持ECB/CBC/CTR/GCM/GMAC模式
  - 真随机数发生器
- 时钟发生电路
  - 片上可配置高速RC振荡器，可配置频率输出8/16/24/32MHz，出厂调校误差+/-0.5%，

8/16MHz全温区变化小于+/-2%

- 低功耗32768Hz晶体振荡器，带有停振检测电路
- 低功耗低速RC振荡器，32KHz
- 高频晶体振荡器，4~32MHz
- PLL，输入1MHz，最高输出64MHz，非分频输出128MHz

1.2 芯片结构框图

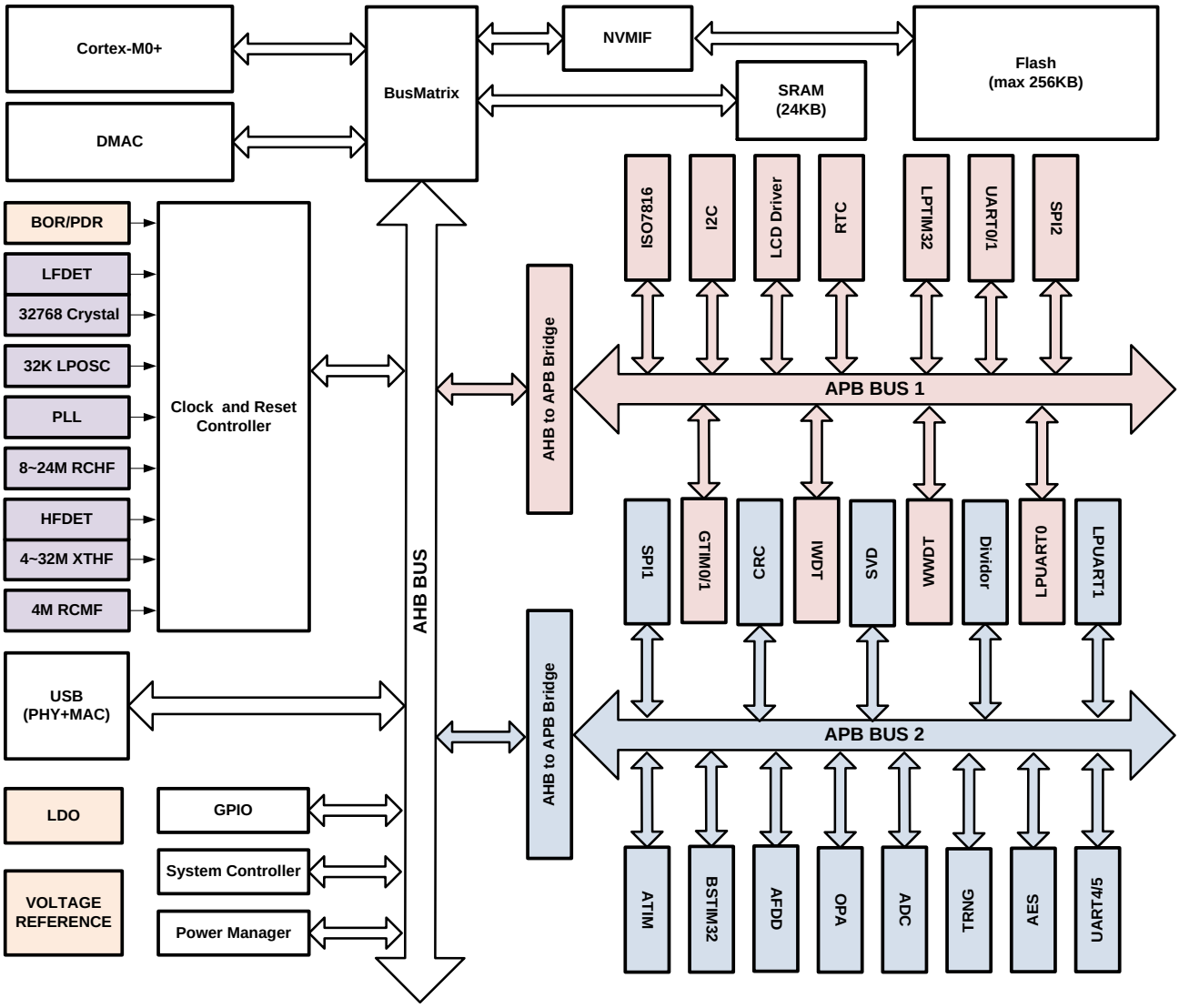


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Flash 容量（Kbytes）	RAM 容量（Kbytes）	封装
FM33LC046U/N	256	24	LQFP64
FM33LC026U/N	128	24	LQFP64
FM33LC016U/N	64	16	LQFP64
FM33LC045U/N	256	24	LQFP48
FM33LC025U/N	128	24	LQFP48
FM33LC015U/N	64	16	LQFP48
FM33LC043U/N	256	24	QFN32
FM33LC023U/N	128	24	QFN32
FM33LC013U/N	64	16	QFN32

表 1-1FM33LC0XX 型号列表

1.4 产品特性对照表

型号		FM33LC046U	FM33LC043U	FM33LC026U	FM33LC023U
CPU		Cortex-M0			
Max Freq.		64MHz			
Flash		256KB		128KB	
RAM		24KB			
AES		1			
RNG		1			
Timer	ATIM	1			
	GTIM	2			
	BSTIM32	1			
	LPTIM32	1			
	systick	1			
RTC/WWDT/IWDT		1/1/1			
SPI		2	TBD	2	TBD
I2C		1	TBD	1	TBD
UART		4	TBD	4	TBD
LPUART		2	TBD	2	TBD
USB1.1 FS		1	TBD	1	TBD
GPIO		54	TBD	54	TBD
LCD		4*32 6*30 8*28	TBD	4*32 6*30 8*28	TBD
OPA		2	TBD	2	TBD
12bit SAR-ADC		16ch	TBD	16ch	TBD
TempSensor		1			

表 1-2FM33LC0xxU 特性列表

型号		FM33LC0x6N	FM33LC0x5N	FM33LC0x3N	FM33LC0x6N	FM33LC0x5N	FM33LC0x3N
CPU		Cortex-M0					
Max Freq.		64MHz					
Flash		256KB 128KB 64KB					
RAM		24KB 24KB 16KB					
AES		1					
RNG		1					
Timer	ATIM	1					
	GTIM	2					
	BSTIM32	1					
	LPTIM32	1					
	systick	1					
RTC/WWDT/IWDT		1/1/1					
SPI		2	2	2	2	2	2
I2C		1	0	-	1	0	-
UART		4	4	4	4	4	4
LPUART		2	2	2	2	2	2
USB1.1 FS		-	-	-	-	-	-
GPIO		54	44	27	54	44	27
LCD		4*32 6*30 8*28	4*24 6*22 8*20	-	4*32 6*30 8*28	4*24 6*22 8*20	-
OPA		2	2	2	2	2	2
12bit SAR-ADC		16ch	11ch	9ch	16ch	11ch	9ch
TempSensor		1					

表 1-3FM33LC0xxN 特性列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 FM33LC0x6U 封装图（LQFP64）

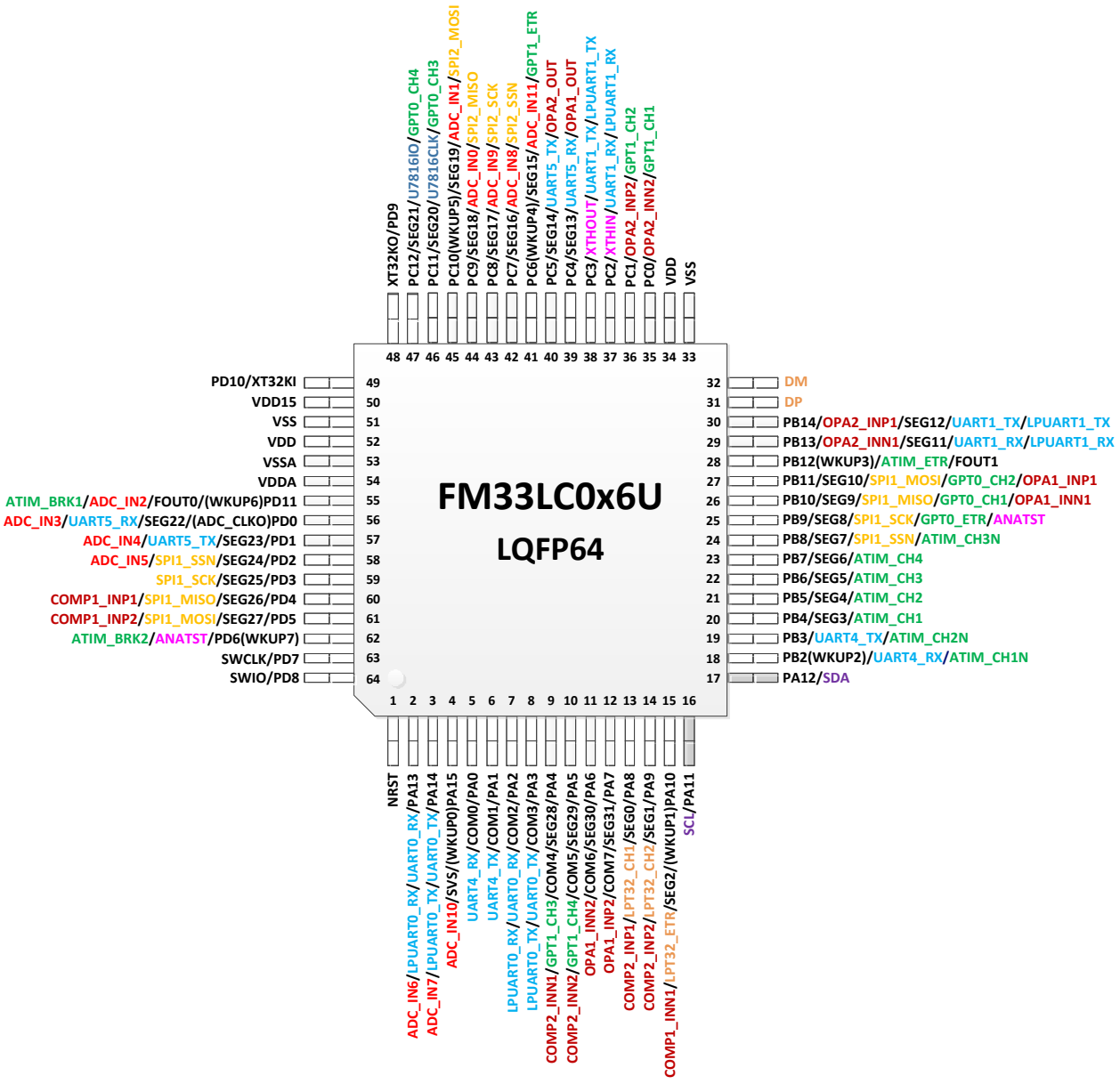


图 2-1FM33LC0x6U LQFP64 封装图

【注】

1、 PA11和PA12为真开漏引脚

2、 PB12为5V tolerant引脚

2.1.2 FM33LC0x6N 封装图（LQFP64）

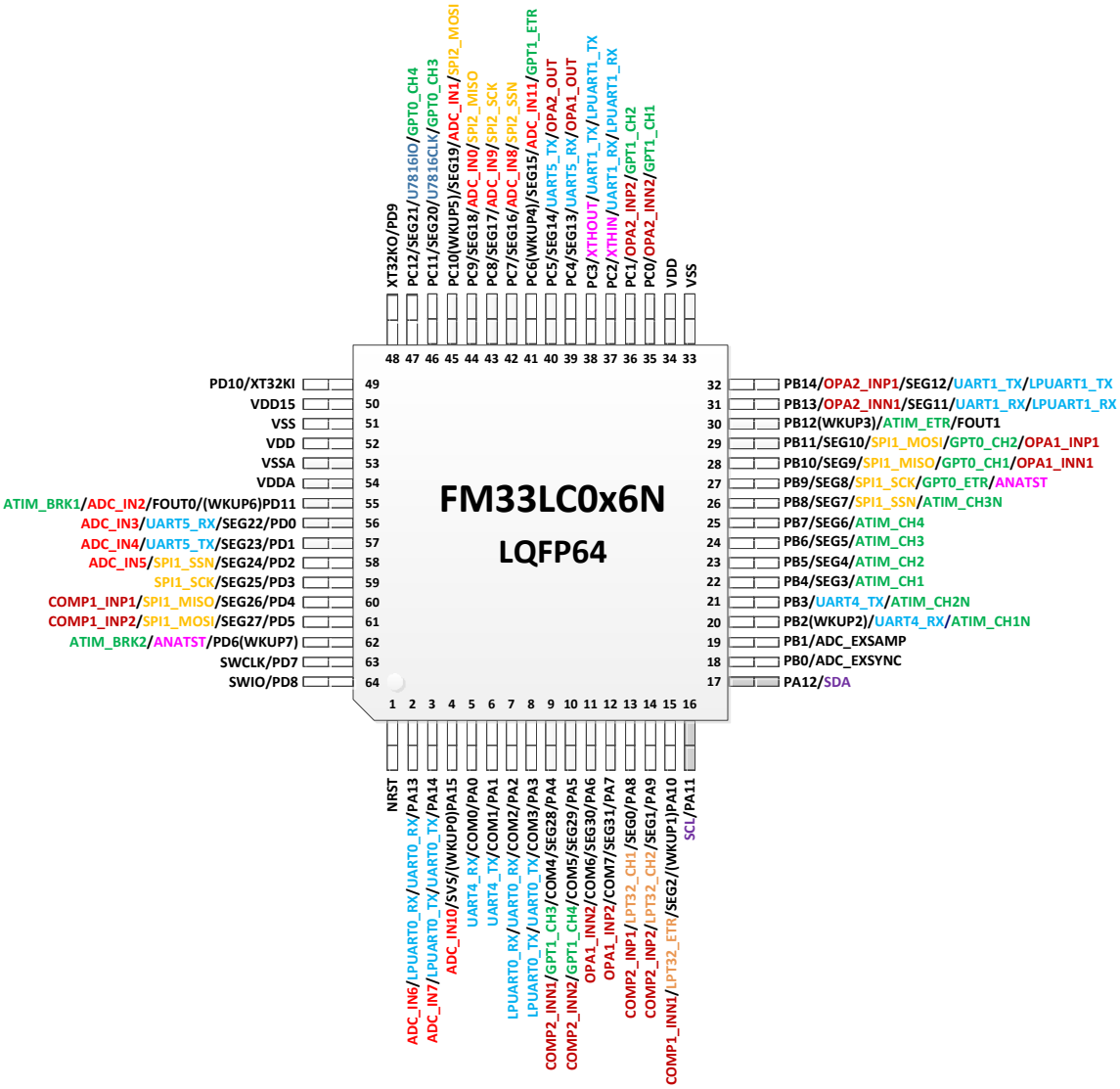


图 2-2FM33LC0x6N LQFP64 封装图

2.1.3 FM33LC0x5N 封装图（LQFP48）

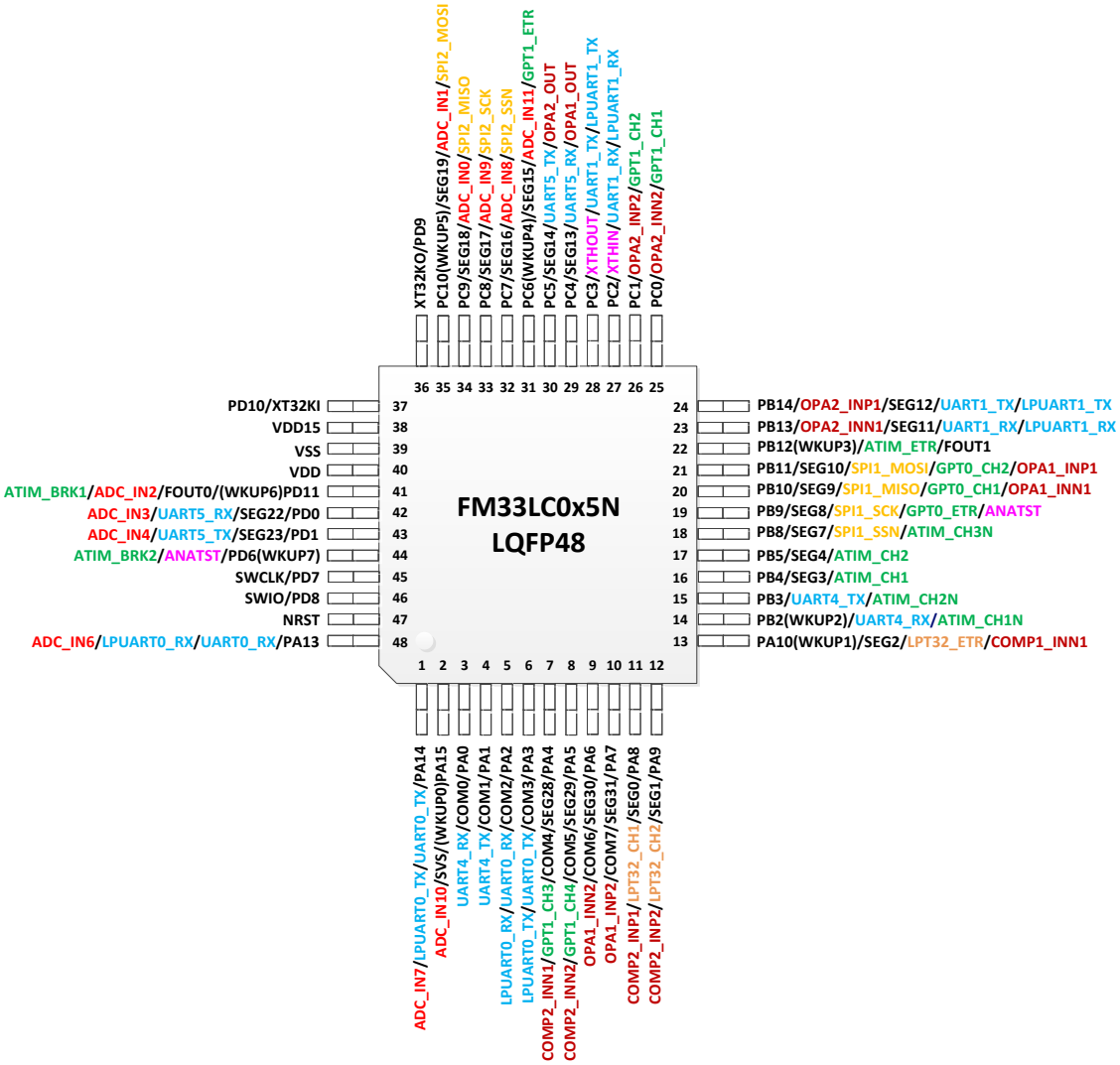


图 2-3FM33LC0x5N LQFP48 封装图

2.1.4 FM33LC0x3N 封装图（QFN32）

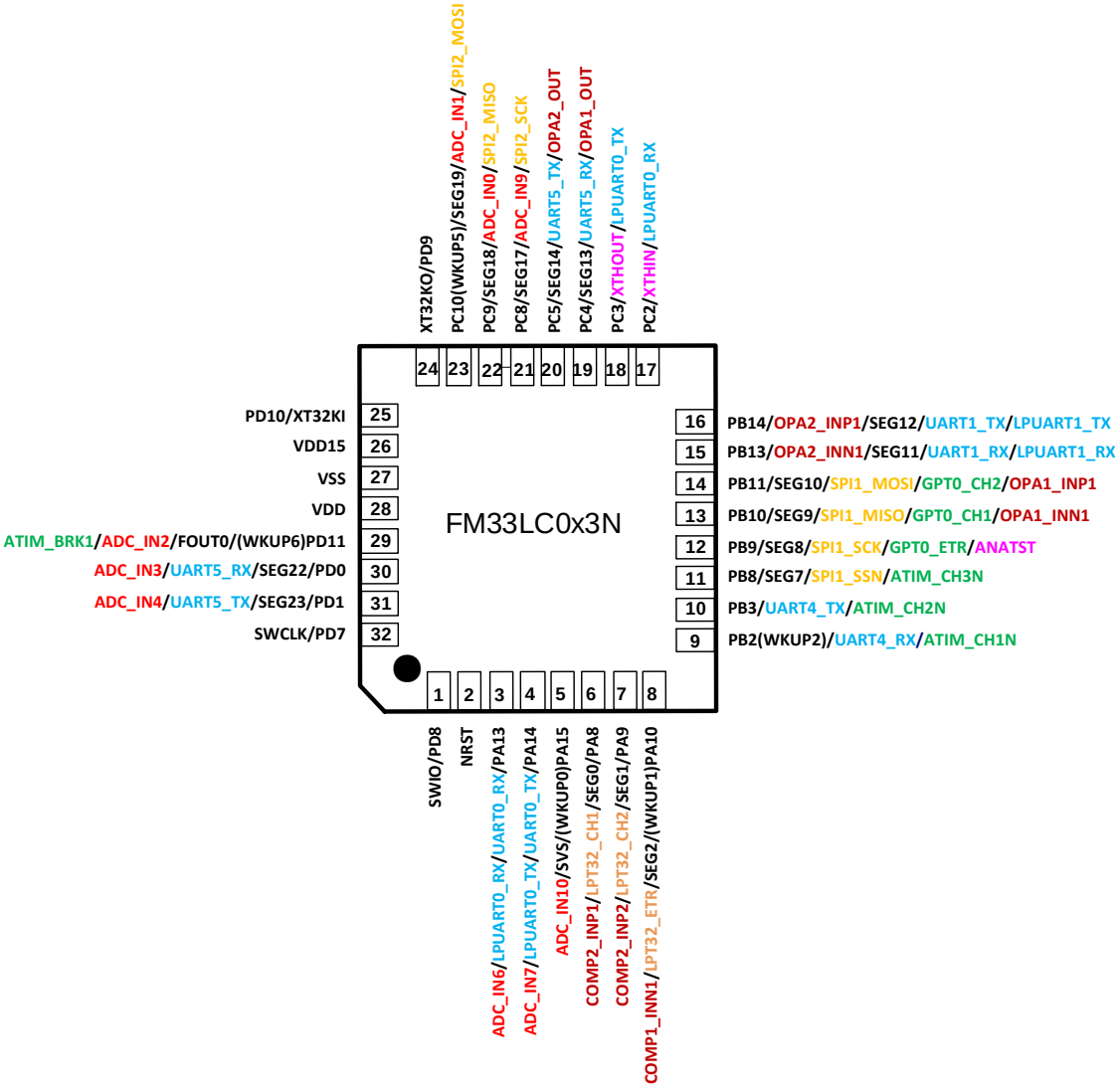


图 2-4 FM33LC0x3NQFN32 封装图

2.1.5 FM33LC0x3U 封装图（QFN32）

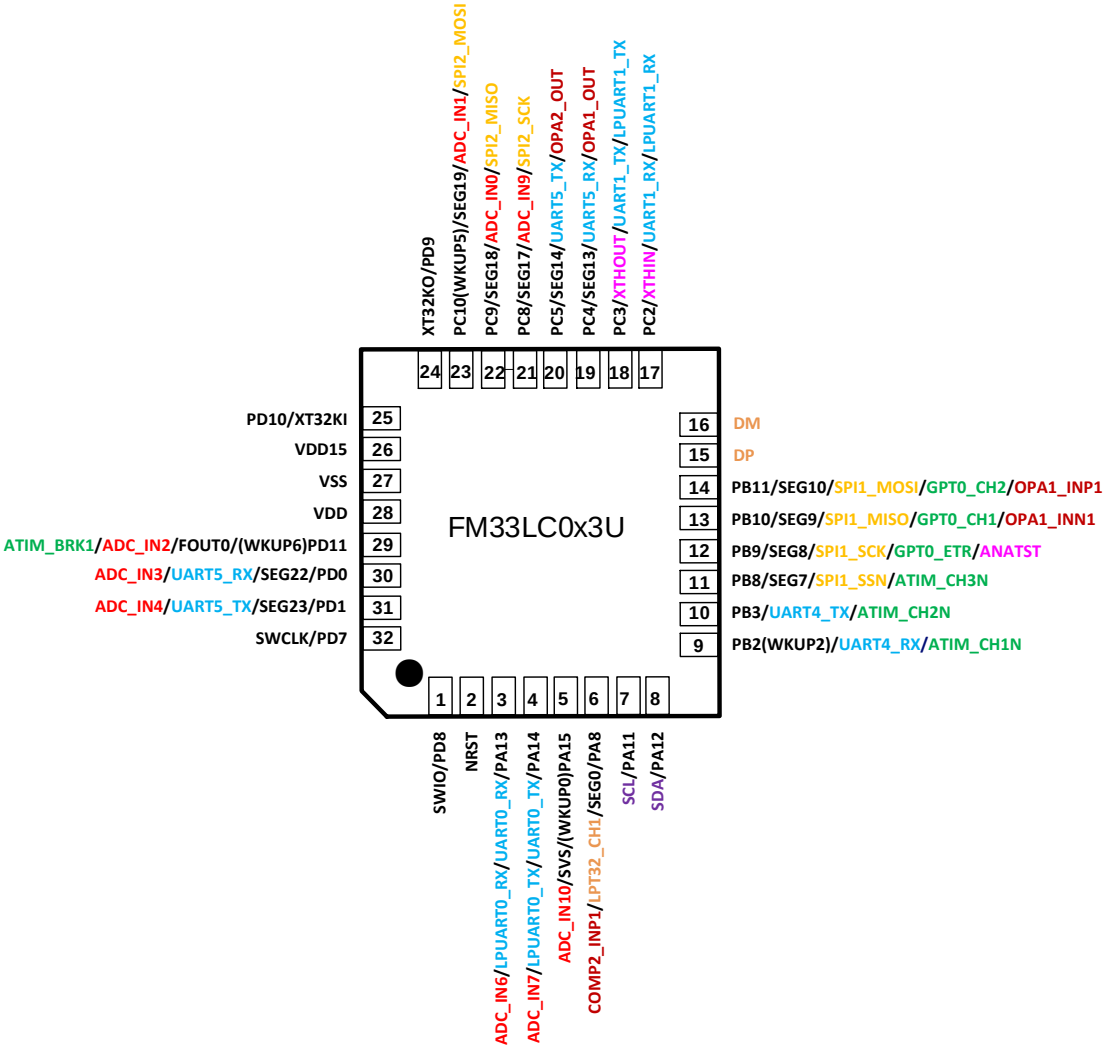


图 2-5 FM33LC0x3U QFN32 封装图

2.1.6 FM33LC0x2N 封装图（TSSOP20）

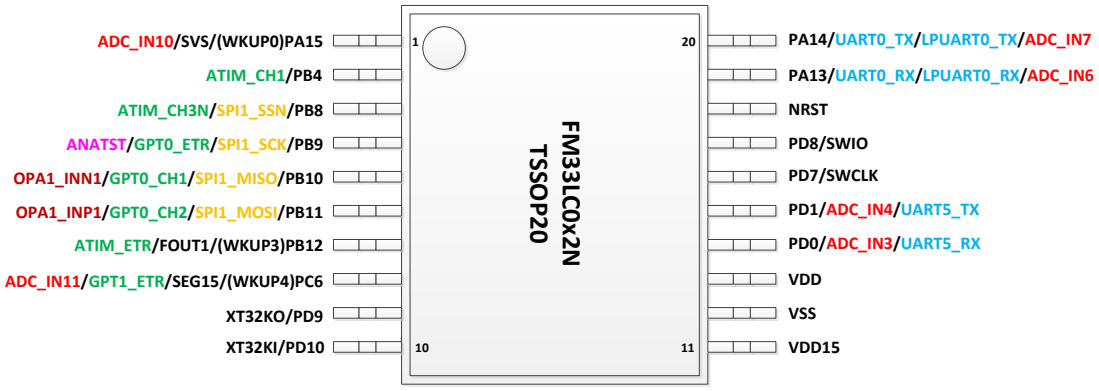


图 2-6 FM33LC0x3N TSSOP20 封装图

2.1.7 引脚功能定义（FM33LC0xxU）

Pin Number			Pin Function	Descriptions
LQFP64	LQFP48	QFN32		
1		2	NRST	全局复位引脚
2		3	PA13	GPIO
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
			ADC_IN6	ADC 输入通道
3		4	PA14	GPIO
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
			ADC_IN7	ADC 输入通道
4		5	PA15	GPIO
			WKUP0	外部唤醒引脚
			SVS	外部电源检测
			ADC_IN10	ADC 输入通道
5			PA0	GPIO
			COM0	LCD 驱动 COM 端
			UART4_RX	UART 接收
6			PA1	GPIO
			COM1	LCD 驱动 COM 端
			UART4_TX	UART 发送
7			PA2	GPIO
			COM2	LCD 驱动 COM 端
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
8			PA3	GPIO
			COM3	LCD 驱动 COM 端
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
9			PA4	GPIO
			COM4/SEG28	LCD 驱动 COM/SEG 端
			GPT1_CH3	通用定时器外部通道
			COMP2_INN1	比较器输入
10			PA5	GPIO
			COM5/SEG29	LCD 驱动 COM/SEG 端
			GPT1_CH4	通用定时器外部通道
			COMP2_INN2	比较器输入
11			PA6	GPIO
			COM6/SEG30	LCD 驱动 COM/SEG 端
			OPA1_INN2	OPA 输入通道
12			PA7	GPIO
			COM7/SEG31	LCD 驱动 COM/SEG 端
			OPA1_INP2	OPA 输入通道
13		6	PA8	GPIO

Pin Number			Pin Function	Descriptions
LQFP64	LQFP48	QFN32		
			SEG0	LCD 驱动 SEG 端
			LPT32_CH1	低功耗定时器外部通道
			COMP2_INP1	比较器输入
14			PA9	GPIO
			SEG1	LCD 驱动 SEG 端
			LPT32_CH2	低功耗定时器外部通道
			COMP2_INP2	比较器输入
15			PA10	GPIO
			WKUP1	外部唤醒引脚
			SEG2	LCD 驱动 SEG 端
			LPT32_ETR	低功耗定时器外部触发输入
			COMP1_INN1	比较器输入
16		7	PA11	GPIO
			SCL	I2C 时钟
17		8	PA12	GPIO
			SDA	I2C 数据
			PB0	GPIO
			ADC_EXSYNC	ADC 外部使能
			PB1	GPIO
			ADC_EXSAMP	ADC 外部采样控制
18		9	PB2	GPIO
			WKUP2	外部唤醒引脚
			UART4_RX	UART 接收
			ATIM_CH1N	高级定时器外部通道
19		10	PB3	外部唤醒引脚
			UART4_TX	UART 发送
			ATIM_CH2N	高级定时器外部通道
20			PB4	GPIO
			SEG3	LCD 驱动 SEG 端
			ATIM_CH1	高级定时器外部通道
21			PB5	GPIO
			SEG4	LCD 驱动 SEG 端
			ATIM_CH2	高级定时器外部通道
22			PB6	GPIO
			SEG5	LCD 驱动 SEG 端
			ATIM_CH3	高级定时器外部通道
23			PB7	GPIO
			SEG6	LCD 驱动 SEG 端
			ATIM_CH4	高级定时器外部通道
24		11	PB8	GPIO
			SEG7	LCD 驱动 SEG 端
			SPI1_SSN	SPI 片选
			ATIM_CH3N	高级定时器外部通道
25		12	PB9	GPIO
			SEG8	LCD 驱动 SEG 端
			ANATST	模拟测试通道

Pin Number			Pin Function	Descriptions
LQFP64	LQFP48	QFN32		
			SPI1_SCK	SPI 时钟
			GPT0_ETR	通用定时器外部触发输入
26		13	PB10	GPIO
			SEG9	LCD 驱动 SEG 端
			OPA1_INN1	OPA 输入
			SPI1_MISO	SPI 数据线
			GPT0_CH1	通用定时器外部通道
27		14	PB11	GPIO
			SEG10	LCD 驱动 SEG 端
			OPA1_INP1	OPA 输入
			SPI1_MOSI	SPI 数据线
			GPT0_CH2	通用定时器外部通道
28			PB12	GPIO
			WKUP3	外部唤醒引脚
			FOUT1	时钟频率输出
			ATIM_ETR	高级定时器外部触发输入
29			PB13	GPIO
			SEG11	LCD 驱动 SEG 端
			OPA2_INN1	OPA 输入
			UART1_RX	UART 接收
			LPUART1_RX	LPUART 接收
30			PB14	GPIO
			SEG12	LCD 驱动 SEG 端
			OPA2_INP1	OPA 输入
			UART1_TX	UART 发送
			LPUART1_TX	LPUART 发送
31		15	DP	USB 数据线
32		16	DM	USB 数据线
33			VSS	地
34			VDD	电源
35			PC0	GPIO
			OPA2_INN2	OPA 输入
			GPT1_CH1	通用定时器外部通道
36			PC1	GPIO
			OPA2_INP2	OPA 输入
			GPT1_CH2	通用定时器外部通道
37		17	PC2	GPIO
			XTHIN	高频晶振输入
			UART1_RX	UART 接收
			LPUART1_RX	低功耗 UART 接收
38		18	PC3	GPIO
			XTHOUT	高频晶振输出
			UART1_TX	UART 发送
			LPUART1_TX	低功耗 UART 发送
39		19	PC4	GPIO
			SEG13	LCD 驱动 SEG 端

Pin Number			Pin Function	Descriptions
LQFP64	LQFP48	QFN32		
			OPA1_OUT	OPA 输出
			UART5_RX	UART 接收
40		20	PC5	GPIO
			SEG14	LCD 驱动 SEG 端
			OPA2_OUT	OPA 输出
			UART5_TX	UART 发送
41			PC6	GPIO
			WKUP4	外部唤醒引脚
			SEG15	LCD 驱动 SEG 端
			GPT1_ETR	通用定时器外部触发输入
			ADC_IN11	ADC 输入通道
42			PC7	GPIO
			SEG16	LCD 驱动 SEG 端
			SPI2_SSN	SPI 片选
			ADC_IN8	ADC 输入通道
43		21	PC8	GPIO
			SEG17	LCD 驱动 SEG 端
			SPI2_SCK	SPI 时钟
			ADC_IN9	ADC 输入通道
44		22	PC9	GPIO
			SEG18	LCD 驱动 SEG 端
			SPI2_MISO	SPI 数据线
			ADC_IN0	ADC 输入通道
45		23	PC10	GPIO
			WKUP5	外部唤醒引脚
			SEG19	LCD 驱动 SEG 端
			SPI2_MOSI	SPI 数据线
			ADC_IN1	ADC 输入通道
46			PC11	GPIO
			SEG20	LCD 驱动 SEG 端
			U7816CLK	7816 接口时钟
			GPT0_CH3	通用定时器外部通道
47			PC12	GPIO
			SEG21	LCD 驱动 SEG 端
			U7816IO	7816 接口数据
			GPT0_CH4	通用定时器外部通道
48		24	PD9	GPIO
			XT32KO	32768Hz 晶振输出脚
49		25	PD10	GPIO
			XT32KI	32768Hz 晶振输入脚
50		26	VDD15	LDO 输出，外接 100nF 电容到地
51		27	VSS	地
52		28	VDD	电源
53			VSSA	模拟地
54			VDDA	模拟电源
55		29	PD11	GPIO

Pin Number			Pin Function	Descriptions
LQFP64	LQFP48	QFN32		
			WKUP6	外部唤醒引脚
			FOUT0	时钟频率输出
			ATIM_BRK1	高级定时器刹车输入
			ADC_IN2	ADC 输入通道
56		30	PD0	GPIO
			SEG22	LCD 驱动 SEG 端
			UART5_RX	UART 接收
			ADC_IN3	ADC 输入通道
57		31	PD1	GPIO
			SEG23	LCD 驱动 SEG 端
			UART5_TX	UART 发送
			ADC_IN4	ADC 输入通道
58			PD2	GPIO
			SEG24	LCD 驱动 SEG 端
			SPI1_SSN	SPI 片选
			ADC_IN5	ADC 输入通道
59			PD3	GPIO
			SEG25	LCD 驱动 SEG 端
			SPI1_SCK	SPI 时钟
60			PD4	GPIO
			SEG26	LCD 驱动 SEG 端
			SPI1_MISO	SPI 数据
			COMP1_INP1	比较器输入
61			PD5	GPIO
			SEG27	LCD 驱动 SEG 端
			SPI1_MOSI	SPI 数据
			COMP1_INP2	比较器输入
62			PD6	GPIO
			WKUP7	外部唤醒引脚
			ANATST	模拟测试通道
			ATIM_BRK2	高级定时器刹车输入
63		32	PD7	GPIO
			SWCLK	SWD 接口时钟
64		1	PD8	GPIO
			SWIO	SWD 接口数据

表 2-1 FM33LC0xxU 引脚列表

2.1.8 引脚功能定义（FM33LC0xxN）

Pin Number				Pin Function	Descriptions
LQFP64	LQFP48	QFN32	TSSOP20		
1	47	2	18	NRST	全局复位引脚
2	48	3	19	PA13	GPIO
				UART0_RX	UART 接收
				LPUART0_RX	低功耗 UART 接收
				ADC_IN6	ADC 输入通道
3	1	4	20	PA14	GPIO
				UART0_TX	UART 发送
				LPUART0_TX	低功耗 UART 发送
				ADC_IN7	ADC 输入通道
4	2	5	1	PA15	GPIO
				WKUP0	外部唤醒引脚
				SVS	外部电源检测
				ADC_IN10	ADC 输入通道
5	3			PA0	GPIO
				COM0	LCD 驱动 COM 端
				UART4_RX	UART 接收
6	4			PA1	GPIO
				COM1	LCD 驱动 COM 端
				UART4_TX	UART 发送
7	5			PA2	GPIO
				COM2	LCD 驱动 COM 端
				UART0_RX	UART 接收
				LPUART0_RX	低功耗 UART 接收
8	6			PA3	GPIO
				COM3	LCD 驱动 COM 端
				UART0_TX	UART 发送
				LPUART0_TX	低功耗 UART 发送
9	7			PA4	GPIO
				COM4/SEG28	LCD 驱动 COM/SEG 端
				GPT1_CH3	通用定时器外部通道
				COMP2_INN1	比较器输入
10	8			PA5	GPIO
				COM5/SEG29	LCD 驱动 COM/SEG 端
				GPT1_CH4	通用定时器外部通道
				COMP2_INN2	比较器输入
11	9			PA6	GPIO
				COM6/SEG30	LCD 驱动 COM/SEG 端
				OPA1_INN2	OPA 输入通道
12	10			PA7	GPIO
				COM7/SEG31	LCD 驱动 COM/SEG 端
				OPA1_INP2	OPA 输入通道

Pin Number				Pin Function	Descriptions
LQFP64	LQFP48	QFN32	TSSOP20		
13	11	6		PA8	GPIO
				SEG0	LCD 驱动 SEG 端
				LPT32_CH1	低功耗定时器外部通道
				COMP2_INP1	比较器输入
14	12	7		PA9	GPIO
				SEG1	LCD 驱动 SEG 端
				LPT32_CH2	低功耗定时器外部通道
				COMP2_INP2	比较器输入
15	13	8		PA10	GPIO
				WKUP1	外部唤醒引脚
				SEG2	LCD 驱动 SEG 端
				LPT32_ETR	低功耗定时器外部触发输入
				COMP1_INN1	比较器输入
16				PA11	GPIO
				SCL	I2C 时钟
17				PA12	GPIO
				SDA	I2C 数据
18				PB0	GPIO
				ADC_EXSYNC	ADC 外部使能
19				PB1	GPIO
				ADC_EXSAMP	ADC 外部采样控制
20	14	9		PB2	GPIO
				WKUP2	外部唤醒引脚
				UART4_RX	UART 接收
				ATIM_CH1N	高级定时器外部通道
21	15	10		PB3	外部唤醒引脚
				UART4_TX	UART 发送
				ATIM_CH2N	高级定时器外部通道
22	16		2	PB4	GPIO
				SEG3	LCD 驱动 SEG 端
				ATIM_CH1	高级定时器外部通道
23	17			PB5	GPIO
				SEG4	LCD 驱动 SEG 端
				ATIM_CH2	高级定时器外部通道
24				PB6	GPIO
				SEG5	LCD 驱动 SEG 端
				ATIM_CH3	高级定时器外部通道
25				PB7	GPIO
				SEG6	LCD 驱动 SEG 端
				ATIM_CH4	高级定时器外部通道
26	18	11	3	PB8	GPIO
				SEG7	LCD 驱动 SEG 端
				SPI1_SSN	SPI 片选
				ATIM_CH3N	高级定时器外部通道
27	19	12	4	PB9	GPIO
				SEG8	LCD 驱动 SEG 端

Pin Number				Pin Function	Descriptions
LQFP64	LQFP48	QFN32	TSSOP20		
				ANATST	模拟测试通道
				SPI1_SCK	SPI 时钟
				GPT0_ETR	通用定时器外部触发输入
28	20	13	5	PB10	GPIO
				SEG9	LCD 驱动 SEG 端
				OPA1_INN1	OPA 输入
				SPI1_MISO	SPI 数据线
				GPT0_CH1	通用定时器外部通道
29	21	14	6	PB11	GPIO
				SEG10	LCD 驱动 SEG 端
				OPA1_INP1	OPA 输入
				SPI1_MOSI	SPI 数据线
				GPT0_CH2	通用定时器外部通道
30	22		7	PB12	GPIO
				WKUP3	外部唤醒引脚
				FOUT1	时钟频率输出
				ATIM_ETR	高级定时器外部触发输入
31	23	15		PB13	GPIO
				SEG11	LCD 驱动 SEG 端
				OPA2_INN1	OPA 输入
				UART1_RX	UART 接收
				LPUART1_RX	LPUART 接收
32	24	16		PB14	GPIO
				SEG12	LCD 驱动 SEG 端
				OPA2_INP1	OPA 输入
				UART1_TX	UART 发送
				LPUART1_TX	LPUART 发送
				DP	USB 数据线
				DM	USB 数据线
33		17		VSS	地
34				VDD	电源
35	25			PC0	GPIO
				OPA2_INN2	OPA 输入
				GPT1_CH1	通用定时器外部通道
36	26			PC1	GPIO
				OPA2_INP2	OPA 输入
				GPT1_CH2	通用定时器外部通道
37	27	18		PC2	GPIO
				XTHIN	高频晶振输入
				UART1_RX	UART 接收
				LPUART1_RX	低功耗 UART 接收
38	28	19		PC3	GPIO
				XTHOUT	高频晶振输出
				UART1_TX	UART 发送
				LPUART1_TX	低功耗 UART 发送
39	29	20		PC4	GPIO

Pin Number				Pin Function	Descriptions
LQFP64	LQFP48	QFN32	TSSOP20		
				SEG13	LCD 驱动 SEG 端
				OPA1_OUT	OPA 输出
				UART5_RX	UART 接收
40	30			PC5	GPIO
				SEG14	LCD 驱动 SEG 端
				OPA2_OUT	OPA 输出
				UART5_TX	UART 发送
41	31		8	PC6	GPIO
				WKUP4	外部唤醒引脚
				SEG15	LCD 驱动 SEG 端
				GPT1_ETR	通用定时器外部触发输入
				ADC_IN11	ADC 输入通道
42	32			PC7	GPIO
				SEG16	LCD 驱动 SEG 端
				SPI2_SSN	SPI 片选
				ADC_IN8	ADC 输入通道
43	33	21		PC8	GPIO
				SEG17	LCD 驱动 SEG 端
				SPI2_SCK	SPI 时钟
				ADC_IN9	ADC 输入通道
44	34	22		PC9	GPIO
				SEG18	LCD 驱动 SEG 端
				SPI2_MISO	SPI 数据线
				ADC_IN0	ADC 输入通道
45	35	23		PC10	GPIO
				WKUP5	外部唤醒引脚
				SEG19	LCD 驱动 SEG 端
				SPI2_MOSI	SPI 数据线
				ADC_IN1	ADC 输入通道
46				PC11	GPIO
				SEG20	LCD 驱动 SEG 端
				U7816CLK	7816 接口时钟
				GPT0_CH3	通用定时器外部通道
47				PC12	GPIO
				SEG21	LCD 驱动 SEG 端
				U7816IO	7816 接口数据
				GPT0_CH4	通用定时器外部通道
48	36	24	9	PD9	GPIO
				XT32KO	32768Hz 晶振输出脚
49	37	25	10	PD10	GPIO
				XT32KI	32768Hz 晶振输入脚
50	38	26	11	VDD15	LDO 输出，外接 100nF 电容到地
51	39	27	12	VSS	地
52	40	28	13	VDD	电源
53				VSSA	模拟地
54				VDDA	模拟电源

Pin Number				Pin Function	Descriptions
LQFP64	LQFP48	QFN32	TSSOP20		
55	41	29		PD11	GPIO
				WKUP6	外部唤醒引脚
				FOUT0	时钟频率输出
				ATIM_BRK1	高级定时器刹车输入
				ADC_IN2	ADC 输入通道
56	42	30		PD0	GPIO
				SEG22	LCD 驱动 SEG 端
				UART5_RX	UART 接收
				ADC_IN3	ADC 输入通道
57	43	31		PD1	GPIO
				SEG23	LCD 驱动 SEG 端
				UART5_TX	UART 发送
				ADC_IN4	ADC 输入通道
58				PD2	GPIO
				SEG24	LCD 驱动 SEG 端
				SPI1_SSN	SPI 片选
				ADC_IN5	ADC 输入通道
59				PD3	GPIO
				SEG25	LCD 驱动 SEG 端
				SPI1_SCK	SPI 时钟
60				PD4	GPIO
				SEG26	LCD 驱动 SEG 端
				SPI1_MISO	SPI 数据
				COMP1_INP1	比较器输入
61				PD5	GPIO
				SEG27	LCD 驱动 SEG 端
				SPI1_MOSI	SPI 数据
				COMP1_INP2	比较器输入
62	44			PD6	GPIO
				WKUP7	外部唤醒引脚
				ANATST	模拟测试通道
				ATIM_BRK2	高级定时器刹车输入
63	45	32		PD7	GPIO
				SWCLK	SWD 接口时钟
64	46	1		PD8	GPIO
				SWIO	SWD 接口数据

表 2-2 FM33LC0xxN 引脚列表

2.1.9 功能引脚分布

功能	引脚	LQFP64 编号	备注
ADC 输入通道	PC9	44	ADC_IN0
	PC10	45	ADC_IN1
	PD11	55	ADC_IN2

功能	引脚	LQFP64 编号	备注
	PD0	56	ADC_IN3
	PD1	57	ADC_IN4
	PD2	58	ADC_IN5
	PA13	2	ADC_IN6
	PA14	3	ADC_IN7
	PC7	42	ADC_IN8
	PC8	43	ADC_IN9
	PA15	4	ADC_IN10
	PC6	41	ADC_IN11
OPA	PB10	26	OPA1_INN1
	PB11	27	OPA1_INP1
	PA6	11	OPA1_INN2
	PA7	12	OPA1_INP2
	PB13	29	OPA2_INN1
	PB14	30	OPA2_INP1
	PC0	35	OPA2_INN2
	PC1	36	OPA2_INP2
	PC4	39	OPA1_OUT
	PC5	40	OPA2_OUT
UART 接收	PA13, PA2	2, 7	UART0_RX
	PB13, PC2	29, 37	UART1_RX
	PA0, PB2	5, 18	UART4_RX
	PC4, PD0	39, 56	UART5_RX
UART 发送	PA14, PA3	3, 8	UART0_TX
	PB14, PC3	30, 38	UART1_TX
	PA1, PB3	6, 19	UART4_TX
	PC5, PD1	40, 57	UART5_TX
LPUART	PA13, PA2	2, 7	LPUART0_RX
	PA14, PA3	3, 8	LPUART0_TX
	PB13, PC2	29, 37	LPUART1_RX
	PB14, PC3	30, 38	LPUART1_TX
SPI	PB8, PD2	24, 58	SPI1_SSN
	PB9, PD3	25, 59	SPI1_SCK
	PB10, PD4	26, 60	SPI1_MISO
	PB11, PD5	27, 61	SPI1_MOSI
	PC7	42	SPI2_SSN
	PC8	43	SPI2_SCK
	PC9	44	SPI2_MISO
	PC10	45	SPI2_MOSI
I2C	PA11	16	SCL
	PA12	17	SDA
7816	PC11	46	U7816CLK
	PC12	47	U7816IO
高级定时器	PB4	20	ATIM_CH1
	PB5	21	ATIM_CH2
	PB6	22	ATIM_CH3

功能	引脚	LQFP64 编号	备注
	PB7	23	ATIM_CH4
	PB2	18	ATIM_CH1N
	PB3	19	ATIM_CH2N
	PB8	24	ATIM_CH3N
	PB12	28	ATIM_ETR
	PD11	55	ATIM_BRK1
	PD6	62	ATIM_BRK2
通用定时器	PB10	26	GPT0_CH1
	PB11	27	GPT0_CH2
	PC11	46	GPT0_CH3
	PC12	47	GPT0_CH4
	PB9	25	GPT0_ETR
	PC0	35	GPT1_CH1
	PC1	36	GPT1_CH2
	PA4	9	GPT1_CH3
	PA5	10	GPT1_CH4
	PC6	41	GPT1_ETR
低功耗定时器	PA8	13	LPT32_CH1
	PA9	14	LPT32_CH2
	PA10	15	LPT32_ETR
WKUP 唤醒	PA15	4	WKUP0
	PA10	15	WKUP1
	PB2	18	WKUP2
	PB12	28	WKUP3
	PC6	41	WKUP4
	PC10	45	WKUP5
	PD11	55	WKUP6
	PD6	62	WKUP7
比较器输入	PA10	15	COMP1_INN1
	PD4	60	COMP1_INP1
	PD5	61	COMP1_INP2
	PA4	9	COMP2_INN1
	PA5	10	COMP2_INN2
	PA8	13	COMP2_INP1
	PA9	14	COMP2_INP2
USB FS	DP	31	DP
	DM	32	DM
外部电源检测	PA15	4	SVS
Debug	PD7	63	SWCLK
	PD8	64	SWIO

表 2-3 引脚功能分布表

2.1.10 封装尺寸图

2.1.10.1 LQFP64

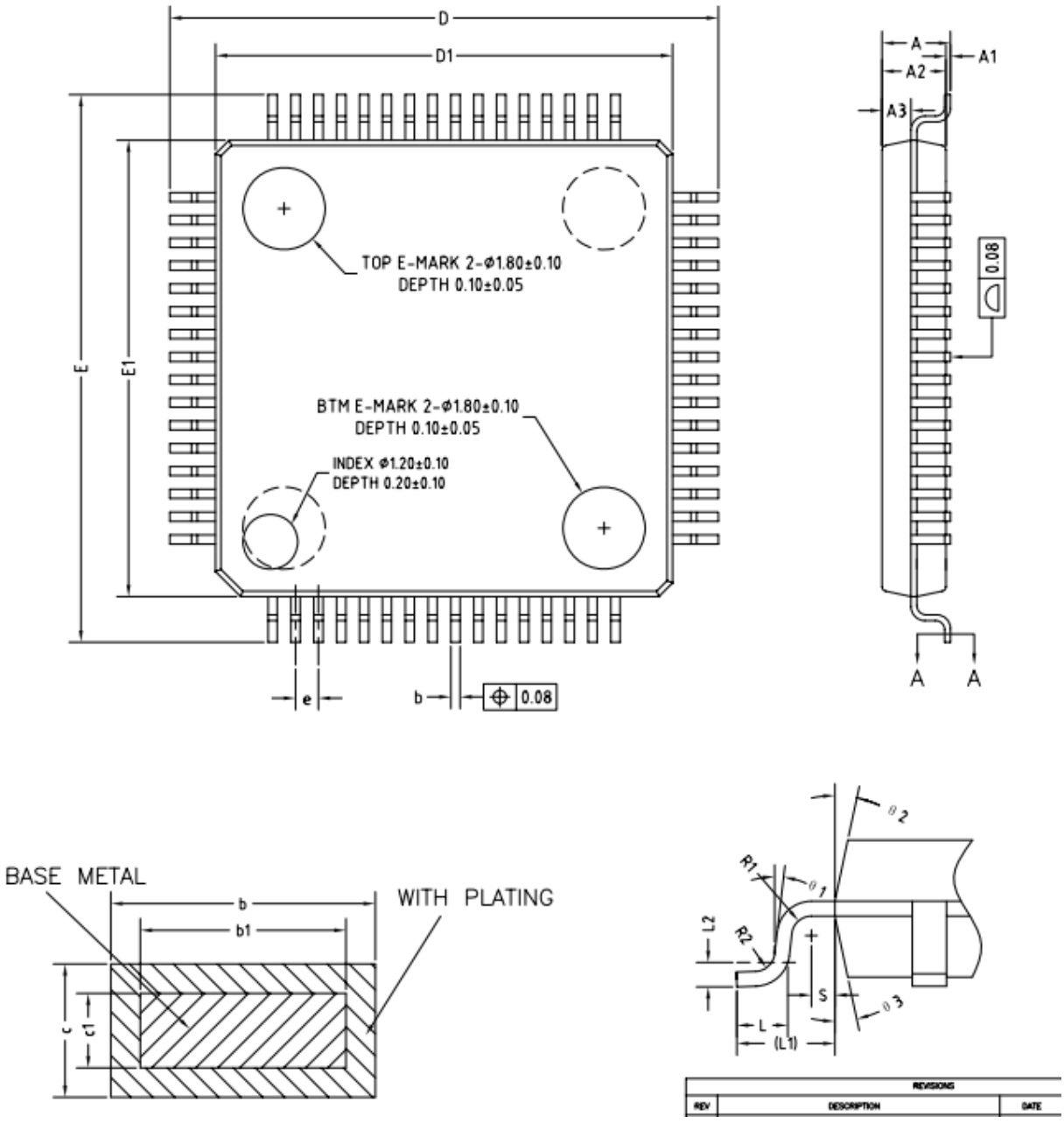


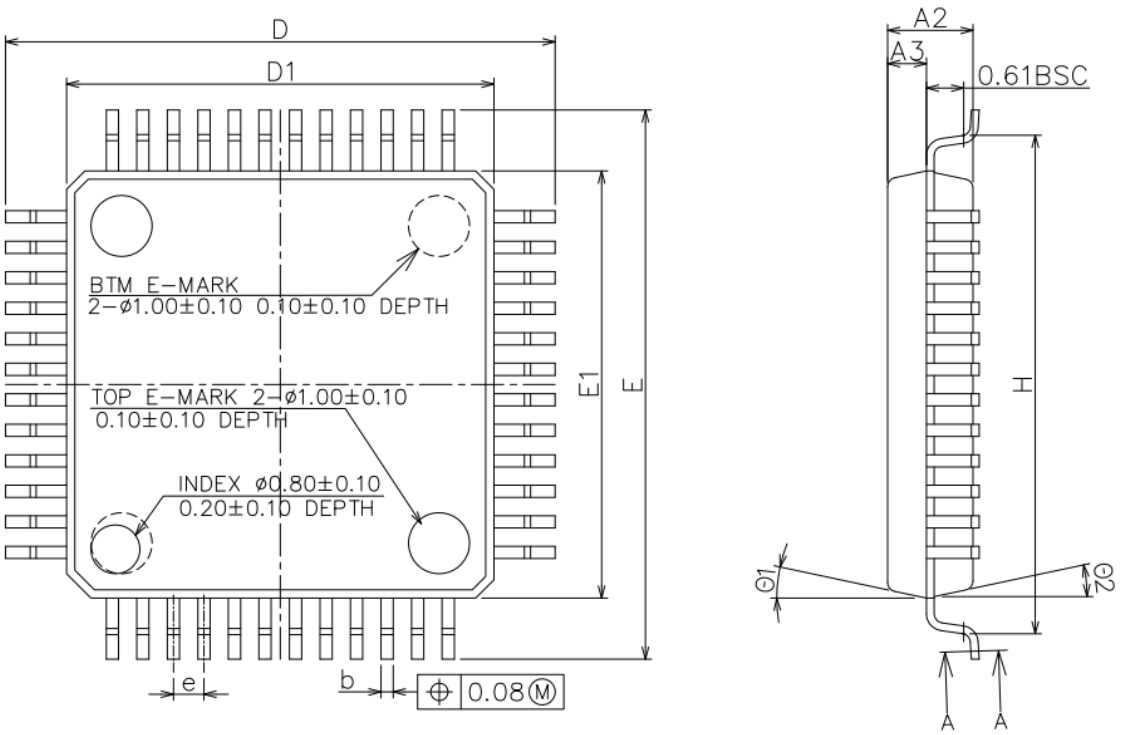
图 2-7 LQFP64 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134

Symbol	MIN	NOM	MAX
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220WMMD-4.

2.1.10.2 LQFP48



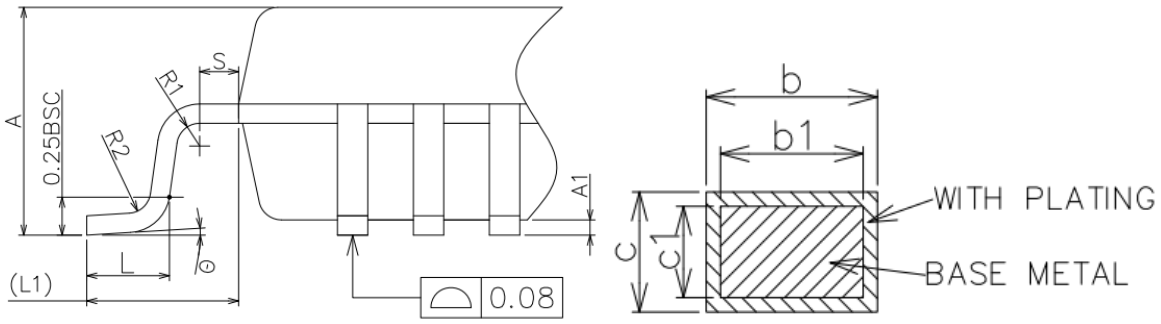


图 2-8LQFP48 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026BDD.

2.1.10.3 QFN32

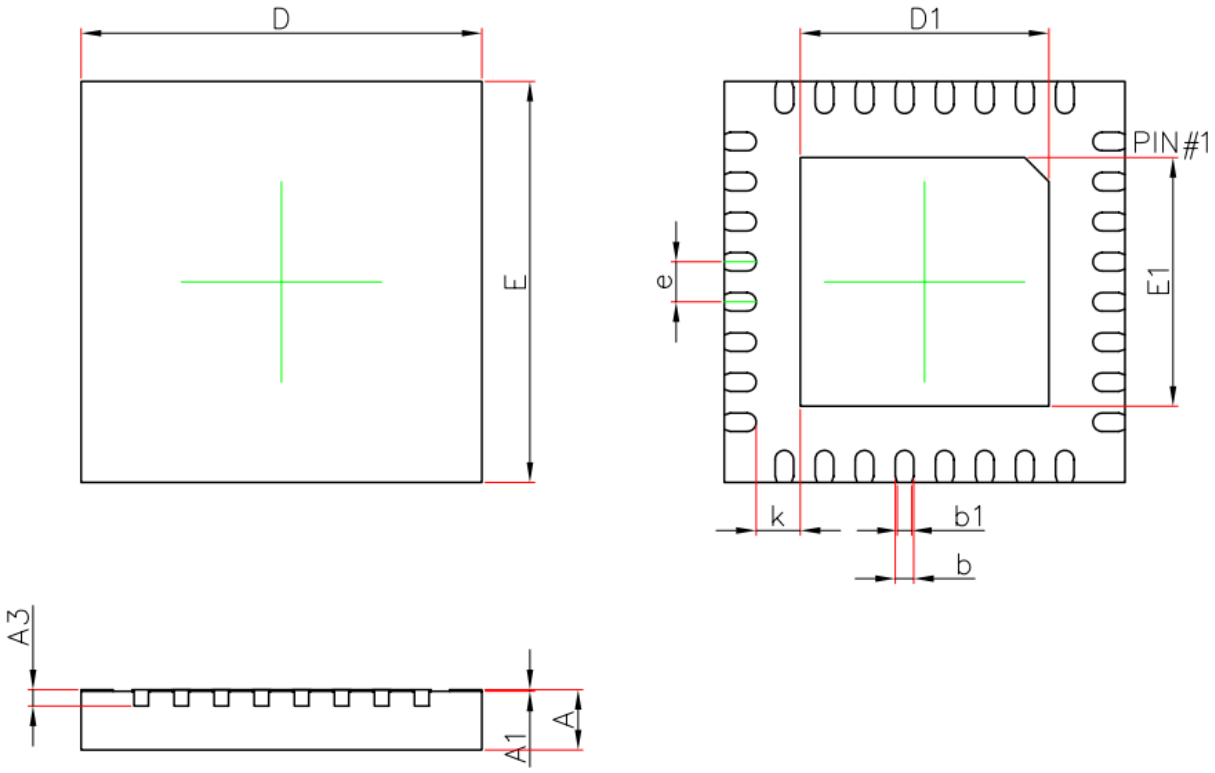


图 2-9QFN32 封装尺寸图

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700	0.800	0.028	0.031
A1	0.000	0.050	0.000	0.002
A3	0.203 REF.		0.008 REF.	
b	0.180	0.300	0.007	0.012
b1	0.130	0.230	0.005	0.009
D	4.900	5.100	0.193	0.201
D1	3.000	3.200	0.118	0.126
E	4.900	5.100	0.193	0.201
E1	3.000	3.200	0.118	0.126
e	0.500 BSC.		0.020 BSC.	
k	0.550 REF.		0.022 REF.	
L	0.324	0.476	0.013	0.019

NOTE:ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220WMMD-4.

3 电参数

3.1 参数说明

电参数章节中所罗列的典型值是大量样本数据分布的中心值，常温下的最大值是由芯片量产测试所保证的。高低温下的电参数通常基于特征参数提取，由大量样本数据分布的平均值加减 3sigma 得到。

3.2 参数测试条件

3.2.1 供电方案

芯片测试时采用下图所示的电源供电方案。

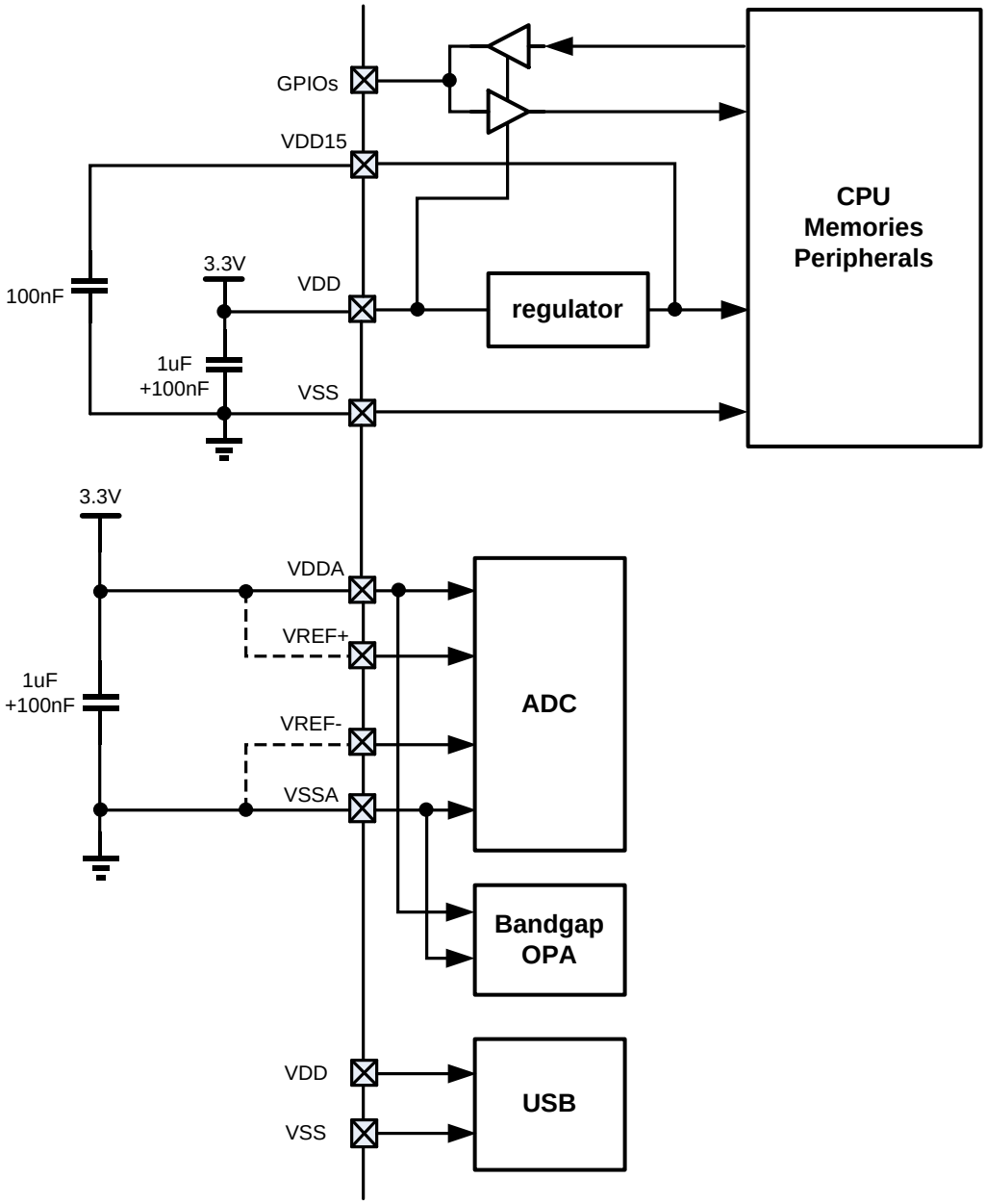


图 3-1FM33LC0x6U 供电方案

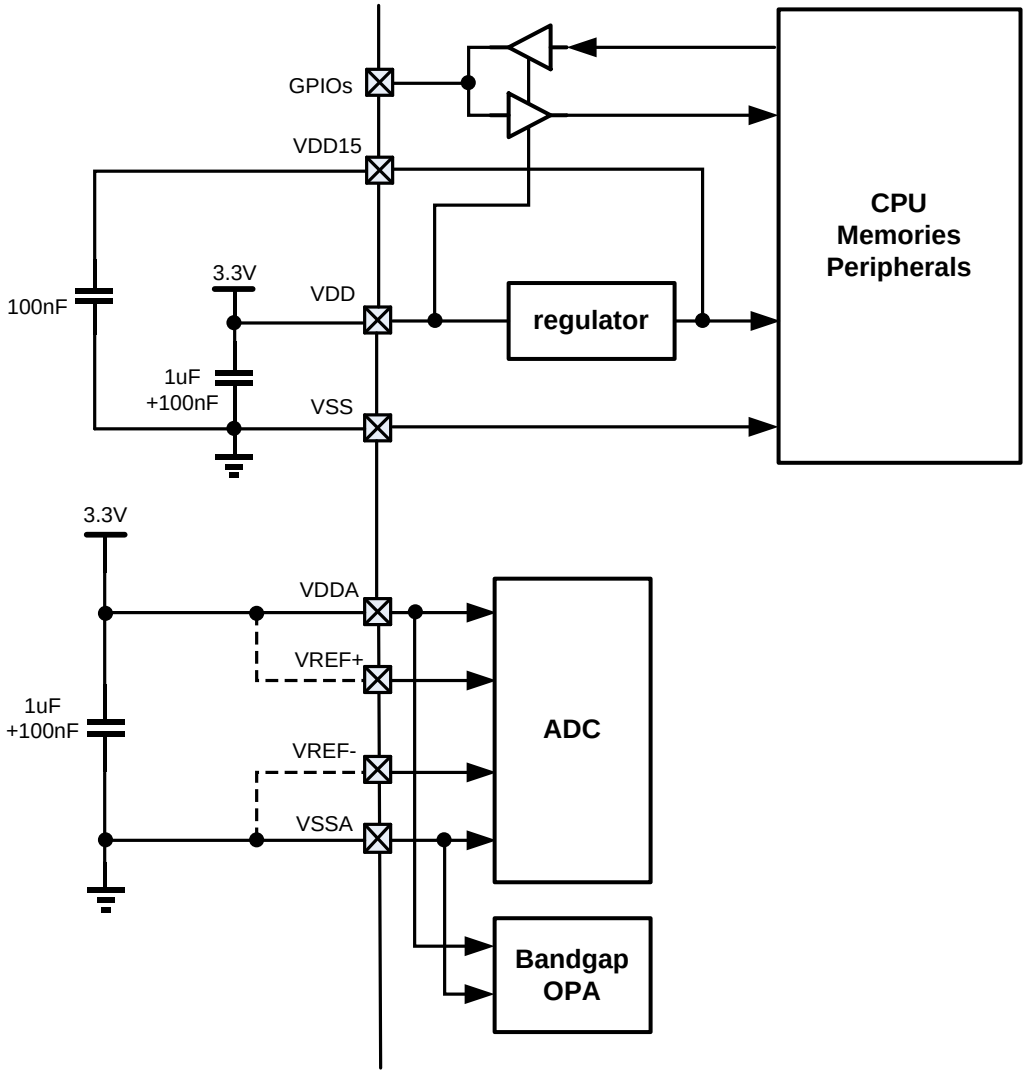


图 3-2FM33LC0x6N 供电方案

3.3 极限参数

对芯片施加的电压、电流等超过极限参数表定义的最大范围时，可能导致芯片不可恢复的损坏；短时间超过极限参数范围则可能影响芯片的可靠性和工作寿命。

Symbol	Parameter		min	max	unit
V_{DD}-V_{SS}	电源电压 （包含 VDD、VDDA）		-0.3	6.5	V
V_{PIN}	管脚电压		V_{SS}-0.3	6.5	V
$\|\triangle V_{DD}\|$	VDD 和 VDDA 之间的压差(1)		-	50	mV
$\|\triangle V_{SS}\|$	所有地引脚之间的压差		-	50	mV
T_A	工作温度		-40	85	℃
T_{STG}	存储温度		-55	150	℃
HBM	ESD HBM 模式 TA=25℃ 测试标准符合 JEDEC JS-001	除 PA11, PA12, PB12 以外的引脚	-	+/-8000	V
		PA11, PA12, PB12	-	+/-5000	
CDM	ESD CDM 模式 TA=25℃ 测试标准符合 JEDEC JS-002		-	+/-1000	V
LU	IO Latchup -(0.5VDD) < VI < (1.5VDD) TA=25℃ 测试标准符合 JESD78E	$I_{trigger}$	-	+/-125	mA
		V_{supply}	-	8.25	V
ΣI_{VDD}	向芯片 VDD 流入的最大电流(source)		-	90	mA
ΣI_{VSS}	从芯片 VSS 流出的最大电流(sink)		-	70	mA
ΣI_{IO}	所有 IO sink 的最大总和电流		-	90	mA
	所有 IO source 的最大总和电流		-	70	mA

表 3-1FM33LC0XX 极限参数

注：

1. 推荐使用相同的电压源对 VDD 和 VDDA 供电。

3.4 性能参数

3.4.1 典型工作条件

Symbol	Parameter	Conditions	min	max	unit
f_{HCLK}	AHB 时钟频率	-	0	64	Mhz
f_{PCLK}	APB 时钟频率	-	0	64	
VDD	典型工作电压范围[1]（使用 USB 功能）	BOR 使能	1.8	3.6	V
		BOR 关闭	1.35	3.6	
	典型工作电压范围[1]（不使用 USB 功能）	BOR 使能	1.8	5.5	V
		BOR 关闭	1.35	5.5	
VDDA	模拟电路工作电压范围（使用 USB 功能）	必须满足 VDDA=VDD	1.8	3.6	V
	模拟电路工作电压范围（不使用 USB 功能）	必须满足 VDDA=VDD	1.8	5.5	V
T_J	结温	T_A=-40~+85C	-40	105	° C

表 3-2FM33LC0XX 典型工作条件

注：

[1] 低温（-40C）上电时，上电复位电压阈值会升高，为了保证可靠复位，电源必须上升到 2.0V 以上才能开始工作。一旦上电过程完成，芯片的最低工作电压由 BOR 下电阈值或 PDR 下电阈值决定。

[2] USB 限制芯片必须工作在不超过 3.6V 电源下。如果不使用 USB，则电源范围最高支持到 5.5V。

3.4.2 功耗参数

芯片出厂时的功耗参数在环境温度下测试，高低温电流参数来自于特征参数提取。

测量功耗参数时，MCU 被配置为如下条件：

- 所有功能引脚被配置为 GPIO 模式，并且关闭输入和输出使能，避免引脚浮空漏电
- 除了特别声明的以外，所有外设被关闭，并停止工作时钟
- 常温下的最大功耗数据代表出厂时的测试上限标准
- 常温下的典型功耗数据代表大量样本分布的中心值
- 除非特别声明，所有功耗数据在 VDD=VDDA=3.3V 的条件下测试获得

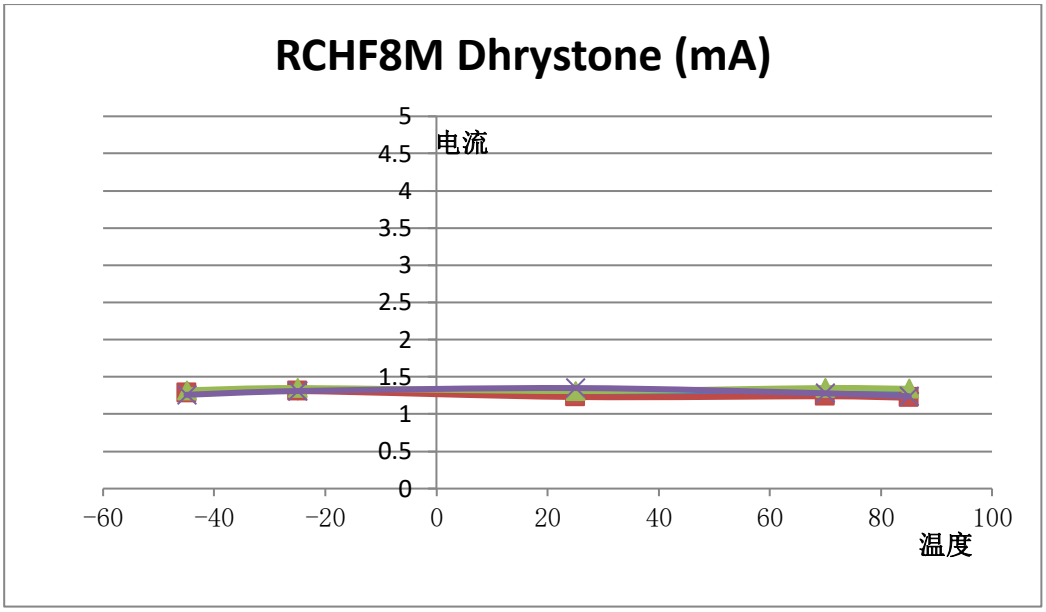
3.4.2.1 Active 模式功耗

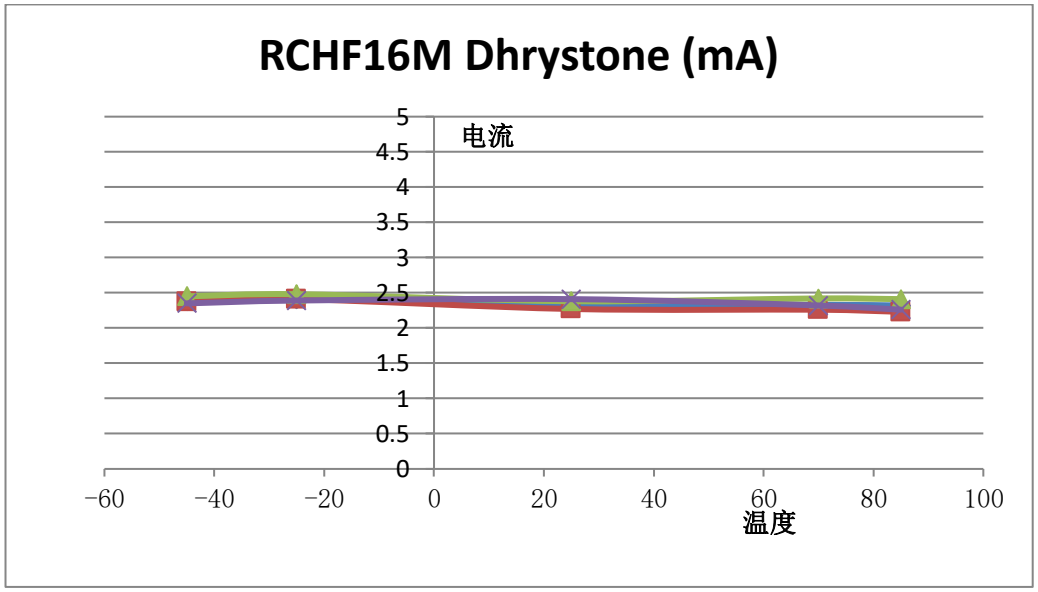
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，Dhrystone	f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	2.5	-	mA
			TA=85℃	-	2.5	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	3.6	-	mA
			TA=85℃	-	3.5	-	
		f_{AHB}=48MHz PLL on Flash 1 wait	TA=25℃	-	5.7	-	mA
			TA=85℃	-	5.65	-	
		f_{AHB}=64MHz PLL on Flash 2 wait	TA=25℃	-	6	-	mA
			TA=85℃	-	5.95	-	
		f_{AHB}=4MHz (RCMF) PLL off Flash 0 wait	TA=25℃	-	0.6	-	mA
			TA=85℃	-	0.6	-	

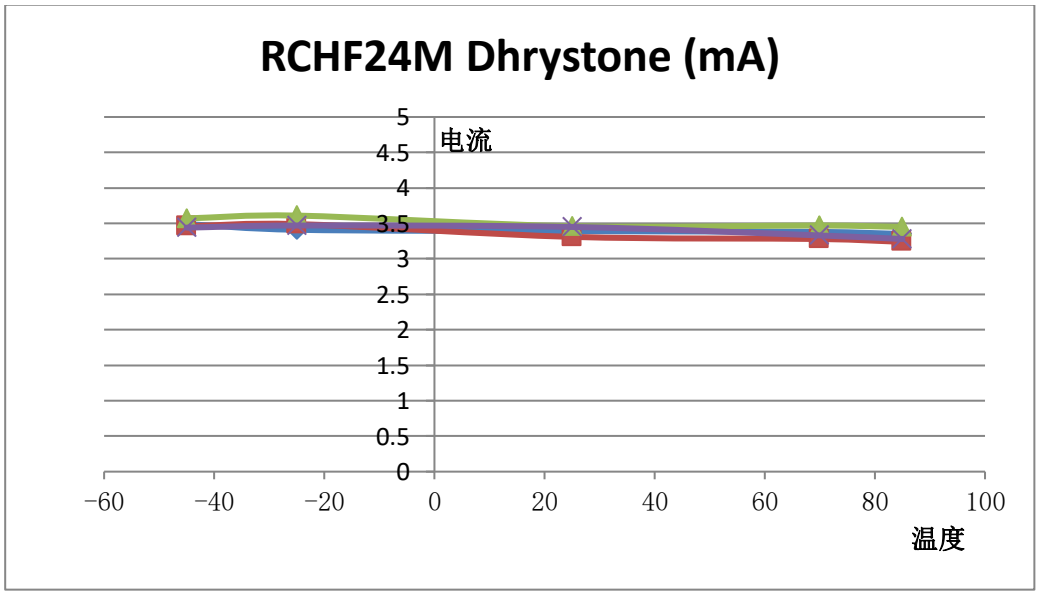
表 3-3ACTIVE 电流参数

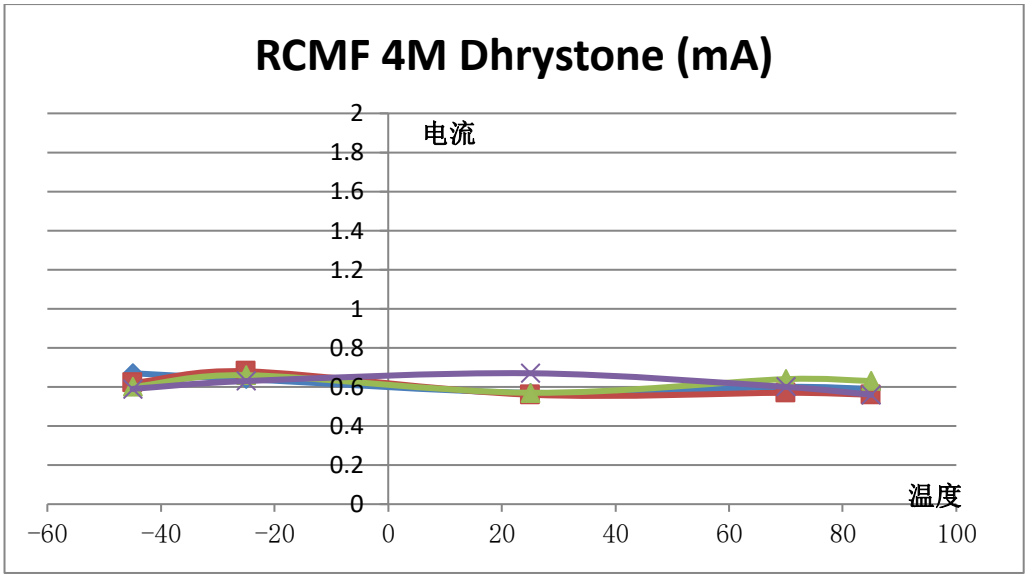
注：上表参数基于特征参数提取，不包含在量产测试中

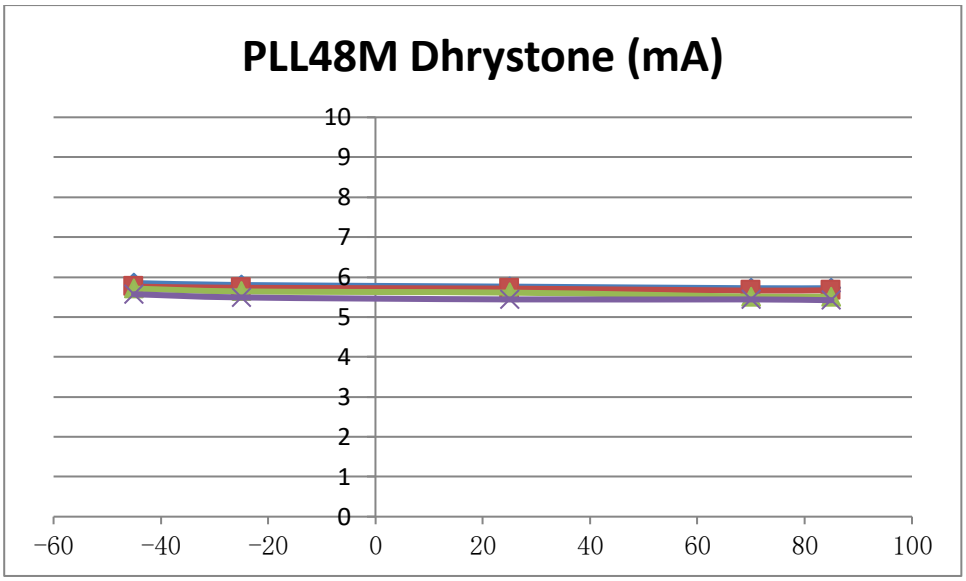
典型功耗-温度曲线（基于特征参数提取，仅供设计参考）

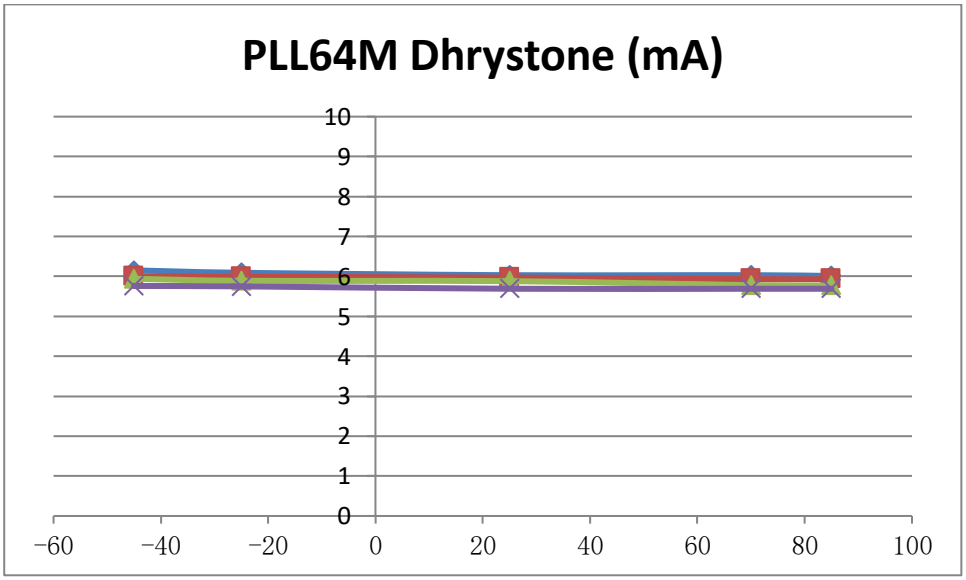

RCHF8M Dhrystone (mA)
电流
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-60
-40
-20
0
20
40
60
80
100
温度


RCHF16M Dhrystone (mA)
电流
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-60
-40
-20
0
20
40
60
80
100
温度


RCHF24M Dhrystone (mA)
电流
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-60
-40
-20
0
20
40
60
80
100
温度


RCMF 4M Dhrystone (mA)
电流
温度
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-60
-40
-20
0
20
40
60
80
100


PLL48M Dhrystone (mA)
10
9
8
7
6
5
4
3
2
1
0
-60
-40
-20
0
20
40
60
80
100

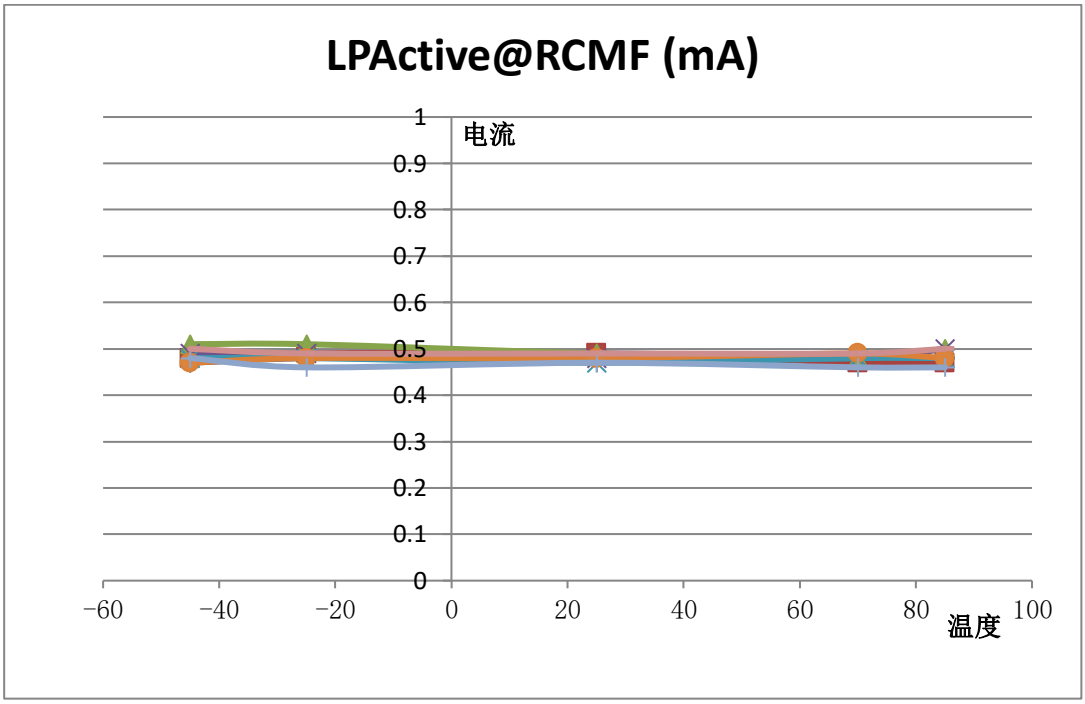

PLL64M Dhrystone (mA)
10
9
8
7
6
5
4
3
2
1
0
-60
-40
-20
0
20
40
60
80
100

3.4.2.2 LP Active 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	LP Active 模式下的功耗，CPU 从 Flash 取指，Dhrystone	f_{AHB}=4MHz (RCMF) PLL, RCHF off Flash 0 wait	TA=25℃	-	0.5	-	mA
			TA=85℃	-	0.5	-	

表 3-4LP ACTIVE 电流参数

典型功耗-温度曲线（基于特征参数提取，仅供设计参考）



3.4.2.3 LP RUN 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{LPRUN}	LP RUN 模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=32768Hz (XTLF) PLL, RCHF, RCMF off Flash 0 wait	TA=25℃		28		uA
			TA=85℃		30		

表 3-5LP RUN 电流参数

3.4.2.4 SLEEP 模式功耗

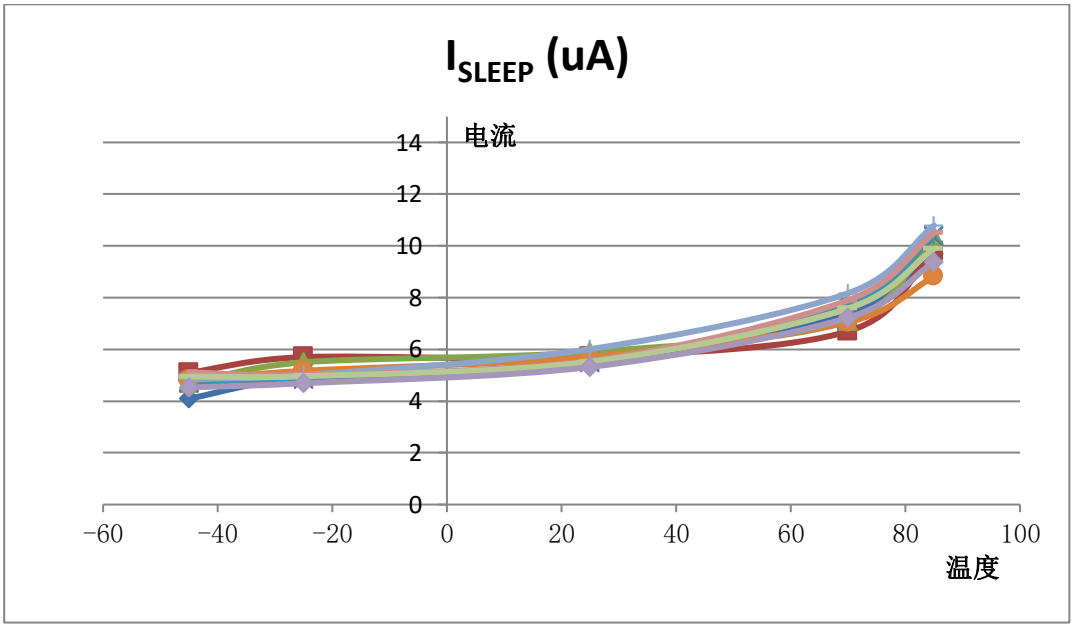
符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
I_{sleep1}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示关闭	TA=25℃	-	5.1	10	uA
			TA=85℃[1]	-	10	-	uA

表 3-6SLEEP 电流参数

[注 1]：此项指标基于特征参数提取，不包含在量产测试中

典型功耗-温度曲线（基于特征参数提取，仅供设计参考）



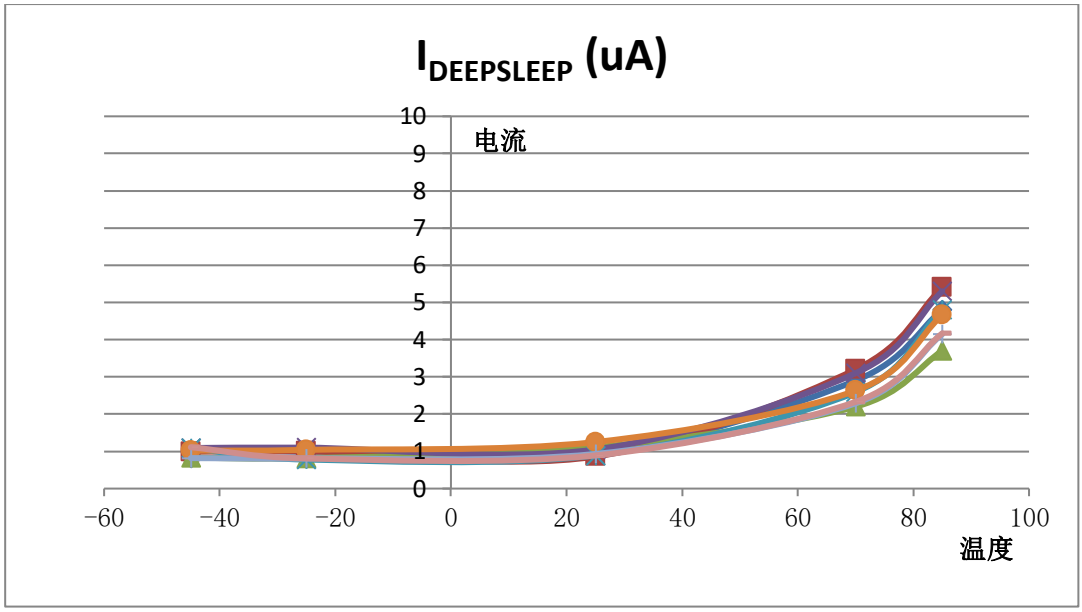
3.4.2.5 DEEPSLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$I_{dpsleep}$	DeepSleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时，XTLF 偏置电流 200nA CPU、RAM、外设数据保持 LCD 显示关闭	TA=25℃	-	1	2	uA
			TA=85℃[1]	-	5	-	uA

表 3-7DEEPSLEEP 电流参数

[注 1]：此项指标基于特征参数提取，不包含在量产测试中

典型功耗-温度曲线（基于特征参数提取，仅供设计参考）


IDEEPSLEEP (uA)
电流
10
9
8
7
6
5
4
3
2
1
0
-60
-40
-20
0
20
40
60
80
100
温度

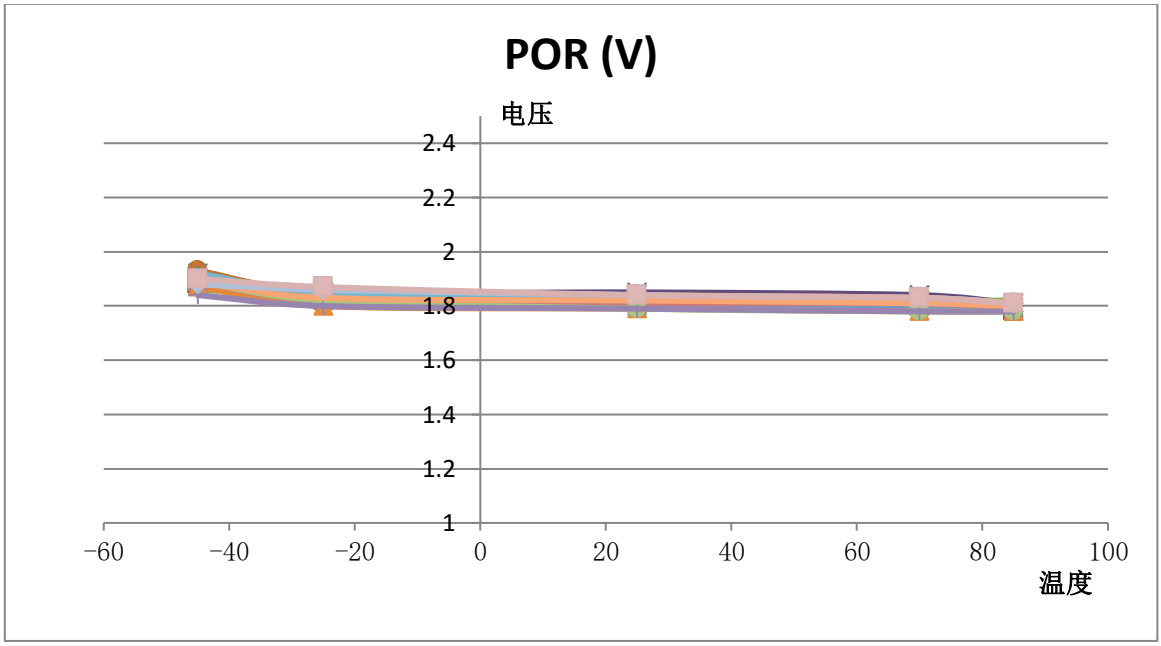
3.4.3 复位和电源监控

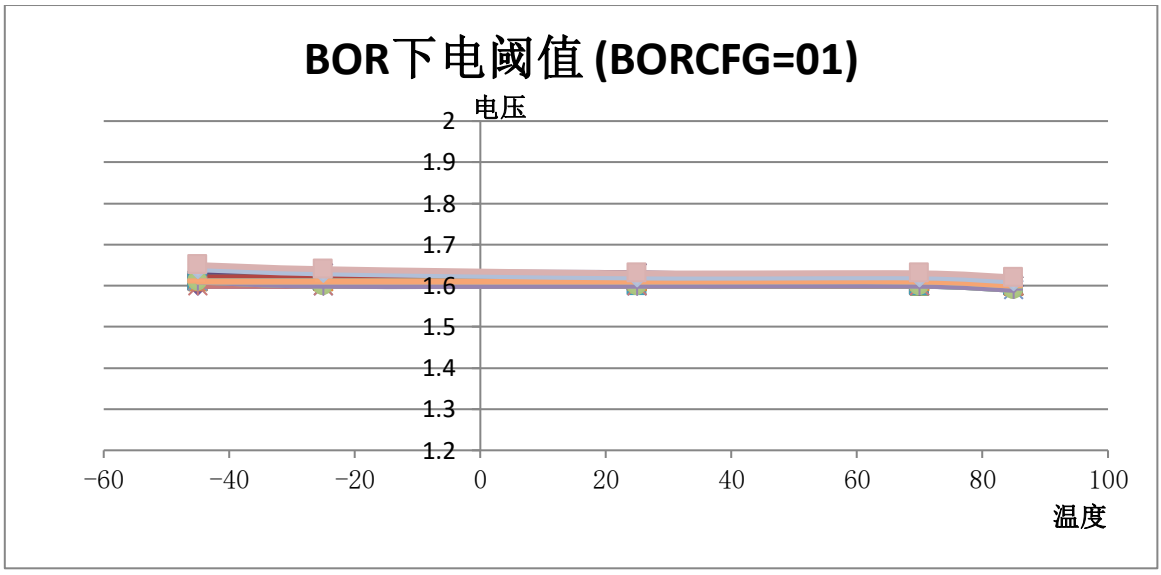
芯片的复位和电源监控参数如下表。

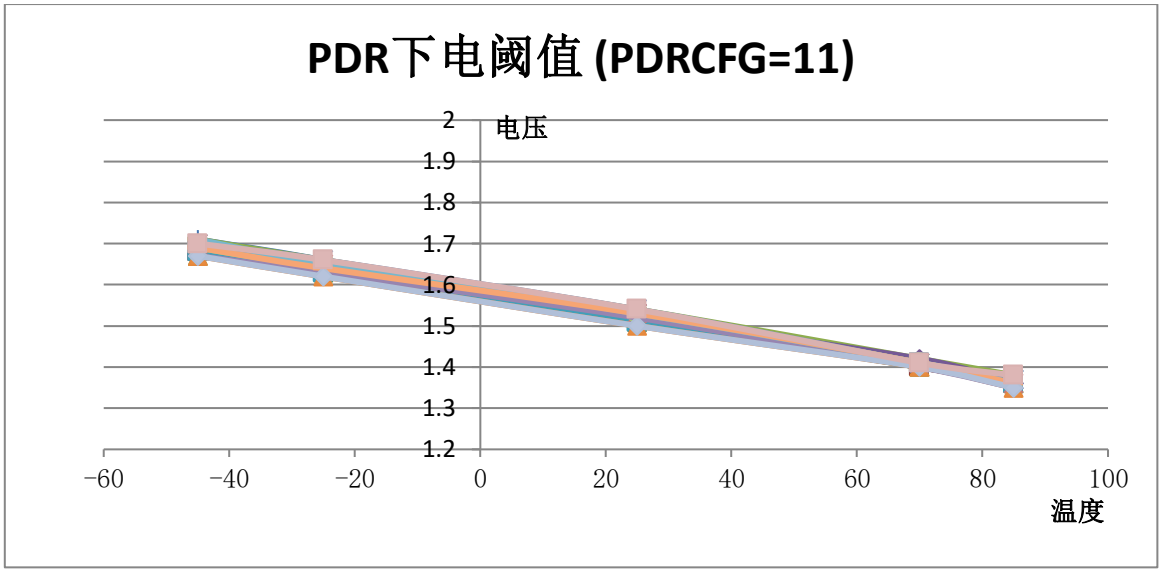
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
t_{VDD}	电源上升速度			2		∞	us/V
	电源下降速度	PDR		200		∞	us/V
		BOR		600		∞	us/V
T_{reset_delay}	上电复位延迟时间				0.5		ms
T_{pdr_filter}	下电复位滤波时间				4		us
V_{POR}	上电复位电压			-	1.8	2.0	V
V_{BOR}	下电复位电压	BORCFG==2'b01		1.58	1.61	1.65	V
V_{PDR}	低功耗下电复位电压	PDRCFG==2'b11		1.34 (T_A=85C)	1.52	1.72 (T_A=-40C)	V
I_{BOR}	BOR 功耗				1.2		uA
I_{PDR}	PDR 功耗				55		nA
V_{SVD}	电压监测阈值电平	SVD[3:0]=0000	Fall		1.800		V
			Rise		1.900		
		SVD[3:0]=0001	Fall		2.014		V
			Rise		2.114		
		SVD[3:0]=0010	Fall		2.229		V
			Rise		2.329		
		SVD[3:0]=0011	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0100	Fall		2.657		V
			Rise		2.757		
		SVD[3:0]=0101	Fall		2.871		V
			Rise		2.971		
		SVD[3:0]=0110	Fall		3.086		V
			Rise		3.186		
		SVD[3:0]=0111	Fall		3.300		V
			Rise		3.400		
		SVD[3:0]=1000	Fall		3.514		V
			Rise		3.614		
		SVD[3:0]=1001	Fall		3.729		V
			Rise		3.829		
		SVD[3:0]=1010	Fall		3.943		V
			Rise		4.043		
		SVD[3:0]=1011	Fall		4.157		V
			Rise		4.257		
		SVD[3:0]=1100	Fall		4.371		V
			Rise		4.471		
		SVD[3:0]=1101	Fall		4.586		V
			Rise		4.686		
		SVD[3:0]=1110	Fall		4.800		V
			Rise		4.900		
		SVD[3:0]=1111	Fall		TBD		V
			Rise		TBD		

表 3-8 复位和电源监控参数

上电复位释放阈值-温度曲线（基于特征参数提取，仅供设计参考）







3.4.4 高精度基准源

芯片内建高精度基准电压源，为 ADC 和 OPA 提供高精度、高稳定性的参考电压。

芯片出厂时，复旦微电子会在特定的电源电压和温度下，使用片内 ADC 采样基准源输出，并将转换结果保存在芯片的 NVR 中，用户应用中可以将这个转换值作为参考基准使用。

符号	参数说明	总线地址
REF_CAL	ADC 对 VREF 输出的转换值 测试条件： T_A=30±1℃ VDDA=3V±10mV	0x1FFF_FB08 低半字保存 ADC 对 VREF 的转换值，高半字为低半字的反码校验
REF_RAW	VREF 输出的电压值 测试条件： T_A=30±1℃ VDDA=3V±10mV	0x1FFF_FB0C 低半字保存 VREF1p2 的实际电压值，高半字为低半字的反码校验 REF_RAW=VREF1p2 * 10000

高精度基准源主要参数：

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
V_{REF}	基准源输出电压[1]	-40℃≤TA≤85℃		1.19	1.202	1.213	V
T_{setup}	内部基准源建立时间	-		-	1	1.4	ms
V_{VREF_MEAS}	出厂时测量转换 VREF 的 VDDA 电压	-		2.99	3	3.01	V
T_{coeff}	内部基准源温度系数	-40℃≤TA≤85℃ VDDA=3.3V			25	85	ppm/℃
T_{S_VREF}	ADC 测量 VREF 时的采样时间	预先使能 VREF Buffer		10			us
T_{ADC_BUF}	VREF buffer 使能后输出完全建立的延迟[1]	VDDA=3.3V ADC 采样值稳定到 1LSB				100	us
I_{REF}	基准源工作电流[2]	TA = 25℃	PTAT_EN=0		1.8		uA

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
		VDDA=3.3V	PTAT_EN=1		2.6		

表 3-9 高精度基准源参数

注：

[1] 基于特征参数提取，不包含在量产测试中

[2] 基于电路仿真

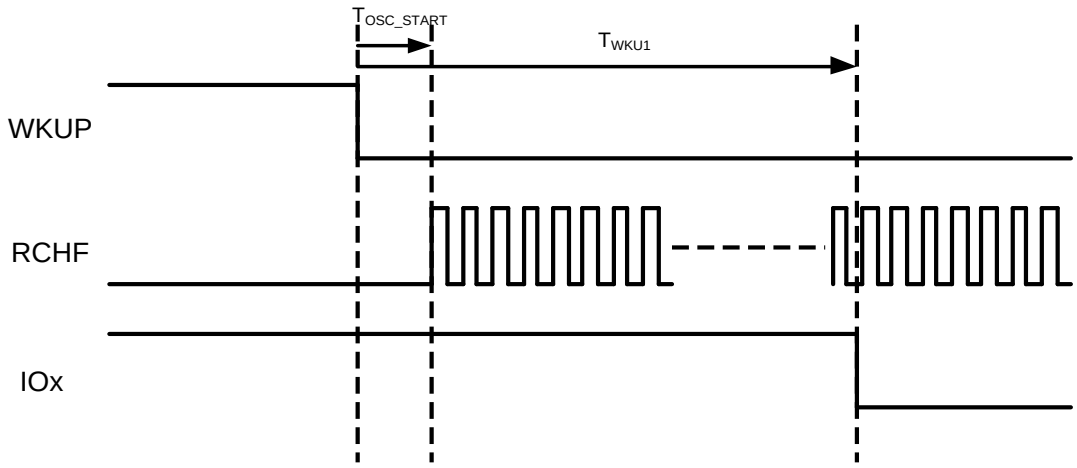
3.4.5 低功耗模式唤醒时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
T_{WKU1}	Sleep/DeepSleep 唤醒时间[1]	使用 WKUP 引脚唤醒，PRIMASK=1 禁止中断；CPU 唤醒后执行程序翻转某个 IO 输出，测量 WKUP 信号边沿到 IO 输出翻转之间的时间 F_{SYSCLK}=8Mhz	-	5.5	-	us
T_{WKU2}	LPRUN 模式唤醒时间		-	0	-	us

表 3-10 唤醒时间参数

[1] 基于特征参数提取

典型唤醒事件波形图，仅供设计参考



上图中 T_{OSC_START} 表示唤醒事件到来后 RCHF 环振起振时间，典型值小于 3us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 5.5us。

如果没有通过 PRIMASK 屏蔽中断，则唤醒事件将使 CPU 进入中断服务程序。CPU 进入中断服务程序的过程将额外引入延迟时间。

注意：以上时间评估使用 RCHF 8Mhz 为唤醒后的工作时钟，如果唤醒后选择 16Mhz 或 24Mhz 频率，则唤醒时间相应缩短。

3.4.6 外部时钟源特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{XTLF}	XTLF 振荡频率	外接 32768Hz 晶体		32768		Hz
T_{start}	XTLF 起振时间	外接 32768Hz 晶体 C_{load}=12pF XTLFIPW==3'b000		1	3	s

表 3-11 低频晶体振荡器参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率	VDD=3.3V	4	-	24	MHz
R_{fb}	反馈电阻	-	-	200	-	KΩ
VDD_{rise}	XTHF 最低工作电压（上电）	8MHz, XTHFCFG=000	2.0	-	-	V
		12MHz, XTHFCFG=000	2.1	-	-	
		24MHz, XTHFCFG=111	2.4	-	-	
		32MHz, XTHFCFG=111	2.85	-	-	
VDD_{fall}	XTHF 最低工作电压（下电）	8MHz, XTHFCFG=000	1.6	-	-	V
		12MHz, XTHFCFG=000	1.85	-	-	
		24MHz, XTHFCFG=111	2.7	-	-	
		32MHz, XTHFCFG=111	2.65	-	-	
IDD	XTHF 工作电流	8MHz, XTHFCFG=000	-	145	-	uA
		24MHz, XTHFCFG=111	-	800	-	
		32MHz, XTHFCFG=111	-	1350	-	
T_{start1}	XTHF 8M 起振时间	VDD=3.3V XTHFCFG=000	-	4.5	-	ms
		VDD=3.3V, XTHFCFG=111	-	0.8	-	ms
T_{start2}	XTHF 24M 起振时间	VDD=3.3V, XTHFCFG=111	-	1.2	-	ms
C_L	负载电容	-	5	-	25	pF

表 3-12 高频晶体振荡器参数

3.4.7 内部时钟源特性

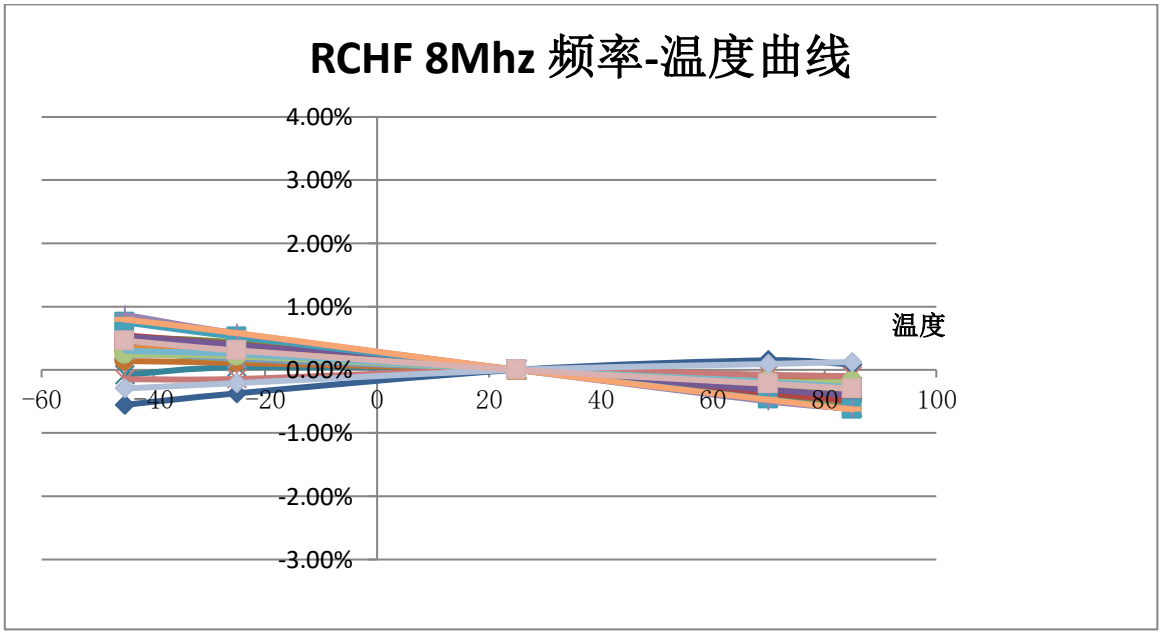
内部高频 RC 振荡器

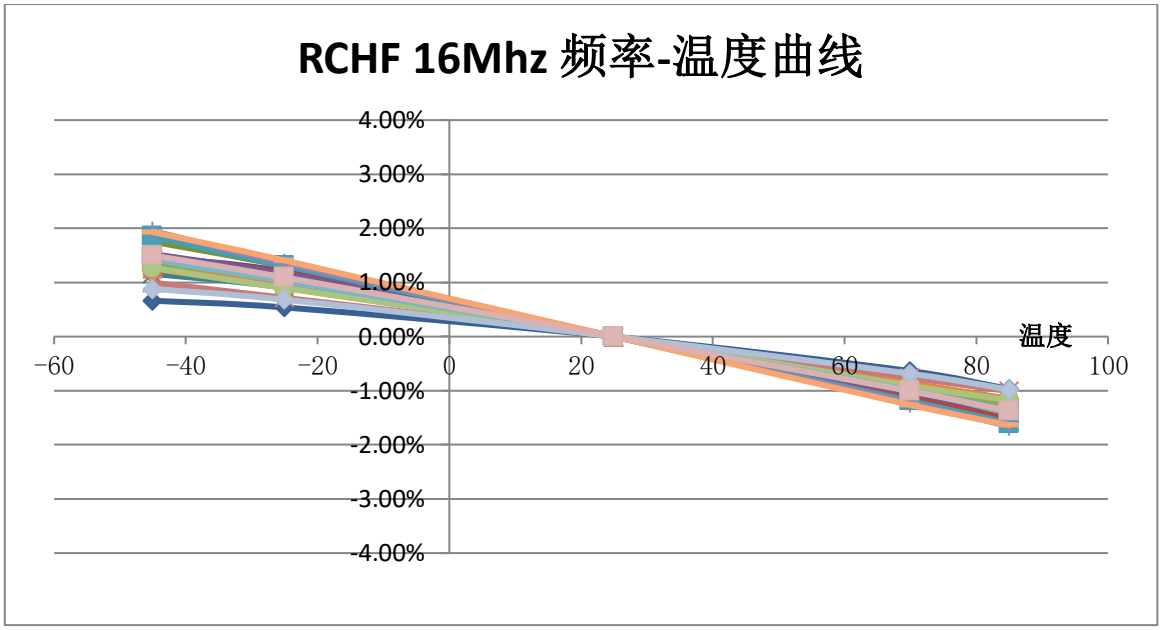
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
f_{RCHF}[1]	RCHF 振荡频率	VDD=1.8~5.5V	FSEL==2'b00	7.92	8	8.08	MHz
			FSEL==2'b01	15.84	16	16.16	
			FSEL==2'b10	23.76	24	24.24	
			FSEL==2'b11		RFU		
ACC_{RCHF}[2]	全温区 RCHF 变化范围	VDD=1.8~5.5V	FSEL==2'b00 T=-40~+85℃	-1	-	1	%
			FSEL==2'b01 T=-40~+85℃	-2	-	2	%
			FSEL==2'b10 T=-40~+85℃	-3	-	3	%

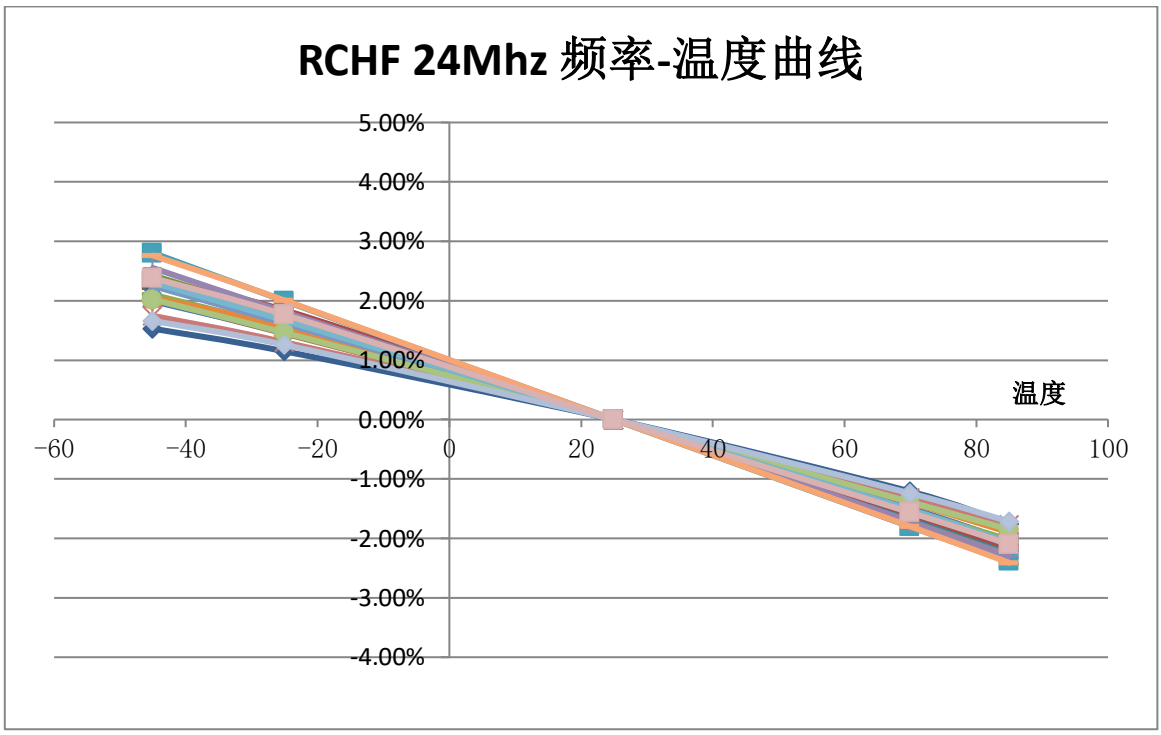
表 3-13 内部高频 RC 振荡器参数

[注1]：此项指标由量产测试保证
[注2]：此项指标基于特征参数提取

典型 RCHF 各档位频率-温度变化曲线（基于特征参数提取，仅供设计参考）







注：通过结合温度传感器，在不同温度范围内做一个简单的软件修调，通常情况下可以使 RCHF 16Mhz 实现全温区+/-1%的精度要求，以及 RCHF 24Khz 实现全温区+/-1.5%的精度要求。关于参考例程和使用方法，请咨询复旦微电子公司。

内部中频 RC 振荡器

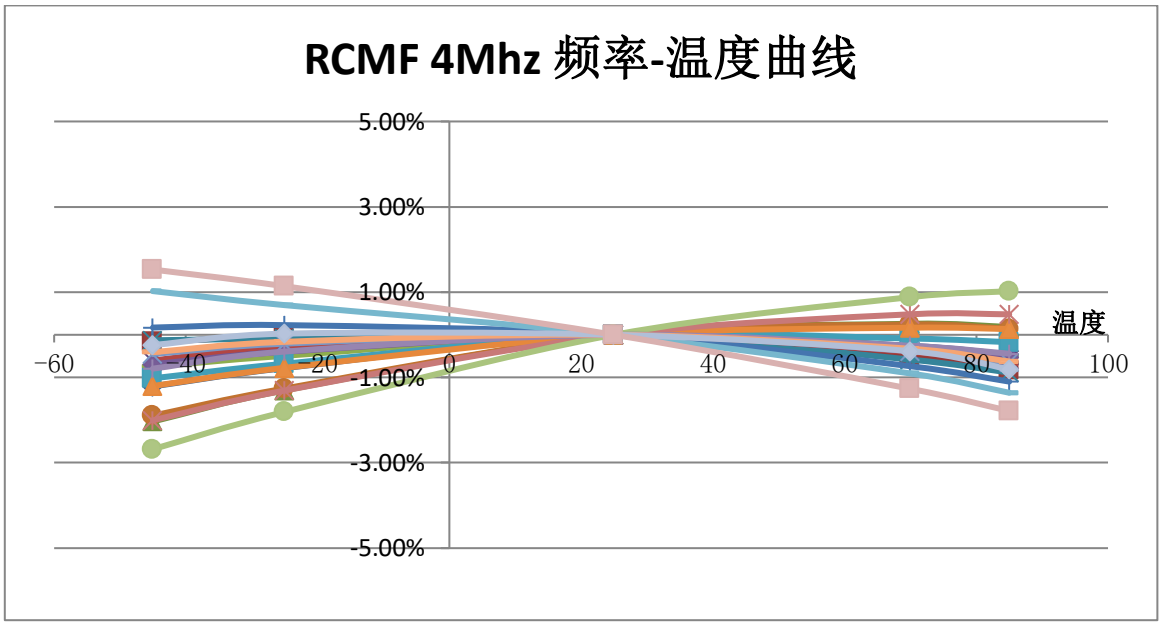
符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{RCMF}	RCMF 低功耗振荡频率	VDD=1.8~5.5V T=25℃	3.6	4	4.4	MHz

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
I_{DD_RCMF}	RCMF 功耗	VDD=1.8~5.5V TA=25℃		20		uA
t_{START}	RCMF 启动时间				10	us
ACC_{RCMF}[2]	全温区 RCMF 变化范围	VDD=1.8~5.5V TA=-40~+85℃	-3	-	3	%

表 3-14 内部 RC 振荡器参数

[注2]：此项指标基于特征参数提取

典型 RCMF 频率-温度变化曲线（基于特征参数提取，仅供设计参考）

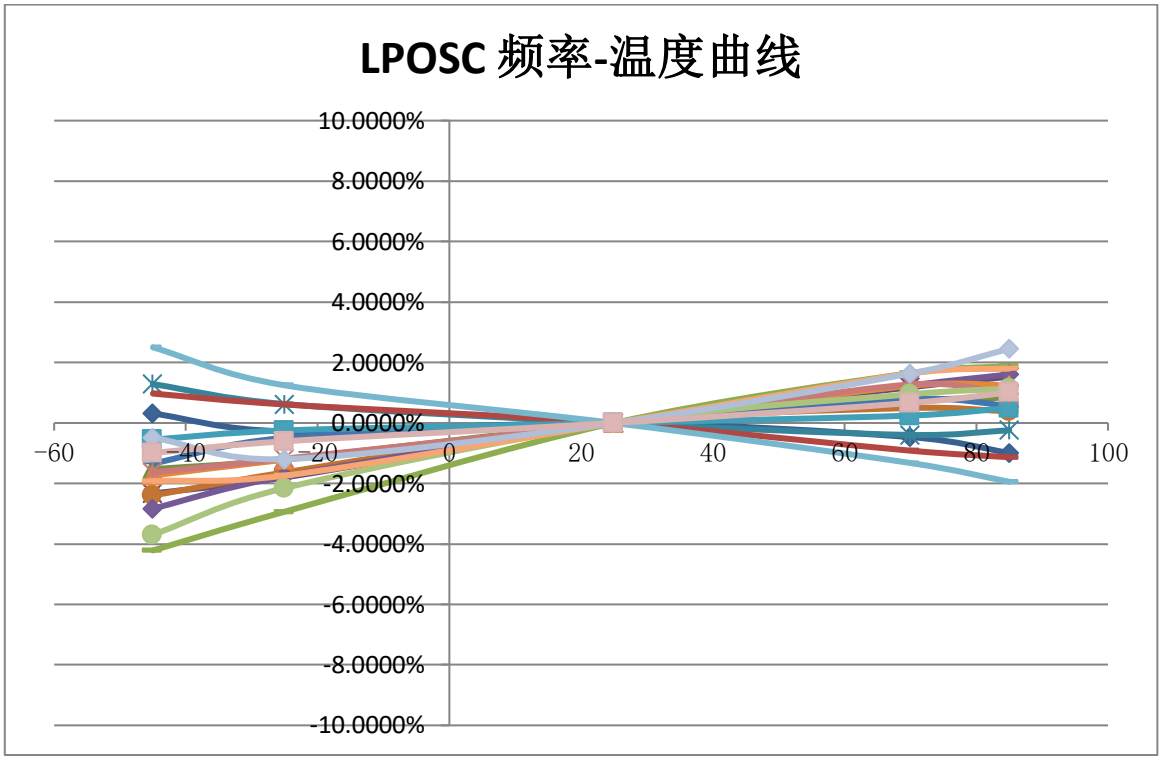


内部低频 RC 振荡器

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{LPOSC}	LPOSC 低功耗振荡频率	VDD=1.8~5.5V T=25℃		32		KHz
I_{DD_LPOSC}	LPOSC 功耗	VDD=1.8~5.5V T=25℃		450		nA
t_{START}	LPOSC 启动时间			50	100	us
ACC_{RCMF}[2]	全温区 LPOSC 变化范围	VDD=1.8~5.5V TA=-40~+85℃	-6	-	4	%

表 3-15 内部低频 RC 振荡器参数

典型 LPOSC 频率-温度变化曲线（基于特征参数提取，仅供设计参考）



3.4.8 PLL 特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{PLL}	PLL 输出频率	VDD=1.8~5.5V T=25℃	32		64	MHz
I_{DD_PLL}	PLL 功耗	输入频率 1Mhz，输出频率 32Mhz		330		uA
		输入频率 1Mhz，输出频率 64Mhz		450		
t_{LOCK}	PLL 锁定时间			65		us

表 3-16PLL 参数

3.4.9 ADC 特性

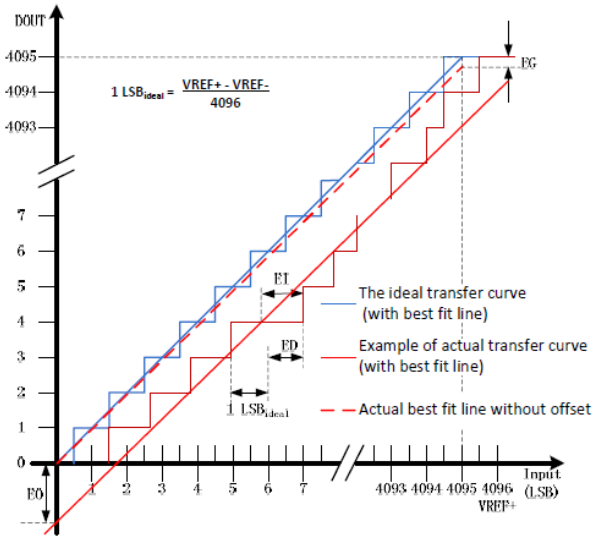
3.4.9.1 性能指标

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
VDDA	工作电压范围			1.8		5.5	V
VREF+	正参考电压			1.5		VDDA	V
VREF-	负参考电压			0		0.5	V
T_J	工作结温范围			-40		125	°C
V_{AIN}	输入电压范围	单端模式		VREF-		VREF+	V
C_s	采样保持电容				8		pF
F_{CLK}	ADC 工作时钟频率					16	MHz
F_S	ADC 采样频率	VDDA=2.5~5.5V				1	Msps
		VDDA=1.8~2.5V				0.5	
T_{SAMP}	采样保持时间			4		384	F_{ADCCLK}
T_{CONV}	转换时间				12		F_{ADCCLK}
I_{VDDA}	ADC 使能时 VDDA 功耗	VDDA=3.3V	Fs=1Msps		2		mA
			Fs=250Ksps		1.8		
		VDDA=5V	Fs=1Msps		2.7		
			Fs=250Ksps		2.4		
ADC 动态性能							
ENOB	有效位数与输入信号频率的关系 VDDA=3.3V VREF+=VDDA F_S=1Msps T_A=25℃	单端模式 F_{AIN}=29KHz			10.9		bits
		单端模式 F_{AIN}=199KHz			9		bits
	有效位数与工作电压的关系 VREF+=VDDA F_{AIN}=29KHz T_A=25℃	VDDA=5.0V F_S=1Msps			11.1		bits
		VDDA=3.3V F_S=1Msps			10.9		bits
		VDDA=2.7V F_S=1Msps			10.7		bits
		VDDA=2.5V F_S=1Msps			10.5		bits
		VDDA=1.8V F_S=0.5Msps			10.8		bits
	低电源电压下采样低频信号输入 VREF+=VDDA F_{AIN}=3KHz T_A=25℃	VDDA=1.8V F_S=1Msps			10.5		bits
	256 倍硬件过采样平均 VDDA=3.3V VREF+=VDDA F_S=1Msps T_A=25℃	单端模式 F_{AIN}=29KHz			12.6		bits

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
SNDR	信噪失真比	VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz T_A=25℃		66.5		dB
		256 倍硬件过采样平均 VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz T_A=25℃		77		
SFDR	无杂散动态范围	VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz T_A=25℃		73		dB
		256 倍硬件过采样平均 VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz T_A=25℃		85.3		
ADC 静态性能						
ED	差分非线性	VDDA=3.3V VREF+=VDDA F_S=1Msps T_A=25℃	-1	-	2	LSB
		VDDA=1.8V VREF+=VDDA F_S=0.5Msps T_A=25℃	-1	-	3	
EI	积分非线性	VDDA=3.3V VREF+=VDDA F_S=1Msps T_A=25℃	-2	-	2	LSB
		VDDA=1.8V VREF+=VDDA F_S=0.5Msps T_A=25℃	-3	-	3	
EO	失调误差	VDDA=3.3V VREF+=VDDA F_S=1Msps T_A=25℃	-5	-	5	LSB
EG	增益误差		-	4	10	LSB

表 3-17ADC 参数

ADC 静态性能指标示意图：



ED = Differential linearity error: maximum deviation between actual steps and the ideal one.

EI = Integral linearity error: maximum deviation between any actual transition and the best fit correlation line.

EO = Offset error: deviation from actual best fit line to the ideal one at the lowest code.

EG = Gain error: deviation of the slope of the best fit line to the ideal slope.

3.4.9.2 输入通道阻抗

下图表示了 ADC 输入通道的阻抗分布。

- ADC_Inx 表示快速外部通道 0~7
- ADC_Iny 表示慢速外部通道 8~11
- R_{IO} 表示引脚输入开关阻抗，R_{SW1} 和 R_{SW2} 表示 ADC 输入 MUX 开关阻抗
- C_S 表示 ADC 内部采样电容，典型值 8pF
- 典型情况下（T=25°C，VDDA=3.3V），R_{IO} = 200Ω，R_{SW1} = 103Ω，R_{SW2} = 206Ω

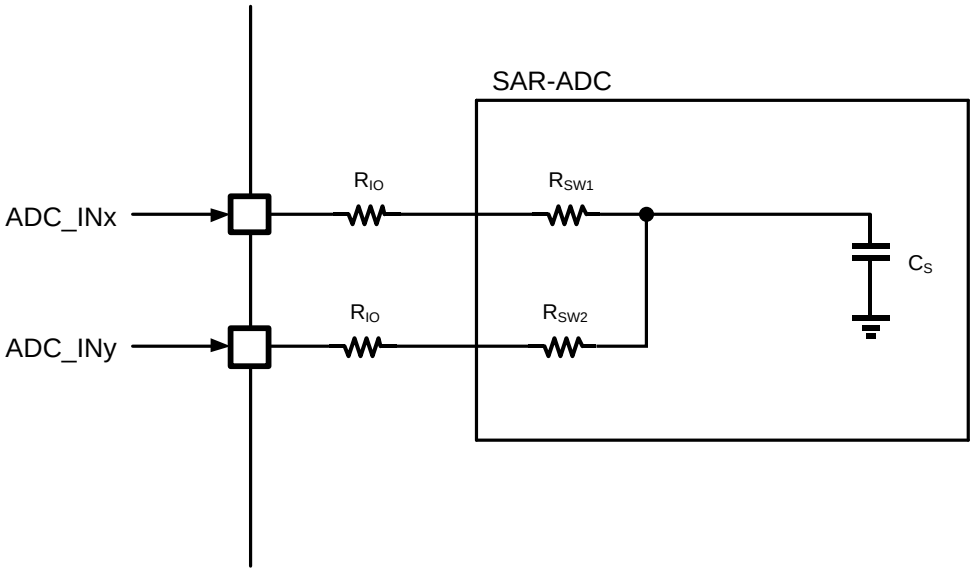


图 3-3ADC 通道输入阻抗

3.4.9.3 采样时间

ADC 输入信号采样时间最小值由被采样的模拟信号源内阻、信号输入通道阻抗、引脚寄生电容、采

样电容共同决定。

下表是不同条件下推荐的最小采样时间，供应用参考。（R_O 表示模拟信号源输出阻抗）

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
快速通道，ADC_IN0~7，VDDA=3.3V，T=25°C						
t_S	采样时间	R_O=0.1KΩ	15	-	-	ns
		R_O=1KΩ	55	-	-	ns
		R_O=5KΩ	235	-	-	ns
		R_O=50KΩ				
		R_O=100KΩ				
慢速通道，ADC_IN8~11，VDDA=3.3V，T=25°C						
t_S	采样时间	R_O=0.1KΩ	20	-	-	ns
		R_O=1KΩ	60	-	-	ns
		R_O=5KΩ	240	-	-	ns
		R_O=50KΩ				
		R_O=100KΩ				
快速通道，ADC_IN0~7，VDDA=1.8V，T=25°C						
t_S	采样时间	R_O=0.1KΩ	60	-	-	ns
		R_O=1KΩ	100	-	-	ns
		R_O=5KΩ	280	-	-	ns
慢速通道，ADC_IN8~11，VDDA=1.8V，T=25°C						
t_S	采样时间	R_O=0.1KΩ	80	-	-	ns
		R_O=1KΩ	120	-	-	ns
		R_O=5KΩ	300	-	-	ns

表 3-18ADC 采样时间

3.4.10 温度传感器

芯片出厂时经过温度定标，定标条件是 VDDA=3.0V，TA=30+/-1°C。在此条件下，使用 ADC 采样并转换温度传感器输出电压，将转换结果保存在 Flash 指定地址。

符号	参数说明	测试条件	数据保存地址
TS_CAL1	温度传感器标定值 1	VDDA=3.0V ± 10mV，T_A=30+/-1°C	0x1FFF_FA90 高半字保存 ADC 数据 低半字为定标温度，0x1E00 表示 30°C

注：根据 TS_CAL1 的数值，可以计算温度定标时温度传感器在 30°C 下的输出电压绝对值。

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
Reso	分辨率[1]	VDDA=VREF+=5V		2.4		LSB/°C
		VDDA=VREF+=3V		3.7		
Slope	输出斜率[1]	T_A=-40~+85°C		3.06		mV/°C

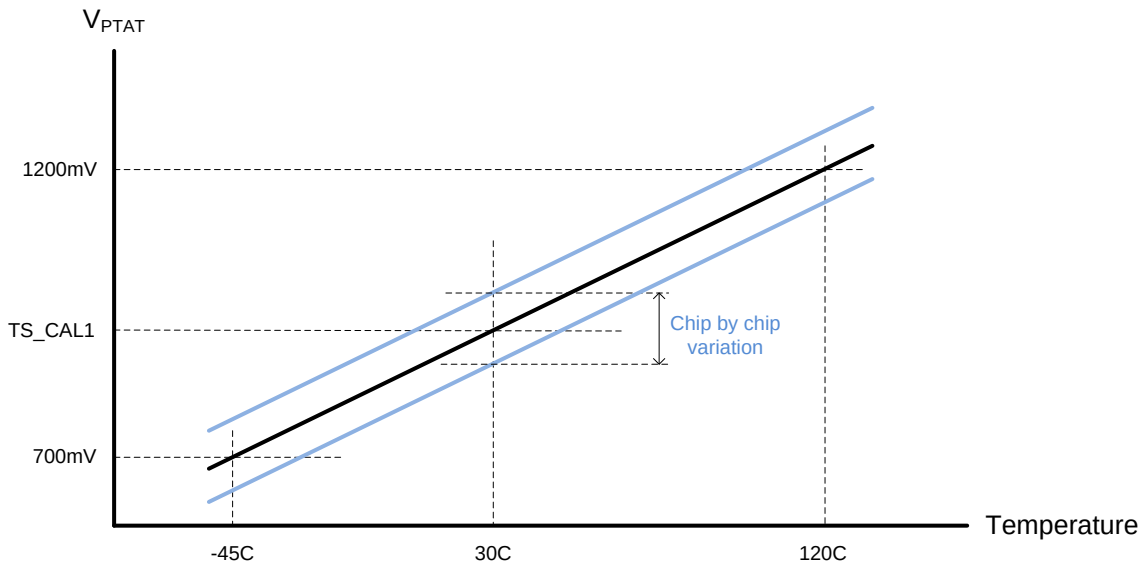
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
		VDDA=1.8~5.5V					
V_{PTAT}	温度传感器输出电压绝对值	VDDA=1.8~5.5V	T_A=30° C	900	930	960	mV
			T_A=85° C	1075	1105	1135	
			T_A=-40° C	675	705	730	
Linerity	全温区线性度[1]			-	+/-1	+/-2	°C
I_{DDA}	温度传感器功耗（不含ADC）[2]	VDDA=3.3V			0.8		uA
t_{START}	温度传感器启动时间，包含输出 buffer 建立时间[2]	VREF1p2 已经使能，置位 PTAT_EN 寄存器、VPTATBUFFER_OUTEN、VPTATBUFFER_EN				50	us
		VREF1p2 未使能				1.4	ms
t_{SAMPLE}	ADC 采样温度传感器输出时要求的采样时间[2]			10	-	-	us

表 3-19 温度传感器参数

[1]基于特征参数提取

[2] 基于电路设计仿真

温度传感器输出曲线示意图如下。



温度传感器输出电压只和芯片基底温度有关，而与芯片当前工作电源电压无关。

3.4.11 运算放大器特性

放大器模式（T_A=25°C）

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
VDDA	工作电压范围		2	-	5.5	V
CMIR	共模输入范围		0.1	-	VDDA-0.1	V
VI_{offset}	输入 offset 电压	VDDA=3.3V 0.3V<VCM<3V	-	+/-1	+/-3	mV
I_B	输入偏置电流	VDDA=3.3V	-	0.5	1	nA
I_{OS}	输入失调电流	VDDA=3.3V	-	-	+/-10	pA
TRIMSTEP_P	低共模输入电压 offset trim 步长	VDDA=3.3V VCM=0.82V		1.2		mV
TRIMSTEP_N	高共模输入电压 offset trim 步长	VDDA=3.3V VCM=2.48V		1.1		mV
I_{LOAD}	驱动电流（Buffer 模式）阻性 4Kohm	VDDA=2V		400		uA
		VDDA=3.3V		700		uA
		VDDA=5V		1100		uA
C_{LOAD}	容性负载				50	pF
CMRR	共模抑制比	VDDA=3.3V VCM=VDDA/2		80		dB
PSRR	电源抑制比	1.8V<VDDA<5V		80		dB
GBW	单位增益带宽	VDDA=3.3V, 1V<VCM<2.4V		6000		KHz
		VDDA=3.3V, VCM<=0.5V VCM>=3V		1000		KHz
SR	Slew Rate（输出电压变化范围是 10%到 90%）	VDDA=3.3V		1.5		V/us
		VDDA=2.0V		1.2		V/us
		VDDA=5.0V		2.0		V/us
AO	开环增益	VDDA=3.3V		74	90	dB
V_{OHSAT}	高饱和电压（Buffer 模式）	R_{load}=4K 输入为 VDDA, VDDA=2.0V		VDDA-350		mV
		R_{load}=4K 输入为 VDDA, VDDA=3.3V		VDDA-400		mV
		R_{load}=4K 输入为 VDDA, VDDA=5.0V		VDDA-550		mV
V_{OLSAT}	低饱和电压（Buffer 模式）	R_{load}=4K 输入为 0, VDDA=2.0V		170		mV
		R_{load}=4K 输入为 0, VDDA=3.3V		220		mV
		R_{load}=4K 输入为 0, VDDA=5.0V		280		mV
Phi	相位裕度	C_L=50pF, R_L=4Kohm		52		°
GM	增益裕度			10		dB
t_{START}	启动时间	Buffer 模式 2.0V=<VDDA<=5.5V		2		us
PGA gain	PGA 增益（OPA1）			2		
				4		
				8		

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
				16		
PGA error	PGA 增益误差	Gain=2	-3	-	3	%
		Gain=4	-5	-	5	
		Gain=8	-5	-	5	
		Gain=16	-7	-	7	
PGA BW	PGA 带宽	Gain=2		GBW/2		
		Gain=4		GBW/4		
		Gain=8		GBW/8		
		Gain=16		GBW/16		
I_{DDA}	功耗	VDDA=3.3V, VCM=VDDA/2		120		uA
		VDDA=3.3V, VCM=0		70		
		VDDA=3.3V, VCM=VDDA		60		

表 3-20OPA 参数

3.4.12 模拟比较器特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
VI_{comp1}	比较器 1 输入电压范围		0	-	VDDA	V
VI_{comp2}	比较器 2 输入电压范围		0	-	VDDA	V
I_{comp1}	比较器 1 工作电流	VDD=3.3V T_A=25°C	-	1.75	2	uA
I_{comp2}	比较器 2 工作电流	VDD=3.3V T_A=25°C	-	1.5	2	uA
T_{setup1}	比较器 1 建立时间	VDD=3.3V	-	8	12	us
T_{setup2}	比较器 2 建立时间	VDD=3.3V	-	8	12	us
$T_{propagation1}$	比较器 1 传播延迟	VDD=3.3V 200mV step, 100mV overdrive	-	0.6	1	us
$T_{propagation2}$	比较器 2 传播延迟	VDD=3.3V 200mV step, 100mV overdrive	-	0.6	1	us
V_{offset}	输入失调电压		-	-	±10	mV

表 3-21 模拟比较器参数

3.4.13 Flash 存储器特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	Flash size		128K	-	256K	bytes
T_{PROG}	32bits Program Time			25		μs
T_{ERASE}	Sector/Page Erase			2		ms
	Chip Erase			8		ms
N_{ED}	Endurance		20,000	100,000		Erase/Write cycles
T_{DR}	Data Retention	T_A=85℃ After 20K cycling	10			yrs

表 3-22 Flash 参数

3.4.14 GPIO 特性

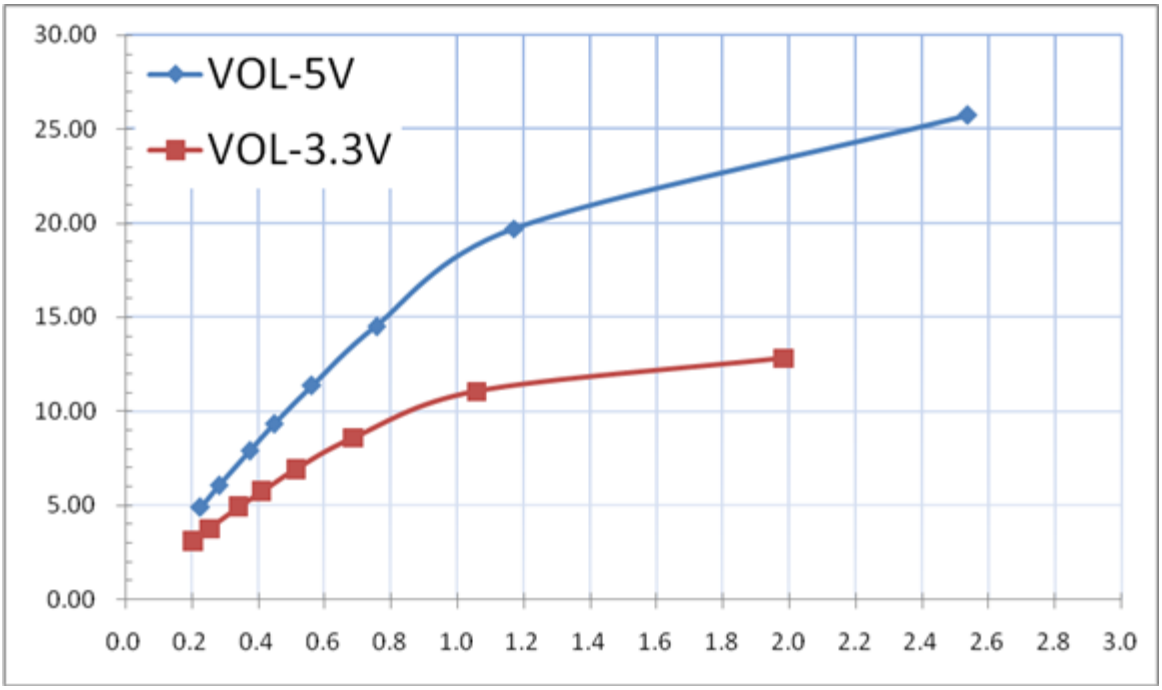
普通 IO

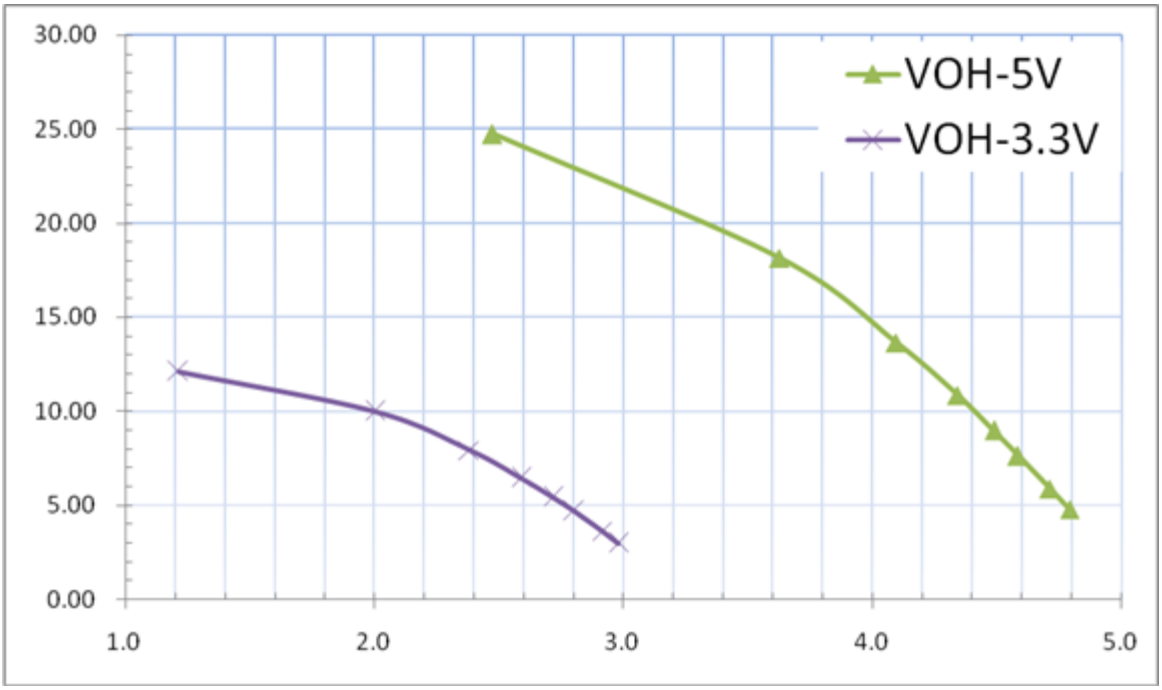
符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
V_{IL}	输入低电平			0		$0.3V_{DD}$	V
V_{IH}	输入高电平			$0.7V_{DD}$		V_{DD}	V
I_{IL}	输入低漏电	V_{IL}=0V		-1		1	μA
I_{IH}	输入高漏电	V_{IH}=3.3V		-1		1	μA
V_{OL}	输出低电平	V_{DD}=3.3V	I_{SINK}=5mA		0.35		V
			I_{SINK}=10mA		0.9		
		V_{DD}=5V	I_{SINK}=10mA		0.5		
			I_{SINK}=15mA		0.8		
V_{OH}	输出高电平	V_{DD}=3.3V	I_{SOURCE}=5mA		2.8		V
			I_{SOURCE}=10mA		2		
		V_{DD}=5V	I_{SOURCE}=10mA		4.4		
			I_{SOURCE}=15mA		3.95		
R_{PU}	弱上拉电阻				100		KΩ

表 3-23 普通 I/O 参数

典型 IO 驱动能力曲线（基于特征参数提取，仅供设计参考）

注：以下图形中，Y 轴单位是 mA，X 轴单位是 V，两条曲线分别代表 5V 和 3.3V 供电下的驱动能力





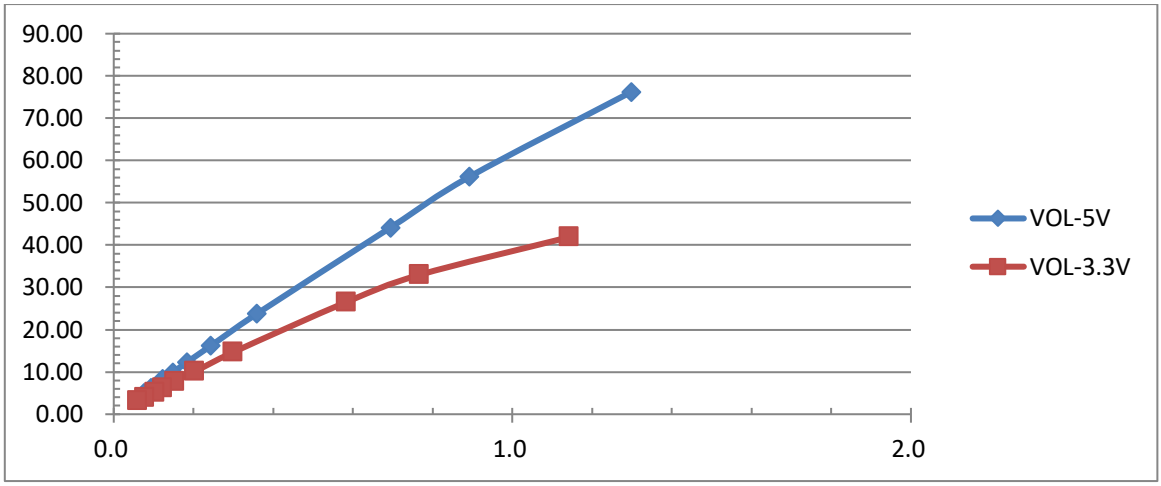
真开漏 IO（PA11、PA12）

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		$0.3V_{DD}$	V
V_{IH}	输入高电平		$0.7V_{DD}$		V_{DD}	V
I_{IL}	输入低漏电	V_{IL}=0V	-1			μA
I_{IH}	输入高漏电	V_{IH}=3.3V			1	μA
V_{OL}	输出低电平	V_{DD}=3.3V I_{SINK}=20mA		0.42		V
		V_{DD}=5V I_{SINK}=20mA		0.3		
V_{OH}	输出高电平		NA			V
R_{PU}	弱上拉电阻		NA			KΩ

表 3-24 真开漏 I/O 参数

典型真开漏 IO 驱动能力曲线（基于特征参数提取，仅供设计参考）

注：以下图形中，Y 轴单位是 mA，X 轴单位是 V，两条曲线分别代表 5V 和 3.3V 供电下的驱动能力



NRST 引脚

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		$0.3V_{DD}$	V
V_{IH}	输入高电平		$0.7V_{DD}$		V_{DD}	V
I_{IL}	输入低漏电	V_{IL}=0V	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=3.3V	-1		1	μA
R_{PU}	上拉电阻			5		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=3.3V		100		ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.8~3.6V -40℃≤T_A≤85℃	50		100	us

表 3-25 NRST 引脚参数

注：

[1] 此项参数基于特征参数提取

GPIO AC 特性

IO	符号	参数说明[1]	测试条件	min	max	单位
非 FM+	Fmax	Maximum frequency	C=30pF,2.7V<Vdd<3.6V	-	45	MHz
			C=30pF,1.6V<Vdd<2.7V	-	22	
			C=10pF,2.7V<Vdd<3.6V	-	80	
			C=10pF,1.6V<Vdd<2.7V	-	40	
	Tr/Tf	Output rise and fall time	C=30pF,2.7V<Vdd<3.6V	-	8.7	ns
			C=30pF,1.6V<Vdd<2.7V	-	16.9	
			C=10pF,2.7V<Vdd<3.6V	-	3.4	
			C=10pF,1.6V<Vdd<2.7V	-	6.7	
FM+	Fmax	Maximum frequency	C=50pF,1.6V<Vdd<3.6V	-	10	MHz
	Tf	Output fall time		-	27	ns

表 3-26 引脚 AC 参数

注：

[1] 依据电路仿真，不在量产测试中测试

3.4.15 LCD 特性

片内电阻分压模式

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
I_{LCD}	片内电阻分压模式下的 LCD 工作电流（空载）[1]	VDD=5V 1/3 偏置，输出空载，4COM		2		uA
V_{LCD}	LCD 偏置电压	-	0.547×VDD		VDD	V

表 3-27 LCD 片内电阻分压

[1]基于特征参数提取，不包含在量产测试中

3.4.16 USB 特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{DD}	USB 工作电压范围		2.805	3.3	3.795	V
V_{DI}	Differential input sensitivity	$\|V_{I(DP)}- V_{I(DM)}\|$	0.2	-	-	V
VCM	Differential common mode voltage		0.8	-	2.5	V
V_{IH}	输入高电平		2.0	-	-	V
V_{IL}	输入低电平		-	-	0.8	V
V_{OL}	输出低电平		0	-	0.3	V
V_{OH}	输出高电平		2.8	-	3.6	V
V_{CRS}	DP/DM 交错点		1.3	-	2.0	V
Z_{DRV}	驱动器输出阻抗		28	-	44	Ohm
t_R	输出上升时间	CL=50pF 10%~90% of $\|V_{OH}- V_{OL}\|$	4	-	20	ns
t_F	输出下降时间	CL=50pF 10%~90% of $\|V_{OH}- V_{OL}\|$	4	-	20	ns
t_{FRMA}	差分上升/下降时间匹配 (t_R/t_F)		90	-	111	%

表 3-28 USB PHY 电特性

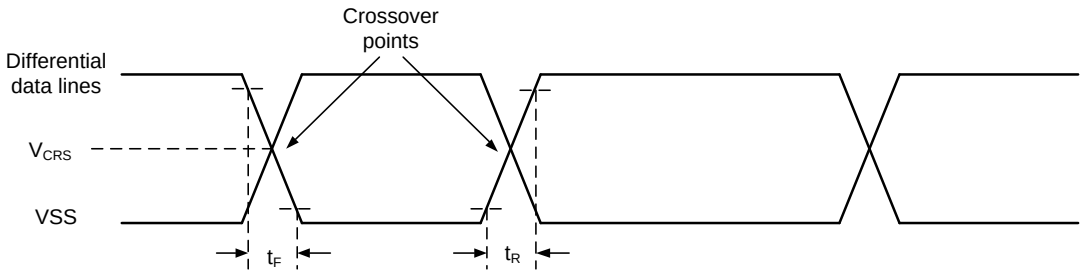


图 3-4USB 差分信号时序图

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F48M	连续校准的 48M 时钟频率		47.88	48	48.12	Mhz
F120M	120M 本振输出频率		112	120	126	Mhz
Duty48M	48M 时钟占空比		45	-	55	%
Duty120M	120M 时钟占空比		45	-	55	%
Tsof	完成 USB 时钟跟踪所需的 SOF 包		10	-	-	

表 3-29 USB 时钟特性

4 电源管理单元（PMU）

4.1 芯片工作电源

4.1.1 电源域划分

- VDD

芯片的主电源（VDD）的典型工作电压范围是 1.8~3.6V。

其中，芯片上电时，复位释放阈值主要由 BOR 电路决定，其典型复位释放电压是 1.8V。如果芯片电源上升时间很短（小于几个 ms），则上电复位释放电压主要由 RC 延迟决定，典型情况下将略低于 1.8V，低温下可能略高于 1.8V。

芯片下电时，如果使能了 BOR，下电复位阈值由 BOR 电路决定，可以由软件配置 BOR_PDRCFG 获得 4 个阈值档位，默认值为 1.6V。如果没有使能 BOR，使能了 PDR，下电复位阈值由 PDR 电路决定，软件可以通过 PDRCFG 配置 3 个档位，默认值 1.4V 左右。

综上，芯片的 VDD 实际工作电压范围将由 BOR 和 PDR 电路配置共同决定。

注意：在任何情况下不得同时关闭 BOR 和 PDR，这样在芯片掉电时可能由于没有产生正常的复位，而导致重新上电时芯片无法正常工作。

- VDDA

VDDA 是专用的模拟电路电源，主要给 ADC、OPA、基准电压等模拟模块供电。VDDA 的工作电压范围是 1.8~5.5V，所有模拟模块在这个电压范围内都可以保证正常工作。

- VREFP

仅有少部分封装形式中有独立的 VREFP 引脚，VREFP 是 ADC 的基准电压输入，在 ADC 工作时，会从 VREFP 引脚抽取几十到上百 uA 的电流。大部分封装中，VREFP 都是和 VDDA 封装在一起的。

- VDD15

VDD15 是芯片内核电源，由一个线性电源稳压器产生 1.5V 电源输出。所有的数字电路、Flash、SRAM 和部分模拟电路工作在这个电源下。VDD15 引脚需要外挂 0.1~1uF 稳压电容。当主电源 VDD 跌落至 1.5V 以下时，稳压器输出将跟随 VDD 变化。

4.1.2 电源结构图

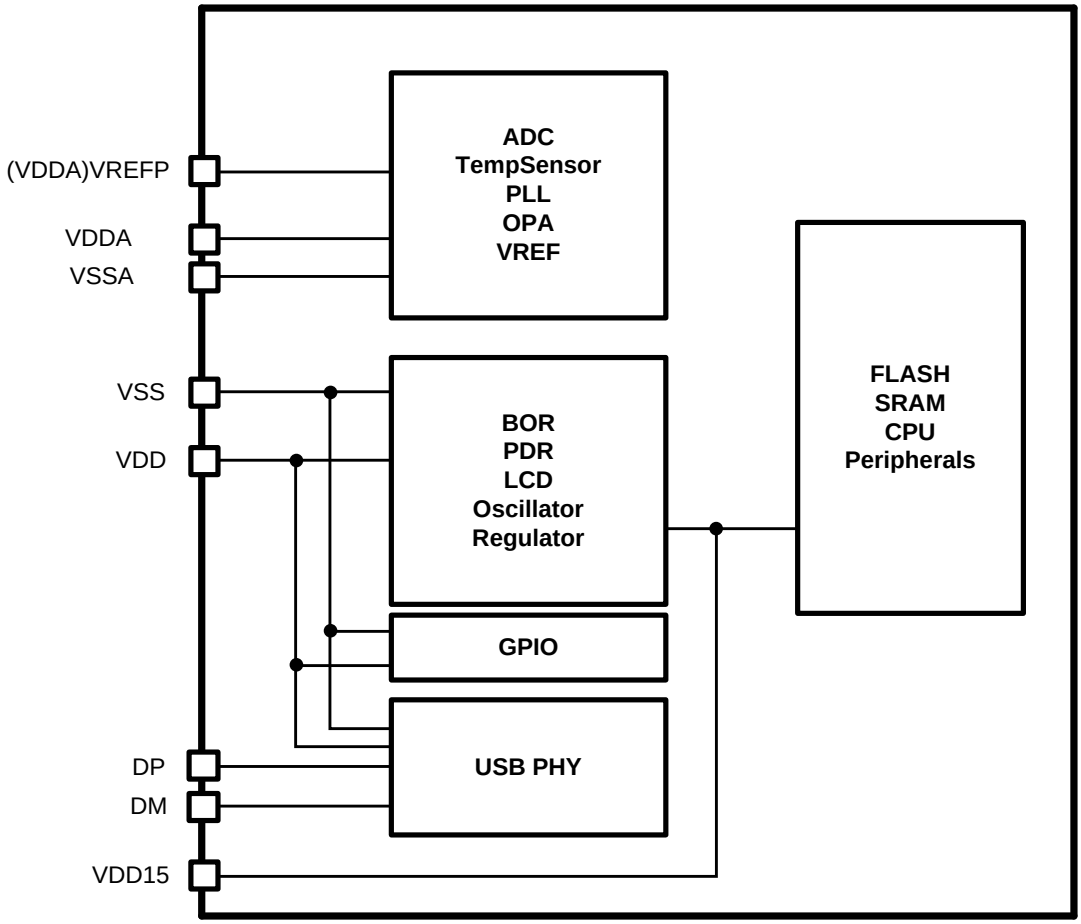


图 4-1 芯片电源结构图

4.1.3 ADC 和基准电压的独立供电

为了提高 ADC 转换精度，降低电源噪声的影响，ADC 和基准电压使用独立的 VDDA 和 VSSA 引脚供电。在系统上，可以对这 VDDA 电源单独滤波，还可以在 PCB 上对 VDDA 和 VSSA 走线进行 shielding，以尽可能屏蔽系统噪声。

FM33LC0xx 的大部分封装形式中，VDDA 和 VSSA 都是独立引脚，而 ADC 基准源 VREFP 和 VDDA 在封装内部相连，VREFN 和 VSSP 在封装内部相连。

但是在某些低管脚数的封装中，可能没有独立的 VDDA 和 VSSA，在这种情况下，VDDA 和 VDD 在封装内部相连，VSSA 和 VSS 在封装内部相连，由于电源地噪声影响，ADC 的性能会有所下降。而在少数管脚数较多的封装中，则会单独引出 VREFP 和 VREFN 引脚，通过独立的 ADC 基准电源引脚，系统可以进一步优化 ADC 的电源环境，来获得最佳性能。VREFP 与 VDDA 独立时，其输入基准可以与 VDDA 不同，允许的输入范围是：

1.8V ≤ VREFP ≤ VDDA

4.2 功耗模式

4.2.1 概述

上电复位后，芯片默认运行在 ACTIVE 模式，此时 CPU 正常从 flash 取指运行，所有外设模块都可以正常工作。芯片支持多种低功耗模式，软件可以在适当的场景下选择合适的低功耗模式，以平衡不同的功耗、性能、唤醒时间和唤醒条件的要求。

芯片支持的功耗模式：

- ACTIVE 模式：正常运行
- LP Active 模式：LDO 进入低功耗模式，CPU 主频不超过 4MHz，所有外设可以运行
- LP Run 模式：LDO 工作在超低功耗模式下，CPU 和外设只能运行在较低频率下
- SLEEP 模式：CPU 停止，Flash 停止，LDO 工作在低功耗模式下，仅部分外设可以运行
- DEEPSLEEP 模式：CPU 停止，Flash 停止，关闭基准电压，LDO 工作在低功耗模式下，仅部分外设可以运行

此外，ACTIVE 模式下的运行功耗也可以通过以下手段降低：

- 降低系统时钟频率
- 关闭不使用的外设的总线时钟和工作时钟

功耗模式	典型功耗	唤醒条件	芯片状态	典型唤醒时间[1]
ACTIVE	95uA/MHz@64MHz 120uA/MHz@48MHz		正常工作 Dhrystone	-
LP Active	500uA	软件主动退出	LDO 进入低功耗模式 Dhrystone	-
LP Run	30uA@32KHz	软件主动退出	低速工作	-
SLEEP	6uA	电源检测中断 比较器中断 RTC 定时中断 IO 引脚中断 WKUPx 唤醒 32K 晶振停振 看门狗复位 NRST 引脚复位	CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 RTC 走时 VREF1p22 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	3us
DEEPSLEEP	1uA		CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 RTC 走时 VREF1p22 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	5us

表 4-1 芯片功耗模式表

注：[1] 典型唤醒时间指从唤醒信号到来，到 CPU 开始执行唤醒中断服务程序的时间间隔。

[2] CPU 自身进入休眠的步骤参见 ARMv6-M 架构参考手册

[3] CPU 试图进入低功耗模式时，如果 Flash 正在擦写，则芯片自动等待 Flash 擦写结束后再进入低功耗模式。

4.2.1 功耗模式与系统频率

在不同功耗模式下，CPU 主频的限制如下图所示：

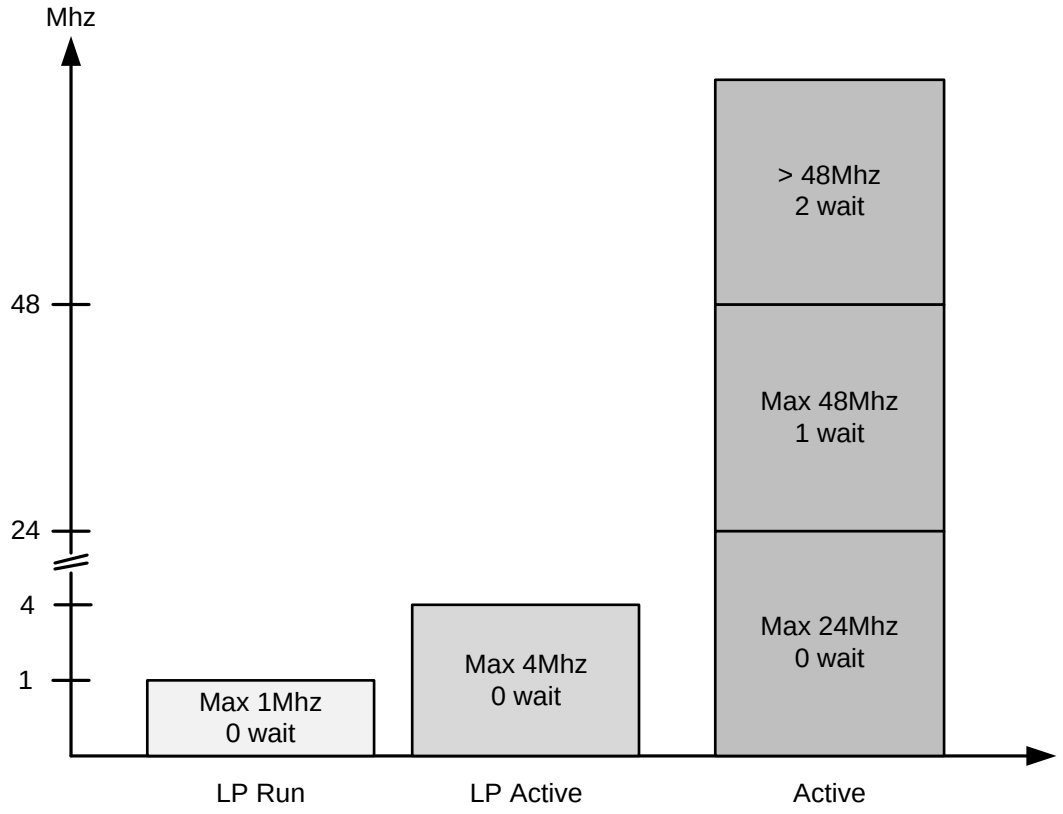


图 4-2 功耗模式与系统主频

不同功耗模式下可以接受的系统频率和可用的时钟源如下表所示。应用软件应严格遵守这个表格的规定，在低功耗模式下使用高主频可能导致系统无法正常运行。

功耗模式	CPU 频率	可用时钟源	Flash wait	外设工作时钟
ACTIVE	<=24Mhz	All	0	All
	>24Mhz, <=48Mhz		1	
	>48Mhz		2	
LP Active	<=4Mhz	RCHF, RCMF, XTLF, LPOSC	0	RCHF, RCMF, XTLF, LPOSC
LP Run	<=1Mhz	RCMF, XTLF, LPOSC	0	RCMF, XTLF, LPOSC

表 4-2 功耗模式和可用系统时钟

4.2.2 Active 模式

芯片正常工作模式。芯片上电复位完成后进入 Active 模式运行，默认的 CPU 频率是 8MHz，最高可以运行到 64MHz。在 Active 模式下所有的数字和模拟外设都可以全速运行。

当主频高于 24MHz 时，访问 Flash 时必须插入 wait cycle，并且建议软件开启 Flash Prefetch 来提升指令执行效率。

4.2.3 LP Active 模式

软件通过置位 LPMCFG.LDO_LPM 寄存器，可以进入 LP Active 模式。此时 LDO 被置于低功耗模式下，本身功耗下降的同时，驱动能力也有所下降。因此在 LP Active 模式下，推荐 CPU 主频不要超过 4MHz。同时，外设模块仍可以使用 RCHF、RCMF、XTLF、LPOSC 工作，但是 PLL 和 XTHF 被硬件强制关闭，无法使用。

LP Active 模式的典型应用场景是，在对 CPU 处理能力要求不高的场景下，令 CPU 待机或者低速运行时，保持 1~2 个外设（如 UART、Timer）长时间正常运行，为一些特殊的低功耗场景提供最优的能效比。

进入 LP Active 模式

- 将系统时钟配置为 4MHz 或更低
- 确保没有外设正在使用 XTHF 或 PLL 时钟
- 置位 LDO_LPM 寄存器

LP Active 模式下的硬件行为

进入 LP Active 之后，硬件自动关闭 XTHF、PLL，随后使 LDO 进入低功耗模式。所有模拟和数字外设都可以工作。

退出 LP Active 模式

- 软件清零 LDO_LPM 寄存器
- 等待几条 NOP 指令
- 根据需要配置系统时钟，恢复正常的 Active 模式运行

LP Active 模式下 CPU 改写 PMOD 寄存器可以直接进入 LP RUN/SLEEP/DEEPSLEEP 模式。

4.2.4 LP Run 模式

当芯片需要低功耗低速运行时，可进入 LP RUN 模式，此时 LDO 进入低功耗模式，内核使用 LSCLK 或

RCMF 分频运行，典型频率 32KHz。在需要高速运行时，软件可主动退出 LP RUN 进入 ACTIVE 模式，然后再将系统时钟切换到较高频率。

进入 LPRUN 模式

进入 LPRUN 的操作步骤：

- 软件将系统时钟（SYSCLK）配置为 LSCLK 或 RCMFPSC
- 配置 PMOD 寄存器为 01
- 如果系统时钟配置不满足以上寄存器条件，则置位异常中断并且禁止进入 LPRUN

LPRUN 模式下的硬件行为

进入 LP Run 之后，硬件自动关闭 RCHF、XTHF、PLL、TRNG，随后使 LDO 进入低功耗模式。SVD、比较器、ADC 仍可以在 LPRUN 模式下工作。由于高速时钟都被关闭，ADC 工作时钟最高只有 RCMF，相当于最快 250Ksps 采样率。

如果软件在 LPRUN 模式下执行 WFI/WFE 指令，CPU 和 Flash 将停止活动，但是外设仍可以继续工作。

退出 LPRUN 模式

按照以下步骤退出 LPRUN 模式：

- 软件将 PMOD 寄存器配置为 00
- 软件根据需要使能 RCHF 或 PLL
- 等待时钟建立后配置系统时钟为 RCHF 或 PLL

LP Run 模式下 CPU 改写 PMOD 寄存器可以返回 ACTIVE，或者进入 SLEEP/DEEPSLEEP 模式。如果返回 ACTIVE，硬件自动将 LDO 置于正常模式，并解除对高速时钟模块的限制。

4.2.5 SLEEP 模式

通过进入 Sleep 模式，可以大幅降低芯片功耗，并处于等待事件唤醒的状态中。

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 指令

LPRUN 模式下的硬件行为

进入 SLEEP 模式后芯片关闭 CPU 时钟，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、XTHF、TRNG，

SVD、OPA、ADC 仍可以在 SLEEP 模式下工作。其中由于高速时钟都被关闭，ADC 工作时钟最高只有 RCMF，相当于最快 250Ksps。

数字外设模块可以使用 RCMF、XTLF、LPOSC 等低速时钟继续工作。

退出 SLEEP 模式

按照以下步骤退出 SLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF
- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

4.2.6 DEEPSLEEP 模式

DEEPSLEEP 是芯片最低功耗模式。

进入 DEEPSLEEP 模式

软件按如下步骤进入 DEEPSLEEP 模式：

- 置位 VREFOFF 寄存器
- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 指令

DEEPSLEEP 模式下的硬件行为

DEEPSLEEP 模式下，芯片自动关闭 CPU 时钟，关闭内部基准源，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、TRNG；SVD、OPA、ADC、比较器仍可以在 DEEPSLEEP 模式下工作。其中由于高速时钟都被关闭，ADC 工作时钟最高只有 RCMF，相当于最快 250Ksps。如果需要 OPA 工作，则不能置位 VREFOFF 寄存器。

数字外设模块可以使用 RCMF、XTLF、LPOSC 等低速时钟继续工作。

退出 DEEPSLEEP 模式

按照以下步骤退出 DEEPSLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF

- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

4.3 唤醒源

唤醒源	应用	可唤醒模式	
		Sleep	DeepSleep
停振检测	可屏蔽，32786Hz 晶振停振时唤醒芯片	√	√
VREF	可屏蔽，在 VREF1p22 建立后产生中断唤醒芯片	√	√
SVD	可屏蔽，在电源电压跌落至阈值以下或升高至阈值以上时唤醒芯片	√	√
比较器	可屏蔽，用于外部事件唤醒	√	√
ADC	可屏蔽，ADC 的各种中断均可用于唤醒	√	√
RTC	可屏蔽，根据需要的唤醒周期设置	√	√
IO 引脚中断	可屏蔽，用于外部事件唤醒	√	√
Debug	不可屏蔽，用于 debug 唤醒	√	√
LPUART	可屏蔽，接收数据唤醒	√	√
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√
NRST	不可屏蔽，用于全局复位	√	√
LPTIM32	可屏蔽，用于定时唤醒	√	√
BSTIM32	可屏蔽，用于定时唤醒	√	√
I2C 从机	可屏蔽，用于从机接收唤醒	√	√

通过Cortex-M0的PRIMASK功能，可以实现以上中断事件唤醒芯片，但是CPU不执行中断处理程序。此时唤醒后CPU将继续从休眠前的指令之后开始运行。

注：芯片从休眠模式唤醒后，软件可以通过查询PMU.WKPFLAG寄存器来快速识别当前的唤醒源，唤醒源的清除需要进入各个外设模块分别完成。

4.4 休眠唤醒后的时钟

当芯片从Sleep/DeepSleep模式唤醒后，芯片以RCHF为时钟源。寄存器将保留休眠前RCHF的频率配置和trim值，因此唤醒后CPU运行频率将由休眠前软件配置寄存器决定（PMU.WKFSEL）。最快情况下芯片唤醒后将以24MHz时钟启动。

休眠时AHBPRES寄存器不会复位，但是SYSCLKSEL寄存器将复位成00（选择RCHF）。因此，如果休眠前系统时钟不是RCHF，则唤醒后将默认使用RCHF，是否经过分频由休眠前的AHBPRES寄存器决定。

4.5 寄存器

offset 地址	名称	符号
PMU(模块起始地址: 0x40000100)		
0x00000000	低功耗控制寄存器 （Power Management Control Register）	PMU_CR
0x00000004	唤醒时间控制寄存器 （Wakeup Time Register）	PMU_WKTR
0x00000008	唤醒源标志查询寄存器 （Wakeup Source Flags Register）	PMU_WKFR
0x0000000C	PMU 中断使能寄存器 （PMU Interrupt Enable Register）	PMU_IER
0x00000010	PMU 中断标志寄存器 （PMU Interrupt and Status Register）	PMU_ISR

4.5.1 低功耗控制寄存器（PMU_CR）

名称	PMU_CR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				LDO_LPM		LDO15EN	LDO15EN_B
位权限	U-0				R/W-01		R-1	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKFSEL		SLPDP	CVS
位权限	U-0				R/W-00		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RFU		PMOD	
位权限	U-0				R/W-00		R/W-00	

位号	助记符	功能描述
31:20	**-**	未实现，读为 0
19:18	**LDO_LPM**	LDO 低功耗模式配置(LDO Low Power Mode) 00/01/11：正常模式 10：LDO 进入低功耗模式 注：当芯片正在对 FLASH 擦、编程时，该寄存器无法改写为 10
17	**LDO15EN**	LDO15 使能标志位(LDO Enable) 1：LDO15 处于工作状态 0：LDO15 被关闭
16	**LDO15EN_B**	LDO15 使能标志反码校验位(LDO Enable Inversed)
15:12	**-**	未实现，读为 0
11:10	**WKFSEL**	Sleep/DeepSleep 唤醒后的系统频率(Wakeup Frequency Config) 00：RCHF-8MHz 01：RCHF-16MHz 10：RCHF-24MHz

位号	助记符	功能描述
		11：RFU
9	**SLPDP**	DeepSleep 控制寄存器(Sleep Deep) 1：DeepSleep 模式使能，下关闭基准电压源 0：常规 Sleep 模式 在 Sleep 下，如果置位了 SLPDP 位即为 DeepSleep 模式； 该位仅在 Sleep 下有效
8	**CVS**	CoreVoltageScaling 配置(Core Voltage Scaling Enable) 0：低功耗模式下不使能内核电压调整 1：低功耗模式下降低内核电压 该位仅在 Sleep/DeepSleep 模式下起作用
7:4	**-**	未实现，读为 0
3:2	**RFU**	Dummy 寄存器
1:0	**PMOD**	低功耗模式配置寄存器(Power Mode) 00：Active mode / LP Active mode 01：LPRUN mode 10：Sleep mode / DeepSleep mode 11：RTCBKP mode

4.5.2 唤醒时间控制寄存器（PMU_WKTR）

名称	PMU_WKTR							
offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					STPCLR	T1A	
位权限	U-0					R/W-0	R/W-01	

位号	助记符	功能描述
31:3	**-**	未实现，读为 0
2	**STPCLR**	Flash Stop 唤醒控制 (Stop clear) 0：Stop 信号等待时钟建立后同步清零 1：Stop 信号异步清零
1:0	**T1A**	可编程额外唤醒延迟 (extra wakeup delay) 在 Sleep/DeepSleep 模式下，RCHF 时钟到来后，根据此寄存器配置等待额外延迟时间 00：0us 01：2us 10：4us 11：8us

4.5.3 唤醒源标志查询寄存器（PMU_WKFR）

名称	PMU_WKFR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ADCWKF	-		RTCWKF	SVDWKF	LFDETWKF	VREFWKF	IOWKF
位权限	R-0	U-0		R-0	R-0	R-0	R-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	I2CWKF	-	LPU1WKF	LPU0WKF	-		COMP2WKF	COMP1WKF
位权限	R-0	U-0	R-0	R-0	U-0		R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					LPTWKF	BSTWKF	DBGWKF
位权限	U-0					R-0	R-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	ADCWKF	ADC 中断唤醒标志，中断撤销时硬件自动清零(ADC 100lock100 Flag, auto to clear)
30:29	-	未实现，读为 0
28	RTCWKF	RTC 中断唤醒标志，中断撤销时硬件自动清零(RTC wakeup flag, auto to clear)
27	SVDWKF	SVD 中断唤醒标志，中断撤销时硬件自动清零(SVD wakeup flag, auto to clear)
26	LFDETWKF	32768Hz 晶体停振中断唤醒标志，中断撤销时硬件自动清零(XTLF fail detect wakeup flag, auto to clear)
25	VREFWKF	VREF1P22 基准源建立中断唤醒标志，中断撤销时硬件自动清零(Vref wakeup flag, auto to clear)
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零(GPIO wakeup flag, auto to clear)
23	I2CWKF	I2C 中断唤醒标志，中断撤销时硬件自动清零(I2C wakeup flag, auto to clear)
22	-	未实现，读为 0
21	LPU1WKF	LPUART1 中断唤醒标志，中断撤销时硬件自动清零(LPUART1 wakeup flag, auto to clear)
20	LPU0WKF	LPUART0 中断唤醒标志，中断撤销时硬件自动清零(LPUART0 wakeup flag, auto to clear)
19:18	-	未实现，读为 0
17	COMP2WKF	比较器 2 中断唤醒标志，中断撤销时硬件自动清零(comparator2 wakeup flag, auto to clear)
16	COMP1WKF	比较器 1 中断唤醒标志，中断撤销时硬件自动清零(comparator1 wakeup flag, auto to clear)
15:11	-	未实现，读为 0
10	LPTWKF	LPTIM32 中断唤醒标志，中断撤销时硬件自动清零(LPTIM wakeup flag, auto to clear)
9	BSTWKF	BSTIM32 中断唤醒标志，中断撤销时硬件自动清零(BSTIM

位号	助记符	功能描述
		wakeup flag, auto to clear)
8	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零(Debugger wakeup flag, write 1 to clear)
7:0	WKPxF	NWKUPx Pin 唤醒标志，软件写 1 清零(WKUP Pin wakeup flag,write 1 to clear)

4.5.4 PMU 中断使能寄存器（PMU_IER）

名称	PMU_IER							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LPRUNEIE	LPACTEIE	SLPEIE	RTCEIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	-	未实现，读为 0
3	**LPRUNEIE**	LPRUN 错误中断使能(LPRUN mode Error Interrupt Enable) 1：使能 LPRUN 错误中断 0：禁止 LPRUN 错误中断
2	**LPACTEIE**	LPACTIVE 错误中断使能 (LPACTIVE mode Error Interrupt Enable) 1：使能 LPACTIVE 错误中断 0：禁止 LPACTIVE 错误中断
1	**SLPEIE**	SLEEP 错误中断使能 (Sleep mode Error Interrupt Enable) 1：使能 SLEEP 错误中断 0：禁止 SLEEP 错误中断
0	**RTCEIE**	RTCBKP 错误中断使能(RTCBKP mode Error Interrupt Enable) 1：使能 RTCBKP 错误中断 0：禁止 RTCBKP 错误中断

4.5.5 PMU 中断标志寄存器（PMU_ISR）

名称	PMU_ISR							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LPRUNEIF	LPACTEIF	SLPEIF	RTCEIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	-	未实现，读为 0
3	**LPRUNEIF**	LPRUN 错误中断标志，硬件置位，软件写 1 清零(LPRUN Error Interrupt Flag,write 1 to clear) 1：在 PMOD=2‘h1 后，系统时钟不符合 LPRUN 模式的定义时置位 0：在 PMOD=2‘h1 后，系统时钟符合 LPRUN 模式的定义
2	**LPACTEIF**	LPACTIVE 错误中断标志，硬件置位，软件写 1 清零 (LPACTIVE Error Interrupt Flag,write 1 to clear) 1：在 LDO15LPM=1’h1 后，系统时钟不符合 LPACTIVE 模式的定义时置位，即系统时钟为 RCHF 且大于 4M，系统时钟为 XTHF/PLL 时 0：在 LDO15LPM=1’h1 后，系统时钟符合 LPACTIVE 模式的定义
1	**SLPEIF**	SLEEP 错误中断标志，硬件置位，软件写 1 清零 (Sleep Error Interrupt Flag,write 1 to clear) 1：在 PMOD=2’h2 后，CPU 执行 WFI/WFE 指令前置位了 SLEEPDEEP 寄存器时置位 0：在 PMOD=2’h2 后，CPU 正确进入 SLEEP
0	**RTCEIF**	RTCBKP 错误中断标志，硬件置位，软件写 1 清零(RTC Error Interrupt Flag,write 1 to clear) 1：在 PMOD=2‘h3 后，未改写 CPU 内部寄存器 SLEEPDEEP=1，然后执行 WFI/WFE 指令；或者系统时钟来自 USB PHY，试图进入 RTCBKP 模式 0：在 PMOD=2‘h3 后，CPU 自身正确进入 DEEP SLEEP

5 内部基准源（VREF）

5.1 概述

FM33LCx0 集成了一个高精度基准源，典型输出电压为 1.2V 左右，在芯片整个工作电源范围内都可以稳定工作。这个基准电压经过 Buffer 输出后，可以被 ADC 采样，也用于比较器的参考电压输入。

在整个工作温度范围内，此基准源的典型温度系数小于 25ppm/°C，同时内建了温度传感器输出，供 ADC 采样并测量当前芯片的基底温度。

软件可以开启或者关闭此基准源，打开基准源后，VREF1p2 输出建立时间小于 1ms，典型功耗为 1.5uA 左右。当打开温度传感器时，VREF1p2 功耗小于 2uA。

软件使能VREF1p2后，芯片内部有一个硬件延迟电路，在等待足够时间保证VREF输出完全建立后，置位VREF_DRY状态标志寄存器，并置位VREF_IF中断标志。软件可以自行定时或根据VREF_RDY寄存器来确认VREF1p2有效建立。

当软件关闭VREF1p2后，VREF_RDY寄存器被自动清零，VREF_IF由软件写1清零。

温度传感器最大支持的测温范围为-55~125°C，温度传感器输出电压随温度变化表现为一条正温度系数的直线，典型斜率为 5.1mV/°C。在芯片出厂前，温度传感器会在 30°C +/- 1°C 的条件下进行标定，在此条件下，-40~+85°C 范围内的温度测量误差在+/-1°C 以内。

5.2 寄存器

offset 地址	名称	符号
VREF(模块起始地址: 0x4001A80C)		
0x0000000C	VREF1p2 控制寄存器 （VREF Control Register）	VREF_CR
0x00000010	VREF1p2 标志寄存器 （VREF Status Register）	VREF_SR
0x00000014	VREF1p2 中断使能寄存器 （VREF Interrupt Enable Register）	VREF_IER
0x00000018	模拟 BUFFER 控制寄存器 （Buffer Control Register）	VREF_BUFCR

5.2.1 VREF1p2 控制寄存器（VREF_CR）

名称	VREF_CR							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PTAT_EN	VREF_EN
位权限	U-0						R/W-0	R/W-1

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	PTAT_EN	Bandgap 温度传感器使能 (Temperature sensor enable) 0：关闭温度传感器输出 1：使能温度传感器输出
0	VREF_EN	VREF1p2 使能寄存器 (Voltage reference enable) 0：关闭 VREF1p2 1：使能 VREF1p2

5.2.2 VREF1p2 标志寄存器（VREF_SR）

名称	VREF_SR							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							FLAG_B
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RDY	IF
位权限	U-0						R-0	R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	FLAG_B	模拟输出的 vref1p2 寄存器电压建立标志 (VREF setable Flag from analog, auto to clear)
7:2	-	RFU：未实现，读为 0
1	RDY	VREF1p2 基准电压建立标志 (VREF Ready Flag,auto to clear) VREF1p2 使能后，通过数字电路延迟置位，软件只读。关闭 VREF1p2 模块后，此寄存器自动清零。 数字电路延迟时间 1.5ms
0	IF	VREF1p2 基准电压建立中断 (VREF Ready Interrupt Flag, write 1 to clear) 0：VREF1p2 没有建立 1：VREF1p2 建立完成 此标志在 VREF1p2 使能后，通过数字电路延迟置位，硬件置位，软件写 1 清零

5.2.3 VREF1p2 中断使能寄存器（VREF_IER）

名称	VREF_IER							
offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	VREF1p2 基准电压建立中断使能 (VREF Ready Interrupt Enable) 0：禁止产生 VREF 建立完成中断 1：允许产生 VREF 建立完成中断 此寄存器为 1 的情况下，当 VREF1p2 建立完成后，将输出中断

位号	助记符	功能描述
		给 CPU

5.2.4 模拟 BUFFER 控制寄存器（VREF_BUFCR）

名称	VREF_BUFCR							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				VPTATBUFFER_OUTEN	VPTATBUFFER_EN	VREFBUFFER_OUTEN	VREFBUFFER_EN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3	VPTATBUFFER_OUTEN	Vptat Buffer 模块开关通道输出使能信号，高电平使能有效。(PTAT Buffer Output Enable)
2	VPTATBUFFER_EN	Vptat Buffer 模块使能信号，高电平使能有效。(PTAT Buffer Enable)
1	VREFBUFFER_OUTEN	Vref Buffer 模块开关通道输出使能信号，高电平使能有效。(VREF Buffer Output Enable)
0	VREFBUFFER_EN	Vref Buffer 模块使能信号，高电平使能有效。(VREF Buffer Enable)

6 CPU

6.1 概述

FM33LC0XX 使用的 CPU 内核为 Cortex-M0，符合 ARMv6-M 架构和编程模型；更多信息请参考 ARM 官网 www.arm.com

其基本特性如下：

- 用户/特权模式
- VTOR（中断向量表重定向）
- NVIC 支持 32 个外部中断
- 数据监视点：1
- 硬件断点：4
- 单周期 32-bit 硬件乘法器
- SWD 调试接口

6.1.1 处理器配置

Feature	Options	FM33LC0XX Config
Interrupts	1~32	32
Data endianness	little/big	little
SysTick Timer	Present or absent	Present
watchpoints	0,1,2	1
breakpoints	0,1,2,3,4	4
halting debug support	Present or absent	Present
multiplier	Fast or Small	Fast
Single-Cycle IO	Present or absent	Absent
wake-up interrupt controller(WIC)	Present or absent	Present
Vector Table Offset Register	Present or absent	Present
Unprivileged/Priviledged support	Present or absent	Present
JTAGnSW	JTAG or SWD for DAP	SWD
Memory Protection Unit	Present or absent	Absent

表 6-1FM33LC0xx CPU 配置简表

6.2 寄存器

主要内核寄存器列表

名字	描述
R0-R12	通用寄存器
MSP (R13)	堆栈指针；Handler 模式下使用 MSP（Main Stack Pointer），Thread 模式下通过 CONTROL 寄存器选择 MSP 或 PSP（Process Stack Pointer）使用
PSP (R13)	
LR (R14)	Link 寄存器，保存子函数/函数调用/异常处理的返回信息
PC (R15)	程序指针
PSR	包含应用程序状态（APSR）、中断程序状态（IPSR）和程序执行状态（EPSR）
PRIMASK	PRIMASK 用于屏蔽指定优先级及以下的所有中断响应
CONTROL	设置 Thread 模式下使用的堆栈指针

表 6-2Cortex-M0 内核寄存器简表

寄存器详细定义参见 ARMv6-M 架构参考手册。

6.3 异常和中断

内核的异常和中断管理通过 NVIC 完成。NVIC 的可编程管理寄存器位于 PPB 总线的 SCS 空间内，NVIC 具有如下特性：

- 支持 32 个外部中断，5 个内部异常
- 1 个 NMI 中断
- 支持中断嵌套
- 向量化的异常入口
- 中断屏蔽

处理器内核接受一个异常请求后，首先会将内核寄存器 R0~R3、R12、R14、PC、xPSR 压入堆栈。链接寄存器 LR（R14）被更新为异常返回时使用的特殊值（EXC_RETURN），然后根据异常向量表定位异常处理程序开始执行。注意在异常处理中没有被自动压栈的寄存器，必须通过软件来保存和恢复。

6.3.1 中断向量表

Position	Priority	Priority type	Acronym	Description	Address
0	-	-	MSP 初值	主栈指针初始化地址	0x0000_0000
1	-3	fixed	Reset	复位向量	0x0000_0004
2	-2	fixed	NMI	WKUPx 中断 低功耗模式错误中断	0x0000_0008
3	-1	fixed	HardFault	HardFault 中断向量	0x0000_000C
4-10	-	-	-	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall 系统服务请求	0x0000_002C
12-13	-	-	-	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	可挂起系统服务请求	0x0000_0038
15	6	settable	Systick	内部定时器中断向量	0x0000_003C
16	7	settable	WWDT	窗口看门狗中断	0x0000_0040
17	8	settable	SVD	电源监测报警中断	0x0000_0044
18	9	settable	RTC	实时时钟中断	0x0000_0048
19	10	settable	FLASH	NVMIF 中断	0x0000_004C
20	11	settable	LFDET	XTLF 或 XTHF 停振检测中断	0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable	IWDT	IWDT 中断	0x0000_0058
23	14	settable	SPI1	SPI 中断	0x0000_005C
24	15	settable	SPI2		0x0000_0060
25	16	settable	LCD	LCD 显示模块中断	0x0000_0064

Position	Priority	Priority type	Acronym	Description	Address
26	17	settable	UART0	UART 中断	0x0000_0068
27	18	settable	UART1		0x0000_006C
28	19	settable	UART4		0x0000_0070
29	20	settable	UART5		0x0000_0074
30	21	settable	HFDET	高频晶体停振检测中断	0x0000_0078
31	22	settable	U7816	U7816 中断	0x0000_007C
32	23	settable	LPUART1	LPUART1 中断	0x0000_0080
33	24	settable	I2C	I2C 中断	0x0000_0084
34	25	settable	USB	USB device 中断	0x0000_0088
35	26	settable	AES	AES 中断	0x0000_008C
36	27	settable	LPTIM	低功耗定时器中断	0x0000_0090
37	28	settable	DMA	DMA 中断	0x0000_0094
38	29	settable	WKUP	WKUP 引脚中断	0x0000_0098
39	30	settable	OPAx	OPAx 中断	0x0000_009C
40	31	settable	BSTIM	基本定时器中断	0x0000_00A0
41	32	settable	COMPx	COMPx 中断	0x0000_00A4
42	33	settable	GPTIM0	通用定时器 1 中断	0x0000_00A8
43	34	settable	GPTIM1	通用定时器 2 中断	0x0000_00AC
44	35	settable	ATIM	高级定时器中断	0x0000_00B0
45	36	settable	VREF	1.2V 内部基准电压建立中断	0x0000_00B4
46	37	settable	GPIO	外部引脚中断	0x0000_00B8
47	38	settable	LPUART0	LPUART0 中断	0x0000_00BC

表 6-3FM33LC0xx 中断向量表

其中WKUP中断可以接到NMI或者38#入口。通过PMU模块的WKSEL寄存器来选择中断入口地址。当配置为38#入口时，可以通过PRIMASK将WKUPx中断屏蔽，唤醒后CPU不进入中断服务程序，而是继续从休眠指令处向下执行。

6.3.2 中断优先级

处理器支持 3 个固定的最高优先级及 4 个可编程优先级。当两个相同优先级的异常同时发生，则异常编号较小的异常将被首先执行。

6.3.3 错误处理

处理器只支持一种硬件错误处理方式：HardFault 异常。HardFault 优先级-1，只有 NMI 能对其抢占。

HardFault 的触发原因包含以下几种情况：

错误类型	错误条件
存储器相关	总线错误。由于在总线传输中使用了非法地址而产生的总线错误。
	试图在 XN 区域内执行程序
程序错误	执行未定义的指令
	试图切换至 ARM 状态

	试图进行非对齐的存储器访问 在更高优先级异常处理中执行 SVC 指令 执行异常返回时 EXC_RETURN 的值非法 当调试未使能时试图执行 BKPT 指令

表 6-4Hardfault 触发原因列表

FM33LC0XX 的 HardFault 触发原因可以通过寄存器查询，以帮助软件开发者定位错误原因。

6.3.4 锁定（Lockup）

当处理器在进行 HardFault 处理的过程中发生了另一个 HardFault，或者 NMI 处理期间发生了 HardFault，则处理器将进入锁定状态（停止执行），并输出 LOCKUP 信号，此时芯片将自动复位处理器内核，而不是等待看门狗溢出。

6.4 调试特性

处理器支持以下调试特性

- 程序的暂停、恢复及单步执行
- 访问内核寄存器和特殊寄存器
- 硬件断点（4 个）
- 软件断点（不限数量的 BKPT 指令）
- 数据监视点（1 个）
- 动态非侵入式存储器访问（无需停止处理器）
- SWD 接口

Cortex-M0 的调试特性是基于 ARM CoreSight 调试架构的，详情请参考《CoreSight Technology System Design Guide》和《ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2》

6.4.1 调试功能引脚

FM33LC0XX 使用 SWD 调试接口，用户模式下最少仅需 4 线（NRST, GND, SWIO, SWCLK）即可实现调试功能。2 线调试引脚可以复用为 GPIO，其功能由软件选择配置。

NRST 引脚用于复位芯片，通过 NRST 与 SWD 的配合，可以使芯片复位后 Halt 在第一条指令处。

调试功能引脚的复用说明参见 I/O 控制章节。

6.4.2 调试状态下的看门狗控制

看门狗在调试模式下可以保持使能或关闭。软件或 Debugger 可以通过 MCUDBGCR 寄存器配置看门狗打开或关闭。

6.4.3 DEBUG 的复位

内核的 DEBUG 部分仅受上下电复位影响，其他系统复位源如看门狗、引脚复位、软件复位等，都不会复位 DAP 电路。这样可以在芯片上电后通过引脚复位使 CPU 内核处于复位状态，但是调试器仍可以正常与 DAP 建立通信并设置断点，在复位放开后可以使 CPU 立即进入调试模式。

建议调试器在系统复位时连接内核(在复位向量处设置断点)。

6.5 寄存器

offset 地址	名称	符号
DBG(模块起始地址：0x40000000)		
0x00000004	DEBUG 配置寄存器 （Debug Configuration Register）	DBG_CR
0x00000008	HardFault 查询寄存器 （HardFault Flag Register）	DBG_HDFR

6.5.1 DEBUG 配置寄存器（DBG_CR）

FM33LC0XX扩展了MCUDBGCR寄存器，用于配置Debug状态下的看门狗和定时器。MCUDBGCR寄存器可以由 SWD 接口或软件改写。

名称	DBG_CR							
offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		AT_STOP	LPT_STOP	GT2_STOP	GT1_STOP	-	BT1_STOP
位权限	U-0		R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						WWDT_STOP	IWDT_STOP
位权限	U-0						R/W-1	R/W-1

位号	助记符	功能描述
31:14	-	RFU：未实现，读为 0
13	AT_STOP	Debug 状态下 ATIM 使能控制位 (Stop ATIM under Debug Enable) 1：Debug 时关闭 ATIM 0：Debug 时保持 ATIM 原来状态
12	LPT_STOP	Debug 状态下 LPTIM32 使能控制位 (Stop LPTIM32 under Debug Enable) 1：Debug 时关闭 LPTIM32 0：Debug 时保持 LPTIM32 原来状态
11	GT1_STOP	Debug 状态下 GPTIM1 使能控制位 (Stop GPTIM1 under Debug Enable) 1：Debug 时关闭 GPTIM1 0：Debug 时保持 GPTIM1 原来状态
10	GT0_STOP	Debug 状态下 GPTIM0 使能控制位 (Stop GPTIM0 under Debug Enable) 1：Debug 时关闭 GPTIM0 0：Debug 时保持 GPTIM0 原来状态

位号	助记符	功能描述
9	-	RFU：未实现，读为 0
8	BT_STOP	Debug 状态下 BSTIM32 使能控制位 (Stop BSTIM under Debug Enable) 1：Debug 时关闭 BSTIM32 0：Debug 时保持 BSTIM32 原来状态
7:2	-	RFU：未实现，读为 0
1	WWDT_STOP	Debug 状态下 WWDT 使能控制位 (Stop WWDT under Debug Enable) 1：Debug 时关闭 WWDT 0：Debug 时保持 WWDT 原来状态
0	IWDT_STOP	Debug 状态下 IWDT 使能控制位 (Stop IWDT under Debug Enable) 1：Debug 时关闭 IWDT 0：Debug 时保持 IWDT 开启

6.5.2 HardFault 查询寄存器（DBG_HDFR）

名称	DBG_HDFR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	SVCUNDEF_FLAG	BKPT_FLAG	TBIT_FLAG	SPECIAL_OP_FLAG	HDF_REQUEST_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	DABORT_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 (Debug Abort Flag for misaligned Address) 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 (Debug Abort Flag for HRESP) 1：总线传输中访问了非法地址导致 HRESP 为高产生错误 0：未访问非法地址
4	SVCUNDEF_FLAG	SVC instructions 未定义标志，写 1 清零 (SVC undefined instruction Flag) if the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set,

位号	助记符	功能描述
		the core should treat SVC instructions as though theywere UNDEFINED。
3	BKPT_FLAG	执行 BKPT 指令标志，写 1 清零 (Break point instruction Flag) 1：执行了 BKPT 指令 0：未执行 BKPT 指令
2	TBIT_FLAG	Thumb-State 标志，写 1 清零 (Thumb state Flag) 1：切换到 ARM 状态 0：处于 Thumb-State
1	SPECIAL_OP_FLAG	特殊指令标志，写 1 清零 (Special OP code Flag) 1：执行了特殊指令代码，如试图在 XN 区域内取指 0：无特殊指令代码被执行
0	HDF_REQUEST_FLAG	hardfault 标志位，任何类型的 hardfault 都会导致该位置位，写 1 清零 (Hardfault Request Flag) 1：hardfault 请求 0：无 hardfault 请求

7 总线与存储

7.1 系统总线

FM33LC0 总线架构包含以下主要部件：

- 两个 Master
  - Cortex-M0 内核
  - DMA 控制器
- 五个 Slave
  - 内部 Flash 存储器
  - 内部 SRAM 存储器
  - GPIO 控制器模块
  - 系统控制模块
  - AHB-APB 总线转接桥

FM33LC0XX 的系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

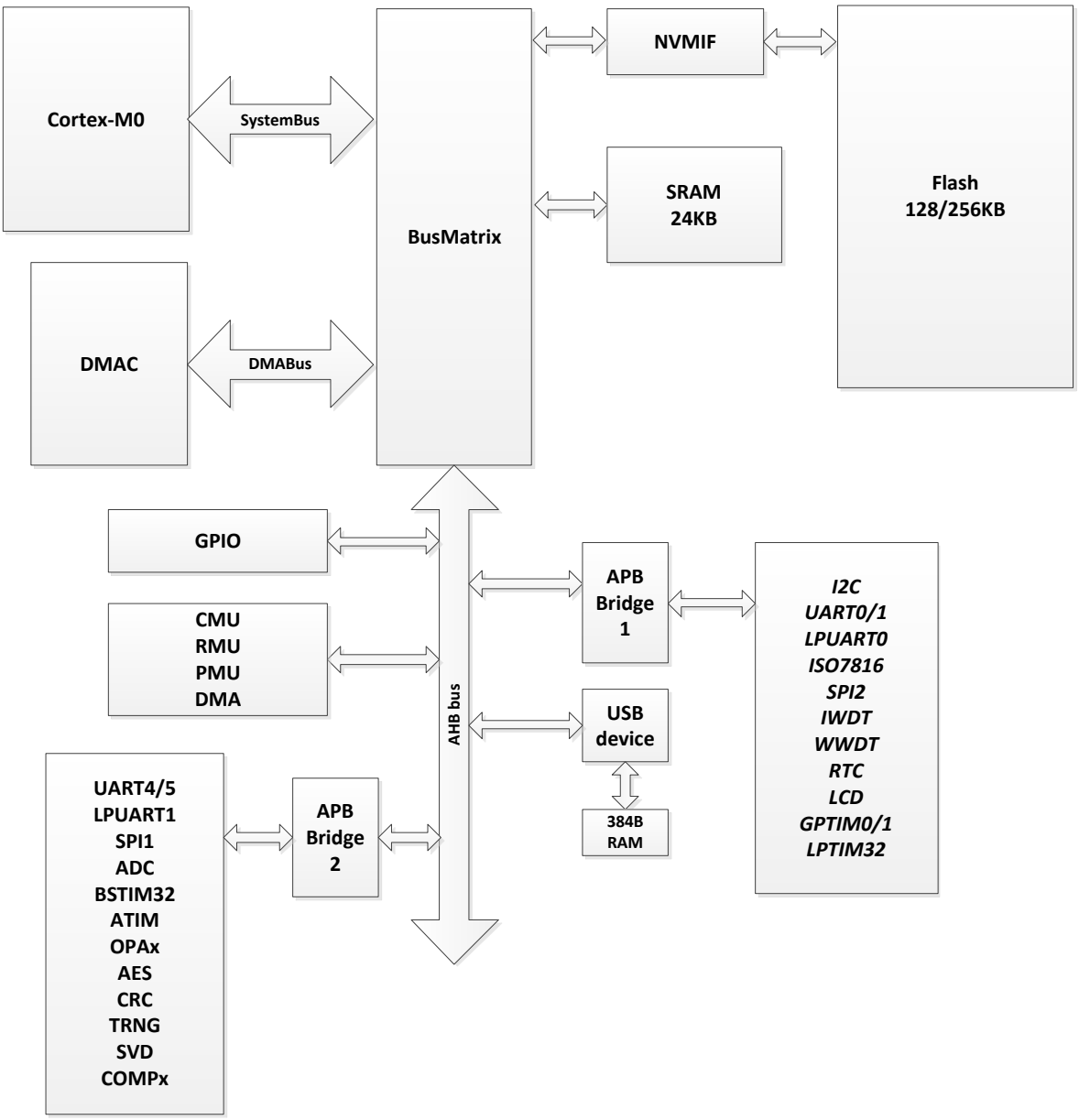

Cortex-M0
SystemBus
BusMatrix
NVMIF
SRAM
24KB
Flash
128/256KB
DMAC
DMABus
GPIO
CMU
RMU
PMU
DMA
AHB bus
APB
Bridge
1
USB
device
384B
RAM
I2C
UART0/1
LPUART0
ISO7816
SPI2
IWDT
WWDT
RTC
LCD
GPTIM0/1
LPTIM32
UART4/5
LPUART1
SPI1
ADC
BSTIM32
ATIM
OPAx
AES
CRC
TRNG
SVD
COMPx
APB
Bridge
2

图 7-1 系统总线示意图

7.2 存储空间分配

7.2.1 概述

Flash 扇区（Sector）大小为 512 字节，每 16 个扇区组成一个 8K 字节的块（Block）。

Flash 包含 4 个 information 扇区，2 个 LDT 扇区，1 个冗余扇区，1 个 DCT 扇区。其中 DCT 和 LDT 为芯片原厂保留扇区，不对用户开放。Information 为用户配置扇区，用于保存用户配置信息。所有 option 扇区在地址上与 Flash 主区域互相隔离。

当芯片从 Flash 启动时，FM33LC0XX 的地址空间分配如下图（256KB Flash，24KB RAM）：

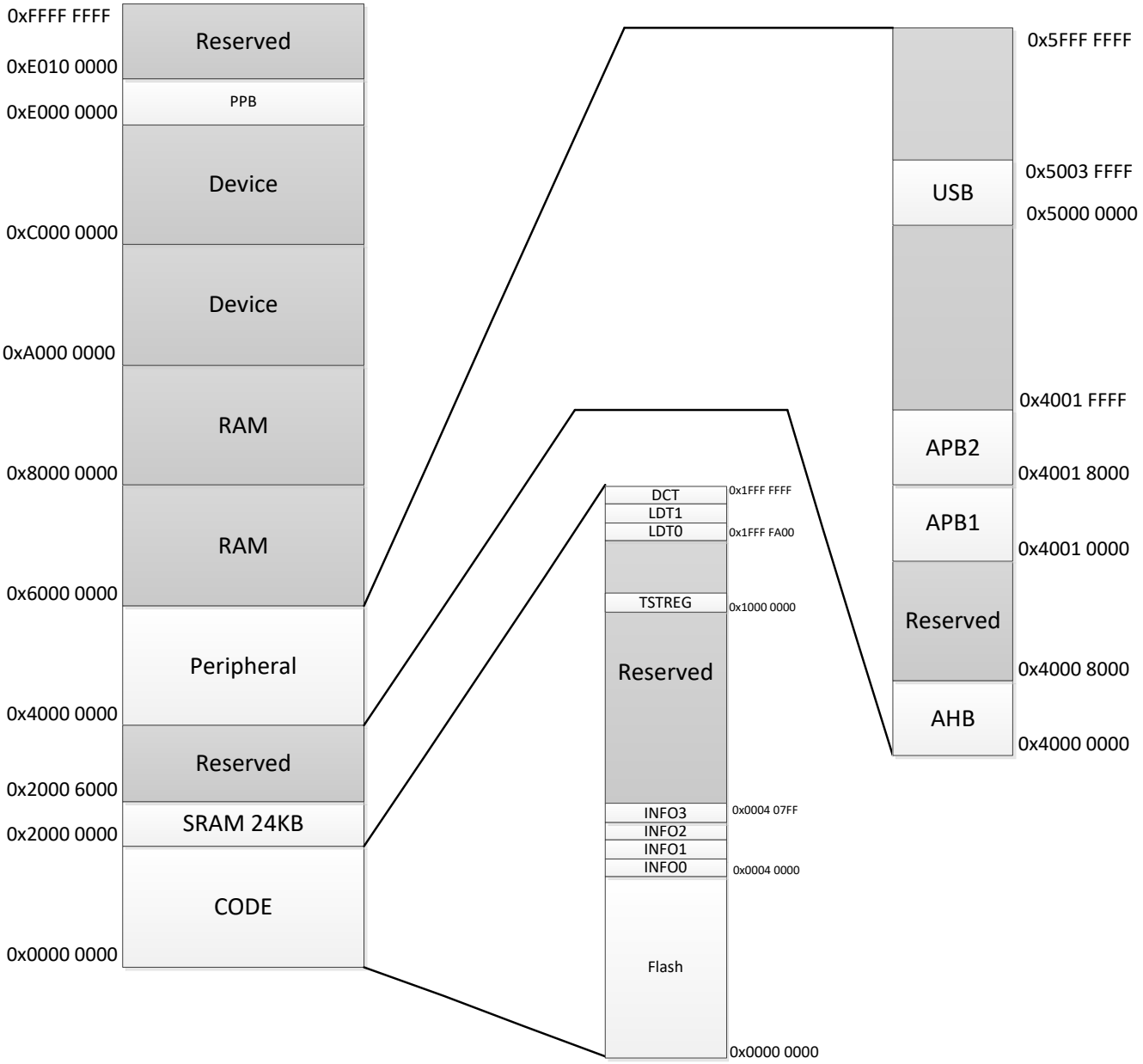


图 7-2FM33LC04x 总线地址

128KB flash+24KB RAM：

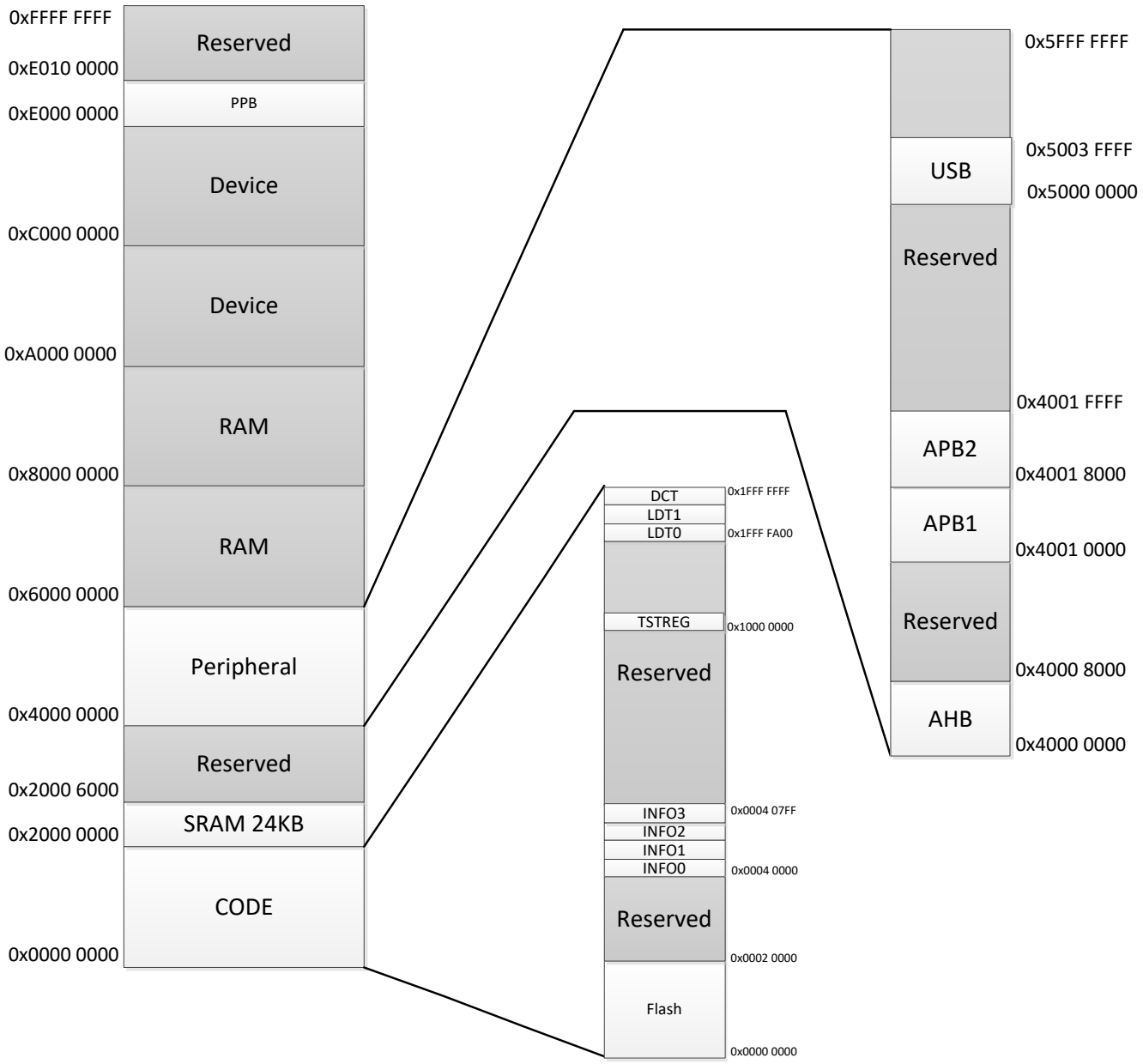


图 7-3FM33LC02x 总线地址

7.2.2 外设模块寄存器地址分配

下表罗列了所有外设模块的地址空间分配范围，每个外设模块占用 1KB 地址空间。

总线	地址边界	空间	外设
AHB	0x4000_0C00~0x4000_0FFF	1KB	GPIO
	0x4000_0000~0x4000_03FF	1KB	SCU, PMU, CMU, RMU
	0x4000_0400~0x4000_07FF	1KB	DMA
	0x4000_1000~0x4000_13FF	1KB	NVMIF
	0x5000_0000~0x5001_FFFF	128KB	USB Controller
	0x5002_0000~0x5003_FFFF	128KB	USB packet RAM
	0x0000_0000~0x0003_FFFF	256KB	Flash main array

	0x1FFF_F000~0x1FFF_FFFF	4KB	Flash Option cell array
	0x2000_0000~0x2000_5FFF	24KB	SRAM
	0x4001_0000~0x4001_7FFF	32KB	APB1
	0x4001_8000~0x4001_FFFF	32KB	APB2
APB1	0x4001_0000~0x4001_03FF	1KB	ISO7816
	0x4001_0400~0x4001_07FF	1KB	LPUART0
	0x4001_0800~0x4001_0BFF	1KB	SPI2
	0x4001_0C00~0x4001_0FFF	1KB	LCD
	0x4001_1000~0x4001_13FF	1KB	RTC
	0x4001_1400~0x4001_17FF	1KB	IWDT
	0x4001_1800~0x4001_1BFF	1KB	WWDT
	0x4001_1C00~0x4001_1FFF	1KB	UART0
	0x4001_2000~0x4001_23FF	1KB	UART1
	0x4001_2400~0x4001_27FF	1KB	I2C
	0x4001_2800~0x4001_2BFF		Reserved
	0x4001_2C00~0x4001_2FFF	1KB	RAMBIST
	0x4001_3000~0x4001_33FF		Reserved
	0x4001_3400~0x4001_37FF	1KB	LPTIM32
	0x4001_3800~0x4001_3BFF	1KB	GTIMER0
	0x4001_3C00~0x4001_3FFF	1KB	GTIMER1
APB2	0x4001_8000~0x4001_83FF	1KB	CRC
	0x4001_8400~0x4001_87FF	1KB	LPUART1
	0x4001_8800~0x4001_8BFF		Reserved
	0x4001_8C00~0x4001_8FFF	1KB	SPI1
	0x4001_9000~0x4001_93FF		Reserved
	0x4001_9400~0x4001_97FF		Reserved
	0x4001_9800~0x4001_9BFF		Reserved
	0x4001_9C00~0x4001_9FFF		Reserved
	0x4001_A000~0x4001_A3FF	1KB	UART4
	0x4001_A400~0x4001_A7FF	1KB	UART5
	0x4001_A800~0x4001_ABFF	1KB	SVD,OPA,COMP
	0x4001_AC00~0x4001_AFFF	1KB	ADC
	0x4001_B000~0x4001_B3FF	1KB	ATIM
	0x4001_B400~0x4001_B7FF	1KB	BSTIM32
	0x4001_B800~0x4001_BBFF	1KB	AES
	0x4001_BC00~0x4001_BFFF	1KB	TRNG

表 7-1 外设模块总线地址列表

7.3 RAM

7.3.1 概述

FM33LC0XX 含有一块 24KB RAM（6K*32），地址空间范围是 0x2000_0000~0x2000_5FFF，软件可以对 SRAM 进行字节、半字、字访问，CPU 和 DMA 都可以以最大系统频率对 SRAM 实现无等待的单周期读写。CPU 也可以从 SRAM 取指执行程序，因此在对程序效率要求高的场合，可以将部分代码导入 SRAM 中，实现最高频率下无等待的执行。

7.4 Flash

7.4.1 概述

FM33LC0XX 使用的 Flash 容量为 64K*32，即 256KB；阵列组织格式包含 page(512B)、sector(2KB)。

Main array 共包含 512 个扇区，支持页擦、扇区擦和全擦。

7.4.2 特殊信息扇区说明

Flash 的 main array 以外还有几个特殊扇区可供用户使用，说明如下：

区域	说明	用途
LDT1	用户选项数据区	用户选项字节（OPTBYTES）
IF	Information 区	4 个 page 共 2KB，供用户使用

7.4.2.1 LDT1 page

LDT1为用户配置信息区，可以在用户模式下改写（仅能使用SWD改写，即用户通过编程器改写）。

LDT1的总线地址是0x1FFF_FC00~0x1FFF_FDFF；Flash内部物理地址是(MCS=1,LDTEN[1:0]=10, ADD=0x0000~0x007F)

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FC00	~OPTBYTES[15:0]	OPTBYTES[15:0]	用户选项字节低半字
0x1FFF_FC04	~OPTBYTES[31:16]	OPTBYTES[31:16]	用户选项字节高半字
0x1FFF_FC08	LOCK1		ACLOCK 配置字，控制低 16 blocks
0x1FFF_FC0C	LOCK2		ACLOCK 配置字，控制高 16 blocks

表 7-2LDT1 数据内容定义

OPTBYTES 选项字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:24	BTSWPEN	启动区交换使能 0x55：允许启动区交换功能 其他：禁止启动区交换	0xFF
23:20	IWDTSLP	配置 IWDT 在低功耗模式下是否允许停止计数 0xA：在 Sleep/DeepSleep/RTCBKP 模式下允许应用停止 IWDT 计数 其他：任何模式下禁止应用停止 IWDT	0xF
19:16	DFLSEN	Data flash 使能 0x5：使能数据 flash，main array 的最高 16KB	0xF

Bitfield	助记符	功能描述	出厂默认
		地址被定义为 data flash 其他：禁止数据 flash	
15:8	ACLKEN	应用代码保护使能 0x33：禁止 ACLOCK 其他：使能 ACLOCK	0x33
7:0	DBRDPEN	调试接口访问保护使能 0xAA：关闭调试接口保护 其他：使能调试接口保护	0xAA

表 7-3 用户选项字节定义

【注】在出厂时，用户软件或者 SWD 接口可以任意改写 OPTBYTES；但是一旦 ACLKEN 或 DBRDP 被使能，用户必须通过 SWD 全擦 flash 后才能重新改写 OPTBYTES。

LOCK 配置字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:0	LOCK1	Block Lock 字 1，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK1[1:0]对应 Block0（Flash 最低地址 8KB 空间），LOCK1[31:30] 对应 Block15（Flash 地址空间 120~128KB），其他以此类推	0xFFFFFFFF
31:0	LOCK2	Block Lock 字 2，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK2[1:0] 对应 Block16（Flash 地址空间 128~136KB），LOCK2[31:30]对应 Block31（Flash 地址空间 248~256KB），其他以此类推	0xFFFFFFFF

表 7-4LOCK 信息定义

LOCK bit 和 Flash 权限锁定地址的对应关系如下表：

Address	LOCK bits
0x0000_0000 ~ 0x0000_1FFF	LOCK1[1:0]
0x0000_2000 ~ 0x0000_3FFF	LOCK1[3:2]
0x0000_4000 ~ 0x0000_5FFF	LOCK1[5:4]
0x0000_6000 ~ 0x0000_7FFF	LOCK1[7:6]
0x0000_8000 ~ 0x0000_9FFF	LOCK1[9:8]
0x0000_A000 ~ 0x0000_BFFF	LOCK1[11:10]
0x0000_C000 ~ 0x0000_DFFF	LOCK1[13:12]
0x0000_E000 ~ 0x0000_FFFF	LOCK1[15:14]
0x0001_0000 ~ 0x0001_1FFF	LOCK1[17:16]
0x0001_2000 ~ 0x0001_3FFF	LOCK1[19:18]
0x0001_4000 ~ 0x0001_5FFF	LOCK1[21:20]

Address	LOCK bits
0x0001_6000 ~ 0x0001_7FFF	LOCK1[23:22]
0x0001_8000 ~ 0x0001_9FFF	LOCK1[25:24]
0x0001_A000 ~ 0x0001_BFFF	LOCK1[27:26]
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]
0x0002_0000 ~ 0x0002_1FFF	LOCK2[1:0]
0x0002_2000 ~ 0x0002_3FFF	LOCK2[3:2]
0x0002_4000 ~ 0x0002_5FFF	LOCK2[5:4]
0x0002_6000 ~ 0x0002_7FFF	LOCK2[7:6]
0x0002_8000 ~ 0x0002_9FFF	LOCK2[9:8]
0x0002_A000 ~ 0x0002_BFFF	LOCK2[11:10]
0x0002_C000 ~ 0x0002_DFFF	LOCK2[13:12]
0x0002_E000 ~ 0x0002_FFFF	LOCK2[15:14]
0x0003_0000 ~ 0x0003_1FFF	LOCK2[17:16]
0x0003_2000 ~ 0x0003_3FFF	LOCK2[19:18]
0x0003_4000 ~ 0x0003_5FFF	LOCK2[21:20]
0x0003_6000 ~ 0x0003_7FFF	LOCK2[23:22]
0x0003_8000 ~ 0x0003_9FFF	LOCK2[25:24]
0x0003_A000 ~ 0x0003_BFFF	LOCK2[27:26]
0x0003_C000 ~ 0x0003_DFFF	LOCK2[29:28]
0x0003_E000 ~ 0x0003_FFFF	LOCK2[31:30]

表 7-5LOCK 位与 Flash 地址对应表

7.4.2.2 Information3 page

Flash还包含4个information扇区，其中information3用于控制BootSwap功能。Information3总线地址是0x0004_0600~0x0004_07FF；

在LDT1中BOOTSWAPEN=0x55的前提下，可以通过INFO3中最低地址数据内容来进行BootSwap操作。当数据为0x5454_ABAB时，芯片将Flash最低两个8KB空间的逻辑地址互换（注意，ACLOCK只按照逻辑地址处理，不考虑实际物理地址），从而实现启动代码无风险升级。

BootSwap示意图如下：

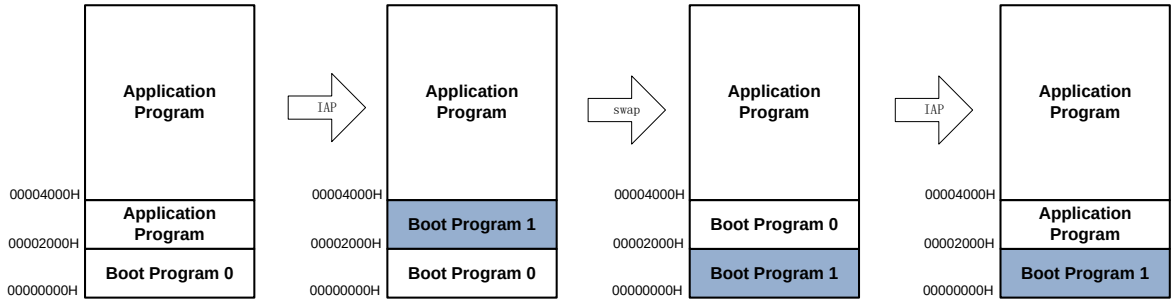


图 7-4Bootswap 示意图

其主要目的是，防止在系统更新启动代码时出现意外中断（停电、异常复位等），如果此时原来的启动代码已经被擦除，将导致芯片重启后无法正常运行。

实现BootSwap功能后，假设启动代码占据0000~1FFF共8KB空间，系统升级时应先将新的启动代码写入2000~3FFF地址，然后使能BootSwap。此时有几种可能性：

- 芯片擦写2000~3FFF地址时掉电，由于原来的启动代码还在，不会影响重启
- 芯片成功写入boot program1，然后使能bootswap并执行软复位，芯片重启后将执行boot program1
- 芯片擦写boot program0时掉电，由于boot program1已经写入，将不影响后续运行

推荐应用按照如下步骤升级：

- 更新application program
- 如需升级boot，先将新的boot程序写入第二个8KB空间
- 配置INFO3，使能BootSwap
- 执行软复位，重启后执行新的boot程序
- 将第二个8KB空间改写为新的应用程序

逻辑地址对Flash物理地址的重映射由芯片自动完成，不论程序还是Debugger都以逻辑地址进行访问。

注意：IF3 的 BootSwap 功能实际只用到本页的最低一个 word，其余地址空间开放读写（无特定功能），软件或 Debugger 都可以任意写入数据。

7.4.2.3 Information1~2 page（Debugger only, lockable）

这2个information扇区开放给用户使用，仅SWD可以擦写，软件可读。

总线地址是0x0004_0200~0x0004_05FF，低地址为IF1，高地址为IF2，总共1KB。

IF2~1这2个扇区各自的最高地址字节为扇区锁定标志，如果SWD将最高地址字节改写为0x55，则芯片复位后当前扇区被禁止编程，SWD进行扇区擦后，可以重新编程。

不论是否有锁定标志，这几个扇区都是SWD和软件可读的。

7.4.2.4 Information0 page（OTP）

IF0是一个OTP page，这个page在用户模式下只能编程一次，编程后不能擦除或改写。IF0的总线地址是0x0004_0000~0x0004_01FF，共512字节。

对IF0 page的读取没有任何限制。

7.4.3 Flash 编程

7.4.3.1 概述

FM33LC0XX 支持以下 Flash 编程方法：

- 在系统编程（ISP）：通过 FMSH 专用编程器或者在线仿真实施芯片编程，使用 SWD 接口
- 在应用编程（IAP）：通过 bootloader 代码实现芯片自编程，用户可定义任意通信接口，可用于实现程序在线升级

编程前必须对 Flash 进行擦除。Flash 支持三种擦除操作：全擦、扇区擦、页擦

7.4.3.2 Flash 擦写时钟

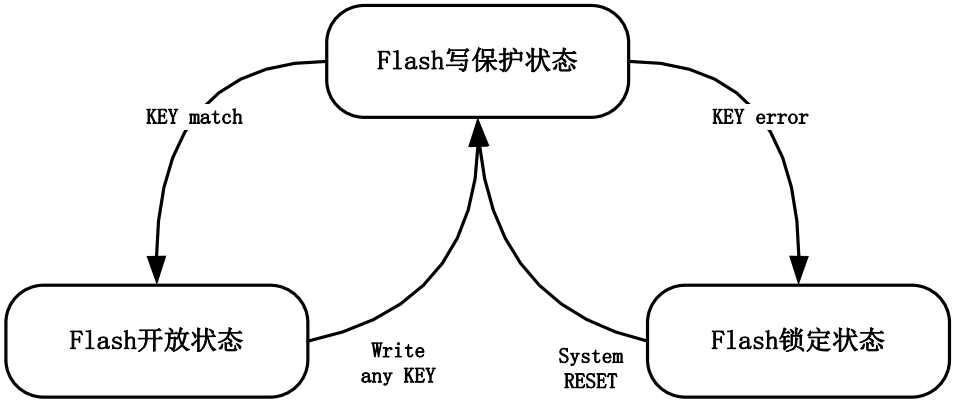
执行 Flash 擦写时使用校准后的 RCHF 时钟，但是系统时钟可以是任意时钟。编程支持的 RCHF 频率为 8M、16M 和 24M。

Flash 擦写时钟独立于 CPU 时钟，因此不会影响程序运行。

7.4.3.3 Flash 擦写方法

FM33LC0XX 支持 Flash 擦除操作，以及单次编程和连续编程。

Flash 擦写前须进行 Key 校验，写入顺序错误或写入值错误，或者在 Flash Key 验证正确之前就进行擦除或编程 Flash 操作将会进入错误状态，并产生相应中断。Flash Key 认证错误之后将禁止擦写 Flash 直到下一次复位。而在正常擦写完成后，向 KEY 寄存器写入任意值都会使状态机返回初始的写保护状态。状态转换如下图：



软件可以通过查询 FLSIF.KEYSTA 来确认当前 Key 输入状态，详情参见寄存器说明。

7.4.3.4 全擦操作（Matrix Erase）

全擦操作只能由 SWD 接口启动，软件禁止进行全擦。全擦操作仅擦除 main array，不会擦除特殊

信息扇区。SWD 可以在制造商或用户模式下启动全擦，操作流程如下：

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D
- SWD 向 Flash 任意地址写擦除请求 0x1234_ABCD
- 芯片启动对 Flash 的全擦，并暂停任何 Master 对 Flash 的访问
- 全擦完成后置位中断标志和全擦标志（全擦标志表示 main array 全部擦除，任何对 main array 的编程将清除此标志）
- 在全擦标志有效的情况下，SWD 可以任意擦写 LDT1，否则擦写 LDT1 被禁止并触发错误中断
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：全擦操作只能擦除Flash的main array，不会影响特殊信息区

7.4.3.5 扇区擦操作（Sector Erase）

SWD 和应用代码都可以执行扇区擦。操作流程如下：

- 配置 ERTYPE 寄存器为 01
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0x3C3C_3C3C
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- 芯片检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

7.4.3.6 页擦操作（Pgae Erase）

SWD 和应用代码都可以执行页擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00 或 11
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的扇区内任意地址写擦除请求 0x1234_ABCD
- 芯片检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志

- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

7.4.3.7 单次编程

单次编程由软件发起，通过总线直接写 Flash，编程最小单位为 32bit，操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 清除连续编程使能寄存器
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 向 Flash 目标地址写数据，如果目标地址被 ACLOCK 锁定，则触发错误标志，如果没有锁定，则执行编程
- 编程完成后置位中断标志
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

7.4.3.8 连续编程

连续编程指通过 DMA 的 Memory 通道一次向 Flash 写入 half-sector（256 字节）。连续编程时 DMA 从 RAM 指定地址读取数据，Flash 目标编程地址必须是 half-sector 对齐的，也就是 Flash 地址低 6 位为 0。采用这种方式时一次编程的数据长度是固定的，主要用于快速大数据量写入。

在启动连续编程期间，DMA 完全占据 Flash 总线，暂停 CPU 对 Flash 的一切访问。连续编程的操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 置位连续编程使能寄存器（DMA 模式使能）
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- 芯片检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 FlashKEY 寄存器写任意值恢复写保护

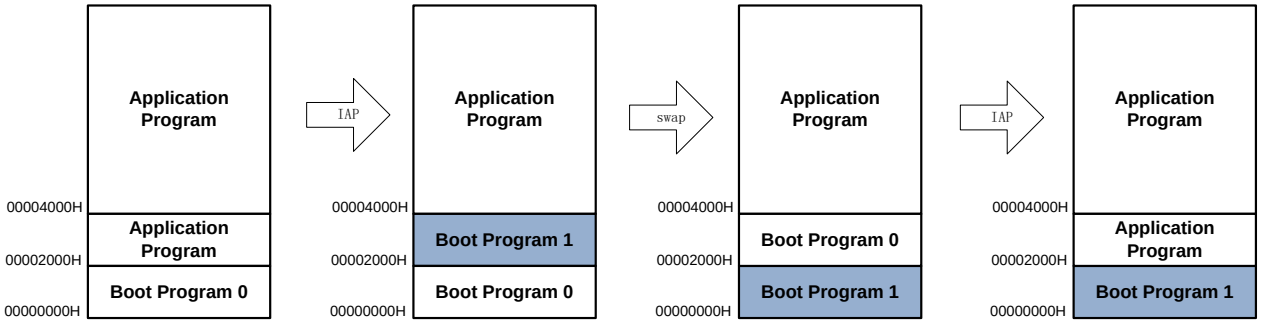
注意：如果 CPU 在 Flash 中取指时进行 Flash 擦写，则 CPU 取指将被暂停，直到擦写操作完成。如果 CPU 跳转到 RAM 中取指运行，则 Flash 擦写不会暂停 CPU 的执行。Flash 擦写过程中，若用

户希望在 RAM 中执行代码时仍然能够实时响应中断，应将中断向量表重新映射到 RAM 中。

7.4.3.9 启动区交换（BootSwap）

BootSwap主要目的是，防止在系统更新启动代码时出现意外中断（停电、异常复位等），如果此时原来的启动代码已经被擦除，将导致芯片重启后无法正常运行。BootSwap功能通过编程information page3最低地址word实现。

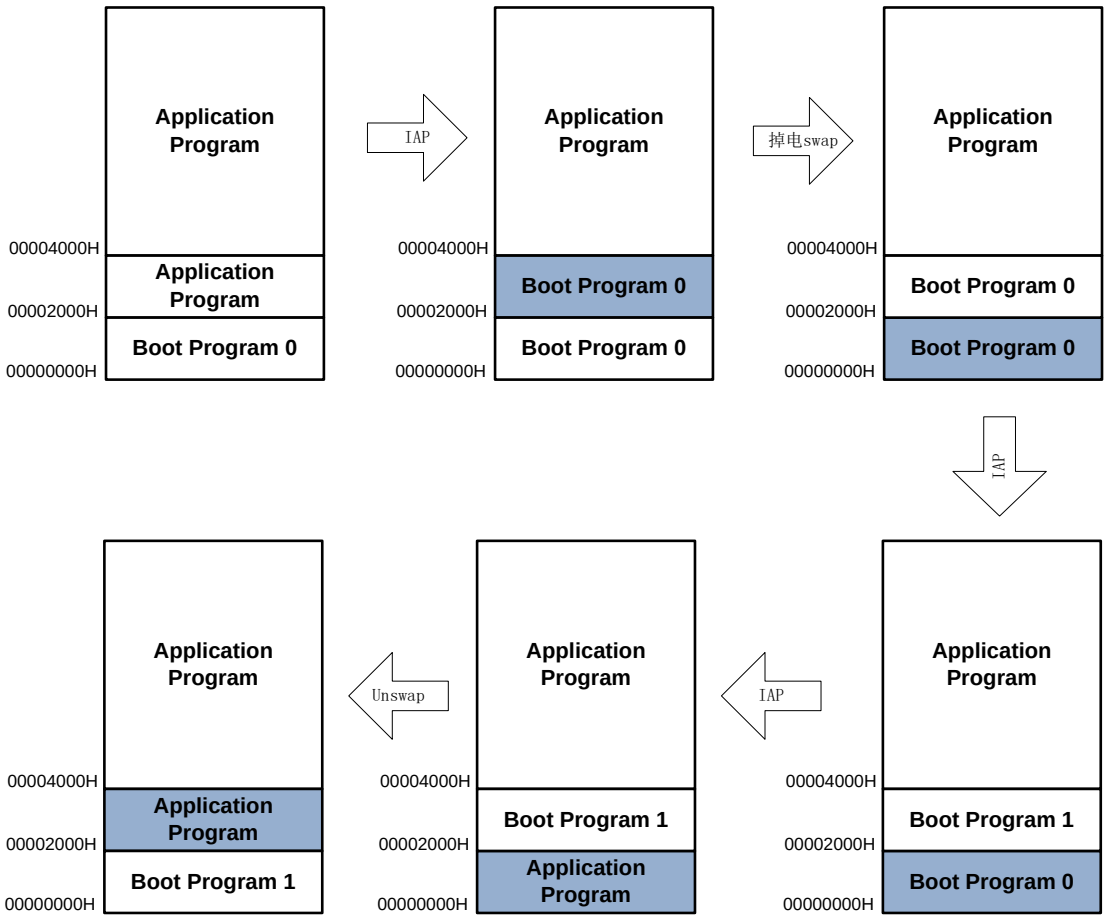
BootSwap示意图如下：



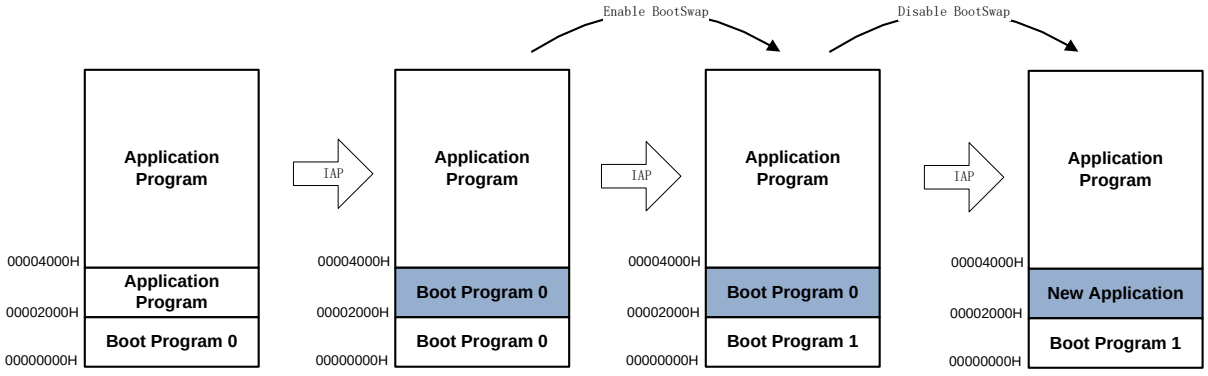
实现BootSwap功能后，假设启动代码占据0000~1FFF共8KB空间，系统升级时应先将新的启动代码写入2000~3FFF地址，然后使能BootSwap。此时有几种可能性：

- 芯片擦写2000~3FFF地址时掉电，由于原来的启动代码还在，不会影响重启
- 芯片成功写入boot program1，然后使能bootswap并执行软复位，芯片重启后将执行boot program1
- 芯片擦写boot program0时掉电，由于boot program1已经写入，将不影响后续运行

另一种BootSwap应用方法如下图，为了保证可靠的更新启动代码，使用2nd 8KB物理空间作为原来Boot程序的备份，如果编程期间发生异常掉电，则触发BootSwap：



如果启动程序更新期间没有发生异常掉电，则可以不执行软复位，无需真正Swap，仅需在更新原来的Boot程序前使能BootSwap，成功更新后撤销BootSwap即可：



推荐应用按照如下步骤升级：

- 更新application program
- 如需升级boot，先将新的boot程序写入第二个8KB空间
- 配置information block，使能BootSwap
- 执行软复位，重启后执行新的boot程序

- 将第二个8KB空间改写为新的应用程序

寄存器标志（FLSIF.BTSF）来表示当前的Boot区是1st 8KB物理地址、还是2nd 8KB物理地址，用于给用户代码查询当前启动情况。

7.4.4 Data Flash

当用户配置 OPTBYTES 使能了 DFLSEN 之后，FM33LC0XX 将开放 16KB 数据 flash 给用户用于数据存储。在使能了 data flash 之后，不同型号产品的 flash 容量划分如下表所示：

型号	Data flash size	Data flash address	Program flash size
FM33LC04x	16KB	0x0003_C000~0x0003_FFFF	240KB
FM33LC02x	16KB	0x0001_C000~0x0001_FFFF	112KB

当使能了 data flash 之后，相应的程序 flash 空间会减少 16KB。Data flash 总是位于 flash 逻辑地址空间的最高 16KB。

Data flash 与 program flash 在访问权限上没有差别，同样受 DBRDP 和 ACLOCK 控制。但是当芯片进行全擦操作时，data flash 不会被擦除。而 data flash 不使能时，芯片执行全擦会擦除 main array 所有数据。

注：在使能 data flash 的情况下，芯片全擦时间显著增加，对于 256KB 容量型号，全擦时间从 8ms 增加到 240ms，对于 128KB 容量型号，全擦时间从 8ms 增加到 112ms。

7.4.5 Flash 的内容保护

Flash 内容保护主要用于保护 Flash 中的用户代码、用户数据和用户配置信息被非授权方读取或篡改。

Flash 保护包含两种类型：Debug 接口读取保护（DBRDP-DeBug ReaD Protection）和应用代码权限保护（ACLOCK-Application Code Block Locking）。Flash 保护的控制通过 LDT1 中的 OPTBYTES 来控制。

7.4.5.1 Debug 接口保护（DBRDP）

DBRDP 的主要目的是防止非授权的第三方通过 debug 接口访问芯片 Flash 内容。

DBRDP 由 LDT1 扇区内的 DBRDPEN 配置字使能或者禁止（0xAA 表示禁止 DBRDP，芯片出厂时默认写为 0xAA）。当 DBRDP 使能时，无法通过 SWD 接口读取或擦写 Flash main array，同时也无法通过 SWD 接口对 RAM 进行访问。

退出 DBRDP 的方法：通过 SWD 对 flash 进行全空间擦除，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 DBRDP，然后复位芯片；复位完成后，芯片将处于无 debug 保护状态。

7.4.5.2 应用代码保护（ACLOCK）

ACLOCK 的主要目的是防止 hacking code 读取或篡改 Flash 中的 application code。通过 ACLOCK 功能，可以设置 CPU 对 Flash 的某些区域只能进行取指操作，不能 read-as-data，也不能擦写。

ACLOCK 以 Block 为单位工作，即对 Flash 保护的颗粒度是 8KB，整个 Flash 包含 32 个 Blocks，对应每个 Block 有 2bit LOCK 信息。出厂时默认的 LOCK 字为 0xFFFF_FFFF，上电 load 的 LOCK 位全部是 11，即默认的无保护状态；当对应 LOCK 位为 01 或 10 时，此 Block 禁止 CPU 擦写和读取，只能取指；当对应的 LOCK 位为 00 时，当前 Block 禁止 CPU 擦写和读取，同时禁止 SWD 擦写和读取。芯片出厂时 LDT0 中关闭 ACLOCK 功能，用户需要通过编程器使能 ACLOCK，并且用户代码编译时要符合 ACLOCK 配置（比如不能将 literal pool 编译到被 LOCK 的 Block）。

ACLOCK 的功能：

- 无保护：所有 Block 允许 CPU 取指、读取、改写，不限制 SWD 访问
- 读写保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不限制 SWD 访问
- 软件和 SWD 保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不允许 SWD 读取、擦写

LOCK 位与 Block 访问权限的关系可以参照下表：

LOCK bit	软件读取	软件取指	SWD 读取和擦写
11	允许	允许	允许
01/10	禁止	允许	允许
00	禁止	允许	禁止

表 7-6LOCK 位权限定义

ACLOCK 信息在芯片复位时 load 到寄存器中，这些寄存器软件也可以置位，但是不能写 0（即只能提升保护等级）。

ACLOCK 不使能时，LOCK 寄存器内容无效。

注：ACLOCK 的权限控制针对 Flash 各个 Block，与 DBRDP 相互独立。对于 SWD 接口而言，DBRDP 的优先级高于 ACLOCK，即 DBRDP 被使能后，不论 ACLOCK 是否起效，SWD 都无法访问 Flash。

注：禁止使用 ACLOCK 关闭 1st block 的读取权限，由于 CPU 复位后首先要从 0 地址读取 MSP 指针，ACLOCK 将导致 CPU 无法正常启动。

退出 ACLOCK 的方法：通过 SWD 对 flash 进行全空间擦除，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 ACLOCK，然后复位芯片；复位完成后，芯片将处于无 ACLOCK 状态。

7.4.5.3 Flash 访问权限说明

Flash 空间访问权限分配：

Flash area	DBRDP	LOCK bits (per Block)[3]	Last byte in page	SWD	Application
Main array	ON	00	x	-	R/E/W/F
		01	x	-	对应 Block 只能取指
		10/11	x	-	对应 Block 只能取指
	OFF	00	x	R/E/W	R/E/W/F
		01	x		对应 Block 只能取指
		10/11	x	对应 Block 无法访问	对应 Block 只能取指
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF3	x	x	x	R/E/W	R/E/W
IF2,1,0	x	x	55	R/E	R
			others	R/E/W	R

表 7-7 Flash 访问权限表

注：

[1] R: Read, E: Erase, W: Write, F: Fetch

[2] 进行 flash 全擦后可以擦写 LDT1

[3] 这里假设 ACLOCKEN 有效。ACLOCKEN 无效的情况下，LOCK bits 不起作用。

7.5 寄存器

offset 地址	名称	符号
FLS(模块起始地址: 0x40001000**)**		
0x00000000	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x00000004	预取指控制寄存器 （Flash Prefetch Control Register）	FLS_PFCR
0x00000008	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x0000000C	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x00000010	ACLOCK 寄存器 2 （Flash Application Code Lock Register2）	FLS_ACLOCK2
0x00000014	Flash 擦写控制寄存器 （Flash Erase/Program Control Register）	FLS_EPCR
0x00000018	Flash Key 输入寄存器 （Flash Key Register）	FLS_KEY
0x0000001C	Flash 中断使能寄存器 （Flash Interrupt Enable Register）	FLS_IER
0x00000020	Flash 标志寄存器 （Flash Interrupt Status Register）	FLS_ISR

7.5.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						WAIT	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1:0	WAIT	Flash 读等待周期配置 (Wait Cycles Config) 00/11：0 wait cycle 01：1 wait cycle 10：2 wait cycles CPU 主频小于等于 24MHz 时，不需要开启 wait；主频大于 24M 小于 48Mhz 时使能 1 wait，主频大于 48Mhz 时使能 2 wait

7.5.2 预取指控制寄存器（FLS_PFCR）

名称	FLS_PFCR							
offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							PRFTEN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	PRFTEN	指令预取指使能，在 WAIT==00 的情况下写 1 无效 (Prefetch Enable) 1：使能 Prefetch 0：禁止 Prefetch

7.5.3 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	-						
位权限	R-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					IF2LOCK	IF1LOCK	-
位权限	U-0					R-0	R-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					DFLSEN	BTSEN	
位权限	U-0					R-0	R-01	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				ACLOCKEN		DBRDPEN	
位权限	U-0				R-01		R-01	

位号	助记符	功能描述
31	IWDTSLP	IWDT 在休眠模式下是否允许应用暂停计数 (IWDT Sleep Enable) 1：允许应用在休眠模式下暂停 IWDT 计数 0：禁止应用在休眠模式下暂停 IWDT 计数
30:19	-	RFU：未实现，读为 0
18	IF2LOCK	Information2 区锁定标志 (IF2 Lock Enable)

位号	助记符	功能描述
		0：未锁定 1：已锁定，锁定后软件不可改写本扇区
17	IF1LOCK	Information1 区锁定标志 (IF1 Lock Enable) 0：未锁定 1：已锁定，锁定后软件不可改写本扇区
16:11	-	RFU：未实现，读为 0
10	DFLSEN	DataFlash 使能 (DataFlash Enable) 0：无 data flash 1：有 data flash
9:8	BTSEN	BootSwap 功能使能 (BootSwap Enable) 00/01/11：禁止 BootSwap 功能 10：允许 BootSwap
7:4	-	RFU：未实现，读为 0
3:2	ACLOCKEN	应用代码权限锁定使能 (AppCode Lock Enable) 00/01/11：ACLOCK 不使能 10：ACLOCK 使能
1:0	DBRDPEN	Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能

7.5.4 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK1[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK1[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK1[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK1[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK1	ACLOCK 配置寄存器低 32bit，分别用于控制 Block15~Block0 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

7.5.5 ACLOCK 寄存器 2（FLS_ACLOCK2）

名称	FLS_ACLOCK2							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK2[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK2[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK2[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK2[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK2	ACLOCK 配置寄存器高 32bit，分别用于控制 Block31~Block16 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock Bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

7.5.6 Flash 擦写控制寄存器（FLS_EPCR）

名称	FLS_EPCR							
offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	ERTYPE	Flash 擦除类型配置 (Erase Type) 00/11：Page Erase

位号	助记符	功能描述
		01：Sector Erase 10：Chip Erase (SWD only)
7:2	-	RFU：未实现，读为 0
1	PREQ	Program Request 软件置位，硬件完成编程后自动清零
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零

7.5.7 Flash Key 输入寄存器（FLS_KEY）

名称	FLS_KEY							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	KEY	Flash 擦写 Key 输入寄存器，软件或者 SWD 在启动擦写前必须正确地向此地址写入合法 KEY 序列。 (Flash Key)

7.5.8 Flash 中断使能寄存器（FLS_IER）

名称	FLS_IER							
offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OTPIE	AUTHIE	KEYIE	CKIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRDIE	ERDIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	OTPIE	OTP 编程错误中断使能，1 有效 (OTP program error Interrupt Enable)
10	AUTHIE	Flash 读写权限错误中断使能，1 有效 (Flash Authentication Error Interrupt Enable)
9	KEYIE	Flash KEY 错误中断使能，1 有效 (Flash Key Error Interrupt Enable)
8	CKIE	擦写定时时钟错误中断使能，1 有效 (Erase/Program Clock Error Interrupt Enable)
7:2	-	RFU：未实现，读为 0
1	PRDIE	编程完成标志中断使能，1 有效 (Program Done Interrupt Enable)
0	ERDIE	擦写完成标志中断使能，1 有效 (Erase Done Interrupt Enable)

7.5.9 Flash 标志寄存器（FLS_ISR）

名称	FLS_ISR							
offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				KEYSTA			BTSF
位权限	U-0				R-000			R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OTPERR	AUTHERR	KEYERR	CKERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRD	ERD
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:17	KEYSTA	Flash 擦写 KEY 输入状态 (Flash Key Status) 000：Flash 写保护状态，未输入 KEY 001：全擦解锁状态 010：扇区擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101/110/111：RFU
16	BTSF	BootSwap 标志寄存器 (BootSwap) 0：启动程序区为 Flash 物理地址 0000H~1FFFH 1：启动程序区为 Flash 物理地址 2000H~3FFFH
15:12	-	RFU：未实现，读为 0
11	OTPERR	OTP page 编程权限错误，硬件置位，软件写 1 清零 (OTP program Error Flag. Write 1 to clear) 1：尝试对已编程的 OTP 字节进行编程 0：无 OTP 编程错误

位号	助记符	功能描述
10	AUTHERR	Flash 读写权限错误，读取 LOCK 块数据或对 LOCK 块擦写时置位，软件写 1 清零。(Flash Authentication Error Flag, write 1 to clear) 1：Flash 访问权限错误 0：Flash 访问没有发生权限错误
9	KEYERR	Flash KEY 错误，硬件置位，软件写 1 清零 (Flash Key Error Flag, write 1 to clear)
8	CKERR	擦写定时时钟错误，NVMIF 擦写 Flash 时如果 RCHF 未使能，则触发 CKERR 中断，软件写 1 清零。(Erase/Program Clock Error Flag, write 1 to clear)
7:2	-	RFU：未实现，读为 0
1	PRD	Program Done，编程完成标志，硬件置位，软件写 1 清零 (Program Done Flag,write 1 to clear)
0	ERD	Erase Done，擦写完成标志，硬件置位，软件写 1 清零(Erase Done Flag,write 1 to clear)

8 复位管理单元（RMU）

8.1 概述

复位电路特点：

- 支持多个复位源，如上下电复位、看门狗复位、软件复位、引脚复位等
- 上下电复位（BOR）监控主电源供电
- BOR 上电复位典型释放电压 1.8V
- BOR 下电复位产生电压软件可配置为 1.75/1.7/1.65/1.6V，可关闭。
- 低功耗下电复位电路（PDR），下电复位电压可配置为 1.25/1.35/1.4/1.5V，可关闭
- 上下电复位信号经过去抖动和延时，抗干扰能力强

进入复位状态时，所有寄存器都恢复到初始值（除 RTC 内部寄存器）；退出复位状态时，MCU 使用内部 RC 振荡器（RCHF，默认频率 8MHz）作为系统时钟。

8.2 模块框图

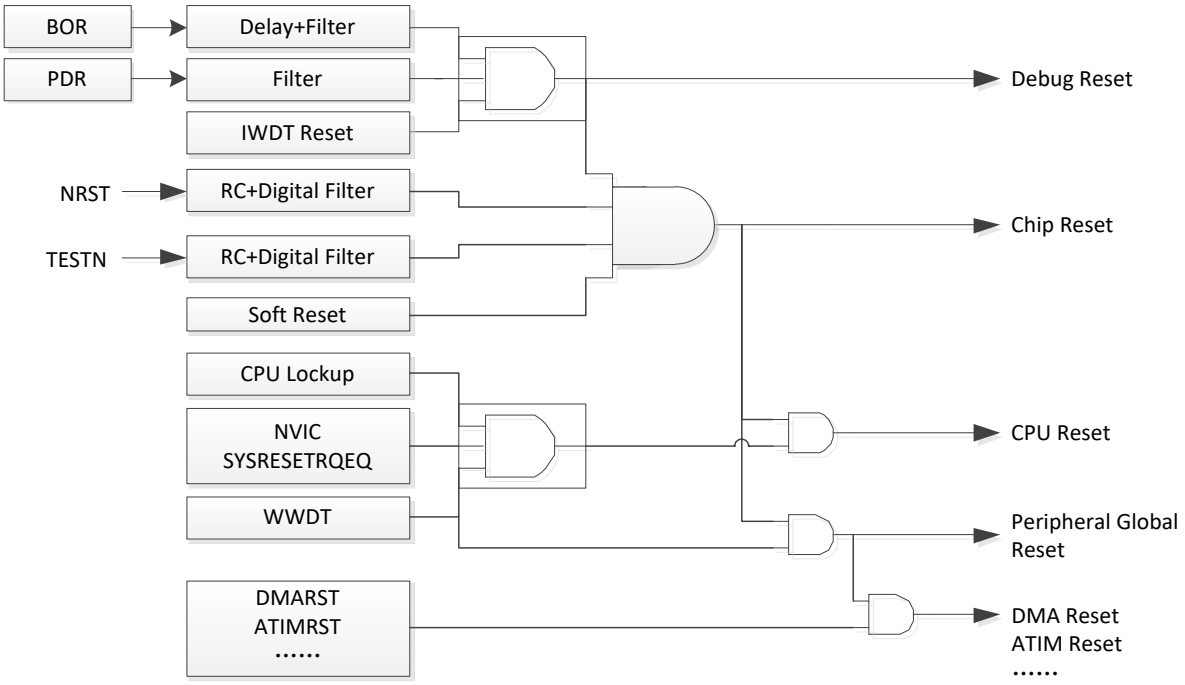


图 8-1 芯片复位源框图

8.3 上下电复位

上下电复位电路监控 VDD 电源，由 BOR 和低功耗 PDR 组成。BOR 复位阈值电压精确度高，但是工作电流较大，因此在不需要精确下电复位阈值的场合，推荐关闭 BOR，仅保留 PDR。

VDD 电源上电期间上电复位信号有效，当 VDD 电压超过 V_{POR} 时，上电复位放开；VDD 跌落到 V_{PDR} 时下电复位有效。为防止电源抖动，保证上电复位电路的抗干扰能力，对上电复位信号进行滤波和延时处理。

V_{POR} 阈值固定为 1.8V，BOR 下电复位阈值和低功耗 PDR 下电复位阈值软件可设置。

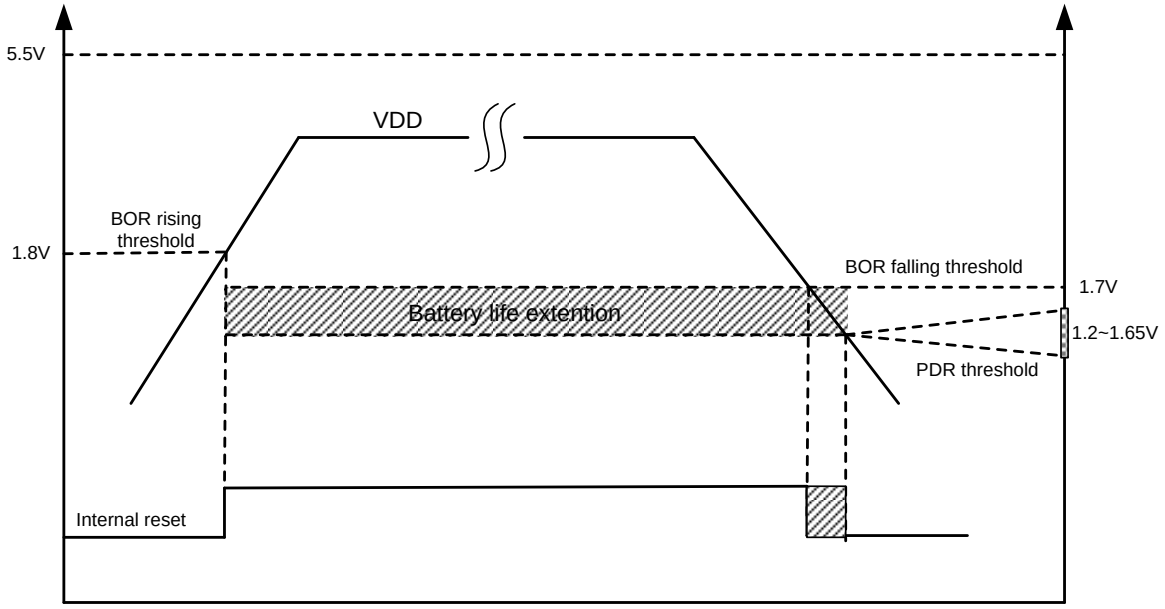


图 8-2 上下电复位示意图

注意：当芯片休眠时，如果要保持高精度BOR下电复位工作，必须保持VREF1p2使能。

8.4 独立看门狗（IWDT）

独立看门狗用于监视系统运行，如果 CPU 运行异常，无法定时清狗，则看门狗在溢出后产生全局复位信号，重启系统，以避免系统锁死。详情请参见 IWDT 章节。

8.5 窗口看门狗（WWDT）

带窗口的看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位全芯片，避免不可预计的后果。详情请参见 WWDT 章节。

8.6 软件复位

软复位由 CPU 写寄存器发起，操作方式为向 SOFTRST 寄存器写 0x5C5C_AABB。

8.7 NRST 引脚复位

NRST 是芯片专用复位引脚，NRST 保持低电平超过 8ms 后，芯片将进入系统复位，但是并不会复位 DEBUG 逻辑。如果芯片处于低功耗模式，NRST 有效也会使芯片复位退出低功耗模式。

8.8 寄存器

offset 地址	名称	符号
RMU(模块起始地址: 0x4001A800**)**		
0x00000000	PDR 控制寄存器 （PDR Control Register）	RMU_PDRCR
0x00000004	BOR 控制寄存器 （BOR Control Register）	RMU_BORCR

8.8.1 PDR 控制寄存器（RMU_PDRCR）

名称	RMU_PDRCR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		EN
位权限	U-0					R/W-11		R/W-1

位号	助记符	功能描述
31:3	**-**	RFU：未实现，读为 0
2:1	**CFG**	下电复位电压配置 (Power Down Reset Config) 00：1.5V 01：1.25V（禁止使用） 10：1.35V 11：1.4V（默认）
0	**EN**	下电复位使能 (Power Down Reset Enable) 0：关闭下电复位 1：使能下电复位

8.8.2 BOR 控制寄存器（RMU_BORCR）

名称	RMU_BORCR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CFG		OFF_BOR_1P0	OFF_BOR_1P2
位权限	U-0				R/W-01		R/W-1	R/W-0

位号	助记符	功能描述
31:4	**-**	RFU：未实现，读为 0
3:2	**CFG**	下电复位电压配置 (Brown-Out-Reset Config) 00：1.7V 01：1.6V（默认） 10：1.65V 11：1.75V
1	**OFF_BOR_1P0**	BOR 上电复位使能，上电后默认关闭 (OFF BOR 1V Enable) 0：使能 BOR 上电复位 1：关闭 BOR 上电复位
0	**OFF_BOR_1P2**	BOR 下电复位使能，上电后默认使能(OFF BOR 1.2V Enable) 0：使能 BOR 下电复位 1：关闭 BOR 下电复位 注意，使能 OFF_BOR_1P2 时必须确保 VREF_EN 置位

9 独立看门狗（IWDT）

9.1 概述

独立看门狗用于监视系统运行，如果 CPU 运行异常，无法定时清狗，则看门狗在溢出后产生全局复位信号，重启系统，以避免系统锁死。独立看门狗在芯片上电后由软件启动，启动后无法关闭，直到芯片发生复位。

为了便于调试，在以下情况下 IWDT 会停止运行：

- 当芯片处于调试模式时，软件可以通过配置 MCUDBGCR 寄存器在调试过程中暂停 IWDT
- 当 OPTBYTES 中 IWDTSLP 有效时，软件可以在休眠模式下暂停 IWDT 计数

IWDT 核心是一个 12bit 向上计数器，复位后从 0 开始递增，计数到 0xFFF 后触发 IWDT 复位。IWDT 复位是一个全局复位，效果等同于上下电复位。

IWDT 使用 LSCLK 工作，结合停振检测电路，确保在 XTLF 低频晶振停振时也不会停止运行。IWDT 带有除 128 预分频器，计数器长度为 12bit。

IWDT 支持可编程窗口功能，软件只能在允许的窗口内清狗，窗口外清狗将触发 IWDT 复位。

9.2 结构框图

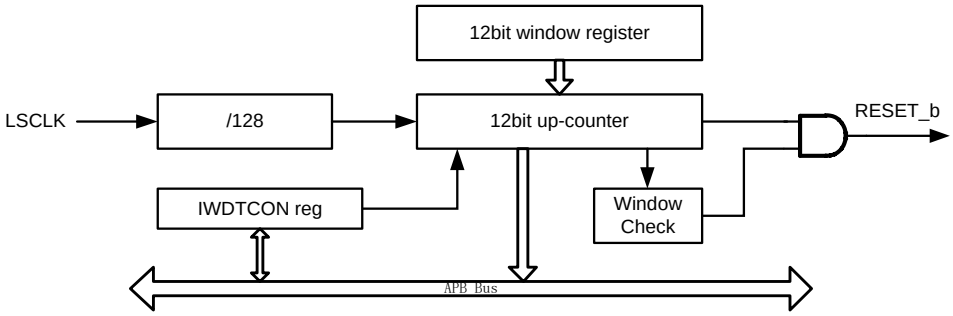


图 9-1 IWDT 结构框图

9.3 IWDT 功能描述

CPU 正常运行时，看门狗应使用较短的溢出周期，而在 SLEEP/DEEPSLEEP 等低功耗模式下，为了使芯片尽可能长时间的停留在低功耗模式下，则看门狗应使用较长的溢出周期。

为了兼容两者的不同应用需求，软件可以实时修改 IWDT 的溢出周期配置。为避免不当操作引发不可预计的后果，软件在更新溢出周期配置时应遵循以下操作步骤：

- 确保看门狗正在运行
- 首先进行一次清狗操作
- 随后改写 IWDTCFG 寄存器，选择合适的溢出周期
- 读 IWDTCFG，确保写入正确
- 溢出周期更新完毕，CPU 正常运行

IWDT 使用 LSCLK 工作，内部预分频 128，分频后的计数器溢出长度可配置为 1~4096（共 8 个可用档位），溢出时间长度计算公式如下：

$tWWDT = T_{LSCLK} * 128 * IWDTOVP$

LSCLK 频率	溢出长度配置	溢出时间（ms）
32768Hz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

表 9-1 IWDT 溢出周期表

9.4 IWDT 窗口功能

IWDT支持可编程清狗窗口功能。IWDT_WIN寄存器用于定义允许的清狗窗口，只有当计数器计数值大于等于IWDT_WIN的值时，清狗操作才是合法的，在窗口之外清狗将会直接出发IWDT复位。

芯片复位后IWDT_WIN为全0，即默认允许软件在任何位置清狗。

软件可以在IWDT运行过程中实时修改IWDT_WIN寄存器。软件清狗时必须读取并确认当前计数值是否在允许清狗的范围内。

当IWDT计数值进入清狗窗口时，IWDT会触发一个中断标志寄存器，通知软件当前计数值已经进入清狗窗口。

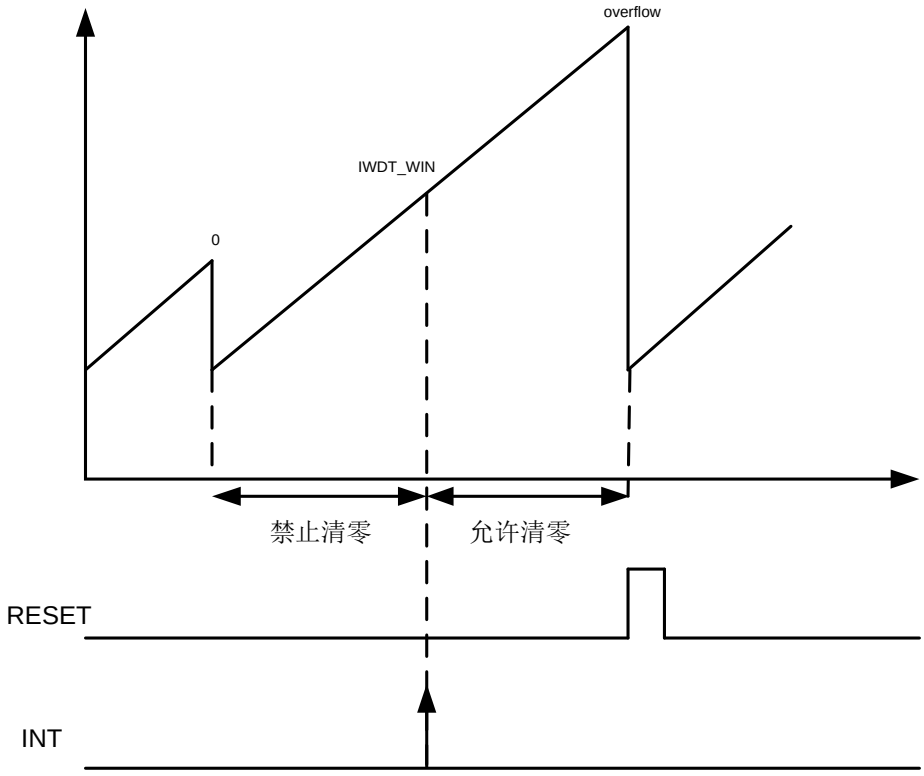


图 9-2 IWDT 窗口示意图

9.5 IWDT 冻结

用户可以通过OPTBYTES配置是否允许IWDT在休眠模式下冻结计数。

当OPTBYTES中的IWDTSLP有效，并且软件将IWDT_FREEZE寄存器置位时，当芯片进入Sleep/DeepSleep模式，IWDT计数值自动冻结（注意不是关闭IWDT，只是计数值保持当前值不再递增）。

9.6 寄存器

offset 地址	名称	符号
IWDT(模块起始地址: 0x40011400)		
0x00000000	IWDT 清除寄存器 （IWDT Service Register）	IWDT_SERV
0x00000004	IWDT 配置寄存器 （IWDT Config Register）	IWDT_CR
0x00000008	IWDT 计数值寄存器 （IWDT Counter Register）	IWDT_CNT
0x0000000C	IWDT 窗口寄存器 （IWDT Window Register）	IWDT_WIN
0x00000010	IWDT 中断使能寄存器 （IWDT Interrupt Enable Register）	IWDT_IER
0x00000014	IWDT 中断标志寄存器 （IWDT Interrupt Status Register）	IWDT_ISR

9.6.1 IWDT 清除寄存器（IWDT_SERV）

名称	IWDT_SERV							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SERV[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SERV[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SERV[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SERV[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	**SERV**	上电复位后 IWDT 默认关闭，软件向此寄存器写入 0x1234_5A5A 后启动 IWDT，此后 IWDT 不可关闭直到下一次芯片复位。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

9.6.2 IWDT 配置寄存器（IWDT_CR）

名称	IWDT_CR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				FREEZE	-		
位权限	U-0				R/W-0	U-0		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					OVP		
位权限	U-0					R/W-001		

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	FREEZE	IWDT 休眠冻结，仅在 OPTBYTES 中 IWDTSLP 配置有效时起作用(Freeze in Sleep Enable) 1：Sleep/DeepSleep 模式下冻结 IWDT 计数 0：Sleep/DeepSleep 模式下保持 IWDT 运行
10:3	-	RFU：未实现，读为 0
2:0	OVP	配置 IWDT 看门狗溢出时间 (Overflow Period) 000：125ms 001：250ms 010：500ms 011：1s 100：2s 101：4s 110：8s 111：16s

9.6.3 IWDT 计数值寄存器（IWDT_CNT）

名称	IWDT_CNT							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	CNT	IWDT 当前计数值，软件只读 (IWDT Counter Value,read only) 由于计数器工作时钟与 APB 总线为异步关系，软件读取计数值时应连续读取 2 次以上，为相同值时才认为是稳定结果

9.6.1 IWDT 窗口寄存器（IWDT_WIN）

名称	IWDT_WIN							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WIN[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WIN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	WIN	IWDT 窗口寄存器 (IWDT Window)

9.6.2 IWDT 中断使能寄存器（IWDT_IER）

名称	**IWDT_IER**							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	IWDT 中断使能 (IWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

9.6.3 IWDT 中断标志寄存器（IWDT_ISR）

名称	**IWDT_ISR**
Offset	0x00000014

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							WINF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WINF	IWDT 进入窗口中断标志，写 1 清零 (Window Flag,write 1 to clear) 0：无中断产生 1：计数值进入清狗窗口

10 窗口看门狗（WWDT）

10.1 功能描述

带窗口的看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位全芯片，避免不可预计的后果。

WWDT 在芯片上电后默认关闭，软件启动 WWDT 后，不能再关闭，直到下一次复位。低功耗休眠模式下 WWDT 停止运行。

为了保证同步性和实时性，WWDT 使用 CPU 时钟工作，内部有一个预分频电路，以产生同步计数使能信号。

在以下情况时 WWDT 产生 CPU 复位：

- 计数器溢出
- 对 WWDT 清零寄存器写 0xAC 以外的值（可用于触发 CPU 软复位）
- 在窗口关闭期内对 WWDT 清零寄存器写 0xAC

当计数器达到溢出时间的 75%时，会触发一个预警中断。

10.2 结构框图

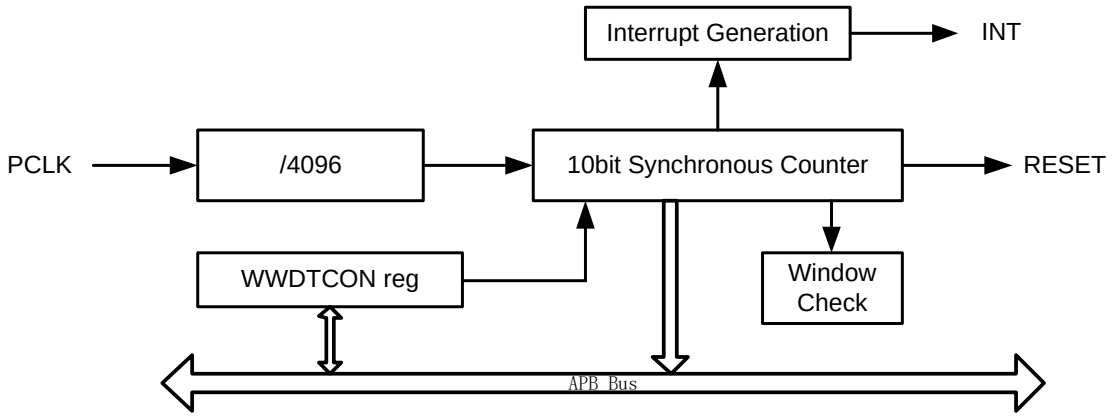


图 10-1WWDT 结构框图

10.3 WWDT 工作方式

WWDT 在芯片复位后默认关闭，软件需对 WWDTCON 寄存器写入 0x5A 来启动 WWDT。WWDT 启动后，如果软件在 open window 内对 WWDTCON 写 0xAC，将清零计数器。WWDT 一旦使能后不能关闭，直到下一次复位，WWDT 复位发生后将会关闭 WWDT。

WWDT 使用 PCLK 工作，内部预分频 4096，分频后的计数器溢出长度可配置为 1~1024（共 8 个可用档位），溢出时间长度计算公式如下：

$tWWDT = T_{APBCLK} * 4096 * N_{CFG}$

下表为计算示例：

APBCLK 频率	溢出长度配置	溢出时间（ms）
48MHz	1	0.085
	4	0.341
	16	1.365
	64	5.461
	128	10.922
	256	21.845
	512	43.69
	1024	87.38
32MHz	1	0.128
	4	0.512
	16	2.048
	64	8.192
	128	16.384
	256	32.768
	512	64.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	129.072
	1024	262.144
8MHz	1	0.512
	4	2.048
	16	8.192
	64	32.768
	128	65.536
	256	131.072
	512	258.144

APBCLK 频率	溢出长度配置	溢出时间（ms）
	1024	524.288

表 10-1 WWDT 溢出周期表

WWDT 只允许在 open window 内进行清除，否则将直接触发复位。使能窗口为计数器的后半周期，软件在清零看门狗之前应注意查询计数值。

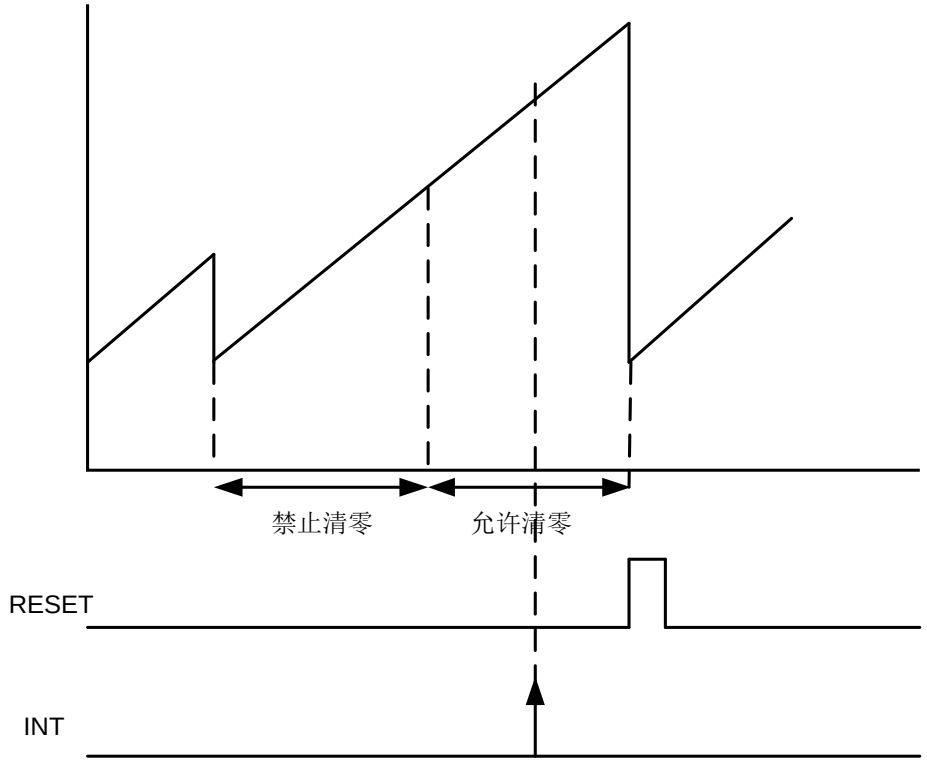


图 10-2WWDT 窗口示意图

10.4 寄存器

offset 地址	名称	符号
WWDT(模块起始地址：0x40011800)		
0x00000000	WWDT 控制寄存器 （WWDT Control Register）	WWDT_CR
0x00000004	WWDT 配置寄存器 （WWDT Config Register）	WWDT_CFGR
0x00000008	WWDT 计数值寄存器 （WWDT Counter Register）	WWDT_CNT
0x0000000C	WWDT 中断使能寄存器 （WWDT Interrupt Enable Register）	WWDT_IER
0x00000010	WWDT 中断标志寄存器 （WWDT Interrupt Status Register）	WWDT_ISR
0x00000014	WWDT 预分频寄存器 （WWDT Prescaler Register）	WWDT_PSC

10.4.1 WWDT 控制寄存器（WWDT_CR）

名称	WWDT_CR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CON							
位权限	W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	CON	当 CPU 向此地址写入 0x5A 时启动 WWDT 定时器 (WWDT Control,write only) 在启动 WWDT 后，当 CPU 向此地址写入 0xAC 时清零计数器

10.4.2 WWDT 配置寄存器（WWDT_CFGR）

名称	WWDT_CFGR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		
位权限	U-0					R/W-011		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	CFG	配置 WWDT 看门狗溢出时间，复位值 011，由于上电后系统时钟默认为 8Mhz，所以默认溢出周期大约 32ms (WWDT Config) 000：T_{PCLK} * 4096 * 1 001：T_{PCLK} * 4096 * 4 010：T_{PCLK} * 4096 * 16 011：T_{PCLK} * 4096 * 64 100：T_{PCLK} * 4096 * 128 101：T_{PCLK} * 4096 * 256 110：T_{PCLK} * 4096 * 512 111：T_{PCLK} * 4096 * 1024

10.4.3 WWDT 计数寄存器（WWDT_CNT）

名称	WWDT_CNT							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CNT[9:8]	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:0	CNT	WWDT 计数寄存器值，软件可通过查询此寄存器了解 WWDT 计时进度 (WWDT Counter value,read only)

10.4.4 WWDT 中断使能寄存器（WWDT_IER）

名称	WWDT_IER
Offset	0x0000000C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	WWDT 中断使能 (WWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

10.4.5 WWDT 中断标志寄存器（WWDT_ISR）

名称	WWDT_ISR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							NOVF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	NOVF	WWDT 75%计时中断标志，写 1 清零 (Near Overflow Flag,write 1 to clear) 0：无中断产生 1：中断标志置位 如果 IE=1，则此寄存器置位将触发中断

10.4.6 WWDT 预分频寄存器（WWDT_PSC）

名称	WWDT_PSC
Offset	0x00000014

名称	WWDT_PSC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				DIV_CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIV_CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	DIV_CNT	WWDT 的 4096 预分频计数器当前计数值，只读 (WWDT prescaler Divider Counte,read only)

11 时钟管理单元（CMU）

11.1 概述

芯片内包含32.768KHz低频晶体振荡电路(XTLF)、4~32MHz高频晶体振荡器、最高24MHz高频RC振荡器(RCHF)、32KHz低功耗内部环振（LPOSC）、4MHz低功耗环振、高频晶体振荡器和一个锁相环（PLL）。芯片内部的时钟产生模块整合这些时钟源，产生各个模块工作所需要的时钟。

特点：

- 系统主时钟可选多个时钟源
- 时钟可在系统运行中实时切换
- 低频晶体振荡器配备停振检测电路
- 部分外设模块独立工作时钟（与 CPU 和总线时钟解耦）
- CPU 和总线最高频率 64MHz

11.2 时钟架构

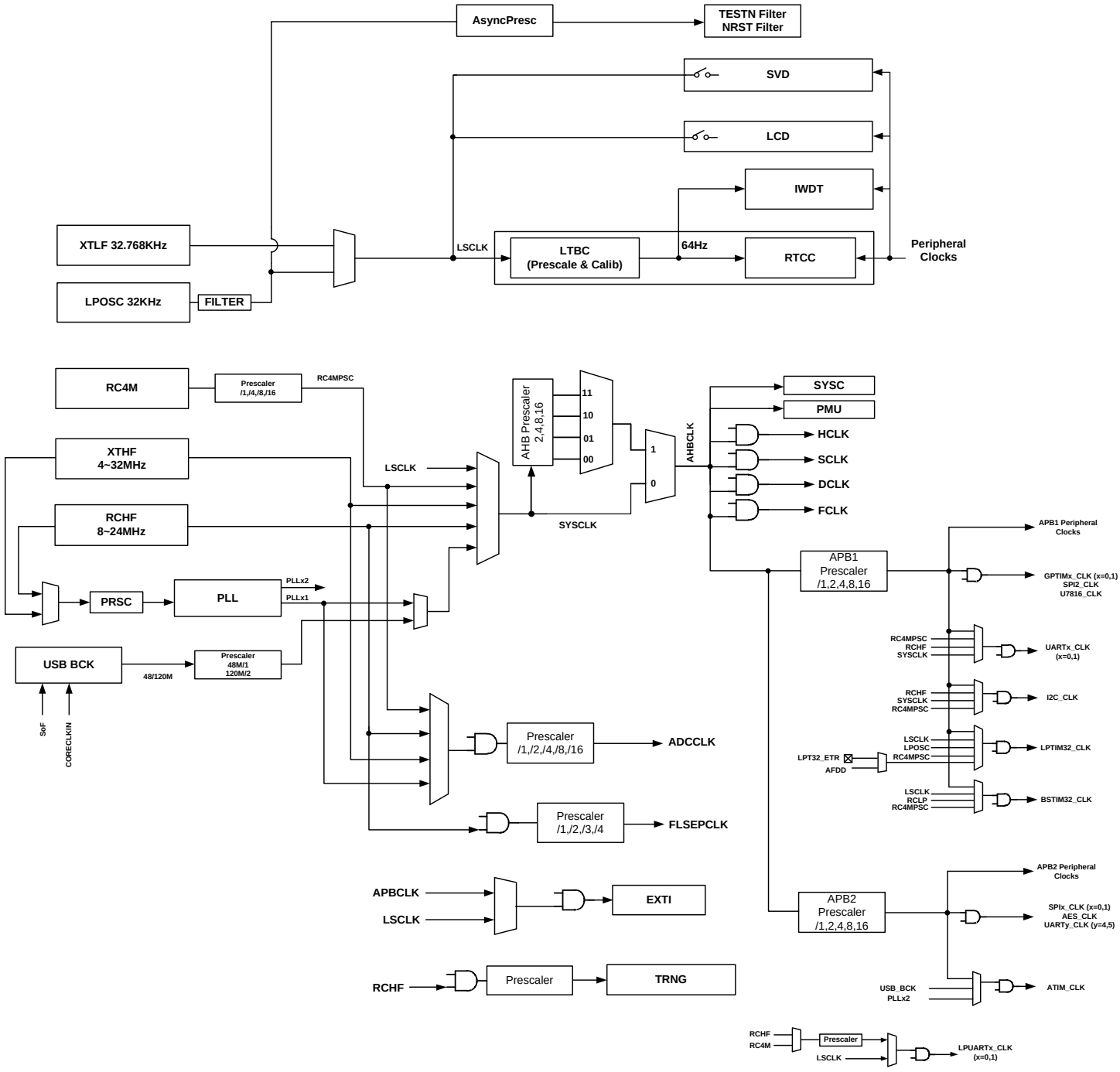


图 11-1 芯片时钟框图

系统主时钟(SYSCLK)可由XTLF、RCHF、LPOSC、PLL、LPOSC、RCMF及它们的分频时钟产生。上电默认使用8MHz RCHF的不分频时钟作为系统主时钟，各外设模块的时钟可以分别独立控制。芯片工作时可以只打开需要工作的模块时钟，其他模块的时钟可关闭，以节省功耗。APB总线时钟APBCLK可以是AHBCLK的分频或同频时钟，用于驱动APB总线上的低速外设。

11.2.1 SYSCLK 切换说明

SYSCLK是系统主时钟，从SYSCLK可以得到AHBCLK和APBCLK等总线时钟以及CPU运行所需的时钟。

SYSCLK选择任何一个时钟源时，硬件都要检查对应时钟源是否开启，如果时钟源没有被使能，则软件切换操作无效，SYSCLKSEL寄存器不会被改写，时钟切换也不会发生。

目标时钟	切换条件
RCHF	RCHF 使能
RCMF	RCMF 使能
XTHF	XTHF 使能且未停振
PLL	PLL 使能，并且： 1，如果 PLL 参考时钟是 XTHF，则 XTHF 必须使能且未停振 2，如果 PLL 参考时钟是 RCHF，则 RCHF 必须使能
XTLF	XTLF 使能且未停振
USB BCK	USB BCK 模块使能（仅 USB 系列型号支持）

表 11-1 系统时钟切换条件

11.2.2 主要时钟说明

时钟	源头	说明
LSCLK	XTLF, LPOSC	32KHz 低频系统时钟，晶体停振时可以切换到 LPOSC 主要用于 RTC、IWDT、引脚滤波、SVD、LCD
SYSCLK	RCHF, PLL, RCLF, LSCLK, XTHF, RCMF, USB_BCK	32K~64MHz，经过分频后得到 AHBCLK
HCLK(AHBCLK)	SYSCLK	AHB 总线时钟，用于驱动 CPU、RAM、Flash 和高速外设
SCLK	SYSCLK	CPU 内核系统时钟
DCLK	SYSCLK	CPU 内核 Debug 时钟（当仿真器连接时这个时钟必须活动）
FCLK	SYSCLK	Free-Running 时钟，提供给 CPU 内核 WIC 模块，以及 APB 桥
APBCLK	AHBCLK	APB 总线时钟，用于驱动低速外设

表 11-2 主要系统时钟说明

11.2.3外设模块的总线时钟和工作时钟

部分外设模块的总线时钟和工作时钟互相独立。

其中总线时钟用于AHB或APB总线访问，在软件访问外设的功能寄存器时，必须先通过外设总线时钟控制寄存器来使能对应的总线时钟。

而外设的工作时钟为外设实际工作使用的时钟，这个时钟可能不同于APBCLK或AHBCLK，外设模

块工作前，需要通过外设工作时钟寄存器来选择所需的时钟源，并打开时钟门控。

而对于工作时钟和总线时钟统一的外设模块，则仅需使能总线时钟就可以正常工作了。

模块	总线时钟	工作时钟
独立工作时钟外设		
UARTx (x=0,1)	APB1CLK	APBCLK
		RCHF
		SYSCLK
		RCMF
LPUART0	APB1CLK	LSCLK
		RCHF
		RCMF
LPUART1	APB2CLK	LSCLK
		RCHF
		RCMF
I2C	APB1CLK	APBCLK
		RCHF
		SYSCLK
		RCMF
ATIM	APB2CLK	APBCLK
		USB_BCK
		PLLx2_CLK
LPTIM32	APB1CLK	APBCLK
		LSCLK
		LPOSC
		LPTIN
BSTIM32	APB2CLK	APBCLK
		LSCLK
		LPOSC
		RCMF_PSC
ADC	APB2CLK	RCMF
		XTHF
		RCHF
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APB2CLK	RCHF
IWDT	APB1CLK	LSCLK
LCD	APB1CLK	LSCLK
RTC	APB1CLK	LSCLK
非独立工作时钟外设		
PMU	AHBCLK	
DMA	AHBCLK	
GPTIMx	APB1CLK	
UARTy (y=4,5)	APB2CLK	

模块	总线时钟	工作时钟
SPI2	APB1CLK	
SPI1	APB2CLK	
7816	APB1CLK	
AES	APB2CLK	
CRC	APB2CLK	
WWDT	APB1CLK	
OPAx	APB2CLK	
COMPx	APB2CLK	

表 11-3 总线时钟和外设工作时钟

11.2.4 休眠模式下的外设时钟

Sleep/DeepSleep模式下，SYSCLK被关闭，因此在休眠模式下AHBCLK和APBCLK都不工作，所有基于AHBCLK或APBCLK的外设都停止工作。但是，使用独立与总线时钟工作的外设仍可以继续工作，比如UART0/1、LPUARTx、I2C、ATIM、LPTIM32、BSTIM32。

为了让上述外设在休眠模式下继续工作，软件需要在休眠前确保上述外设使用SYSCLK和总线时钟以外的时钟工作。

11.2.5 LSCLK 切换逻辑

LSCLK是供RTC、IWDT、SVD和LCD驱动使用的低速时钟，典型频率32K左右。LSCLK的源头是XTLF或者LPOSC，芯片支持两者之间的自动切换或者由软件手动切换。

LSCLK的自动切换功能，由LSACTS寄存器配置，仅在XTLF使能的情况下有效，此时假设XTLF是主时钟，LPOSC是备份时钟，仅用于防止XTLF异常停振。所以LSCATS只在XTLF使能的情况下起作用，当XTLF出现意外停振，FDET输出的停振信号将LSCLK自动切换到LPOSC。

在LSCLK自动切换不使能的情况下，软件可以通过LSCLKSEL寄存器实现对LSCLK的手动切换。

芯片上电复位后，XTLFEN=0，LFDET输出不停振，LSCLKSEL默认选中LPOSC，XTLF无输出；因此上电后LSCLK默认是LPOSC。

此后，软件如果希望使用XTLF，则应当使能XTLF并轮询LFDET输出，直到确认XTLF起振后，将LSCATS置位，或者清零LSCLKSEL。

当LSCLK用作系统时钟（SYSCLK）时，建议软件打开停振自动切换功能。

11.3 USB PHY BCK

USB PHY的内建时钟源可以用总线SoF或者芯片内部参考时钟来做时钟自校准，获得准确的

48/96/120M输出。

当PHY使用芯片内部时钟作为参考时钟时，需要CMU模块输出参考时钟，其频率可以是12M或者32768Hz。其时钟源可以来自于晶振或内部环振，如下图所示：

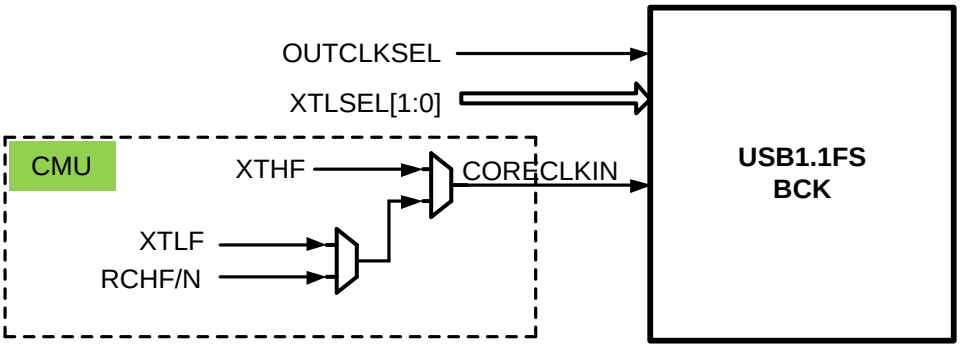


图 11-2USB PHY 参考时钟源选择

11.4 高频 RC 振荡器(RCHF)

11.4.1 概述

高频RC振荡器典型振荡频率为8/16/24MHz，可用作系统主时钟，MCU在此频率下工作，可达到性能与功耗的平衡。为满足不同应用对MCU执行速度的需求，高频RC振荡器的输出频率可调，最高能达到24MHz。RCHF输出频率可以进行调校，出厂前调校到目标频率的+/-1%以内，8MHz输出全温区（-40~+85°C）频率变化小于+/-1%，16MHz输出全温区（-40~+85°C）频率变化小于+/-2%。

11.4.2 软件使用说明

芯片上电后默认使用RCHF 8MHz时钟工作，硬件电路会自动从Flash读取8MHz校准值，保证8MHz频率误差小于+/-0.5%。

如果软件需要使用其他频率，则按照以下步骤操作：

- 改写RCHF_CR.FSEL
- 从Flash（0x1FFF_FB40、0x1FFF_FB3C、0x1FFF_FB38）读取频率调校值（分别对应8/16/24MHz）
- 将频率调校值写入RCHF_TR寄存器，即可得到目标频率误差小于+/-1%的时钟

Flash中的RCHF校准参数数据格式如下：

AHB地址	bit[31:16]	bit[15:0]	说明
0x1FFF_FB38	{9'b0000_0000_0, ~RCHF24M_TRIM}	{9'b1111_1111_1, RCHF24M_TRIM}	RCHF 24Mhz调校值（软件装载）
0x1FFF_FB3C	{9'b0000_0000_0, ~RCHF16M_TRIM}	{9'b1111_1111_1, RCHF16M_TRIM}	RCHF 16Mhz调校值（软件装载）
0x1FFF_FB40	{9'b0000_0000_0, ~RCHF8M_TRIM}	{9'b1111_1111_1, RCHF8M_TRIM}	RCHF 8Mhz调校值（上电后自动装载）

表 11-4 RCHF 校准数据

RCHF校准值为7bit数据，上述flash地址中的高16bit和低16bit分别保存校准值和反码校验字。如果正反码校验失败，校准值应被丢弃。

11.5 中频 RC 振荡器(RCMF)

11.5.1 概述

RCMF是一个低功耗中频环振，典型频率4MHz，典型功耗仅20uA左右，用于CPU低功耗低速运行。RCMF出厂时经过测试校准，软件使用前可以从0x1FFF_FB44地址读取校准值，并写入RCMFTR寄存器，即可获得常温下误差小于+/-1%的4M时钟。

Flash中的RCMF校准参数数据格式如下：

AHB地址	bit[31:16]	bit[15:0]	说明
0x1FFF_FB44	{9'b0000_0000_0, ~RCMF_TRIM}	{9'b1111_1111_1, RCMF_TRIM}	RCMF调校值

表 11-5 RCMF 校准数据

RCMF校准值为7bit数据，上述flash地址中的高16bit和低16bit分别保存校准值和反码校验字。如果正反码校验失败，校准值应被丢弃。

11.6 高精度低功耗 RC 振荡器(LPOSC)

11.6.1 概述

LPOSC功耗极低，仅几百nA，可以用于XTLF的备份时钟，或者单独使用。

在ACTIVE和LPRUN模式下默认开启，不能关闭。

如果使能了XTLF停振自动切换备份时钟的功能，当XTLF停振时LPOSC被强制开启。

当软件关闭XTLF时，LFDET不会输出停振信号，如果此时软件也关闭了LPOSC，则IWDT和RTC都停止运行；如果软件希望在休眠模式下保持LPOSC运行，则需要保证在休眠前使能LPOSC。

功耗模式	状态	控制说明
Active/LPRun	保持开启	不可关闭
Sleep/DeepSleep	可以保持开启或者关闭	根据 RCC.LPM_LPOSC_OFF 寄存器的设置，保持 LPOSC 使能或者关闭 如果 LPOSC 关闭，则 XTLF 停振时根据寄存器配置决定是否自动启动 LPOSC 并输出 32KHz
Sleep/DeepSleep 唤醒	自动开启	唤醒时总是自动启动 LPOSC

表 11-6 LPOSC 控制状态

Flash中的LPOSC校准参数数据格式如下：

AHB地址	bit[31:16]	bit[15:0]	说明
0x1FFF_FB20	{8’b0000_0000, ~LPOSC_TRIM}	{8’b1111_1111, LPOSC_TRIM}	LPOSC调校值

表 11-7 LPOSC 校准数据

LPOSC校准值为8bit数据，上述flash地址中的高16bit和低16bit分别保存校准值和反码校验字。如果正反码校验失败，校准值应被丢弃。

11.7 低频晶体振荡电路(XTLF)

11.7.1 概述

低频晶体振荡电路通过外接32768Hz晶体提供稳定的振荡源，功耗极低，主要用来给实时时钟(RTC)模块提供输入时钟。XTLF的振荡强度可调，用户可根据需要选择振荡强度，达到振荡能力与功耗的平衡。XTLF的反馈电阻集成在芯片内部，用户需要在振荡引脚上外加负载电容。

芯片内部集成了一个停振检测电路，用来检测XTLF是否停振。一旦检测到XTLF停振，将产生XTLF停振中断，通知CPU及时处理。

软件可以使能或关闭XTLF。为了提高抗干扰能力，采用4bit的XTLFEN控制位，4bit复位值为0101，必须改写为1010才能关闭XTLF，其他任何数据都会保持XTLF使能。

11.7.2 工作方式

XTLF上电后默认关闭，软件启动，默认使用中等强度，以缩短起振时间，相应的振荡功耗也较大。典型的起振时间小于1s。当振荡器充分起振后，软件可以通过配置寄存器降低振荡功耗。

11.7.3 停振检测

FM33LC0XX带有片上低频晶体停振检测电路，详情请参见FDET章节。

11.8 高频晶体振荡电路(XTHF)

11.8.1 概述

通过外接高频晶体，XTHF能够为MCU提供高精度的高频时钟源。静态和负载电容应尽可能靠近XTHF引脚布置，其中负载电容大小应合理选择，以适配所选用的晶体类型。

XTHF可以适配4~32MHz晶体。软件可以通过XTHFEN寄存器使能或关闭XTHF时钟。

11.8.2 工作方式

XTHF上电后默认关闭。上电复位完成后，软件可以根据需要打开XTHF。由于晶振引脚与GPIO复用，软件使能XTHF前，需要将PC2和PC3引脚配置为模拟功能。

11.8.3 停振检测

FM33LC0XX带有片上高频晶体停振检测电路，详情请参见FDET章节。

11.9 锁相环(PLL)

11.9.1 概述

锁相环输入参考时钟可以是RCHF或XTHF分频，最高输出频率可达64MHz以及其2倍频。软件使用PLL作为系统时钟前，需配置输入参考时钟和倍频系数。

11.9.2 软件应用指南

出于可靠性考虑，软件需注意以下几点：

- 软件选择PLL输入时必须保证RCHF或XTHF为使能状态
- PLL输出选为SYSCLK时不能关闭PLL
- 软件应等待PLL锁定后再将SYSCLK配置为PLL输出

配置PLL输出64MHz，并使系统以64MHz主频运行：

- 配置PLLCON寄存器，选择输入时钟源和输出时钟频率
- 设置Flash wait cycle为2
- 使能Flash预取指（可选）
- 将AHB时钟选择为PLL输出

11.10 低功耗模式下的时钟源

在低功耗模式下，部分时钟源被硬件强制关闭，而另外一部分时钟源则仍可以保持工作。具体参见下表：

时钟源	LPRUN/Sleep/DeepSleep	说明
RCHF	X	硬件强制关闭
PLL	X	
XTHF	X	
USB_BCK	X	
RCMF	O	软件配置使能或关闭
LPOSC	O	
XTLF	O	

表 11-8 低功耗下的时钟源

11.11 休眠唤醒的时钟处理

当芯片从Sleep/DeepSleep模式下唤醒时，硬件自动打开RCHF并恢复到休眠前的频率输出；同时将SYSCLKSEL寄存器复位成00，将系统时钟选为RCHF，而AHBPRES寄存器不会被复位，保持休眠前的状态；因此芯片唤醒后默认将使用RCHF或者其分频时钟工作。

11.12 寄存器

offset 地址	名称	符号
RCC(模块起始地址:0x40000200)		
0x00000000	LOCKUP 复位控制寄存器 （Lockup reset Control Register）	RCC_LKPCR
0x00000004	软件复位寄存器 （Software Reset Register）	RCC_SOFTRST
0x00000008	复位标志寄存器 （Reset Flag Register）	RCC_RSTFR
0x0000000C	系统时钟控制寄存器 （System Clock Control Register）	RCC_SYSCLKCR
0x00000010	RCHF 控制寄存器 （RCHF Control Register）	RCC_RCHFCR
0x00000014	RCHF 调校寄存器 （RCHF Trim Register）	RCC_RCHFTR
0x00000018	PLL 控制寄存器 （PLL Control Register）	RCC_PLLCR
0x0000001C	LPOSC 控制寄存器 （LPOSC Control Register）	RCC_LPOSCCR
0x00000020	LPOSC 调校寄存器 （LPOSC Trim Register）	RCC_LPOSCTR
0x00000024	XTLF 控制寄存器 （XTLF Control Register）	RCC_XTLFCR
0x00000028	外设总线时钟控制寄存器 1 （Peripheral bus Clock Control Register1）	RCC_PCLKCR1
0x0000002C	外设总线时钟控制寄存器 2 （Peripheral bus Clock Control Register2）	RCC_PCLKCR2
0x00000030	外设总线时钟控制寄存器 3 （Peripheral bus Clock Control Register3）	RCC_PCLKCR3
0x00000034	外设总线时钟控制寄存器 4 （Peripheral bus Clock Control Register4）	RCC_PCLKCR4
0x00000038	LSCLK 选择寄存器 （LSCLK Select Register）	RCC_LSCLKSEL
-	-	-
-	-	-
0x00000044	AHB Master 控制寄存器 （AHB Master Control Register）	RCC_AHBMCR
-	-	-
0x00000050	外设复位使能寄存器 （Peripheral Reset Enable Register）	RCC_PRSTEN
0x00000054	AHB 外设复位寄存器 （AHB Peripherals Reset Control Register）	RCC_AHBRSTCR
0x00000058	APB 外设复位寄存器 1 （APB Peripherals Reset Control Register1）	RCC_APBRSTCR1
0x0000005C	APB 外设复位寄存器 2 （APB Peripherals Reset Control Register2）	RCC_APBRSTCR2
0x00000060	XTHF 控制寄存器 （XTHF Control Register）	RCC_XTHFCR

offset 地址	名称	符号
RCC(模块起始地址:0x40000200)		
0x00000064	RCMF 控制寄存器 （RCMF Control Register）	RCC_RCMFCR
0x00000068	RCMF 调校寄存器 （RCMF Trim Register）	RCC_RCMFTR
0x0000006C	外设工作时钟控制寄存器 1 （Peripheral Operation Clock Control Register1）	RCC_OPCCR1
0x00000070	外设工作时钟控制寄存器 2 （Peripheral Operation Clock Control Register2）	RCC_OPCCR2
0x00000074	PHY 控制寄存器 （PHY Control Register）	RCC_PHYCR
0x00000078	PHY BCK 控制寄存器 （PHY BCK Control Register）	RCC_PHYBCKCR

11.12.1 LOCKUP 复位控制寄存器（RCC_LKPCR）

名称	RCC_LKPCR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						LKUPRST_EN	-
位权限	U-0						R/W-0	U-0

位号	助记符	功能描述
31:2	**-**	RFU：未实现，读为 0
1	**LKUPRST_EN**	LOCKUP 复位使能 (Lockup Reset Enable) 1：使能 SC000 LOCKUP 复位 0：屏蔽 SC000 LOCKUP 复位
0	**-**	RFU：未实现，读为 0

11.12.2 软件复位寄存器（RCC_SOFTRST）

名称	RCC_SOFTRST							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SOFTRST[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	SOFTRST[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFTRST[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SOFTRST[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	**SOFTRST**	软件写 0x5C5C_AABB 触发全局复位 (software reset,write only)

11.12.3 复位标志寄存器（RCC_RSTFR）

名称	RCC_RSTFR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			MDFN_FLAG	NRSTN_FLAG	TESTN_FLAG	PORN_FLAG	PDRN_FLAG
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-1	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SOFTN_FLAG	IWDTN_FLAG	-	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
位权限	U-0		R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:12	**-**	RFU：未实现，读为 0
12	**MDFN_FLAG**	模式诊断超时复位标志，高有效，硬件职位，软件写 1 清零(MDFN reset Flag, write 1 to clear)
11	**NRSTN_FLAG**	NRST 引脚复位标志，高有效，软件写 1 清零 (NRST reset Flag, write 1 to clear)
10	**TESTN_FLAG**	TESTN 引脚复位标志，高有效，软件写 1 清零 (TESTN reset Flag, write 1 to clear)
9	**PORN_FLAG**	上电复位标志，高有效，软件写 1 清零 (Power-up-reset Flag, write 1 to clear)
8	**PDRN_FLAG**	下电复位标志，高有效，软件写 1 清零 (Power-down-reset Flag, write 1 to clear)
7:6	**-**	RFU：未实现，读为 0
5	**SOFTN_FLAG**	软件复位标志，高有效，软件写 1 清零 (Software reset flag, write 1 to clear)
4	**IWDTN_FLAG**	IWDT 复位标志，高有效，软件写 1 清零 (IWDT reset flag, write 1 to clear)

位号	助记符	功能描述
3	-	RFU：未实现，读为 0
2	**WWDTN_FLAG**	WWDT 复位标志，高有效，软件写 1 清零 (WWDT reset flag, write 1 to clear, write 1 to clear)
1	**LKUPN_FLAG**	LOOKUP 复位标志，高有效，软件写 1 清零 (Lockup reset flag, write 1 to clear)
0	**NVICN_FLAG**	NVIC 复位标志，高有效，软件写 1 清零 (NVIC reset flag,write 1 to clear)

11.12.4 系统时钟控制寄存器（RCC_SYSCLKCR）

名称	RCC_SYSCLKCR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				LSCATS	-	SLP_EN EXTI	-
位权限	U-0				R/W-1	U-0	R/W-1	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		APBPRES2			APBPRES1		
位权限	U-0		R/W-000			R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					AHBPRES		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		-		BCKOSEL	SYSCLKSEL		
位权限	R/W-00		U-0		R/W-0	R/W-000		

位号	助记符	功能描述
31:28	-	RFU：未实现，读为 0
27	LSCATS	LSCLK 自动切换使能 (LSCLK automatic switch Enable) 0：当检测到 XTLF 异常停振时，不会自动将 LSCLK 切换到 LPOSC，软件可以通过写LSCLKSEL 寄存器手动切换到LPOSC 1：当检测到 XTLF 异常停振时，自动使能 LPOSC 并将 LSCLK 切换到 LPOSC
26	-	RFU：未实现，读为 0
25	SLP_ENEXTI	Sleep/DeepSleep 模式下 EXTI 采样设置 (EXTI under Sleep mode Enable) 1：Sleep/DeepSleep 模式下使能外部引脚中断采样（采样时钟为 LSCLK） 0：Sleep/DeepSleep 模式下禁止外部引脚中断采样（将无法产生 EXTI 中断）
24:22	-	RFU：未实现，读为 0
21:19	APBPRES2	APB2 时钟分频选择(APB2 bus clock Prescaler) 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频 注：该寄存器位于 PD Domain

位号	助记符	功能描述
18:16	APBPRES1	APB1 时钟分频选择 (APB1bus clock Prescaler) 000/001/010/011：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
15:11	-	RFU：未实现，读为 0
10:8	AHBPRES	AHB 时钟分频选择 (AHB bus clock Prescaler) 000/001/010/011：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
7:6	RFU	Dummy 寄存器
5:4	-	RFU：未实现，读为 0
3	BCKOSEL	USB PHY BCK 输出时钟选择信号(USB clock select) 0：选择 48M BCK 输出做为系统时钟源 1：选择 120M BCK 输出的两分频做为系统时钟源
2:0	SYSCLKSEL	系统时钟源选择 (System clock select) 000：RCHF 001：XTHF 010：PLL 011：RCHF 100：RCMFPSC 101：LSCLK 110：LPOSC 111：USBBCK

11.12.5 RCHF 控制寄存器（RCC_RCHFCR）

名称	RCC_RCHFCR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				FSEL			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:16	FSEL	RCHF 频率选择寄存器 (RCHF 180lock180ncy Select)

位号	助记符	功能描述
		0000：8MHz 0001：16MHz 0010：24MHz 其他：RFU
15:1	-	RFU：未实现，读为 0
0	EN	RCHF 使能寄存器 (RCHF Enable) 1：使能 RCHF 0：关闭 RCHF

11.12.6 RCHF 调校寄存器（RCC_RCHFTR）

名称	RCC_RCHFTR							
offest	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TRIM						
位权限	U-0	R/W-100 0000						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	TRIM	RCHF 频率调校寄存器，7’h00 表示频率最低，7’h7F 表示频率最高，调校范围为中心频率+/-30%，调校步长为中心频率 0.5% 上电后芯片自动从 NVR1 读取 8MHz 调校值并写入此寄存器 软件使用其他频率时，可以自行从 NVR1 指定地址读取调校信息并写入此寄存器，从而确保输出频率准确。 (RCHF Trim)

11.12.7 PLL 控制寄存器（RCC_PLLCR）

名称	RCC_PLLCR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	DB						
位权限	U-0	R/W-010 1111						
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC			OSEL	-	INSEL	EN
位权限	R-0	R/W-000			R/W-0	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:23	-	RFU：未实现，读为 0
22:16	DB	PLL 倍频比 (PLL Divide Boost) 0011111：输出 32 倍频 0101111：输出 48 倍频
15:8	-	RFU：未实现，读为 0
7	LOCKED	PLL 锁定标志，软件通过查询此寄存器确认 PLL 已经处于锁定状态 (PLL is Locked) 1：PLL 已锁定 0：PLL 未锁定
6:4	REFPRSC	PLL 参考时钟预分频（目标是产生 1MHz 参考时钟给 PLL） (PLL reference clock prescaler) 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：12 分频 101：16 分频 110：24 分频 111：32 分频
3	OSEL	PLL 输出选择寄存器 (PLL output select) 0：选择 PLL 一倍输出作为数字电路内的 PLL 时钟 1：选择 PLL 两倍输出作为数字电路内的 PLL 时钟
2	-	RFU：未实现，读为 0
1	INSEL	PLL 输入选择寄存器 (PLL reference input select) 0：RCHF 1：XTHF
0	EN	PLL 使能寄存器 (PLL enable) 1：使能 PLL 0：关闭 PLL

11.12.8 LPOSC 控制寄存器（RCC_LPOSCCR）

名称	RCC_LPOSCCR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					LPO_CHOP_EN	LPO_ENB	LPM_LPO_OFF
位权限	U-0					R/W-1	R-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	LPO_CHOP_EN	LPOSC Chopper 使能寄存器 (LPOSC chopper enable)
1	LPO_ENB	LPOSC 使能标志信号，只读，供软件查询 LPOSC 使能状态 (LPOSC Enable Bar,read only) 0：LPOSC 处于开启状态 1：LPOSC 处于关闭状态
0	LPM_LPO_OFF	休眠模式下 LPOSC 控制寄存器 (LPOSC Off in LowPowerMode Enable) 1：休眠模式下 LPOSC 关闭 0：休眠模式下 LPOSC 开启 【注1】此控制寄存器仅在芯片睡眠模式下有效，非 Sleep/Deepsleep 模式下 LPOSC 保持开启 【注 2】XTLF 异常停振时，根据 LSCATS 寄存器配置决定是否自动使能 LPOSC

11.12.9 LPOSC 调校寄存器（RCC_LPOSCTR）

名称	RCC_LPOSCTR							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LPOTRIM							
位权限	R/W-1000 1101							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	LPOTRIM	LPOSC 调校值寄存器 (LPOSC trimming) 0000 0000：频率最低 1111 1111：频率最高

11.12.10 XTLF 控制寄存器（RCC_XTLFCR）

名称	RCC_XTLFCR
Offset	0x00000024

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				EN			
位权限	U-0				R/W-xxxx			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					IPW		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:8	EN	XTLF 使能寄存器，上电默认 XTLF 关闭 (Enable) 1010：关闭 XTLF 和 FDET 0101：使能 XTLF 和 FDET 其他：RFU 当 XTLF 工作时，软件必须写入 1010 才能将其关闭，当 XTLF 不工作时，软件必须写入 0101 才能启动
7:3	-	RFU：未实现，读为 0
2:0	IPW	XTLF 工作电流选择，电流越大表示振荡强度越高，上电复位后使用 000 档位起振，正常工作时推荐使用 100 或 011 档位，实际应根据适配晶体的实测负阻特性选择合适的电流大小 (XTLF current select) 000：450 nA 001：400 nA 010：350 nA 011：300 nA 100：250 nA 101：200 nA 110：150 nA 111：100 nA

11.12.11 外设总线时钟控制寄存器 1（RCC_PCLKCR1）

名称	RCC_PCLKCR1							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DCU_PCE	-						
位权限	R/W-1	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD_PCE	ANAC_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RTC_PCE	USB_PCE	LPT_PCE
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31	DCU_PCE	DCU 总线时钟使能，高使能 (Debug Control Unit APB bus clock enable)
30:8	-	RFU：未实现，读为 0
7	PAD_PCE	PADCFG 总线时钟使能，高使能 (GPIO controller APB bus clock enable)
6	ANAC_PCE	ANAC 总线时钟使能，高使能 (Analog controller APB bus clock enable)
5	IWDT_PCE	IWDT 总线时钟使能，高使能 (IWDT APB bus clock enable)
4	SCU_PCE	SCU 总线时钟使能，高使能 (System controller APB bus clock enable)
3	PMU_PCE	PMU 总线时钟使能，高使能 (PMU APB bus clock enable)
2	RTC_PCE	RTC 总线时钟使能，高使能 (RTC APB bus clock enable)
1	USB_PCE	USB 总线时钟使能，高使能 (USB device APB bus clock enable)
0	LPT_PCE	LPTIM32 总线时钟使能，高使能(LPTIM APB bus clock enable)

11.12.12 外设总线时钟控制寄存器 2（RCC_PCLKCR2）

名称	RCC_PCLKCR2							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						HDIV_PCE	ADC_PCE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT_PCE	RAMBIST_PCE	FLASH_PCE	DMA_PCE	LCD_PCE	AES_PCE	TRNG_PCE	CRC_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	HDIV_PCE	硬件除法器总线时钟使能，高使能 (Hardware Divider APB bus clock enable)
8	ADC_PCE	ADC 总线时钟使能，高使能 (ADC controller APB bus clock enable)
7	WWDT_PCE	WWDT 总线时钟使能，高使能 (WWDT APB bus clock enable)
6	RAMBIST_PCE	RAMBIST 总线时钟使能，高使能 (RAMBIST APB bus clock

位号	助记符	功能描述
		enable)
5	FLASH_PCE	Flash 擦写控制器总线时钟使能，高使能 (Flash interface APB bus clock enable)
4	DMA_PCE	DMA 总线时钟使能，高使能 (DMA APB bus clock enable)
3	LCD_PCE	LCD 总线时钟使能，高使能 (LCD APB bus clock enable)
2	AES_PCE	AES 总线时钟使能，高使能 (AES APB bus clock enable)
1	RNG_PCE	RNG 总线时钟使能，高使能 (RNG APB bus clock enable)
0	CRC_PCE	CRC 总线时钟使能，高使能 (CRC APB bus clock enable)

11.12.13 外设总线时钟控制寄存器 3（RCC_PCLKCR3）

名称	RCC_PCLKCR3							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							I2C_PCE
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					LPUART1_PCE	-	U7816_PCE
位权限	U-0					R/W-0	U-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LPUART0_PCE	UCIR_PCE	UART5_PCE	UART4_PCE	-		UART1_PCE	UART0_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SPI2_PCE	SPI1_PCE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24	I2C_PCE	I2C 总线时钟使能，高有效(I2C APB bus clock enable)
23:19	-	RFU：未实现，读为 0
18	LPUART1_PCE	LPUART1 总线时钟使能，高有效(LPUART1 APB bus clock enable)
17	-	RFU：未实现，读为 0
16	U7816_PCE	7816 总线时钟使能，高有效(U7816 APB bus clock enable)
15	LPUART0_PCE	LPUART 总线时钟使能，高有效(LPUART0 APB bus clock enable)
14	UCIR_PCE	UART 红外调制工作时钟使能，高有效(UART infra-red APB bus clock enable)
13	UART5_PCE	UART5 总线时钟使能，高有效(UART5 APB bus clock enable)
12	UART4_PCE	UART4 总线时钟使能，高有效(UART4 APB bus clock enable)
11:10	-	RFU：未实现，读为 0
9	UART1_PCE	UART1 总线时钟使能，高有效(UART1 APB bus clock enable)
8	UART0_PCE	UART0 总线时钟使能，高有效(UART0 APB bus clock enable)
7:2	-	RFU：未实现，读为 0
1	SPI2_PCE	SPI2 总线时钟使能，高有效(SPI2 APB bus clock enable)

位号	助记符	功能描述
0	SPI1_PCE	SPI1 总线时钟使能，高有效(SPI1 APB bus clock enable)

11.12.14 外设总线时钟控制寄存器 4（RCC_PCLKCR4）

名称	RCC_PCLKCR4							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			AT_PCE	GT1_PCE	GT0_PCE	-	BT_PCE
位权限	U-0			R/W-0	R/W-0	R/W-0	U-0	R/W-0

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4	AT_PCE	高级定时器总线时钟使能，高有效(ATIM APB bus clock enable)
3	GT1_PCE	通用定时器 1 总线时钟使能，高有效(GPTIM1 APB bus clock enable)
2	GT0_PCE	通用定时器 0 总线时钟使能，高有效(GPTIM0 APB bus clock enable)
1	-	RFU：未实现，读为 0
0	BT_PCE	基本定时器 0 总线时钟使能，高有效(BSTIM APB bus clock enable)

11.12.15 LSCLK 选择寄存器（RCC_LSCLKSEL）

名称	**RCC_LSCLKSEL**							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEL							
位权限	R/W-0000 0001							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	SEL	当 LSCLK 为 XTLF 时，软件对此地址写 0x55，会将 LSCLK 源头切换到 LPOSC 当 LSCLK 为 LPOSC 时，软件对此地址写 0xAA，会将 LSCLK 源头切换到 XTLF 写任意其他值，不改变当前 LSCLK；此寄存器仅在 LSCATS 为 0 时有效

11.12.16 AHB Master 控制寄存器（RCC_AHBMCR）

名称	RCC_AHBMCR							
Offset	0x00000044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							MPRIL
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	MPRIL	AHB Master 优先级配置寄存器(Priority Config) 0：DMA 优先 1：CPU 优先

11.12.17 外设复位使能寄存器（RCC_PRSTEN）

名称	RCC_PRSTEN							
Offset	0x00000050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PERHRSTEN[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PERHRSTEN[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PERHRSTEN[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PERHRSTEN[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	PERHRSTEN	外设模块复位使能，32bit 虚寄存器，只写 (Peripheral Reset Enable,write only) 软件对此地址写 0x1357_9BDF，使能外设复位功能，此后可以通过外设模块复位寄存器复位各个模块 软件对此地址写任意其他数据，将关闭外设复位功能

11.12.18 AHB 外设复位寄存器（RCC_AHBRSTCR）

名称	RCC_AHBRSTCR							
Offset	0x00000054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						USBRST	DMARST
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	USBRST	USB 模块复位，软件写 1 复位，写 0 撤销复位 (USB reset Enable) 0：不复位 1：复位
0	DMARST	DMA 模块复位，软件写 1 复位，写 0 撤销复位 (DMA reset Enable) 0：不复位 1：复位

11.12.19 APB 外设复位寄存器 1（RCC_APBRSTCR1）

名称	RCC_APBRSTCR1							
Offset	0x00000058							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	-				GPT1RST	GPT0RST
位权限	R/W-0	R/W-0	U-0				R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							LCDRST
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-	U7816RST	-			SPI2RST	-	
位权限	U-0	R/W-0	U-0			R/W-0	U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	LPUART0RST	-		I2C1RST	-		LPT32RST
位权限	U-0	R/W-0	U-0		R/W-0	U-0		R/W-0

位号	助记符	功能描述
31	UART5RST	UART5 模块复位，软件写 1 复位，写 0 撤销复位 (UART5 reset Enable) 0：不复位 1：复位
30	UART4RST	UART4 模块复位，软件写 1 复位，写 0 撤销复位 (UART4 reset Enable) 0：不复位 1：复位
29:26	-	RFU：未实现，读为 0
25	GPT1RST	GPTIM1 模块复位，软件写 1 复位，写 0 撤销复位 (GPTIM1 reset Enable) 0：不复位 1：复位
24	GPT0RST	GPTIM0 模块复位，软件写 1 复位，写 0 撤销复位 (GPTIM0 reset Enable) 0：不复位 1：复位
23:17	-	RFU：未实现，读为 0
16	LCDRST	LCD 模块复位，软件写 1 复位，写 0 撤销复位 (LCD reset Enable) 0：不复位 1：复位
15	-	RFU：未实现，读为 0
14	U7816RST	U7816 模块复位，软件写 1 复位，写 0 撤销复位 (U7816 reset Enable) 0：不复位 1：复位
13:11	-	RFU：未实现，读为 0
10	SPI2RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 (SPI2 reset) 0：不复位 1：复位
9:7	-	RFU：未实现，读为 0
6	LPUART0RST	EUART0 模块复位，软件写 1 复位，写 0 撤销复位 (LPUART0 reset Enable) 0：不复位 1：复位
5:4	-	RFU：未实现，读为 0
3	I2C1RST	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 (I2C1 reset Enable) 0：不复位 1：复位
2:1	-	RFU：未实现，读为 0
0	LPT32RST	LPTIM32 模块复位，软件写 1 复位，写 0 撤销复位 (LPTIM

位号	助记符	功能描述
		resetEnable) 0：不复位 1：复位

11.12.20 APB 外设复位寄存器 2（RCC_APBRSTCR2）

名称	RCC_APBRSTCR2							
Offset	0x0000005C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ATRST	-		BT32RST	-			ADCCRST
位权限	R/W-0	U-0		R/W-0	U-0			R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	OPARST	-		HDVRST	AESRST	CRCRST	RNGRST
位权限	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			UART1RST	UART0RST	-	SPI1RST	UCIRRST
位权限	U-0			R/W-0	R/W-0	U-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LPUART1RST	-						
位权限	R/W-0	U-0						

位号	助记符	功能描述
31	ATRST	ATIM 模块复位，软件写 1 复位，写 0 撤销复位 (ATIM reset Enable) 0：不复位 1：复位
30:29	-	RFU：未实现，读为 0
28	BTRST	BSTIM32 模块复位，软件写 1 复位，写 0 撤销复位 (BSTIM32 reset Enable) 0：不复位 1：复位
27:25	-	RFU：未实现，读为 0
24	ADCCRST	ADC 控制器复位，软件写 1 复位，写 0 撤销复位 (ADC controller reset Enable) 0：不复位 1：复位
23	ADCRST	ADC 模块复位，软件写 1 复位，写 0 撤销复位 (ADC reset Enable) 0：不复位 1：复位
22	OPARST	运放模块复位，软件写 1 复位，写 0 撤销复位 (OPA reset Enable) 0：不复位 1：复位
21:20	-	RFU：未实现，读为 0
19	HDVRST	硬件除法器复位，软件写 1 复位，写 0 撤销复位 (Hardware Divider Reset Enable)

位号	助记符	功能描述
		0：不复位 1：复位
18	AESRST	AES模块复位，软件写1复位，写0撤销复位 (AES reset Enable) 0：不复位 1：复位
17	CRCRST	CRC 模块复位，软件写 1 复位，写 0 撤销复位 (CRC reset Enable) 0：不复位 1：复位
16	RNGRST	RNG 模块复位，软件写 1 复位，写 0 撤销复位 (RNG reset Enable) 0：不复位 1：复位
15:13	-	RFU：未实现，读为 0
12	UART1RST	UART1 模块复位，软件写 1 复位，写 0 撤销复位 (UART1 reset Enable) 0：不复位 1：复位
11	UART0RST	UART0 模块复位，软件写 1 复位，写 0 撤销复位 (UART0 reset Enable) 0：不复位 1：复位
10	-	RFU：未实现，读为 0
9	SPI1RST	SPI1 模块复位，软件写 1 复位，写 0 撤销复位(SPI1reset Enable) 0：不复位 1：复位
8	UCIRRST	红外调制模块复位，软件写 1 复位，写 0 撤销复位(UCIR reset Enable) 0：不复位 1：复位
7	LPUART1RST	LPUART1 模块复位，软件写 1 复位，写 0 撤销复位(LPUART1 reset Enable) 0：不复位 1：复位
6:0	-	RFU：未实现，读为 0

11.12.21 XTHF 控制寄存器（RCC_XTHFCR）

名称	RCC_XTHFCR							
Offset	0x00000060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-					CFG		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:11	-	RFU：未实现，读为 0
10:8	CFG	XTHF 振荡强度配置 (XTHF oscillation strength config) 000：最弱 111：最强
7:1	-	RFU：未实现，读为 0
0	EN	XTHF 使能寄存器 (XTHF enable) 0：关闭 XTHF 1：使能 XTHF

11.12.22 RCMF 控制寄存器（RCC_RCMFCR）

名称	RCC_RCMFCR							
Offset	0x00000064							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						PSC	
位权限	U-0						R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:18	-	RFU：未实现，读为 0
17:16	PSC	RCMF 输出预分频 (Prescaler) 00：不分频 01：4 分频 10：8 分频 11：16 分频
15:1	-	RFU：未实现，读为 0
0	EN	RCMF 使能寄存器 (RCMFEnable) 0：关闭 RCMF 1：打开 RCMF

11.12.23 RCMF 调校寄存器（RCC_RCMFTR）

名称	RCC_RCMFTR

Offset	0x00000068							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TRIM						
位权限	U-0	R/W-100 0000						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	TRIM	RCMF 频率调校寄存器，7’h00 表示频率最低，7’h7F 表示频率最高，调校范围为中心频率+/-30%，调校步长为中心频率 1% (RCMF trimming)

11.12.24 外设工作时钟控制寄存器 1（RCC_OPCCR1）

名称	RCC_OPCCR1							
Offset	0x0000006C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	EXTICKS	LPUART1CKE	LPUART0CKE	LPUART1CKS		LPUART0CKS	
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			I2CCKE	-		I2CCKS	
位权限	U-0			R/W-0	U-0		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ATCKE	-					UART1CKE	UART0CKE
位权限	R/W-0	U-0					R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ATCKS		-		UART1CKS		UART0CKS	
位权限	R/W-00		U-0		R/W-00		R/W-00	

位号	助记符	功能描述
31	EXTICKE	EXTI 工作时钟使能，高有效(External interrupt operation clock Enable)
30	EXTICKS	EXTI 中断采样时钟选择 (External interrupt sampling clock select) 1：外部引脚中断使用 LSCLK 采样 0：外部引脚中断使用 HCLK 采样 *建议在关闭所有 EXTI 中断的情况下设置，设置完成后再使能 EXTI 中断
29	LPUART1CKE	LPUART1 工作时钟使能，高有效 (LPUART1 operation clock

位号	助记符	功能描述
		enable)
28	LPUART0CKE	LPUART0 工作时钟使能，高有效 (LPUART0 operation clock enable)
27:26	LPUART1CKS	LPUART1 工作时钟选择 (LPUART1 operation clock select) 00：LSCLK 01：RCHF 分频 10：RCMF 分频 11：RFU
25:24	LPUART0CKS	LPUART0 工作时钟选择 (LPUART0 operation clock select) 00：LSCLK 01：RCHF 分频 10：RCMF 分频 11：RFU
23:21	-	RFU：未实现，读为 0
20	I2CCKE	I2C 工作时钟使能
19:18	-	RFU：未实现，读为 0
17:16	I2CCKS	I2C 主机工作时钟选择 (I2C operation clock select) 00：APBCLK 01：RCHF 10：SYSCLK 11：RCMF_PSC
15	ATCKE	ATIM 工作时钟使能寄存器，高有效 (ATIM operation clock Enable)
14:10	-	RFU：未实现，读为 0
9	UART1CKE	UART1 工作时钟使能，高有效 (UART1 operation clock enable)
8	UART0CKE	UART0 工作时钟使能，高有效 (UART0 operation clock enable)
7:6	ATCKS	ATIM 工作时钟源选择寄存器 (ATIM operation clock select) 00：APBCLK2 01：USB PHY BCK 120M 10：APBCLK2 11：PLL 两倍频
5:4	-	RFU：未实现，读为 0
3:2	UART1CKS	UART1 工作时钟选择 (UART1 operation clock select) 00：APBCLK 01：RCHF 10：SYSCLK 11：RCMF_PSC
1:0	UART0CKS	UART0 工作时钟选择 (UART0 operation clock select) 00：APBCLK 01：RCHF 10：SYSCLK 11：RCMF_PSC

11.12.25 外设工作时钟控制寄存器 2（RCC_OPCCR2）

名称	RCC_OPCCR2							
Offset	0x00000070							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	RNGPRSC			-	ADCPRSC		

位权限	U-0	R/W-000			U-0	R/W-000		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	USBREFCKE	FLASHCKE	RNGCKE	ADCCKE	USBREFCKS		ADCCKS	
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			LPTCKE	-		LPTCKS	
位权限	U-0			R/W-0	U-0		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BTCKE	-		BTCKS	
位权限	U-0			R/W-0	U-0		R/W-00	

位号	助记符	功能描述
31	-	RFU：未实现，读为 0
30:28	RNGPRSC	随机数发生器工作时钟分频 (RNG operation clock prescaler) 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
27	-	RFU：未实现，读为 0
26:24	ADCPRSC	ADC 工作时钟预分频 (ADC operation clock prescaler) 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110/111：RFU
23	USBREFCKE	USB 参考时钟使能 (USB reference clock enable) 0：关闭 USB 参考时钟输出 1：使能 USB 参考时钟输出
22	FLASHCKE	Flash 擦写时钟使能，高有效 (Flash erase/program clock enable)
21	RNGCKE	随机数发生器工作时钟使能，高有效 (RNG operation clock enable)
20	ADCCKE	ADC 工作时钟使能，高有效 (ADC operation clock enable)
19:18	USBREFCKS	USB 参考时钟源选择 (USB reference clock select) 00/11：XTLF（32768Hz） 01：XTHF（12MHz） 10：RCHF 分频
17:16	ADCCKS	ADC 工作时钟选择 (ADC operation clock select) 00：RCMF_PSC 01：RCHF 10：XTHF 11：PLL
15:13	-	RFU：未实现，读为 0
12	LPTCKE	LPTIM 工作时钟使能，高有效 (LPTIM operation clock enable)

位号	助记符	功能描述
11:10	-	RFU：未实现，读为 0
9:8	LPTCKS	LPTIM 工作时钟选择 (LPTIM operation clock select) 00：APBCLK1 01：LSCLK 10：LPOSC 11：RCMF_PSC
7:5	-	RFU：未实现，读为 0
4	BTCKE	BSTIM 工作时钟使能，高有效 (BSTIM operation clock enable)
3:2	-	RFU：未实现，读为 0
1:0	BTCKS	BSTIM 工作时钟源选择 (BSTIM operation clock select) 00：APBCLK2 01：LSCLK 10：LPOSC 11：RCMF_PSC

11.12.26 PHY 控制寄存器（RCC_PHYCR）

名称	RCC_PHYCR							
offset	0x00000074							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			PHY_PONRST_B	PD	PLVREADY_33V	BCKPD	NONCRY_RSTB
位权限	U-0			R/W-0	R/W-1	R/W-0	R/W-1	R/W-0

位号	助记符	功能描述
31:5	**-**	RFU：未实现，读为 0
4	**PHY_PONRST_B**	USB PHY 复位，低有效；上电默认为 0，软件写 1 后撤销复位 (PHY Power-On_Reset Bar Enable) 0：复位 USB PHY 1：撤销复位 USB PHY **注：当系统时钟来自 PHY 时，该寄存器无法写 0；** *在 FM33LC0xx 系列中，在使用 USB 进行通信或使用 USB BCK 时钟前，必须先置位此寄存器，撤销 USB PHY 的复位*
3	**PD**	PHY transceiver power down 控制信号 (PHY Power Down Enable) 1：PHY 收发器处于待机模式 0：PHY 收发器处于工作模式

位号	助记符	功能描述
2	**PLVREADY_33V**	VDD15D 电源建立标志，在使用 USB PHY 之前，软件需要将此寄存器置位 (Power Low Voltage Ready Enable) 1：VDD15D 电源已经建立 0：VDD15D 电源未建立
1	**BCKPD**	PHY 的 BCK 模块使能 (Built-in-Clock Power Down Enable) 1：BCK 不使能，不输出时钟 0：BCK 使能，输出时钟 **注：当系统时钟来自 PHY 时，该寄存器无法写 1；**
0	**NONCRY_RSTB**	BCK 模块的复位信号 (Non-Crystal Reset Bar Enable) 1：BCK 复位释放 0：BCK 复位有效 **注：当系统时钟来自 PHY 时，该寄存器无法写 0；**

11.12.27 PHY BCK 控制寄存器（RCC_PHYBCKCR）

名称	RCC_PHYBCKCR							
Offset	0x00000078							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CK48M_EN
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CLK_RDY	-						OUTCLKSEL
位权限	R	U-0						R/W-0

位号	助记符	功能描述
31:9	**-**	RFU：未实现，读为 0
8	**CK48M_EN**	48M 时钟输出使能 (Clock 48Mhz Enable) 1：允许 PHY 输出 48M 时钟，USB 通信前必须置位 0：禁止 PHY 输出 48M 时钟
7	**CLK_RDY**	时钟跟踪完成标志 (Clock Ready Flag,read only) 1：表示时钟跟踪完成 0：时钟跟踪尚未完成 *注：当 OUTCLKSEL=1 的情况下，此寄存器保持为 0*
6:1	**-**	RFU：未实现，读为 0
0	**OUTCLKSEL**	时钟跟踪源选择 (Output Clock Select) 1：使用 CORECLKIN 参考时钟来作为时钟跟踪源 0：使用 USB 总线下发的 SOF 来做时钟跟踪

12 停振检测（FDET）

12.1 低频停振检测

FM33LC0XX带有片上低频晶体停振检测电路，使能后可以持续检测XTLF输出，当发现XTLF停振时，产生报警中断，软件可以通过LSCATS寄存器决定是否自动将LSCLK切换到LPOSC。

当LSCATS=1时，FDET检测到XTLF停振时，硬件会自动使能LPOSC并将LSCLK切换为LPOSC输出；当LSCATS=0时，停振检测只会产生报警中断，并不会自动切换时钟。

在XTLF停振状态下，软件也可以通过置位LSCATS来切换XTLF。

停振检测电路总是与XTLF同时打开或关闭，无法单独关闭，一旦XTLF使能，停振检测电路就会自动打开；当XTLF关闭时，停振检测也会自动关闭，避免误触发停振报警。

12.2 高频停振检测

FM33LC0XX带有片上高频晶体停振检测电路，与XTHF电路一起使能或关闭。停振检测使能后可以持续检测XTHF输出，当发现XTHF停振时，会产生报警中断，同时产生高级定时器刹车信号；如果XTHF正在被直接或者间接的用作系统工作时钟（直接指SYSCLK选为XTHF，间接指SYSCLK选为PLL同时PLL使用XTHF为输入参考时钟），则停振信号将自动使能RCHF并将SYSCLK切换到RCHF，以避免高频晶体意外停振导致系统死机。

停振检测电路总是与XTHF同时打开或关闭，无法单独关闭，一旦XTHF使能，停振检测电路就会自动打开；当XTHF关闭时，停振检测也会自动关闭，避免误触发停振报警。

HFDET停振检测阈值约200KHz，即XTHF时钟频率低于200KHz时触发停振报警；HFDET工作电流约1.2uA。

12.3 寄存器

offset 地址	名称	符号
FDET(模块起始地址:0x4001A838)		
0x00000000	停振检测中断使能寄存器（XTLF Oscillation Fail Detection Interrupt Enable Register）	FDET_IER
0x00000004	停振检测中断标志寄存器（XTLF Oscillation Fail Detection Interrupt Status Register）	FDET_ISR

12.3.1 停振检测中断使能寄存器（FDET_IER）

名称	FDET_IER							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						HFDET_IE	LFDET_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	HFDET_IE	XTHF 高频检测报警中断使能，1 有效 (XTHF fail detect interrupt enable)
0	LFDET_IE	XTLF 低频检测报警中断使能，1 有效 (XTLF fail detect interrupt enable)

12.3.2 停振检测中断标志寄存器（FDET_ISR）

名称	FDET_ISR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						HFDETO	LFDETO
位权限	U-0						R-0	R-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						HFDETIF	LFDETIF
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	HFDETO	高频晶体停振检测模块输出 (XTHF fail detect output) 1：XTHF 未停振 0：XTHF 停振
8	LFDETO	低频晶体停振检测模块输出 (XTLF fail detect output) 1：XTLF 未停振 0：XTLF 停振
7:2	-	RFU：未实现，读为 0
1	HFDETIF	高频停振检测中断标志寄存器，XTHF 停振时硬件异步置位，软件写 1 清零；只有在 FFDETO 不为 0 的情况下才能够清除此寄存器 (XTHF fail detect interrupt flag, write 1 to clear)
0	LFDETIF	低频停振检测中断标志寄存器，XTLF 停振时硬件异步置位，软件写 1 清零；只有在 LFDETO 不为 0 的情况下才能够清除此寄存器 (XTLF fail detect interrupt flag,write 1 to clear)

13 电源电压监测（SVD）

13.1 概述

电源检测电路主要用来监测外部主电源的供电情况，及时检测到外部主电源欠压或恢复的情况，并给出中断信号。电源检测电路可关断或周期使能以节省功耗。

特点：

- 监测主电源，电压低于或高于设定的阈值时产生中断
- 低压检测范围 1.8V~4.8V，15 级可编程阈值档位，档位间隔 0.214V
- 电压检测迟滞窗口 0.1V
- 可关断或间歇式工作
- 支持 1 个外部通道直接输入与内部基准电压源比较
- 外部通道支持 100mV 窗口

13.2 结构框图

下图是电源检测电路的模块框图。

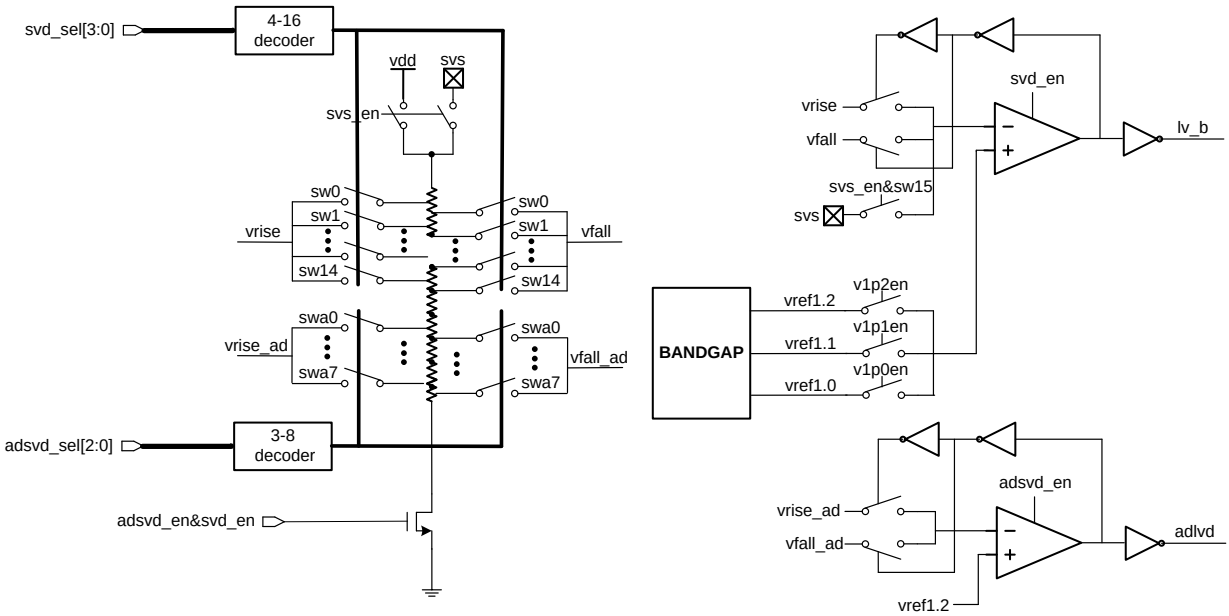


图 13-1 低压检测电路框图

SVD共有15个内部通道和1个外部通道，内部通道用于芯片电源检测，外部通道用于外部输入信号与内部基准电压比较。

SVD工作时序示意图：

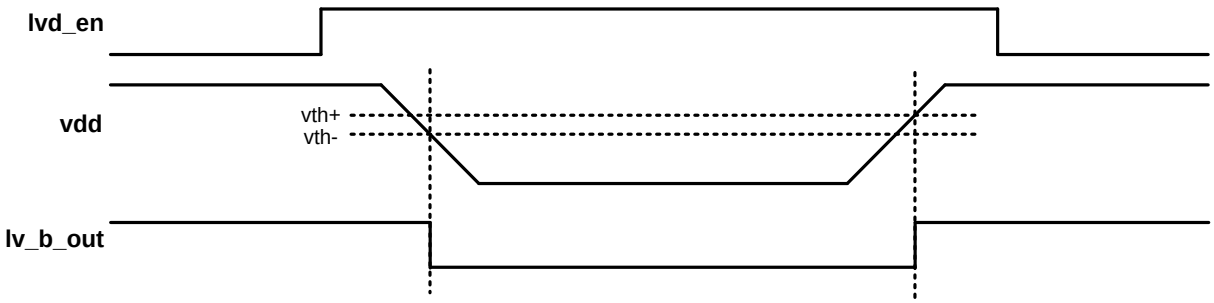


图 13-2SVD 工作时序

13.3 引脚定义

SVD模块可以直接检测芯片电源（VDD），也可以通过SVS引脚（PA15）检测外部电压信号。

检测SVS输入时，需要将PA15的FCR寄存器配置为11（analog function）。

13.4 功能描述

电源检测电路可以用来检测主电源电压及外部电压。电源电压通过分压电阻产生15级检测电平，检测范围1.8V~4.8V，每级相差0.214V；另外还支持1路外部输入检测电平，共16级检测电平。通过16选1 MUX送入比较器，与内部参考电压相比较，根据低压报警阈值设置，若待检测电平低于参考电压，引起输出电压跳变，会产生欠压中断，通知MCU及时处理该事件；而当VDD恢复至阈值以上（有大约0.1V迟滞窗口），则会产生欠压恢复中断。

电源检测电路可由软件配置使能或禁止工作。为节省功耗，使能时又可分为常使能和间歇工作两种模式。间歇工作时，可通过设置节DSEF寄存器设置开启时间间隔。

常使能条件下SVD从欠压到过压有0.1V回滞窗口，而间歇使能情况下没有回滞窗口；对于内部通道，可以通过软件配合，即欠压中断后人为设置一个较高的过压阈值，来解决窗口问题。而对于SVS通道，则需要特殊设计，由数字电路锁存上一次间歇窗口的判决结果，作为本次间歇窗口中的阈值选择依据，从而实现SVS的下降阈值和上升阈值选择。相应的，BG需要输出1.0V、1.1V、1.2V三个基准电压。

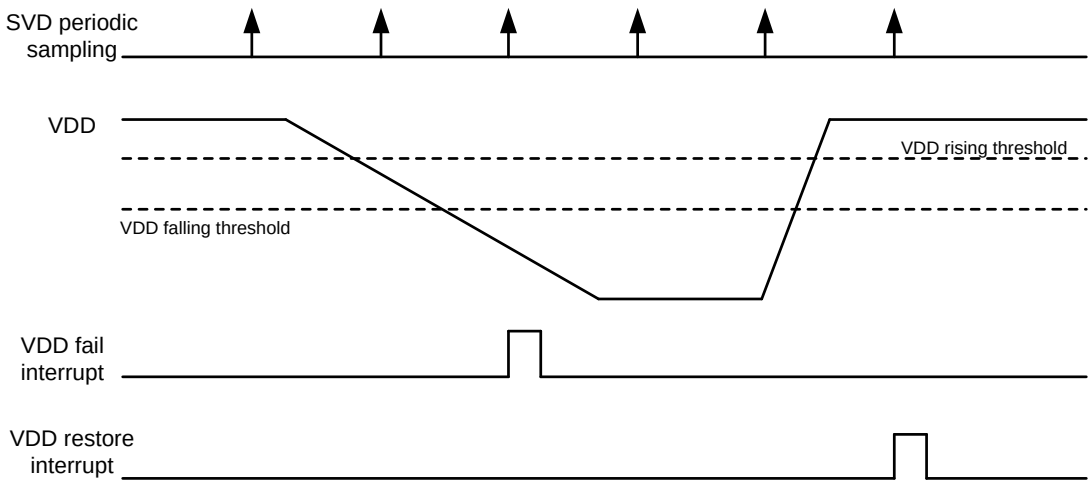


图 13-3 电源检测电路间歇工作模式

间歇工作时，当软件使能SVD的间隙使能后，SVD并不一定会立刻工作，而是要等待下一个开启窗口到来。而常使能情况下，软件开启SVD后经过一到两个LSCLK时钟同步周期后，SVD就会开始工作。SVD开启后到输出稳定建立大约需要100us时间，软件读取SVD输出时需要注意。

如果芯片进入休眠模式后关闭了所有时钟，又希望使用SVD，则需要在休眠前将SVD设置为常使能，并且关闭数字滤波功能。

工作模式说明：

- 在常使能/内部通道模式下，检测阈值有窗口，下降阈值和上升阈值窗口为0.1V，不使能到使能时检测下降阈值。
- 在间歇使能/内部通道模式下，在每次间歇使能启动时（即不使能到使能时）检测下降阈值，因此没有阈值窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将阈值档位恢复。
- 在常使能/外部通道模式下，输入的基准电压为三档位输入，分别为0.8V、0.75V、0.7V，检测阈值没有窗口，需要软件配合，即在检测到欠压时，软件将阈值档位调高一档；在检测到非欠压时，软件将档位恢复。
- 在间歇使能/外部通道模式下，输入的基准电压为三档位输入，分别为0.8V、0.75V、0.7V，检测阈值没有窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将档位恢复。

13.5 间歇使能模式

在休眠模式下可以通过间歇使能来降低SVD的平均功耗。在DeepSleep模式下，VREF1p22默认关闭，此时间歇启动SVD的同时还要启动VREF1p22，数字电路需要先等待VREF1p22建立，才能采样SVD输出。

VREF1p22使能后硬件自动等待1.5ms启动时间，然后再等待不少于100us SVD建立时间，硬件才锁存SVD输出信号。SVD开启窗口中，SVD加Vref1p22总功耗小于3uA。如果开启间隔是62.5ms，则平均电流大约为70nA；如果开启间隔是1s，则平均电流小于5nA。

13.6 外部电源检测

SVD除了可以检测芯片电源，也可以对外部电压信号进行掉电或上电检测。

外部电源检测通过SVS引脚（PA15）实现，SVS的输入可以采用外部电阻分压或内部电阻分压后，再输入到比较器进行检测。当外部待检测电压高于芯片电源时，推荐使用外部电阻分压的方式，得到低于芯片电源的SVS输入；当外部待检测电压低于或等于芯片电源时，则可以直接输入到SVS管脚，通过内部电阻分压进行检测。使用SVS前需要将PA15引脚设置为模拟功能。

下图为外部电阻分压的外部电源检测：

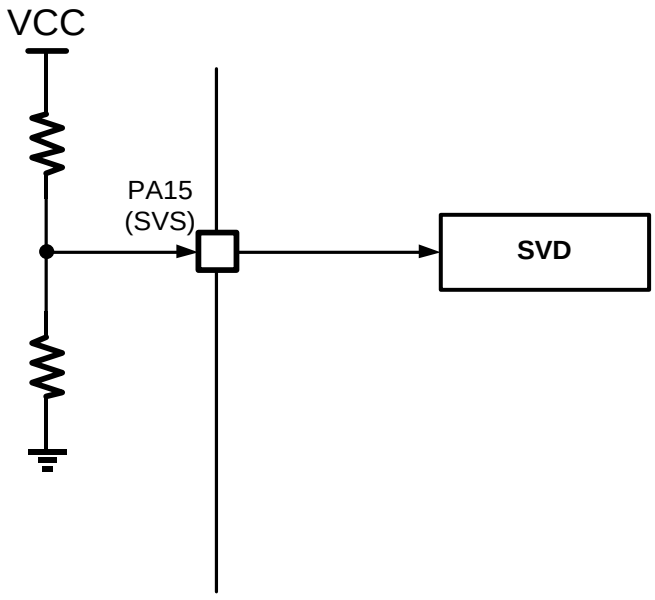


图 13-4 外部电阻分压用于外部电源检测

下图为内部电阻分压的外部电源检测：

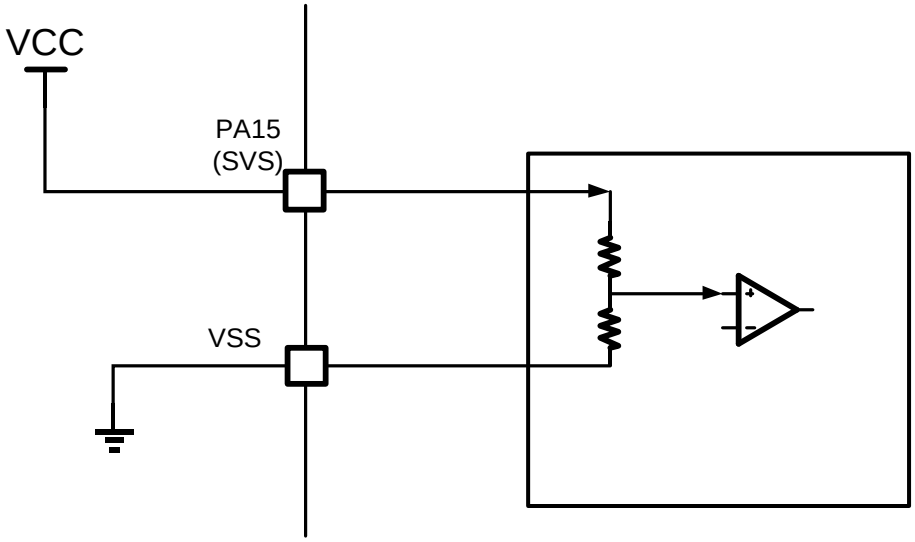


图 13-5 内部电阻分压用于外部电源检测

寄存器配置方法如下表：

SVSEN	SVDLVL	说明
0	X	外部电源检测通道关闭，仅检测内部电源电压
1	1111	外部电压输入不做内部分压，直接输入到比较器与内部基准电压比较
	0000~1110	外部电压输入先经过内部电阻分压，然后再输入到比较器与内部基准电压比较 分压后的档位参见后续章节描述

13.7 电源检测阈值

通过SVSEN和SVDLVL寄存器可以选择电压检测对象和检测阈值。

内部电源检测：SVSEN = 0，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 100，比较基准1.2V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.800	1.900
0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257

1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	N/A	N/A

内部电源检测：SVSEN = 0，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 010，比较基准1.1V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	N/A	N/A

内部电源检测：SVSEN = 0，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 001，比较基准1.0V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476
0110	2.572	2.655
0111	2.750	2.833
1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	N/A	N/A

外部电压检测：SVSEN =1，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 100，比较基准1.2V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.800	1.900
0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	1.2	1.2

外部电压检测：SVSEN =1，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 010，比较基准1.1V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	1.1	1.1

外部电压检测：SVSEN =1，{VREF1P2EN, VREF1P1EN, VREF1P0EN} = 001，比较基准1.0V

SVDLVL	上升阈值（V）	下降阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476

0110	2.572	2.655
0111	2.750	2.833
1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	1.0	1.0

13.8 寄存器

offset 地址	名称	符号
SVD(模块起始地址:0x4001A824)		
0x00000000	SVD 配置寄存器 （SVD Config Register）	SVD_CFGR
0x00000004	SVD 控制寄存器 （SVD Control Register）	SVD_CR
0x00000008	SVD 中断使能寄存器 （SVD Interrupt Enable Register）	SVD_IER
0x0000000C	SVD 状态和标志寄存器 （SVD Interrupt Status Register）	SVD_ISR
0x00000010	SVD 参考电压选择寄存器 （SVD reference Voltage Select Register）	SVD_VSR

13.8.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				ADSVD_SEL			ADSVD_EN
位权限	U-0				R/W-110			R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LVL				DFEN	MOD	ITVL	
位权限	R/W-0000				R/W-1	R/W-0	R/W-00	

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:9	ADSVD_SEL	ADC 电源检测档位配置 (ADC supply monitor select) 000：3.300V 001：3.514V 010：3.729V 011：3.943V 100：4.157V 101：4.371V 110：4.586V 111：4.800V
8	ADSVD_EN	ADC 电源检测功能，在 5V 系统供电方案下建议在使用 ADC 之前打开这个功能 (ADC supply monitor enable) 使能 ADC 电源检测功能不需要使能 SVDEN
7:4	LVL	SVD 报警阈值设置，档位定义参见 13.7 电源检测阈值 (SVD voltage level)

位号	助记符	功能描述
3	DFEN	数字滤波使能（SVDMODE=1 时必须置 1）(digital filter enable) 1：启动 SVD 输出的数字滤波 0：关闭 SVD 输出的数字滤波
2	MOD	SVD 工作模式选择，配置模式后还要置位 SVDEN 才会启动 SVD (SVD working mode) 1：间歇使能模式 0：常使能模式 注意：间歇使能模式下必须开启数字滤波
1:0	ITVL	SVD Interval，SVD 间歇使能间隔 (SVD periodic interval) 00：62.5ms 01：256ms 10：1s 11：4s

13.8.2 SVD 控制寄存器（SVD_CR）

名称	SVD_CR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SVSEN	EN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	TE	SVD 测试使能，避免写 1 (SVD test enable)
7:2	-	RFU：未实现，读为 0
1	SVSEN	SVS 外部电源检测通道控制信号 (SVS external monitor channel enable) 0：SVS 通道关闭 1：SVS 通道使能 当 EN=1 时，根据 SVDLVL 寄存器可以设置 SVS 输入后是否经过内部电阻分压；如果 LVL=1111，则 SVS 输入不做分压，如果 LVL != 1111，则 SVS 输入经过内部电阻分压。
0	EN	SVD 使能 (SVD enable) 1：启动 SVD 0：关闭 SVD

13.8.3 SVD 中断使能寄存器（SVD_IER）

名称	SVD_IER							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PFIE	PRIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	PFIE	电源跌落中断使能寄存器(Power Fall interrupt enable) 1：允许电源跌落中断 0：禁止中断
0	PRIE	电源恢复中断使能寄存器(Power Rise interrupt enable) 1：允许电源恢复中断 0：禁止中断

13.8.4 SVD 状态和标志寄存器（SVD_ISR）

名称	SVD_ISR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADLVDO
位权限	U-0							R-x
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SVDO
位权限	U-0							R-x
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SVDR	-					PFF	PRF
位权限	R-x	U-0					R/W-0	R/W-0

位号	助记符	功能描述
31:17	-	RFU：未实现，读为 0
16	ADLVDO	ADC 电源检测输出信号，当 ADC 电源低于设定阈值时输出 1，软件只读 (ADC supply monitor output)
15:9	-	RFU：未实现，读为 0

位号	助记符	功能描述
8	SVDO	SVD 电源检测输出 (SVD output) 1：电源电压高于 SVD 当前阈值 0：电源电压低于 SVD 当前阈值
7	SVDR	SVD 输出锁存信号，数字电路锁存的 SVD 状态 (SVD registered output)
6:2	-	RFU：未实现，读为 0
1	PFF	电源跌落中断标志寄存器，电源电压跌落到 SVD 阈值之下时置位，软件写 1 清零 (Power fall flag,write 1 to clear)
0	PRF	电源恢复中断标志寄存器，电源电压上升到 SVD 阈值之上时置位，软件写 1 清零 (Power rise flag，write1to clear)

13.8.5 SVD 参考电压选择寄存器（SVD_VSR）

名称	SVD_VSR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit0	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					V1P2EN	V1P1EN	V1P0EN
位权限	U-0					R/W-1	R/W-0	R/W-1

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	V1P2EN	1.2V 基准输入使能信号 (1.2V reference enable) 1：使能 1.2V 基准输入 0：关闭 1.2V 基准输入
1	V1P1EN	1.1V 基准输入使能信号 (1.1V reference enable) 1：使能 1.1V 基准输入 0：关闭 1.1V 基准输入
0	V1P0EN	1.0V 基准输入使能信号 (1.0V reference enable) 1：使能 1.0V 基准输入 0：关闭 1.0V 基准输入

14 AES 硬件运算单元（AES）

14.1 功能描述

AES单元主要功能如下：

- 支持解密密钥扩展
- 支持128bit/192bit/256bit的密钥长度
- 支持ECB，CBC，CTR，GCM
- 支持DMA进行自动数据传输
- 支持GF（2^128）域下的乘法，支持GMAC

14.2 工作模式

AES有4种工作模式，通过配置MODE[1:0]寄存器设置。

模式1：用存储在AES_KEYRx寄存器中的密钥加密。

模式2：密钥扩展，把初始存储在AES_KEYRx寄存器的加密密钥覆盖成在密钥扩展完成后存储在内部寄存器的密钥计算结果。

模式3：用存储在AES_KEYRx寄存器中的解密密钥（预计算的）解密。

模式4：用存储在AES_KEYRx寄存器中的加密密钥进行密钥扩展和解密。（在CTR模式下不使用）

首先通过配置MODE[1:0]寄存器确定工作模式，MODE寄存器必须在AES使能前（EN=0时）才能够配置。KEY寄存器也应该在AES使能前配置。之后配置数据流处理模式寄存器CHMOD[1:0]，在CBC/CTR/GCM模式下还需要配置IV寄存器。

接着可以使能EN，在模式1/模式3/模式4下，AES模块等待软件往AES_DINR寄存器写入输入数据，写4次写完128bit后AES开始计算。在模式2时，使能EN后就马上进行密钥扩展运算了。

计算完成后标志CCF会置起，如果CCFIE=1，会产生一个中断信号。软件再从AES_DOUTR寄存器中读4次共128bit的结果。

AES还支持DMA模式。通过配置DMAOUTEN=1和DMAINEN=1，AES可以配合DMA连续的处理数据，无需CPU的介入。

错误标志RDERR和WRERR会在一次错误的读写操作时置起，如果ERRIE使能，还会产生相应的错

误中断。AES在产生错误后还会继续正常工作。

通过重置EN寄存器能够在任何时候复位AES模块。

14.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

14.3.1 ECB 模式

默认的工作模式，该模式下无需使用IV寄存器，每个block单独进行加解密计算。加解密流程如图14-1和图14-2所示。

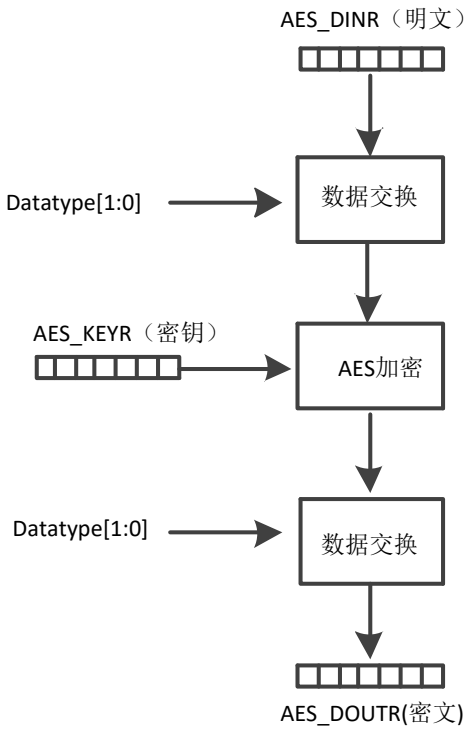


图 14-1 ECB 模式加密流程

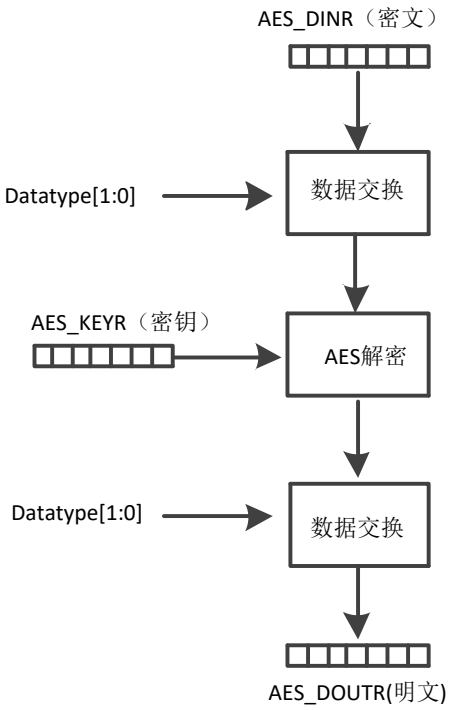


图 14-2 ECB 模式解密流程

14.3.2 CBC 模式

每个block的明文数据与前一block的加密结果异或后作为加密的数据输入。第一个block需要一个初始的IVRx寄存器值。加密时异或操作在加密前而解密时异或操作在加密后。工作流程如图14-3和图14-4所示。

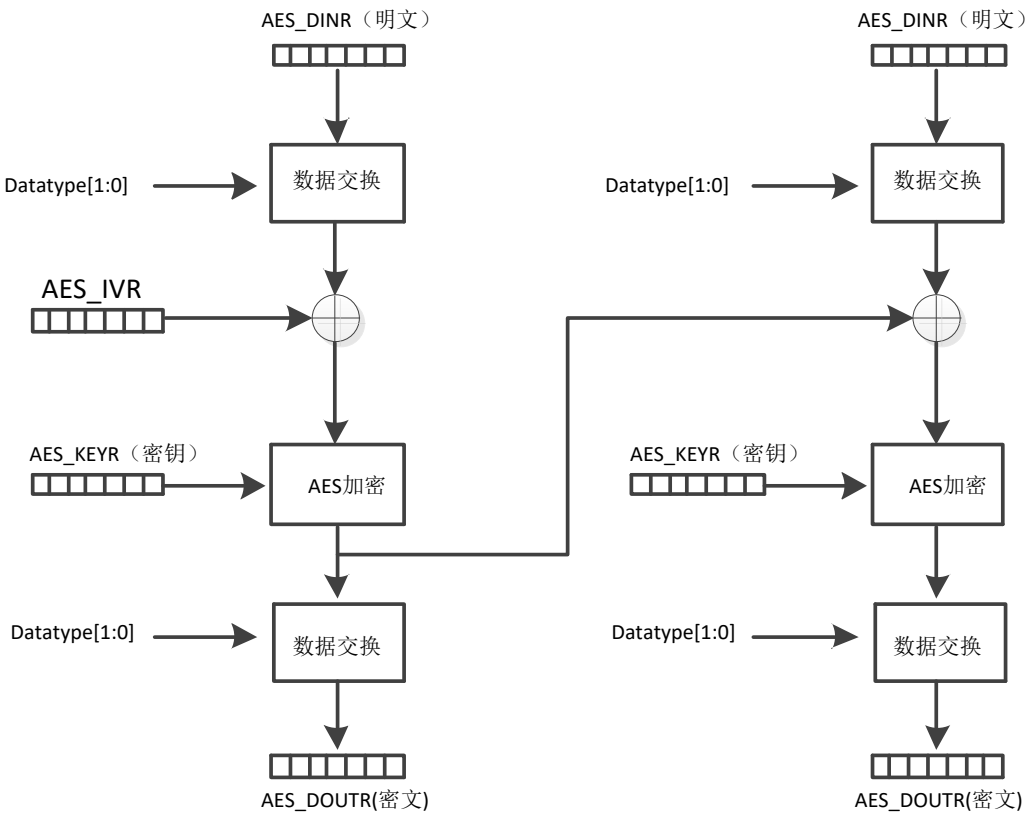


图 14-3 CBC 加密过程

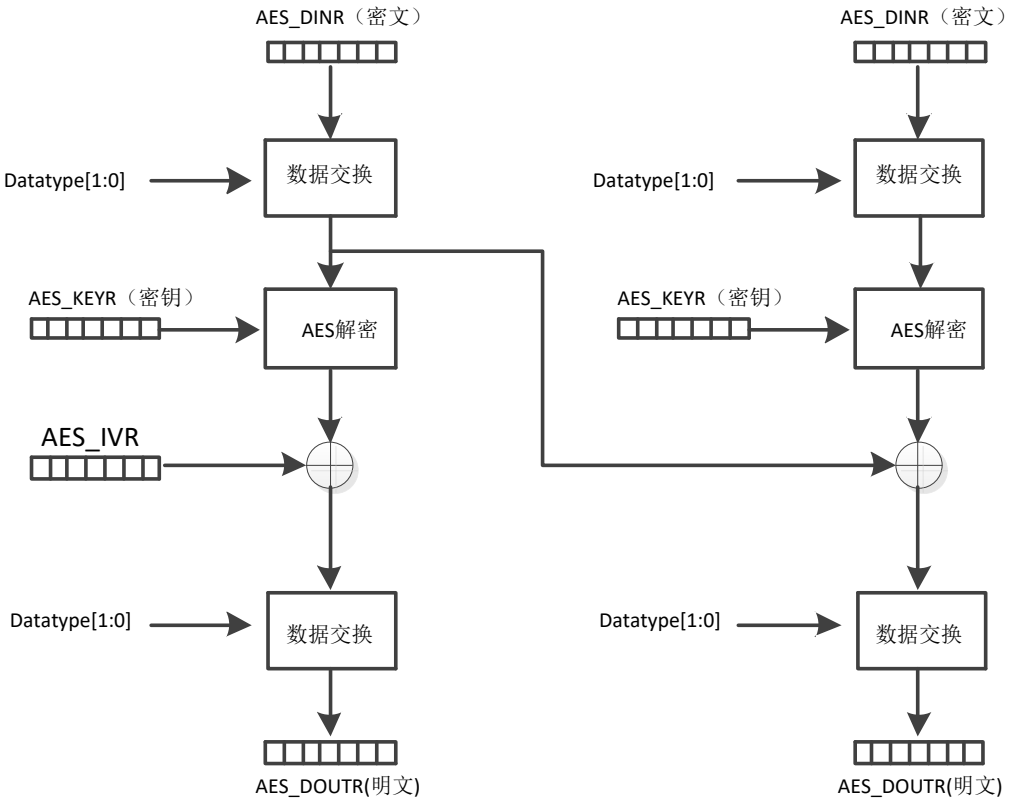


图 14-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

14.3.3 暂停模式

如果一个更高优先级的数据需要处理，当前的数据运算是可以暂停的。暂停的数据处理在加解密运算模式下都能够恢复。仅在CPU参与的模式下可用，DMA模式下不可用。

正确的工作流程为：数据在一个block的结果被读完后暂停。

通过对EN bit写0暂停AES。软件读AES_IVRx寄存器中的值并存储，在恢复运算时该值需要被写入AES_IVRx寄存器。

流程如图14-5所示

图 14-5 暂停模式流程

14.3.4 CTR 模式

该模式下，一个32bit的计数器和一个随机数被用作加解密模块的输入。结果与明文数据进行异或。流程如图14-6和图14-7所示。

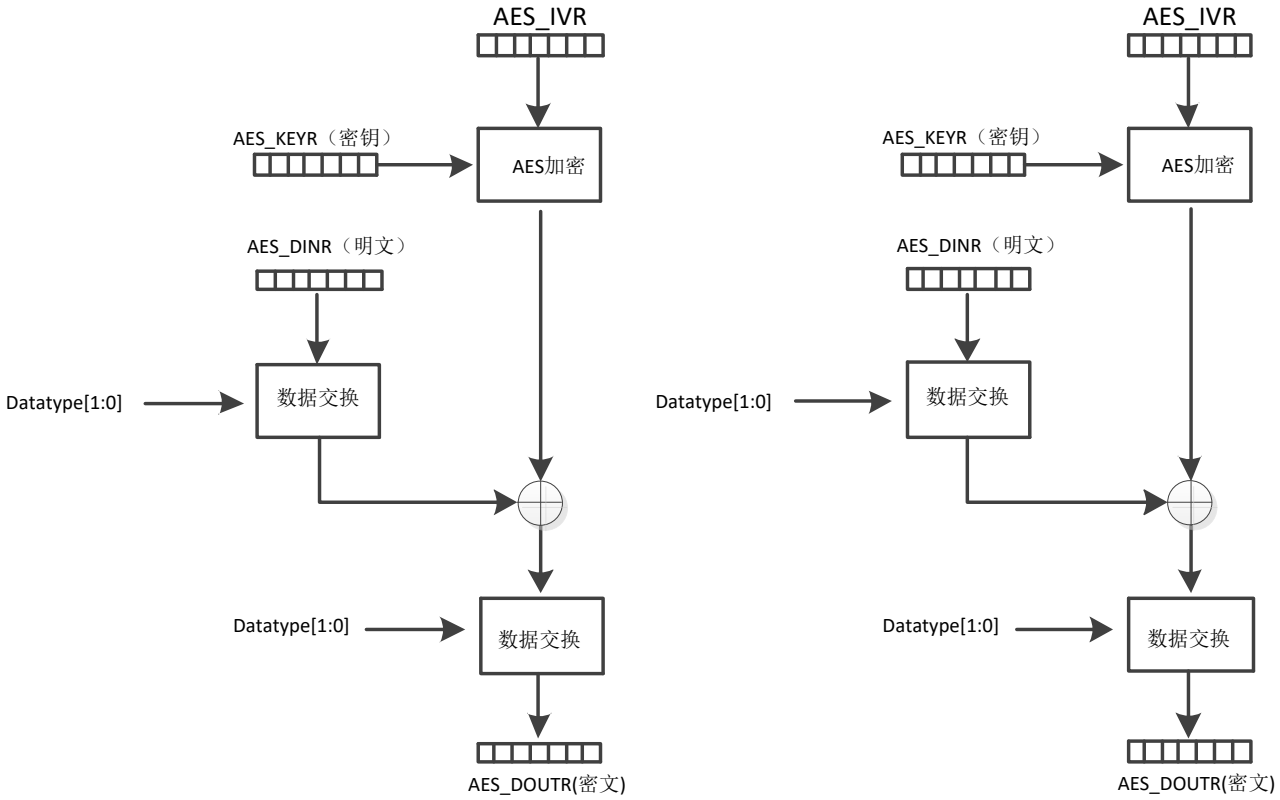


图 14-6 CTR 加密流程

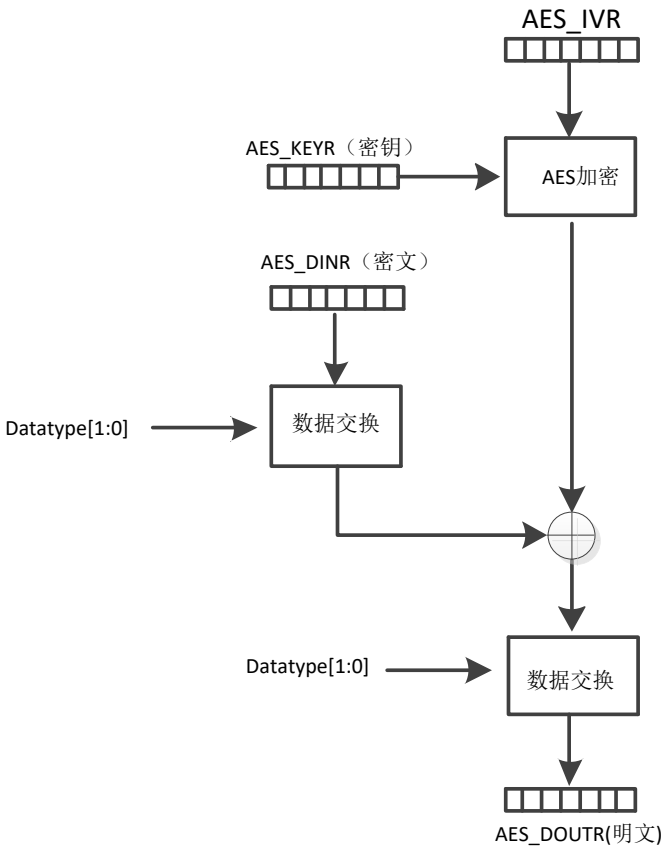


图 14-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图14-8所示

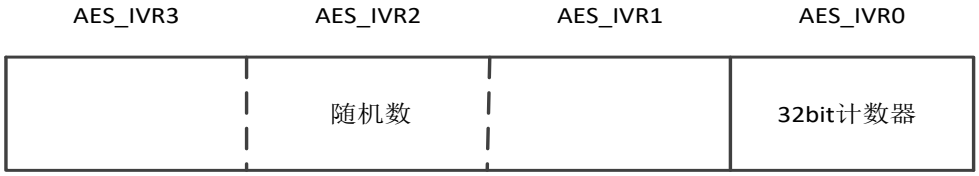


图 14-8 32 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

14.3.5 CTR 模式下的暂停模式

与CBC下暂停模式类似。参考CBC下暂停模式。

14.3.6 GCM 模式

具体可以参考文档The Galois/Counter Mode of Operation (GCM)

GCM的加密按照以下公式定义：

$$
\begin{aligned}
H &= E(K, 0^{128}) \\
Y_0 &= \begin{cases} IV \| 0^{31}1 & \text{if } \mathrm{len}(IV) = 96 \\ \mathrm{GHASH}(H, \{\}, IV) & \text{otherwise.} \end{cases} \\
Y_i &= \mathrm{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \\
C_i &= P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n-1 \\
C_n^* &= P_n^* \oplus \mathrm{MSB}_u(E(K, Y_n)) \\
T &= \mathrm{MSB}_t(\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0))
\end{aligned}
$$

其中GHASH函数的定义为 $\mathrm{GHASH}(H, A, C) = X_{m+n+1}$，其中X的定义为

$$
X_i = \begin{cases} 0 & \text{for } i = 0 \\ (X_{i-1} \oplus A_i) \cdot H & \text{for } i = 1, \ldots, m-1 \\ (X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H & \text{for } i = m \\ (X_{i-1} \oplus C_i) \cdot H & \text{for } i = m+1, \ldots, m+n-1 \\ (X_{m+n-1} \oplus (C_m^* \| 0^{128-u})) \cdot H & \text{for } i = m+n \\ (X_{m+n} \oplus (\mathrm{len}(A) \| \mathrm{len}(C))) \cdot H & \text{for } i = m+n+1. \end{cases}
$$

GCM模式的加解密流程如图14-9，图14-10所示。

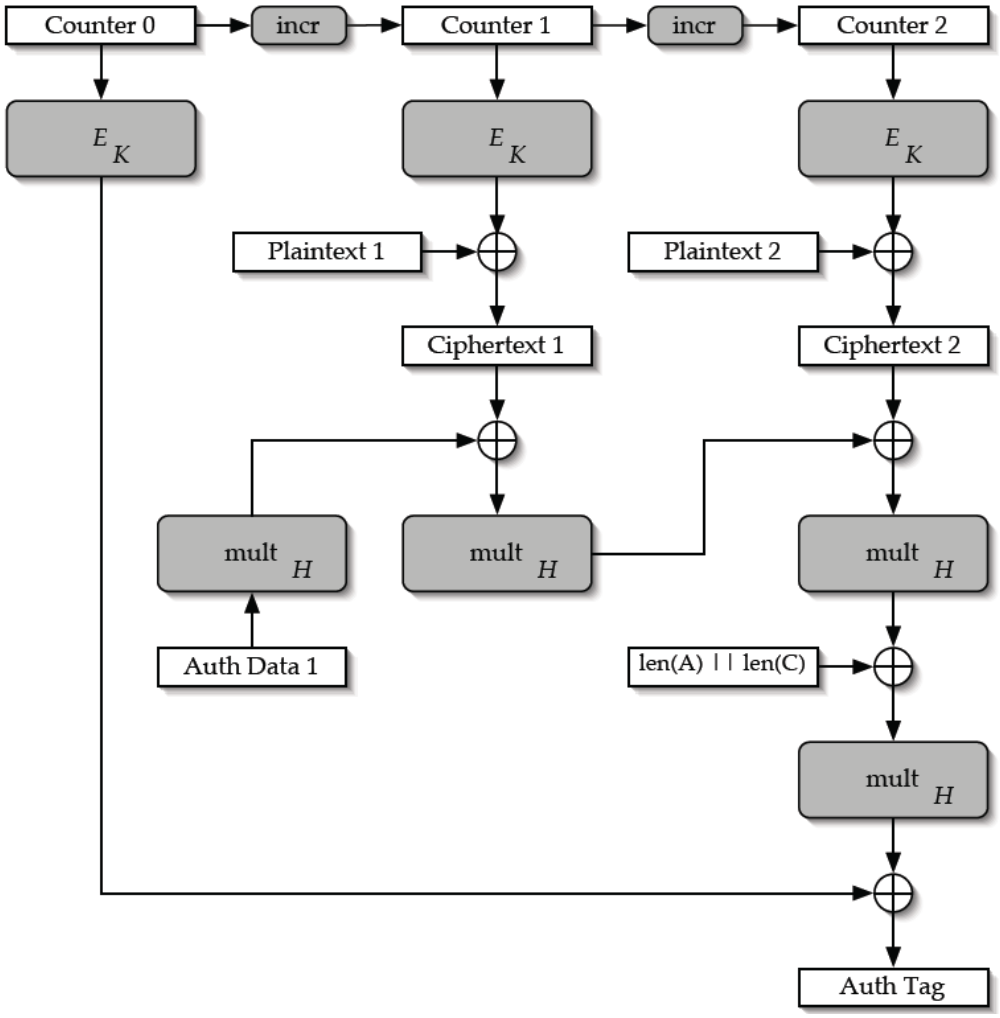


图 14-9 GCM 加密流程

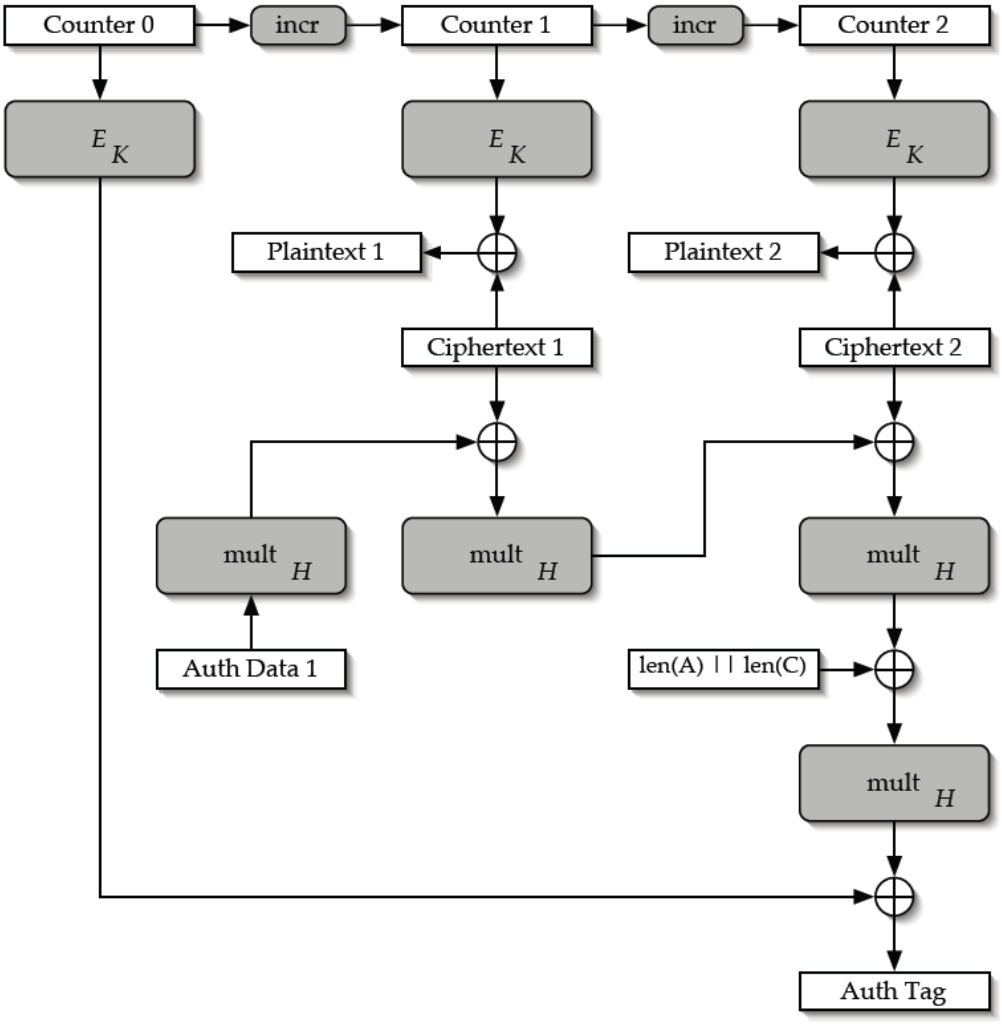


图 14-10 GCM 解密流程

图中Ek表示AES加密模块。multH模块是一个GF（2^128）域上的乘法。Incr表示计数器加一。

GCM模式由软件配合实现，硬件提供一个AES模块和multH模块供软件调度。GCM模式加解密的过程与CTR模式相同。认证过程通过软件调度multH模块实现。

14.3.7 MultH 模块

GF（2^128）上的乘法使用如下算法实现。

Algorithm 1 Multiplication in $GF(2^{128})$. Computes the value of $Z = X \cdot Y$, where X, Y and $Z \in GF(2^{128})$.

$Z \leftarrow 0, V \leftarrow X$
for $i = 0$ to 127 **do**
  if $Y_i = 1$ **then**
    $Z \leftarrow Z \oplus V$
  end if
  if $V_{127} = 0$ **then**
    $V \leftarrow \text{rightshift}(V)$
  else
    $V \leftarrow \text{rightshift}(V) \oplus R$
  end if
end for
return Z

MultH模块的输入输出寄存器复用AES的寄存器。模块框图如图14-11所示。

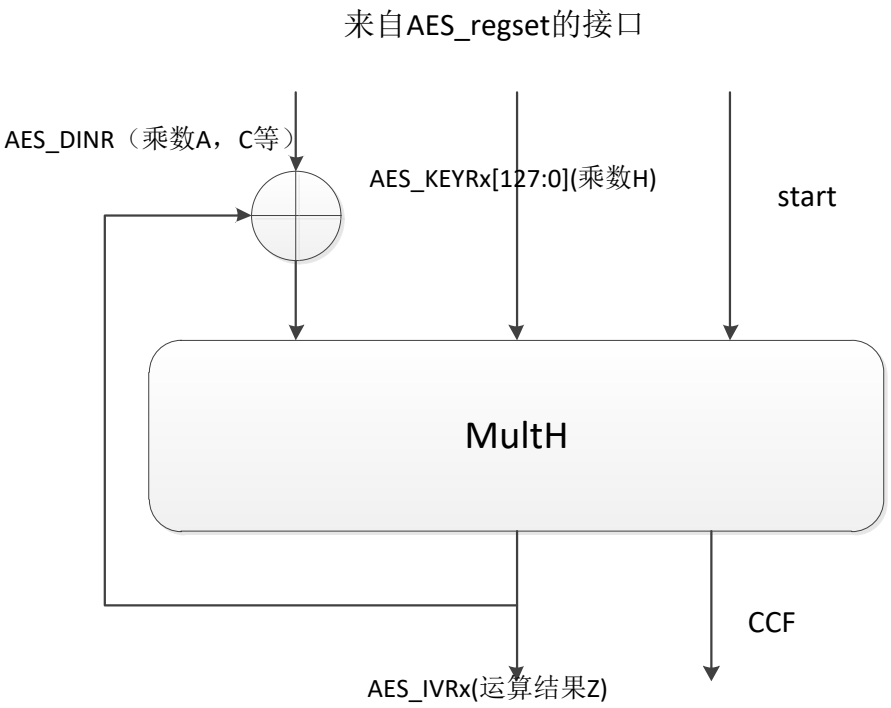


图 14-11 multH 模块框图

multH模块的输入寄存器复用AES的输入寄存器AES_DINR和AES_KEYx的低128bit。输出寄存器复用AES_IVR寄存器。使用时配置CHMOD[1:0]寄存器为MultH模式，接着配置好AES_KEYx和AES_IVR寄存器输入和输出各128bit，使能EN，向AES_DINR输入数据，等待CCF置起即计算完成。

注意：因为复用了寄存器，调用multH会冲掉AES的寄存器。所以使用完multH模块进行计算后如要再进行AES计算，需要重新写相关寄存器。

14.3.8 推荐的 GCM 流程

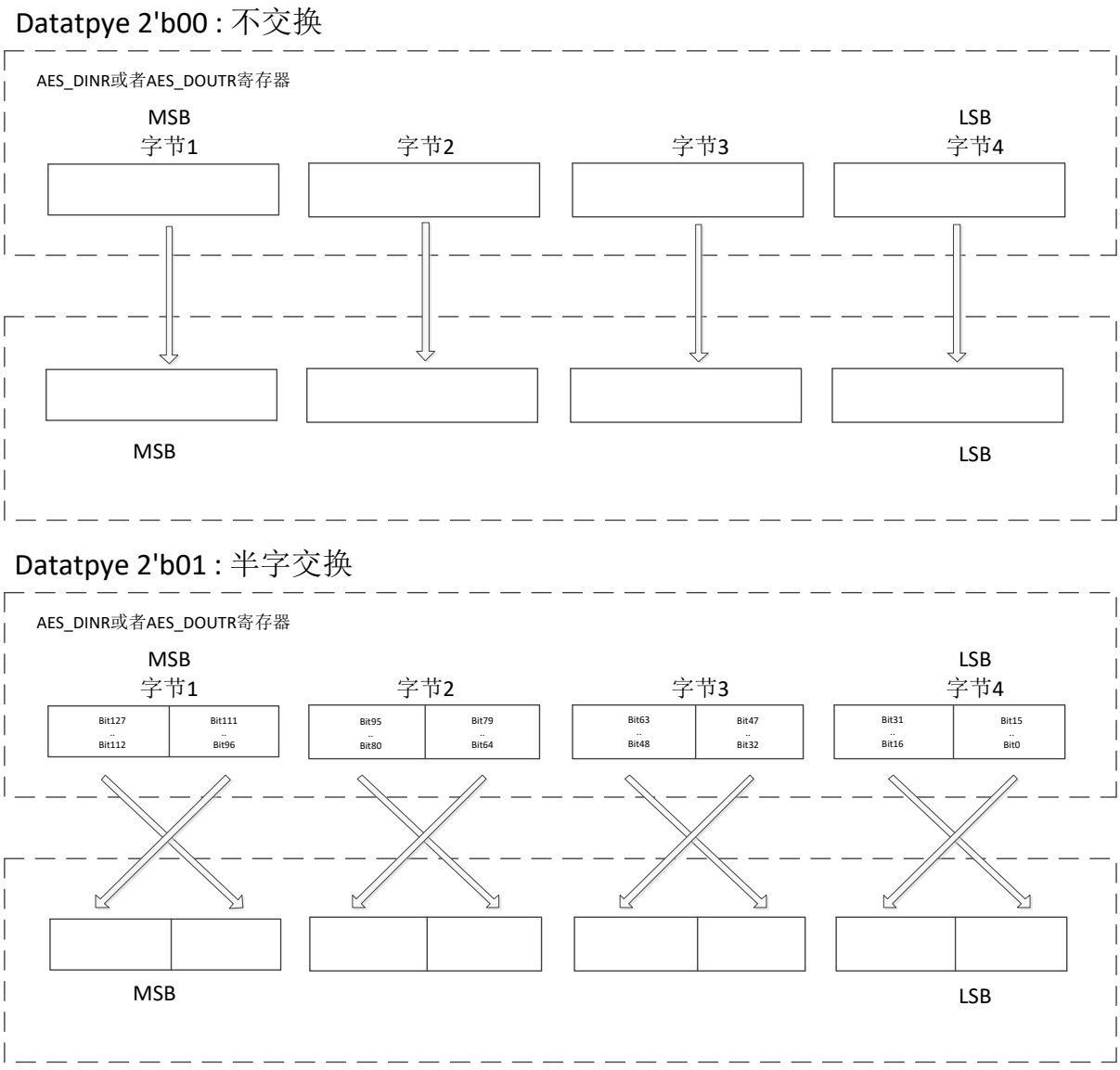
GCM模式的实现需要软硬件配合，本文档提供一种推荐的使用方法。

GCM模式的加解密过程和CTR模式相同。认证过程时仅使用MultH模块而不用AES加解密。

- 调用一次AES模块计算H。并存储。
- 调用一次AES模块计算E（K，Y0），并存储。
- 使用CTR模式开始连续数据的AES加解密操作。IV寄存器初值为Y1
- 使用multH模块连续计算GHASH结果
- 最终GHASH的结果异或上E（K，Y0）即可计算得到tag的值。

14.4 数据类型

AES一次读写32bit数据，每32bit可以根据DATATYPE[1:0]寄存器的设置按照不同的方式交换数据的顺序。如图14-12所示。



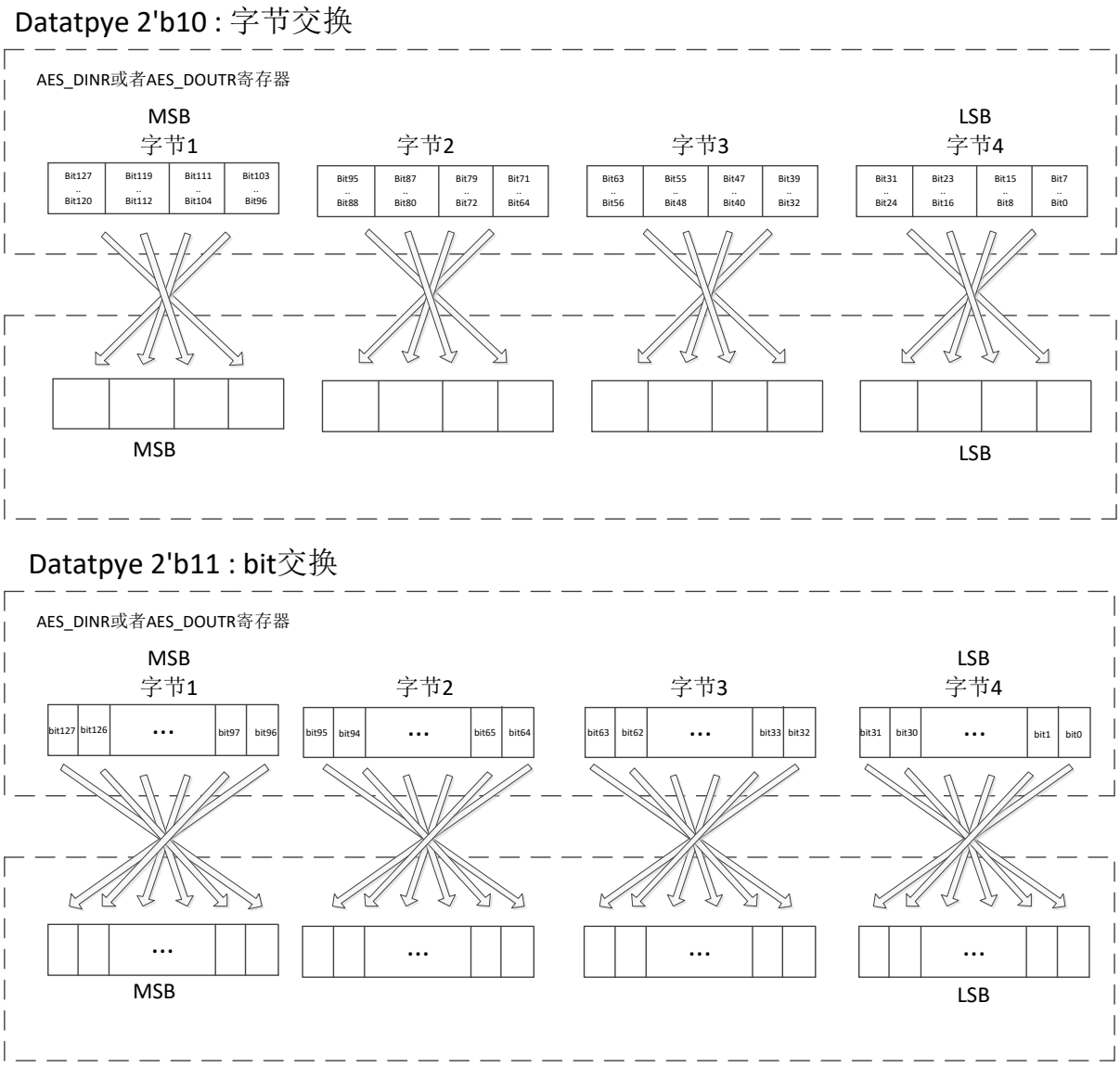


图 14-12 根据数据类型存储数据的示意图

14.5 工作流程

14.5.1 模式 1：加密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=00，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYRx寄存器，CTR和CBC模式下写AES_IVRx寄存器
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起

- 从AES_DOUTR分4次读出加密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行加密

步骤5-7如图所示。

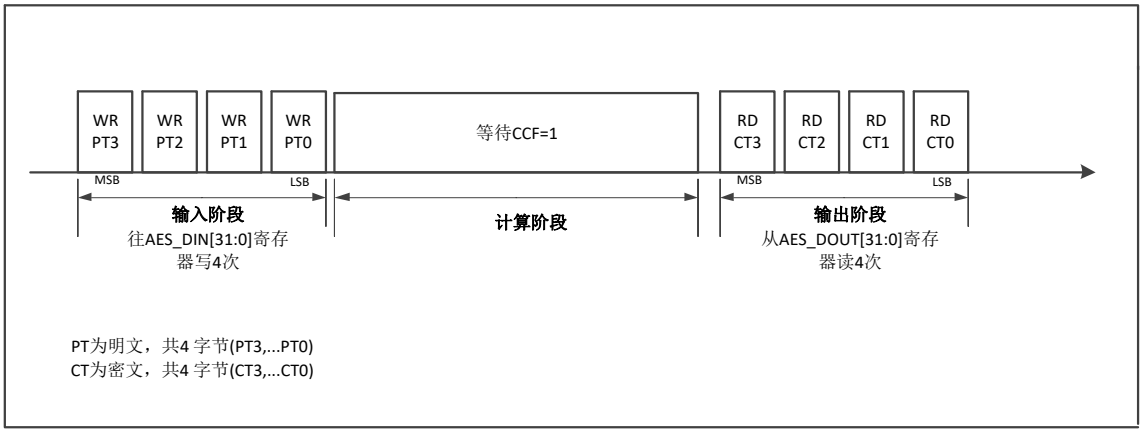


图 14-13 模式 1：加密流程

14.5.2 模式 2：密钥扩展

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=01，CHMOD[1:0]寄存器的值不关心。
- 写AES_KEYRx寄存器。
- 写EN=1，使能AES
- 等待CCF标志置起
- 清除CCF标志，扩展完的key自动写回AES_KEYRx寄存器。如果需要的话可以读取AES_KEYRx寄存器获取结果。想要重新计算扩展密钥，重复步骤3,4,5,6。

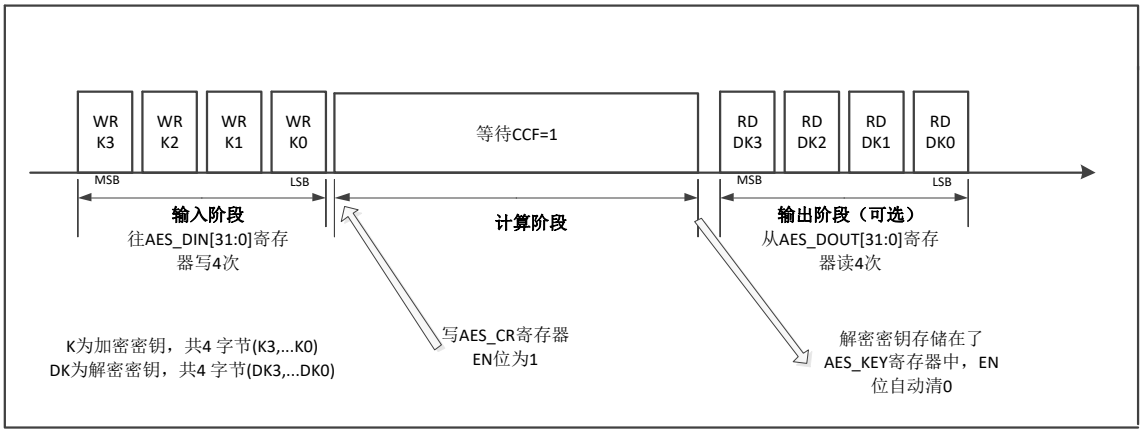


图 14-14 模式 2 示意图

14.5.3 模式 3：解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=10，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYRx寄存器（如果已经通过模式2计算得到了扩展密钥则可跳过这个步骤），CTR和CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

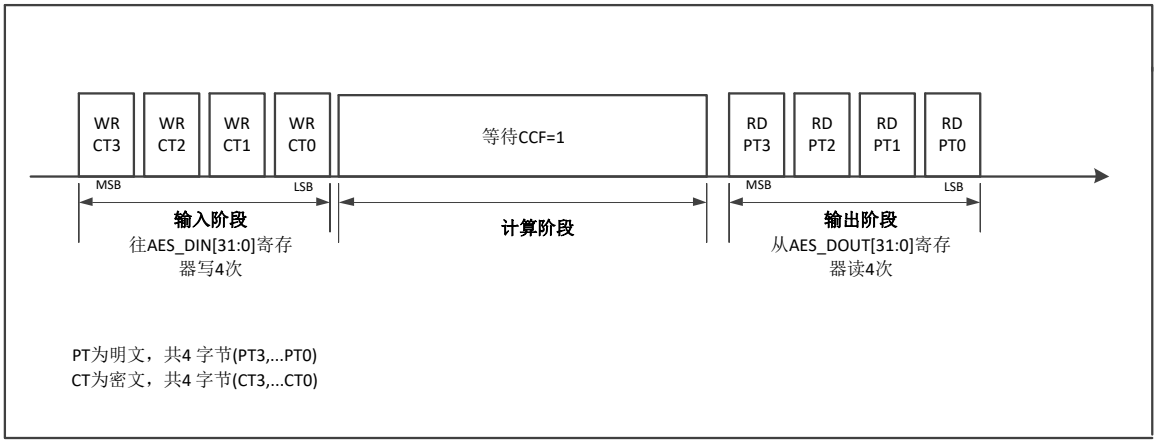


图 14-15 模式 3 示意图

14.5.4 模式 4：密钥扩展+解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=11，设置流数据处理模式寄存器CHMOD[1:0]。该模式在CTR模式下被禁止使用。如果设置mode[1:0]=11，CHMOD[1:0]=10，将强制进入CTR解密模式。
- 写AES_KEYRx寄存器，CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

注意：该模式下AES_KEYRx寄存器内存储的一直是加密密钥，扩展密钥每次都会在内部被重新计算而不会被存储到AES_KEYRx寄存器中。

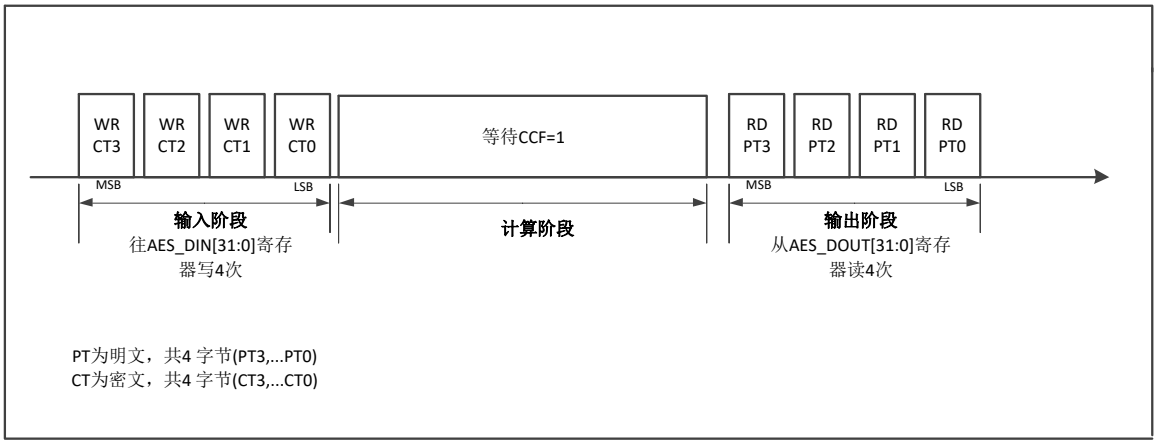


图 14-16 模式 4 示意图

14.5.5 使用 MultH 模块

- 复位EN 重置AES模块。
- 设置流数据处理模式寄存器CHMOD[1:0]=11。该模式下mode[1:0]寄存器的值不能够是01配置在模式2：密钥扩展下。同时配置mode[1:0]=01和CHMOD[1:0]=11会由于mode寄存器优先值更高而进行密钥扩展操作。
- 写AES_KEYRx寄存器，高128bit为上一次计算输出值，若为第一轮计算，则初始值为0x00000000。低128bit为H的值。
- 写EN=1，使能multH模块。
- 写AES_DINR 寄存器4次。MultH模块会把上一次的计算结果异或上AES_DINR寄存器输入的值做为multH模块的一个乘数。所以把上一轮的计算结果赋为0x00000000，即实现了直接把AES_DINR寄存器输入的值做为multH模块的一个乘数的功能。
- 等待CCF标志置起
- 从AES_IVR寄存器中读出计算结果。
- 对于同一个H，重复步骤5,6进行连续计算。即可实现了一个GMAC的功能。

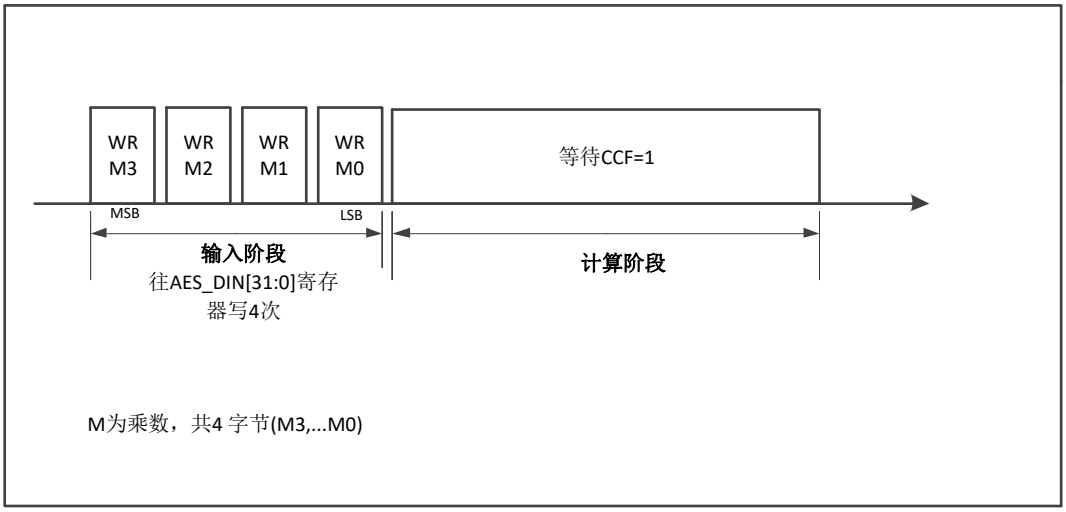


图 14-17 multH 模块使用流程示意图

14.6 DMA 接口

- 一个输入的请求通道：当DMAINEN为1时，每当AES在需要输入数据写入AES_DINR寄存器的时候发起一个DMA的请求。
- 一个输入的请求通道：当DMAOUTEN为1时，每当AES在需要从AES_DOUTR寄存器输出数据的时候发起一个DMA的请求。

每个阶段产生4次请求，在AES模块被关闭前对DMA的请求会一直产生。AES计算完128比特后就自动取新数据进行下次计算。

注意：DMA模式下DMAOUTEN=1时，CCF标志可能为高。

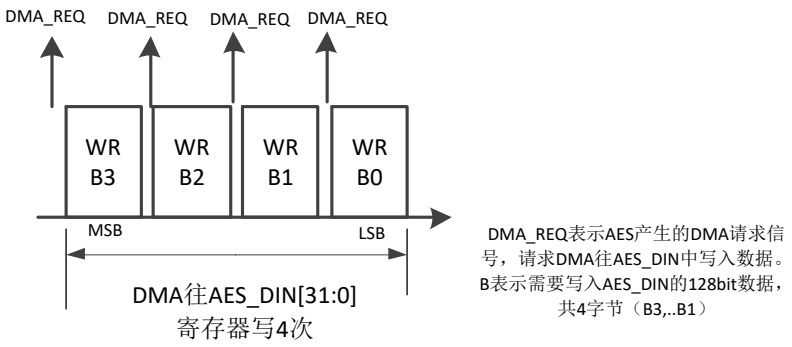


图 14-18 输入时 DMA 请求和数据传输示意图

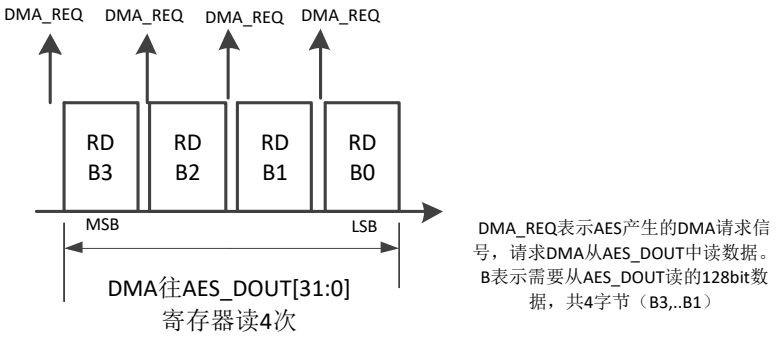


图 14-19 输出时 DMA 请求和数据传输示意图

14.6.1 MultH 模块与 DMA 间接口

MultH计算也可以通过DMA计算。当DMAINEN为1及CHMOD[1:0]=11时，每当AES在需要输入数据写入AES_DINR寄存器的时候发起一个DMA的请求。该模式下配置DMAOUTEN=1无效AES不会产生DMA请求。

14.7 错误标志

在计算和输入阶段发生一个读操作，置起RDERR。

在计算和输出阶段发生一个写操作，置起WRERR。

产生错误后AES模块不会被硬件自动停止，会像正常一样继续运算。

14.8 寄存器

offset 地址	名称	符号
AES(模块起始地址:0x4001B800)		
0x00000000	AES 控制寄存器 （AES Control Register）	AES_CR
0x00000004	AES 中断使能寄存器 （AES Interrupt Enable Register）	AES_IER
0x00000008	AES 中断标志寄存器 （AES Interrupt Status Register）	AES_ISR
0x0000000C	AES 数据输入寄存器 （AES Data Input Register）	AES_DIR
0x00000010	AES 数据输出寄存器 （AES Data Output Register）	AES_DOR
0x00000014	AES 秘钥寄存器 0 （AES Key Register 0）	AES_KEY0
0x00000018	AES 秘钥寄存器 1 （AES Key Register 1）	AES_KEY1
0x0000001C	AES 秘钥寄存器 2 （AES Key Register 2）	AES_KEY2
0x00000020	AES 秘钥寄存器 3 （AES Key Register 3）	AES_KEY3
0x00000024	AES 秘钥寄存器 4 （AES Key Register 4）	AES_KEY4
0x00000028	AES 秘钥寄存器 5 （AES Key Register 5）	AES_KEY5
0x0000002C	AES 秘钥寄存器 6 （AES Key Register 6）	AES_KEY6
0x00000030	AES 秘钥寄存器 7 （AES Key Register 7）	AES_KEY7
0x00000034	AES 初始向量寄存器 0 （AES Initial Vector Register 0）	AES_IVR0
0x00000038	AES 初始向量寄存器 1 （AES Initial Vector Register 1）	AES_IVR1
0x0000003C	AES 初始向量寄存器 2 （AES Initial Vector Register 2）	AES_IVR2
0x00000040	AES 初始向量寄存器 3 （AES Initial Vector Register 3）	AES_IVR3

14.8.1 AES 控制寄存器（AES_CR）

名称	AES_CR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-	KEYLEN		DMAOEN	DMAIEN	-	-	-
位权限	U-0	R/W-00		R/W-0	R/W-0	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	CHMOD		MODE		DATATYP		EN
位权限	U-0	R/W-00		R/W-00		R/W-00		R/W-0

位号	助记符	功能描述
31:15	-	RFU：未实现，读为 0
14:13	KEYLEN	AES 加密密钥长度，AESEN=1 时不可修改。(Key Length) 00：128bit 01：192bit 10：256bit 11：保留
12	DMAOEN	DMA 数据自动读出使能(DMA output enable) 0：不开启 1：开启 该位置位后在模式 1，模式 3 和模式 4 下 AES 模块会自动产生 AES->RAM 的传输请求。模式 2 下不会产生。
11	DMAIEN	开启 DMA 数据自动写入使能(DMA input enable) 0：不开启 1：开启 该位设置为 1 后在模式 1，模式 3 和模式 4 以及 MultH 模式下 AES 模块会自动产生 RAM->AES 的传输请求。模式 2 下不会产生。
10:7	-	RFU：未实现，读为 0
6:5	CHMOD	AES 数据流处理模式，AESEN=1 时不可修改。(Cipher Mode) 00：ECB 01：CBC 10：CTR 11：使用 MultH 模块
4:3	MODE	AES 工作模式，AESEN=1 时不可修改。(operation MODE) 00：模式 1：加密 01：模式 2：密钥扩展 10：模式 3：解密 11：模式 4：密钥扩展+解密 CTR 模式下配置成模式 4 将自动进入 CTR 的解密模式。即在 CHMOD=2’b10 时配置 MODE=2’b11，AES 将按照 MODE=2’b10 的情形执行。
2:1	DATATYP	选择数据类型，AESEN=1 时不可修改。具体交换规则可参考 AES 数据类型章节。(Data type) 00：32bit 数据不交换 01：16bit 数据半字交换 10：8bit 数据字节交换 11：1bit 数据比特交换
0	EN	AES 使能 (AES enable) 0：不使能 1：使能 在任何时候清除 AESEN 位都能够复位 AES 模块 在模式 2 下该位会在一次计算完成后硬件自动清 0

14.8.2 AES 中断使能寄存器（AES_IER）

名称	AES_IER							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR_IE	RDERR_IE	CCF_IE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	WRERR_IE	写错误中断使能，1 有效。(Write Error interrupt enable)
1	RDERR_IE	读错误中断使能，1 有效。(Read Error interrupt enable)
0	CCF_IE	AES 计算完成中断使能，1 有效。(Cipher Complete Interrupt enable)

14.8.3 AES 中断标志寄存器（AES_ISR）

名称	AES_ISR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR	RDERR	CCF
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	WRERR	写错误标志：在计算或输出阶段发生写操作时置位，软件写 1 清零 (Write Error Flag,write 1 to clear)
1	RDERR	读错误标志：在计算或输入阶段发生读操作时置位，软件写 1 清

位号	助记符	功能描述
		零 (Read Error Flag,write 1 to clear)
0	CCF	AES 计算完成标志，软件写 1 清零 (Cipher Complete Flag,write 1 to clear) 1：计算完成 0：计算没有完成

14.8.4 AES 数据输入寄存器（AES_DIR）

名称	AES_DIR							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DIN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	DIN	数据输入寄存器，当 AES 需要输入加解密数据时，应该往该寄存器连续写 4 次。(AES Data Input) 模式 1（加密）：把明文从 MSB 到 LSB 分 4 次写入。 模式 2（密钥扩展）：无需使用数据输入寄存器 模式 3 和模式 4（解密）：把密文从 MSB 到 LSB 分 4 次写入。 MultH 模式：把乘数 A 或 C 从 MSB 到 LSB 分 4 次写入。

14.8.5 AES 数据输出寄存器（AES_DOR）

名称	AES_DOR							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DOUT[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DOUT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DOUT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DOUT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	DOUT	数据输出寄存器，当 AES 计算完成后，可以分四次读出加解密的结果。(AES Data Output,read only) 模式 1（加密）：把密文从 MSB 到 LSB 分 4 次读出。 模式 2（密钥扩展）：无需使用数据输输出寄存器 模式 3 和模式 4（解密）：把明文从 MSB 到 LSB 分 4 次输出。 MultH 模式：运算结果存储在 IVR 寄存器中，无需读取 AES_DOUTR 寄存器。

14.8.6 AES 秘钥寄存器 x（AES_KEYx）

名称	AES_KEYx(x=0,1,2,3,4,5,6,7)							
Offset	0x00000014 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEYx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEYx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEYx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEYx[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	KEYx	AES 运算秘钥，最长 256bit，AESKEY0 存放秘钥最低 32bit，AESLKEY7 存放秘钥最高 32bit。(AES Key) AESKEY0~3 在 MultH 模式下存放 H[127:0]

14.8.7 AES 初始向量寄存器 x（AES_IVRx）

名称	AES_IVRx(x=0,1,2,3)							
Offset	0x00000034 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IVRx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IVRx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IVRx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IVRx[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	IVRx	AES 运算 128bit 初始向量，在 MultH 模式下保存运算结果。 (AES Initial Vector Registers)

15 随机数发生器（TRNG）

15.1 概述

FM33LC0XX使用2个Galois真随机噪声源作为真随机数种子，配合简单在线检测（32位全0全1检测）、LFSR后处理、伪随机LFSR共同组成芯片的随机数发生器。

TRNG的启动测试和完整的在线测试功能需要固件实现。

Galois噪声源的采样和LFSR建议使用4MHz时钟。两次取32bit随机数之间的间隔不得小于32个时钟周期。

真随机数发生器通过了FIPS PUB140-2测试，成功率99.9%。

15.2 功能描述

15.2.1 随机数产生

下图为真随机数发生器结构框图。

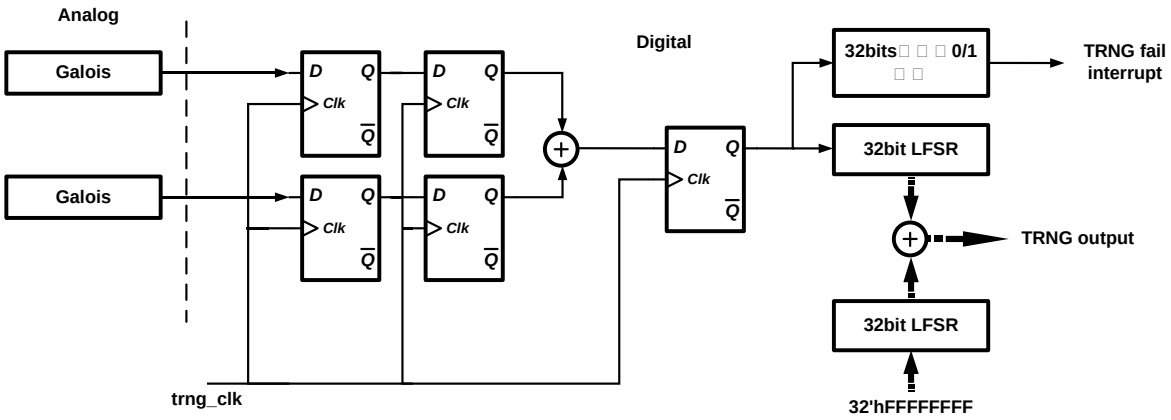


图 15-1 真随机数模块框图

真随机噪声源为2个Galois环振，Galois环振输出在数字电路内部异或并使用系统时钟采样，然后进行LFSR后处理。LFSR后处理之前经过随机数在线检测，如果发现连续32bit全0或全1的情况，则产生TRNG失效报警中断。同时为了避免小概率的真随机数性能不良情况，另外使用一组LFSR以32’hFFFFFFFF为初始值，与后处理LFSR同步运算，并以两组LFSR按位异或后的结果作为最终的32bit随机数输出。

15.2.2 工作时钟

随机数发生器的工作时钟采用RCHF的分频时钟，独立于APBCLK。为了保证随机数质量，一般建议应用使用4M时钟作为随机数工作时钟，并根据4M目标频率配置CMU模块中随机数工作时钟分频寄存器（OPCCON2.RNGPRSC）。工作时钟示意图如下：

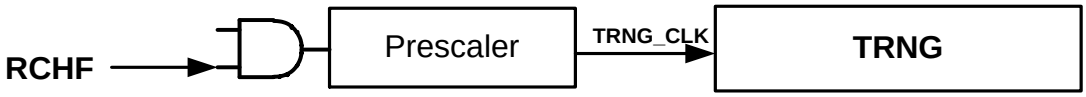


图 15-2 真随机数模块工作时钟

15.2.3 随机数读取

当随机数模块被使能后，真随机噪声源和LFSR后处理模块同时开始工作。软件通过读取RNGOUT寄存器，每次读出32bit随机数。由于LFSR循环移位周期是32cycle，为保证随机数质量，应用应保

证两次读取RNGOUT之间的间隔大于32个TRNG_CLK周期。

举例来说，假设TRNG_CLK为4MHz，则两次读取RNGOUT的间隔不应小于8us。

15.2.4 CRC 运算

用作随机数后处理的LFSR也可用于进行CRC计算。

在进行CRC运算时，两组32bit LFSR分别作为输入数据寄存器和CRC运算寄存器，一次可以运算32bit数据的CRC结果。CRC运算前CPU需查询当前LFSR是否被占用，如LFSR空闲，方可以使用CRC功能。

CPU一旦启动CRC运算，LFSR自动置为复位值，随后进行32bit运算，运算结束后清除CRC启动寄存器，不产生中断；软件启动CRC后应连续查询启动寄存器状态，直到运算结束后再读取结果。

CRC多项式：

CRC32=X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+X0

软件操作流程：

- 查询LFSR_BUSY，确认LFSR不在运行中
- 将待运算数据写入CRCDATA0~3
- 置位CRC_EN
- 查询并等待CRC_EN被清零
- 从LFSROUT0~3读出运算结果

15.3 寄存器

地址	名称	符号
RNGCTL(模块起始地址:0x4001A868)		
0x00000000	随机数控制寄存器 （Random Number Generator Control Register）	RNGCTL_CR

15.3.1 随机数控制寄存器（RNGCTL_CR）

名称	RNGCTL_CR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	EN	RNG 使能寄存器，软件写 1 启动 (RNG enable) 1：启动 RNG 0：关闭 RNG

15.4 寄存器

offset 地址	名称	符号
RNG(模块起始地址：0x4001BC00)		
0x00000004	随机数/CRC 结果输出寄存器 （Random Number Generator Data Output Register）	RNG_DOR
0x00000010	RNG 标志寄存器 （Random Number Generator Status Register）	RNG_SR
0x00000014	CRC 控制寄存器 （CRC Control Register）	RNG_CRCCR
0x00000018	CRC 输入数据寄存器 （CRC Data input Register）	RNG_CRCDIR
0x0000001C	CRC 标志寄存器 （CRC Status Register）	RNG_CRCSR

15.4.1 随机数/CRC 结果输出寄存器（RNG_DOR）

名称	RNG_DOR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RNGOUT[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RNGOUT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RNGOUT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RNGOUT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	RNGOUT	随机数生成结果或 CRC 运算结果寄存器 (RNG output,read only)

15.4.2 RNG 标志寄存器（RNG_SR）

名称	RNG_SR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						LFSREN	RNF
位权限	U-0						R-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	LFSREN	LFSR 状态标志，只读 (LFSR Flag,read only) 1：LFSR 在运行中，不可进行 CRC 验证 0：LFSR 不在运行中，可进行 CRC 验证 注：本寄存器不会引起模块中断，仅供查询
0	RNF	随机数生成失败标志 (Random Number Fail Flag) 1：随机数未能通过质量检测 0：随机数通过质量检测

15.4.3 CRC 控制寄存器（RNG_CRCCR）

名称	RNG_CRCCR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CRCEN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	CRCEN	CRC 使能控制寄存器，软件写 1 启动 CRC，运算完成后硬件自动清零 (CRC enable) 1：CRC 使能 0：CRC 关闭

15.4.4 CRC 输入数据寄存器（RNG_CRCDIR）

名称	RNG_CRCDIR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRCIN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CRCIN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CRCIN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CRCIN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	CRCIN	CRC 运算数据输入寄存器 (CRC data input)

15.4.5 CRC 标志寄存器（RNG_CRCSR）

名称	RNG_CRCSR

Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CRCDONE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	CRCDONE	CRC 计算完成标志，软件写 0 清零 (CRC calculation done Flag,write 0 to clear) 1：CRC 计算完成 0：CRC 计算未完成

16 运算放大器（OPA）

16.1 概述

- 两个独立运算放大器。
- 输入电压范围 rail-to-rail。
- GBW 为 1.6MHz。
- 典型功耗小于 123uA。
- 低功耗模式小于 1.4uA，低功耗模式仅用于比较器模式。
- 最大驱动电流 500uA。
- 支持 standalone 模式、buffer 模式、PGA 模式（x2, x4, x8, x16，仅 OPA1）、比较器模式
- 典型输入 offset +/-3mv，支持用户校准
- OPA 输出可连接 ADC，用于输入信号预放大和阻抗匹配

16.2 结构框图

下图是单个OPA1的结构框图：

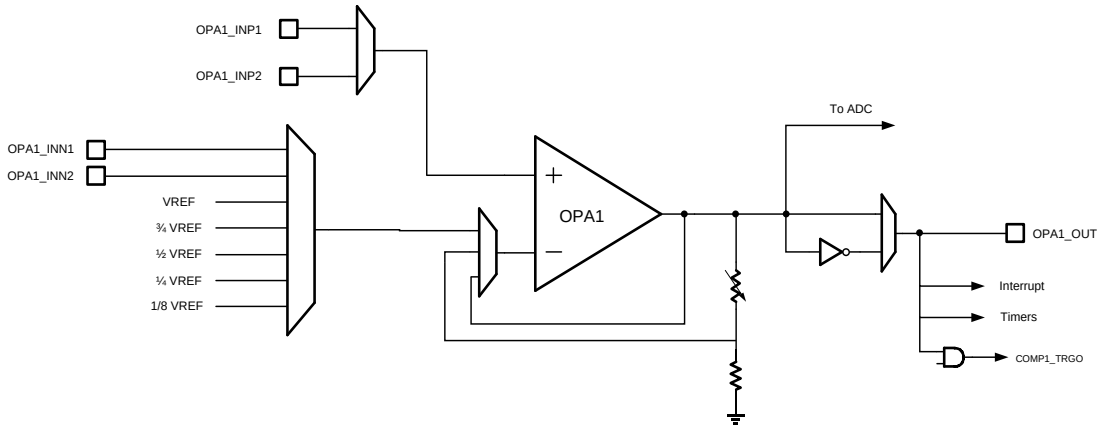


图 16-1OPA1 电路框图

下图为OPA2的结构框图：

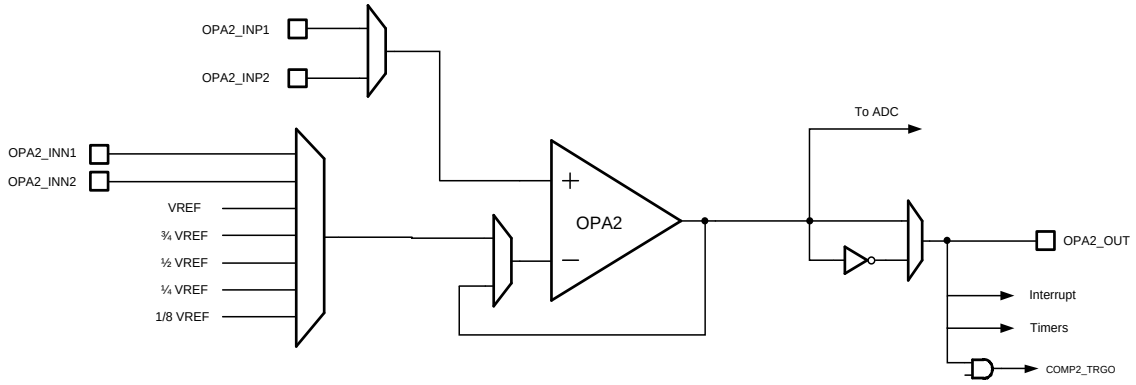


图 16-2OPA2 电路框图

根据寄存器配置选择AMUX不同通路，可以实现不同的开环和闭环应用，比如比较器（输入可配置）、buffer、PGA（内置反馈电阻）、独立运放。输出可以从IO引出，或接给ADC，也可以产生数字信号或中断输出。

下图是OPA作为ADC前端放大应用时的连接关系示意图：

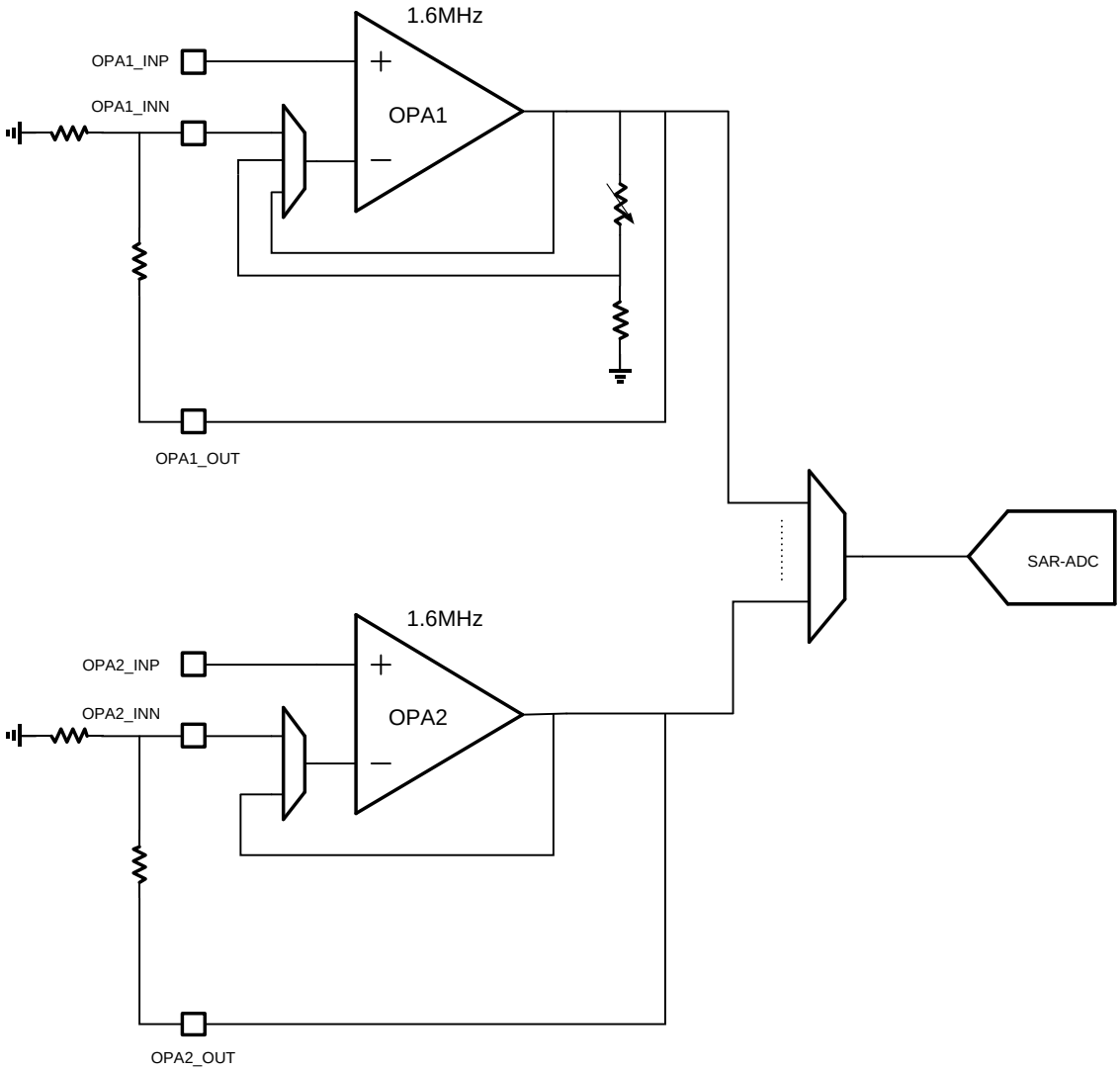


图 16-3OPA 用作 ADC 前端放大

16.3 引脚定义

OPA 模块有多个模拟输入输出端口，被复用到多个 GPIO 上。

引脚	OPAx	符号	功能
PB10	OPA1	OPA1_INN1	运放负端输入
PB11		OPA1_INP1	运放正端输入
PA6		OPA1_INN2	运放负端输入
PA7		OPA1_INP2	运放正端输入
PC4		OPA1_OUT	运放输出
PB13	OPA2	OPA2_INN1	运放负端输入
PB14		OPA2_INP1	运放正端输入
PC0		OPA2_INN2	运放负端输入
PC1		OPA2_INP2	运放正端输入
PC5		OPA2_OUT	运放输出

表 16-1OPA 引脚列表

16.4 功能描述

OPA支持standalone模式、buffer模式和PGA模式（x4, x8, x16, x32）

16.4.1 Standalone 模式

此模式下OPA的输入和输出都直接连接到芯片GPIO的电阻通道上，通过在片外连接反馈电阻，用户可以灵活调节运放的负反馈增益，如下图所示：

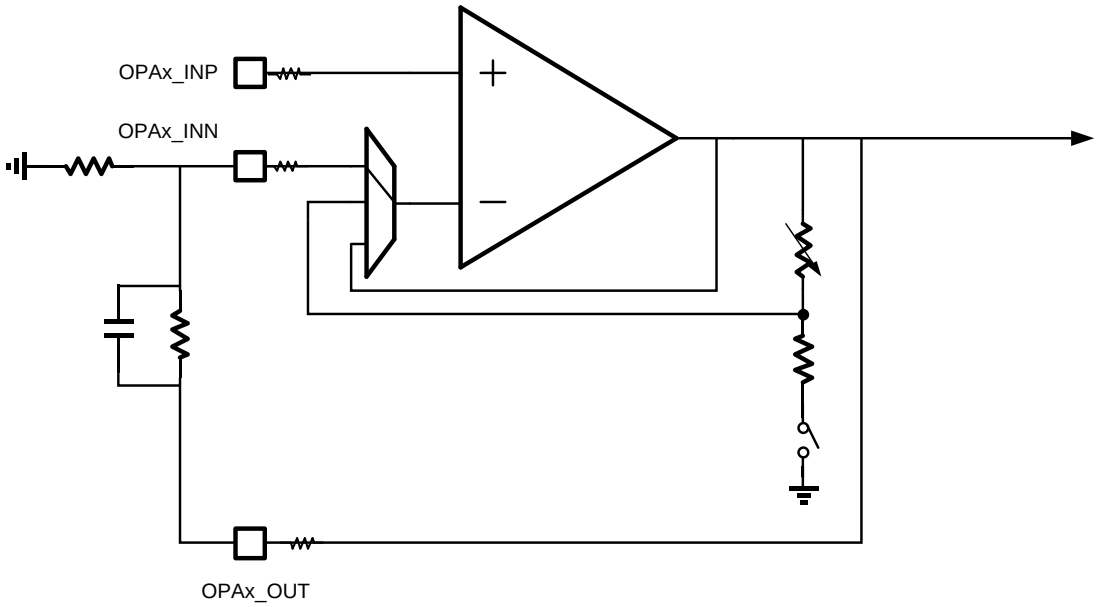


图 16-4OPA 独立模式

注意PAD电阻通道内部的ESD电阻。

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入IO
- 配置OPAxCR.OPAxMOD为00，即standalone模式
- 使能OPAx

16.4.2 比较器模式

在standalone模式下，如果断开片外反馈电阻，则可以提供比较器功能，比较器正端来自GPIO模拟通道输入，负端来自GPIO或者VREF1p22及其buffer分压。注意只有比较器模式下才会产生比较器中断和触发信号输出。

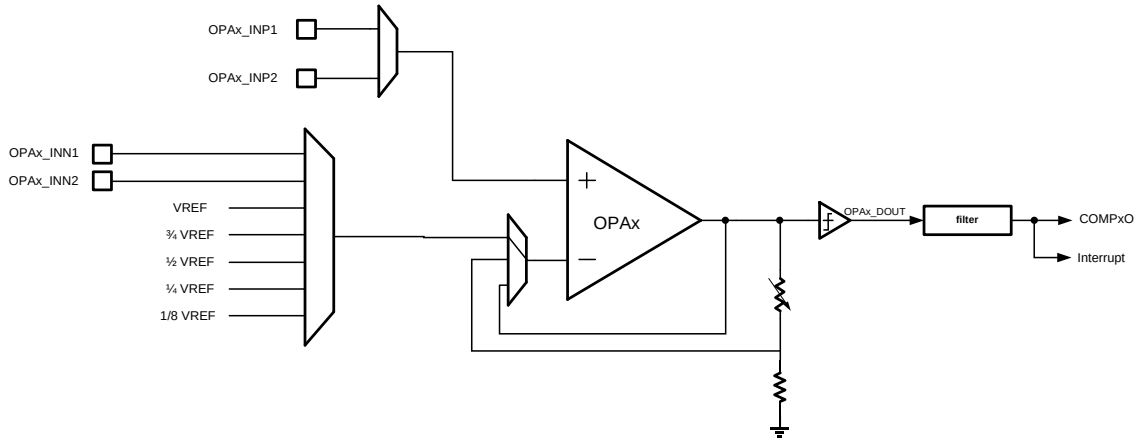


图 16-5OPA 比较器模式

VREF分压结构如下图所示，其中VREF Buffer输出可以调校，避免VREF本身离散性的影响。

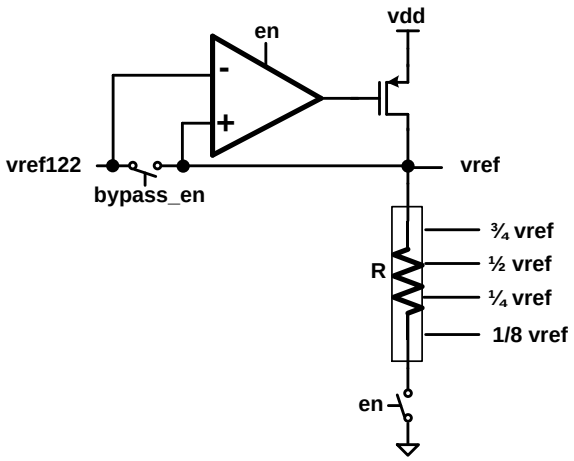


图 16-6OPA 比较器基准电压产生

OPA1和OPA2共享同一个VREF BUFFER，通过OPA1CR.BUENB使能，OPA1.BUFBYP可以配置是否bypass这个BUFFER。

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入信号源
- 使能VREF和VREF_BUFFER
- 等待VREF buffer输出建立
- 配置OPAxCR.OPAxMOD为01，即比较器模式
- 使能OPAx

比较器输出数字滤波

OPA比较器模式下，其输出为数字信号，支持数字滤波功能。

数字滤波可以通过CMPxDF寄存器使能或禁止。在使能数字滤波后，数字电路使用APBCLK连续采样比较器输出，只有当3拍采样结果一致时，才认为是合法电平。下图为数字滤波示意图。

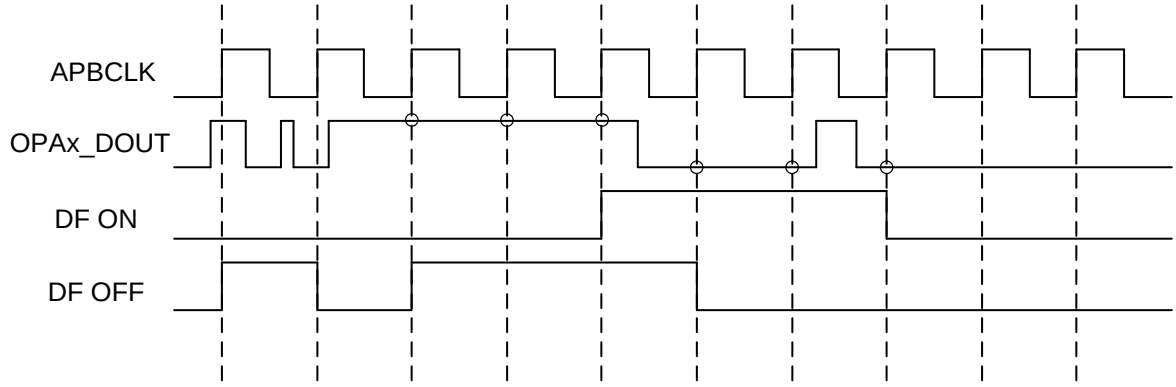


图 16-7OPA 比较器输出数字滤波

16.4.2.1 比较器的功耗模式

比较器支持三种功耗模式：

模式	功耗	输出延迟	使用条件
快速模式	80uA	<10us	BGQS 使能，OPAxLPM=0
低功耗模式	<10uA	<30us	BGQS 使能，OPAxLPM=1
超低功耗模式	1.5uA	200us	BGQS 关闭，OPAxLPM=1

表 16-2OPA 比较器模式

比较器的快速模式和低功耗模式需要使能BGQS，因此能在Active/LP Active/LP Run/Sleep模式下工作。而DeepSleep模式下由于关闭了BGQS，只能使用超低功耗模式。

16.4.3 Buffer 模式

buffer模式下OPA可用于为ADC输入提供阻抗调整，当输入信号频率与OPA的GBW相适应时，配置为buffer模式的OPA可以增强ADC输入信号的驱动能力。

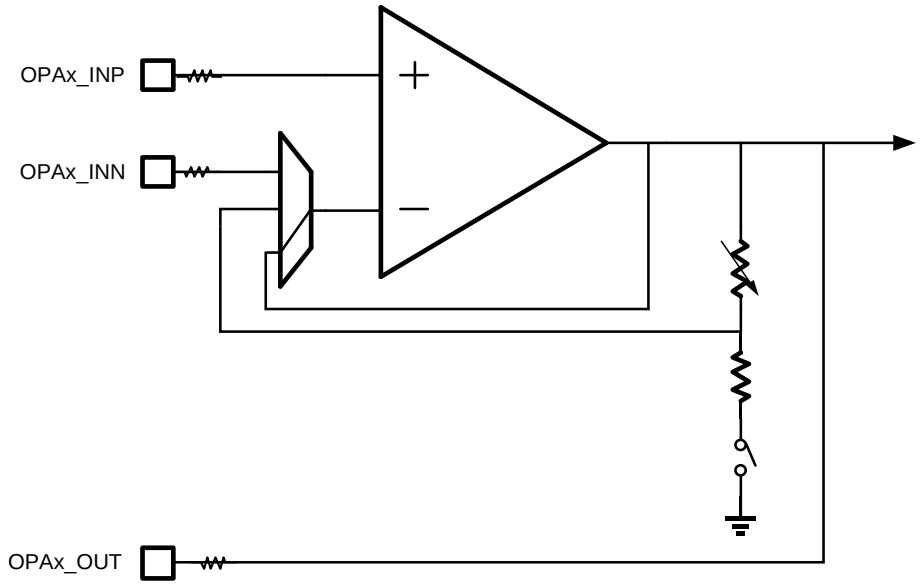


图 16-8OPA 缓冲器模式

软件配置方法：

- 配置OPAxCR.VPSEL和VNSEL选择输入IO
- 配置OPAxCR.OPAxMOD为11，即buffer模式
- 使能OPAx

16.4.4 PGA 模式

PGA模式下，通过调整片内电阻阻值，可以实现固定增益的放大效果，无需连接片外反馈电阻。仅OPA1支持PGA模式，其支持的增益为x2, x4, x8, x16

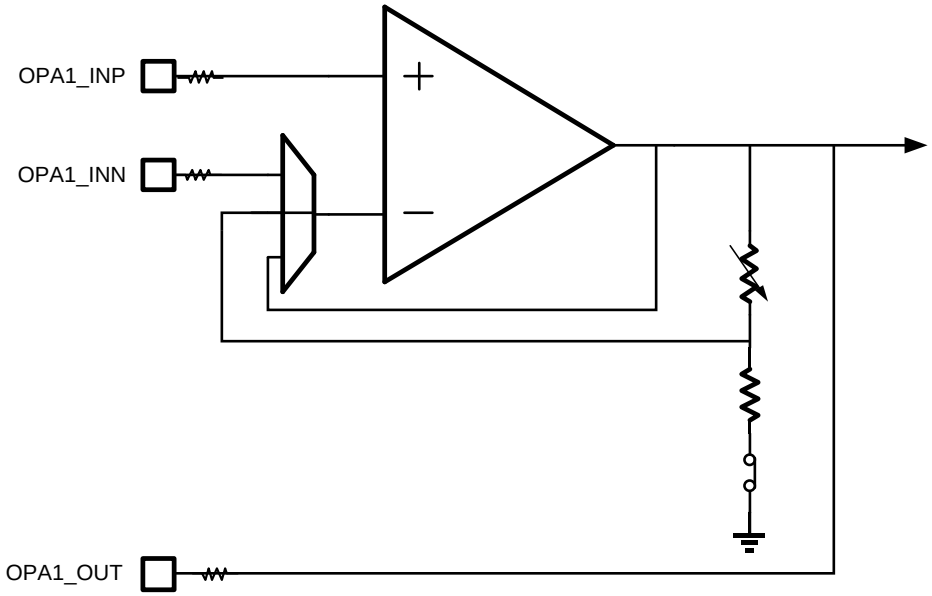


图 16-9OPA PGA 模式

通过配置VN_SEL寄存器，并在OPA1_OUT和OPA1_INN之间连接片外电容，可以实现环路滤波，如下图所示：

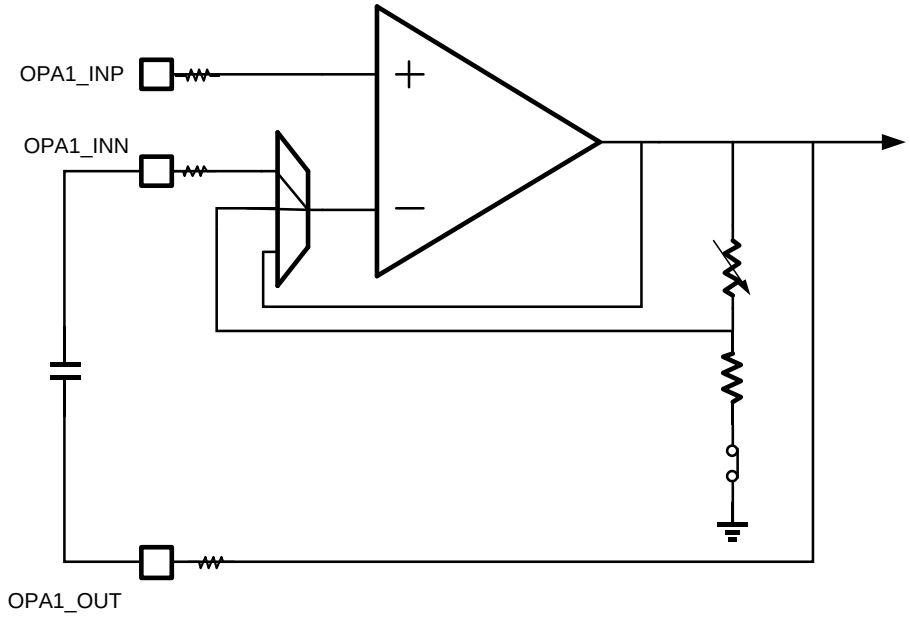


图 16-10OPA 环路滤波

软件配置方法：

- 配置OPA1CR.VPSEL和VNSEL选择输入IO
- 配置OPA1CR.OPA1MOD为10，即PGA模式
- 配置OPA1CR.PGA_GAIN选择增益倍数

- 如果需要片外环路滤波，置位OPA1CR.VN_EXC
- 使能OPA1

16.4.5 Offset 校准

校准功能用于抵消运放输入offset电压。为了避免封装应力影响，校准可以在成测时进行，也可以由用户在reflow焊接后进行。校准完全由软件操作实现，软件通过调整输入差分对镜像电流的大小来补偿固有输入offset电压，并通过读取OPA输出来实现校准。校准N差分对时正反向输入被短接到3/4VDD，校准P差分对时正反向输入被短接到1/4VDD。

用户校准需按以下步骤进行：

- 将OPA设置为BUFFER模式
- 在OPA_INP引脚上施加3/4 VDD的电压，用于校准N端offset
- 使能NCAL_EN，改写OPAx_NCAL寄存器，直到OPA_OUT输出电压等于OPA_INP
- 关闭NCAL_EN并保存校准值
- 在OPA_INP引脚上施加1/4 VDD电压，用于校准P端offset
- 使能PCAL_EN，改写OPAx_PCAL寄存器，直到OPA_OUT输出电压等于OPA_INP
- 关闭PCAL_EN并保存校准值

注意：用户offset校准的精度很大程度上取决于输入电压精度和输出电压测量精度；校准电路的典型调校步长是1.5mV

16.4.6 低功耗比较器

OPA在比较器模式下可以进入低功耗模式，此时典型功耗1.3uA，通过OPA1LPM和OPA2LPM寄存器配置进入，以获得低功耗比较器的功能。

在Buffer模式、PGA模式下不能进入低功耗模式。

休眠模式下使用比较器唤醒

当芯片处于Sleep/DeepSleep休眠模式时，可以通过低功耗比较器输出的上升中断或者下降中断来唤醒芯片。中断的产生不需要时钟，因此可以在深度休眠的情况下实现MCU异步唤醒。

16.4.7 中断及触发信号输出

当OPA配置为比较器模式时，可以产生比较器中断和其他外设模块的触发信号。

比较器中断

OPA配置为比较器后，可以在比较器输出上升沿、下降沿上分别产生独立的中断事件。OPAIE寄存器可以使能或禁止中断输出。OPAxIF标志寄存器在中断事件发生时置位，软件写1清零。软件也可以通过OAPSTA寄存器直接读取比较器的输出值。

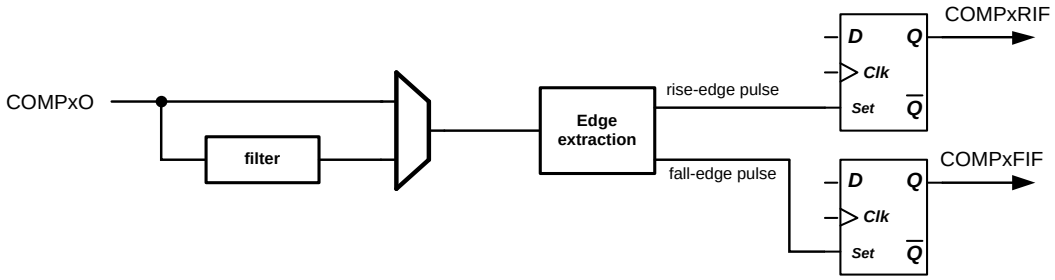


图 16-11OPA 比较器中断产生

16.4.8 低功耗模式下的 OPA

由于OPA电源由VDDA供电，没有低功耗限制，可以在任何低功耗模式下保持工作，由软件配置决定。

注意，OPA工作需要BGQS，因此当OPA使能的情况下进入DeepSleep或RTCBKP时，数字电路不能关闭BGQS。OPA配置为比较器模式时，不需要BGQS，但是需要使能VREF1p22。

16.5 寄存器

offset 地址	名称	符号
OPA1(模块起始地址：0x4001A844)		
0x00000000	OPA1 控制寄存器 （OPA1 Control Register）	OPA1_CR
0x00000004	OPA1 校准寄存器 （OPA1 Calibration Register）	OPA1_CALR
0x00000008	OPA1 中断使能寄存器 （OPA1 Interrupt Enable Register）	OPA1_IER
0x0000000C	OPA1 中断标志寄存器 （OPA1 Interrupt Status Register）	OPA1_ISR
OPA2(模块起始地址：0x4001A854)		
0x00000000	OPA2 控制寄存器 （OPA2 Control Register）	OPA2_CR
0x00000004	OPA2 校准寄存器 （OPA2 Calibration Register）	OPA2_CALR
0x00000008	OPA2 中断使能寄存器 （OPA2 Interrupt Enable Register）	OPA2_IER
0x0000000C	OPA2 中断标志寄存器 （OPA2 Interrupt Status Register）	OPA2_ISR

16.5.1 OPA1 控制寄存器（OPA1_CR）

名称	OPA1_CR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BUFENB	BUFBYP	-					
位权限	R/W-1	R/W-0	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				VNSEL			VPSEL
位权限	U-0				R/W-111			R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF	VN_EXC	PGA_GAIN		MOD		LPM	EN
位权限	R/W-0	R/W-0	R/W-00		R/W-00		R/W-0	R/W-0

位号	助记符	功能描述
31	BUFENB	VREF BUFFER 使能信号 (VREF buffer enable bar) 0：使能 VREF BUFFER 1：关闭 VREF BUFFER
30	BUFBYP	VREF BUFFER 旁路控制 (VREF buffer bypass enable) 0：不 bypass VREF BUFFER 1：bypass VREF BUFFER
29:12	-	RFU：未实现，读为 0
11:9	VNSEL	OPA1 负端输入选择 (OPA1 Negtive Input Select) 000：OPA1_INN1

位号	助记符	功能描述
		001：OPA1_INN2 010：VREF 011：3/4 VREF 100：1/2 VREF 101：1/4 VREF 110：1/8 VREF 111：1/8 VREF
8	VPSEL	OPA1 正端输入选择 (OPA1 Positive Input Select) 0：OPA1_INP1 1：OPA1_INP2
7	DF	OPA1 比较器模式输出数字滤波使能（仅 OPA 配置为比较器模式下有效） (OPA1 Comparator mode digital filter enable) 0：关闭比较器输出数字滤波 1：打开比较器输出数字滤波
6	VN_EXC	OPA1 负端连接 GPIO，仅 OPA1MOD=10 时有效 (OPA1 Negtive Input Connected to GPIO enable) 0：PGA 模式下 OPA1 负端不连接 GPIO 1：PGA 模式下 OPA1 负端同时连接到 GPIO
5:4	PGA_GAIN	PGA 增益选择 (PGA gain select) 00：PGA 增益 x2 01：PGA 增益 x4 10：PGA 增益 x8 11：PGA 增益 x16
3:2	MOD	OPA1 工作模式 (OPA1 mode) 00：standalone 模式 01：比较器模式 10：PGA 模式 11：buffer 模式
1	LPM	OPA1 低功耗控制寄存器 (OPA1 low power mode) 0：正常模式 1：低功耗模式
0	EN	OPA1 使能寄存器 (OPA1 enable) 0：关闭 OPA1 1：使能 OPA1

16.5.2 OPA1 校准寄存器（OPA1_CALR）

名称	OPA1_CALR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	NCAL_EN	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			NCAL				
位权限	U-0			R/W-0 0000				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PCAL_EN	-						

位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			PCAL				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31	NCAL_EN	OPA1 负端输入校准使能，1 有效 (Negative input calibration enable)
30:21	-	RFU：未实现，读为 0
20:16	NCAL	OPA1 负输入端校准 trim 信号，最高位为符号位 (OPA1 negative input calibration) 01111：输出电压减小最大 00001：输出电压减小最小 00000：输出电压不变 10000：输出电压不变 10001：输出电压增加最小 11111：输出电压增加最大
15	PCAL_EN	OPA1 正端输入校准使能，1 有效 (Positive input calibration enable)
14:5	-	RFU：未实现，读为 0
4:0	PCAL	OPA1 正输入端校准 trim 信号，最高位为符号位 (OPA1 positive input calibration) 01111：输出电压减小最大 00001：输出电压减小最小 00000：输出电压不变 10000：输出电压不变 10001：输出电压增加最小 11111：输出电压增加最大

16.5.3 OPA1 中断使能寄存器（OPA1_IER）

名称	OPA1_IER							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						FIE	RIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	FIE	OPA1 比较器模式输出下降沿中断使能

位号	助记符	功能描述
		(OPA1 comparator mode fall interrupt enable) 1：使能中断输出 0：禁止中断输出
0	RIE	OPA1 比较器模式输出上升沿中断使能 (OPA1 comparator mode rise interrupt enable) 1：使能中断输出 0：禁止中断输出

16.5.4 OPA1 中断标志寄存器（OPA1_ISR）

名称	OPA1_ISR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OUT	-						
位权限	R	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						FIF	RIF
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	OUT	OPA1 比较器模式输出电平，只读 (OPA1 comparator mode output,read only)
14:2	-	RFU：未实现，读为 0
1	FIF	OPA1 比较器模式输出下降沿中断标志，硬件置位，软件写 1 清零 (OPA1 comparator mode fall interrupt flag,write 1 to clear)
0	RIF	OPA1 比较器模式输出上升沿中断标志，硬件置位，软件写 1 清零 (OPA1 comparator mode rise interrupt flag,write 1 to clear)

16.5.5 OPA2 控制寄存器（OPA2_CR）

名称	OPA2_CR							
offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				VNSEL			VPSEL

位权限	U-0				R/W-111			R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF	-			MOD		LPM	EN
位权限	R/W-0	U-0			R/W-00		R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:9	VNSEL	OPA2 负端输入选择 (Negative input select) 000：OPA2_INN1 001：OPA2_INN2 010：VREF 011：3/4 VREF 100：1/2 VREF 101：1/4 VREF 110：1/8 VREF 111：1/8 VREF
8	VPSEL	OPA2 正端输入选择 (Positive input select) 0：OPA2_INP1 1：OPA2_INP2
7	DF	OPA2 比较器模式输出数字滤波使能（仅 OPA 配置为比较器模式下有效）(OPA2 comparator mode digital filter enable) 0：关闭比较器输出数字滤波 1：打开比较器输出数字滤波
6:4	-	RFU：未实现，读为 0
3:2	MOD	OPA2 工作模式 (OPA2 mode) 00：standalone 模式 01：比较器模式 10：RFU 11：buffer 模式
1	LPM	OPA2 低功耗控制寄存器 (OPA2 low power mode) 0：正常模式 1：低功耗模式
0	EN	OPA2 使能寄存器 (OPA2 enable) 0：关闭 OPA2 1：使能 OPA2

16.5.6 OPA2 校准寄存器（OPA2_CALR）

名称	OPA2_CALR							
offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	NCAL_EN	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			NCAL				
位权限	U-0			R/W-0 0000				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PCAL_EN	-						

位权限	R/W-0	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			PCAL				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31	NCAL_EN	OPA2 负端输入校准使能，1 有效(Negative input calibration enable)
30:21	-	RFU：未实现，读为 0
20:16	NCAL	OPA2 负输入端校准 trim 信号，最高位为符号位(OPA2 negative input calibration) 01111：输出电压减小最大 00001：输出电压减小最小 00000：输出电压不变 10000：输出电压不变 10001：输出电压增加最小 11111：输出电压增加最大
15	PCAL_EN	OPA2 正端输入校准使能，1 有效(Positive input calibration enable)
14:5	-	RFU：未实现，读为 0
4:0	PCAL	OPA2 正输入端校准 trim 信号，最高位为符号位(OPA2 positive input calibration) 01111：输出电压减小最大 00001：输出电压减小最小 00000：输出电压不变 10000：输出电压不变 10001：输出电压增加最小 11111：输出电压增加最大

16.5.7 OPA2 中断使能寄存器（OPA2_IER）

名称	OPA2_IER							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						FIE	RIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	FIE	OPA2 比较器模式输出下降沿中断使能

位号	助记符	功能描述
		(OPA2 comparator mode fall interrupt enable) 1：使能中断输出 0：禁止中断输出
0	RIE	OPA2 比较器模式输出上升沿中断使能 (OPA2 comparator mode rise interrupt enable) 1：使能中断输出 0：禁止中断输出

16.5.8 OPA2 中断标志寄存器（OPA2_ISR）

名称	OPA2_ISR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OUT	-						
位权限	R	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						FIF	RIF
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	OUT	OPA2 比较器模式输出电平，只读 (OPA1 comparator mode output)
14:2	-	RFU：未实现，读为 0
1	FIF	OPA2 比较器模式输出下降沿中断标志，硬件置位，软件写 1 清零 (OPA2 comparator mode fall interrupt flag,write 1 to clear)
0	RIF	OPA2 比较器模式输出上升沿中断标志，硬件置位，软件写 1 清零 (OPA2 comparator mode rise interrupt flag,write 1 to clear)

17 模拟比较器（Comparator）

17.1 概述

芯片集成2个比较器，支持以下特性：

- 2个比较器，轨到轨输入
- 灵活的输入选择
  - IO引脚输入
  - 内部基准电压及其分压
- 中断事件可唤醒MCU
- 输出信号可连接到GPIO，也可以作为触发源连接到定时器、ADC

17.2 结构框图

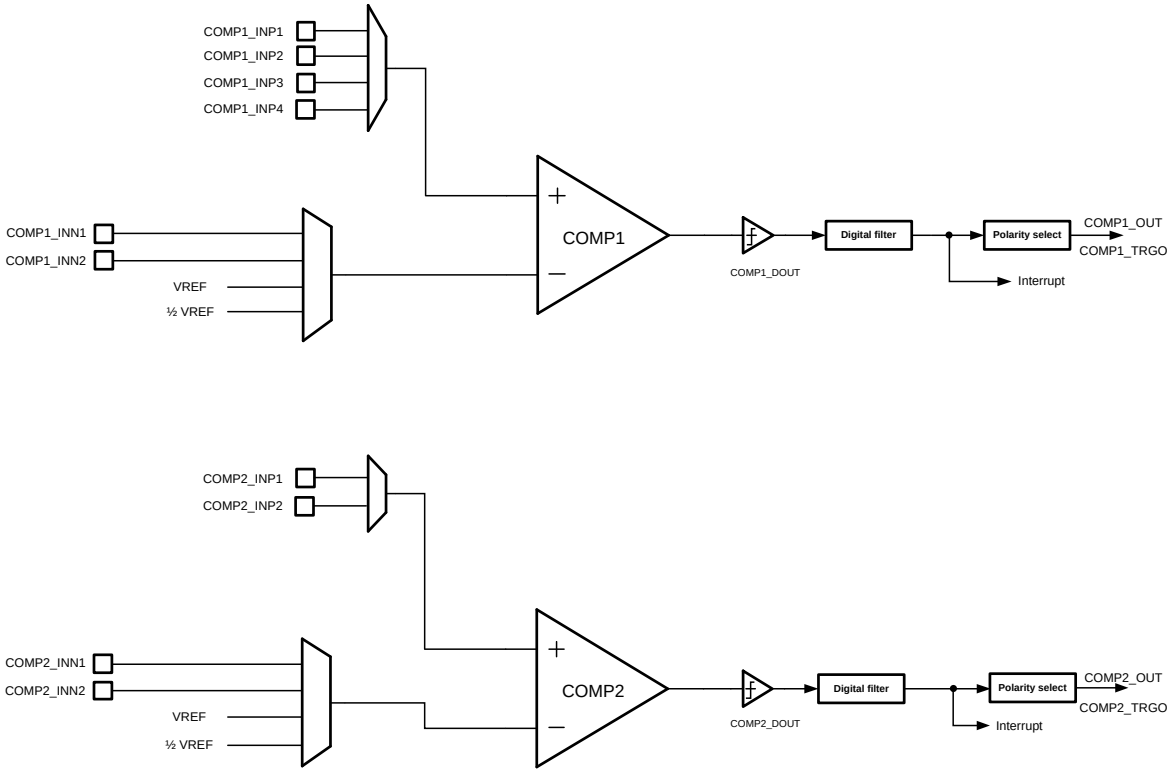


图 17-1 比较器电路框图

比较器结构如上图所示，基准电压VREF输入经过BUFFER模块后输出VREF和VREF分压。两个比较器完全相同，输入端连接方式如结构图所示，两个比较器产生输出信号输出至数字电路。

比较器输入电压范围0~VDD，建立时间小于15us，传输延迟小于2us。

17.3 引脚定义

比较器输入可以来自于GPIO模拟通道，同时其输出可以从FOUT0和FOUT1送到片外使用。

引脚	COMPx	符号	功能
PD4	COMP1	COMP1_INP1	比较器正端输入
PD5		COMP1_INP2	
PA10		COMP1_INN1	比较器负端输入
PD11		COMP1_OUT(FOUT0)	比较器输出
PA8	COMP2	COMP2_INP1	比较器正端输入
PA9		COMP2_INP2	
PA4		COMP2_INN1	比较器负端输入
PA5		COMP2_INN2	
PB12		COMP2_OUT(FOUT1)	比较器输出

表 17-1 比较器引脚列表

17.4 功能描述

17.4.1 基本功能

比较器比较正端输入电压和负端输入电压，当正端电压高于负端电压时，输出逻辑高电平，反之输出逻辑低电平。

正端输入电压可以从多个引脚输入选择，负端输入电压可以选择引脚输入或者内部基准电压。

比较器输出的逻辑信号可以产生中断信号。

17.4.2 输出数字滤波

比较器输出支持数字滤波功能。滤波方式是采用APBCLK连续采样比较器输出，保持3次采样周期数为相同电平时，才认为这个电平有效。

下图数字滤波的示意图：

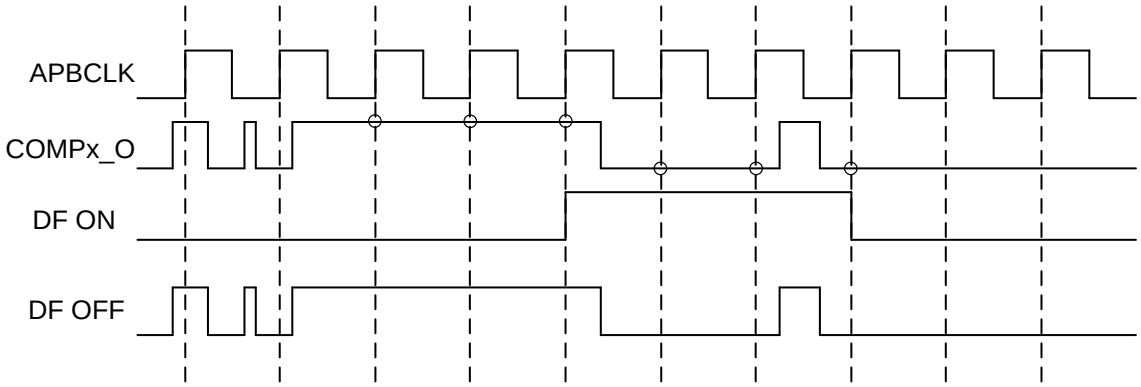


图 17-2 数字滤波波形示意图

17.4.3 内部比较基准选择

比较器负端可以选择输入芯片内部比较基准电压，比较基准来自于VREF及其分压。比较器与OPA共享同一套分压电路，比较器1使用的内部基准由OPA1控制寄存器OPA1CR.VNSEL控制，比较器2使用的内部基准由OPA2控制寄存器OPA2CR.VNSEL控制。

下面以COMP1为例说明软件配置方法：

- 清零OPA1CR.BUFENB寄存器，使能OPA1内部基准电压缓冲器
- 配置COMP1CR.V1NSEL为10，选择VREF 1.25V为比较基准
- 置位CMP1EN，使能比较器1

17.4.4 中断及触发信号输出

比较器输出可以产生中断和其他外设模块的触发信号。

比较器中断

比较器输出可以在上升沿、下降沿上分别产生独立的中断事件。CMPxIE寄存器可以使能或禁止中断输出。CMPxIF标志寄存器在中断事件发生时置位，软件写1清零。软件也可以通过CMPxO寄存器直接读取比较器的输出值。

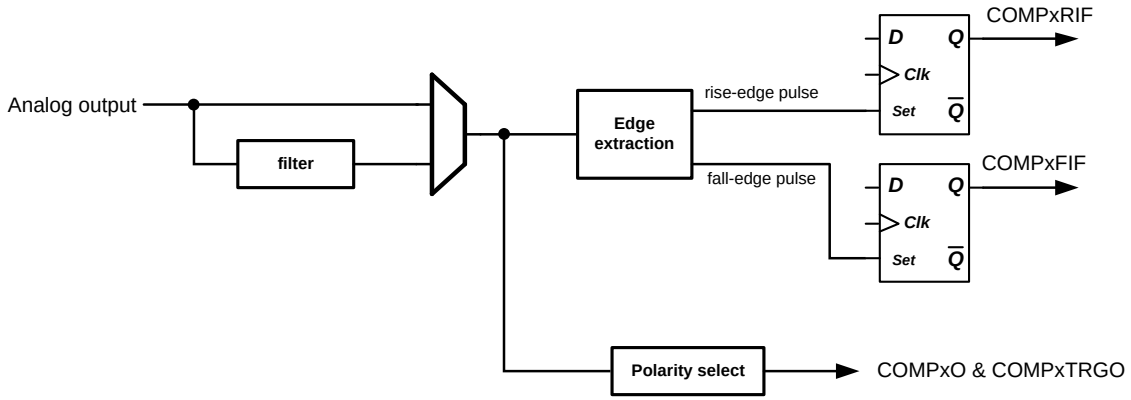


图 17-3 比较器中断产生

比较器触发信号输出

可以在比较器输出的上升沿、下降沿分别或同时产生触发信号输出。需要输出触发信号时，必须使能COMP总线时钟，当触发事件发生时，在APBCLK的上升沿产生一个APBCLK周期的高电平触发信号。触发信号可以被连接到定时器的内部触发输入，或者ADC的内部触发输入。

CMPxSEL寄存器可以配置在比较器输出的什么边沿上产生触发信号输出，TRGOEN用于使能或禁止触发信号输出。

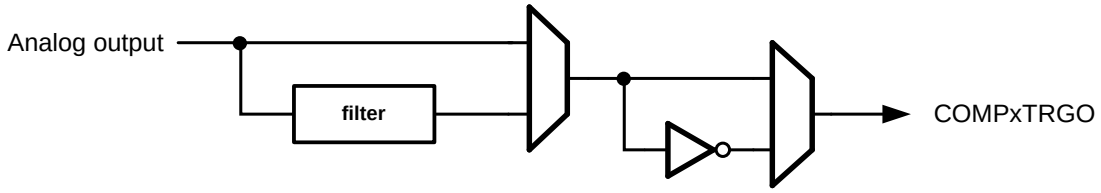


图 17-4 比较器触发输出信号产生

17.4.5 比较器输出连接

比较器的输出可以被连接到GPTIMx的输入，以便定时器自动记录比较器输出翻转次数。

具体连接关系请参考27.4.4内部触发信号（ITRx）的捕捉。

比较器的输出也可以作为启动ADC转换的触发信号，请参考32.4.7转换触发。

17.5 寄存器

模块起始地址：0x4001A870

offset 地址	名称	符号
COMP(模块起始地址：0x4001A870)		
0x00000000	COMP 控制寄存器 1 （ComparatorControl Register 1）	COMP_CR1
0x00000004	COMP 控制寄存器 2 （Comparator Control Register 2）	COMP_CR2
0x00000008	COMP 中断配置寄存器 （Comparator Interrupt Config Register）	COMP_ICR
0x0000000C	COMP 中断标志寄存器 （Comparator Interrupt Status Register）	COMP_ISR

17.5.1 COMP 控制寄存器 1（COMP_CR1）

名称	COMP_CR1							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CMP1O
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		POL	V1PSEL		V1NSEL		CMP1EN
位权限	U-0		R/W-0	R/W-00		R/W-00		R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	CMP1O	比较器 1 输出，软件只读 (Comparator1 Output)
7:6	-	RFU：未实现，读为 0
5	POL	比较器 1 输出极性控制 (Comparator1 output Polarity) 0：输出不取反 1：输出取反
4:3	V1PSEL	比较器 1 正极输入选择 (Comparator1 positive input select) 00：COMP1_INP1 (PD4) 01：COMP1_INP2 (PD5) 10：COMP1_INP3 (PB12) 11：RFU
2:1	V1NSEL	比较器 1 负极输入选择 (Comparator1 negative input select) 00：COMP1_INN1 01：RFU 10：VREF = 1.2V 11：VREF/2 = 0.6V

位号	助记符	功能描述
0	CMP1EN	比较器 1 使能位 (Comparator1 Enable) 0：关闭比较器 1 1：使能比较器 1

17.5.2 COMP 控制寄存器 2（COMP_CR2）

名称	COMP_CR2							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CMP2O
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		POL	-	V2PSEL	V2NSEL		CMP2EN
位权限	U-0		R/W-0	U-0	R/W-0	R/W-00		R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	CMP2O	比较器 2 输出，软件只读 (Comparator1 Output)
7:6	-	RFU：未实现，读为 0
5	POL	比较器 2 输出极性控制 (Comparator1 output Polarity) 0：输出不取反 1：输出取反
4	-	RFU：未实现，读为 0
3	V2PSEL	比较器 2 正极输入选择 (Comparator1 positive input select) 0：COMP2_INP1 (PA8) 1：COMP2_INP2 (PA9)
2:1	V2NSEL	比较器 2 负极输入选择 (Comparator1 negative input select) 00：COMP2_INN1 (PA4) 01：COMP2_INN2 (PA5) 10：VREF = 1.2V 11：VREF/2 = 0.6V
0	CMP2EN	比较器 2 使能位 (Comparator1 Enable) 0：关闭比较器 1 1：使能比较器 1

17.5.3 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CMP2DF	CMP1DF
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	CMP2SEL		CMP1SEL		CMP2IE	CMP1IE
位权限	U-0	U-0	R/W-00		R/W-00		R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	CMP2DF	比较器 2 数字滤波使能 (Comparator2 Digital Filter enable) 0：禁止数字滤波 1：使能数字滤波
8	CMP1DF	比较器 1 数字滤波使能 (Comparator1 Digital Filter enable) 0：禁止数字滤波 1：使能数字滤波
7:6	-	RFU：未实现，读为 0
5:4	CMP2SEL	比较器 2 中断源选择 (Comparator2 interrupt edge select) 00/11：比较器 2 输出上升或下降沿产生中断 01：比较器 2 输出上升沿产生中断 10：比较器 2 输出下降沿产生中断
3:2	CMP1SEL	比较器 1 中断源选择 (Comparator1 interrupt edge select) 00/11：比较器 1 输出上升或下降沿产生中断 01：比较器 1 输出上升沿产生中断 10：比较器 1 输出下降沿产生中断
1	CMP2IE	比较器 2 中断使能 (Comparator2 Interrupt Enable) 1：允许中断 0：禁止中断
0	CMP1IE	比较器 1 中断使能 (Comparator1 Interrupt Enable) 1：允许中断 0：禁止中断

*注：为了避免误触发中断，应在关闭中断使能的情况下设置中断源选择寄存器。

17.5.4 COMP 中断标志寄存器（COMP_ISR）

名称	COMP_ISR							
offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						CMP2IF	CMP1IF
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	CMP2IF	比较器 2 中断标志，硬件置位，软件写 1 清零 (Comparator2 Interrupt Flag,write 1 to clear)
0	CMP1IF	比较器 1 中断标志，硬件置位，软件写 1 清零 (Comparator1 Interrupt Flag,write 1 to clear)

18 硬件除法器（HDIV）

18.1 概述

硬件除法器模块用于帮助软件加速除法运算。该硬件除法器是一个有符号数整数除法器，可以输出32bit被除数和16bit除数，输出23bit商和32bit余数。

特点：

- 有符号整数运算（二进制补码格式）
- 32bit被除数、16bit除数
- 输出32bit商和32bit余数
- 除以0警告
- 一次计算需要8个24MHz周期

18.2 工作流程

软件按照如下步骤调用硬件除法器。

- 向DIVEND寄存器写入32bit被除数（二进制补码）
- 向DIVSOR寄存器写入16bit除数（二进制补码）
- 硬件除法器在软件写入DIVSOR后自动开始运算，同时置位BUSY寄存器
- 软件查询BUSY标志，直到运算完成后BUSY自动清零
- 查询DIV_BY_0标志
- 读取QUOT寄存器中的商
- 读取REMD寄存器中的余数

18.3 寄存器

offset 地址	名称	符号
HDIV(模块起始地址:0x40019000)		
0x00000000	被除数寄存器 （Dividend Register）	HDIV_END
0x00000004	除数寄存器 （Divisor Regsiter）	HDIV_SOR
0x00000008	商寄存器 （Quotient Register）	HDIV_QUOT
0x0000000C	余数寄存器 （Reminder Register）	HDIV_REMD
0x00000010	状态标志寄存器 （Status Register）	HDIV_SR

18.3.1 被除数寄存器（HDIV_END）

名称	HDIV_END							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIVEND[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DIVEND[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVEND[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVEND[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	DIVEND	32bit 有符号被除数 (Dividend)

18.3.2 除数寄存器（HDIV_SOR）

名称	HDIV_SOR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVSOR[15:8]							
位权限	R/W-0000 0000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVSOR[7:0]							
位权限	R/W-0000 0001							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DIVSOR	16bit 有符号除数 (Divisor)

18.3.3 商寄存器（HDIV_QUOT）

名称	HDIV_QUOT							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QUOT[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QUOT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QUOT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QUOT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	QUOT	32bit 有符号商(QUTO ,read only) （仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

18.3.4 余数寄存器（HDIV_REMD）

名称	HDIV_REMD							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REMD[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REMD[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0

15:0	REMD	16bit 有符号余数 (Reminder,read only) （仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

18.3.5 状态标志寄存器（HDIV_SR）

名称	HDIV_SR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DIV_BY_0	BUSY
位权限	U-0						R-0	R-0

位号	助记符	功能描述
31:2	-	RFU：未实现，读为 0
1	DIV_BY_0	除数为 0 标志 (divided by 0 flag,read only) 1：除数为 0 0：除数不为 0
0	BUSY	运算过程指示 (Busy flag,read only) 1：HDIV 在计算过程中，结果未就绪 0：计算完毕，结果就绪 软件在写入除数后，HDIV 开始计算，软件应查询 BUSY 为低后再读取商和余数寄存器

19 I²C

19.1 概述

I²C 模块实现 MCU 与外部 I2C 接口器件之间的同步通信，硬件实现串并转换。支持 I2C 的主机和从机模式，不支持多主机模式。

特点：

- 1 路独立 I2C 接口
- 支持主机和从机模式，不支持多主机模式
- 支持 7 位或 10 位从机地址
- 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)和 Fm+(1Mbps)
- 支持 DMA，主机和从机独立 DMA 通道
- 低功耗从机设计，可以在没有系统时钟的情况下收发数据
- 支持异步从机地址匹配唤醒、数据帧接收完成唤醒或 START 检测唤醒

19.2 结构框图

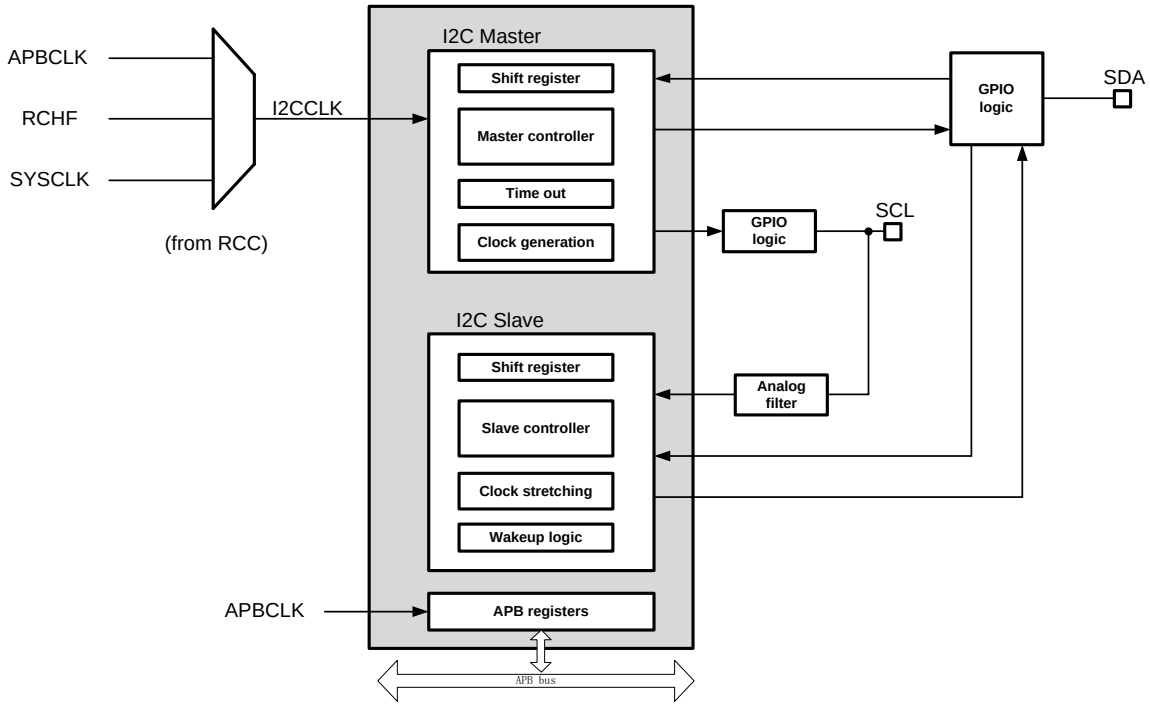


图 19-1 I2C 模块框图

19.3 引脚定义和上拉电阻范围

I2C 模块使用 2 个真开漏引脚与外部器件通信，工作时需要外接上拉电阻：

引脚	I2Cx	符号	功能
PA11	I2C	SCL	I2C 时钟
PA12		SDA	I2C 数据

表 19-1I2C 引脚列表

I2C 总线协议规定了 standard-mode、fast-mode 和 fast-mode plus 的信号最大上升时间，以及 IO 能够支

Symbol	Parameter	Conditions	Standard-mode		Fast-mode	
			Min	Max	Min	Max
V_{IL}	LOW-level input voltage[1]		–0.5	$0.3V_{DD}$	–0.5	$0.3V_{DD}$
V_{IH}	HIGH-level input voltage[1]		$0.7V_{DD}$	[2]	$0.7V_{DD}$	[2]
V_{hys}	hysteresis of Schmitt trigger inputs		-	-	$0.05V_{DD}$	-
V_{OL1}	LOW-level output voltage 1	(open-drain or open-collector) at 3 mA sink current; V_{DD}>2V	0	0.4	0	0.4
V_{OL2}	LOW-level output voltage 2	(open-drain or open-collector) at 2 mA sink current[3]; $V_{DD}\leq 2V$	-	-	0	$0.2V_{DD}$
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	3	-
		V_{OL} = 0.6 V[4]	-	-	6	-
t_{of}	output fall time from $V_{Ihmin}$to V_{ILmax}		-	250[5]	1 × (V_{DD} / 5.5 V)[6]	250[5]
t_{SP}	pulse width of spikes that must be suppressed by the input filter		-	-	0	50[8]
I_i	input current each I/O pin	$0.1V_{DD}< V_I< 0.9V_{Ddmax}$	–10	+10	–10[9]	+10[9]

C_i	capacitance for each I/O pin[10]		-	10	-	10

表 19-2I2C 协议电参数表

下表则定义了总线信号允许的最大上升时间和下降时间。

Symbol	Parameter	Conditions	Standard-mode		Fast-mode	
			Min	Max	Min	Max
f_{SCL}	SCL clock frequency		0	100	0	400
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters (see Remark in Section 4.1)	5.0	-	-	-
		I^2C-bus devices	0[3]	-[4]	0[3]	-[4]
$t_{SU;DAT}$	data set-up time		250	-	100[5]	-
t_r	rise time of both SDA and SCL signals		-	1000	20	300
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	$1 \times (V_{DD} / 5.5\ V)$	300
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-
C_b	capacitive load for each bus line[10]		-	400	-	400
$t_{VD;DAT}$	data valid time[11]		-	3.45[4]	-	0.9[4]
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45[4]	-	0.9[4]
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	$0.1V_{DD}$	-	$0.1V_{DD}$	-
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	$0.2V_{DD}$	-	$0.2V_{DD}$	-

表 19-3I2C 协议时序参数表

根据以上协议规范，我们可以计算外部上拉电阻的合理范围。

假设总线信号从 VIL=0.3VDD 上升到 VIH=0.7VDD，则充电时间可以计算为：

$$V(t1) = 0.3 \times V_{DD} = V_{DD}\,(1 - e^{-t1\,/\,RC}); \quad t1 = 0.3566749 \times RC$$

$$V(t2) = 0.7 \times V_{DD} = V_{DD}\,(1 - e^{-t2\,/\,RC}); \quad t2 = 1.2039729 \times RC$$

$$T = t2 - t1 = 0.8473 \times RC$$

根据总线容性负载大小和协议对信号最大上升时间的要求，我们可以计算上拉电阻的最大值：

$$R_{p(\max)} = \frac{t_r}{0.8473 \times C_b}$$

而上拉电阻的最小值则由总线电源电压 VDD 和 IO 电流 sink 能力决定，FM33LC0XX 的 I2C 引脚 sink 能力是标准/快速模式 3mA，Fm+模式 20mA。

$$R_{p(\min)} = \frac{VDD - V_{OL(\max)}}{I_{OL}}$$

19.4 时钟

I2C 主机和从机都采用了双时钟结构：

- 主机和从机的总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 I2C 内部寄存器时，必须使能 PCLK。
- 主机的数据收发时钟用 I2CCLK 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK、RCMF，能够独立于 APBCLK 工作。必须使能 I2CCLK 才能进行数据收发。
- 从机的数据收发时钟使用 SCL 总线时钟输入，因此无需系统时钟就可以进行数据收发

PCLK 和 I2CCLK 的控制都在 CMU 模块内完成，进行 I2C 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 I2C 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，I2C 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 I2C 以较高的波特率进行数据通信。

理论上 PCLK 和波特率时钟之间没有相对关系的约束，波特率时钟可以快于或者慢于 PCLK。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

19.5 接口时序

19.5.1 接口时序图

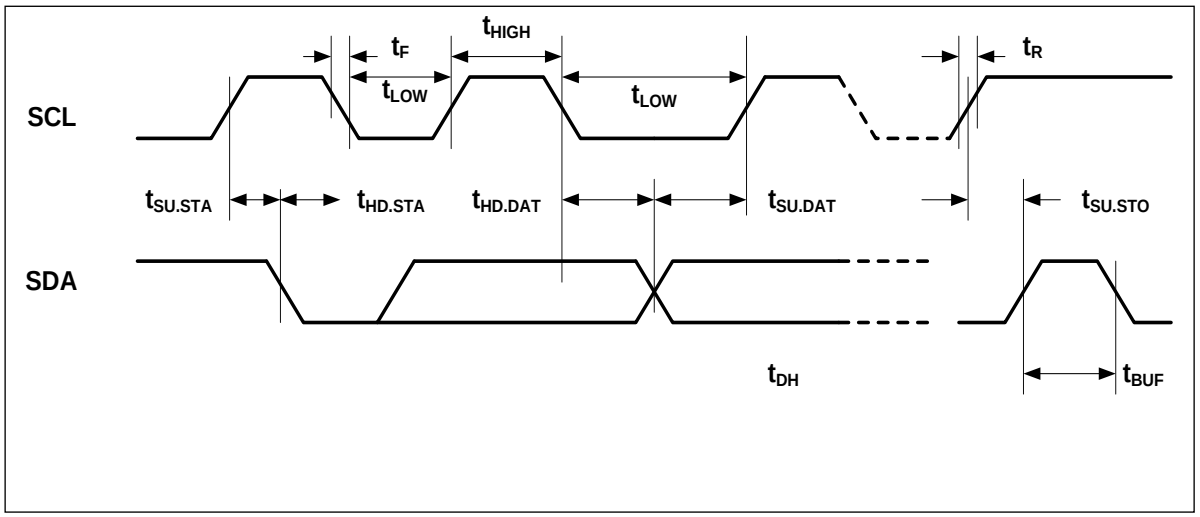


图 19-2 I²C 总线时序

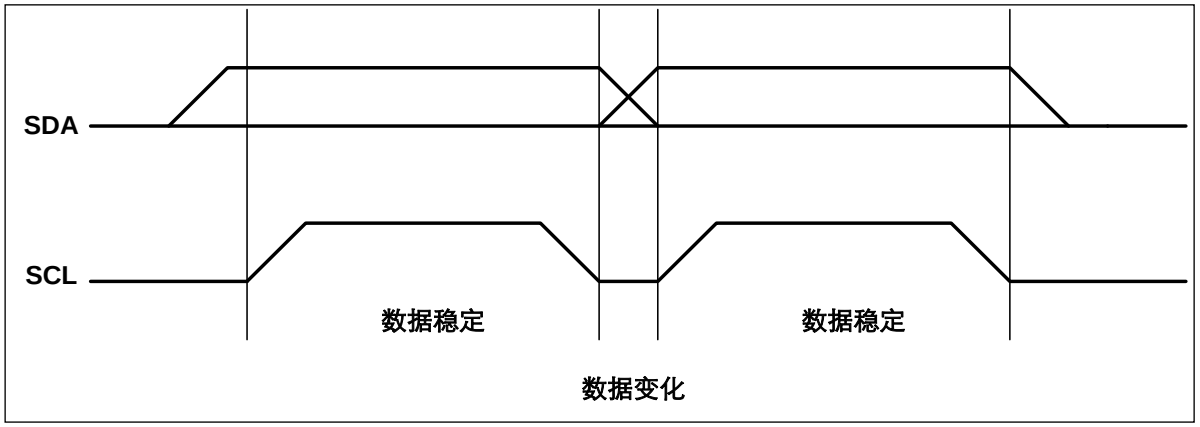


图 19-3 数据有效时序

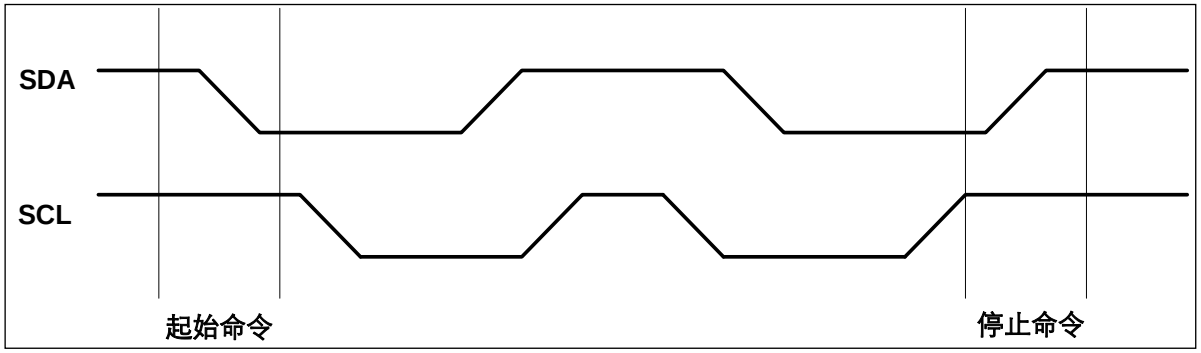


图 19-4 起始（Start）与停止(Stop)命令定义

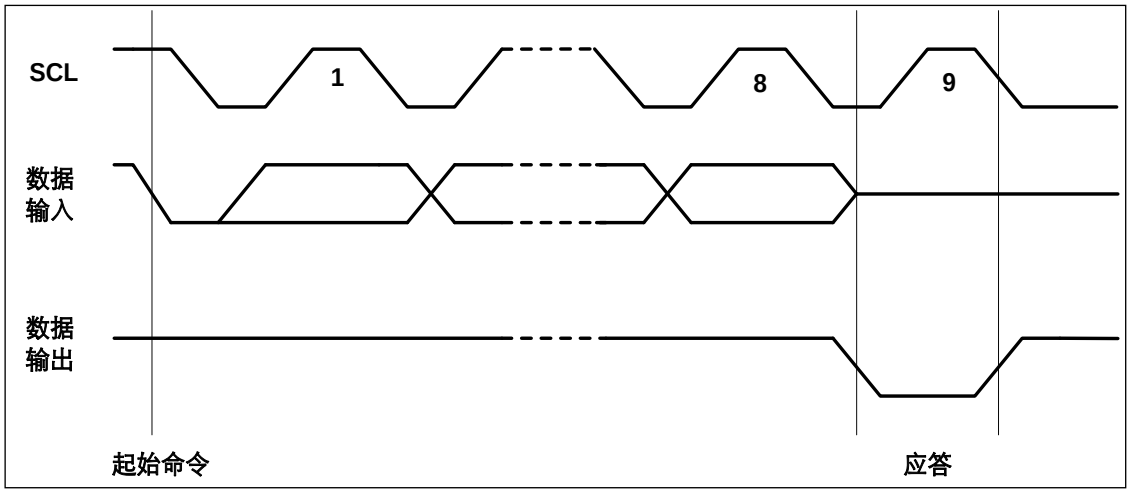


图 19-5 输出应答(ACK)

19.5.2 接口时序描述

时钟有效时序：SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化（参见图 19-3）；当数据在 SCL 为高时变化，将视为下文所述的一个起始或停止命令。

起始命令：当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始（参见图 19-4）。

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使 EEPROM 进入等待态低功耗模式（参见图 19-4）。

输出应答：SDA 上的数据都是以 8 位为一组串行输入和输出的，MSB 先发，接收方在收完每个字节后应当在第 9 个周期回发一个回应 acknowledge 位（以下简称 ack），ack 的时钟由主机提供。发送方在 ack 期间悬空 SDA，接收方须将 SDA 拉低，确保 ack 时钟高电平期间 SDA 为低，形成有效的 ack 信号(参见图 19-5)。

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	F_{SCL}	0	100	0	400	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	—	0.6	—	us
启动条件保持时间	$T_{HD:STA}$	4.0	—	0.6	—	us
时钟低电平时间	T_{LOW}	4.7	—	1.3	—	us
时钟高电平时间	T_{HIGH}	4.0	—	0.6	—	us
数据输入建立时间	$T_{SU:DAT}$	250	—	100(4)	—	ns
数据输入保持时间	$T_{HD:DAT}$	5.0 0(2)	— 3.45(3)	— 0(2)	— 0.9(3)	us us
SDA 和 SCL 上升时间	T_R	—	1000	20+0.1Cb(5)	300	ns
SDA 和 SCL 下降时间	T_F	—	300	20+0.1Cb(5)	300	ns
停止条件建立时间	$T_{SU:STO}$	4.0	—	0.6	—	us
总线空闲时间	T_{BUF}	4.7	—	1.3	—	us

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	
总线的容性负载	Cb	—	400	—	400	Pf
噪声容限低值	V_{nL}	$0.1V_{DD}$	—	$0.1V_{DD}$	—	V
噪声容限高值	V_{nH}	$0.2V_{DD}$	—	$0.2V_{DD}$	—	V

表 19-4 I^2C 接口时序要求

19.6 I²C 工作模式

I2C模块支持以下工作模式：

- 主机接收
- 主机发送
- 从机接收
- 从机发送

芯片上电后I2C模块默认关闭，主机和从机都不工作。软件需要根据应用选择模块工作模式，通过设置MSPEN来使能主机通信，或设置SSPEN来使能从机通信。

主机和从机不能同时工作，因为他们复用相同的IO引脚作为SCL和SDA，原则上禁止软件同时将MSPEN和SSPEN置1。

19.7 I²C 从机地址格式

I2C总线协议定义了以下保留地址，对其中多数保留地址，I2C从机硬件不做合法性判断，软件可以根据收到的地址进行自定义的处理。

但是对于10bit从机地址应用，即SSPCON.A10EN=1的情况下，要求1st字节必须以11110开头，否则将触发ADDR_ERROR错误标志。而在SSPCON.A10EN=0的情况下，如果从机收到了11110开头的地址字节，也会置位ADDR_ERROR错误标志。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

表 19-5I2C 从机保留地址定义

19.8 I²C 初始化

进行I2C通信前必须正确的初始化I2C模块，建议软件按照以下步骤进行初始化操作：

- 清零RCC模块的I2CRST寄存器，确保I2C模块不处于复位状态
- 置位RCC模块的I2C_APBEN寄存器，使能I2C模块寄存器总线接口时钟
- 配置RCC模块的I2C_CKSEL和I2C_CKEN寄存器，选择并使能I2C工作时钟（如果是从机模式，不需要这个步骤）
- 根据需要配置模拟滤波使能（SCL和SDA输入模拟滤波，>50ns）

19.8.1 IO 配置

FM33LC0XX 最多有两组引脚用于数据传输，开始 I2C 通信前需将对应引脚的 FCR 寄存器设置为 AF：

SDA：PA12/PD12

SCL：PA11/PB15

注意，如果 PA11 和 PB15 同时配置为 SDA 功能，则 PA11 被连接到 I2C 模块，PB15 无效；如果 PA12 和 PD12 同时配置为 SCL 功能，则主机模式下两个引脚都会输出 SCL 信号，从机模式下只有 PA12 被连接到 I2C 从机的 SCL 输入。

PA11 和 PA12 是强驱动真 OD 引脚，必须搭配外部总线上拉电阻使用，并且具备 20mA sink 电流能力，能够支持 Fm+模式。

19.8.2 主机波特率配置

I2C 主机需要在使能前配置通信波特率，而从机不需要配置。

MSPBRG[8:0]波特率配置寄存器用于产生通信波特率。MSPBRG 是 9 bit 波特率分频系数，波特率计算公式如下：

$T_{SCL}=2T_{BRG}$

$T_{BRG}=2\times T_{I2CCLK}\times(MSPBRG[8:0]+1)$；$T_{I2CCLK}$ 为 I2C 工作时钟周期，即：

$$MSPBRG = F_{I2CCLK}/(4*F_{SCL}) - 1$$

例如对于 100k 波特率，若 I2C 工作时钟为 8M，则 MSPBRG=19。

19.8.3 从机的输入模拟滤波和输出延迟

模拟滤波功能仅针对 SCL 引脚，并且只有从机的 SCLi 输入信号上可以使能模拟滤波功能。

同时，从机的 SDA 输出延迟，通过在 SDAo 上增加大于 300ns 的模拟延迟，来确保 SDA 相对于 SCL 下降沿的输出保持时间。

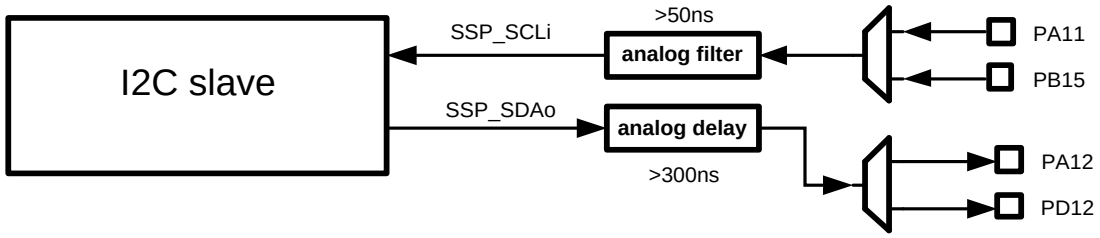


图 19-6 从机信号滤波

19.9 I²C 主机功能

FM33LC0XX的I2C主机模式不支持多主机总线，因此挂在总线上的其他设备都是从机。总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。

I2C总线通信总是由主机发起，主机模式支持7bit或10bit寻址。

19.9.1 7bit 寻址

在7bit寻址时，主机发送的第一个字节包含从机地址和传输方向位（$R/\overline{W}$），根据$R/\overline{W}$决定后续传输是主机向从机写入数据（$R/\overline{W}$=0）或主机从从机读取数据（$R/\overline{W}$=1）。

名字	Slave Address Byte							
位	7	6	5	4	3	2	1	0
位名	address							R/W

位描述：

位号	助记符	功能描述
7-1	address	Slave device address
0	R/W	0: Write 表示发送数据（master 发送） 1: Read 表示请求数据（slave 回发）

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

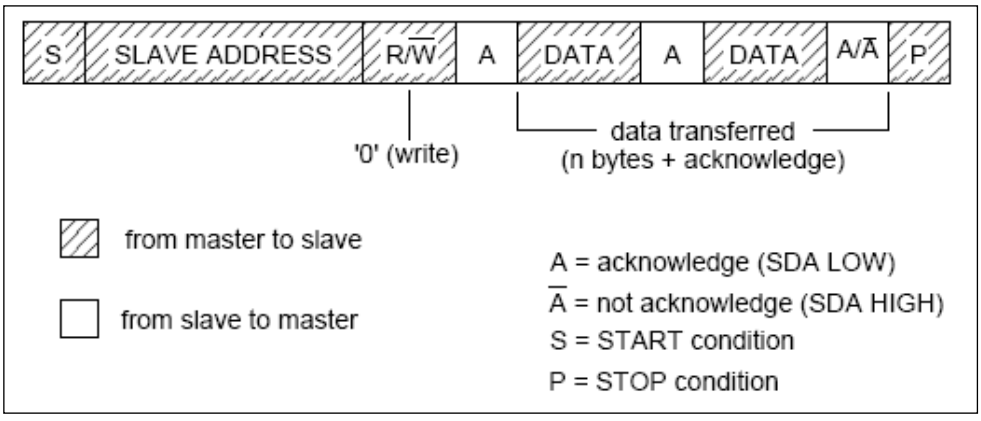


图 19-7 主机向 7 位地址从机写入数据时的帧格式

1、主机发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

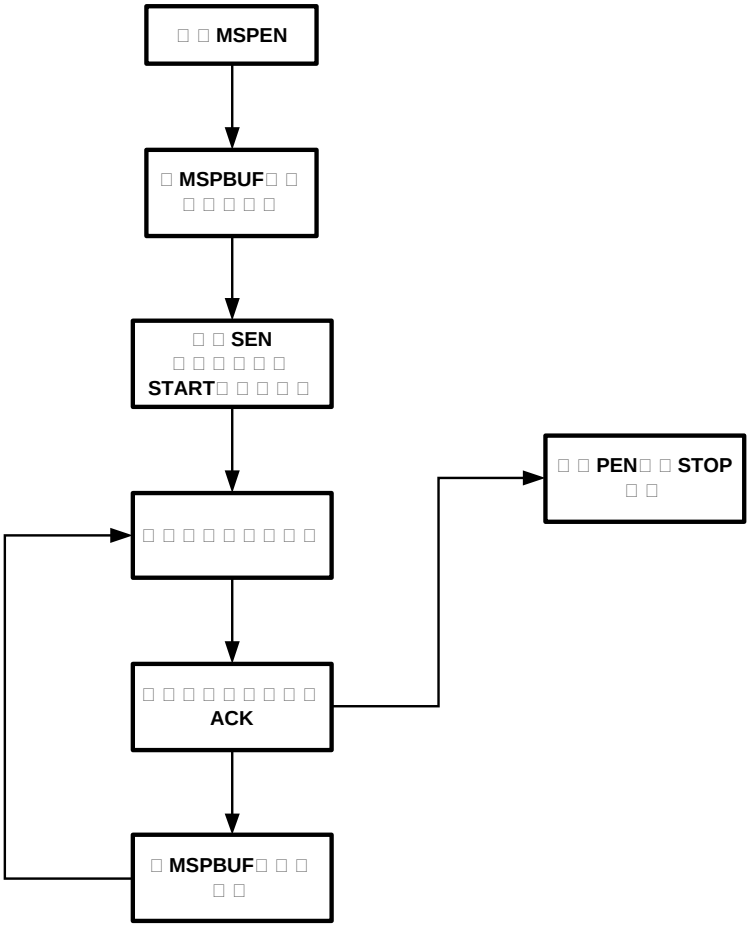


图 19-8I2C 软件发送数据流图

I2C主机对7位地址从机写入数据的波形示意图如下：

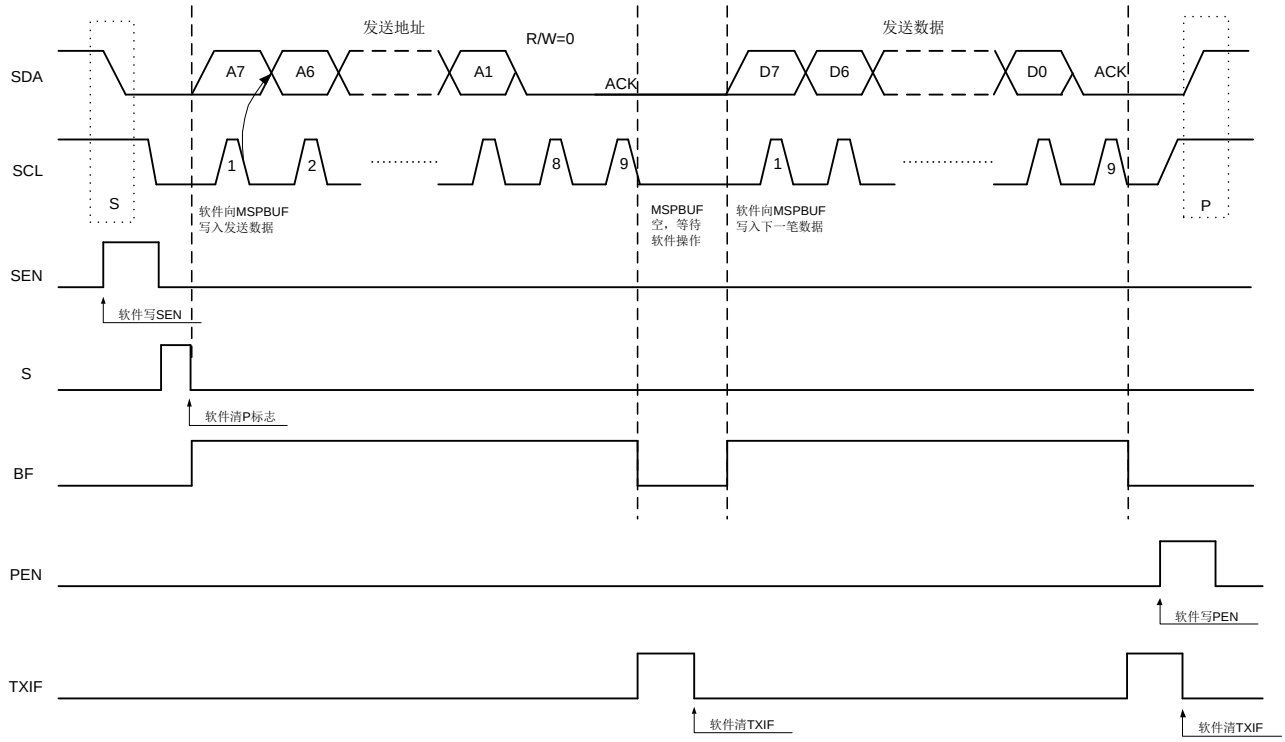


图 19-9I2C 主机对 7 位地址从机发送数据流图

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

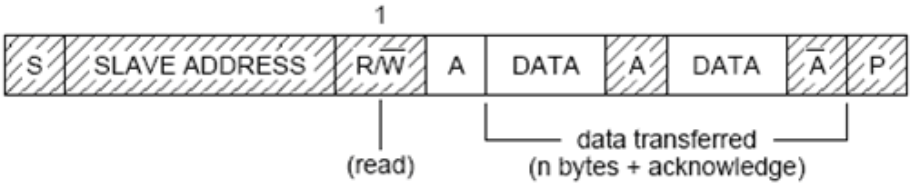


图 19-10 主机从 7 位地址从机读取数据时的帧格式

1、 主机发起 START 时序

2、 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 此时设置 MSPCON.RCEN 为 1，主机自动转为接收状态

4、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

5、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

6、 主机发送 STOP 时序终止读取

软件启动 I2C 接收的操作流程如下图：

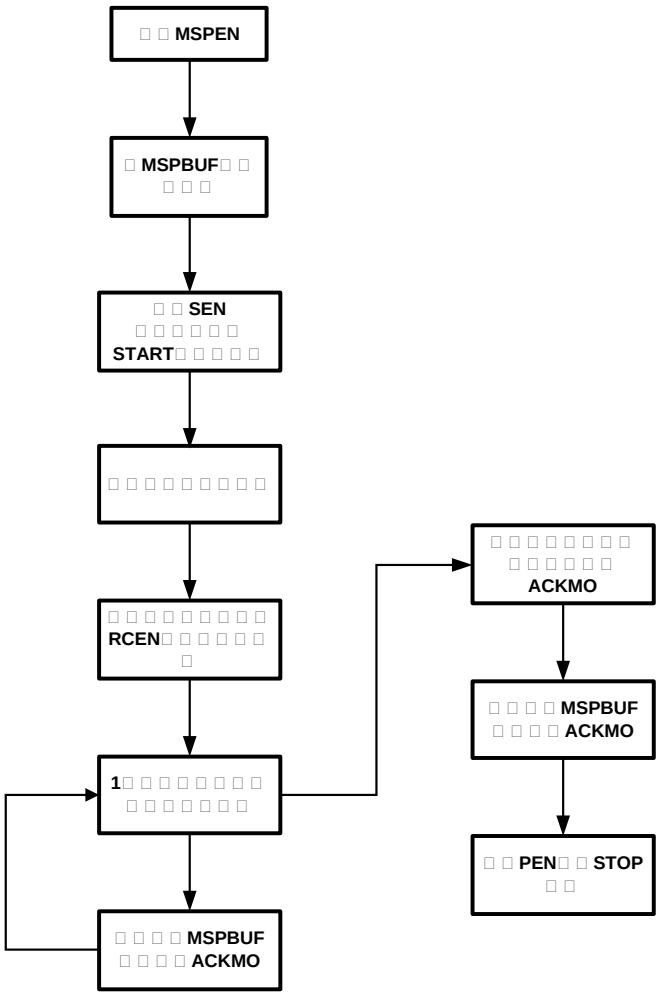


图 19-11I2C 软件发送数据流图

主机每次接收完从机发送的数据后，根据ACKMO寄存器回发响应。ACKMO复位值为0，即默认状态下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKMO寄存器改写为1。ACKMO为1的情况下，主机在发送完响应后会自动清零ACKMO。

I2C主机从7位地址从机读出数据的波形示意图如下：

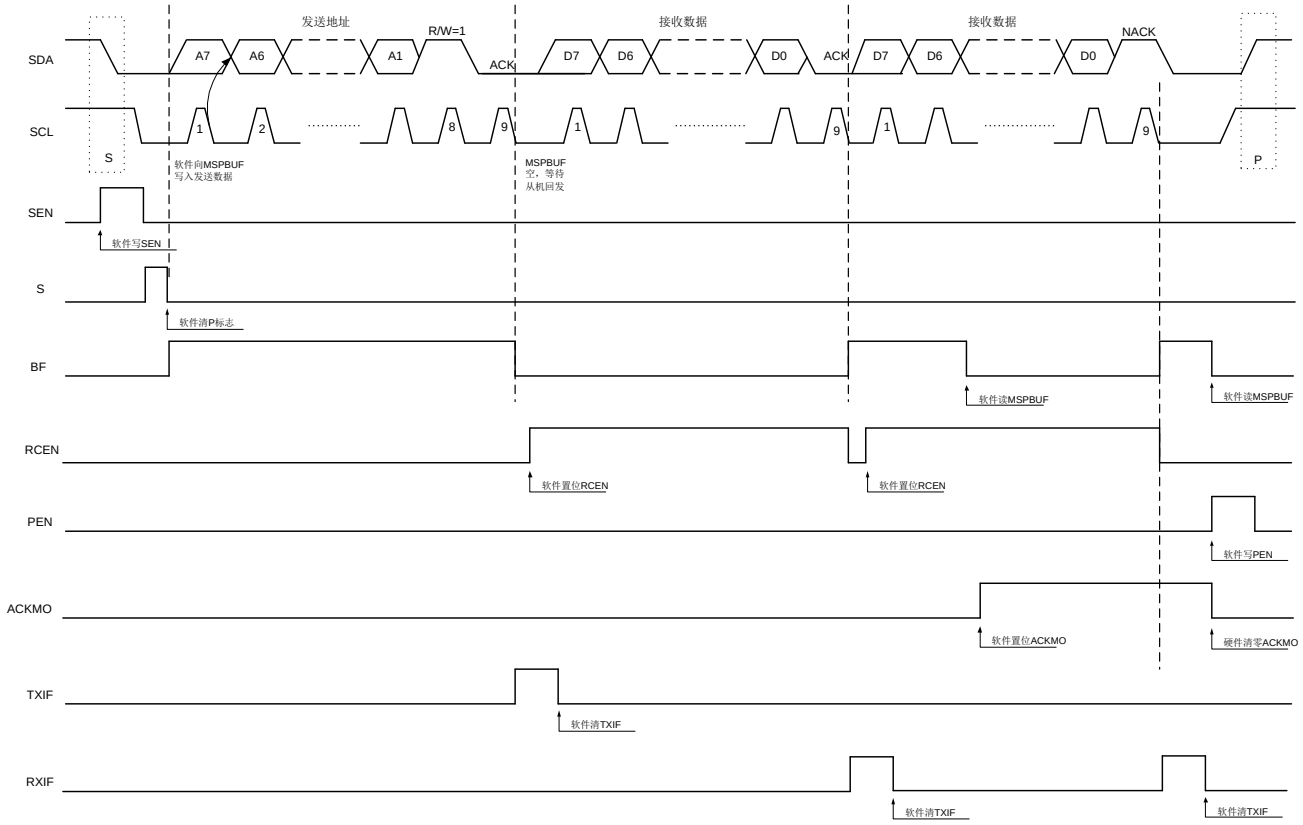


图 19-12I2C 从 7 位地址从机读取数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

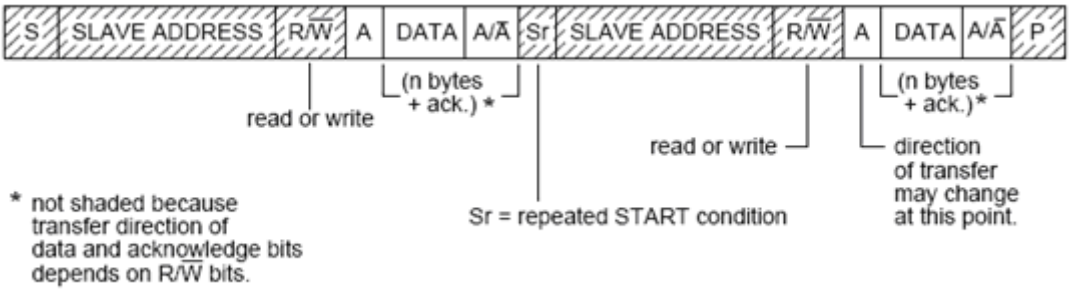


图 19-13 双向数据通信帧格式

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和从机地址字节来修改传输方向。

19.9.2 10bit 寻址

在10bit寻址时，主机发送的第一个字节包含部分从机地址（11110_A9_A8）和传输方向位（$R/\overline{W}$），第二个字节包含剩余从机地址（A7~A0）。两个字节地址发送完成后，再进行数据传输。

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

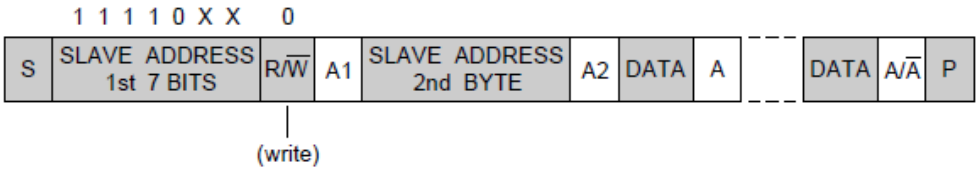


图 19-14 10bit 寻址，主机向从机写入数据

1、主机发起 START 时序

2、主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

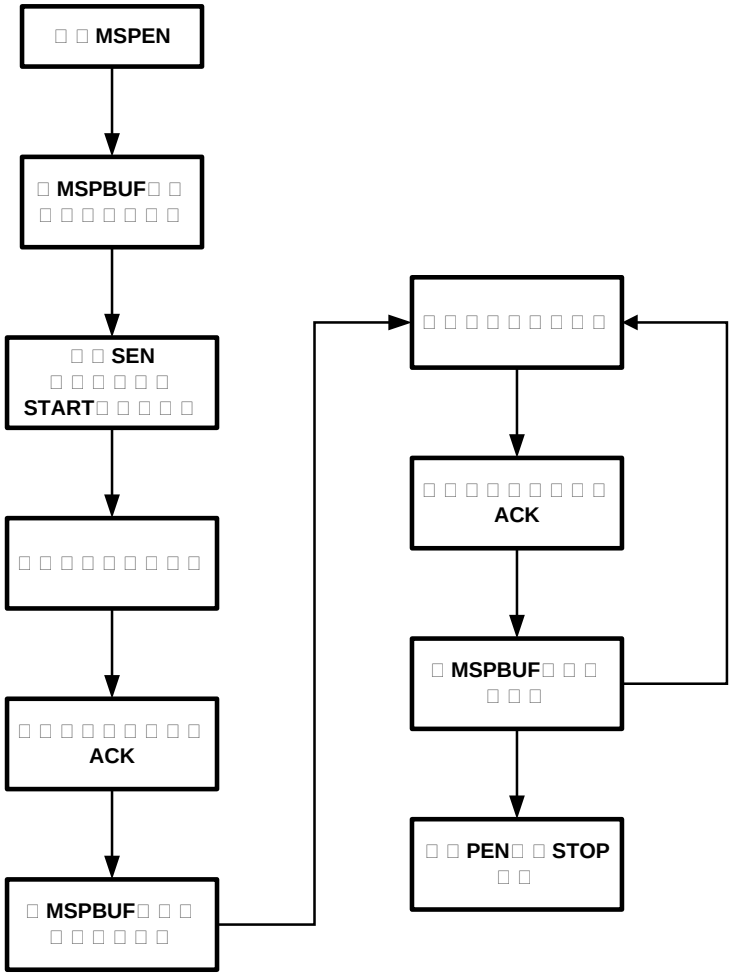


图 19-15I2C 软件发送数据流图

主机从从机读取数据

典型的10bit寻址，主机从从机读取数据的数据流图如下图所示。

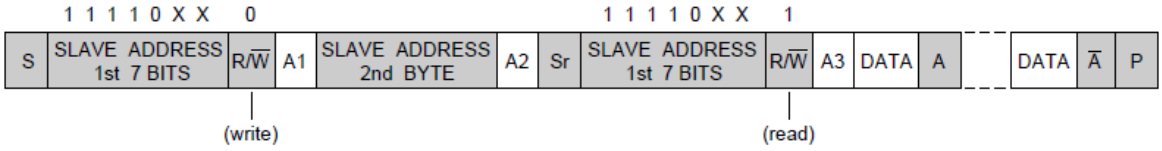


图 19-16 10bit 寻址，主机从从机读取数据

1、 主机发起 START 时序

2、 主机发送第一字节从机地址，包含 5 位前导码 11110、2 位从机地址最高位和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 ReSTART 时序

5、 主机再次发送第一字节从机地址，将 R/W 为改为 0

6、 此时设置 MSPCON.RCEN 为 1，主机转为接收状态

7、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

8、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

9、 主机发送 STOP 时序终止读取

软件启动 I2C 接收的操作流程如下图：

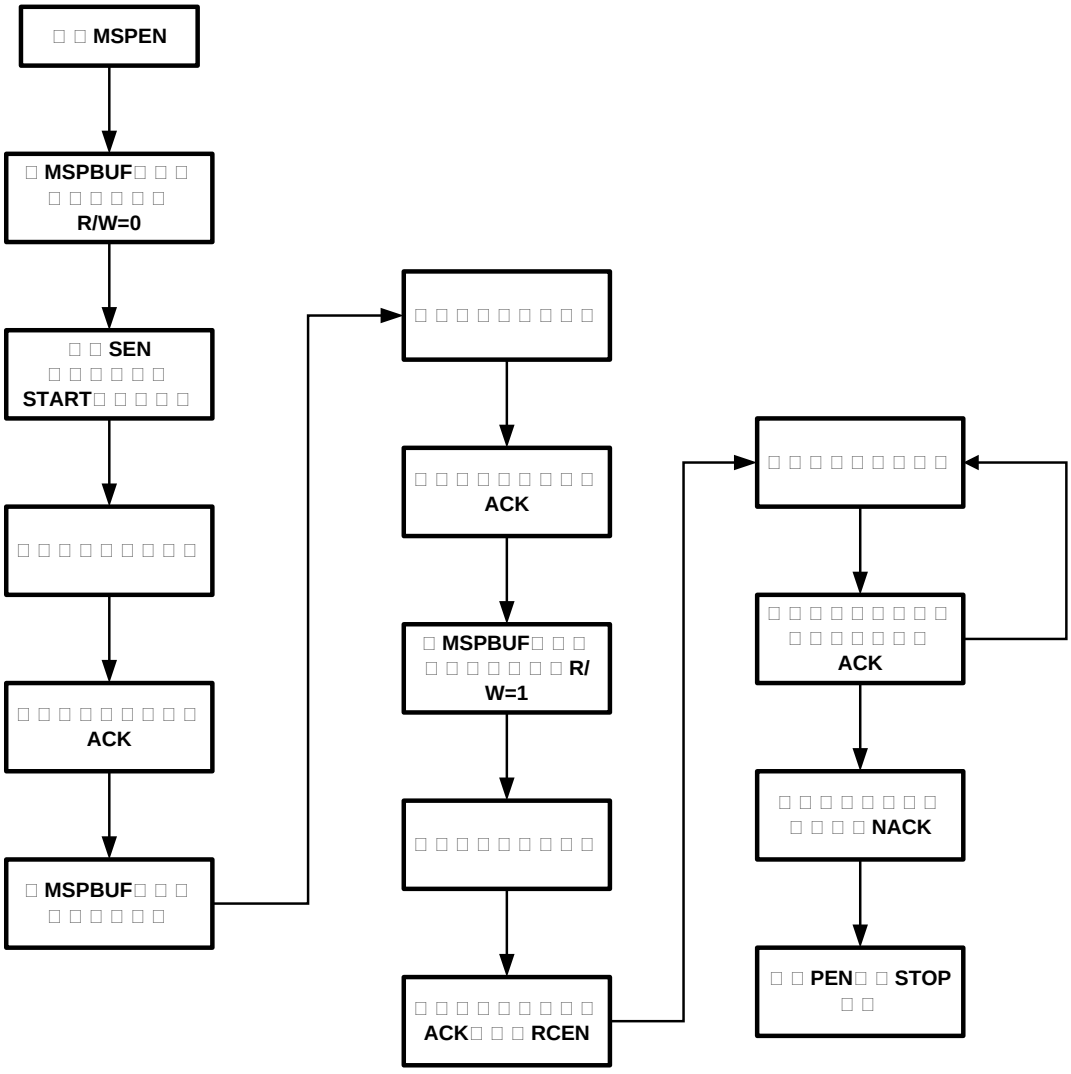


图 19-17I2C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以

有数据读取。

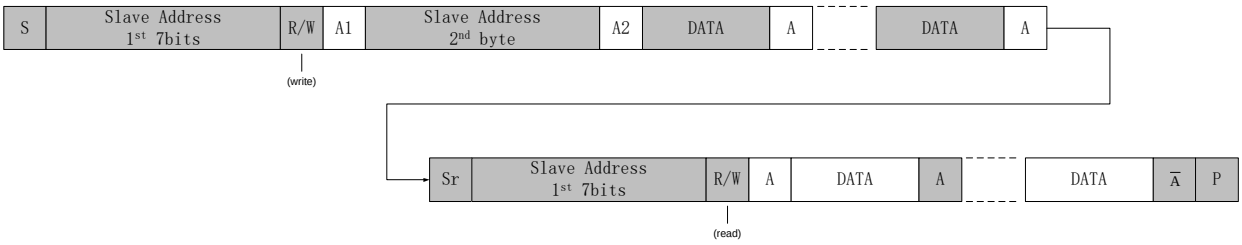


图 19-18I2C 软件发送数据流图

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送ReSTART时序和1st从机地址字节来修改传输方向。

19.9.3 DMA

I2C 主机支持 DMA，需要注意的是，必须在 I2C 模块的总线时钟（APBCLK）使能的情况下，才能使用 DMA 功能。

主机使用DMA向从机写入数据

主机使用 DMA 发送数据时，包括从机地址字节和发送数据在内的所有数据都需要事先写入 RAM 中，并通过 DMA 请求发送出去。软件应事先将目标 DMA 通道配置为 I2C_TX。

在 DMAEN=1 的情况下，MSPEN 置位，如果数据缓存寄存器 MSPBUF 为空，I2C 模块将产生 DMA 请求，DMA 模块响应请求后将 RAM 中的待发数据写入 MSPBUF，同时 I2C 模块自动置位 SEN 产生 START 时序，开始数据发送（第一个字节是从机地址）。DMA 发送模式下，I2C 并不检查发送数据的合法性，软件必须保证 RAM 中的数据是正确的。

每个字节发送完成后，I2C 检查从机 ACK，如果 ACK 正确则产生新的 DMA 请求，如果收到 NACK 则产生 NACK 中断，并不再产生 DMA 请求。

当DMA完成指定长度的数据发送后，产生DMA传输完成中断，此时可以又软件置位PEN产生STOP时序，也可以由 I2C 硬件根据 DMA 传输完成信号自动置位 PEN 产生 STOP 时序。可以通过设置 AUTOEND 寄存器来选择所需的策略。

主机使用 DMA 进行发送的流程如下图：

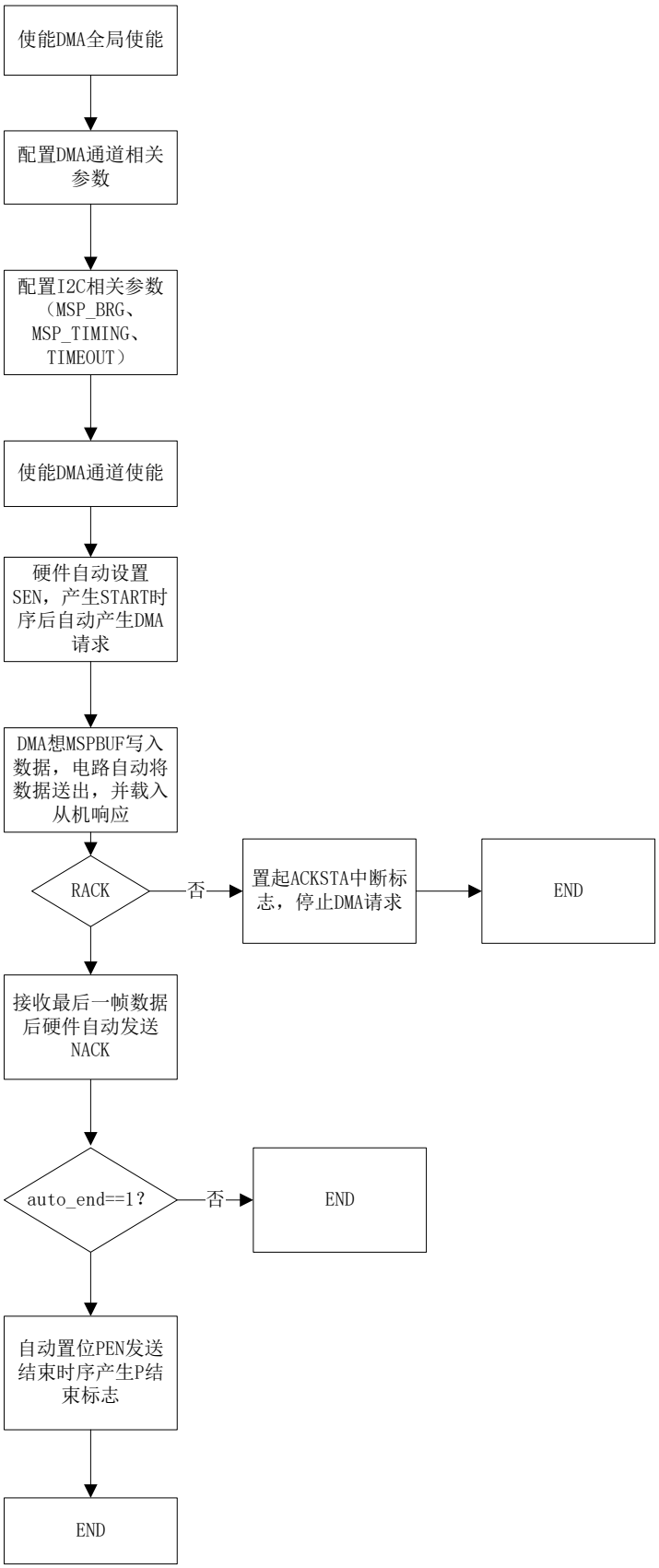


图 19-19 I2C 主机 DMA 发送流程图

主机使用DMA从从机读取数据

这种场景下，从机寻址字节必须由软件发送。软件应事先将目标 DMA 通道配置为 I2C_RX。

软件首先发送完从机地址后，设置 MSP_DMAEN=1，然后使能对应的 DMA 通道，I2C 自动进入接收模式，并在每个字节接收完成后产生 DMA 请求，通知 DMA 来读取 MSPBUF 内容，同时向从机回发 ACK。

当 DMA 传输达到指定长度后，DMA 的传输完成标志将通知 I2C 回发 NACK。随后根据 AUTOEND 寄存器配置，可以由软件或硬件置位 PEN 产生 STOP 时序。

注意：当I2C主机通过DMA进行数据接收时，在不同AUTOEND配置和相同DMA传输长度（CHxTSIZE）配置下，DMA接收字节数会有差别。当AUTOEND=0时，接收字节数为CHxTSIZE+1；当AUTOEND=1时，接收字节数为CHxTSIZE。

主机使用DMA进行接收的流程如下图：

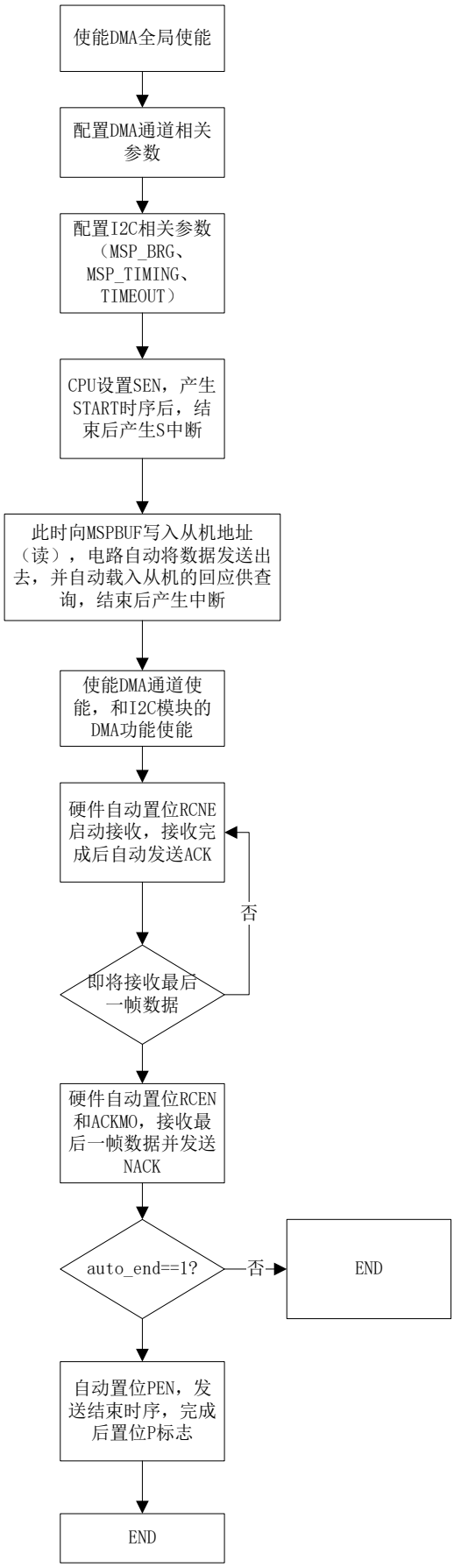


图 19-20 I2C 主机 DMA 接收流程图

19.9.4 SCL 延展（Slave Clock Stretching）

I2C 总线运行低速从机通过拉低 SCL 的方式通知主机暂停数据通信。I2C 主机必须支持这一特性，因此在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，需要自动检查总线上 SCL 的实际电平，如果不是高电平，意味着从机正在进行 SCL 延展，主机会持续监控 SCL 电平，直到 SCL 为高，才开始后续操作。

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。

19.9.5 超时机制

I2C主机还实现了超时机制，即发现从机长时间拉低SCL导致总线无法通信的情况下，产生超时报警中断并返回IDLE状态。

当主机检测到SCL延展，其内部定时器开始计时，主机设定的SCL延展超时的时长最长是4096个SCL周期，假设波特率为100K，则超时周期大约是40ms，如果波特率是400K，则超时周期大约是10ms。通过12bit的TIMEOUT寄存器，软件可以设置超时周期。软件必须在MSPEN为0的情况下设置TIMEOUT寄存器，此寄存复位值为0xFFF，即表示最长4096*T_{SCL}的超时周期，当检测到SCL延展后，TIMEOUT寄存器开始向下递减，当计数到0时，计数停止，TIMEOUT寄存器被复位到0xFFF，同时触发超时中断。因此通过修改TIMEOUT初始值，可以设定超时周期。

$T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

当发生TIMEOUT中断时，建议软件复位I2C模块。

此功能可以被关闭，如果关闭硬件超时，软件也可以通过定时器结合SCL引脚状态判断来自行实现任意长度的超时判决。

19.9.6 可编程时序

I2C 模块的主机模式提供了灵活的时序编程特性，允许用户定义 SCL 时钟的低电平宽度、高电平宽度，SDA 数据的建立和保持时间。

通过 MSPBRG 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立保持时间长度。

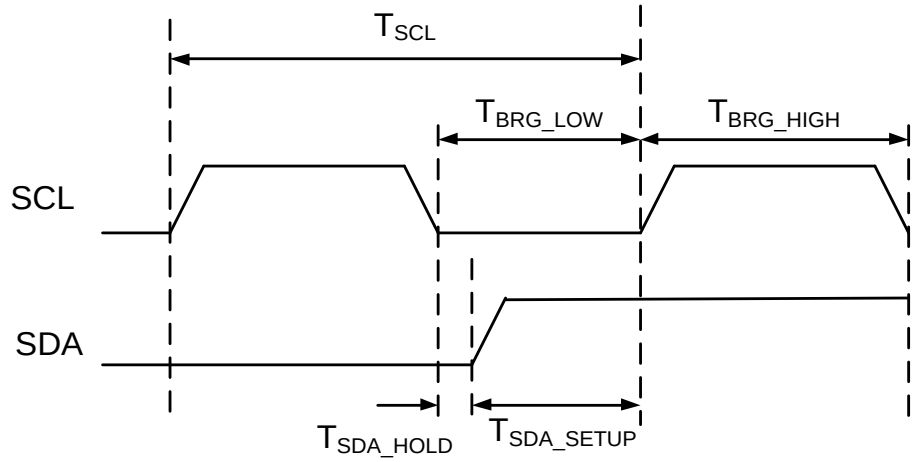


图 19-21 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG\_LOW} + T_{BRG\_HIGH}$

$T_{SDA\_SETUP} = T_{BRG\_LOW} - T_{SDA\_HOLD}$

注意，应用中对MSPBGRH、MSPBRGL和SDAHD寄存器的配置必须满足以下要求，如果违反这些要求将导致异常的总线时序：

MSPBRGH>= 2

MSPBRGL >=2

MSPBRGL-1>= SDAHD >=1

TIMEOUT >= 1

19.10 I²C 从机功能

I2C 从机的工作不需要系统时钟，因此可以在芯片休眠的状态下进行数据收发和唤醒。

从机接收完 1 字节数据后，产生中断通知 CPU 处理数据，在 CPU 取走数据前硬件可以将 SCL 拉低（软件控制使能），通知发送方正忙，发送方应暂停发送直到 SCL 放开。若接收方无法响应 ACK，发送方检测 ACK 失败后应发送 P 终止通信或者发送 Sr 开始新的通信。

从机发送完 1 字节数据后，产生中断通知 CPU，硬件拉低 SCL 令主机等待，CPU 响应中断并准备好下一字节数据后再放开 SCL，主机继续发送 SCL 使从机继续数据发送。

19.10.1 从机寻址

根据 SSPCON.A10EN 寄存器状态，从机可以支持 7bit 或者 10bit 寻址过程。从机地址由 SLAVE_ADDR 寄存器定义。

对于 10bit 从机地址应用，即 SSPCON.A10EN=1 的情况下，要求 1st 字节必须以 11110 开头，否则将触发 ADDR_ERROR 错误标志。而在 SSPCON.A10EN=0 的情况下，如果从机收到了 11110 开头的地址字节，也会置位 ADDR_ERROR 错误标志。

19.10.2 从机发送数据

推荐操作流程：

- 从机接收到地址字节（R/W=1），回发ACK，产生地址匹配中断
- 由于R/W=1，硬件自动进行SCL延展，从机进入发送状态
- 软件响应中断，查询R/W标志，确认是从机发送
- 软件将待发送数据写入SSPBUF
- 硬件自动释放SCL
- 新的SCL到来，SSPBUF移位输出到SDA总线
- 接收ACK并产生发送完成中断
- 重复数据发送过程直到接收到STOP时序，或接收到主机NACK

下图是一个典型的从机数据发送波形示意图：

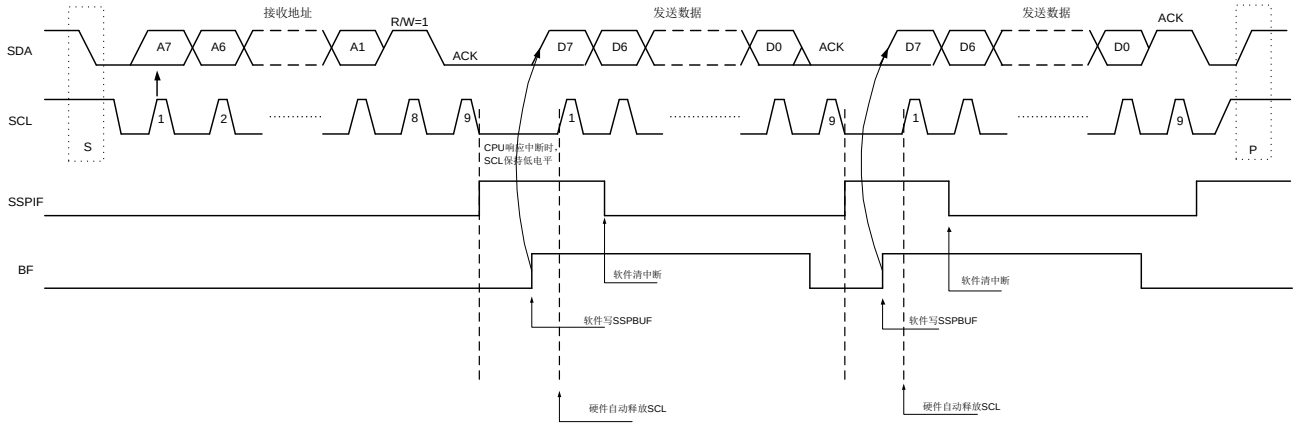


图 19-22 从机数据发送波形

在从机发送流程中，当从机收到正确地址时，ADM标志置位，地址字节不会被写入SSPBUF，因此BF标志不会置位。硬件自动拉低SCL信号等待软件写入SSPBUF，当软件写SSPBUF后BF置位，同时硬件释放SCL。

19.10.3 从机接收数据

推荐操作流程：

- 从机接收到地址字节（R/W=0），回发ACK，产生地址匹配中断
- 由于R/W=0，硬件自动进行SCL延展，从机保持接收状态
- 软件响应中断，查询R/W标志，确认是从机接收
- 软件读SSPBUF硬件自动释放SCL，开始接收数据
- 主机数据字节到来，字节接收完成后硬件置位BF标志
- 从机回发ACK，并产生接收完成中断
- 硬件自动进行SCL延展（SCLSEN=1）
- 软件响应中断，读取SSPBUF，硬件自动清零BF标志
- 硬件自动释放SCL
- 重复数据接收过程直到接收到STOP时序，或者软件将ACKEN置0

下图是一个典型的从机数据接收波形示意图（SCLSEN=1）：

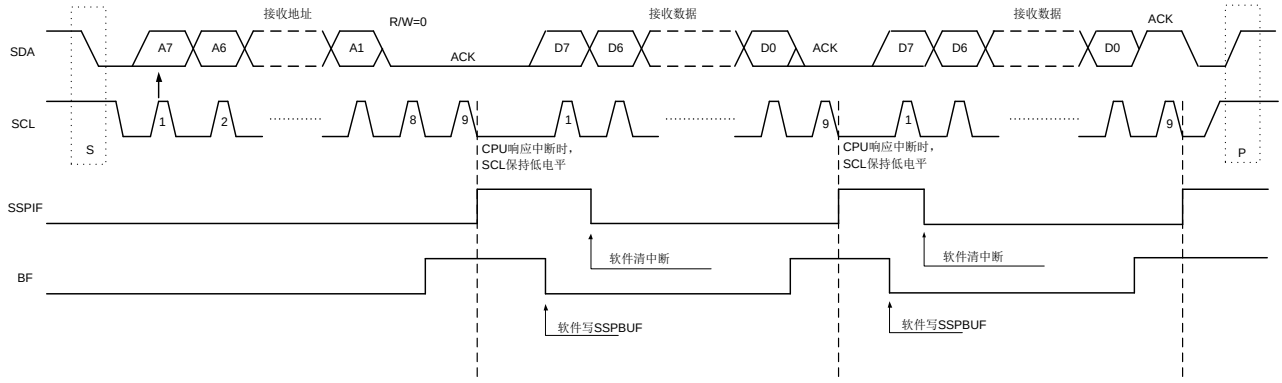


图19-23从机数据接收波形

从机接收过程中，从机首先收到地址字节，地址匹配的情况下，ADM标志置位，地址字节将被写入SSPBUF并置位BF标志，然后硬件拉低SCL。当软件读取SSPBUF后，BF标志自动清零，硬件释放SCL，可以进行后续数据接收。

注意：从机接收流程中地址字节会被写入SSPBUF并导致BF置位，软件需要读取SSPBUF来清零BF并释放SCL。而从机发送流程中地址字节不会被写入SSPBUF因此也不会置位BF标志。

从机接收数据可以被动结束通信或主动结束通信。

如果主机主动下发STOP，则从机被动结束本次通信。或者，软件在中断处理程序中将ACKEN寄存器清零，则从机在接收完下一个字节后，将回发NACK，主机接收NACK后将下发STOP结束本次通信。

从机SCL延展

I2C从机默认使能SCL延展（slave clock stretching），但是软件可以关闭这个功能（SCLSEN寄存器）以适应不支持从机SCL延展的主机。

当SCL延展使能的情况下，数据接收完成后，软件只有在SCL延展期间读取接收缓冲区时，才能清零BF标志。如果接收中出现了数据溢出，SSPOV标志置位，此时硬件回发NACK，并且SCL不再被延展，以便主机下发STOP；SSPOV置位的情况下，建议软件等待STOP标志置位，再读取接收缓冲区清零BF标志。

接收数据溢出

当从机接收缓冲区满（BF=1）时，如果又收到新的数据，则发生接收溢出，SSPOV标志置位。接收缓冲区中的老数据将被新的数据覆盖。只有在从机关闭了SCL延展功能的情况下，才有可能发生接收数据溢出。

下图是SCLSEN=0情况下发生数据接收溢出的示意图：

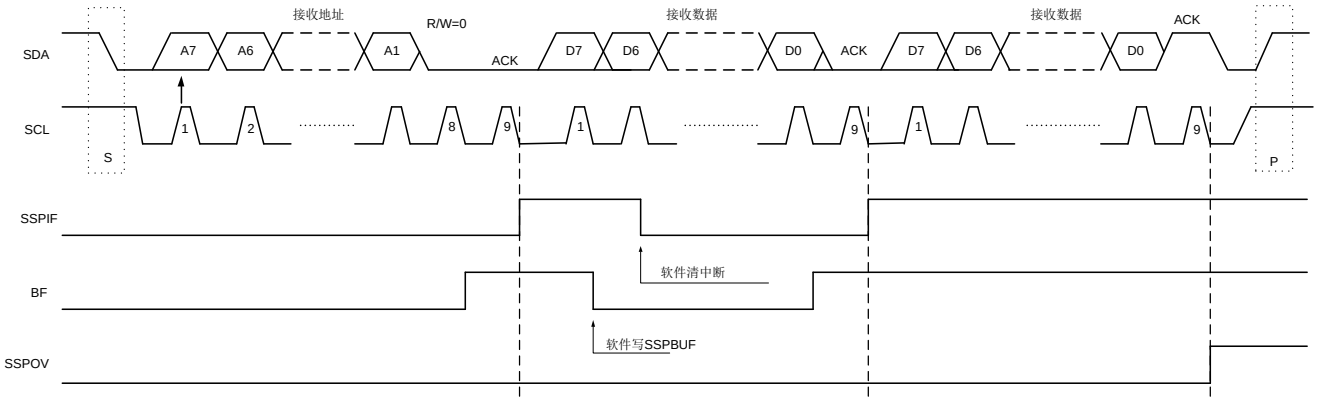


图19-24从机数据接收波形（SCLSEN=0，接收溢出）

19.10.4 从机低功耗接收唤醒

由于I2C从机不需要系统时钟即可工作，因此可以在休眠模式下接收数据并唤醒MCU。

I2C从机支持START时序唤醒、地址匹配唤醒和数据接收完成唤醒。

软件设置流程：

- 关闭I2C主机
- 设置从机地址
- 根据所需的唤醒事件，设置SE、ADME或BFE中断使能
- 设置对应的GPIO为I2C功能
- 置位SSPEN，启动I2C从机
- 进入休眠模式等待数据接收
- 当唤醒事件到来后，软件查询唤醒源，处理I2C数据传输

19.10.5 DMA

I2C 从机支持 DMA 操作，需要注意的是，必须在 I2C 的总线时钟（APBCLK）使能的情况下才能进行 DMA 操作。总线时钟被用于产生 DMA 请求并接收 DMA 应答。

从机使用DMA接收数据

当 I2C 从机接收到正确的地址后，产生 ADM 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 0 表示主机准备向从机写入数据。此时软件可以配置特定 DMA 通道为 I2C_RX，并使能 I2C 从机的 DMAEN；随后每次从机完成一个字节的接收，将产生 DMA 请求，通知 DMA 来读取 SSPBUF。

结束 DMA 从机接收有两种可能：

1）数据传输长度还未达到 DMA 长度配置，主机就下发了 STOP 时序，软件应响应 STOP 中断并主动处理这种情况；

2）数据传输长度达到 DMA 长度配置，但是由于 DMA 请求是在从机回发 ACK 后产生，所以软件应响应 DMA 传输完成中断，并将 ACKEN 清零，这样从机会在接收完下一个字节后，回发 NACK，结束本次通信。

从机使用 DMA 进行接收的流程如下图：

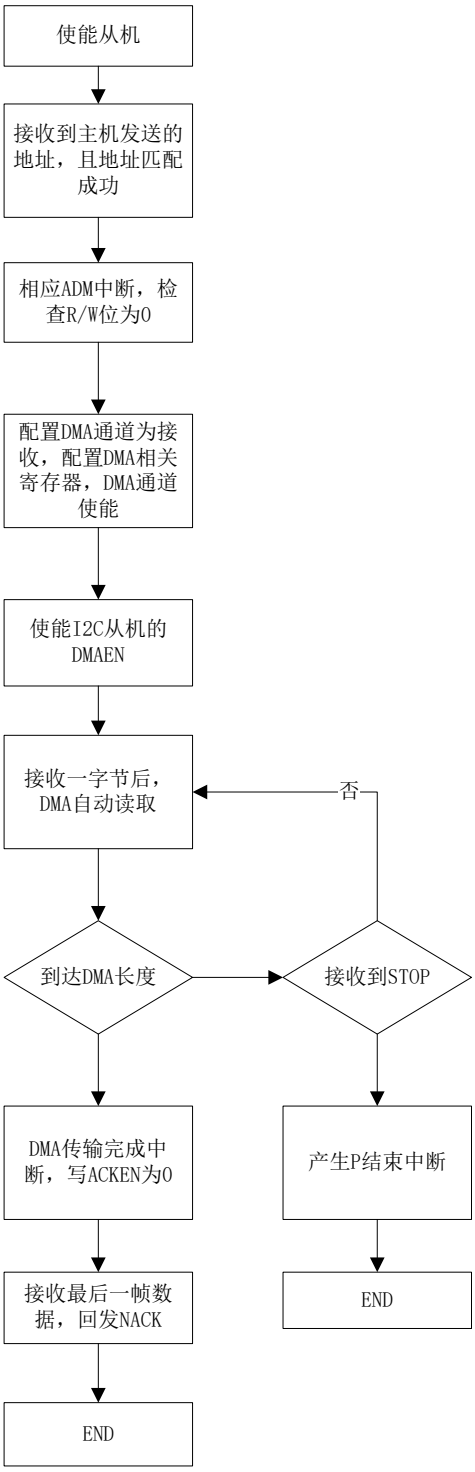


图 19-25 I2C 从机 DMA 接收流程图

从机使用DMA发送数据

当 I2C 从机接收到正确的地址后，产生 ADM 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 1 表示主机准备从从机读出数据。此时软件需要先读取 SSPBUF 清除 BF 标志，然后配置特定 DMA 通道为 I2C_TX，并使能 I2C 从机的 DMAEN；随后当从机数据缓存 SSPBUF 为空时，将

产生 DMA 请求，通知 DMA 写入 SSPBUF。

只有主机回发 NACK 才能结束读取操作。当读取数据长度大于 DMA 设置的传输长度时，由于 DMA 不再响应 I2C 请求，从机将一直拉低 SCL，直到软件关闭 I2C 从机模块。

从机使用 DMA 进行发送的流程如下图：

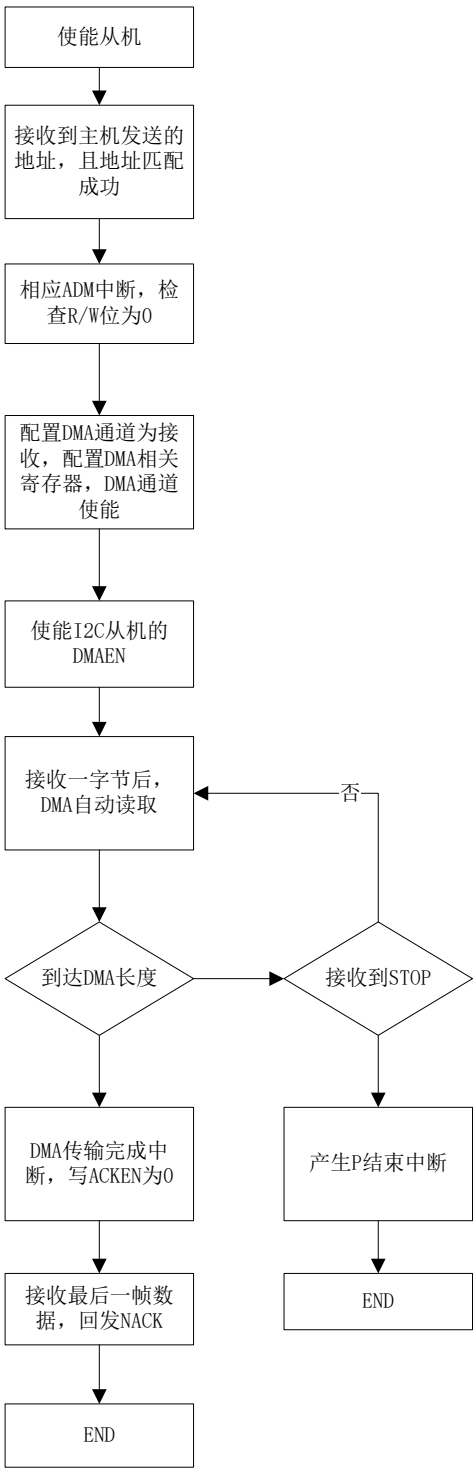


图 19-26 I2C 从机 DMA 发送流程图

19.10.6 从机时序

由于从机的数据收发只使用 SCL 进行，因此需要一些模拟延迟来实现 SDA 的数据建立和保持时间控制，而 SCL 的时序完全由主机控制。

从机时序控制如下图所示。根据 I2C 协议要求，SDA 相对于 SCL 下降沿的数据保持时间最小为 0ns，即从机使用 SCL 下降沿发送数据即可满足要求。但是考虑到总线上 SCL 波形的实际下降时间，为了更好的覆盖保持时间要求，在 SDA 输出上额外加入大于 300ns 的 RC 延迟。这个延迟仅需施加在 I2C 从机的 SDA 输出上（SSP_SDAO）

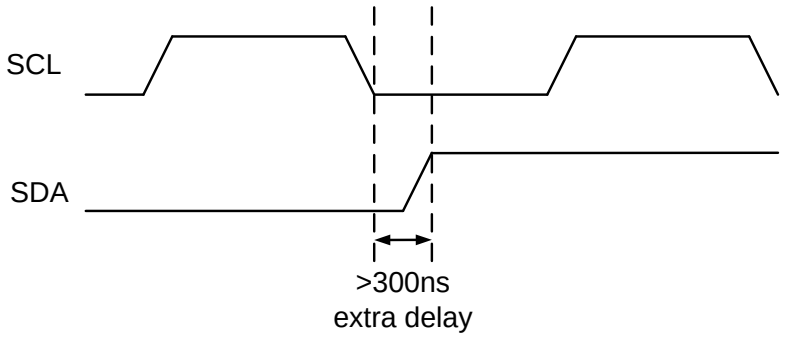


图 19-27 SDA 输出延迟波形

19.11 寄存器

offset 地址	名称	符号
I2C(模块起始地址:0x40012400)		
0x00000000	I2C 主机配置寄存器 （I2C Master Config Register）	I2C_MSPCFGR
0x00000004	I2C 主机控制寄存器 （I2C Master Control Register）	I2C_MSPCR
0x00000008	I2C 主机中断使能寄存器 （I2C Master Intterupt Enable Register）	I2C_MSPIER
0x0000000C	I2C 主机中断标志寄存器 （I2C Master Interrupt Status Register）	I2C_MSPISR
0x00000010	I2C 主机状态寄存器 （I2C Master Status Register）	I2C_MSPSR
0x00000014	I2C 主机波特率寄存器 （I2C Master Baud rate Generator Register）	I2C_MSPBGR
0x00000018	I2C 主机收发缓存寄存器 （I2C Master transfer Buffer）	I2C_MSPBUF
0x0000001C	I2C 主机时序控制寄存器 （I2C Master Timing Control Register）	I2C_MSPTCR
0x00000020	I2C 主机超时寄存器 （I2C Master Time-Out Register）	I2C_MSPTOR
0x00000024	I2C 从机控制寄存器 （I2C Slave Control Register）	I2C_SSPCR
0x00000028	I2C 从机中断使能寄存器 （I2C Slave Interrupt Enable Register）	I2C_SSPIER
0x0000002C	I2C 从机中断标志寄存器 （I2C Slave Interrupt Status Register）	I2C_SSPISR
0x00000030	I2C 从机状态寄存器 （I2C Slave Status Register）	I2C_SSPSR
0x00000034	I2C 从机收发缓存寄存器 （I2C Slave transfer Buffer）	I2C_SSPBUF
0x00000038	I2C 从机地址寄存器 （I2C Slave Address Register）	I2C_SSPADR

19.11.1 I2C 主机配置寄存器（I2C_MSPCFGR）

名称	I2C_MSPCFGR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						AUTOEND	MSP_DMAEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	TOEN	MSPEN
位权限	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:18	-	RFU：未实现，读为 0
17	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
16	MSP_DMAEN	主机 DMA 使能 (Master DMA enable) 0：关闭 DMA 功能 1：使能 DMA 功能
15:2	-	RFU：未实现，读为 0
1	TOEN	SCL 拉低超时使能 (Time Out enable) 1：使能超时功能，超时周期由 MSPTO 寄存器定义 0：关闭超时功能
0	MSPEN	I2C 主机模块使能控制位 (Master enable) 1：I2C 主机使能 0：I2C 主机禁止

19.11.2 I2C 主机控制寄存器（I2C_MSPCR）

名称	I2C_MSPCR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RCEN	PEN	RSEN	SEN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3	RCEN	主机接收模式下，接收使能位 (Receive enable) 1：主机接收使能 0：接收禁止 主机通信中，软件在发送完地址字节后，通过置位 RCEN 将传输方向切换为主机接收，然后可以接收来自于从机的数据。RCNE 在接收过程中保持为 1，直到软件置位 PEN 发送 STOP 时序。
2	PEN	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RSEN	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start

位号	助记符	功能描述
		Enable)
0	SEN	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

19.11.3 I2C 主机中断使能寄存器（I2C_MSPIER）

名称	I2C_MSPIER							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOLE	OVTE	SE	PE	NACKE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	WCOLE	WCOL 中断使能寄存器 (Write collision interrupt enable) 1：允许写冲突中断 0：禁止写冲突中断
5	OVTE	SCL 超时中断使能寄存器 (SCL overtime enable) 1：允许超时中断 0：禁止超时中断
4	SE	START 时序中断使能寄存器 (Start interrupt enable) 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	STOP 时序中断使能寄存器 (Stop interrupt enable) 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
1	TXIE	I2C 主机发送完成中断使能(Trasnmit done interrupt enable) 1：允许发送完成中断 0：禁止发送完成中断
0	RXIE	I2C 主机接收完成中断使能(Receive done interrupt enable) 1：允许接收完成中断 0：禁止接收完成中断

19.11.4 I2C 主机中断标志寄存器（I2C_MSPISR）

名称	I2C_MSPISR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOL	OVT	S	P	ACKSTA	TXIF	RXIF
位权限	U-0	R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	WCOL	写冲突检测位，MCU 只能在完成 START 时序或发送完成一帧读写之后才能写 MSPBUF，否则发生写冲突；硬件置位，软件写 1 清零(Write Collision Interrupt Flag,write 1 to clear) 1：发送写冲突 0：未发生冲突
5	OVT	SCL 超时中断标志，仅在 TOEN 为 1 时工作(SCL OverTime Interrupt Flag) 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	START 时序发送完成中断标志，硬件置位，软件读取后清零(Start Interrupt flag)
3	P	STOP 时序发送完成中断标志，硬件置位，软件读取后清零(Stop interrupt flag)
2	ACKSTA	主控发送模式下，来自从机的回应信号；当主机发送后收到 NACK，此标志可以产生中断；硬件置位，软件写 1 清零。(Acknowledge Status Flag ,write 1 to clear) 1：从机回应 NACK 0：从机回应 ACK
1	TXIF	I2C 主机发送完成中断标志，硬件置位，软件写 1 清零(Trasnmit done interrupt flag,write 1 to clear) 此标志寄存器在主机接收完从机回发的 ACK 或 NACK 后置位。
0	RXIF	I2C 主机接收完成中断标志，硬件置位，软件写 1 清零(Receive done interrupt flag,write 1 to clear) 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

19.11.5 I2C 主机状态寄存器（I2C_MSPSR）

名称	I2C_MSPSR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BUSY	RW	-	BF	-	ACKMO
位权限	U-0		R-0	R-0	U-0	R-0	U-0	R/W-0

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5	BUSY	I2C 通信状态位 (Busy) 1：接口处于读写状态，正在进行数据传输， 0：已完成数据传输
4	RW	I2C 传输方向状态位 (Read/Write) 1：主机从从机读取数据 0：主机向从机写入数据
3	-	RFU：未实现，读为 0
2	BF	缓冲器满状态位 (Buffer Full) 接收： 1：接收完成，MSPBUF 满 0：接收未完成，MSPBUF 空 发送： 1：正在发送，MSPBUF 满 0：发送完成，MSPBUF 空
1	-	RFU：未实现，读为 0
0	ACKMO	主控接收模式下，主机回应信号的状态 (Ack Master output) 1：主机回发 NACK 0：主机回发 ACK ***注意：必须在 P 标志寄存器被清零的情况下，软件才能置位 ACKMO***

19.11.6 I2C 主机波特率寄存器（I2C_MSPBGR）

名称	I2C_MSPBGR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							MSPBRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSPBRGH[7:0]							
位权限	R/W-0001 0011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MSPBRGL[8]
位权限	U-0							R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBRGL[7:0]							
位权限	R/W-0001 0011							

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24:16	MSPBRGH	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 (Master SCL High level length)
15:9	-	RFU：未实现，读为 0
8:0	MSPBRGL	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 (Master SCL Low level length)

19.11.7 I2C 主机收发缓存寄存器（I2C_MSPBUF）

名称	I2C_MSPBUF							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBUF							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	MSPBUF	MSPBUF[7:0]：数据的读写通过对 MSPBUF 的操作完成。发送时，对 MSPBUF 执行写操作，同时也载入数据收发移位寄存器(MSPSR)；接收时，MSPBUF 与 MSPSR 组成双缓冲结构，读出数据为 MSPBUF 的数据。接收完一个字节的数据，MSPSR 将数据载入 MSPBUF，同时置位 I2CIF。MSPSR 不是直接寄存器，没有物理地址。 (Master data Buffer)

19.11.8 I2C 主机时序控制寄存器（I2C_MSPTCR）

名称	I2C_MSPTCR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SDAHD[7:0]							
位权限	R/W-0000 1010							

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2C 工作时钟计数 (SDA hold delay) 注意：最小有效值为 1，最大有效值为 MSPBRGL

19.11.9 I2C 主机超时寄存器（I2C_MSPTOR）

名称	I2C_MSPTOR							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TIMEOUT[11:8]			
位权限	U-0				R/W-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUT[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	TIMEOUT	定义从机 SCL 低电平延展超时周期，软件可以在 MSPEN=0 的情况下改写 (SCL stretching Time Out) $T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

19.11.10 I2C 从机控制寄存器（I2C_SSPCR）

名称	I2C_SSPCR							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						SCLSEN	SSP_DMAEN
位权限	U-00000000						R/W-1	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ACKEN	SDAO_DLYEN	SCLI_ANFEN	A10EN	SSPEN
位权限	U-0			R/W-1	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	SCLSEN	I2C 从机时钟延展使能 (SCL Stretching Enable) 0：禁止 slave clock stretching 1：使能 slave clock stretching *注意：当从机使用 DMA 通信时，必须将 SCLCEN 置 1*
8	SSP_DMAEN	I2C 从机 DMA 使能 (Slave DMA enable) 1：使能 DMA 功能 0：关闭 DMA 功能
7:5	-	RFU：未实现，读为 0
4	ACKEN	ACK 使能位：(Slave Ack Enable) 1 = slave 接收完成后将回发 ACK 0 = slave 不回发 ACK
3	SDAO_DLYEN	SDA 从机输出延迟使能 (SDA output delay enable) 0：bypass 从机 SDA 输出延迟 1：使能从机 SDA 输出延迟
2	SCLI_ANFEN	SCL 从机输入模拟滤波使能 (SCL input analog filter enable) 0：bypass 模拟滤波 1：使能模拟滤波
1	A10EN	10 位地址使能：(10bit Slave address enable) 1 = slave 使用 10bit address 0 = slave 使用 7bit address
0	SSPEN	I2C 从机使能位 (Slave enable) 1：使能 I2C 从机 0：关闭 I2C 从机

19.11.11 I2C 从机中断使能寄存器（I2C_SSPIER）

名称	I2C_SSPIER
Offset	0x00000028

名称	I2C_SSPIER							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADEE	SE	PE	WCOLE	SSPOVE	ADME	TXIE	RXIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7	ADEE	从机地址错误中断使能，1 有效 (Address Error Enable)
6	SE	Start 中断使能，1 有效 (Start interrupt enable)
5	PE	Stop 中断使能，1 有效 (Stop interrupt enable)
4	WCOLE	WCOL 中断使能，1 有效 (Write Collision Enable)
3	SSPOVE	SSPOV 中断使能，1 有效 (Slave Buffer Overflow Enable)
2	ADME	从机地址匹配中断使能，1 有效 (Address Match Enable)
1	TXIE	发送完成中断使能，1 有效 (Transmit interrupt enable)
0	RXIE	接收完成中断使能，1 有效 (Receive interrupt enable)

19.11.12 I2C 从机中断标志寄存器（I2C_SSPISR）

名称	I2C_SSPISR							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADE	S	P	WCOL	SSPOV	ADM	TXIF	RXIF
位权限	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7	ADE	从机地址格式错误，硬件置位，软件写 1 清零 (Address Error flag,write 1 to clear) 在 7 位地址情况下收到 11110 开头的地址字节，或者在 10 位

位号	助记符	功能描述
		地址情况下第一个字节不以 11110 开头时，触发 ADEE
6	S	检测到 start 时序，硬件置位，软件读取后自动清零 (Start flag)
5	P	检测到 stop 时序，硬件置位，软件读取后自动清零 (Stop flag)
4	WCOL	写冲突标志，硬件置位，软件写 1 清零 (Write Collision flag, write 1 to clear) 1：在 BF=1 的情况下，软件向 SSPBUF 写入新的数据 0：无写入冲突 当 WCOL 发生时，新的数据将被丢弃
3	SSPOV	SSPBUF 溢出标志，硬件置位，软件写 1 清零 (Slave buffer overflow flag,write 1 to clear) 1：在 BF=1 的情况下，从机又接收到新的数据 0：没有接收溢出 如果从机使能 SCL 延展，不会出现接收数据溢出的情况；因此 SSPOV 只可能在 SCLSEN=0 的情况下被置位。
2	ADM	从机地址匹配标志，硬件置位，软件写 1 清零 (Address Matched flag ,write 1 to clear) 1：接收到的 7bit 或 10bit 地址与 SLAVE_ADDR 寄存器内容一致 0：接收到的地址与 SLAVE_ADDR 不一致
1	TXIF	I2C 从机发送完成标志，硬件置位，软件写 1 清零 (Transmit interrupt flag,write 1 to clear)
0	RXIF	I2C 从机接收完成标志，硬件置位，软件写 1 清零 (Receive interrupt flag，write 1 to clear)

19.11.13 I2C 从机状态寄存器（I2C_SSPSR）

名称	I2C_SSPSR							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				BUSY	RW	DA	BF
位权限	U-0				R-0	R-0	R-0	R-0

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3	BUSY	从机通信标志 (Busy) 1：从机数据收发中 0：从机空闲

位号	助记符	功能描述
2	RW	读写方向状态寄存器 (Read/Write) 1：slave 收到 R/W=1，slave 需要发送数据给 master 0：slave 处于接收数据状态
1	DA	data/address 帧指示 1：上一字节收到的是数据 0：上一字节收到的是地址
0	BF	从机数据缓冲区满标志 (Buffer Full flag) 1：SSPBUF 满 0：SSPBUF 空

19.11.14 I2C 从机收发缓存寄存器（I2C_SSPBUF）

名称	I2C_SSPBUF							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPBUF							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	SSPBUF	SSPBUF[7:0]：数据的读写通过对 SSPBUF 的操作完成。发送时，对 SSPBUF 执行写操作，同时也载入数据收发移位寄存器 (SSPSR)；接收时，SSPBUF 与 SSPSR 组成双缓冲结构，读出数据为 SSPBUF 的数据。接收完一个字节的数据，SSPSR 将数据载入 SSPBUF，同时置位 I2CIF。SSPSR 不是直接寄存器，没有物理地址 (Slave Buffer)

19.11.15 I2C 从机地址寄存器（I2C_SSPADR）

名称	I2C_SSPADR							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

名称	I2C_SSPADR							
位名	-						SSPADDR[9:8]	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPADDR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:0	SSPADDR	从机地址寄存器 (Slave Address) A10EN = 1 10 位地址都有效 A10EN = 0 仅低 7 位有效

20 UART

20.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 4路独立通道（UART0, UART1, UART4, UART5）
- 全双工通信口
- UART具有数据接收完成/接收错误中断，并提示错误类型
- 可配置数据长度，支持6、7、8、9bits
- 可配置的停止位-支持1个停止位或2个停止位
- 可配置为红外调制输出功能，且载波频率可设置，及载波占空比可设置
- 支持DMA
- 支持接收超时机制

20.2 结构框图

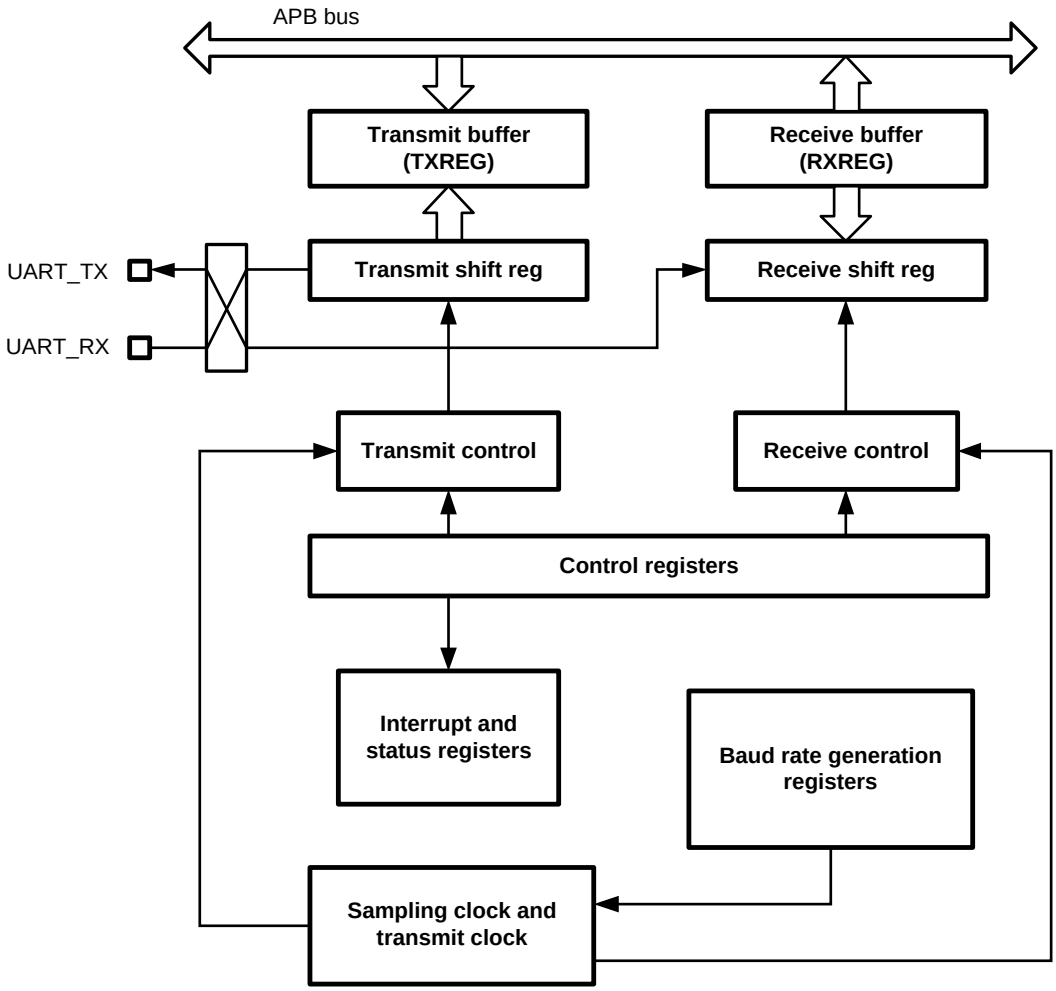


图 20-1 UART 接口时序

20.3 引脚定义

UART 模块使用 2 个引脚与外部器件通信，每个 UART 的收发信号可能被映射到不同的 GPIO 上。

下表为 FM33LC0x6 的引脚映射关系

引脚		UARTx	符号	功能
PA2	PA13	UART0	UART0_RX	数据接收
PA3	PA14		UART0_TX	数据发送
PC2	PB13	UART1	UART1_RX	数据接收
PC3	PB14		UART1_TX	数据发送
PA0	PB2	UART4	UART4_RX	数据接收
PA1	PB3		UART4_TX	数据发送
PC4	PD0	UART5	UART5_RX	数据接收
PC5	PD1		UART5_TX	数据发送

下表为 FM33LG0x6 的引脚映射关系

引脚		UARTx	符号	功能
PA2	PA13	UART0	UART0_RX	数据接收
PA3	PA14		UART0_TX	数据发送
PC2	PB13	UART1	UART1_RX	数据接收
PC3	PB14		UART1_TX	数据发送
PA0	PB2	UART4	UART4_RX	数据接收
PA1	PB3		UART4_TX	数据发送
PC4	PD0	UART5	UART5_RX	数据接收
PC5	PD1		UART5_TX	数据发送

表 20-1 UART 引脚列表

当 UART 功能被同时映射到多个引脚上时：

- PA2 和 PA13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC2 和 PB13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC4 和 PD0 同时配置为数字外设功能
  - 只有 PC4 上的 RX 信号会输入到模块内部
- PA0 和 PB2 同时配置为数字外设功能
  - 只有 PA0 上的 RX 信号会输入到模块内部
- UART 发送功能被同时映射到多个 GPIO 上时，这些引脚会同时发送数据

20.4 UART 类型区分

FM33L0xx集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART0/1	UART4/5	LPUART0/1
DMA 支持	Y	Y	Y
半双工/全双工	Y	Y	Y
红外发射	Y	Y	-
双时钟域（工作时钟独立于总线）	Y	-	Y
休眠唤醒	Y	-	Y
接收超时	Y	-	-
发送延迟	Y	-	-
数据长度	6、7、8、9bits		

表 20-2 UART 类型列表

20.5 UART 字符描述

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

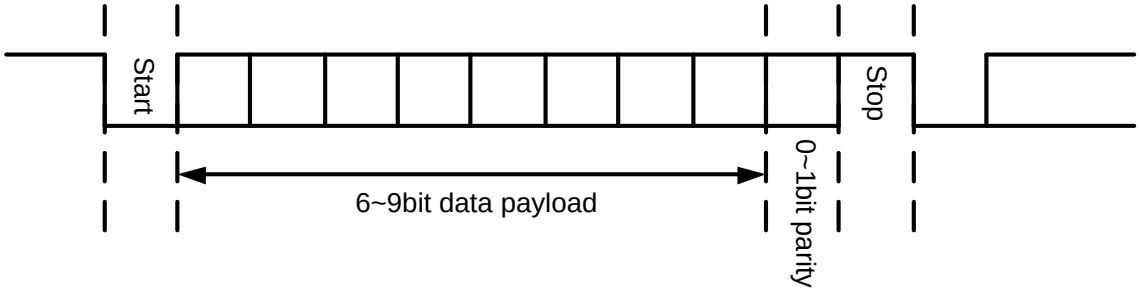


图 20-2 UART 字符描述

UART 支持多种帧格式，由 UARTxCSR.PDSEL 寄存器和 UARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 20-3 UART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

20.6 功能描述

20.6.1 时钟结构

UART0 和 UART1 采用了双时钟结构：

- 总线寄存器时钟用 pclk 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 UART 内部寄存器时，必须使能 pclk
- 数据收发时钟用 uclk 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK、RCMF，能够独立于 APBCLK 工作。必须使能 uclk 才能进行数据收发。

Pclk 和 uclk 的控制都在 CMU 模块内完成，进行 UART 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 UART0 和 UART1 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，UART 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 UART 以较高的波特率进行数据通信。

理论上 pclk 和 uclk 之间没有相对关系的约束，uclk 可以快于或者慢于 pclk。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

与 UART0 和 UART1 不同的是，UART4 和 UART5 采用单时钟结构，此时 uclk=pclk，UART 的数据收发时钟也是来源于 APBCLK 的。

20.6.2 位接收采样

UART 对接收数据进行波特率的 16 倍过采样，并在每个 bit 的中间位置进行三中取二的多数判决，以提高对信号噪声的抑制能力。

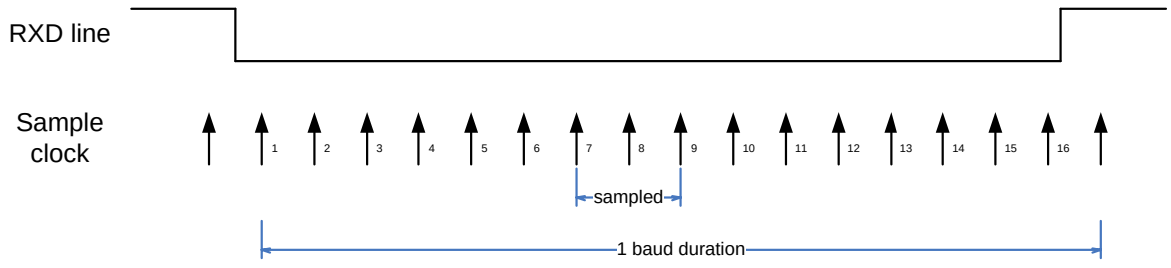


图 20-3 位接收采样

接收移位寄存器收到的 bit 位是多数判决的结果。例如三次采样结果是 001，则判决为 0；如果是 011，则判决为 1。

由于 UART 对输入信号进行 16 倍过采样，要求 SPBRG 配置不能小于 16，即 UART 工作时钟必须至少是波特率的 16 倍。

20.6.3 数据发送

在发送模式下，UART 的串行数据发送电路主要包括一个发送移位寄存器(TSR)，TSR 功能是将数据逐个移位送出。待发数据必须先写到发送缓冲区中。当软件置位 TXEN 寄存器后，如果发送缓冲区非空，UART 将缓冲区数据载入 TSR 并开始移位输出。

注：由于寄存器操作时钟和波特率时钟是异步关系，当发送开始时，需要等待波特率时钟到来，因此从 TXEN 置位到 UART 开始发送 Start 位之间，有最大 1 个 baud 的延迟。

TXBE 和 TXSE 是发送中断标志位，分别表示发送缓冲区空和 TSR 空，软件可以选择在合适的时间点产生发送完成中断。

一般情况下，一开始 TSR 寄存器是空的，数据的发送需先设定波特率 SPBRG，使能发送模块(设定 TXEN 为 1)，然后写入 TXBUF 寄存器开始发送。也可以在设定好波特率 SPBRG 后，先写入 TXBUF 寄存器，然后再设定 TXEN 使能发送模块来开始数据发送。如果在数据发送过程中将发送模块使能位 TXEN 清 0，那么数据发送工作就会被中断，发送模块也会被复位。

下图为UART异步发送的例子。这个示例中软件首先向TXBUF写入数据，然后通过置位TXEN启动发送。

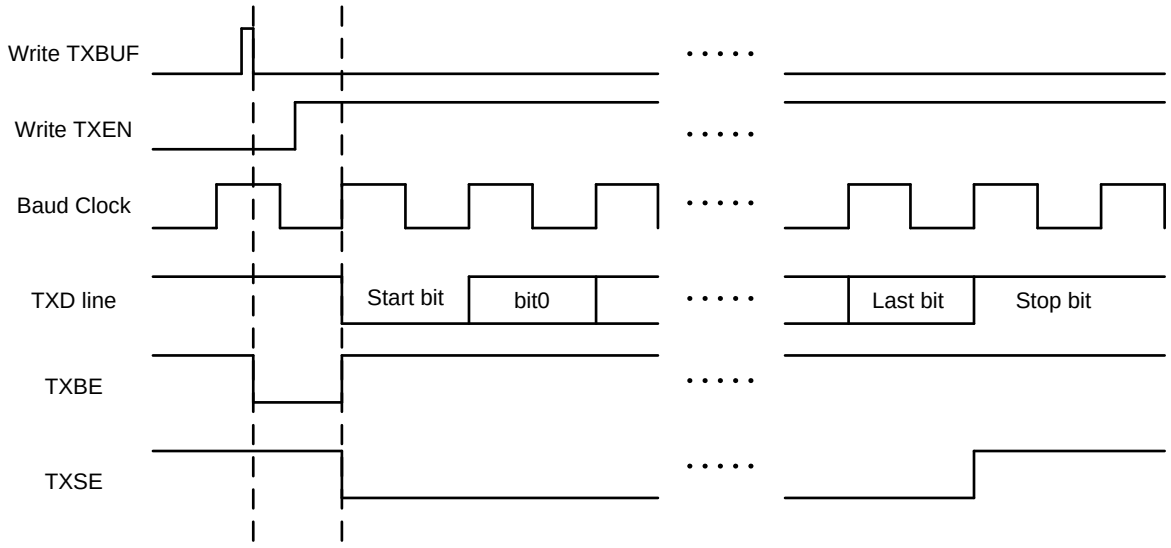


图 20-4 UART 异步发送波形 1

上图中推荐的操作步骤如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 TXSE_IE 或者 TXBE_IE
- 决定数据发送的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 如果希望发送的串行数据红外调制，向 IRCON 寄存器写入合适的值来获得相应的调制频率和占空比，并置位 TXIREN
- 将待发送的数据写入 TXBUF 寄存器（自动启动发送）
- 使能发送模块：置位 TXEN

软件也可以先置位TXEN再写入TXBUF，此时UART会在数据写入TXBUF后立刻开始发送流程。

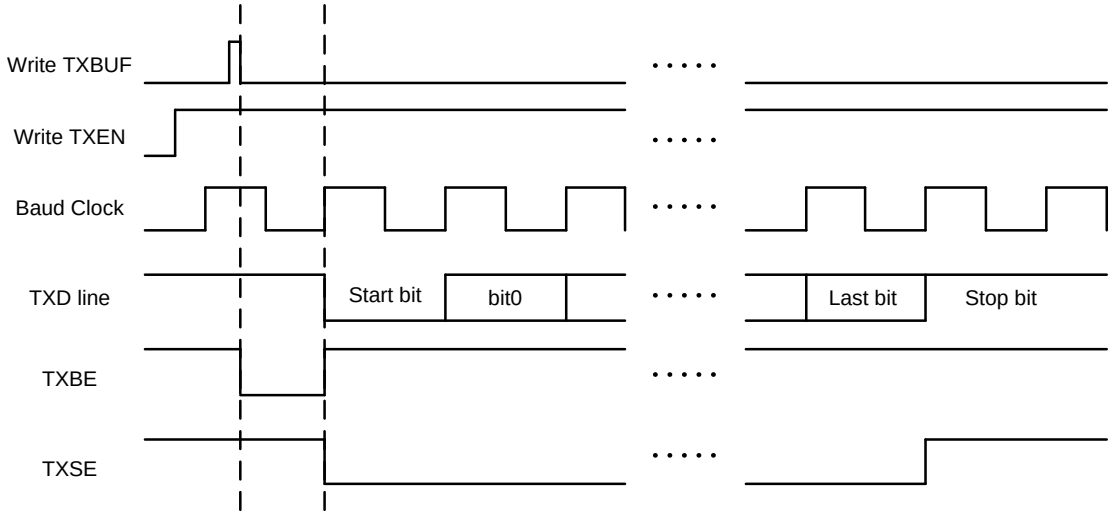


图 20-5 UART 异步发送波形 2

当TXBUF为空时，软件可以立即写入下一个待发送数据，以实现连续无间隔的数据发送。

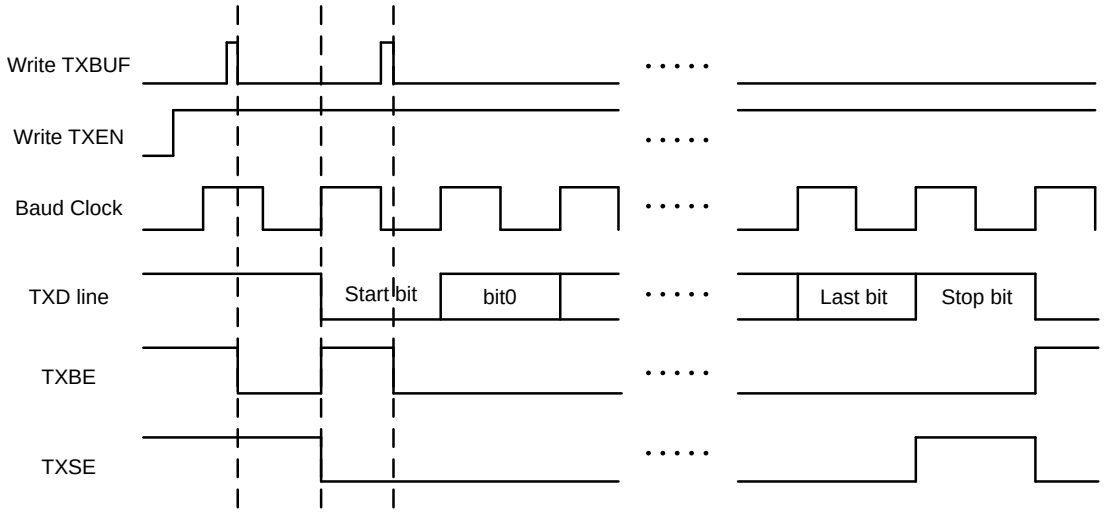


图 20-6 UART 异步发送波形 3

20.6.4 数据接收

UART 的串行数据接收电路主要包括一个接收移位寄存器(RSR)。当接收到停止位后，RSR 就把接收到的数据送入接收缓冲区(RXBUFFER)，传送完成后，在每次接收数据送入接收缓冲区后将中断标志 RXBF 置 1。当接收缓冲区已满时，RSR 接收到一帧数据后仍会将其写入接收缓冲区，即覆盖缓冲区中原有数据，并且再次置位 RXBF，同时发生接收溢出错误，OERR 被置 1；软件写 1 或者读取 RXBUF 都可以清除 OERR 标志。

接收过程中，如果没有检测到正确的停止位，则发生帧格式错，FERR 被置 1；如果发生奇偶校验错，标志位 PERR 被置 1。

推荐的异步接收操作如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 RXBF_IE
- 设置数据接收的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 使能接收模块：置位 RXEN
- 在一帧接收完毕时，RXBF 位会置 1，如果 RXBF_IE 位为 1，将会产生中断
- 读取 PERR、FERR、OERR 寄存器，判断是否有数据错误或者溢出
- 读取 RXBUF 寄存器中的接收数据

20.6.5 低功耗休眠唤醒（UART0/1）

UART0和UART1支持RXD下降沿触发的芯片休眠唤醒。当置位了NEWUP寄存器之后，RXD输入上的下降沿事件（低电平持续时间>100ns）将会使芯片从休眠模式下唤醒，借助这个功能，可以实现UART0/1在休眠模式下接收数据。

软件配置方法如下：

- 配置UART寄存器，使能NEWUP
- 配置UART工作时钟为RCHF，根据需要配置波特率分频寄存器
- 将对应GPIO配置为UART数据接收功能
- 置位RXEN，使能接收
- 软件设置芯片进入休眠，等待UART接收事件

20.6.6 使用 DMA 进行 UART 收发

当 UART 模块被使能后，UART 模块在发送缓冲寄存器空和接收缓冲寄存器满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 UART 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 UART 请求，并完成 RAM 和 UART 之间的数据搬运。

应用举例：使用 DMA 进行 UART0 接收

- 将 DMA 通道 1 或 3 配置为 RXD0
- 设置对应通道参数：RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 UART 模块参数
- 使能 UART 模块接收使能，等待数据接收
- 收到数据后 UART 自动产生 DMA 请求
- DMA 响应请求，读取 UART 接收缓存寄存器，写入指定 RAM 地址

20.6.7 DMA 模式下的发送完成中断

当 UART 通过 DMA 进行数据发送时，DMA 会在指定长度的数据传输完成后产生 DMA 通道中断。但是当通道中断产生时，最后一帧数据刚刚被写入 UART 发送缓冲区，还未被发送出去。

通过配置 DMATXIFCFG 寄存器，可以实现 DMA 传输完成、并且最后一帧数据发送完成的情况下，产生一个发送完成中断（缓冲区空或者移位寄存器空），以便实现所有数据全部发送出去后，再中断 CPU 的应用场景。

软件工作流程说明如下：

- 配置DMA通道为UART发送
- 关闭DMA通道中断使能
- 置位DMATXIFCFG寄存器，仅允许最后一帧数据产生中断输出
- 准备待发送数据，使能DMA
- 置位UART TXBE_IE或TXSE_IE寄存器，允许中断产生
- UART连续发送，直到最后一帧，发送期间不会产生TXBE或TXSE中断
- 最后一帧发送完成后，UART产生TXBE或TXSE中断

下表假设 UART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
0	x	1~N	每帧发送完成后置位	不产生
1	0	1~N	每帧发送完成后置位	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

表 20-4 DMA 发送中断

20.7 波特率发生

20.7.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 UART4 和 UART5，F_{CLK} 就是 APBCLK；对于 UART0 和 UART1，F_{CLK} 是独立与 APBCLK 的工作时钟。

为了支持全双工通信，接收和发送波特率单独产生；

下表是常用系统时钟频率下的波特率：

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300	0	80000	299.9991	-0.00031	106667

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

表 20-5 常用时钟频率下波特率计算

20.7.1 波特率自适应

利用 Timer 的 Capture 功能，可以实现波特率自适应功能。可实现的一种方法为，外部 UART 设备按约定的数据内容(比如 0xF8)发送一帧，由 Timer 对该帧数据的高电平脉宽进行计数，MCU 读取 Timer 捕捉结果计算得到波特率因子，并写入波特率发生寄存器中，作为波特率发生的时钟分频计数值 X 使用。这时接收状态复位，重新等待起始位，以写入的波特率因子所产生的波特率接收数据。参考 Timer 章节。

20.8 红外调制

TZBRG 寄存器保存一个 11 位的分频系数 X，其值为 0~2047 之间的任一整数。所有 UART 共用一个红外调制频率发生器。

红外调制频率计算公式：

$FIR = F_{APBCLK} / (TZBRG+ 1)$

红外调制的方式为：发送数据 0 时调制红外频率，发送数据 1 时不调制。为满足 PNP 和 NPN 两种红外驱动管的需求，寄存器 IRFLAG 位控制红外调制输出的极性。IRFLAG=0 时为正极性输出，适合 PNP 管驱动；IRFLAG=1 时为负极性输出，适合 NPN 管驱动。

TH 寄存器用于配置红外调制占空比

占空比：Y = (TZBRG[10:4] * TH) /(TZBRG+1)

当 TH=4’b0000 时，占空比为 Y = (TZBRG[10:1]+1)/(X+1)；

当 TZBRG[10:4]=7’h00 时，占空比为 Y = TH/(TZBRG[3:0]+ 1)；若此时 TH>TZBRG [3:0]，则红外调制时钟 IRCLK 为固定高电平。

当红外调制极性反向时（IRFLAG=1），占空比也为 1-Y

红外调制波形见下图：

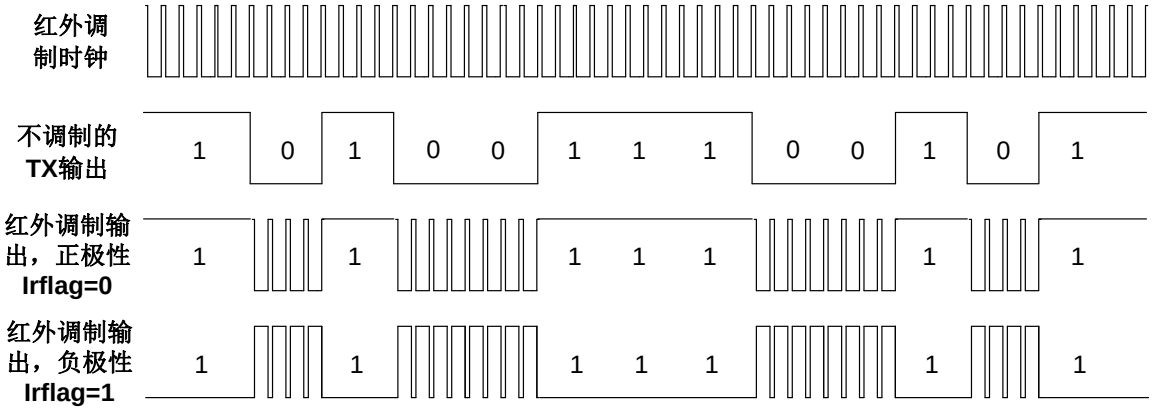


图 20-7 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

20.9 接收超时

针对 MODBUS 等时间敏感型应用，设计了接超时机制。当使能 RXTOEN 寄存器后，超时计数器以波特率时钟计数，当每次收到一个完整的数据帧，将清零超时计数器并重新开始计数。超时溢出的上限值可以由软件配置，最大 255 波特。

注：UART4 和 UART5 不支持接收超时功能。

20.10 发送延迟

通过 TXDLY_LEN 寄存器，可以控制两个数据帧发送之间的间隔时间，单位是波特。发送延迟是从上一帧最后一个 STOP 位结束，到下一帧起始位之间的间隔。

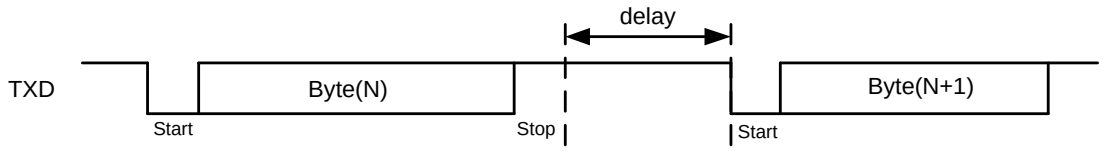


图 20-8 UART 发送延迟

注：UART4 和 UART5 不支持发送延迟功能。

20.11 寄存器

offset 地址	名称	符号
UART 公共寄存器(模块起始地址：0x40019C00)		
0x00000000	红外调制配置寄存器 （Infrared modulation Control Register）	UART_IRCR
UART0 寄存器(模块起始地址：0x40011C00**)**		
0x00000000	UART0 控制状态寄存器 （UART0 Control Status Register）	UART0_CSR
0x00000004	UART0 中断使能寄存器 （UART0 Interrupt Enable Register）	UART0_IER
0x00000008	UART0 中断标志寄存器 （UART0 Interrupt Status Register）	UART0_ISR
0x0000000C	UART0 超时和延迟寄存器 （UART0 Time-Out and Delay Register）	UART0_TODR
0x00000010	UART0 接收缓冲寄存器 （UART0 Receive Buffer）	UART0_RXBUF
0x00000014	UART0 发送缓冲寄存器 （UART0 Transmit Buffer）	UART0_TXBUF
0x00000018	UATR0 波特率产生寄存器 （UART0 Baud rate Generator Register）	UART0_BGR
UART1 寄存器(模块起始地址：0x40012000**)**		
0x00000000	UART1 控制状态寄存器 （UART1 Control Status Register）	UART1_CSR
0x00000004	UART1 中断使能寄存器 （UART1 Interrupt Enable Register）	UART1_IER
0x00000008	UART1 中断标志寄存器 （UART1 Interrupt Status Register）	UART1_ISR
0x0000000C	UART1 超时和延迟寄存器 （UART1 Time-Out and Delay Register）	UART1_TODR
0x00000010	UART1 接收缓冲寄存器 （UART1 Receive Buffer）	UART1_RXBUF
0x00000014	UART1 发送缓冲寄存器 （UART1 Transmit Buffer）	UART1_TXBUF
0x00000018	UATR1 波特率产生寄存器 （UART1 Baud rate Generator Register）	UART1_BGR
UART4 寄存器(模块起始地址：0x4001A000**)**		
0x00000000	UART4 控制状态寄存器 （UART4 Control Status Register）	UART4_CSR
0x00000004	UART4 中断使能寄存器 （UART4 Interrupt Enable Register）	UART4_IER
0x00000008	UART4 中断标志寄存器 （UART4 Interrupt Status Register）	UART4_ISR
0x00000010	UART4 接收缓冲寄存器 （UART4 Receive Buffer）	UART4_RXBUF
0x00000014	UART4 发送缓冲寄存器 （UART4 Transmit Buffer）	UART4_TXBUF
0x00000018	UATR4 波特率产生寄存器 （UART4 Baud rate Generator Register）	UART4_BGR
UART5 寄存器(模块起始地址：0x4001A400**)**		
0x00000000	UART5 控制状态寄存器	UART5_CSR

offset 地址	名称	符号
	（UART5 Control Status Register）	
0x00000004	UART5 中断使能寄存器 （UART5 Interrupt Enable Register）	UART5_IER
0x00000008	UART5 中断标志寄存器 （UART5 Interrupt Status Register）	UART5_ISR
0x00000010	UART5 接收缓冲寄存器 （UART5 Receive Buffer）	UART5_RXBUF
0x00000014	UART5 发送缓冲寄存器 （UART5 Transmit Buffer）	UART5_TXBUF
0x00000018	UATR5 波特率产生寄存器 （UART5 Baud rate Generator Register）	UART5_BGR

20.11.1 红外调制配置寄存器（UART_IRCR）

名称	UART_IRCR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG[10:8]		
位权限	R/W-0	R/W-0000				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG[7:0]							
位权限	R/W-1101 0010							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性 (Infra Red) 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数 (Tranmission High Duty)
10:0	**TZBRG**	红外调制频率 (Transmission Baud Rate)

注：此寄存器用于控制UART0/1/3

20.11.2 UARTx 控制状态寄存器（UARTx_CSR）

名称	**UARTx_CSR(x=0,1,4,5)**							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						TXIREN	RXTOEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-00		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23:18	-	未实现：读为0
17	**TXIREN**	发送红外调制使能 (Transmit Infra-red modulation Enable) 1：使能红外调制发送 0：关闭红外调制发送
16	**RXTOEN**	接收超时使能 (Receive Time-Out Enable) 1：使能接收超时功能 0：关闭接收超时功能
15:13	-	未实现：读为0
12	**IOSWAP**	RX 和 TX 引脚交换 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
11	**NEWUP**	UART RX下降沿唤醒功能使能寄存器（仅UART0和UART1有效）(Negedge Wakeupenable) 1：使能RX下降沿唤醒 0：禁止RX下降沿唤醒
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 (DMA transmit interrupt enable) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit config) 0：1位停止位 1：2 位停止位
7:6	**PDSEL**	每帧的数据长度选择；此寄存器对数据发送和接收同时有效 (Payload data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据

位号	助记符	功能描述
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

20.11.3 UARTx 中断使能寄存器（UARTx_IER）

名称	**UARTx_IER(x=0,1,4,5)**							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	U-0				R/W-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWUP_IE	-					TXBE_IE	TXSE_IE
位权限	R/W-0	U-0					R/W-0	R/W-0

位号	助记符	功能描述
31:12	**-**	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能,1 有效 (Receive Time-Out Interrupt Enable) （仅 UART0 和 UART1 有效）
10	**RXERR_IE**	接收错误中断使能，1 有效(Receive Error Interrupt Enable)
9	**-**	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7	**NEWUP_IE**	RX 下降沿异步检测中断使能，1 有效 (Negedge Wakeup Interrupt Enable)
6:2	**-**	**未实现：读为 0**
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Empty Interrupt Enable)

20.11.4 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0,1,4,5)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					PERR	FERR	OERR
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RXTO	-		RXBF
位权限	U-0				R/W-0	U-0		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWKF	-					TXBE	TXSE
位权限	R/W-0	U-0					R-0	R/W-0

位号	助记符	功能描述
31:19	-	未实现：读为0
18	PERR	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error,write 1 to clear)
17	FERR	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	OERR	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 或者读取 RXBUF 时清零 接收溢出时，接收缓冲器中原有的数据被新数据覆盖。 (RX buffer Overflow Error flag,write 1 to clear)
15:12	-	未实现：读为0
11	RXTO	接收超时中断标志，硬件置位，软件写 1 清零 (Receive Time-Out flag,write 1 to clear) （仅 UART0 和 UART1 有效）
10:9	-	未实现：读为0
8	RXBF	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full flag write 1 to clear)
7	NEWKF	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零 (Negedge Wakeup Flag write 1 to clear) （仅 UART0 和 UART1 有效）
6:2	-	未实现：读为 0
1	TXBE	发送缓存空中断标志，硬件置位，软件写入 TXBUF 时清零 (Transmit Buffer Empty flag)
0	TXSE	发送缓存空且移位寄存器发送完成中断标志，硬件置位，软件写 1 或者软件写发送缓存时清零 (Transmit Shift register Empty flag,write 1 to clear)

20.11.5 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(x=0,1)							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15:8	**TXDLY_LEN**	发送延迟，最大 255baud (Transmit Delay Length)
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud (Receive Time-Out Length)

20.11.6 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF(x=0,1,4,5)							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	**-**	未实现：读为0
8:0	**RXBUF**	接收数据缓冲寄存器数据 (Receive buffer)

7位收发时，接收的7bits数据存入RXBUF[6:0]

20.11.7 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF(x=0,1,4,5)							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	**-**	未实现：读为0
8:0	**TXBUF**	发送数据缓冲寄存器数据 (Transmit Buffer)

7位收发时，发送的7bits数据写入TXBUF[6:0]

20.11.8 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR(x=0,1,4,5)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG[15:8]							
位权限	R/W-0000 0011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG[7:0]							
位权限	R/W-0100 0001							

位号	助记符	功能描述
31:16	**-**	未实现：读为0
15:0	**SPBRG**	波特率产生器寄存器值(Serial Port Baud Rate Generation)

波特率计算详见36.5波特率发生章节

注：当SPBRG <= 0x000F时，UARTDIV=16'H000F;

当SPBRG >0x000F时，UARTDIV=SPBRG;

21 LPUART

21.1 概述

LPUART 是增强型异步串行通信接口，其工作时钟可以选择总线时钟（APBCLK）、32768Hz 晶振时钟（XTLF）、32KHz 低功耗环振时钟（LPOSC）、高频环振时钟（RCHF）、系统时钟（SYSCLK）、低功耗中频环振时钟（RCMF）。其中，当工作时钟选择为 XTLF 或者 LPOSC 时，可以支持到最高 9600 波特率的数据接收，此时 LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

特点：

- 异步数据收发
- 2路独立LPUART（LPUART0, LPUART1）
- 标准UART帧格式
  - 1bit起始位
  - 可配置数据长度，支持6、7、8、9bits
  - 奇校验、偶校验或无校验位
  - 1或2bit停止位
  - 与UARTx共享红外调制输出
- 可编程数据极性
- 当工作时钟为XTLF或LPOSC时，支持Sleep/DeepSleep模式下的数据收发
- 中断标志
  - 接收Buffer满
  - 接收Buffer溢出
  - 接收帧格式错误
  - 接收校验位错误
  - START检测
  - 数据匹配
  - 发送完成
- 休眠模式下唤醒芯片
  - RXD下降沿唤醒
  - 起始位检测唤醒
  - 1字节接收完成唤醒
  - 1字节数据匹配唤醒
- 支持DMA（Sleep/DeepSleep/RTCBKP模式下不支持）

21.2 结构框图

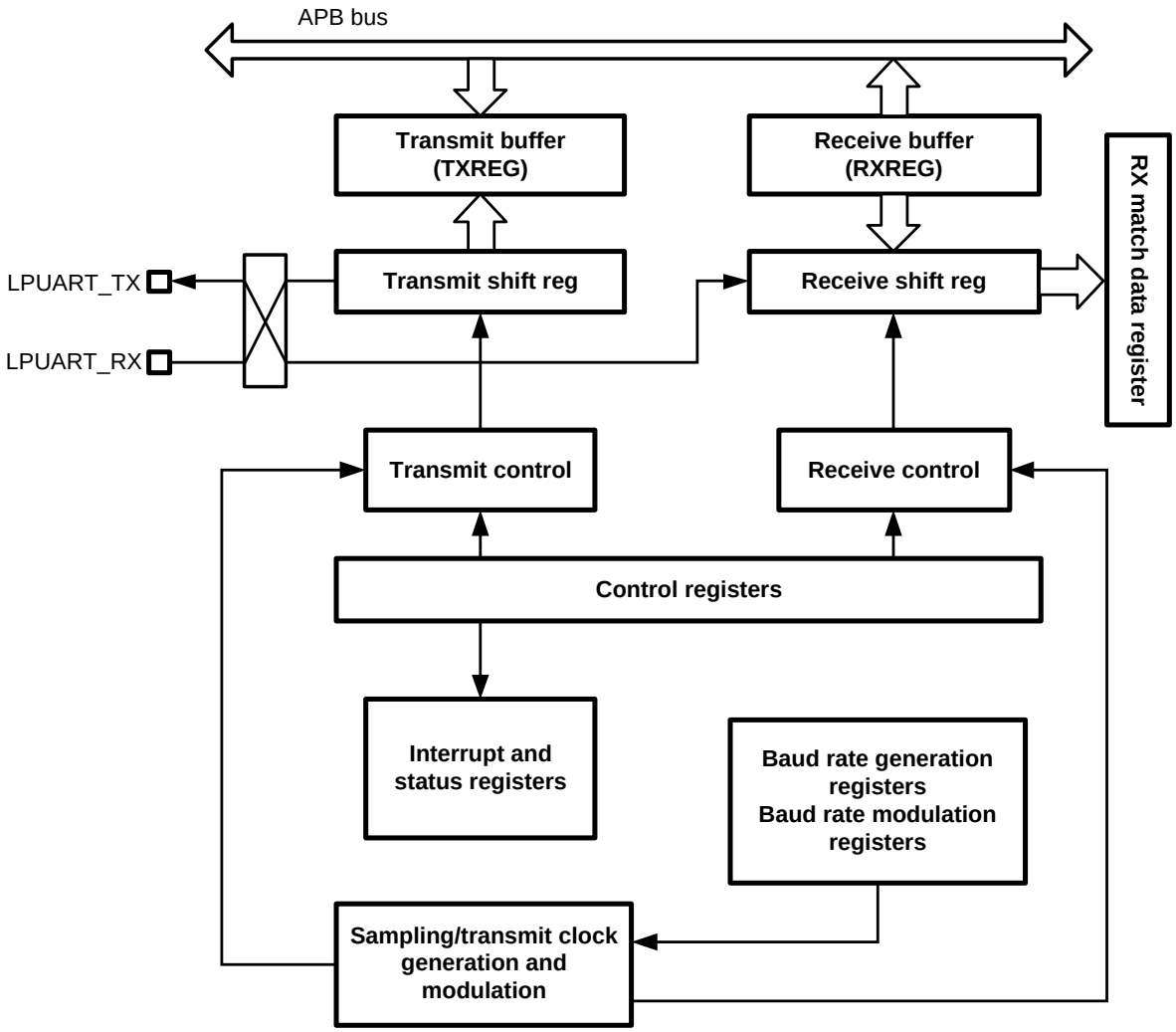


图 21-1LPUART 结构框图

21.3 引脚定义

LPUART 模块使用 2 个引脚与外部器件通信，每个 UART 的收发信号可能被映射到不同的 GPIO 上。

引脚		UARTx	符号	功能
PA13	PA2	LPUART0	LPUART0_RX	数据接收
PA14	PA3		LPUART0_TX	数据发送
PC2	PB13[1]	LPUART1	LPUART1_RX	数据接收
PC3	PB14[1]		LPUART1_TX	数据发送

[1] 仅FM33LG0x6型号有

当 LPUART 功能被同时映射到多个引脚上时：

- PA2 和 PA13 同时配置为数字外设功能
  - 只有 PA2 上的 RX 信号会输入到模块内部
- PC2 和 PB13 同时配置为数字外设功能
  - 只有 PC2 上的 RX 信号会输入到模块内部
- LPUART 发送功能被同时映射到多个 GPIO 上时，这些引脚会同时发送数据

21.4 工作时钟

LPUART 使用独立于 APBCLK 的时钟进行数据收发，工作前需要在 CMU 模块中配置相关寄存器。

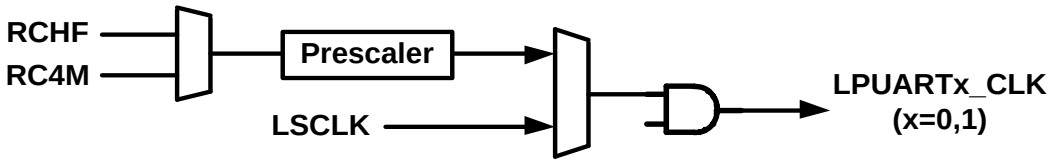
LPUART 可以使用 XTLF 或者 LPOSC 工作。由于 LPOSC 精度不高，在使用 LPOSC 进行 LPUART 通信前必须先进行时钟校准，将 LPOSC 校准到+/-1%以内。

LPOSC 校准不应使用 XTLF，因为有 XTLF 的情况下可以使用 XTLF 进行通信。因此 LPOSC 校准电路应该使用 RCHF 工作，推荐参考输入为 8MHz。

在 ACTIVE 模式下，LPUART 也可以使用 RCHF 工作，此时时钟精度会高于 LPOSC，以获得更好的时序容错性能。使用 RCHF 工作时，prescaler 电路对 RCHF 进行预分频，获得与 32768Hz 相近的时钟频率，比如 RCHF 为 8M/16M/24M 时，prescaler 分频系数应为分别 244/488/732。

在 ACTIVE 和 LP ACTIVE 模式下，LPUART 可以使用 RCMF 时钟工作。RCMF 的温度系数相对 RCHF 较差，因此建议 LPUART 工作前先进行 RCMF 校准。

LPUART 工作时钟结构参见下图，这部分功能和寄存器在 CMU 模块实现。



21.5 字符描述

LPUART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

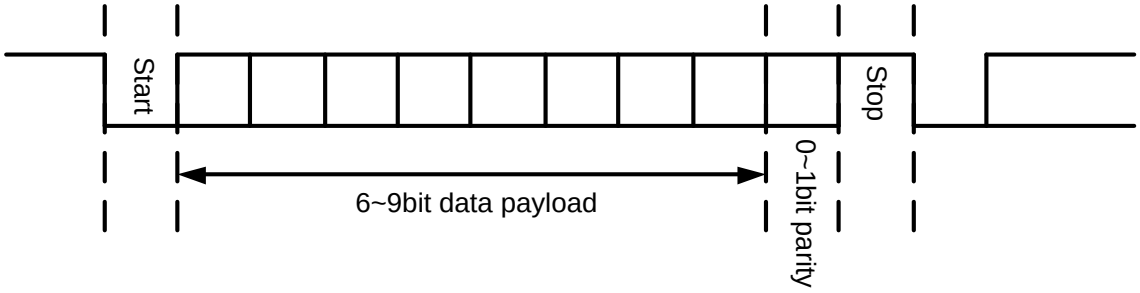


图 21-2 字符描述

LPUART 支持多种帧格式，由 LPUARTxCSR.PDSEL 寄存器和 LPUARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 21-1LPUART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

21.6 功能描述

21.6.1 位接收采样

低功耗串口模式下，工作时钟频率仅为 32Khz 左右，此时标准串口无法支持 9600bps 通信，因此需要引入 bit 调制设计。

软件需要根据通信波特率的不同合理配置调制控制寄存器 MCTL，建议的配置参数表如下：

Baud	MCTL											
	Bit0（start）	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

表 21-2LPUART 位调制参数表

21.6.2 接收流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开接收使能
- 等待中断事件

21.6.3 发送流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开发送使能
- 等待中断事件

21.6.4 使用 DMA 进行 LPUART 收发

当 LPUART 模块被使能后，LPUART 模块在发送缓冲寄存器空和接收缓冲寄存器满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 LPUART 外设，设置

RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 LPUART 请求，并完成 RAM 和 LPUART 之间的数据搬运。

应用举例：使用 DMA 进行 LPUART1 接收

- 将 DMA 通道 0 或 3 配置为 LPUART1_RX
- 设置对应通道参数：RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 LPUART1 模块参数
- 使能 LPUART1 模块接收使能 LPUEN.RXEN=1，等待数据接收
- 收到数据后 LPUART1 自动产生 DMA 请求
- DMA 响应请求，读取 LPUART1 接收缓存寄存器，写入指定 RAM 地址

21.6.5 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep、DeepSleep 模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性，通过LPUxCR.RXEV选择唤醒事件为START位、一帧接收完成、一帧数据匹配或RXD下降沿检测
- 配置LPUEN寄存器打开接收使能
- 软件进入Sleep/DeepSleep

21.6.6 LPRUN 模式下的数据 DMA 收发

通过 LPUART 和 DMA，软件可以实现 LPRUN 模式下一定数据量的 LPUART 自动收发，而无需 CPU 干预，同时保证典型条件下全芯片功耗小于 10uA。

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置DMA通道控制寄存器，选择LPUART收发
- 如果需要发送数据，将待发数据写入RAM中指定位置
- 配置DMA数据收发长度和RAM指针

- 将系统主时钟选为LSCLK
- 软件进入LPRUN
- 配置LPUEN寄存器打开发送接收使能
- 如CPU无额外工作，可以主动进入WFI/WFE，等待中断唤醒

21.6.7 DMA 模式下的发送完成中断

当 LPUART 通过 DMA 进行数据发送时，DMA 会在指定长度的数据传输完成后产生 DMA 通道中断。但是当通道中断产生时，最后一帧数据刚刚被写入 LPUART 发送缓冲区，还未被发送出去。

通过配置 DMATXIFCFG 寄存器，可以实现 DMA 传输完成、并且最后一帧数据发送完成的情况下，产生一个发送完成中断（缓冲区空或者移位寄存器空），以便实现所有数据全部发送出去后，再中断 CPU 的应用场景。

软件工作流程说明如下：

- 配置DMA通道为LPUART发送
- 关闭DMA通道中断使能
- 置位LPUART TXBE_IE或TXSE_IE寄存器，允许中断产生
- 置位DMATXIFCFG寄存器，仅允许最后一帧数据产生中断输出
- 准备待发送数据，使能DMA
- LPUART连续发送，直到最后一帧，发送期间不会产生TXBE或TXSE中断
- 最后一帧发送完成后，LPUART产生TXBE或TXSE中断

下表假设 LPUART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	LPUART interrupt
0	x	1~N	每帧发送完成后置位	不产生
1	0	1~N	每帧发送完成后置位	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

表 21-3LPUART DMA 中断说明

21.7 寄存器

offset 地址	名称	符号
LPUART0 寄存器(模块起始地址：0x40010400**)**		
0x00000000	LPUART0 控制状态寄存器 （LPUART0 Control Status Register）	LPUART0_CSR
0x00000004	LPUART0 中断使能寄存器 （LPUART0 Interrupt Enable Register）	LPUART0_IER
0x00000008	LPUART0 中断标志寄存器 （LPUART0 Interrupt Status Register）	LPUART0_ISR
0x0000000C	LPUART0 波特率调制寄存器 （LPUART0 Baud rate Modulation Register）	LPUART0_BMR
0x00000010	LPUART0 接收缓存寄存器 （LPUART0 Receive Buffer Register）	LPUART0_RXBUF
0x00000014	LPUART0 发送缓存寄存器 （LPUART0 Transmit Buffer Register）	LPUART0_TXBUF
0x00000018	LPUART0 数据匹配寄存器 （LPUART0 data Matching Register）	LPUART0_DMR
LPUART1 寄存器(模块起始地址：0x40018400**)**		
0x00000000	LPUART1 控制状态寄存器 （LPUART1 Control Status Register）	LPUART1_CSR
0x00000004	LPUART1 中断使能寄存器 （LPUART1 Interrupt Enable Register）	LPUART1_IER
0x00000008	LPUART1 中断标志寄存器 （LPUART1 Interrupt Status Register）	LPUART1_ISR
0x0000000C	LPUART1 波特率调制寄存器 （LPUART1 Baud rate Modulation Register）	LPUART1_BMR
0x00000010	LPUART1 接收缓存寄存器 （LPUART1 Receive Data Register）	LPUART1_RXBUF
0x00000014	LPUART1 发送缓存寄存器 （LPUART1 Transmit Data Register）	LPUART1_TXBUF
0x00000018	LPUART1 数据匹配寄存器 （LPUART1 data Matching Register）	LPUART1_DMR

21.7.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）

名称	LPUARTx_CSR(x=0,1)							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				WKBYTE_CFG	-	RXEV	
位权限	U-0				R/W-0	U-0	R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	DMATXIFCFG	BITORD	STOPCFG
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN

位权限	R/W-00	R/W-00	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	LPUART 通信标志，只读 (Busy) 1：LPUART 正在通信中 0：LPUART 空闲
23:20	-	未实现：读为0
19	**WKBYTE_CFG**	数据接收唤醒条件配置 (Wakeup Byte Config) 1：接收完1字节，并且奇偶校验和STOP位都正确，才触发唤醒中断 0：接收完1字节，不检查校验位和STOP位，直接触发唤醒中断
18	-	未实现：读为0
17:16	**RXEV**	唤醒中断事件配置，用于控制何种事件下向 CPU 提供唤醒中断 (Receive Wakeup Event) 00：START 位检测唤醒 01：1byte 数据接收完成 10：接收数据匹配成功 11：RXD 下降沿检测
15:12	-	未实现：读为0
11	**IOSWAP**	RX 和 TX 引脚交换 (IO swapping) 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在LPUART通过DMA进行发送时有效 (DMA Transmit Interrupt Config) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit Config) 0：1位停止位 1：2位停止位
7:6	**PDSEL**	每帧数据长度选择；此寄存器对数据发送和接收同时有效 (Payload Data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 (Parity) 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反

位号	助记符	功能描述
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

21.7.2 LPUARTx 中断使能寄存器（LPUARTx _IER）

名称	LPUARTx_IER(x=0,1)							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RXEV_IE	-	RXERR_IE	-	RXBF_IE
位权限	U-0			R/W-0	U-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE_IE	TXSE_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:13	**-**	未实现：读为0
12	**RXEV_IE**	接收唤醒事件中断使能，1 有效 (Receive Event Interrupt Enable)
11	**-**	未实现：读为0
10	**RXERR_IE**	接收错误中断使能，1 有效 (Receive Error Interrupt Enable)
9	**-**	未实现：读为0
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
7:2	**-**	未实现：读为 0
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Interrupt Enable)

21.7.3 LPUARTx 中断标志寄存器（LPUARTx _ISR）

名称	LPUARTx_ISR(x=0,1)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							RXEVF
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				TXOV	PERR	FERR	OERR

位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE	TXSE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**RXEVF**	接收唤醒事件中断标志，硬件置位，软件写 1 清零 (Receive Event Interrupt Flag,write 1 to clear) 中断标志触发源由 LPUxCR.RXEV 寄存器配置。
23:20	-	未实现：读为0
19	**TXOV**	发送缓存溢出错误，硬件置位，软件写1清零 (Transmit Overflow Error flag,write 1 to clear) 当发送缓存中的数据还未进入移位寄存器发送时，软件向发送缓存写入新数据，将触发TXOV标志置位。
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error flag,write 1 to clear)
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	**OERR**	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 清零 (Receive Buffer Overflow Error flag,write 1 to clear)
15:9	-	未实现：读为0
8	**RXBF**	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full flag, write 1 to clear)
7:2	-	未实现：读为 0
1	**TXBE**	发送缓存空中断标志，硬件置位，写入 TXBUF 时清零 (Transmit Buffer Empty flag,write 1 to clear)
0	**TXSE**	发送缓存空且发送移位寄存器空中断标志，硬件置位，软件写 1 或者发送数据被载入移位寄存器时清零 (Transmit Shift register Empty flag, write 1 to clear)

21.7.4 LPUARTx 波特率调制寄存器（LPUARTx _BMR）

名称	LPUARTx_BMR(x=0,1)							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				MCTL[11:8]			
位权限	U-0				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MCTL[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	BAUD
位权限	U-0	R/W-000

位号	助记符	功能描述
31:28	**-**	未实现：读为0
27:16	**MCTL**	LPUART 每个 bit 的位宽调制控制信号 (Bit Modulation Control)
15:3	**-**	未实现：读为0
2:0	**BAUD**	波特率控制（bps） 000：9600 001：4800 010：2400 011：1200 100：600 101/110/111：300

21.7.5 LPUARTx 接收缓存寄存器（LPUARTx _RXBUF）

名称	LPUARTx_RXBUF(x=0,1)							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	**-**	**未实现：读为0**
8:0	**RXBUF**	接收数据缓存寄存器 (Receive Buffer)

21.7.6 LPUARTx 发送缓存寄存器（LPUARTx _TXBUF）

名称	LPUARTx_TXBUF(x=0,1)							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-							TXBUF[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**TXBUF**	发送数据缓存寄存器 (Transmit Buffer)

21.7.7 LPUARTx 数据匹配寄存器（LPUARTx _DMR）

名称	LPUARTx_DMR(x=0,1)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MATD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MATD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	-	**未实现：读为0**
8:0	**MATD**	第一帧接收比较数据，如果 RXEV=10，当接收到的第一帧数据与 MATD 相同时，触发 RXEVF 中断，可以用于休眠模式下的数据接收唤醒。(Matched Data)

22 SPI

22.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过 4 线交换数据的串行同步通讯手段。芯片提供了 2 个 SPI 接口模块，可配置为主设备或从设备，实现与外部的 SPI 通信。

特点：

- 全双工4线串行同步收发（SCLK, MOSI, MISO, SSN）
- MISO和MOSI可交换引脚顺序
- 半双工4线串行同步收发（SCLK, SDATA, SSN, DCN）
- 2路独立通道
- 主从模式
- 可编程时钟极性和相位
- 可编程比特速率
- 可编程数据字长（8/16/24/32bits）
- 最大波特率为$F_{APBCLK}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持DMA

22.2 结构框图

下图为 SPI 模块的结构示意图。

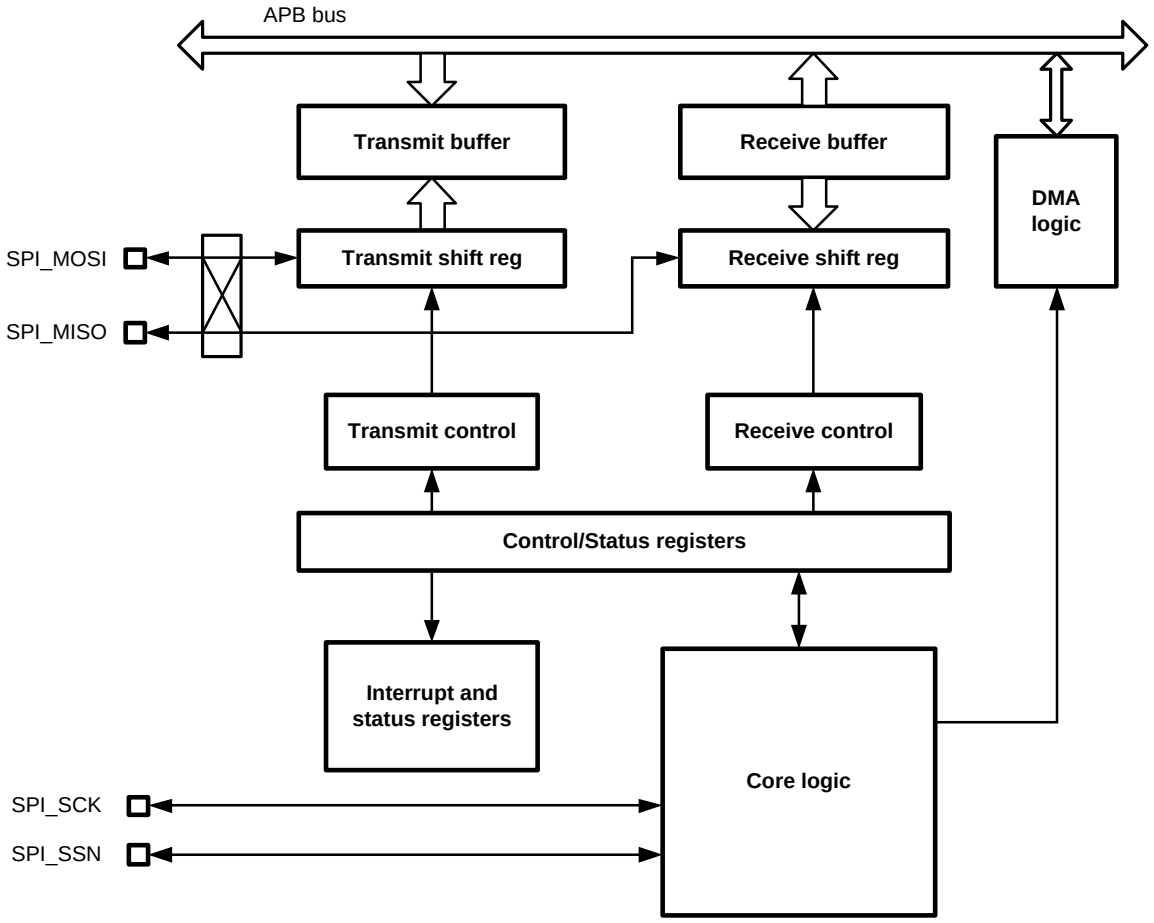


图 22-1 SPI 结构框图

22.3 引脚定义

SPI 模块使用 4 个引脚与外部器件通信，在全双工和半双工模式下，这四个引脚的功能定义有所不同，如下表所示：

引脚	SPIx	全双工	功能	半双工	功能
PB8/PD2	SPI1	SSN	片选信号	SSN	片选信号
PB9/PD3		SCLK	时钟	SCLK	时钟
PB10/PD4		MISO	主机输入从机输出	DCN	命令/数据标识
PB11/PD5		MOSI	主机输出从机输入	SDATA	双向数据
PC7	SPI2	SSN	片选信号	SSN	片选信号
PC8		SCLK	时钟	SCLK	时钟
PC9		MISO	主机输入从机输出	DCN	命令/数据标识
PC10		MOSI	主机输出从机输入	SDATA	双向数据

22.4 接口时序

为了兼容不同的 SPI 外设，SPI 串行时钟的时序可以通过时钟相位选择位（CPHA）和时钟极性选择位(CPOL)设置产生 4 种不同组合。为保证数据正确传输，主从器件的时序配置必需一致。

当处于从器件模式或 SPI 系统使能位（SPE）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

22.4.1 CPHA=0

CPHA=0 时，SPI 模块在串行时钟的第一个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，SPI 在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，SPI 在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据。

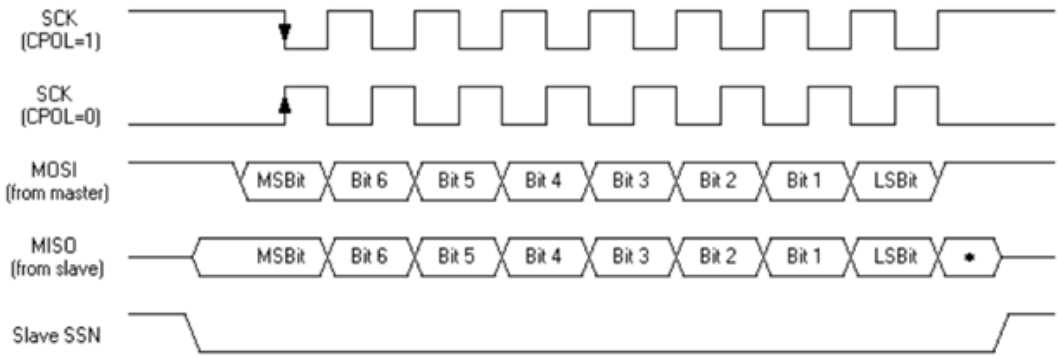


图 22-2 SPI 数据/时钟时序图（CPHA=0）

22.4.2 CPHA=1

CPHA=1 时，SPI 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据。

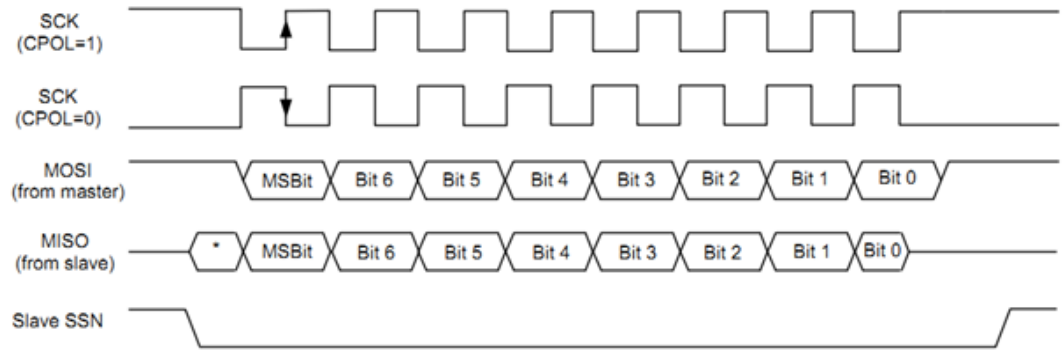


图 22-3 SPI 数据/时钟时序图（CPHA=1）

22.4.1 4 线半双工模式（主机）

4线半双工模式可以支持与点阵液晶或TFT屏的交互通信。在这种模式下，通过DCN信号的高低来区分当前发送的是命令帧还是数据帧。双向数据都通过SDATA（MOSI）引脚收发，由硬件自动完成数据方向切换。FM33LC0XX的SPI仅支持4线半双工主机模式，不支持从机模式。

所有通信都由主机发起，主机首先发送命令帧，然后再进行数据帧传输。命令帧和数据帧通过DCN信号线区分。主机可以通过4线半双工接口向从机写入数据，或从从机读取数据。

4线半双工写操作

软件通过清零HD_RW寄存器，表示当前主机要发起写操作。

主机发起写操作前，首先发送写命令帧。当写命令帧发送完毕后，如果发送缓冲区为空，则硬件将拉高SSN并停止SCLK发送；如果发送缓冲区已经写入了新的数据，则硬件会连续发送后续的数据帧。

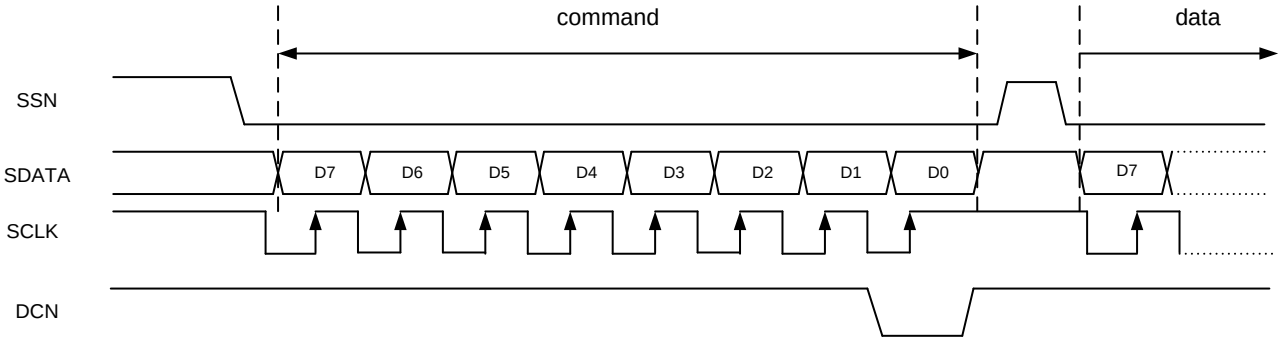


图 22-4 4 线半双工写操作

DCN在第8个时钟上升沿采样判决，如果为0，表示当前帧是命令帧。发送命令帧前，软件需要将DCN寄存器写0，命令帧发送完成后硬件自动将DCN寄存器置位。

4线半双工读操作

软件通过置位HD_RW寄存器，表示当前主机要发起读操作。

4线半双工读操作支持8位、24位和32位读取。主机发起读操作时，首先发送读命令帧。当读命令帧发送完毕后，可以根据寄存器配置发送1个dummy cycle，在dummy cycle期间，SCLK时钟正常发送，但是主机不驱动SDATA，也不接受SDATA输入。

完成命令帧和dummy cycle（可选）后，4线半双工SPI自动进入接收状态，SDATA信号改由从机驱动，主机收到的数据帧将被写入接收缓冲区。每个数据帧接收完成后，将置位RXBF中断标志寄存器。软件应及时读取接收缓冲区中的数据，如果接收缓冲区和接收移位寄存器都处于满状态，硬件会停止SCLK发送，暂停从从机读取数据，直到软件或DMA读取了接收缓冲区。

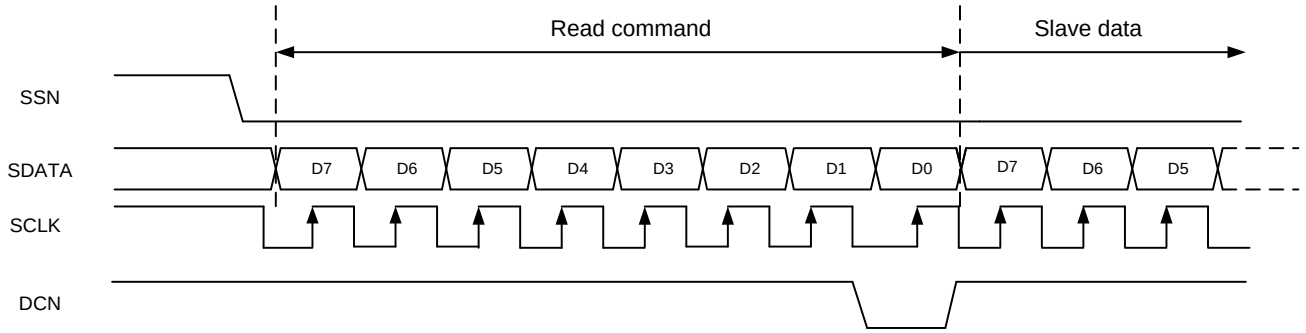


图 22-5 4 线半双工读操作（无 dummy cycle）

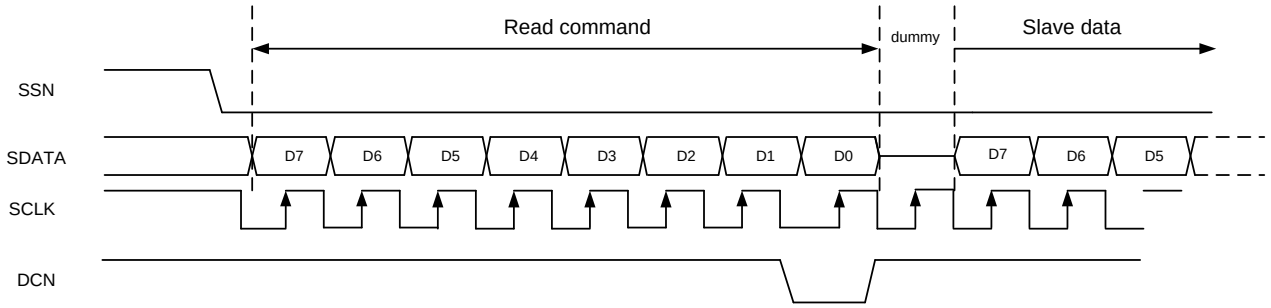


图 22-6 4 线半双工读操作（有 dummy cycle）

22.5 功能描述

22.5.1 I/O 配置

主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。数据传输时 MSB 在前。

主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。数据传输时 MSB 在前。

串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，如图 22-2 所示，当 SPI 配置为主器件时，SSN 引脚必须接

高电平，当 SPI 配置为从器件时，SSN 引脚必须接低电平。

SPI 主从器件的连接如**错误!未找到引用源。**所示：

主从器件的 MOSI、MISO 和 SCK 分别连在一起，主器件的 SSN 必须接高电平，从器件的 SSN 必须接低电平。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

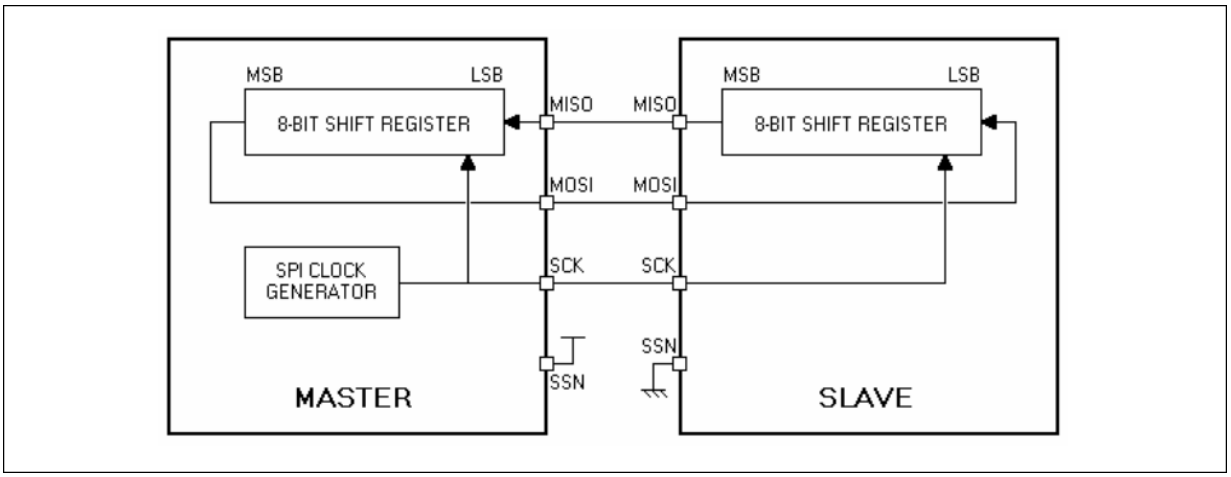


图 22-7 SPI Master/SPI Slave 互连

22.5.2 全双工数据通信

SPI模块默认为全双工通信，如果需要实现连续不间断的数据通信，软件需要确保TX BUFFER非空。即使软件只用SPI进行数据接收，由于SPI的全双工属性，软件仍需要对TX BUFFER进行写操作，此时写入的是无效数据，可根据MOSI无效状态配置写入全0或全F。

发送缓冲区

软件或DMA将待发送数据写入发送缓冲区（SPIxTXBUF寄存器），当发送开始时，硬件将数据从发送缓冲区拷贝到移位寄存器并开始发送。数据从发送缓冲区转移至移位寄存器后，发送缓存空标志（TXBE）被置位，表示可以向TXBUF写入新数据；如果TXIE寄存器置位，则产生中断。通过向TXBUF写入数据，可以清零TXBE寄存器。

如果在移位寄存器移位完成前，新的数据被写入发送缓冲区，则可以保证连续不断的数据发送。在TXBE为0的情况下写TXBUF，则会产生数据冲突，参见22.5.6数据冲突。

接收缓冲区

当SPI完成一帧数据接收后，收到的数据将从移位寄存器拷贝到接收缓冲区（SPIxRXBUF寄存器），同时RXBF标志被置位，表示RXBUF中已有数据待处理。如果RXIE寄存器置位，则产生中断。通过读取RXBUF可以清零RXBF标志。

在RXBF没有置位的情况下读RXBUF，将返回上一次接收到的数据；如果应用没有及时处理RXBF，新的数据在RXBF置位的情况下完成接收，则产生数据冲突，参见22.5.6数据冲突。

BUSY标志

当SPI正在进行数据收发时，BUSY寄存器置位。此寄存器在某些场景下可以用来判断最后一帧数据是否传输完毕。比如TXBE只是表示数据已经进入移位发送，但是真正发送完成，需要等待BUSY标志清零。

如何启动SPI通信

主机模式下，建议遵循以下步骤启动SPI通信：

- 应用配置SPI模块
- 置位SPIEN
- 向TXBUF写入数据，SPI模块自动开始发送SCK并进行数据收发

从机模式下，建议应用在主机开始发送SCK之前完成配置和使能，并将第一帧待发送数据写入TXBUF，等待主机发送SCK开始通信。

如何结束SPI通信

主机模式下，建议遵循以下步骤结束SPI通信：

- 等待RXBF和TXBE标志置位，此时移位寄存器中还有最后一帧数据正在发送
- 查询BUSY标志，直到BUSY为0，最后一帧数据收发完成
- 关闭SPI模块，如果需要，读取最后一帧接收数据

从机模式下，应用可以在读取任意一帧数据后关闭SPI模块，关闭前已经被移入移位寄存器的数据将被忽略。

22.5.3 TX-ONLY 模式

某些时候SPI通信是半双工的，在主机仅需进行发送的情况下，通过置位TXO寄存器进入TX-ONLY模式，此时MISO收到的数据不会被写入RX Buffer中，相应的也不会置位RXBF中断标志。

通过置位TXO_AC，可以实现TXO自动清零功能。在TX-ONLY模式下，如果TX buffer空（TXBE置位）并且发送移位寄存器空，则TXO寄存器自动清零，退出TX-ONLY状态。

22.5.4 RX-ONLY 模式

SPI主机仅需进行接收的情况下，通过置位RXO寄存器进入RX-ONLY模式，此时SPI模块无需软件

对TX Buffer进行写操作，即可进行连续不断的数据接收，此时MOSI将保持IDLE电平，并且不会置位TXBE中断标志寄存器。

22.5.5 主机 SSN 控制

SPI模块主机支持硬件或软件控制SSN信号。

当SSNSEN寄存器清零时，SSN由硬件电路控制；如果SSNM寄存器置位，则SPI每发完一帧数据后，将拉高SSN，SSN高电平时间由WAIT寄存器配置（若干个SCK时钟周期）；

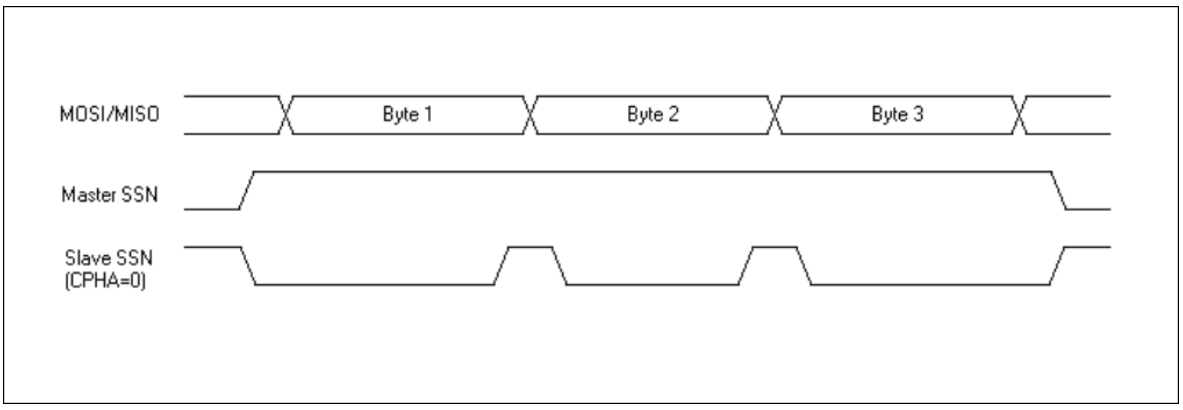


图 22-8 SPI SSN 时序图（SSNM=1，CPHA=0）

如果SSNM寄存器复位，则SPI每发完一帧数据后不会拉高SSN，而是直接进入下一帧数据发送。

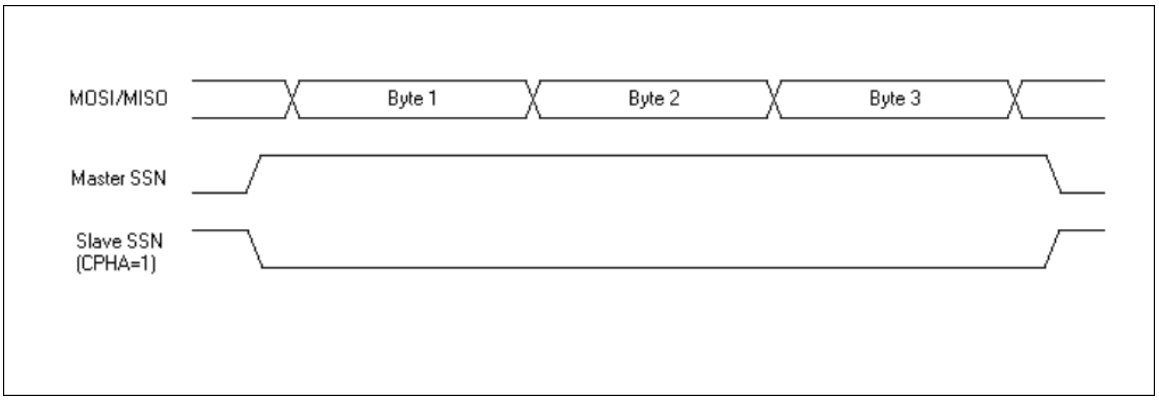


图 22-9 SPI SSN 时序图（SSNM=0）

当SSNSEN寄存器置位时，SSN直接由软件控制。软件通过写SPIxCR2.SSN寄存器位，可以直接操作SPI主机发送的SSN电平。

22.5.6 数据冲突

当 SPI 的 TX Buffer 数据尚未被读进移位寄存器，或者 SPI 的 RX Buffer 中的数据未被软件或 DMA

读取时，对 TX Buffer 或 RX Buffer 的写操作会产生对应的冲突错误，TXCOL/RXCOL 位会置起，产生中断。导致冲突的写入数据将被忽略。数据冲突错误在主从模式下都会产生。

对 TX Buffer 的写操作，由芯片内部的 Master 模块发起，包括 CPU、DMA 等等。对 RX Buffer 的写操作，则由外部 SPI 器件发起。

当数据冲突发生时，TX Buffer 和 RX Buffer 内原有数据不会被刷新，新写入的数据丢失。

22.5.7 使用 DMA 进行 SPI 收发

当 SPI 模块被使能后，SPI 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 SPI 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 SPI 请求，并完成 RAM 和 SPI 之间的数据搬运。

注意：如果使用 DMA 进行收发全双工通信，软件应先使能 DMA 发送通道，再使能 DMA 的接收通道；反之可能会导致 SPI 额外发送一个字节的 dummy 数据。

使用 DMA 进行 SPI 接收

- 将 DMA 通道 3 或 5 配置为 SPI_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 SPI 模块参数
- 使能 SPI 模块，等待数据接收
- 收到数据后 SPI 自动产生 DMA 请求
- DMA 响应请求，读取 SPI 接收缓存寄存器，写入指定 RAM 地址
- 当指定长度的 DMA 传输结束后，DMA 将忽略后续请求并产生传输完成中断，软件应处理中断并关闭 SPI
- 如果关闭 SPI 前又有数据被接收，软件可以通过写 RXBFC 清除 RXBUF

使用 DMA 进行 SPI 发送

DMA发送过程与上述接收过程类似，主要差别是，当指定长度的DMA传输结束后，软件不能立即关闭SPI，因为此时最后一帧数据还在移位发送中，因此软件需要查询BUSY标志直到移位发送结束，再关闭SPI模块。

数据帧长度与 RAM 数据组织方式

SPI 传输帧长度可以配置为 8、16、24、32bit。

当数据帧长度为 8bit 时，DMA 每次搬运 1byte，4 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data3，data2，data1，data0 }

当数据帧长度为 16bit 时，DMA 每次搬运 2bytes，2 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data1，data0 }

当数据帧长度为 24bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址，但是有效数据仅占用 RAM 字内低 24bit：

RAM word: { 8'h0，data0 }

当数据帧长度为 32bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址：

RAM word: { data0 }

22.6 寄存器

offset 地址	名称	符号
SPI1 寄存器(模块起始地址:0x40018C00)		
0x00000000	SPI1 控制寄存器 1 （SPI1 Control Register1）	SPI1_CR1
0x00000004	SPI1 控制寄存器 2 （SPI1 Control Register2）	SPI1_CR2
0x00000008	SPI1 控制寄存器 3 （SPI1 Control Register3）	SPI1_CR3
0x0000000C	SPI1 中断使能寄存器 （SPI1 Interrupt Enable Register）	SPI1_IER
0x00000010	SPI1 中断状态寄存器 （SPI1 Status Register）	SPI1_ISR
0x00000014	SPI1 发送数据缓冲寄存器 （SPI1 Transmit Buffer）	SPI1_TXBUF
0x00000018	SPI1 接收数据缓冲寄存器 （SPI1 Receive Buffer）	SPI1_RXBUF
SPI2 寄存器(模块起始地址:0x40010800)		
0x00000000	SPI2 控制寄存器 1 （SPI2 Control Register1）	SPI2_CR1
0x00000004	SPI2 控制寄存器 2 （SPI2 Control Register2）	SPI2_CR2
0x00000008	SPI2 控制寄存器 3 （SPI2 Control Register3）	SPI2_CR3
0x0000000C	SPI2 中断使能寄存器 （SPI2 Interrupt Enable Register）	SPI2_IER
0x00000010	SPI2 中断状态寄存器 （SPI2 Status Register）	SPI2_ISR
0x00000014	SPI2 发送数据缓冲寄存器 （SPI2 Transmit Buffer）	SPI2_TXBUF
0x00000018	SPI2 接收数据缓冲寄存器 （SPI2 Receive Buffer）	SPI2_RXBUF

22.6.1 SPIx 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1(x=1,2)							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	MSPA	SSPA	MM
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	CPOL	CPHA
位权限	R/W-00		R/W-000			R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	**IOSWAP**	MOSI 和 MISO 引脚交换 (IO swapping) 0：默认引脚顺序 1：交换引脚顺序
10	**MSPA**	Master Sampling Position Adjustment，Master 对 MISO 信号的采样位置调整，用于高速通信时补偿 PCB 走线延迟 1：采样点延迟半个 SCK 周期 0：不调整
9	**SSPA**	Slave Sending Position Adjustment，Slave MISO 发送位置调整 1：提前半个 SCK 周期发送 0：不调整
8	**MM**	Master/Slave 模式选择。 1：Master 模式 0：Slave 模式
7:6	**WAIT**	Master 模式下，每发送完一帧后加入至少(1+WAIT)个 SCK cycle 等待时间，再传输下一帧的数据。如果 SSN 由硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。
5:3	**BAUD**	Master 模式波特率配置位: 000： $f_{APBCLK}/2$ 001： $f_{APBCLK}/4$ 010： $f_{APBCLK}/8$ 011： $f_{APBCLK}/16$ 100： $f_{APBCLK}/32$ 101： $f_{APBCLK}/64$ 110： $f_{APBCLK}/128$ 111： $f_{APBCLK}/256$ 当通信正在进行的时候，不能修改这些位。
2	**LSBF**	帧格式 (LSB First) 0：先发送 MSB 1：先发送 LSB 注：当通信在进行时不能改变该位的值。
1	**CPOL**	时钟极性选择 (Clock Polarity) 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
0	**CPHA**	时钟相位选择 (Clock Phase) 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

22.6.2 SPIx 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2(x=1,2)							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN	-			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	U-0			R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	**DUMMY_EN**	4 线半双工协议下是否在读操作中插入 dummy cycle (Dummy cycle Enable) 0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	-	RFU：未实现，读为 0
11	**RXO**	RXONLY 控制位，此寄存器置位时，SPI 可以连续接收，无需软件写 TXBUF (Receive Only mode) 1：启动 Master 的单接收模式 0：关闭单接收模式（收发全双工）
10:9	**DLEN**	通信数据字长配置 (Data Length) 00：8bit 01：16bit 10：24bit 11：32bit
8	**HALFDUPLEX**	通信模式选择 (Half-Duplex mode) 0：标准 SPI 模式，4 线全双工 1：DCN 模式，4 线半双工
7	**HD_RW**	半双工模式下主机读写操作配置 (Read/Write config for Half-Duplex mode) 0：4 线半双工协议下主机写入从机 1：4 线半双工协议下主机读取从机
6	**CMD8b**	半双工模式下定义 command 帧长度 (Command 8 bits) 1：command 帧固定为 8bit 0：command 帧长度由 DLEN 定义
5	**SSNM**	Master 模式下 SSN 控制模式选择 (SSN mode) 1：每发送完一帧后 Master 拉高 SSN，维持高电平时间由 WAIT 寄存器控制 0：每发送完一帧后 Master 保持 SSN 为低
4	**TXO_AC**	TXONLY 硬件自动清空的使能 (TXONLY auto-clear enable) 1：TXONLY 硬件自动清零有效，软件使能 TXO 后，等待发送完毕后，硬件清零 0：关闭 TXONLY 硬件自动清零
3	**TXO**	TXONLY 控制位 (Transmit Only mode enable) 1：启动 Master 的单发送模式 0：关闭单发送模式（收发全双工）
2	**SSN**	Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平

位号	助记符	功能描述
		1：SSN 输出高电平 0：SSN 输出低电平
1	**SSNSEN**	Master 模式下，软件控制 SSN 使能 (SSN Software Enable) 1：Master 模式下 SSN 输出由软件控制 0：Master 模式下 SSN 输出由硬件自动控制
0	**SPIEN**	SPI 使能 (SPI enable) 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

22.6.3 SPIx 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3(x=1,2)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TXBFC	RXBFC	MERRC	SERRC
位权限	U-0				W-0	W-0	W-0	W-0

位号	助记符	功能描述
31:4	**-**	RFU：未实现，读为 0
3	**TXBFC**	Transmit Buffer Clear，软件写 1 清除发送缓存，写 0 无效
2	**RXBFC**	Receive Buffer Clear，软件写 1 清除发送缓存，写 0 无效
1	**MERRC**	Master Error Clear，软件写 1 清除 HSPISTA.MERR 寄存器
0	**SERRC**	Slave Error Clear，软件写 1 清除 HSPISTA.SERR 寄存器

22.6.4 SPIx 中断控制寄存器（SPIx_IER）

名称	SPIx_IER(x=1,2)							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	ERRIE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	**-**	RFU：未实现，读为 0
2	**ERRIE**	SPI 错误中断使能 (Error Interrupt Enable)
1	**TXIE**	发送完成中断使能 (Transmit Interrupt Enable)
0	**RXIE**	接收完成中断使能 (Receive Interrupt Enable)

22.6.5 SPIx 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR(x=1,2)							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DCN_TX	-	RXCOL	TXCOL	BUSY
位权限	U-0			R/W-1	U-0	R/W-0	R/W-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MERR	SERR	-			TXBE	RXBF
位权限	U-0	R-0	R-0	U-0			R-1	R-0

位号	助记符	功能描述
31:13	**-**	RFU：未实现，读为 0
12	**DCN_TX**	半双工模式下（HALFDUPLEX=1），配置在每个数据帧的最后 bit 发送的 DCN 信号电平 (Data/Command transmit config) 0：DCN=0，表示命令帧 1：DCN=1，表示数据帧 软件应在发送前设置 DCN_TX 寄存器，如果 DCN_TX=0，硬件在完成一帧发送后，自动将 DCN_TX 置 1，即默认只会发送一个命令帧，后续都是数据帧。
11	**-**	RFU：未实现，读为 0
10	**RXCOL**	接收缓存溢出，软件写 1 清零 (Receive Collision flag,write 1 to flag)
9	**TXCOL**	发送缓存溢出，软件写 1 清零 (Transmit Collision flag,write 1 to clear)
8	**BUSY**	SPI 空闲标志，只读(busy flag) 1：SPI 传输进行中 0：SPI 传输空闲
7	**-**	RFU：未实现，读为 0
6	**MERR**	Master Error 标志(Master Error flag) 当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	**SERR**	Slave Error 标志(Slave Error flag) 当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位
4:2	**-**	RFU：未实现，读为 0

位号	助记符	功能描述
1	**TXBE**	TX Buffer Empty 标志位(TX Buffer Empty flag) 1：发送缓存空，软件写 TXBUF 清零 0：发送缓存满
0	**RXBF**	RX Buffer Full 标志位(RX Buffer Full flag) 1：接收缓存满，软件读 RXBUF 清零 0：接收缓存空

22.6.6 SPIx 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF(x=1,2)							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	TXBUF[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	TXBUF[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	TXBUF[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	**TXBUF**	SPI 发送缓存 (Transmit Buffer)

22.6.7 SPIx 接收缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF(x=1,2)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	RXBUF[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	RXBUF[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	**RXBUF**	SPI 接收缓存 (Receive Buffer)

23 智能卡接口（ISO7816）

23.1 概述

智能卡接口（7816）是外部智能卡通过 2 线交换 8 位数据的串行同步通讯手段。芯片提供了 2 个 7816 主机接口模块。

- 1路7816接口
- 具备卡时钟输出端口，输出频率在1MHz~5MHz之间可设
- 位传输方向可配置，支持MSB First或LSB First
- 错误信号宽度可配置为1/1.5/2个ETU
- 发送数据支持传输错误重发机制，重发次数可配置为0~3次
- 支持EGT可设0~256，并支持多种超时中断
- 具有数据接收完成/接收错误中断，并提示错误类型
- 发送中断产生条件可配置为缓冲区空或移位寄存器空
- 支持DMA接口

23.2 结构框图

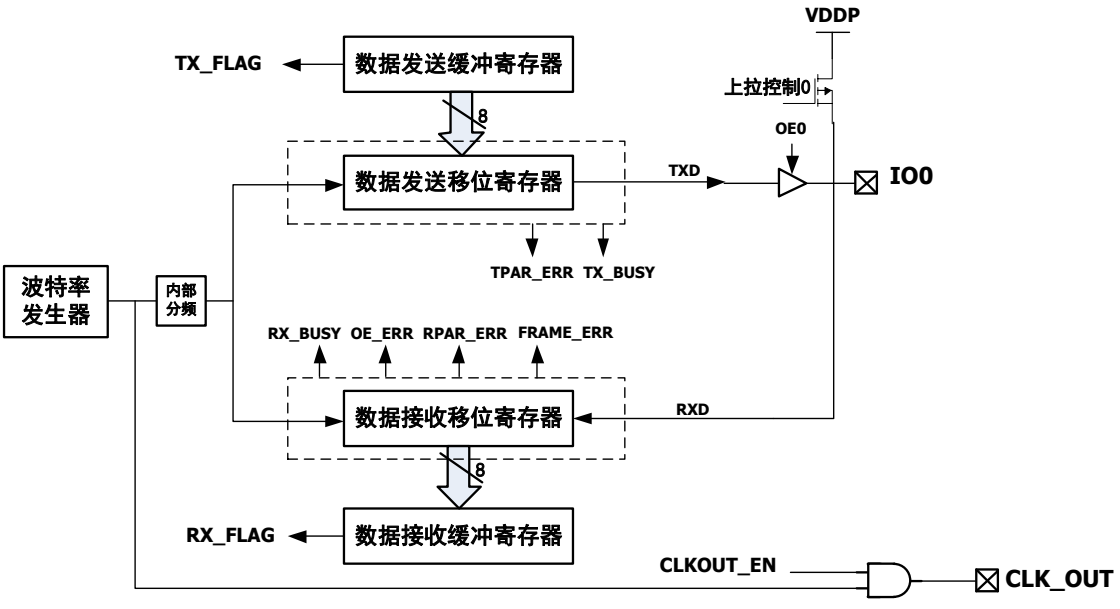


图 23-1 ISO7816 结构框图

23.3 接口时序

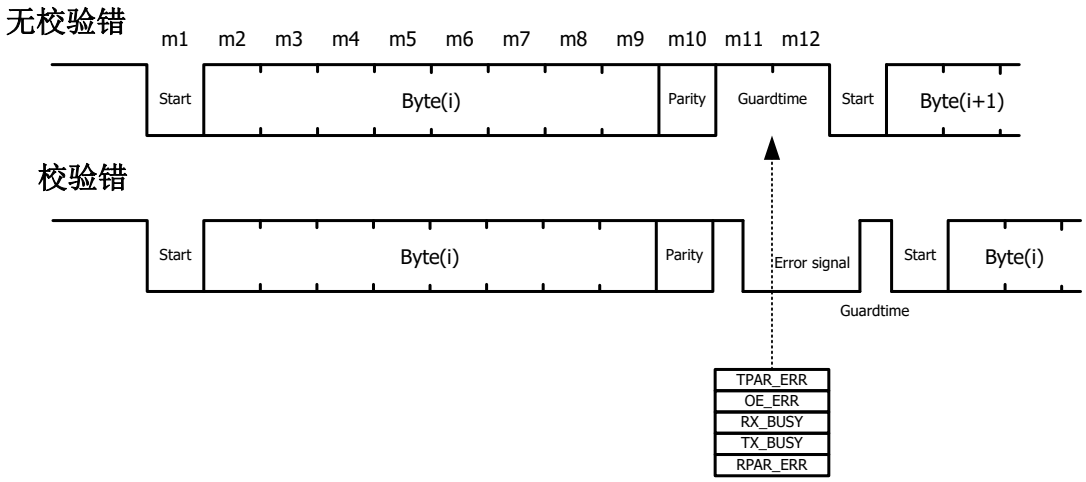


图 23-2 ISO7816 数据帧结构

参照 7816 协议标准，7816 基本接口时序如下：

- 一个起始位后跟8个数据位及1个校验位，以1ETU或2ETU的GUARDTIME结束。
- 单字节数据长度最小为11ETU或12ETU。
- 第10.5个ETU接收电路校验接收数据，若校验正确，则插入2个ETU的GUARDTIME，确保数据长度为12ETU，并在第11个ETU时令RX_BUSY无效并产生可能的OE_ERR标志，完成数据发送；若接收校验出错，则在第10.5ETU拉低IO，产生ERROR SIGNAL。ERROR SIGNAL最短1个ETU，最长2个ETU。并在第11个ETU根据需要产生RPAR_ERR标志。
- 第11个ETU时发送电路未采样到ERROR SIGNAL，则说明发送数据正确，数据发送完成，令TX_BUSY无效。
- 若第11个ETU发送电路采样到ERROR SIGNAL，则说明发送数据错误，根据设定产生需要的TPAR_ERR或等待2个ETU后重发数据。
- 所有中断标志尽可能都在同一时刻产生，使得MCU可以正确及时处理中断。

23.4 功能描述

23.4.1 数据接收

7816 数据接收过程：

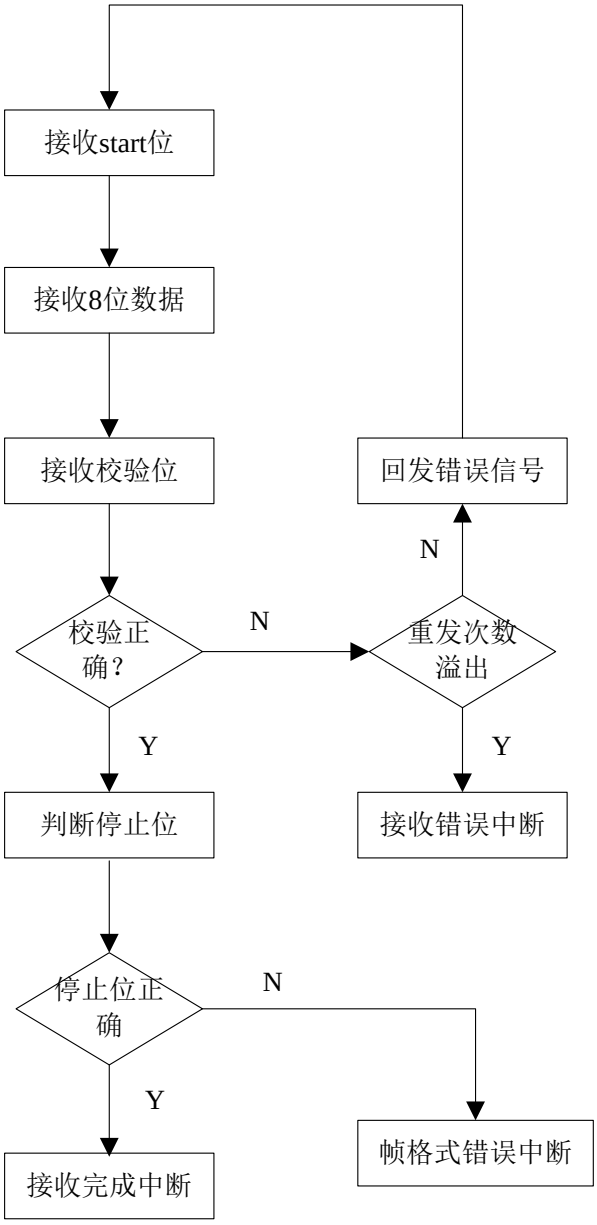


图 23-3 ISO7816 数据接收过程

23.4.2 数据发送

在 TXEN 开启的时候，软件只要向 TXBUF 写入数据，硬件在相应 IO 口空闲的条件下会自动发送数据，软件可以在发送过程中向 TXBUF 写入数据，硬件会在前一帧发送结束后继续发送下一帧。当进行数据发送时，内部输入端口自动关闭，即电路正常应用模式下不能收到自己发出的数据。要注意的是，由于本设计中只有一级缓存，软件两次写 TXBUF 的间隔不能太短，如果在状态机把数据装入移位寄存器开始发送之前又写 TXBUF，会把前面的数据冲掉。注意在发送时，软件至少要等硬件将前一笔数据移入移位寄存器以后才能写下一笔数据，软件可以监视 TX_FLAG，TX_FLAG 为 1 表示发送缓存寄

存器空，数据已经进入移位寄存器发送，可以向 TXBUF 写入下一笔数据。

7816 数据发送流程：

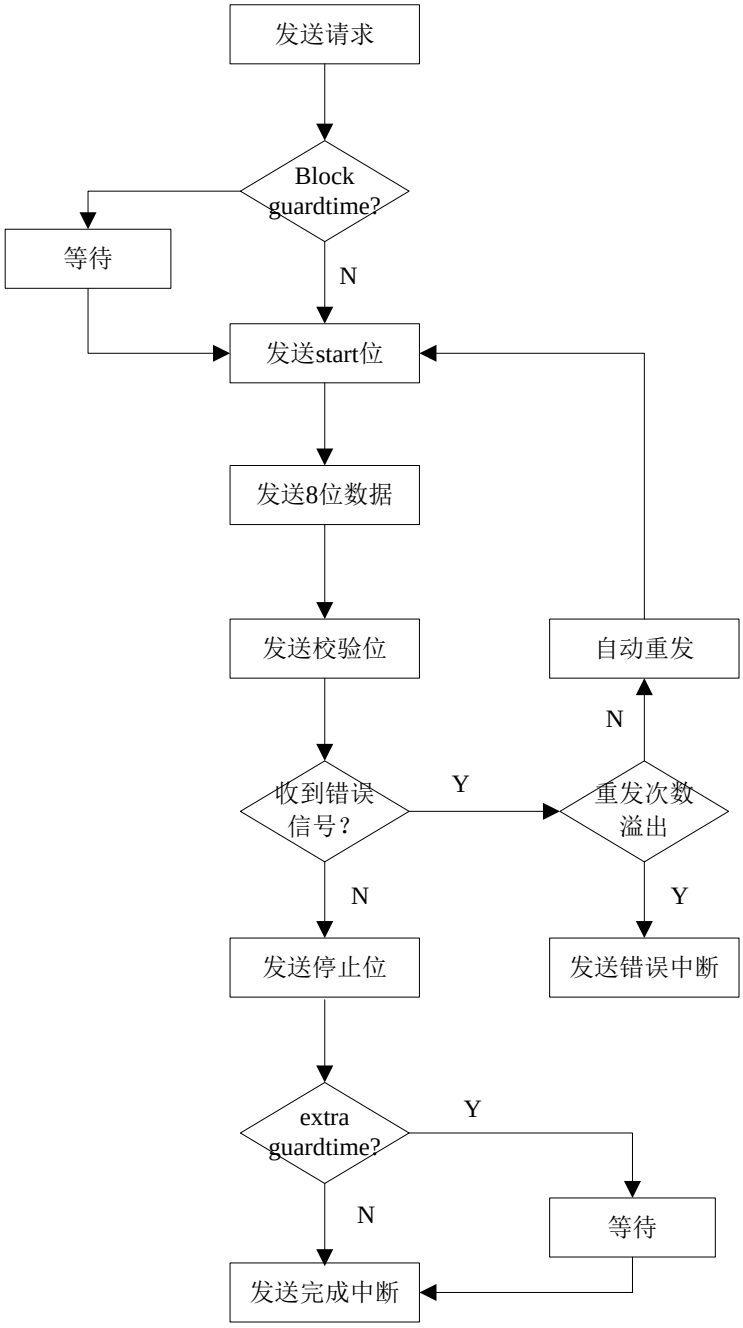


图 23-4 ISO7816 数据发送过程

23.4.3 使用 DMA 进行 7816 收发

当 7816 模块被使能后，7816 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 7816 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 7816 请求，并完成 RAM 和 SPI 之间的数据搬运。

应用举例：使用 DMA 进行 7816 接收

- 将 DMA 通道 0 配置为 U7816_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 7816 模块参数
- 使能 7816 模块，等待数据接收
- 收到数据后 7816 自动产生 DMA 请求
- DMA 响应请求，读取 7816 接收缓存寄存器，写入指定 RAM 地址

23.5 寄存器

offset 地址	名称	符号
ISO7816(模块起始地址:0x40010000)		
0x00000000	U7816 控制寄存器 （U7816 Control Register）	U7816_CR
0x00000004	U7816 帧格式寄存器 （U7816 Frame Format Register）	U7816_FFR
0x00000008	U7816 额外保护时间寄存器 （U7816 Extra Guard Time Register）	U7816_EGTR
0x0000000C	U7816 工作时钟分频寄存器 （U7816 Prescaler Register）	U7816_PSC
0x00000010	U7816 波特率寄存器 （U7816 Baud rate Generator Register）	U7816_BGR
0x00000014	U7816 数据接收缓存寄存器 （U7816 Receive Buffer）	U7816_RXBUF
0x00000018	U7816 数据发送缓存寄存器 （U7816 Transmit Buffer）	U7816_TXBUF
0x0000001C	U7816 中断使能寄存器 （U7816 Interrupt Enable Register）	U7816_IER
0x00000020	U7816 中断状态标志寄存器 （U7816 Interrupt Status Register）	U7816_ISR

23.5.1 U7816 控制寄存器（U7816_CR）

名称	U7816_CR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		TXEN	RXEN	CKOEN	HPUAT	HPUEN	RFUI
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5	TXEN	U7816 通道发送使能控制位 (Transmit Enable) 1：通道发送使能，可发送数据 0：通道发送禁止，不可发送数据，并关断输出端口，将 SCL 信号转化为低电平
4	RXEN	U7816 通道接收使能控制位 (Receive Enable) 1：通道接收使能，可接收数据 0：通道接收禁止，不可接收数据，并关断输入端口

位号	助记符	功能描述
3	CKOEN	U7816 时钟 CLK 输出使能控制位 (Clock output Enable) 1：7816 时钟输出使能 0：7816 时钟输出禁止
2	HPUAT	U7816 通道数据发送强上拉电阻自动有效控制位 (High-Pullup Automatically) 1：数据发送时上拉电阻自动有效，接收态上拉电阻无效 0:数据发送时上拉电阻自动有效功能禁止,上拉电阻由 HPUEN，LPUEN 控制
1	HPUEN	U7816 通道强上拉使能控制位 (High-Pullup Enable) 1：强上拉有效 0：强上拉无效
0	RFUI	保留位

23.5.2 U7816 帧格式寄存器（U7816_FFR）

名称	U7816_FFR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				SFREN	ERSW		ERSGD
位权限	U-0				R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BGTEN	REP_T	PAR		RFREN	TREPEN	RREPEN	DICONV
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-1	R/W-1	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	SFREN	Guard Time 发送长度控制位 (Send long Frame Enable) 1：Guard time 为 3 etu 0：Guard time 为 2 etu
10:9	ERSW	ERROR SIGNAL 宽度选择 (Error Signal Width) 11：ERROR SIGNAL 宽度为 1ETU; 10：ERROR SIGNAL 宽度为 1.5ETU; 01：ERROR SIGNAL 宽度为 2ETU; 00：ERROR SIGNAL 宽度为 2ETU;
8	ERSGD	ERROR SIGNAL 后 GUARDTIME 宽度选择（仅在发送时有效）(Error Signal Guard Time) 1：ERROR SIGNAL 后 GUARDTIME 为 1~1.5ETU。 0：ERROR SIGNAL 后 GUARDTIME 为 2~2.5ETU。 ERROR SIGNAL 宽度为整数 ETU 时 GUARDTIME 为 1.5 或 2.5ETU；ERROR SIGNAL 宽度为 1.5ETU 时 GUARDTIME 为 1 或 2ETU
7	BGTEN	BGT 控制位。控制接收->发送之间是否插入 BGT。BGT 是接收

位号	助记符	功能描述
		->发送之间需要的最小时间（block guard time enable） 1：BGT 使能，插入 Block guard time(12 etu); 0：BGT 禁止，不插入 Block guard time(12 etu);
6	REP_T	控制接收数据奇偶校验出错时自动重发次数 (Repeated Times) 1：3 次 0：1 次
5:4	PAR	奇偶校验类型选择 (Parity) 00：Even 01：Odd 10：Always 1 11：不校验，处理
3	RFREN	Guard Time 接收长度控制位 (Receive short Frame) 1：Guard time 为 1 etu 0：Guard time 为 2 etu
2	TREPEN	发送数据奇偶校验错的处理方式选择 (Transmit Repeat Enable) 1：收到奇偶校验出错标志（error signal），根据 T＝0 协议自动进行回发。在单一 byte 重复发送次数超过 REP_T 后，置 tx_parity_err 标志，进行中断 0：收到 Error signal 时不进行自动回发，置 tx_parity_err 标志，直接中断
1	RREPEN	接收数据奇偶校验错的处理方式选择 (Receive Repeat Enable) 1：奇偶校验错，根据 T=0 协议自动回发 ERROR SIGNAL。单一 BYTE 连续接收次数超过 REP_T 后，置 RX_PARITY_ERR 标志，进行中断 0：奇偶校验错，不自动发送 ERROR SIGNAL，置 RX_PARITY_ERR 标志，进行中断
0	DICONV	传输次序，编码方式选择 (bit Direction Conversion) 1：反向编码，先收发 MSB；(收发数据+校验位)反逻辑电平 0：正向编码，先收发 LSB ； (收发数据+校验位)正逻辑电平

23.5.3 U7816 额外保护时间寄存器（U7816_EGTR）

名称	U7816_EGTR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXEGT							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	TXEGT	发送时插入的 Extra Guard Time 时间（以 ETU 为单位） (Transmit Extra Guard Time)

23.5.4 U7816 工作时钟分频寄存器（U7816_PSC）

名称	U7816_PSC							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CLKDIV				
位权限	U-0			R/W-0 0011				

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4:0	CLKDIV	U7816 时钟输出分频控制寄存器(Clock Divider)，控制 7816 工作时钟分频数。 U7816 工作时钟与 APBCLK 的分频关系： $F_{7816}=F_{APBCLK}/(CLKDIV+1)$ 特殊情况：CLK_DIV 设置成 0 或 1 时，$F_{7816}=F_{APBCLK}/2$ 注：7816 协议规定的工作时钟范围是 1~5MHZ。

23.5.5 U7816 波特率寄存器（U7816_BGR）

名称	U7816_BGR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				PDIV[11:8]			
位权限	U-0				R/W-0001			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDIV[7:0]							
位权限	R/W-0111 0011							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	PDIV	U7816 预分频控制寄存器(Pre-Divider)，控制 7816 通信分频比（波特率） Baud ＝ F_{7816}/(PDIV ＋ 1) *注意：PDIV 最小可用值是 0x1，应用禁止配置 0x0*

23.5.6 U7816 数据接收缓存寄存器（U7816_RXBUF）

名称	U7816_RXBUF							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	RXBUF	U7816 数据接收缓存寄存器 (Receive Buffer)

23.5.7 U7816 数据发送缓存寄存器（U7816_TXBUF）

名称	U7816_TXBUF							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0

位号	助记符	功能描述
7:0	TXBUF	U7816 数据发送缓存寄存器 (Transmit Buffer)

23.5.8 U7816 中断使能寄存器（U7816_IER）

名称	U7816_IER							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RXIE	TXIE	LSIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	RXIE	数据接收中断使能位。对应 RXIF 中断标志位 (Receive Interrupt Enable) 1：当 RXIF 寄存器置位时产生接收完成中断 0：禁止接收完成中断
1	TXIE	数据发送中断使能位。对应 TXIF 中断标志位(Transmit Interrupt Enable) 1：当 TXIF 寄存器置位时产生发送完成中断 0：禁止发送完成中断
0	LSIE	线路状态中断使能位。对应 ERRIF 中断标志位(Line Status Interrupt Enable) 1：当 ERRIF 寄存器置位时产生线路错误中断 0：禁止线路错误中断

23.5.9 U7816 中断状态标志寄存器（U7816_ISR）

名称	U7816_ISR							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					WAIT_RPT	TXBUSY	RXBUSY
位权限	U-0					R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TPARERR	RPARERR	FRERR	OVERR

位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RXIF	TXIF	ERRIF
位权限	U-0					R-0	R-1	R-0

位号	助记符	功能描述
31:19	-	RFU：未实现，读为 0
18	WAIT_RPT	U7816 接口发送了错误信号，正在等待对方重发数据；(Waiting for Repeat flag) 状态机进入发送错误信号状态时置位，收到数据起始位或者进入发送状态时硬件清零； 软件只读。
17	TXBUSY	发送数据忙标志。（发送完成后自动清零）(Transmission busy flag) 1：处于数据发送状态，发送移位寄存器正在发送数据。（开始发送起始位置 1，停止位中间清零） 0：数据发送空闲
16	RXBUSY	接收数据忙标志。（接收完成后自动清零）(Receiving busy flag) 1：处于数据接收状态，接收移位寄存器正在接收数据。（收到起始位置 1，收到停止位清零，若接收数据出错需重发，则回发 error signal 时清零。即数据及校验位接收之后，无论是否需要重发，都需要及时清除该标志） 0：数据接收空闲
15:12	-	RFU：未实现，读为 0
11	TPARERR	发送数据奇偶校验错误标志位。硬件置位，写 1 清零 (Transmit Parity Error,write 1 to clear)
10	RPARERR	接收数据奇偶校验错误标志位。硬件置位，写 1 清零 (Receive Parity Error flag,write 1 to clear)
9	FRERR	接收帧格式错误标志位。硬件置位，写 1 清零 (Frame Error flag,write 1 to clear) 1：帧格式有错误，接收到的 frame 字节长度有误或接收到的 frame 或者 stop 位有误 0：接收数据时无奇偶校验错误
8	OVERR	接收溢出错误标志位。硬件置位，写 1 清零 (Receive Overflow Error,write 1 to clear) 1：接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效。原接收缓冲寄存器内数据被新覆盖 0：无溢出错误
7:3	-	RFU：未实现，读为 0
2	RXIF	接收完成标志(Receive interrupt flag)，U7816 接口控制器每收到 1byte 数据，根据接收的通道相应发出一次中断。硬件置位，读数据接收缓冲寄存器清零 1：接收到 1byte 数据，数据接收缓冲器满 0：未接收到数据，数据接收缓冲器空
1	TXIF	发送缓冲区空标志(Transmit interrupt flag)，上电复位后此标志就自动置位，表示缓冲区空，可以写入数据。软件写入数据后标志自动清除，数据从发送缓存移入移位寄存器后置 1 1：数据发送缓冲器空 0：数据发送缓冲器内有数据待发送
0	ERRIF	错误标志(Error interrupt flag)，寄存器配置出错或传输过程中出错。此 bit 是 TPARERR、RPARERR、FRERR、OVERR 的或。

位号	助记符	功能描述
		软件通过清除以上错误标志寄存器来清除此 bit。

24 DMA

24.1 概述

- 7通道外设PDMA，支持Peripherals<>RAM传输
- 1通道存储器MDMA，支持Flash<>RAM传输
- 外设DMA传输由外设请求触发，DMA工作期间不影响CPU运行
- 外设通道最大传输长度65536字节（64KB），支持byte/half-word/word传输
- Flash->RAM通道最大传输长度8192字节，只支持word传输
- 支持Flash连续编程（RAM->Flash），需要预先进行擦除，一次编程固定为256字节
- RAM指针递增、递减
- 可产生半程中断和全程中断
- 通道优先级可配置（4级优先级）

24.2 工作原理

外设 DMA 为 Peripheral<>RAM 通道，采用外设请求触发方式进行数据传输，每个外设通道都可以支持外设->RAM 或者 RAM->外设的数据传输，并且根据目标外设类型的不同，自适应选择 byte/half-word/word 传输方式。DMA 作为 Master，在收到 request 后将发起 AHB transactions 进行数据操作，外设目标地址根据通道接入选择自动定位，RAM 目标地址则根据寄存器配置定位。

每个 channel 可以从多个外设中选择一个作为 source 或 destination，同时软件可以设置通道优先级，当两个通道同时要访问 RAM 时，由优先级决定谁先访问，另一个通道将被挂起，直到优先通道访问完毕。

外设请求可以是准备发送（RAM/Flash->Peripheral）或接收完成（Peripheral->RAM），数据传输通过 AHB 总线完成，当 DMA 访问外设时，CPU 对同一个外设的访问将引起冲突，哪个 Master 访问被挂起取决于 BusMatrix 设置的仲裁优先级。这里需要注意的是，由于大部分外设都被挂在 APB 总线上，APB 映射到 AHB 仅为一个 slave，因此当 DMA 访问 APB 中任意外设时，CPU 即使访问 APB 下的其他外设，也同样会引起总线仲裁。通过 DIR 寄存器可以配置每个通道的传输方向，软件必须保证传输方向配置与实际挂载到这个通道上的外设请求相一致。比如通道 1 当前挂载的外设请求是 UART0 接收，则必须将 DIR 寄存器配置为 0（数据从外设读出，写入 RAM），每次 UART0 接收完一帧数据，将发送 RXD0 请求给 DMA，DMA 响应请求后，从 UART0 接收缓存寄存器读取数据，如果 DIR 被错误的配置为 1，则 DMA 对 UART0 接收缓存寄存器的写操作将被 UART0 忽略。

软件可设置 DMA 的存储器指针，用于配置 DMA 传输的起始地址，可以选择指针递增或递减方式。另有 TRFLEN 寄存器配置传输次数，根据起始地址和传输次数，计算得到终止地址，当存储器指针指向终止地址时，本次传输结束，关闭通道。

当 channel 被使能后，DMA 就准备好接受通道所选中的外设请求。当配置传输长度一半的字节被传输后，一个 HTIF（Half transfer interrupt flag）中断置位；当配置传输长度全部完成后，TCIF（Transfer complete interrupt flag）中断置位。上述中断都可以被相应的中断使能寄存器屏蔽。

在 DMA 一个完整 transfer block 完成之前，软件随时可以关闭 channel 使能，此时 DMA 将被挂起，如果软件此后重新使能通道，则 DMA 继续执行之前挂起的操作。

24.3 结构框图

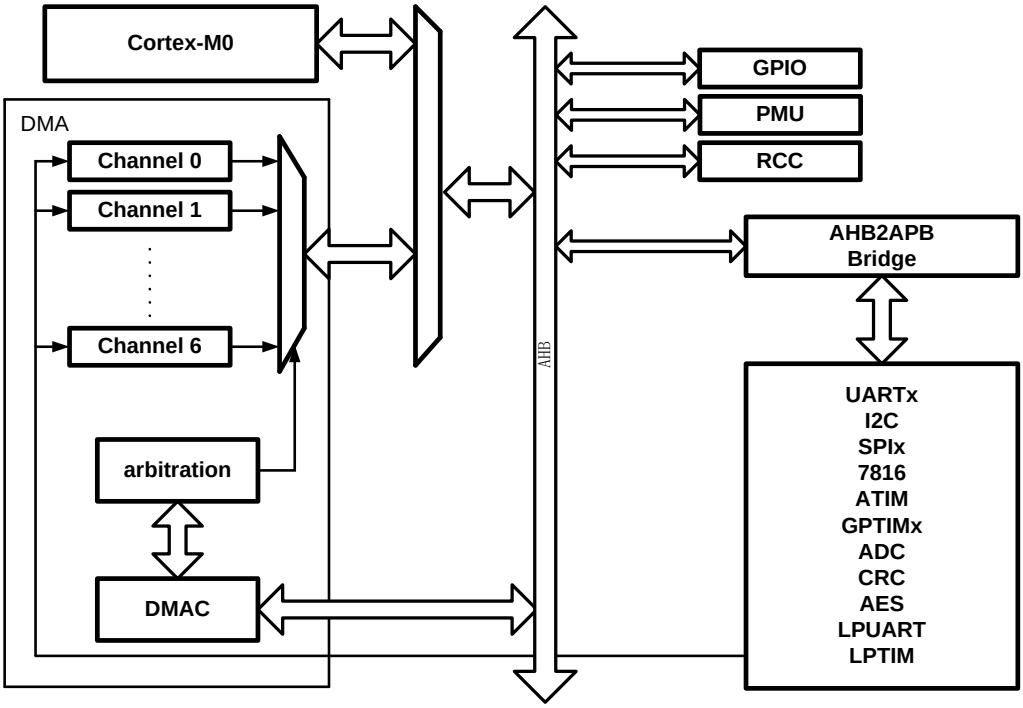


图 24-1 DMA 结构框图

24.4 工作流程

DMA 寄存器配置：

图 24-2 DMA 寄存器配置

DMA 对请求响应分成两部分处理：通道请求处理过程和数据搬运过程。

- 通道请求处理

  a) DMA 接受到请求，跳到步骤 b

  b) 判断是否有其他通道正在搬运数，若有，则停留在步骤 b 直至其他通道当次搬运完成；若无，进一步判断是否有其他同时置起的请求信号，若有，则判断当前通道优先级是否高于其他通道，若是，则跳到步骤 c 并向数据搬运过程发起请求，若否，则停留在步骤 b 直至其他通道当次搬运完成

  c) 并等待数据搬运完成响应信号，得到响应则，跳到步骤 d，否则停在步骤 c

  d) 数据搬运长度+1，判断是否达到设定长度，若是则产生通道使能关闭脉冲；判断请求是否释放，若是，则跳到步骤 a，若否，则停留在步骤 d 判断数据传输达到设定长度，否则跳到步骤 a

- 数据搬运

  a) 等待通道请求处理过程发起请求

  b) 向 HADDR 写源地址

  c) 向 HADDR 写目的地址，同时读取 HRDATA

  d) 将读到的 HRDATA 数据写到 HWDATA

  e) 向通道请求处理过程发出搬运完成响应，并跳到步骤 a

DMA 工作的流程如下图所示：

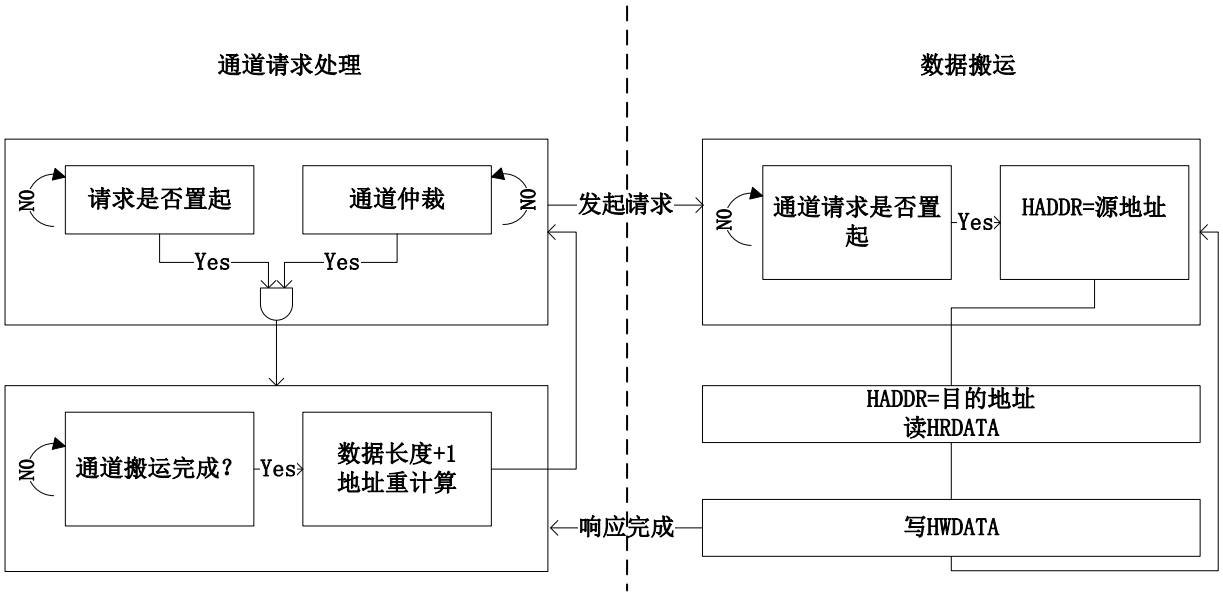


图 24-3 DMA 工作流程

24.5 访问带宽

DMA 外设通道支持字节/半字/字访问，每个通道都可以通过通道控制寄存器中的 BDW 位来配置传输带宽。

24.6 通道控制

24.6.1 DMA 请求映射

DMA 共有 7 个优先级可配的外设通道，每个通道可接受 8 个请求响应，根据每个通道的配置寄存器选择其中一个请求送入通道控制器，通道控制器根据各个通道的 busy 状态和优先级选择其中一个通道请求进行响应处理，外设请求映射如下。

编号	外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
0	***ADC***	ADC				ADC		
1	***SPI1***				SPI1_RX	SPI1_TX	SPI1_RX	SPI1_TX
2	***SPI2***			SPI2_RX		SPI2_TX	SPI2_RX	SPI2_TX
3	***UART0***		RXD0	TXD0	RXD0	TXD0		
4	***UART1***				RXD1	TXD1	RXD1	TXD1
5	***UART4***			RXD4	TXD4			
6	***UART5***					RXD5		TXD5
7	***LPUART0***	LPUART0_RX	LPUART0_TX		LPUART0_RX		LPUART0_TX	
8	***LPUART1***	LPUART1_TX		LPUART1_RX			LPUART1_RX	LPUART1_TX
9	***U7816***						U7816RX	U7816TX
10	***I2C***		I2C_RX	I2C_TX		I2C_RX		I2C_TX
11	***AES***	AES_IN	AES_OUT					
12	***CRC***	CRC						
13	***ATIM***	ATIM_CH1	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV		
14	***GTIM1***	GTIM1_CH1	GTIM1_CH2	GTIM1_CH3	GTIM1_CH4			GTIM_TRIG GTIM1_UEV
15	***GTIM2***	GTIM2_CH1	GTIM2_CH2	GTIM2_CH3	GTIM2_CH4		GTIM2_TRIG GTIM2_UEV	
16	***LPTIM32***		LPT32_CH1				LPT32_CH2	
		8	8	8	8	8	8	8

表 24-1DMA 通道映射

注意，ATIM_TRIG、ATIM_COM和ATIM_UEV请求，都仅针对高级定时器的DMA Burst模式，即这些请求到来时，DMA都是要访问ATIM的DMAR寄存器，因此这三个请求可以合并到一个通道上完成；同理，通用定时器的GTIMx_TRIG和GTIMx_UEV请求也可以合并到一个通道上。

外设请求映射通过CHxSSEL寄存器配置，上表中从上到下分别表示CHxSSEL=0~7情况下有效的外设请求信号。比如针对通道0，当CH0SSEL=2时，被选中的外设请求是EUART1_TX，即EUART1的数据发送DMA请求被连接到DMA通道0的请求输入。

24.6.2 通道优先级

DMA 总共有 7 个外设通道，每个通道的优先级别可以通过寄存器配置为：very high,high,low,very low。当多个通道配置为相同优先级别时，通道序号越大，优先级别越低。

DMA 每搬运完一次数据都会重新进行通道请求选择，假设通道 0 传输长度为 3，通道 1 传输长度为 2。当通道 0 完成第二次传输准备进行第三次数据搬运时，通道 1 请求响应置起，这时通道控制器根据通道优先级切换至通道 1 数据搬运，直至通道 1 数据全部搬运完成，通道寄存器再切换回通道 0 完成剩下的数据搬运。

24.6.3 传输方向定义

在 DMA 通道定义规则中，_RX 表示 DMA 从外设读取数据，写入 RAM，_TX 表示 DMA 从 RAM/Flash 读取数据，写入外设。

软件在配置每个通道的外设分配之后，还需要配置 CHx_DIR 寄存器设定通道传输方向，错误的方向设置会导致 DMA 无法正常工作。

24.6.4 循环模式

外设 DMA 通道支持循环模式（Circular mode）。循环模式下，当 CHxTSIZE 寄存器定义的传输长度完成后，DMA 不会自动停止，而是返回 RAM 指针寄存器定义的起始地址，继续传输。DMA 的半程中断和全程中断还是会正常置起，DMA 不会终止传输，直到软件关闭通道。

通过置位 CHxCTRL.CIRC 寄存器使能循环模式。

存储 DMA 通道不支持循环模式。

24.7 寄存器

offset 地址	名称	符号
DMA(模块起始地址:0x40000400)		
0x00000000	DMA 全局控制寄存器 （DMA Global Control Register）	DMA_GCR
0x00000004	通道 0 控制寄存器 （Channel 0 Control Register）	DMA_CH0CR
0x00000008	通道 0 存储器指针寄存器 （Channel 0 Memory Address Register）	DMA_CH0MAD
0x0000000C	通道 1 控制寄存器 （Channel 1 Control Register）	DMA_CH1CR
0x00000010	通道 1 存储器指针寄存器 （Channel 1 Memory Address Register）	DMA_CH1MAD
0x00000014	通道 2 控制寄存器 （Channel 2 Control Register）	DMA_CH2CR
0x00000018	通道 2 存储器指针寄存器 （Channel 2 Memory Address Register）	DMA_CH2MAD
0x0000001C	通道 3 控制寄存器 （Channel 3 Control Register）	DMA_CH3CR
0x00000020	通道 3 存储器指针寄存器 （Channel 3 Memory Address Register）	DMA_CH3MAD
0x00000024	通道 4 控制寄存器 （Channel 4 Control Register）	DMA_CH4CR
0x00000028	通道 4 存储器指针寄存器 （Channel 4 Memory Address Register）	DMA_CH4MAD
0x0000002C	通道 5 控制寄存器 （Channel 5 Control Register）	DMA_CH5CR
0x00000030	通道 5 存储器指针寄存器 （Channel 5 Memory Address Register）	DMA_CH5MAD
0x00000034	通道 6 控制寄存器 （Channel 6 Control Register）	DMA_CH6CR
0x00000038	通道 6 存储器指针寄存器 （Channel 6 Memory Address Register）	DMA_CH6MAD
0x0000003C	通道 7 控制寄存器 （Channel 7 Control Register）	DMA_CH7CR
0x00000040	通道 7 Flash 指针寄存器 （Channel 7 Flash Address Register）	DMA_CH7FLSAD
0x00000044	通道 7 RAM 指针寄存器 （Channel 7 RAM Address Register）	DMA_CH7RAMAD
0x00000048	DMA 中断状态标志寄存器 （DMA Interrupt Status Register）	DMA_ISR

24.7.1 DMA 全局控制寄存器（DMA_GCR）

名称	DMA_GCR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	DMA_GCR							
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						ADDRERR_EN	EN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	**-**	RFU：未实现，读为 0
1	**ADDRERR_EN**	DMA 错误地址中断使能 (DMA address error interrupt enable) 1：允许错误地址中断 0：禁止错误地址中断
0	**EN**	DMA 全局使能 (DMA enable) 1：DMA 使能 0：DMA 关闭

24.7.2 通道 x 控制寄存器（DMA_CHxCR）

名称	DMA_CHxCR(x=0,1,2,3,4,5,6)							
Offset	0x00000004 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TSIZE[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		INC	SSEL		
位权限	U-0		R/W-00		R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	DIR	BDW		CIRC	FTIE	HTIE	EN
位权限	U-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:16	**TSIZE**	Channelx 传输长度，最大 65536 次传输 *注意：禁止将 CHxTSIZE 配置为 0*
15:14	**-**	RFU：未实现，读为 0
13:12	**PRI**	Channelx 优先级 (Channels Priority) 00：Low 01：Medium 10：High 11：Very High
11	**INC**	RAM 地址增减设置 (Channelx Ram address Incremental) 1：RAM 地址递增 0：RAM 地址递减

位号	助记符	功能描述
10:8	**SSEL**	Channelx 外设请求映射 (Channelx request Source Select) 每个通道可以接受 8 个外设请求，外设请求的映射参见 24.6.1DMA 请求映射
7	**-**	RFU：未实现，读为 0
6	**DIR**	通道传输方向 (Direction) 0：从外设读取数据写入 RAM 1：从 RAM 读取数据写入外设
5:4	**BDW**	传输带宽设置 (Bandwidth) 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	**CIRC**	循环缓冲模式 (Circular mode enable) 0：关闭循环模式 1：使能循环模式
2	**FTIE**	Channelx 传输完成中断使能 (Channelx Finished-Transfer Interrupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**HTIE**	Channelx 半程传输完成中断使能 (Channelx Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channelx 使能 (Channelx Enable) 1：启动通道 0 0：关闭通道 0

24.7.3 通道 x 存储器指针寄存器（DMA_CHxMAD）

名称	DMA_CHxMAD(x=0,1,2,3,4,5,6)							
Offset	0x00000008 + x*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MEMAD[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MEMAD[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MEMAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**MEMAD**	Channelx 存储器指针地址，DMA 传输启动前软件向此寄存器写入存储器目标地址。(Channel x Memory Address pointer) 当指针指向空地址时，DMA 访问将触发 hardfault

位号	助记符	功能描述
		当指针指向 Flash 时，禁止向 Flash 写入数据。 软件可以查询当前 DMA 传输的目标存储器地址。

24.7.4 通道 7 控制寄存器（DMA_CH7CR）

名称	DMA_CH7CR							
Offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				TSIZE[11:8]			
位权限	U-0				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		-	DIR	RI	FI
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					FTIE	HTIE	EN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:28	**-**	RFU：未实现，读为 0
27:16	**TSIZE**	Channel7 传输长度 (Channel 7 Transfer Size)，1-4096 次传输，仅在 Flash->RAM 传输时有效，RAM->Flash 传输为固定长度 64 次传输
15:14	**-**	RFU：未实现，读为 0
13:12	**PRI**	Channel7 优先级 (Channel 7 Priority) 00：Low 01：Medium 10：High 11：Very High
11	**-**	RFU：未实现，读为 0
10	**DIR**	Channel7 传输方向 (Channel 7 Direction) 1：Flash->RAM 传输 0：RAM->Flash 传输
9	**RI**	Channel7 RAM 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Ram Incremental) 1：RAM 地址递增 0：RAM 地址递减
8	**FI**	Channel7 Flash 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Flash Incremental) 1：Flash 地址递增 0：Flash 地址递减
7:3	**-**	RFU：未实现，读为 0
2	**FTIE**	Channel7 传输完成中断使能 (Channel 7 Finished-Transfer Interupt Enable) 1：使能传输完成中断 0：关闭传输完成中断

位号	助记符	功能描述
1	**HTIE**	Channel7 半程传输完成中断使能 (Channel 7 Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channel7 使能 (Channle 7 Enable) 1：启动通道 0 0：关闭通道 0

24.7.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）

名称	DMA_CH7FLSAD							
Offset	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	FLSAD[14:8]						
位权限	U-0	R/W-000 0000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FLSAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:15	**-**	RFU：未实现，读为 0
14:0	**FLSAD**	Channel7 Flash 指针地址（word 地址），DMA 传输启动前软件向此寄存器写入 Flash 目标地址，DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 Flash Address pointer) 软件可以查询当前 DMA 传输的目标 Flash 地址 此寄存器低位（bit5-0）仅在 Flash->RAM 传输中有效，RAM->Flash 传输中默认对齐 Flash 的 half-sector 起始地址

24.7.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）

名称	DMA_CH7RAMAD							
Offset	0x00000044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RAMAD[11:8]			
位权限	U-0				R/W-0000			

名称	DMA_CH7RAMAD							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	**RAMAD**	Channel7 RAM 字指针地址，DMA 传输启动前软件向此寄存器写入 RAM 目标地址（word 地址），DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 RAM Address pointer) 软件可以查询当前 DMA 传输的目标 RAM 地址

24.7.7 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
Offset	0x00000048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADDRERR
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHFT[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHHT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:17	-	RFU：未实现，读为 0
16	**ADDRERR**	DMA 传输地址错误标志，硬件置位，软件写 1 清零 (DMA address error flag,write 1 to clear)
15:8	**CHFT**	DMA 通道 x 传输完成标志(DMA channel Finished-Transfer Flag,write 1 to clear)，硬件置位，软件写 1 清零 1：对应通道传输完成 0：对应通道传输未完成
7:0	**CHHT**	DMA 通道 x 传输半程标志 (DMA channel Half-Transfer Flag,write 1 to clear)，硬件置位，软件写 1 清零

25 CRC

25.1 概述

循环冗余校验(Cyclic Redundancy Check，CRC)是最为常用的计算机和仪表数据通信的校验方法，FM33LC0XX中CRC计算单元为完全独立模块，通过软件控制可进行7816、I2C、UART和SPI模块有串行数据流接口的收发CRC计算和校验。

此外，CRC也可进行Flash内容的完整性校验。通过结合DMA，可以实时计算Flash中程序内容的CRC结果，并生成一个完整性签名，与程序一同保存在Flash中。通过校验这个CRC签名，可以验证Flash内容是否正确、完整。

- 支持7/8/16/32位CRC，支持任意多项式
- 初值可设置
- CRC快速算法，1个时钟周期完成8bit CRC运算，4个时钟周期完成32bit CRC运算
- 支持输入输出数据顺序自动调整（以字节、半字、或全字为单位）
- 支持对输出结果异或

25.2 软件配置过程

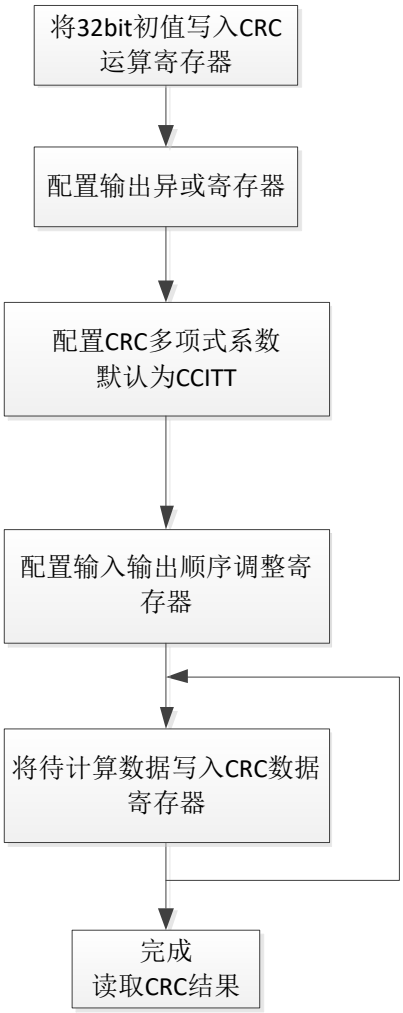


图 25-1CRC 运算流程图

CRC 配置及计算流程如下：

- CRC开始计算的时候，配置运算移位寄存器中的初始值，范围是0x0000_0000~0xFFFF_FFFF。
- 配置输出异或寄存器CRC_XOR
- 软件需配置好输入REFLECTIN处理使能；输出REFLECTOUT和XOROUT处理使能
- 软件将需要计算CRC码的数据放入数据寄存器(CRC_DR)，然后自动开始计算逐次移位。
- 计算完毕后，结果数据回写到数据寄存器，软件根据当前计算状态BUSY位来判断是否能取结果：
- 若多项式为7bit多项式则结果为CRC_DR[6:0]，若多项式为8bit多项式则结果为CRC[7:0]，若多项式为16bit多项式则结果为CRC_DR[15:0]，若多项式为32bit多项式则结果为CRC_DR[31:0]；
- 计算完前一次CRC后，数据寄存器中会保留前一次结果，作为后续数据的移位寄存器初始值。在多次连续触发CRC计算后，软件最终读取的是累积计算的完整数据的CRC值。

25.3 Golden 数据

提供 Golden 数据表格供应用中测试及校验使用。

多项式	输入序列	初始值(16 进制)		
		全 0	全 F	6363
		CRC 计算结果（16 进制）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96
CRC-16	5A5A	5DD9	DDD4	9696
	11223344	7D35	7D11	4698
CRC-CCITT	5A5A	1ACB	07C4	1877
	11223344	DD33	59F3	DD06

表 25-1CRC golden 数据

25.4 DMA 接口

CRC与DMA之间通道为单向的（RAM->CRC）。CRC模块可以通过DMA模块读取并校验RAM数据，其工作流程如图所示。CRC向DMA发起请求，DMA接收请求后，读取RAM并将数据写入CRC模块的CRCDR寄存器中。CRC模块接收到数据后，撤销DMA请求并开始计算校验值，校验完成后，CRC模块重新置起DMA请求。

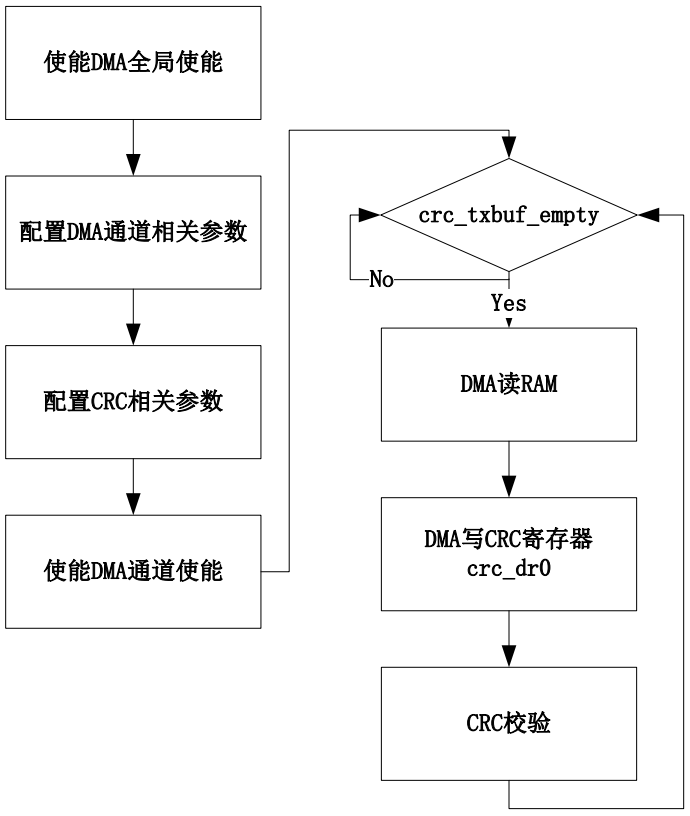


图25-2使用DMA对RAM中的数据进行CRC运算

25.5 Flash 数据完整性校验

通过CRC和DMA可以进行Flash内容的完整性校验，解决方案是：

- 首先通过DMA将Flash内容搬运到RAM中
- 通过DMA将RAM中缓存的Flash数据搬运到CRC模块进行计算
- 重复以上过程直到完成所有Flash数据的校验

25.6 寄存器

offset 地址	名称	符号
CRC(模块起始地址:0x40018000)		
0x00000000	CRC 数据寄存器 （CRC Data Register）	CRC_DR
0x00000004	CRC 控制状态寄存器 （CRC Control Register）	CRC_CR
0x00000008	CRC LFSR 寄存器 （CRC Linear Feedback Shift Register）	CRC_LFSR
0x0000000C	CRC 输出异或寄存器 （CRC output XOR Register）	CRC_XOR
0x0000001C	CRC 多项式寄存器 （CRC Polynominal Register）	CRC_POLY

25.6.1 CRC 数据寄存器（CRC_DR）

名称	CRC_DR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**DR**	CRCDR 用于作为数据输入寄存器，并且在运算结束后保存 CRC 计算结果。(CRC Data Register) 作为输入时：若 word 操作使能，则对 CRCDR[31:0]进行计算，共 4 次 byte 运算（由低到高）；否则对 CRCDR[7:0]进行计算，共 1 次 byte 运算。 保存结果时：若为 7 位多项式结果保存在 CRCDR[6:0]，若为 8 位多项式结果保存在 CRCDR[7:0]，若为 16 位多项式结果保存在 CRCDR[15:0]，若为 32 位多项式结果保存在 CRCDR[31:0]。

25.6.2 CRC 控制状态寄存器（CRC_CR）

名称	CRC_CR							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OPWD	PARA
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-10	

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	**OPWD**	WORD 操作使能 (Operation by Word) 0：字节操作，CRC 计算仅针对 CRCDR 最低字节进行 1：字操作，CRC 计算针对 CRCDR 全部 4 字节进行
8	**PARA**	CRC 快速计算使能 (Parallel Calculation) 0：串行运算，计算 1 个字节需要 8 个时钟周期 1：并行计算，计算 1 个字节需要 1 个时钟周期
7:6	**RFLTIN**	CRC 输入反转控制 (Reflected Input) 00：输入不反转 01：输入按字节反转 10：输入按半字反转 11：输入按字反转 例如：计算数据为 0x11223344， 如果 RFLTIN==01，则将数据变为 0x8844CC22，再进行计算 如果 RFLTIN==10，则将数据变为 0x448822CC，再进行计算 如果 RFLTIN==11，则将数据变为 0x22CC4488，再进行计算
5	**RFLTO**	CRC 输出反转控制 (Reflected Output) 0：输入不反转 1：输入按字节反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 0x1234，则输出的结果为 0x2C48 如果 RFLTO==0，则直接输出 0x1234 注意：此结果不一定为最终输出结果，还需要看 XOR 是否为 1，详见本寄存器 bit2 说明
4	**RES**	CRC 结果标志位，只读 (Result Flag) 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 (Busy) 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 (Output XORed with CRC_XOR register enable) 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**SEL**	CRC 多项式位宽选择 (Polynomial width Selection) 00：32 位 01：16 位 10：8 位

位号	助记符	功能描述
		11：7 位

25.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**LFSR**	CRC 线性反馈移位寄存器 (Linear Feedback Shift Register) 运算开始前可以由软件写入 CRC 初始值

25.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	XOR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	XOR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XOR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XOR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**XOR**	CRC 运算结果异或寄存器 (eXclusive OR) 当 CRC_CR.XOR 为 1 时，CRC 结果输出前将异或此寄存器的数据。

25.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	POLY[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	POLY[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	POLY[15:8]							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0010 0001							

位号	助记符	功能描述
31:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

26 高级定时器（ATIM）

26.1 概述

FM33LC0XX包含一个高级定时器。

高级定时器包含一个16bit自动重载计数器及一个可编程预分频器。

高级定时器可以支持多种应用，包括如捕捉、输出比较、PWM、带死区插入的互补PWM。

26.2 主要特性

- 16bit向上、向下、双向自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- 4个独立通道可用于输入捕捉、输出比较、PWM、单脉冲输出
- 可编程死区插入的互补输出
- 独立工作时钟，最高频率120MHz
- 重复计数器，支持定时器多个循环后更新状态
- 两路刹车引脚输入、比较器刹车、SVD刹车，刹车信号滤波和极性选择，刹车信号组合配置
- 支持在以下事件发生时产生中断或DMA事件

  - 计数器上/下溢出，计数器初始化（软件或硬件 trigger）

  - Trigger 事件（计数器启动、停止、初始化、内外部触发）

  - 输入捕捉

  - 输出比较

  - 刹车输入
- 支持增量正交编码器和霍尔传感器

术语：

- OCxREF信号为高称为有效电平（active），为低称为无效电平（inactive）
- 空闲模式（IDLE mode）：相对运行模式而言，指发生刹车事件时MOE=0
- 输出禁止（output disables）：GPIO输出使能关闭，不受TIMER驱动
- Off-state：GPIO输出使能打开，但是TIMER输出无效状态
- 无效状态（inactive state）：互补通道中的一路或两路输出无效电平时的状态

26.3 结构框图

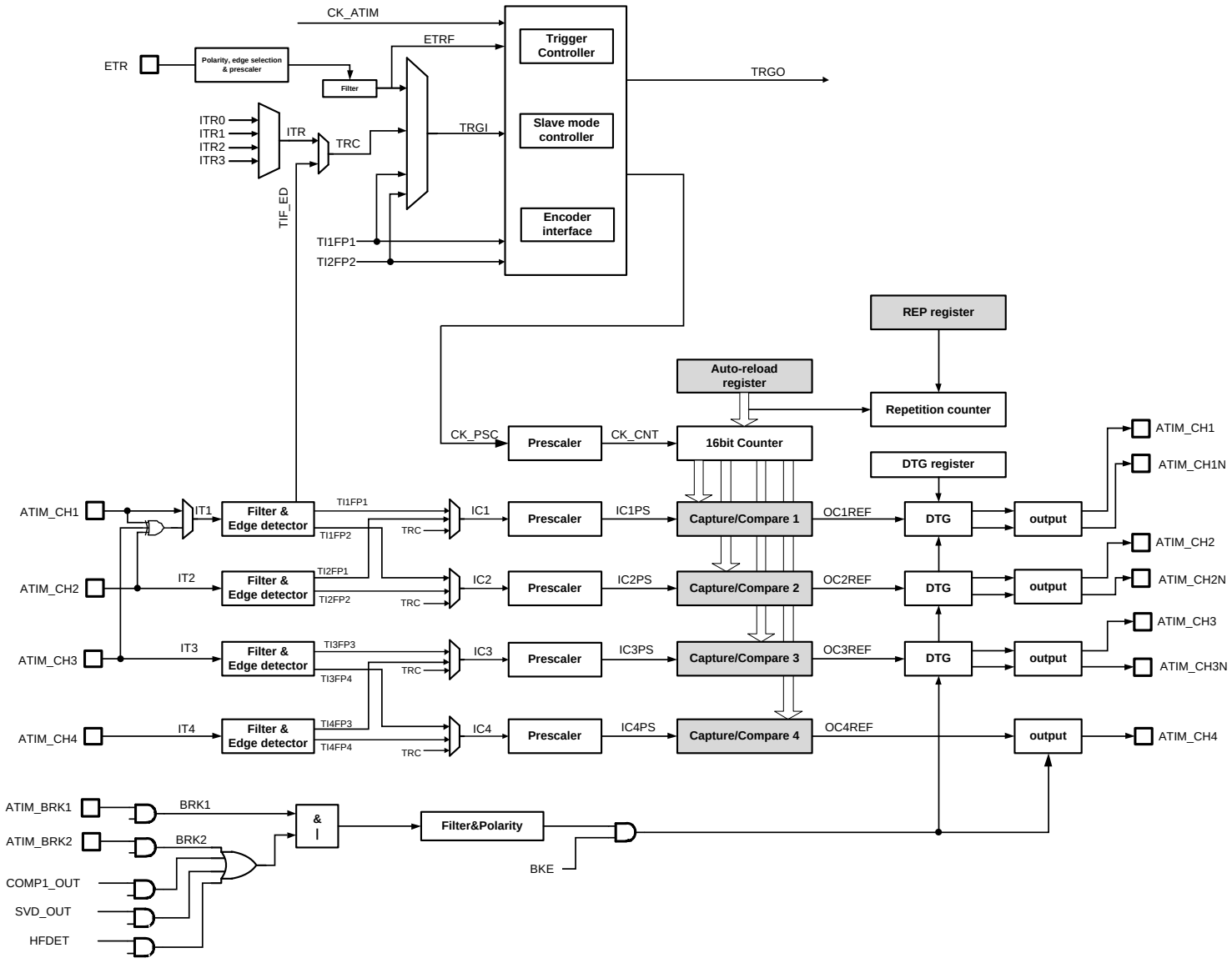


图 26-1 高级定时器结构框图

26.4 功能描述

26.4.1 定时单元

高级定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（ATIM_CNT）
- 预分频寄存器（ATIM_PSC）
- 自动重载寄存器（ATIM_ARR）
- 重复计数寄存器（ATIM_RCR）

ARR包含预装载功能，该功能通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=0时，对ARR寄存器执行写入，写入数据将直接传入到影子寄存器；当ARPE=1时，对ARR寄存器执行写入的数据在update event（ATIM_CNT上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发ARR更新（UEV）。

ATIM_CNT工作时钟由ATIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

ATIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际不改写影子寄存器，只有当新的update event到来时，才会从PSC更新至影子寄存器。因此在CNT计数过程中，软件可以实时改写PSC，而新的预分频比将在下一更新事件发生时被采用。

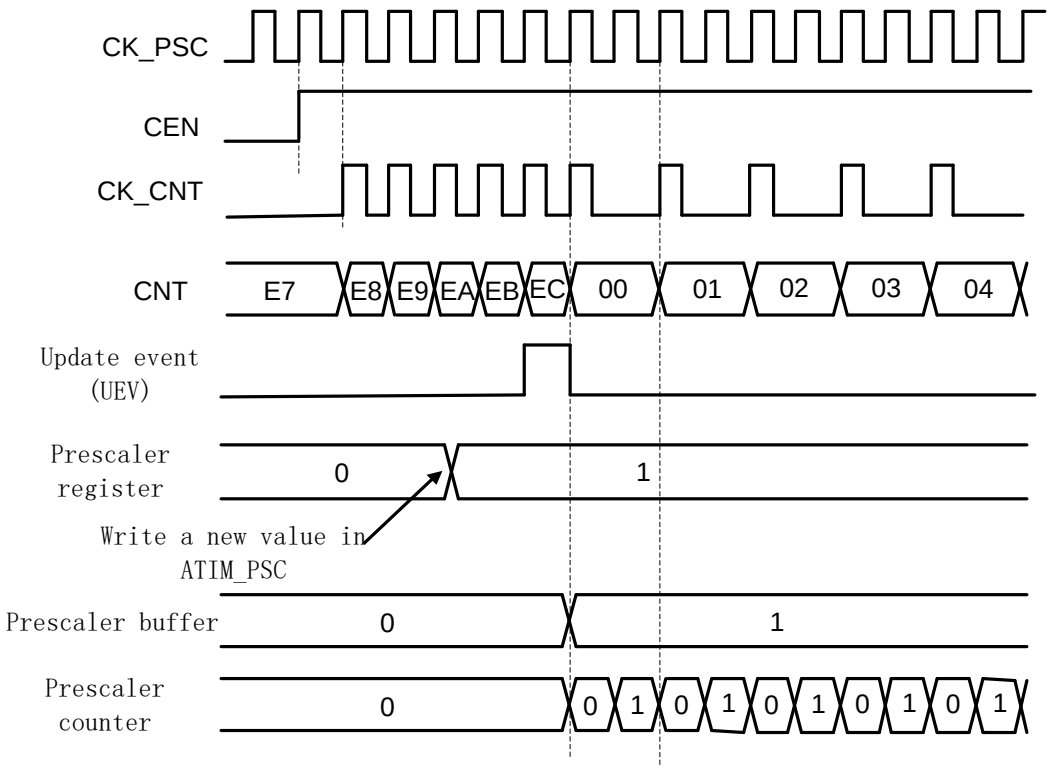


图 26-2 预分频从 1 变为 2 的波形

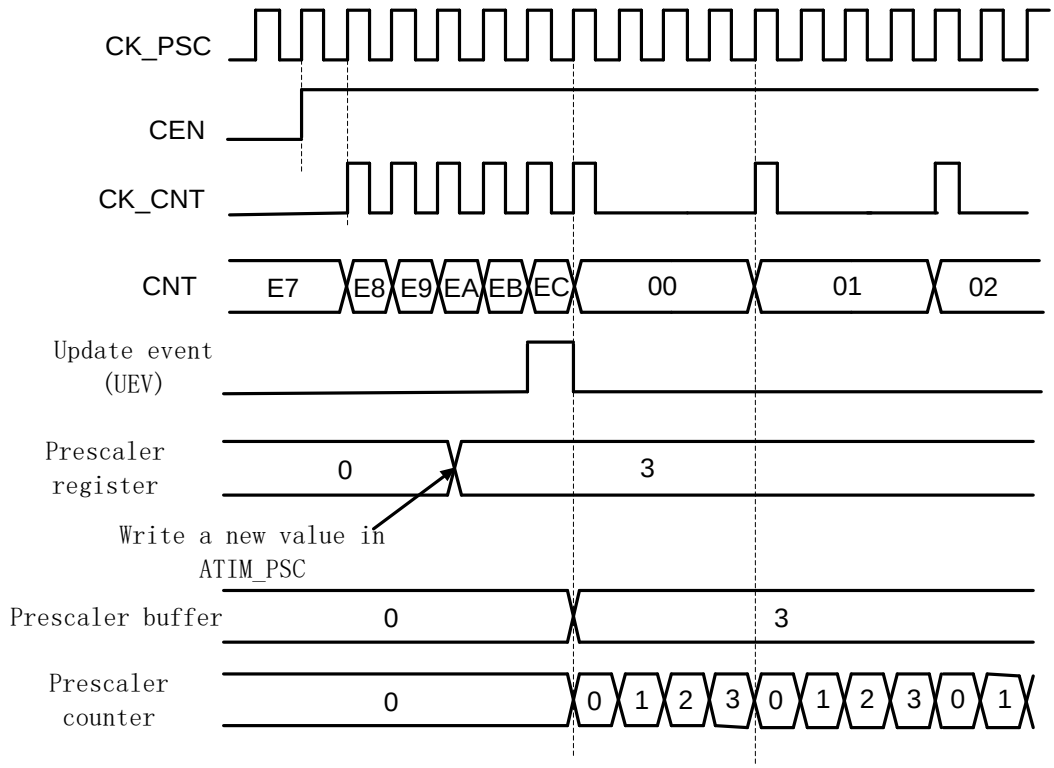


图 26-3 预分频从 1 变为 4 的波形

26.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

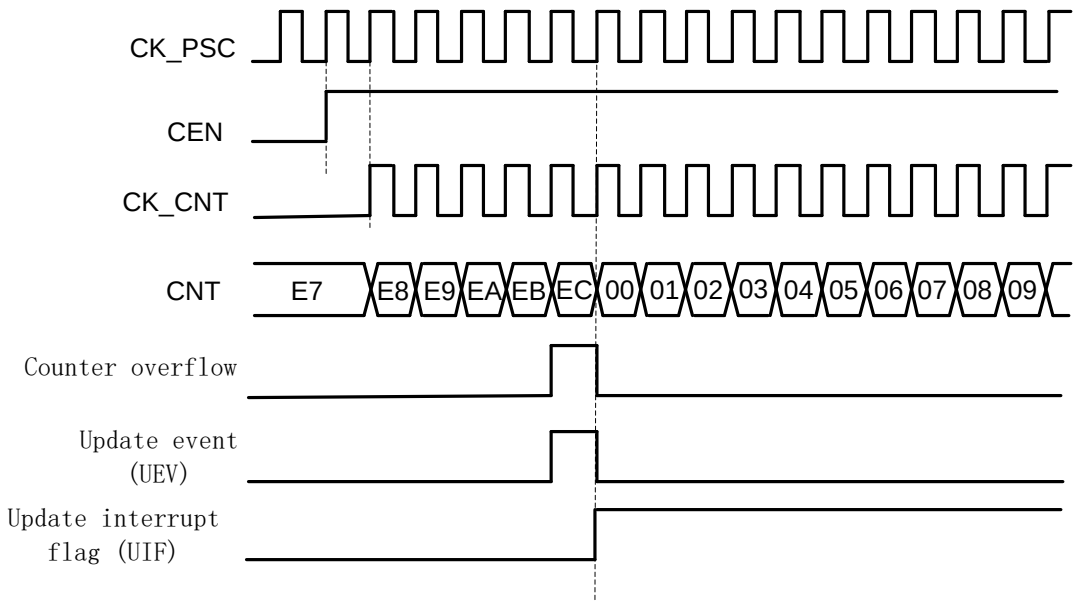


图 26-4 向上计数波形，内部时钟不分频

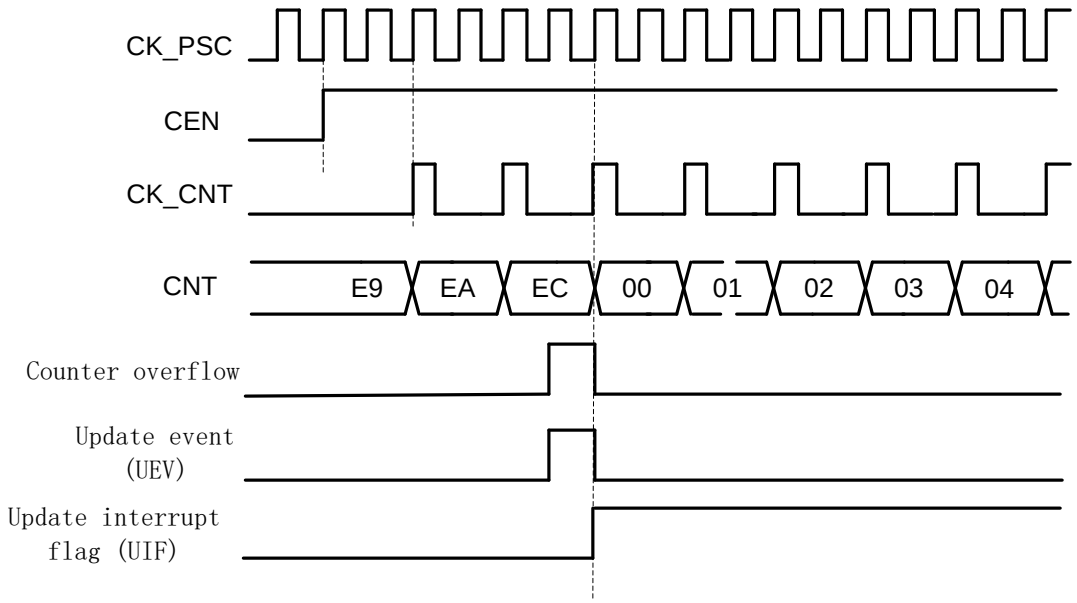


图 26-5 向上计数波形，内部时钟 2 分频

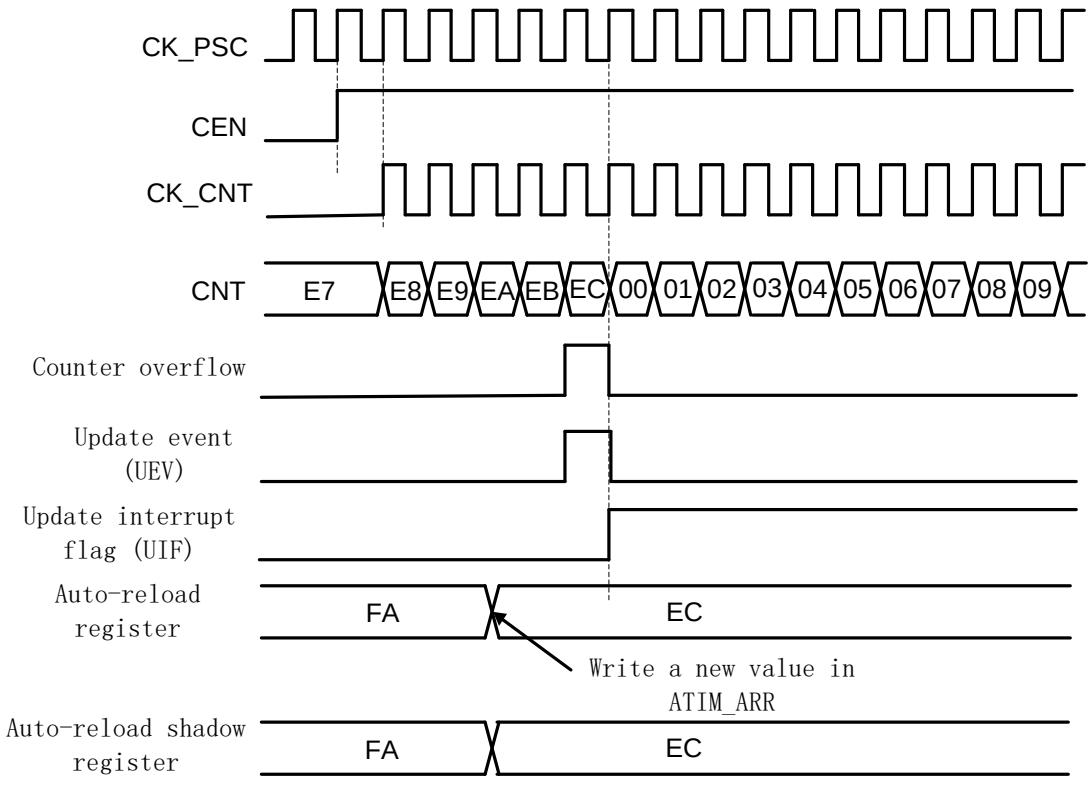


图 26-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

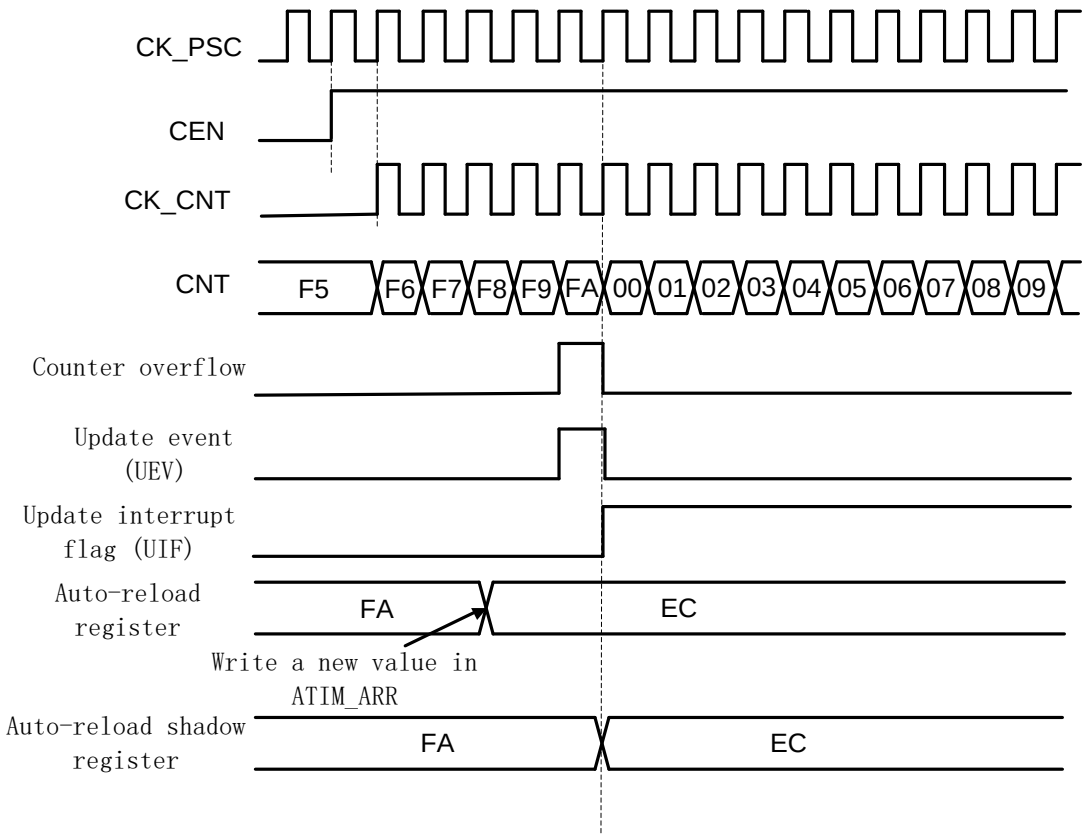


图 26-7ARPE=1（ATIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从ARR值开始递减，到0后产生下溢出事件，并且重新从ARR开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

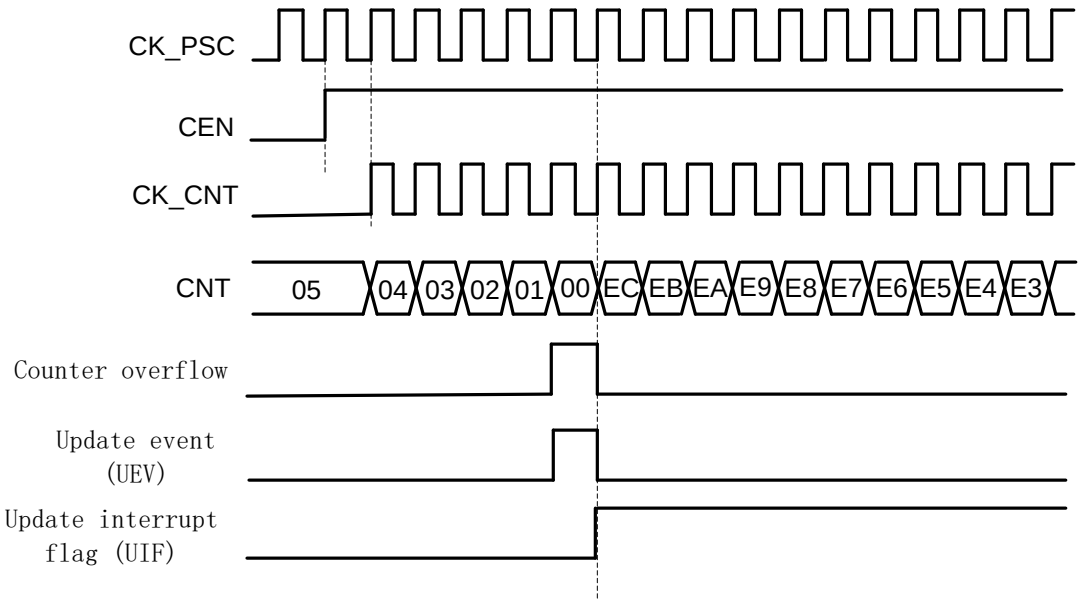


图 26-8 向下计数，内部时钟不分频

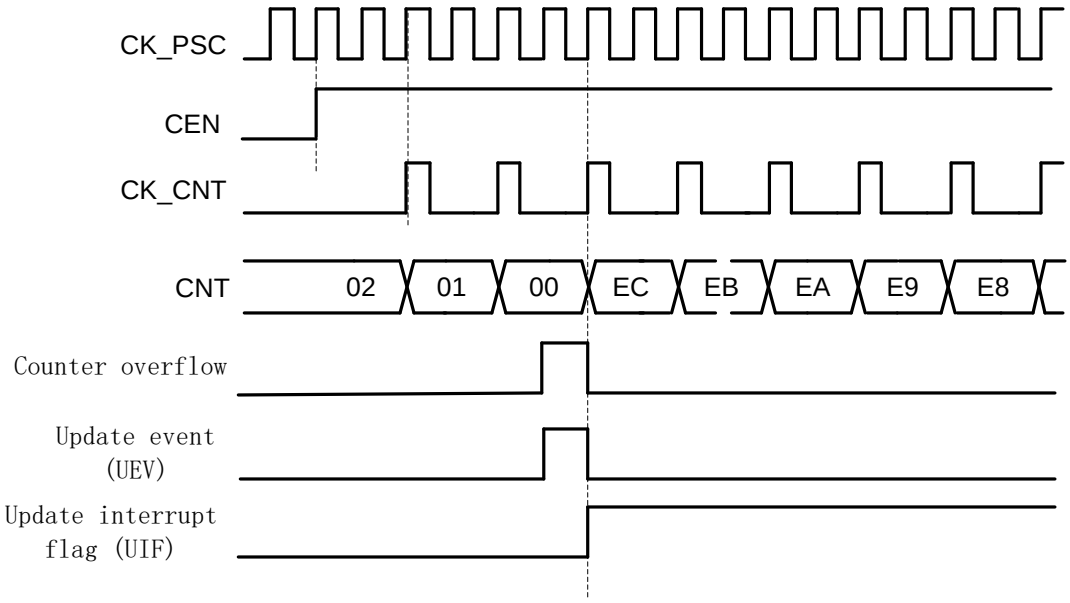


图 26-9 向下计数，内部时钟 2 分频

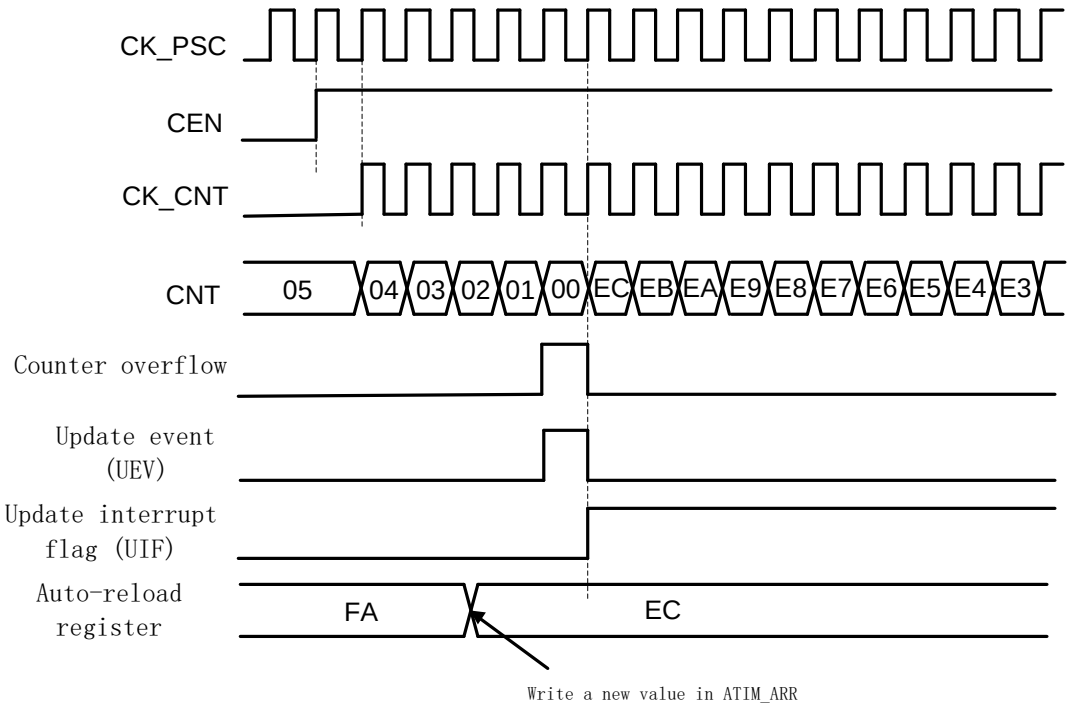


图 26-10 向下计数，内部时钟 2 分频

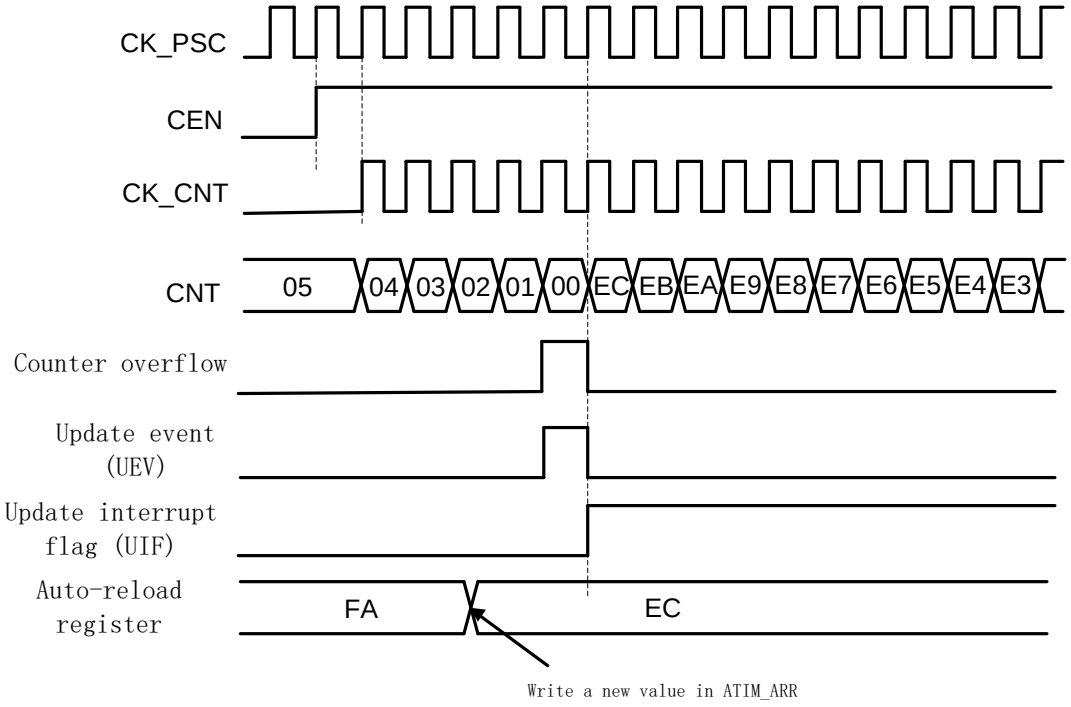


图 26-11 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从0开始向上计数，到ARR-1产生上溢出事件，然后从ARR开始向下计数到1，产生下溢出事件，再从0重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当CMS!=00时为中心对齐计数，当CMS=01时，输出比较功能仅在向下计数时有效，当CMS=10时，输出比较功能仅在向上计数时有效，当CMS=11时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在overflow和underflow的事件上都会更新 ARR、PSC和RCR的影子寄存器。

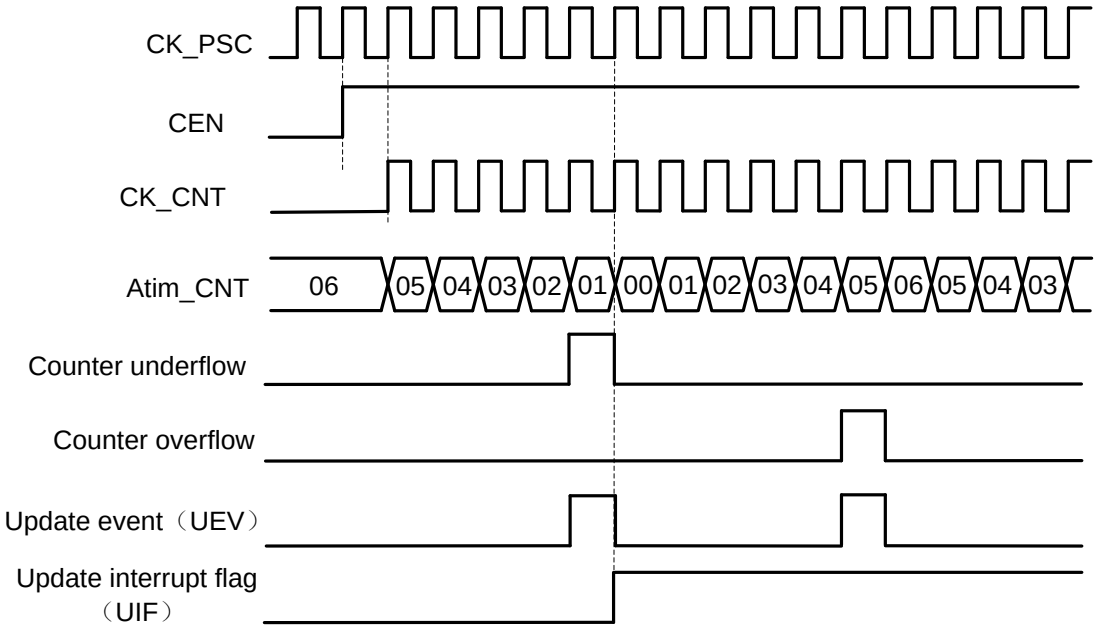


图 26-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

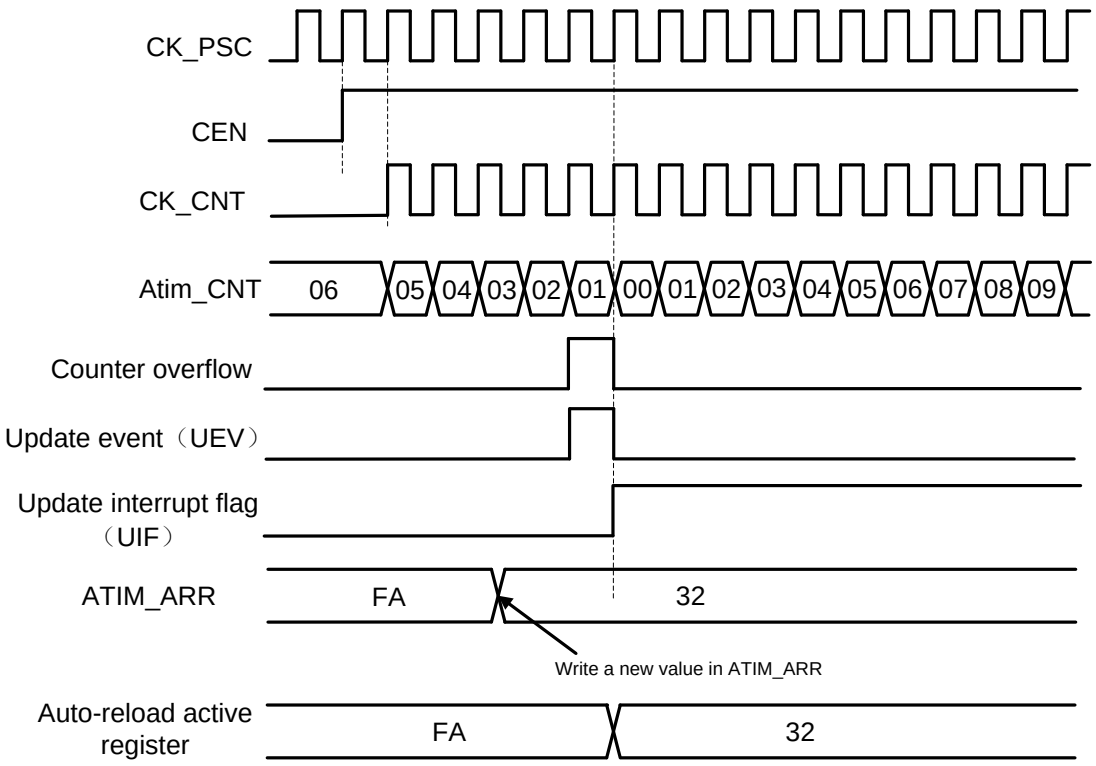


图 26-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

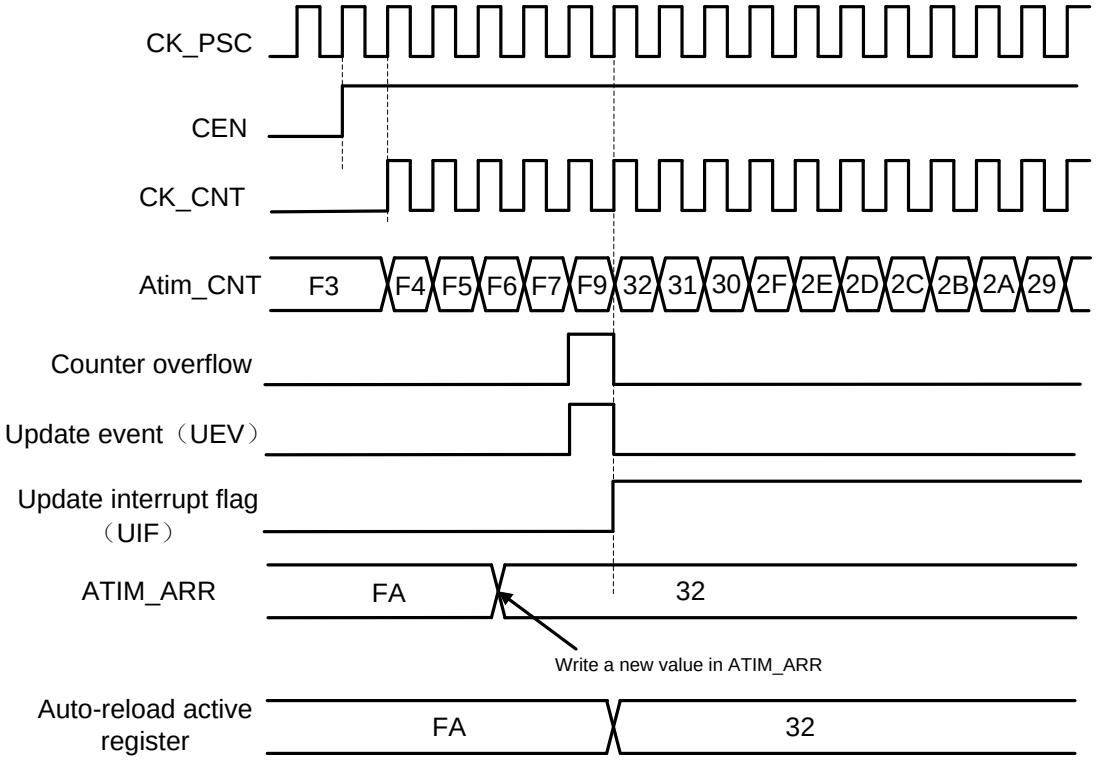


图 26-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

26.4.3 重复计数器

Update event在计数器overflow或underflow，并且重复计数器为0 的情况下产生。这意味着ARR、PSC、CCR（比较/捕捉寄存器，输出比较模式下）的preload寄存器会在N+1次overflow或underflow之后，才将数据传输给影子寄存器，其中N是RCR寄存器值。

重复计数器在以下情况下递减：

- 向上计数模式下发生上溢出
- 向下计数模式下发生下溢出
- 中心计数模式下每次上溢出或者下溢出

注意，当update event由软件或slave mode controller触发时，更新事件会立即发生，而不管当前RCR是什么值，同时重复计数器也会被立即更新为RCR的值。

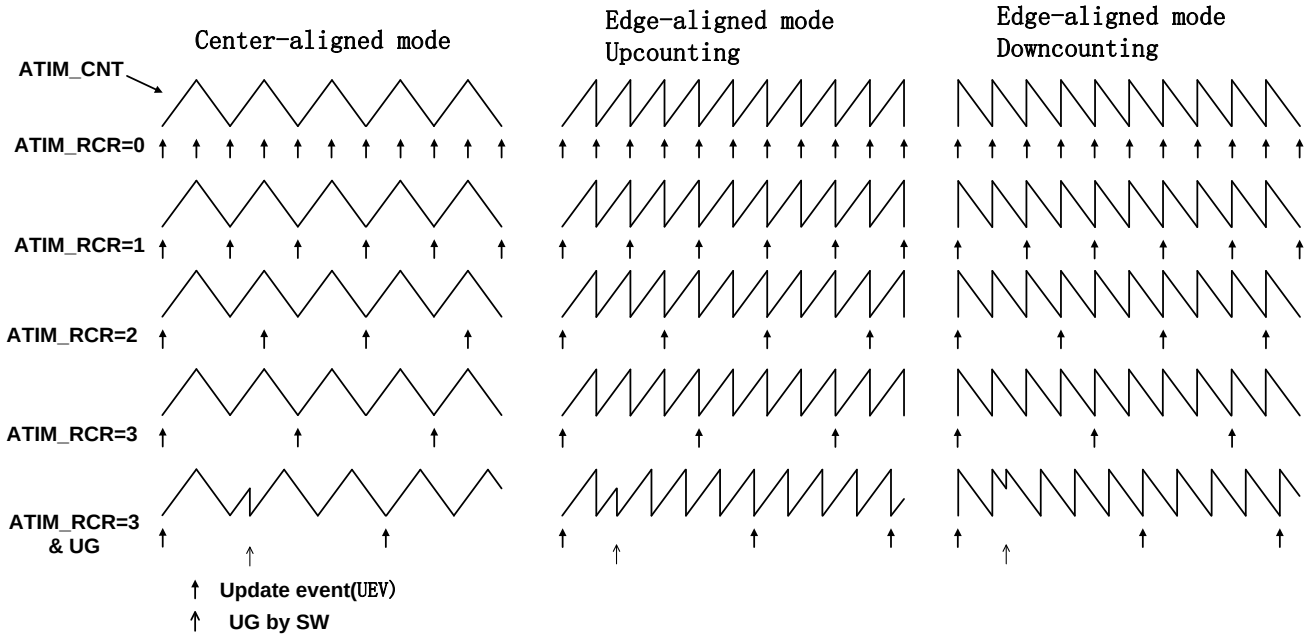


图 26-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

26.4.4 Preload 寄存器

以下功能寄存器支持preload功能：

- 自动重载寄存器ARR
- 重复计数寄存器RCR
- 预分频寄存器PSC（不可关闭preload功能）
- 通道控制寄存器CCR
- CcxE和CcxNE控制寄存器
- OcxM控制寄存器

以上寄存器，除了PSC之外，都可以由软件选择使能或者禁止preload功能。

具备preload功能的寄存器，包含两组物理实体：

- Shadow register（影子寄存器）：实际定时器正在使用的寄存器
- Preload register（预装载寄存器）：软件可以访问的寄存器

当禁止preload时，具备preload功能的寄存器特性如下：

- Preload寄存器可以实时由软件访问、改写
- Shadow寄存器与Preload寄存器同步更新

如果使能了preload，则：

- 所有软件操作访问的是preload寄存器
- 当update event发生时，所有preload寄存器内容将同步被转移到对应的shadow寄存器

26.4.5 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（Tix）——外部时钟模式1
- 外部引脚触发输入（ETR）——外部时钟模式2
- 内部触发（ITRx）——使用一个timer的触发输出（TGO）作为计数时钟

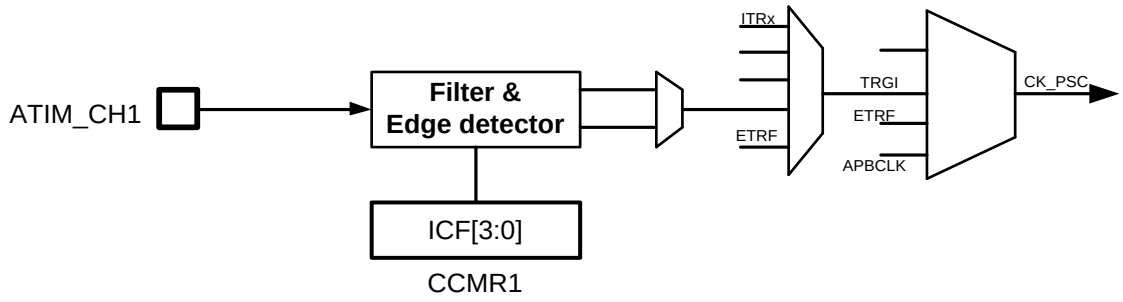


图 26-16 ATIM 时钟源框图

26.4.5.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

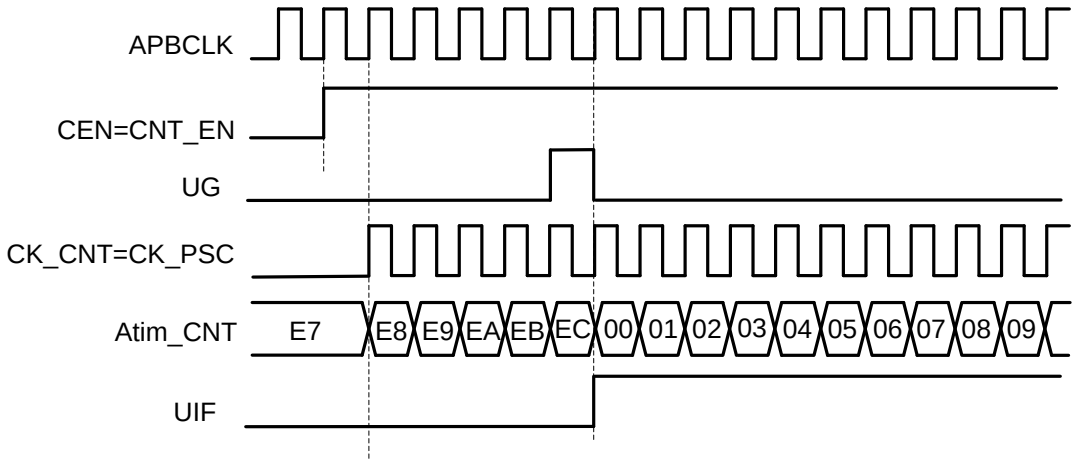


图 26-17 内部时钟源模式，时钟分频因子为 1

26.4.5.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置SMS=111，计数边沿可以配置为上升或下降沿。

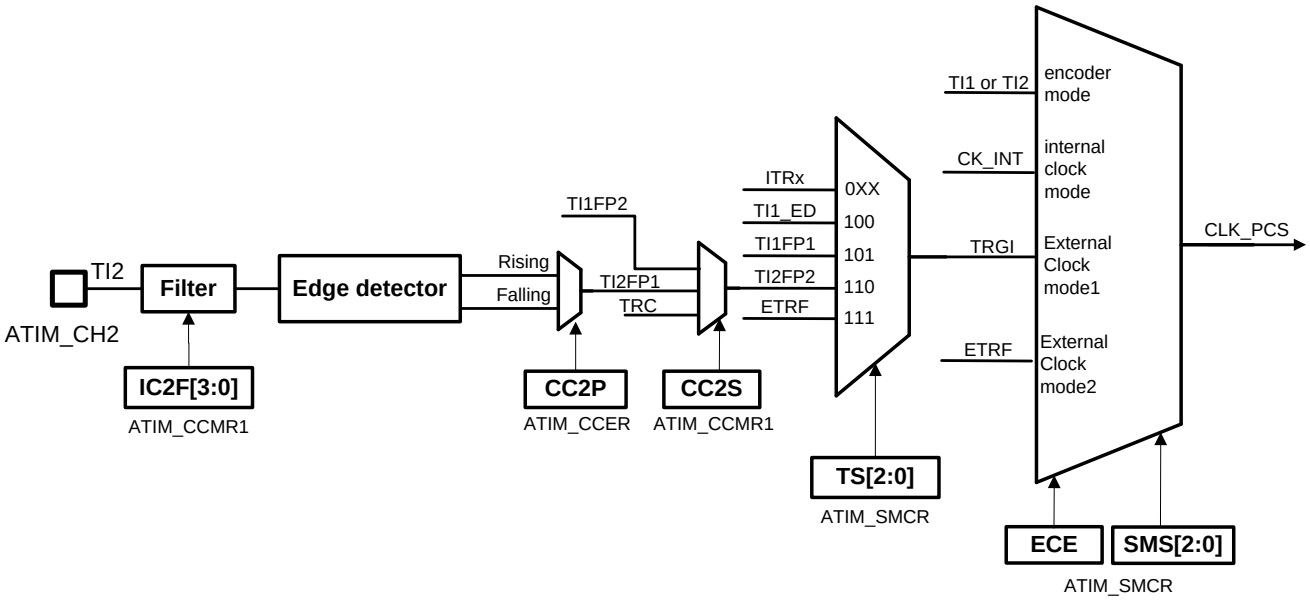


图 26-18TI2 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发TIF标志

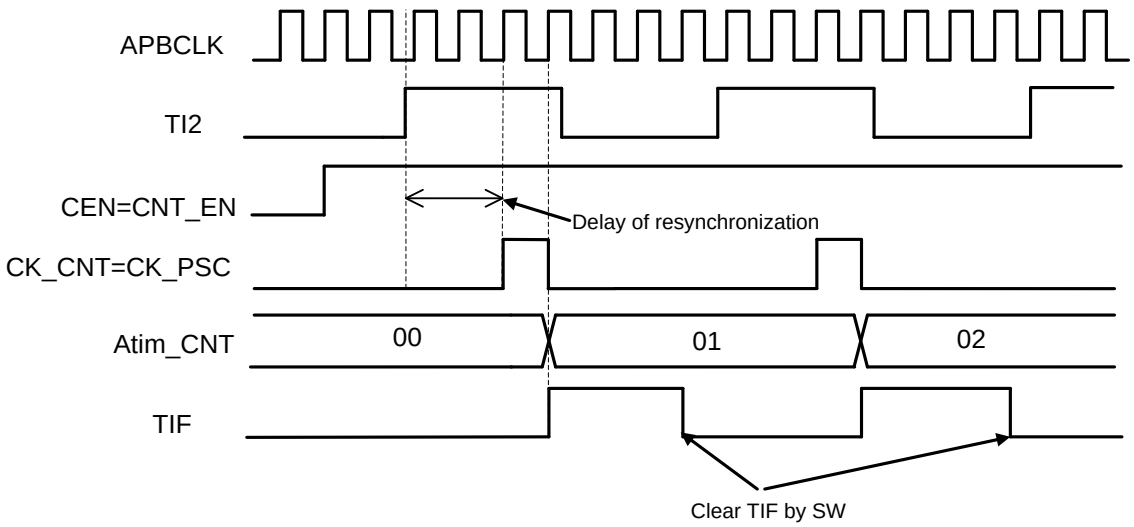


图 26-19 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能ATIM的内部时钟（APBCLK），因为ATIM要使用APB_CLK来对

外部输入时钟进行同步和滤波。在外部时钟模式1下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的2级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于2个APB_CLK周期。

此模式下只有通道1和2的输入可以用做时钟输入，所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置ATIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置ATIM_SMCR.SMCR=111
- 选择触发输入源，配置ATIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置ATIM_CCER.CC2E=1
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

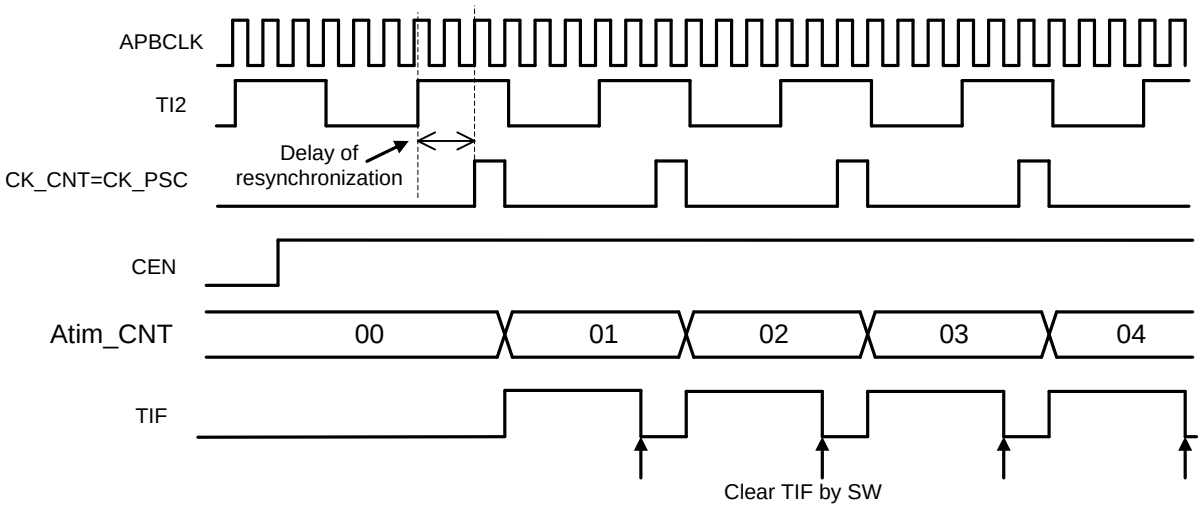


图 26-20 外部时钟模式 1 下的时序

26.4.5.3 外部时钟模式 2

此模式下使用ATIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

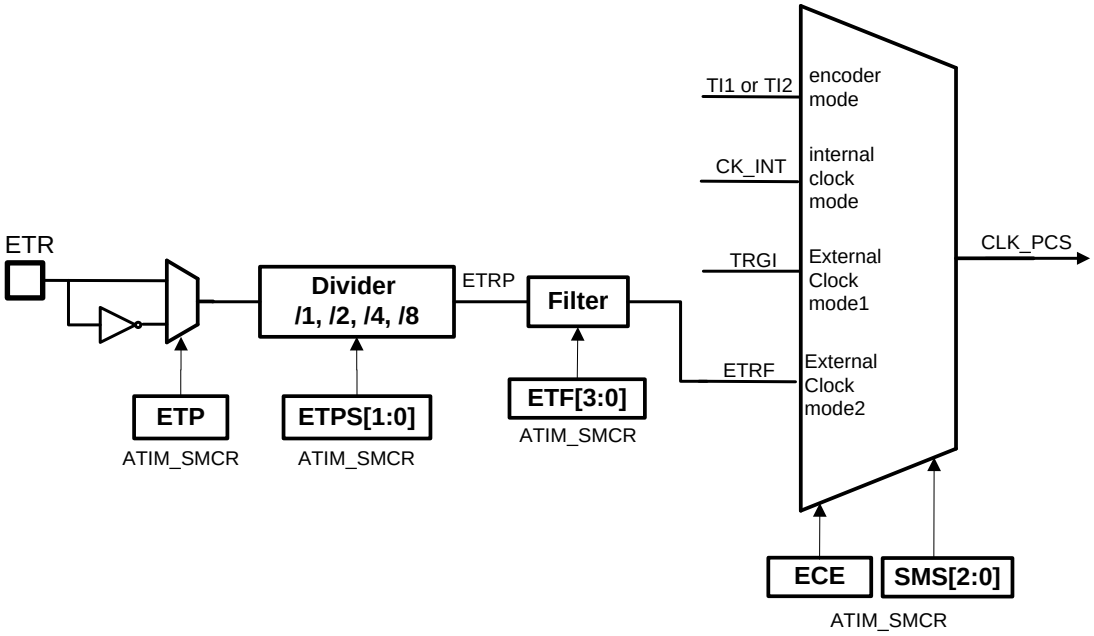


图 26-21 外部触发输入框图

下图是使用ETR二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于ETR输入上升沿。

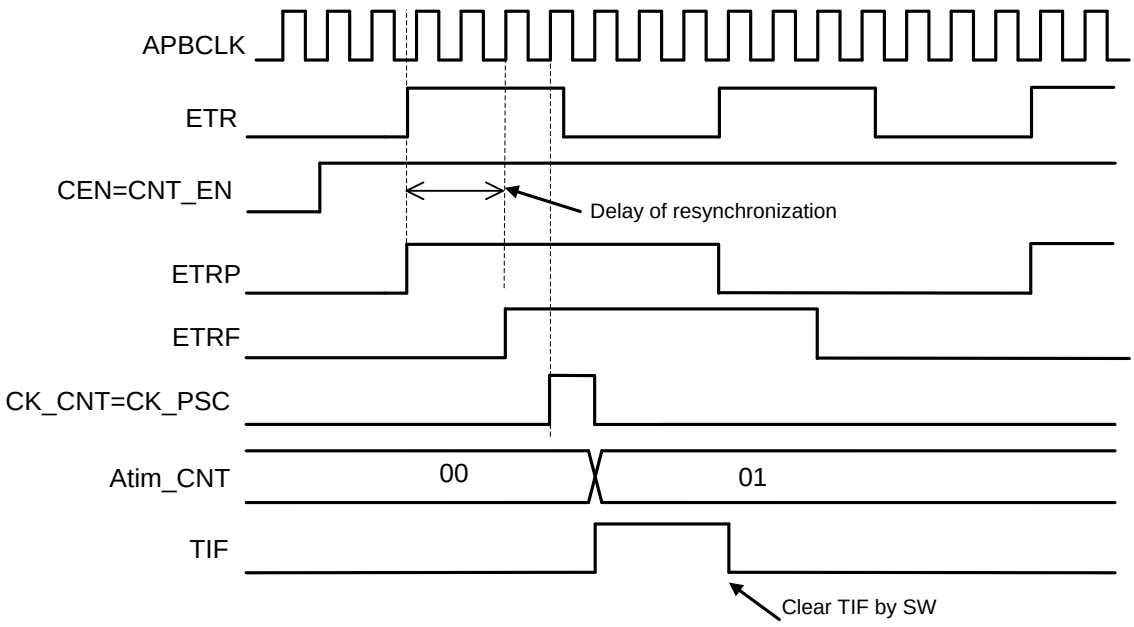


图 26-22 外部时钟模式 2 下的时序 1

与外部时钟模式1的主要差别是，ETR输入直接被分频后再进行滤波，产生CK_PSC时钟，这意味着可以支持ETR输入频率高于APB_CLK的应用场景，这种情况下，需要首先对ETR输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1，ATIM_SMCR.SMS=000
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

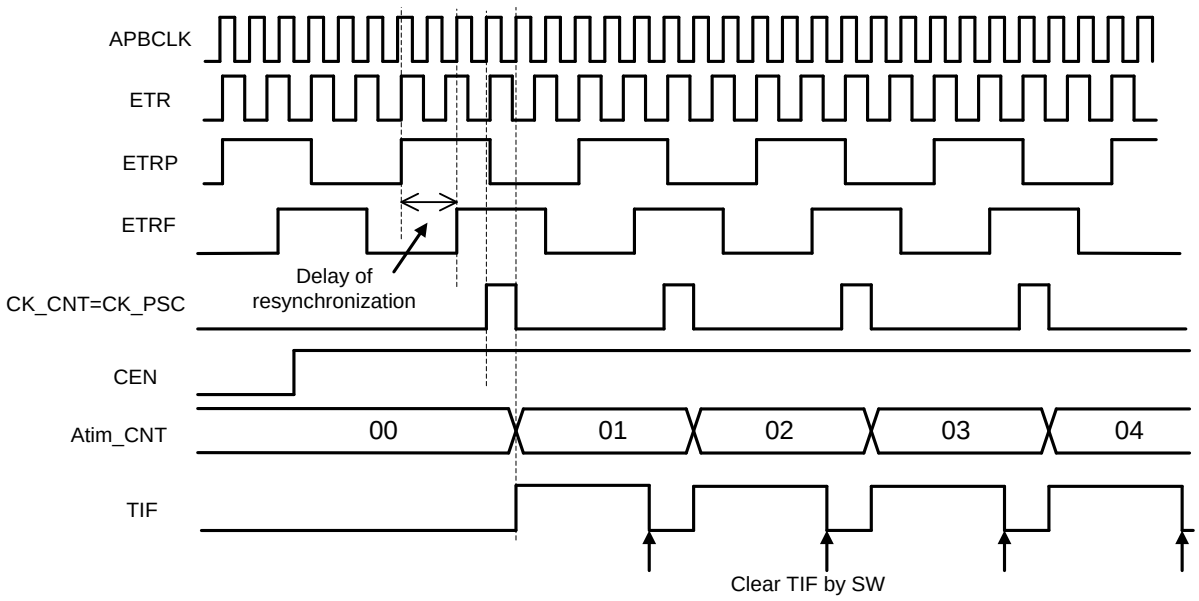


图26-23外部时钟模式2下的时序2

在使用外部时钟模式2时，仍可以将ATIM配置为slave模式：比如使用ETR输入计数，同时使用另一个Timer的TRGO作为触发信号，当触发事件到来时，复位计数器重新开始计数。

26.4.6 内部触发信号（ITRx）

ATIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIM	GPTIM0_TRGO	GPTIM1_TRGO	COMP1	COMP2

26.4.7 捕捉/比较通道

ATIM包含4个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样Tix输入并产生滤波后的信号TixF，然后边沿检测和极性选择产生对应的TixFPx信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

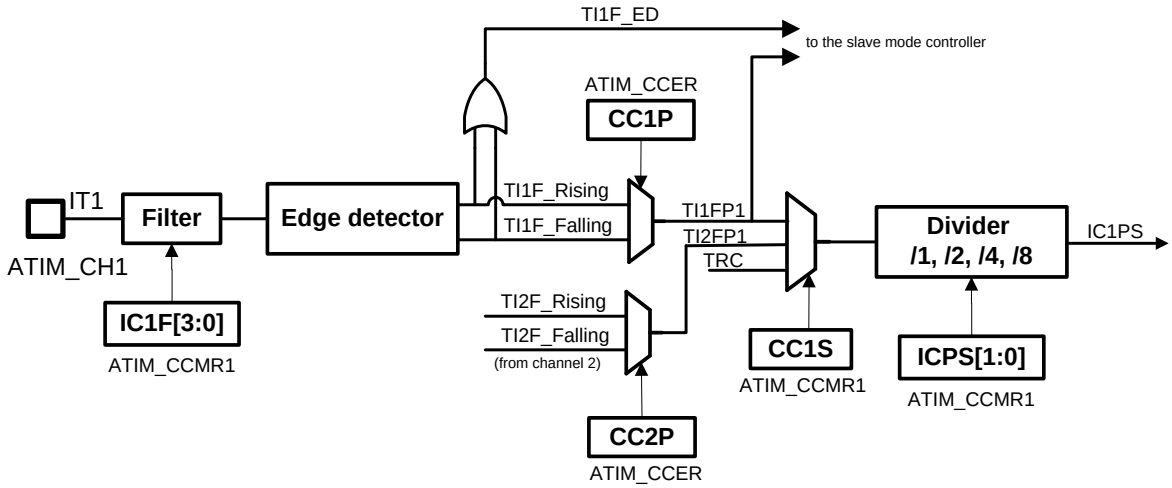


图26-24捕获/比较通道(通道1输入部分)

输出级电路会产生一个输出基准信号OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道1~3支持互补输出和死区插入，通道4则比较简单，不支持互补输出。

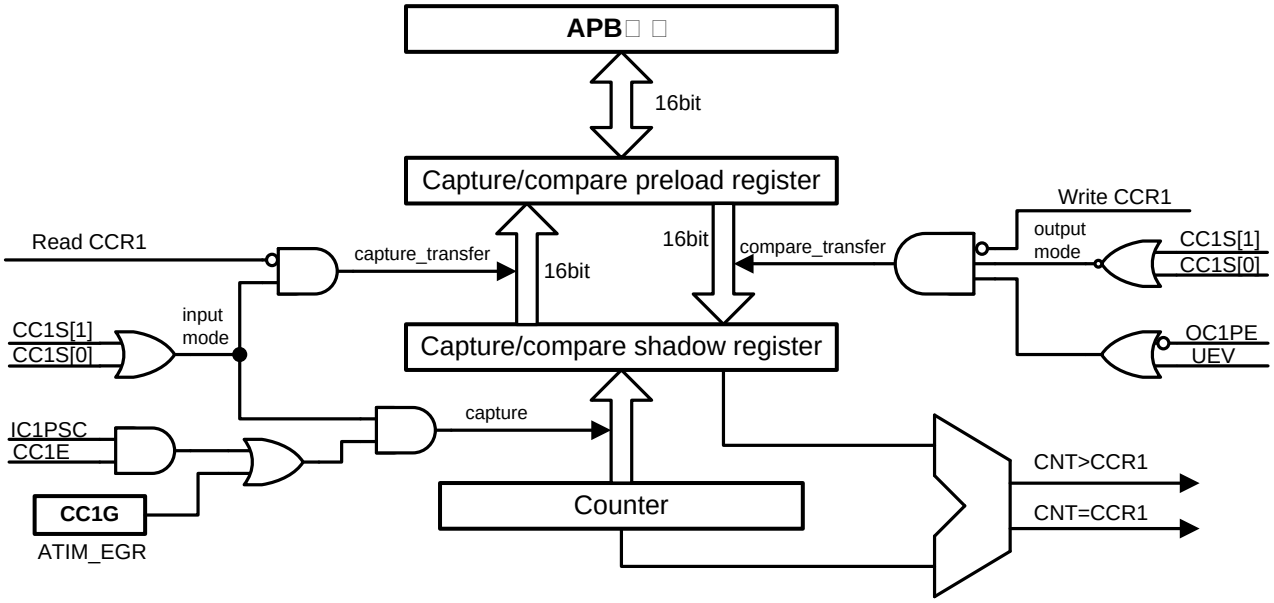


图26-25捕获/比较通道1的主电路

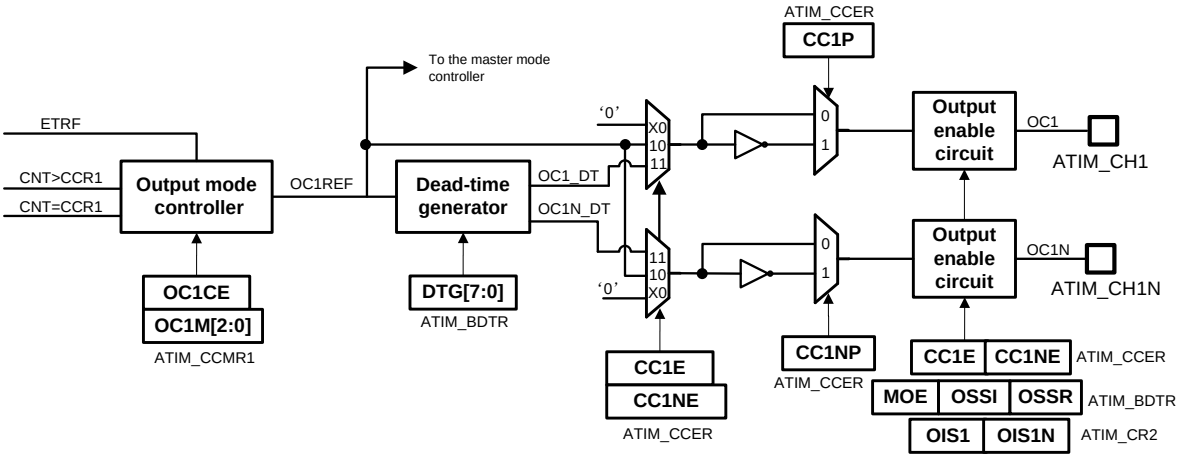


图26-26捕获/比较通道的输出部分(通道1至3)

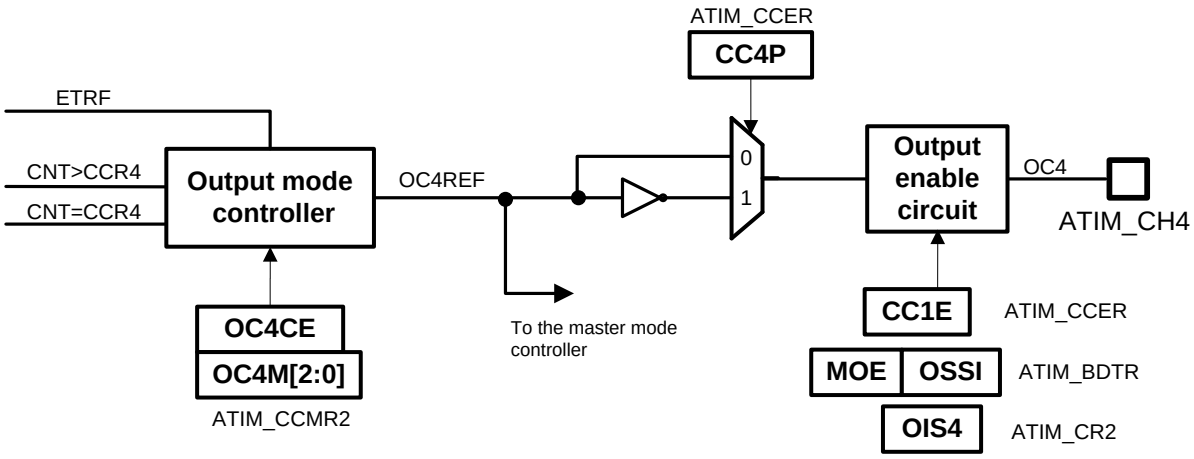


图26-27捕获/比较通道的输出部分(通道4)

捕捉/比较寄存器（CCR）包含preload寄存器和shadow寄存器，软件读写总是访问preload寄存器。在捕捉模式下，捕捉值保存在shadow寄存器中并复制到preload寄存器。在比较模式下，preload寄存器的值被拷贝到shadow寄存器用来与计数器比较。

26.4.8 输入捕捉模式

当Icx信号上出现预期的电平变换，将触发一次capture，当前计数器值被锁存进CCR，与此同时，CcxIF中断标志置位，并且可以触发对应的中断或者DMA请求。如果一个捕捉事件在CcxIF为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR中上次捕捉值被覆盖）。CcxIF可以由软件清零，或者通过读取CCR寄存器自动清零。CcxOF标志通过软件写1清零。

通过两个或更多通道配合，可以实现PWM信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从TI1引脚输入，芯片内部将滤波后的信号取上升沿得到TI1FP1，将滤波后的信号取下降沿得到TI1FP2，将TI1FP1输入给捕捉通道1，将TI1FP2输入给捕捉通道2，即可实现通道1对输入信号上升沿捕捉，同时通道2对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过CCR1和CCR2寄存器的值，即可计算输入信号的周期和占空比。

实现在TI1输入的上升沿捕获计数器的值到ATIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置ATIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置ATIM_CCER.CC1E=1

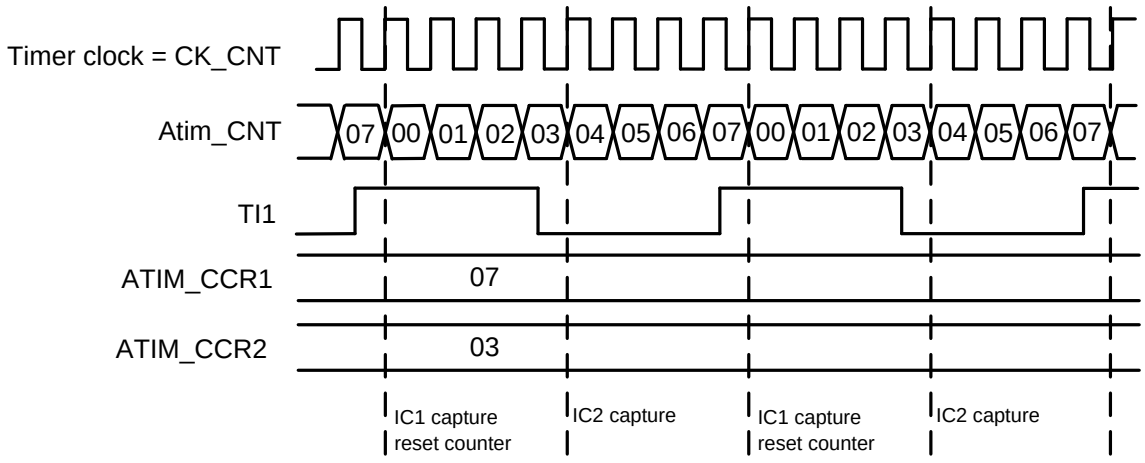


图26-28PWM输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0确保之后通道配置成功
- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置ATIM_CCMR1.CC1S=01, ATIM_CCMR1.CC2S=10
- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=1
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]，ATIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0] ，ATIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

26.4.9 软件 Force 输出

在比较输出模式下，软件可以直接将OCxREF force成特定电平，而独立于CCR和计数器的比较结果。

软件通过写OcxM=101寄存器，可以直接将OCxREF强制为有效（OCxREF固定为高有效），通过写OcxM=100可以直接将OCxREF强制为无效（低电平）。但是软件force操作不会取消比较过程，CCR和计数器的比较还会一直进行。

26.4.10 输出比较模式

输出比较模式下，当CCR与计数器值相等，OCxREF可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA请求可以发送（改写配置寄存器？）。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

- 选择计数时钟（内部、外部、预分频等）
- 向ARR和CCR寄存器写入期望数据
- 根据需要设置中断使能和DMA使能
- 选择输出模式
- 使能计数器

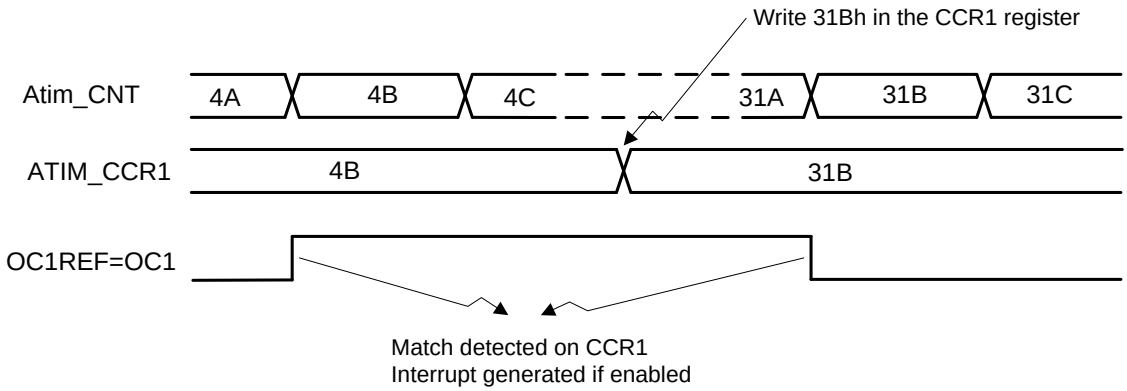


图26-29输出比较模式，翻转OC1

在不使能preload的情况下，软件可以随时改写CCR寄存器实现对输出波形的实时控制。如果使能了preload，则CCR shadow寄存器仅在下一次update event发生时更新为preload寄存器的内容。

26.4.11 PWM 输出

PWM模式可以输出脉宽调制信号，其周期由ARR寄存器决定，占空比由CCR寄存器决定。

输出信号的极性可以由CCxP寄存器配置。PWM模式工作中，CNT和CCR实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM输出也支持边缘对齐和中央对齐模式。

PWM边缘对齐模式

在向上计数的情况下，配置为PWM模式1时，OCxREF信号在CNT<CCR时为高电平，否则为低电平。如果CCR值大于ARR值，则OCxREF被固定为1；如果CCR为0则OCxREF被固定为0.

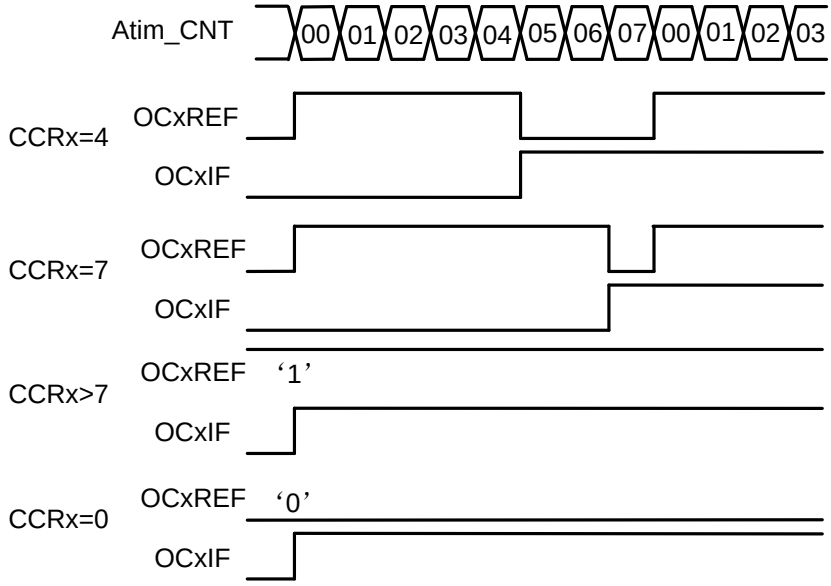


图26-30边沿对齐的PWM波形(ARR=7)

在向下计数时，OCxREF电平高低定义与向上计数时相同。

PWM中央对齐模式

OCxREF电平定义与边缘对齐模式相同。下图是一个示例：

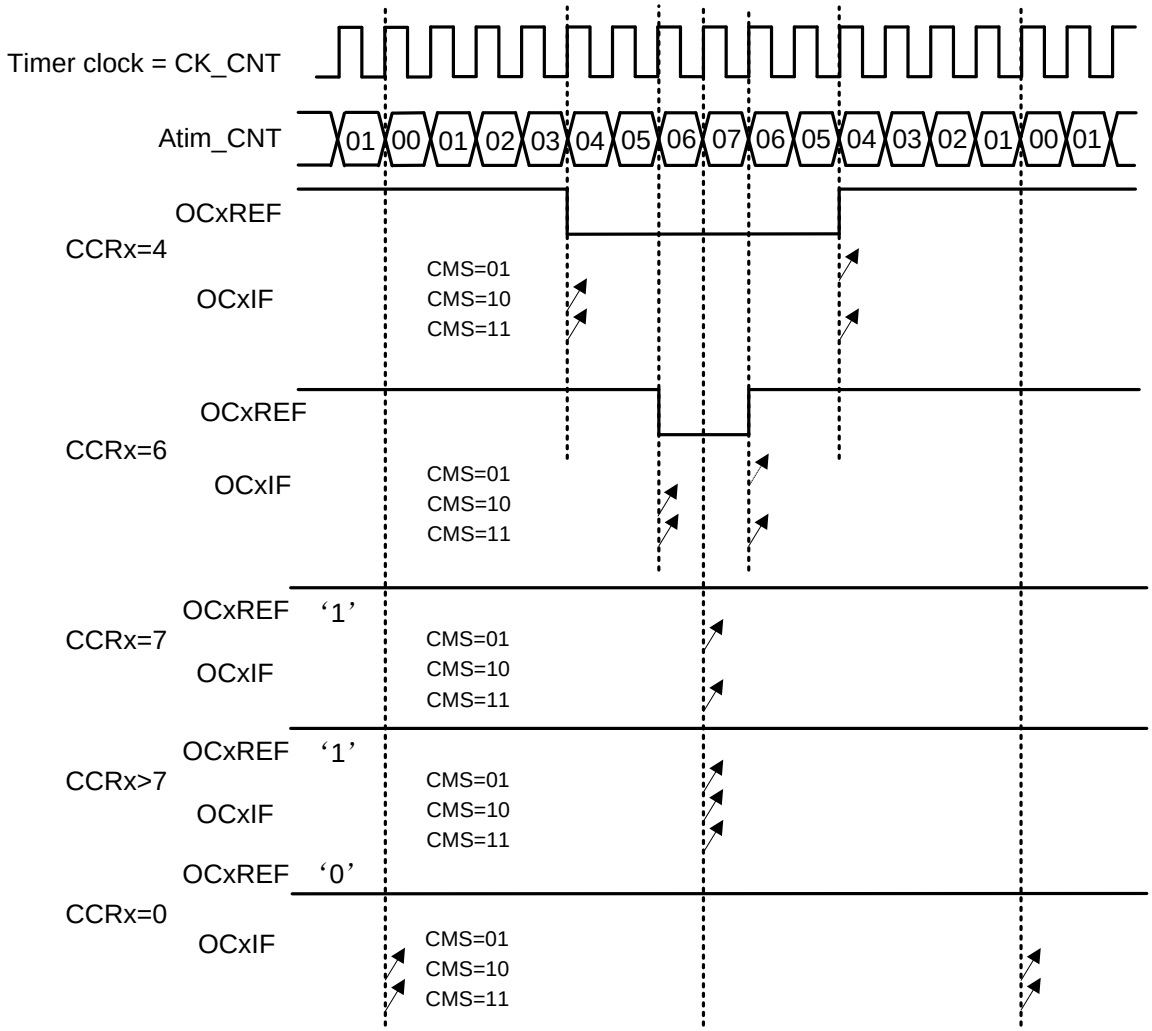


图26-31中央对齐的PWM波形(APR=7)

当启动中央对齐计数时，一开始的计数方向是由DIR寄存器决定的；随后在计数过程中，DIR寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过UG寄存器做一次update，并且在计数过程中不要改写计数器。

26.4.12 互补输出和死区插入

ATIM的通道1~3支持互补输出和死区插入。DTG[7:0]寄存器用于设置死区时间（对所有通道同时有效）。输出信号Ocx与参考信号OCxREF同相，OcxN与参考信号反相；Ocx的上升沿是OCxREF上升沿的delay，OcxN的上升沿是OCxREF下降沿的delay。

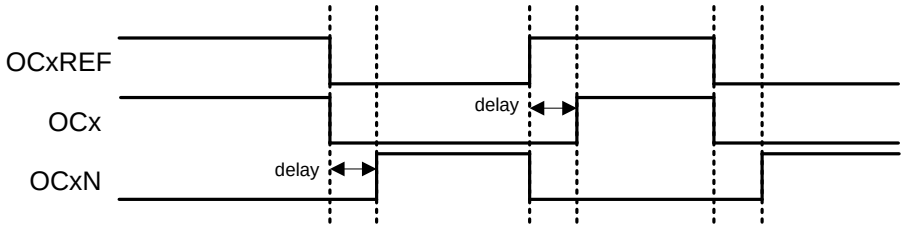


图26-32带死区插入的互补输出

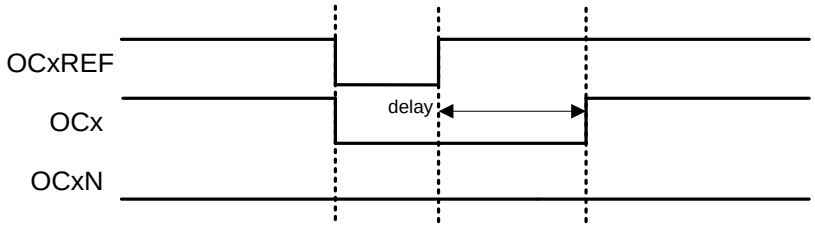


图26-33死区波形延迟大于负脉冲

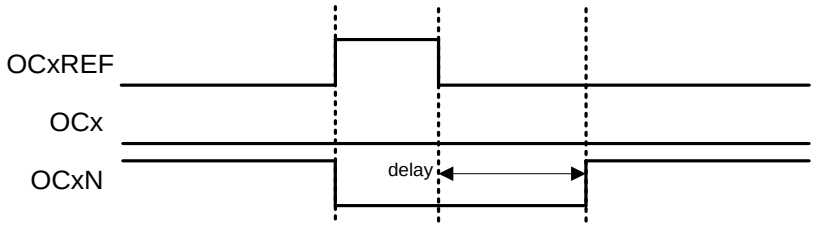


图26-34死区波形延迟大于正脉冲

26.4.13 刹车功能

刹车功能可以使用外部BRK引脚输入的2路刹车信号，或者比较器、SVD、XTHF停振检测产生的有效输出；上电复位后刹车电路被禁止，用户通过置位BKE寄存器使能刹车功能；2路刹车输入可以配置为相与或者相或操作。组合后的刹车信号可以配置有效极性，以及数字滤波。

刹车输入控制逻辑如下图所示：

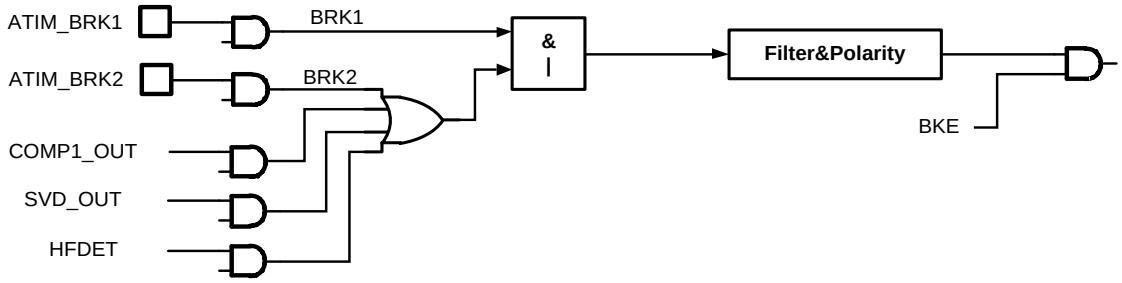


图26-35刹车控制逻辑

ATIM_BRKx复用GPIO功能，当GPIO设置为数字外设功能时，其输入信号直接连接到ATIM的刹车输入上；当GPIO设置为其他功能时，ATIM的刹车输入端口被固定成1。通过BRKxGATE寄存器，可以控制门控后的BRKx信号的实际电平，软件能够灵活的将不使用的BRKx设置为0或者1电平，以适应后级逻辑电路的需要。

当一个刹车事件发生时：

- 输出使能寄存器被异步清零，可以通过OSSI寄存器选择输出被强制为inactive/idle/reset状态
- 每个输出通道被驱动为OISx寄存器定义的电平
- 当互补输出使能时，输出被异步置位成inactive和reset状态，死区插入电路开始工作，在死区时间后驱动输出为OISx和OISxN定义的电平
- 刹车标志寄存器置位，根据配置可以触发中断或DMA
- 如果使能了自动输出（AOE=1），输出使能位（MOE）将在下一个update event发生时被自动置位；否则MOE将保持为0直到被软件重新置位。

注意BRK信号是电平有效的，因此在BRK保持有效的情况下，无法使能MOE，同时刹车标志BIF也无法清除。

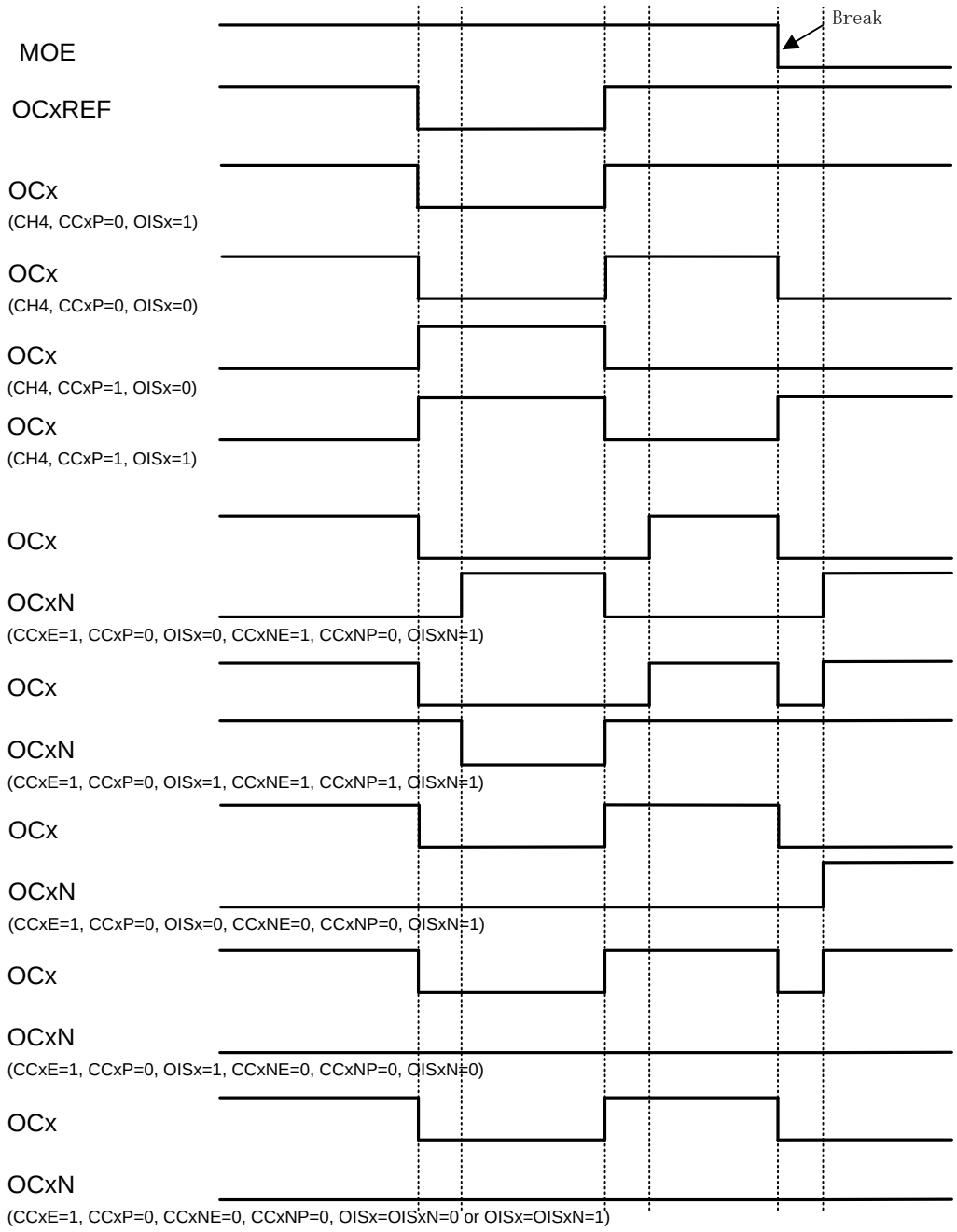


图26-36响应刹车的输出

26.4.14 互补输出通道信号状态逻辑表

以下是控制寄存器和互补输出通道的状态对应表，其中 MOE 为定时器总输出使能位，OSSI 定义 IDLE 状态（MOE=0）下是关闭 IO 输出还是进入 off state，OSSR 定义 RUN 状态（MOE=1）下的是关闭 IO 输出还是进入 off state。

控制寄存器					输出状态	
MOE	**OSSI**	**OSSR**	**CcxE**	**CcxNE**	**Ocx 输出状态**	**OcxN 输出状态**
1	X	0	0	0	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	输出关闭（不由ATIM驱动）,OcxN=0,OcxN_EN=0
		0	0	1	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		0	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Output Disabled (not driven by the timer) OcxN=0, OcxN_EN=0
		0	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) Ocx=CCxP, Ocx_EN=0	Output Disabled (not driven by the timer) OcxN=CCxNP, OcxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) Ocx=CCxP, Ocx_EN=1	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		1	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Off-State (output enabled with inactive state) OcxN=CCxNP, OcxN_EN=1
		1	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
0	0	X	0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	0		0	1	输出关闭（不由ATIM驱动） 如果无时钟：Ocx=CCxP, Ocx_EN=0, OcxN=CCxNP, OcxN_EN=0 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	0		1	0		
	0		1	1		
	1		0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	1		0	1	Off-state（输出使能，inactive输出） 如果无时钟：Ocx=CCxP, Ocx_EN=1, OcxN=CCxNP, OcxN_EN=1 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	1		1	0		
	1		1	1		

表 26-1 通道状态表

26.4.15 6-step PWM 输出

当某个通道使用互补输出时，OcxM, CcxE, CcxNE寄存器支持preload功能，preload寄存器的值在换相（COM）事件发生时被装载到shadow寄存器中。用户因此可以预先设置下一步配置，并在COM事件发生时同步更新所有通道。COM事件可以由软件写ATIM_EGR中的COM位触发，或者由TRGI上升沿硬件触发。

当COM事件发生时，换相标志寄存器置位，并且可以产生中断或DMA请求。

下图是一个6步换相控制的例子，当COM事件发生时，三个例子显示不同配置下的输出变化。

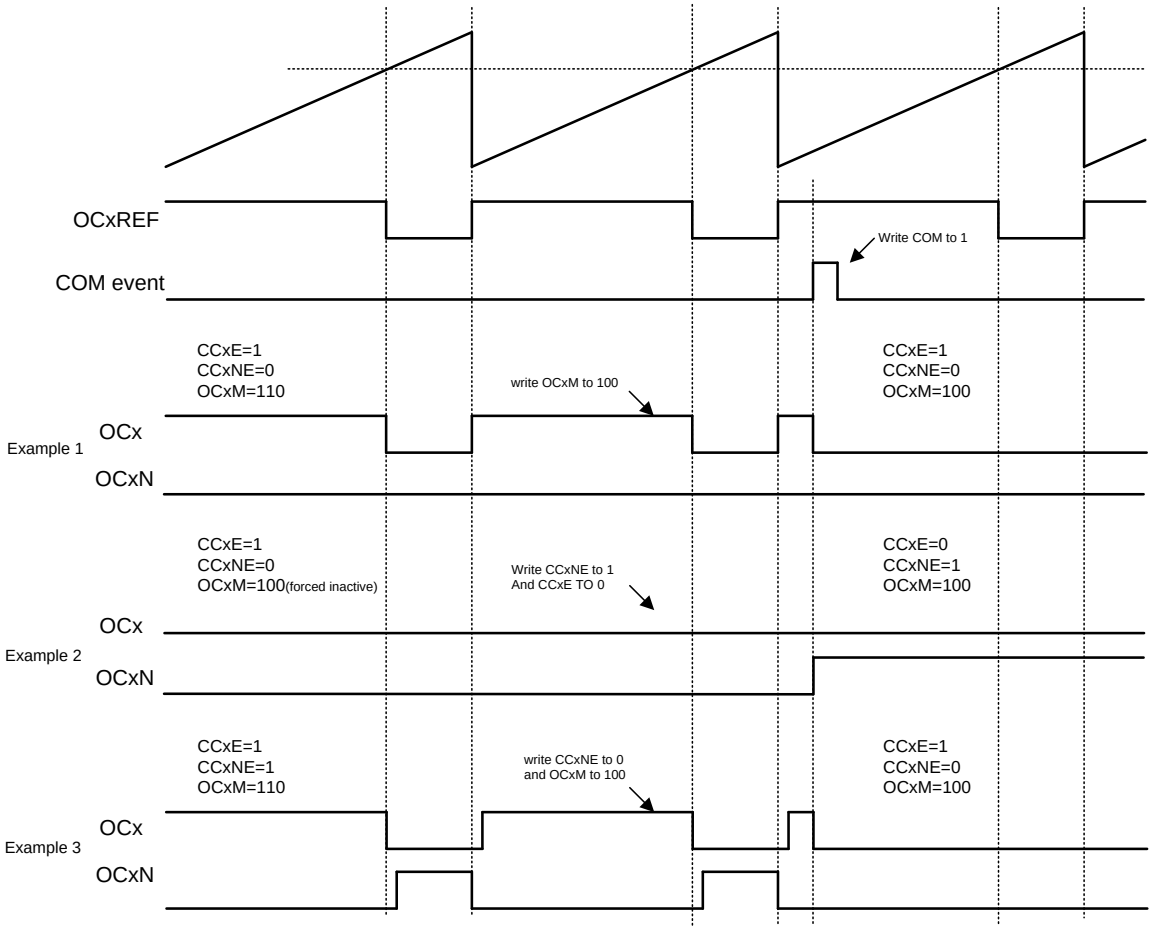


图26-37产生六步PWM，使用COM的例子(OSSR=1)

26.4.16 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次update event到来时，计数器会自动停止。只有当CCR和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求CNT<CCR<=ARR，在向下计数时，要求CNT>CCR

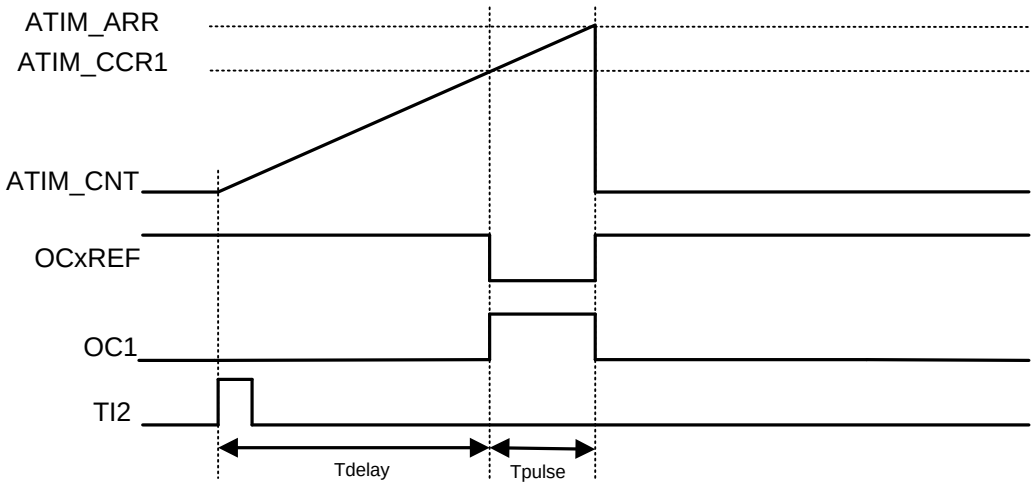


图26-38单脉冲模式的例子

上图是以TI2输入为计数器触发信号，计数值等于CCR后OCxREF输出低电平，计数到ARR后OCxREF回到高电平，并且计数器回滚到0，停止计数。

实现上述功能TI2作为输入触发的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC2P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置ATIM_CCER.CC2E=1

实现上述功能OC1作为输出的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置ATIM_CCMR1.CC1S=00

- 选择计数有效沿，配置ATIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置ATIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- ATIM_CCR1的值决定了Tdelay
- ATIM_ARR和ATIM_CCR1的差值决定了Tpulse（ATIM_ARR-ATIM_CCR1）
- 设置为单脉冲模式，配置ATIM_CR1.OPM=1

26.4.17 外部事件清除 OCxREF

OCxREF的有效状态未高电平，通过对外部ETR引脚施加高电平，可以直接拉低OCxREF，直到下一次update event。此功能仅在输出比较和PWM模式下有效。使能此功能需要将OcxCE置1。

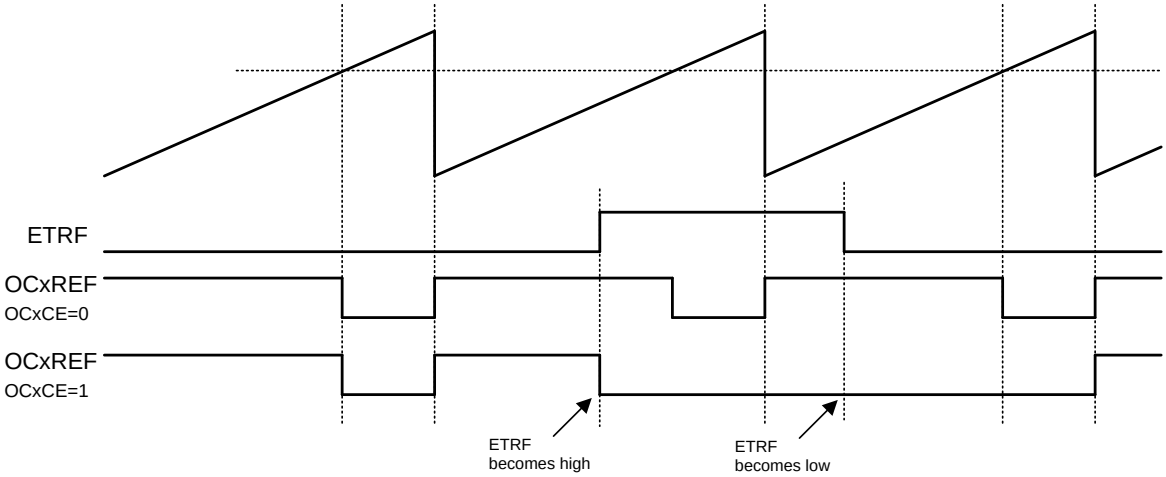


图26-39 ETR信号清除ATIM的OCxREF

26.4.18 编码器接口模式（encoder interface）

编码器接口模式涉及到两个外部输入信号，ATIM根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（TI1 对应TI2，TI2对应TI1）	TI1信号		TI2信号	
		上升	下降	上升	下降
仅在TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在TI1 和TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表26-2encoder interface计数方式

比如在计数器以TI1信号为时钟计数时，如果TI1上升沿采样到TI2为高电平，则计数器递减；如果TI1下降沿采样到TI2为高电平，则计数器递增。

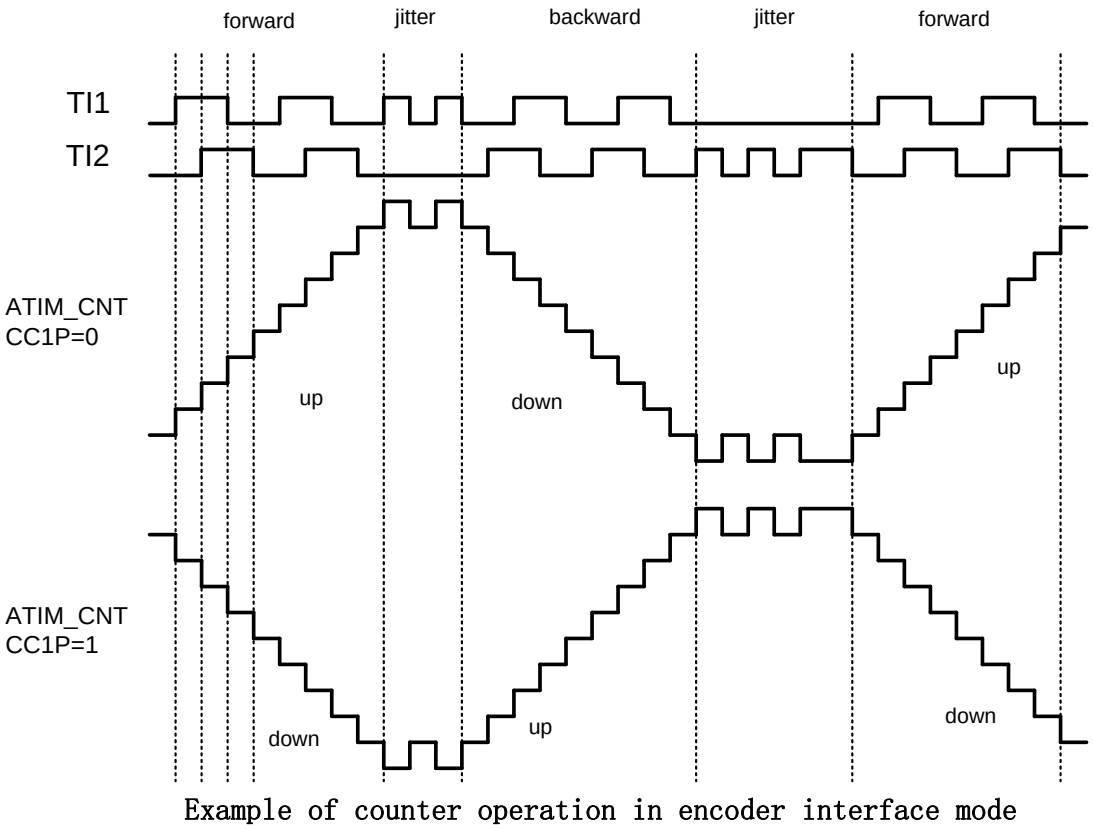


图26-40编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_CH2功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01，ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置ATIM_SMCR.SMS[2:0]=011
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

26.4.19 TIM 从机模式

ATIM作为slave时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致TIM内部所有preload寄存器重新初始化，CNT回到0开始计数。以下图为例，计数器正常计数，外部TI1输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

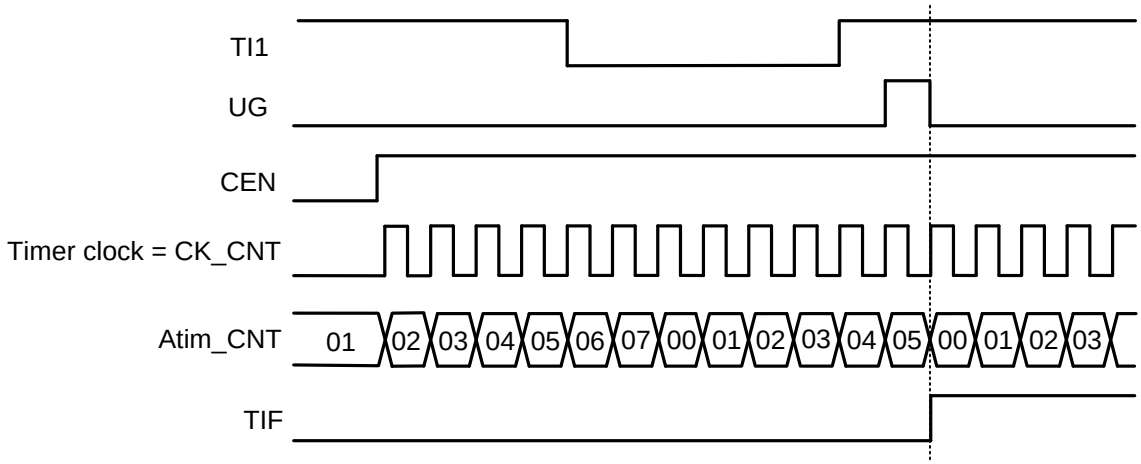


图26-41复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置ATIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

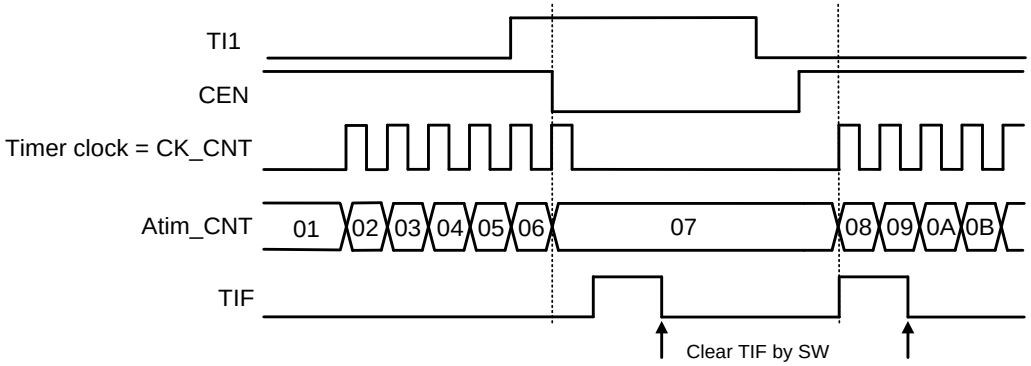


图26-42门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

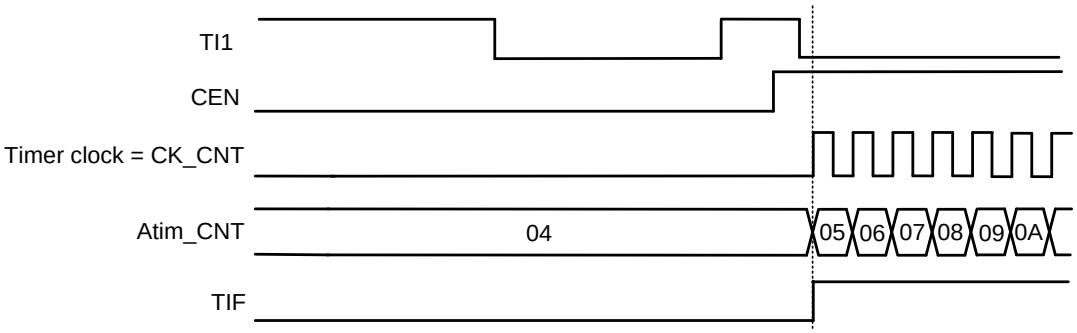


图26-43触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将ETR设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到TI1的上升沿之后，计数器开始以ETR输入的上升沿计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

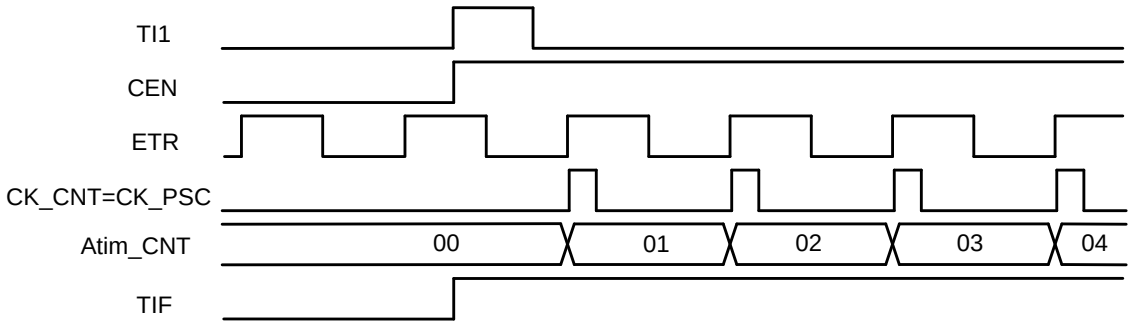


图26-44外部时钟模式2＋触发模式下的时序

26.4.20 DMA 访问

ATIM支持7种DMA请求，分别为4个CC通道请求、外部触发请求、用户软件触发请求和COM触发请求。

其中每个CC通道各自产生一个DMA请求，在捕捉模式下用于将CCRx中的内容传输给RAM，在比较模式下则用于将RAM中的数据写入CCRx；CC通道的DMA请求可以配置为单次传输或Burst传输（CCxBURSTEN），单次传输仅访问CCRx寄存器，Burst传输则根据DCR寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件、软件触发事件和COM事件也可以产生DMA请求，当这些请求发生时，会启动DMA Burst传输，向ATIM内部1个或多个寄存器写入数据，或者从ATIM读取1个或多个寄存器值。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

表 26-3DMA 请求配置

26.4.21 DMA Burst

ATIM支持DMA和DMA-Burst访问，可以配置ATIM在特定事件发生时触发DMA请求，可以将CCR中的捕捉结果写入RAM，或者从RAM中将一个或多个寄存器内容写入ATIM的preload寄存器中。

DMA-Burst支持一个事件触发连续多次DMA请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA控制器需将外设目标地址指向一个虚拟寄存器ATIM_DMAR。在特定的定时器事件发生时，ATIM会连续发射多个DMA请求。每个DMA对ATIM_DMAR的写操作都会被ATIM重新定向到实际的功能寄存器上。

DBL寄存器用于设置DMA burst长度，DBA寄存器用于设置DMA访问ATIM内部的基地址（相对于ATIM_CR的offset）。

26.4.22 输入异或功能

通道1~3的输入信号可以被异或起来之后，接入到通道1的滤波和边沿电路输入，用于通道1的输入捕捉或者触发。

ATIM_CR2寄存器的TI1S位用于选择通道1的输入是否来自于三个通道输入的异或。

26.4.23 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

Debug时当定时器被停止后，其输出会被禁止（MOE清零），根据寄存器配置，此时的输出信号可以被force成inactive或由GPIO模块控制。DMA-Burst模式下，DMA所有访问都要指向DMAR虚拟寄存器，由ATIM自动根据访问来累加内部offset地址。DBA寄存器用于指定ATIM内部首次DMA传输的目标地址，而DBL用于指定Burst长度

26.5 寄存器

offset 地址	名称	符号
ATIM(模块起始地址:0x4001B000)		
0x00000000	ATIM 控制寄存器 1 （ATIM Control Register1）	ATIM_CR1
0x00000004	ATIM 控制寄存器 2 （ATIM Control Register2）	ATIM_CR2
0x00000008	ATIM 从机模式控制寄存器 （ATIM Slave Mode Control Register）	ATIM_SMCR
0x0000000C	ATIM DMA 和中断使能寄存器 （ATIM DMA and Interrupt Enable Register）	ATIM_DIER
0x00000010	ATIM 状态寄存器 （ATIM Interrupt Status Register）	ATIM_ISR
0x00000014	ATIM 事件产生寄存器 （ATIM Event Generation Register）	ATIM_EGR
0x00000018	ATIM 捕捉/比较模式寄存器 1 （ATIM Capture/Compare Mode Register1）	ATIM_CCMR1
0x0000001C	ATIM 捕捉/比较模式寄存器 2 （ATIM Capture/Compare Mode Register2）	ATIM_CCMR2
0x00000020	ATIM 捕捉/比较使能寄存器 （ATIM Capture/Compare Enable Register）	ATIM_CCER
0x00000024	ATIM 计数器寄存器 （ATIM Counter Register）	ATIM_CNT
0x00000028	ATIM 预分频寄存器 （ATIM Prescaler Register）	ATIM_PSC
0x0000002C	ATIM 自动重载寄存器 （ATIM Auto-Reload Register）	ATIM_ARR
0x00000030	ATIM 重复计数寄存器 （ATIM Repetition Counter Register）	ATIM_RCR
0x00000034	ATIM 捕捉/比较寄存器 1 （ATIM Capture/Compare Register1）	ATIM_CCR1
0x00000038	ATIM 捕捉/比较寄存器 2 （ATIM Capture/Compare Register2）	ATIM_CCR2
0x0000003C	ATIM 捕捉/比较寄存器 3 （ATIM Capture/Compare Register3）	ATIM_CCR3
0x00000040	ATIM 捕捉/比较寄存器 4 （ATIM Capture/Compare Register4）	ATIM_CCR4
0x00000044	ATIM 刹车和死区控制寄存器 （ATIM Break and Deadtime Register）	ATIM_BDTR
0x00000048	ATIM DMA 控制寄存器 （ATIM DMA Control Register）	ATIM_DCR
0x0000004C	ATIM DMA 访问寄存器 （ATIM DMA Access Register）	ATIM_DMAR
0x00000060	ATIM 刹车输入控制寄存器 （ATIM Break Control Register）	ATIM_BKCR

26.5.1 ATIM 控制寄存器 1（ATIM_CR1）

名称	ATIM_CR1
Offset	0x00000000

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CKD	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	**CKD**	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比）(Counter cloc Divider) 00：$t_{DTS}=t_{CK\_INT}$ 01：$t_{DTS}=2*t_{CK\_INT}$ 10：$t_{DTS}=4*t_{CK\_INT}$ 11：RFU，禁止使用
7	**ARPE**	Auto-reload 预装载使能 (Auto-Reload Preload Enable) 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	**CMS**	计数器对齐模式选择 (Counter Mode Selection) 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	**DIR**	计数方向寄存器 (counter Direction) 0：向上计数 1：向下计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读
3	**OPM**	单脉冲输出模式 (One Pulse Mode) 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	**URS**	更新请求选择 (Update Request Selection) 0：以下事件能够产生 update 中断或 DMA 请求 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	**UDIS**	禁止 update (Update Disable) 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器

位号	助记符	功能描述
		- 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	**CEN**	计数器使能 (Counter Enable) 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

26.5.2 ATIM 控制寄存器 2（ATIM_CR2）

名称	ATIM_CR2							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	CCUS	-	CCPC
位权限	R/W-0	R/W-000			R/W-0	R/W-0	U-0	R/W-0

位号	助记符	功能描述
31:15	**-**	RFU：未实现，读为 0
14	**OIS4**	参考 OIS1
13	**OIS3N**	参考 OIS1N
12	**OIS3**	参考 OIS1
11	**OIS2N**	参考 OIS1N
10	**OIS2**	参考 OIS1
9	**OIS1N**	定义 OC1N 的输出 IDLE 状态 (Output Idle State for OC1N) 0：当 MOE=0 时，经过 dead time 后，OC1N=0 1：当 MOE=0 时，经过 dead time 后，OC1N=1
8	**OIS1**	定义 OC1 的输出 IDLE 状态 (Output Idle State for OC1) 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=0 1：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=1
7	**TI1S**	ATIM 输入 TI1 选择 (Timer Input 1 Selection) 0：ATIM_CH1 引脚连接到 TI1 输入 1：ATIM_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 (Master Mode Selection) 000：ATIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器

位号	助记符	功能描述
		010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	**CCDS**	捕捉/比较 DMA 选择 (Capture/Compare DMA Selection) 0：捕捉/比较事件发生时发送 DMA 请求 1：Update Event 发生时发送 DMA 请求
2	**CCUS**	捕捉/比较控制寄存器更新选择 (Capture/Compare Update Selection) 0：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们仅在置位 COMG 寄存器时更新 1：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们在置位 COMG 寄存器或者 TRGI 上升沿时更新
1	**-**	RFU：未实现，读为 0
0	**CCPC**	捕捉/比较预装载控制 (Capture/Compare Preload Control enable) 0：CcxE, CcxNE, OcxM 寄存器不使能 preload 1：CcxE, CcxNE, OcxM 寄存器使能 preload 注意：此寄存器仅在拥有互补输出功能的通道上有效

26.5.3 ATIM 从机模式控制寄存器（ATIM_SMCR）

名称	ATIM_SMCR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
位权限	R/W-0	R/W-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	R/W-0	R/W-000			U-0	R/W-000		

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15	**ETP**	外部触发信号极性配置 (External Trigger Polarity) 0：高电平或上升沿有效 1：低电平或下降沿有效

位号	助记符	功能描述
14	**ECE**	外部时钟使能 (External Clock Enable) 0：关闭外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	**ETPS**	外部触发信号预分频寄存器 (External Trigger Prescaler) 外部触发信号 ETRP 的频率最多只能是 ATIM 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8 分频
11:8	**ETF**	外部触发信号滤波时钟和长度选择 (External Trigger Filter) 0000：无滤波 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	主机从机模式选 (Master Slave Mode) 0：无动作 1：触发模式下，TRGI 触发的动作被延迟，以便于通过 TRGO 实现当前定时器和从机定时器同步
6:4	**TS**	触发选择，用于选择同步计数器的触发源 (Trigger Source) 000：内部触发信号（ITR0） 001：内部触发信号（ITR1） 010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2） 111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	**-**	RFU：未实现，读为 0
2:0	**SMS**	从机模式选择 (Slave Mode Selection) 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP2 边沿，根据 TI1 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP1 边沿，根据 TI2 电平

位号	助记符	功能描述
		高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

26.5.4 ATIM DMA 和中断使能寄存器（ATIM_DIER）

名称	ATIM_DIER							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 (CC4 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
18	**CC3BURSTEN**	捕捉比较通道 3 的 DMA 模式配置 (CC3 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
17	**CC2BURSTEN**	捕捉比较通道 2 的 DMA 模式配置 (CC2 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
16	**CC1BURSTEN**	捕捉比较通道 1 的 DMA 模式配置 (CC1 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	-	RFU：未实现，读为 0
14	**TDE**	外部触发 DMA 请求使能 (Triggered DMA Enable) 0：从机模式下，禁止外部触发事件产生 DMA 请求 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器）

位号	助记符	功能描述
13	**COMDE**	COM 事件 DMA 请求使能 (COM event DMA Enable) 0：COM 事件发生时，禁止产生 DMA 请求 1：COM 事件发生时，允许产生 DMA 请求
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 (CC4 DMA Enable) 0：禁止 CC4 DMA 请求 1：允许 CC4 DMA 请求
11	**CC3DE**	捕捉比较通道 3 的 DMA 请求使能 (CC3 DMA Enable) 0：禁止 CC3 DMA 请求 1：允许 CC3 DMA 请求
10	**CC2DE**	捕捉比较通道 2 的 DMA 请求使能 (CC2 DMA Enable) 0：禁止 CC2 DMA 请求 1：允许 CC2 DMA 请求
9	**CC1DE**	捕捉比较通道 1 的 DMA 请求使能 (CC1 DMA Enable) 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	更新事件 DMA 请求使能 (Update Event DMA Enable) 0：Update Event 发生时，禁止产生 DMA 请求 1：Update Event 发生时，允许产生 DMA 请求
7	**BIE**	刹车事件中断使能 (Break event Interrupt Enable) 0：禁止刹车事件中断 1：允许刹车事件中断
6	**TIE**	触发事件中断使能 (Trigger event Interrupt Enable) 0：禁止触发事件中断 1：允许触发事件中断
5	**COMIE**	COM 事件中断使能 (COM event Interrupt Enable) 0：禁止 COM 事件中断 1：允许 COM 事件中断
4	**CC4IE**	捕捉/比较通道 4 中断使能 (CC4 Interrupt Enable) 0：禁止捕捉/比较 4 中断 1：允许捕捉/比较 4 中断
3	**CC3IE**	捕捉/比较通道 3 中断使能 (CC3 Interrupt Enable) 0：禁止捕捉/比较 3 中断 1：允许捕捉/比较 3 中断
2	**CC2IE**	捕捉/比较通道 2 中断使能 (CC2 Interrupt Enable) 0：禁止捕捉/比较 2 中断 1：允许捕捉/比较 2 中断
1	**CC1IE**	捕捉/比较通道 1 中断使能 (CC1 Interrupt Enable) 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	更新事件中断使能 (Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

26.5.5 ATIM 状态寄存器（ATIM_ISR）

名称	**ATIM_ISR**							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC4, write 1 to clear) 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC3, write 1 to clear) 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC2, write 1 to clear) 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断 (Over-Capture Interrupt Flag for CC1, write 1 to clear) 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0：无 overcapture 事件 1：在 CC1IF 标志为 1 的情况下发生新的捕捉
8	-	RFU：未实现，读为 0
7	**BIF**	刹车事件中断标志，硬件置位，软件写 1 清零 (Break Interrupt Flag, write 1 to clear)
6	**TIF**	触发事件中断标志，硬件置位，软件写 1 清零 (Trigger Interrupt Flag, write 1 to clear)
5	**COMIF**	COM 事件中断标志，硬件置位，软件写 1 清零 (COM Interrupt Flag, write 1 to clear)
4	**CC4IF**	捕捉/比较通道 4 中断标志 (CC4 Interrupt Flag, write 1 to clear) 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 (CC3 Interrupt Flag, write 1 to clear) 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 (CC2 Interrupt Flag, write 1 to clear) 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 (CC1 Interrupt Flag, write 1 to clear) 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果 CC1 通道配置为输入：发生捕捉事件时置位，软件写 1 清零，或者软件读 ATIM_CCR1 自动清零。
0	**UIF**	更新事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to clear) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复结束，并且 UDIS=0 的情况下，计数器发生溢出

位号	助记符	功能描述
		-URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

26.5.6 ATIM 事件产生寄存器（ATIM_EGR）

名称	ATIM_EGR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
位权限	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

位号	助记符	功能描述
31:8	**-**	RFU：未实现，读为 0
7	**BG**	软件刹车，软件置位此寄存器产生刹车事件，硬件自动清零 (Break Generate)
6	**TG**	软件触发，软件置位此寄存器产生触发事件，硬件自动清零 (Trigger Interrupt Flag)
5	**COMG**	软件 COM 事件，硬件置位，软件写 1 清零 (COMG Generate)
4	**CC4G**	捕捉/比较通道 4 软件触发，参考 CC1G (CC4 Generate)
3	**CC3G**	捕捉/比较通道 3 软件触发，参考 CC1G (CC3 Generate)
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G (CC2 Generate)
1	**CC1G**	捕捉/比较通道 1 软件触发 (CC1 Generate) 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 (User Generate) 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

26.5.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR1							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC		CC1S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE (OC2 Clear Enable)
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M (OC2 Mode)
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE (OC2 Preload Enable)
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE (OC2 Fast Enable)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 Channel Selection) 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7	**OC1CE**	输出比较 1 清零使能 (OC1 Clear Enable) 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M**	输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 (OC1 Mode) 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低；在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC1REF 在 CNT<CCR1 时置低，否则置高；在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	**OC1PE**	输出比较 1 预装载使能 (OC1 Preload Enable) 0：CCR1 preload 寄存器无效，CCR1 可以直接写入 1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器

位号	助记符	功能描述
		的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 (OC1 Fast Enable) 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况（只用于触发模式） 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC1S**	捕捉/比较 1 通道选择 (CC1 Channel Selection) 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

输入捕捉模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:12	**IC2F**	输入捕捉 2 滤波 (IC2 Filter)
11:10	**IC2PSC**	输入捕捉 2 预分频 (IC2 Prescaler)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 Channel Selection) 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC3 映射到 TI2 10：CC2 通道配置为输入，IC3 映射到 TI1 11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	**IC1F**	输入捕捉 1 滤波 (IC1 Filter) 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	输入捕捉 1 预分频 (IC1 Prescaler) 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC1S**	捕捉/比较 1 通道选择 (CC1 Channel Selection)

位号	助记符	功能描述
		00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

26.5.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR2							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC4F				IC4PSC		CC4S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC		CC3S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15	**OC4CE**	输出比较 4 清零使能，参考 OC1CE (OC4 Clear Enable)
14:12	**OC4M**	输出比较 4 模式配置，参考 OC1M (OC4 Mode)
11	**OC4PE**	输出比较 4 预装载使能，参考 OC1PE (OC4 Preload Enable)
10	**OC4FE**	输出比较 4 快速使能，参考 OC1FE (OC4 Fast Enable)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 Channel Selection) 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7	**OC3CE**	输出比较 1 清零使能(OC3 Clear Enable) 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC3M**	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 (OC3 Mode) 000：输出比较寄存器 CCR3 和计数器 CNT 的比较结果不会影响输出 001：CCR3=CNT 时，将 OC1REF 置高

位号	助记符	功能描述
		010：CCR3=CNT 时，将 OC1REF 置低 011：CCR3=CNT 时，翻转 OC1REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低；在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC3REF 在 CNT<CCR3 时置低，否则置高；在向下计数时，OC3REF 在 CNT>CCR3 时置高，否则置低
3	**OC3PE**	输出比较 3 预装载使能 (OC3 Preload Enable) 0：CCR3 preload 寄存器无效，CCR3 可以直接写入 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC3FE**	输出比较 3 快速使能 (OC3 Fast Enable) 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC3S**	捕捉/比较 3 通道选择 (CC4 Channel Selection) 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

输入捕捉模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:12	**IC4F**	输入捕捉 4 滤波 (IC4 Filter)
11:10	**IC4PSC**	输入捕捉 4 预分频 (IC4 Prescaler)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 3 滤波 (IC3 Filter) 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6

位号	助记符	功能描述
		1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频 (IC3 Prescaler) 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC3S**	捕捉/比较 3 通道选择 (CC3 channel Selection) 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

26.5.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER）

名称	ATIM_CCER							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:14	**-**	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P (CC4 Polarity)
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E (CC4 Enable)
11	**CC3NP**	捕捉/比较 3 互补输出极性，参考 CC1NP (CC3N Polarity)
10	**CC3NE**	捕捉/比较 3 互补输出使能，参考 CC1NE (CC3N Enable)
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P (CC3 Polarity)
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E (CC3 Enable)
7	**CC2NP**	捕捉/比较 2 互补输出极性，参考 CC1NP (CC2N Polarity)
6	**CC2NE**	捕捉/比较 2 互补输出使能，参考 CC1NE (CC2N Enable)

位号	助记符	功能描述
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P (CC2 Polarity)
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E (CC2 Enable)
3	**CC1NP**	捕捉/比较 1 互补输出极性 (CC1N Polarity) 0：OC1N 高电平为 active 1：OC1N 低电平为 active
2	**CC1NE**	捕捉/比较 1 互补输出使能 (CC1N Enable) 0：OC1N 无效，OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定
1	**CC1P**	捕捉/比较 1 输出极性 (CC1 Polarity) CC1 通道配置为输出时 0：OC1 高电平 active 1：OC1 低电平 active CC1 通道配置为输入时 0：非取反模式–捕捉在 IC1 的上升沿进行 1：取反模式–捕捉在 IC1 的下降沿进行
0	**CC1E**	捕捉/比较 1 输出使能 (CC1 Enable) CC1 通道配置为输出时 0：OC1 不 active 1：OC1 active CC1 通道配置为输入时 0：关闭捕捉功能 1：使能捕捉功能

26.5.10 ATIM 计数器寄存器（ATIM_CNT）

名称	ATIM_CNT							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:0	**CNT**	计数器值 (Counter)

26.5.11 ATIM 预分频寄存器（ATIM_PSC）

名称	ATIM_PSC

Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 (Prescaler) $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

26.5.12 ATIM 自动重载寄存器（ATIM_ARR）

名称	ATIM_ARR							
offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 (Auto-Reload Register) 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

26.5.13 ATIM 重复计数寄存器（ATIM_RCR）

名称	ATIM_RCR
Offset	0x00000030

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REP[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	**REP**	重复计数值。(Repetition) REP 不为 0 时，每次 update 条件发生时 REP 递减，当 REP=0 时触发 update 事件

26.5.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1）

名称	**ATIM_CCR1**							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**CCR1**	捕捉/比较通道 1 寄存器 (Capture/Compare channel 1 Register) **如果通道 1 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC1 输出 **如果通道 1 配置为输入：** CCR1 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR1 为只读

26.5.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2）

名称	ATIM_CCR2							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	CCR2	捕捉/比较通道 2 寄存器 (Capture/Compare channel 2 Register) **如果通道 2 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 **如果通道 2 配置为输入：** CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

26.5.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）

名称	ATIM_CCR3							
Offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	CCR3	捕捉/比较通道 3 寄存器 (Capture/Compare channel 3 Register) **如果通道 3 配置为输出：**

位号	助记符	功能描述
		这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

26.5.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4）

名称	ATIM_CCR4							
Offset	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:0	**CCR4**	捕捉/比较通道 4 寄存器 (Capture/Compare channel 4 Register) **如果通道 4 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC4 输出 **如果通道 4 配置为输入：** CCR4 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR4 为只读

26.5.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR）

名称	ATIM_BDTR							
Offset	0x00000044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	DTG
位权限	R/W-0000 0000

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	**MOE**	输出使能主控 (Master Output Enable) 此寄存器控制所有通道的输出使能，每个通道独立的输出使能还需要 CcxE 和 CcxNE 来控制。MOE 由软件置位，或者在 AOE=1 的情况下硬件触发自动置位。当刹车输入有效时，MOE 被硬件异步清零。 0：关闭 OC 和 OCN 输出，具体 IO 输出状态由 OSSI 决定 1：使能 OC 和 OCN 输出（仍需各个通道的 CcxE 和 CcxNE 状态来决定是否输出）
14	**AOE**	自动输出使能 (Automatic Output Enable) 0：MOE 仅能由软件置位 1：MOE 可以软件置位，或者由 update 事件自动置位
13	**BKP**	刹车极性 (Break Polarity) 0：刹车输入为低电平有效 1：刹车输入为高电平有效
12	**BKE**	刹车使能 (Break Enable) 0：禁止刹车输入 1：允许刹车输入
11	**OSSR**	运行状态下的输出关闭状态选择 (Off-State Select in Run mode) 仅在 MOE=1 的情况下，针对使能了互补输出的通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 驱动 GPIO 为无效状态
10	**OSSI**	IDLE 状态下的输出关闭状态选择 (Off-State Select in IDLE mode) 仅在 MOE=0 的情况下，针对输出通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 先驱动空闲状态，待死区时间结束后，启动无效状态
9:8	**LOCK**	寄存器写保护配置 (register write LOCK) 00：无写保护 01：保护等级 1 – DTG, OISx, OISxN, BKE, BKP, AOE 不能改写 10：保护等级 2 –在等级 1 基础上，CCxP, CCxNP, OSSR, OSSI 不能改写 11：保护等级 3 –在等级 2 基础上，OcxM, OcxPE 在相应通道配置为输出时不能改写 注意：LOCK 寄存器在被写入非 00 值之后无法再改写，写保护后的寄存器只有在 ATIM 模块被复位后才能重新写入。
7:0	**DTG**	死区时间插入，用于配置互补输出插入的死区时间长度 (Dead Time Generation) 000/001/010/011：DT=DTG[7:0] * t_{DTS} 100/101：DT=(64+DTG[5:0]) * 2 * t_{DTS} 110：DT=(32+DTG[4:0]) * 8 * t_{DTS} 111：DT=(32+DTG[4:0]) * 16 * t_{DTS}

26.5.19 ATIM DMA 控制寄存器（ATIM_DCR）

名称	ATIM_DCR							
Offset	0x00000048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			R/W-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 (DMA Burst Length) 对 ATIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~18 00000：长度=1 00001：长度=2 00010：长度=3 00011：长度=4 00100：长度=5 00101：长度=6 00110：长度=7 00111：长度=8 01000：长度=9 01001：长度=10 01010：长度=11 01011：长度=12 01100：长度=13 01101：长度=14 01110：长度=15 01111：长度=16 10000：长度=17 10001：长度=18 其他：无效值，禁止写入
7:5	-	RFU：未实现，读为 0
4:0	**DBA**	DMA 基地址，定义指向寄存器的偏移地址 (DMA Burst Address) 00000：ATIM_CR1 00001：ATIM_CR2 00010：ATIM_SMCR …… 注意：当 DBA+DBL 超出了 ATIM 寄存器地址范围，则实际 burst 传输到 ATIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

26.5.20 ATIM DMA 访问寄存器（ATIM_DMAR）

名称	ATIM_DMAR							
Offset	0x0000004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**DMAR**	DMA burst 访问寄存器 (DMA burst access Register) 在使用 DMA burst 传输时，将 DMA 通道外设地址设置为 ATIM_DMAR，ATIM 会根据 DBL 的值产生多次 DMA 请求

26.5.21 ATIM 刹车输入控制寄存器（ATIM_BKCR）

名称	ATIM_BKCR							
Offset	0x00000060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						BRK2GATE	BRK1GATE
位权限	U-0						R/W-1	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRKF				BRKCOMB	HFDET_BRKEN	SVD_BRKEN	COMP_BRKEN
位权限	R/W-0000				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU，未实现，读为 0
9	**BRK2GATE**	ATIM_BRK2 引脚输入门控信号 (Break 2 Gate) 0：将 ATIM_BRK2 的输入门控成 0 1：不门控
8	**BRK1GATE**	ATIM_BRK1 引脚输入门控信号 (Break 1 Gate) 0：将 ATIM_BRK1 的输入门控成 0

位号	助记符	功能描述
		1：不门控
7:4	**BRKF**	刹车信号的滤波时钟和长度选择 (Break Filter) 0000：无滤波 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3	**BRKCOMB**	刹车组合控制 (Break Combination) 0：两路刹车信号相或 1：两路刹车信号相与
2	**HFDET_BRKEN**	XTHF 停振检测刹车信号使能 (HFDET Break Enable) 0：禁止 HFDET 刹车信号 1：使能 HFDET 刹车信号
1	**SVD_BRKEN**	SVD 刹车信号使能 (SVD Break Enable) 0：禁止 SVD 刹车信号 1：使能 SVD 刹车信号
0	**COMP_BRKEN**	比较器输出刹车信号使能 (Comparator Break Enable) 0：禁止比较器刹车信号 1：使能比较器刹车信号

27 通用定时器（GPTIM）

27.1 概述

FM33LC0XX包含2个通用定时器。

通用定时器包含一个16bit自动重载计数器及一个可编程预分频器。

通用定时器可以支持多种应用，包括如捕捉、输出比较、PWM。

27.2 主要特性

- 16bit向上、向下、双向计数自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- 灵活的计数时钟源选择，使用部分时钟可以在休眠模式下运行
- 4个独立通道可用于输入捕捉、输出比较、PWM（边缘或中心对齐模式）、单脉冲输出
- 支持与其他定时器级联
- 支持在以下事件发生时产生中断
  - 计数器溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕捉
  - 输出比较

27.3 结构框图

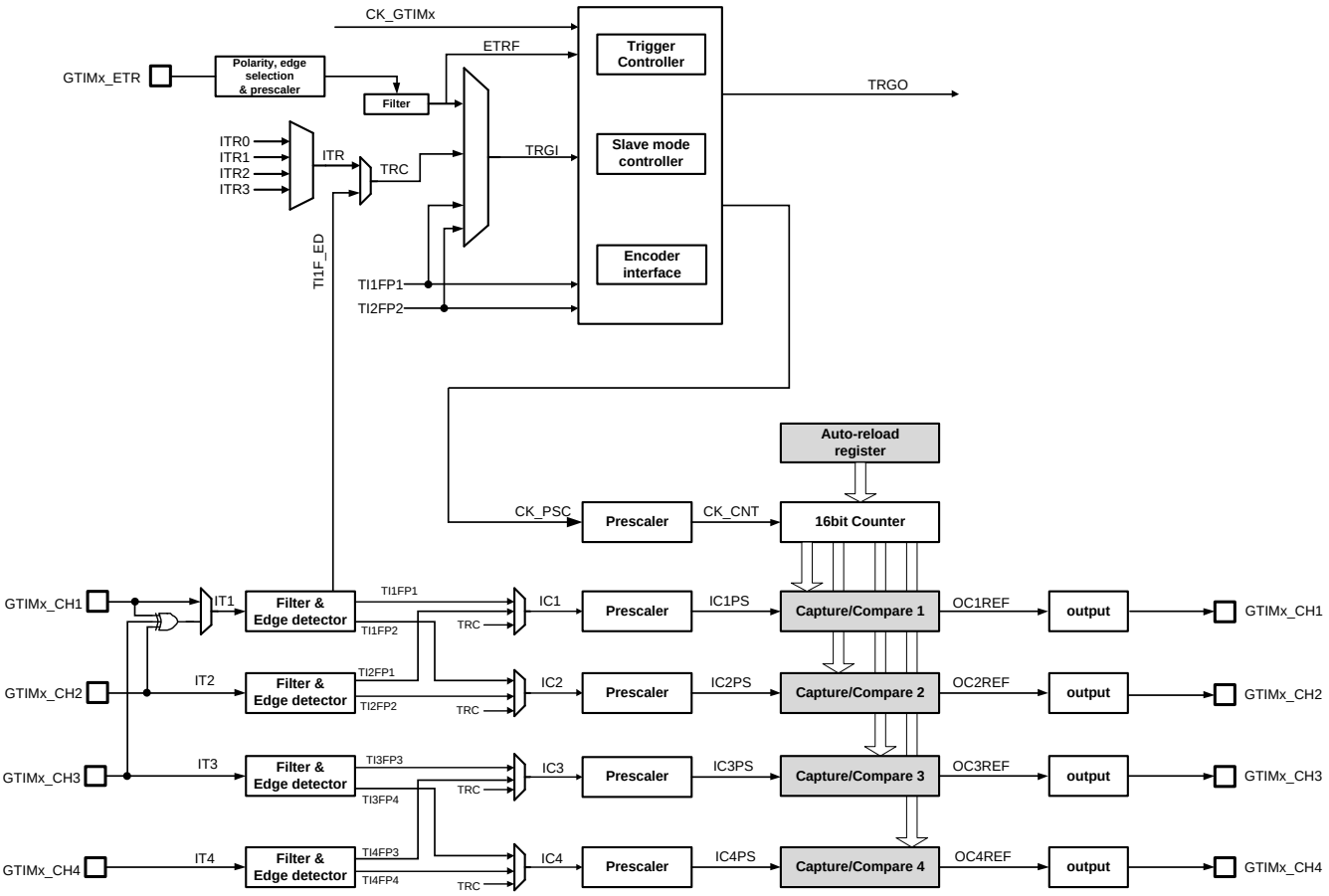


图27-1通用定时器架构示意图

27.4 功能描述

27.4.1 定时单元

高级定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（GPTIM_CNT）
- 预分频寄存器（GPTIM_PSC）
- 自动重载寄存器（GPTIM_ARR）
- 重复计数寄存器（GPTIM_RCR）

ARR包含预装载功能，该功能通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=0时，对ARR寄存器执行写入，写入数据将直接传入到影子寄存器；当ARPE=1时，对ARR寄存器执行写入的数据在update event（GPTIM_CNT上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发ARR更新（UEV）。

GPTIM_CNT工作时钟由GPTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

GPTIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际不改写影子寄存器，只有当新的update event到来时，才会从PSC更新至影子寄存器。因此在CNT计数过程中，软件可以实时改写PSC，而新的预分频比将在下一更新事件发生时被采用。

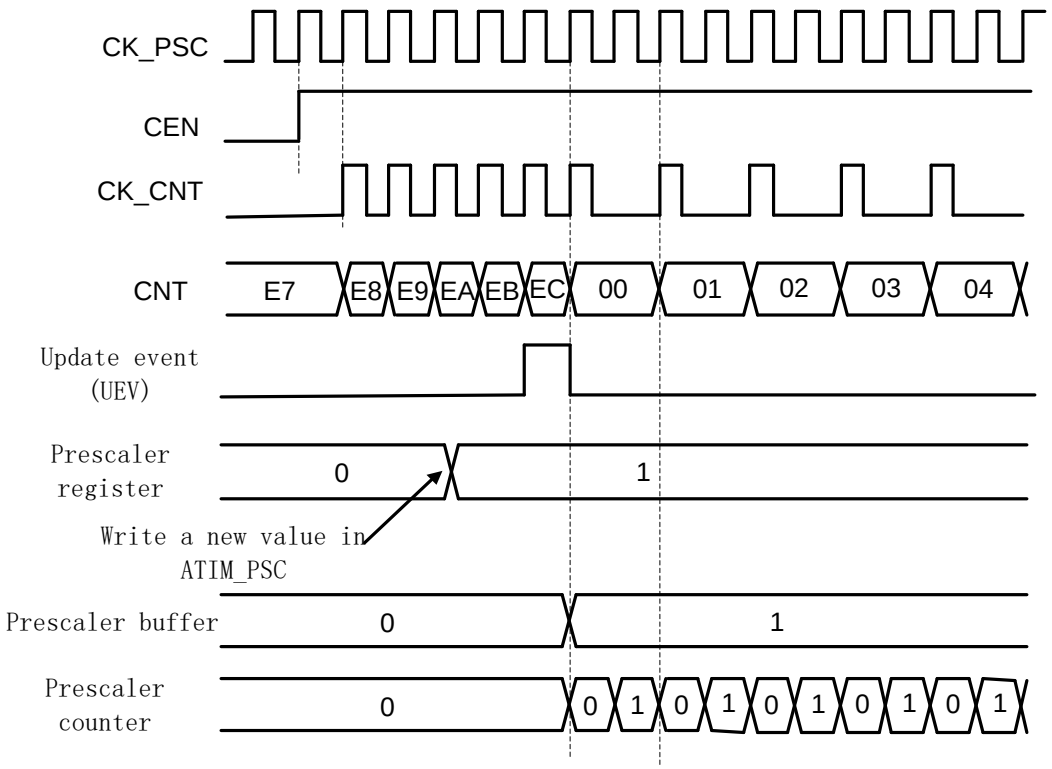


图 27-2 预分频从 1 变为 2 的波形

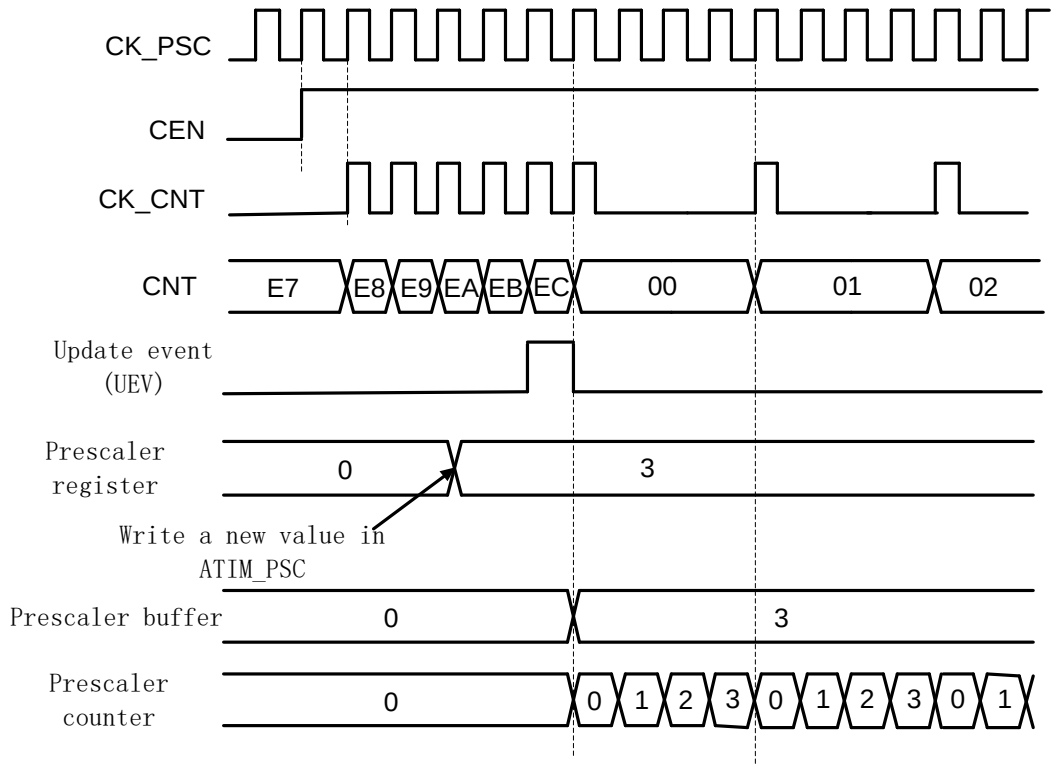


图 27-3 预分频从 1 变为 4 的波形

27.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

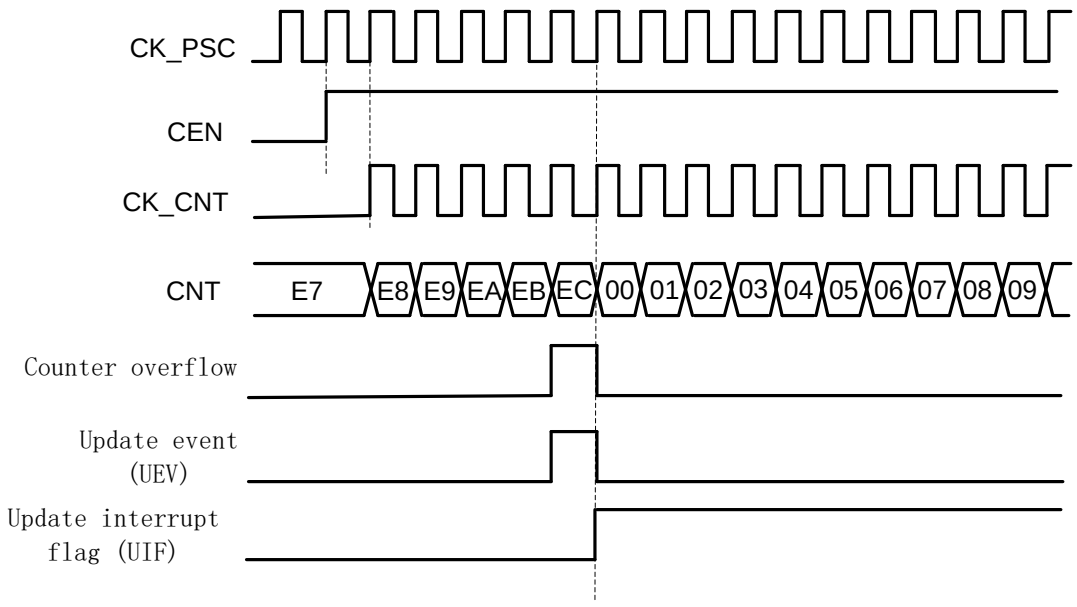


图 27-4 向上计数波形，内部时钟不分频

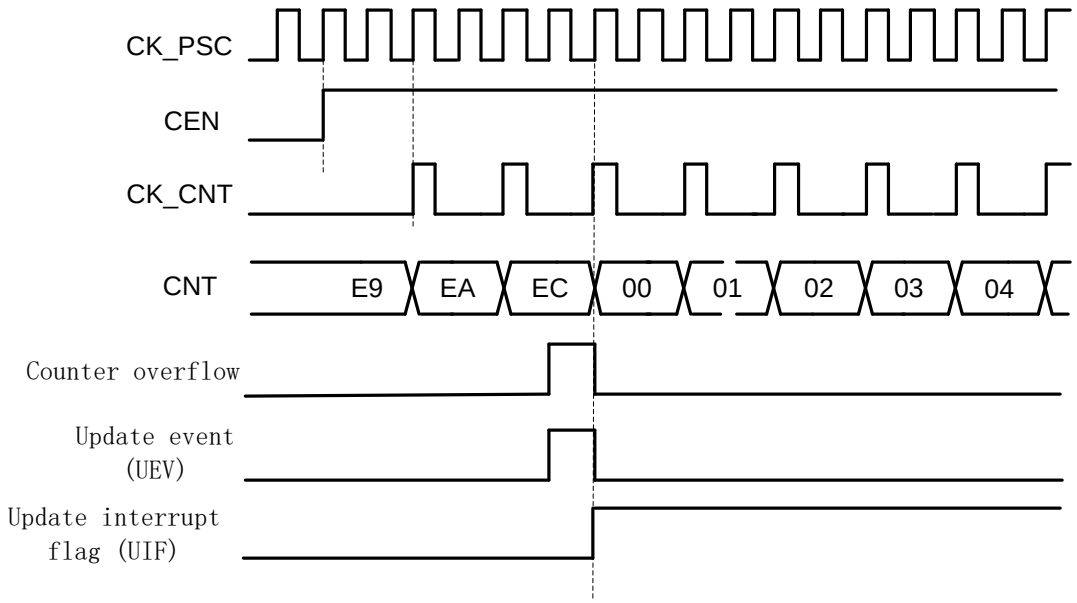


图 27-5 向上计数波形，内部时钟 2 分频

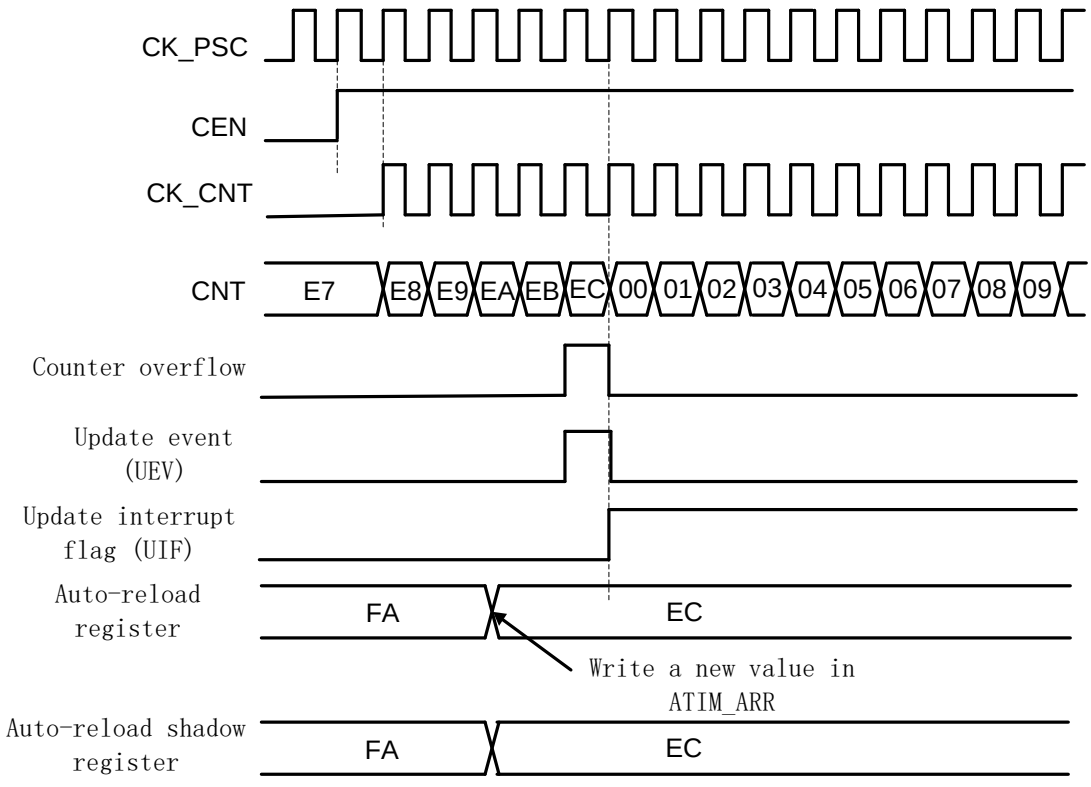


图 27-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

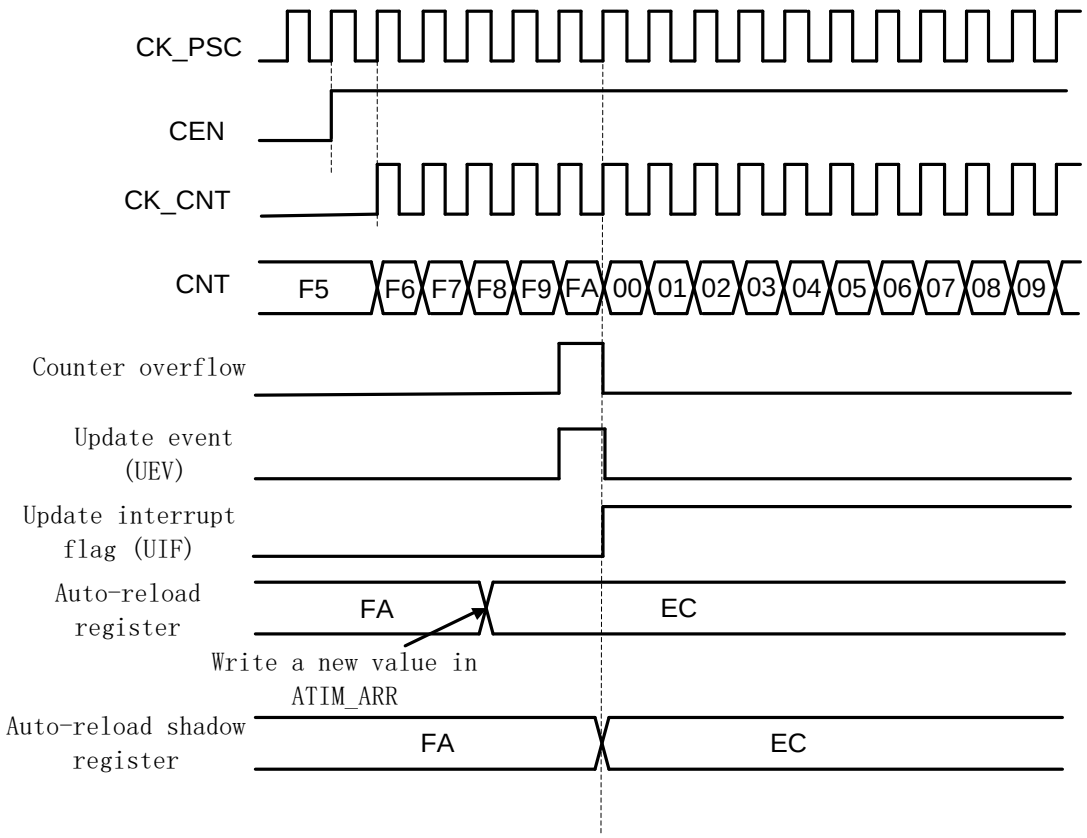


图 27-7ARPE=1（ATIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从ARR值开始递减，到0后产生下溢出事件，并且重新从ARR开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

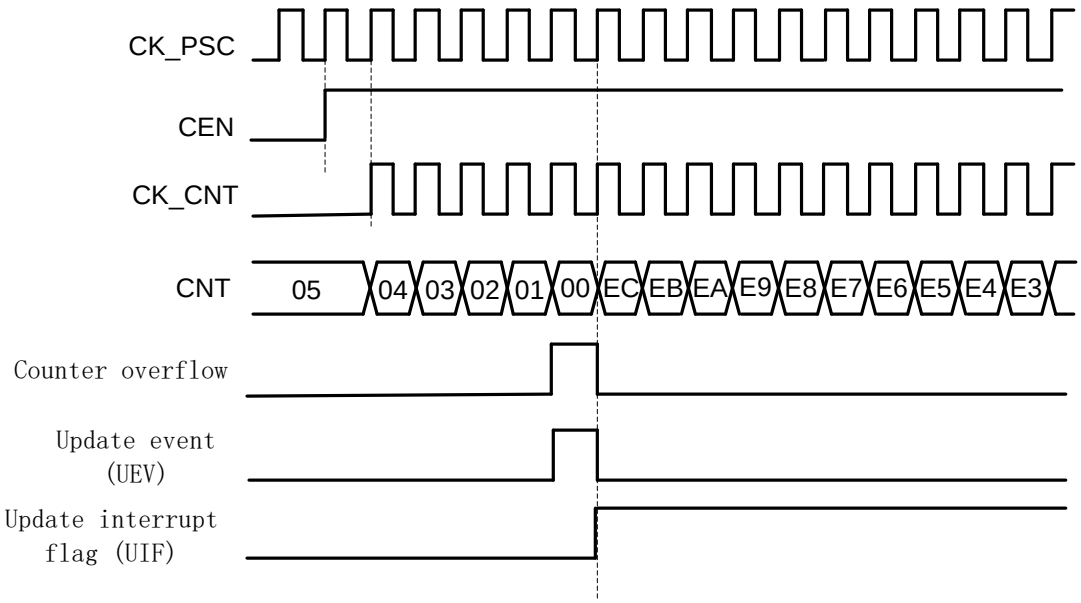


图 27-8 向下计数，内部时钟不分频

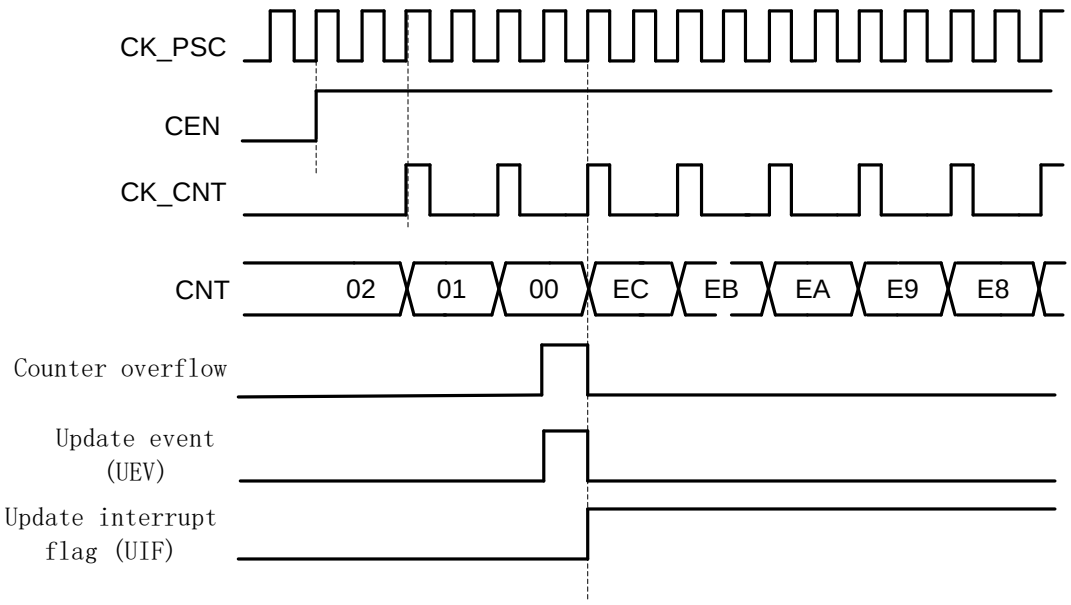


图 27-9 向下计数，内部时钟 2 分频

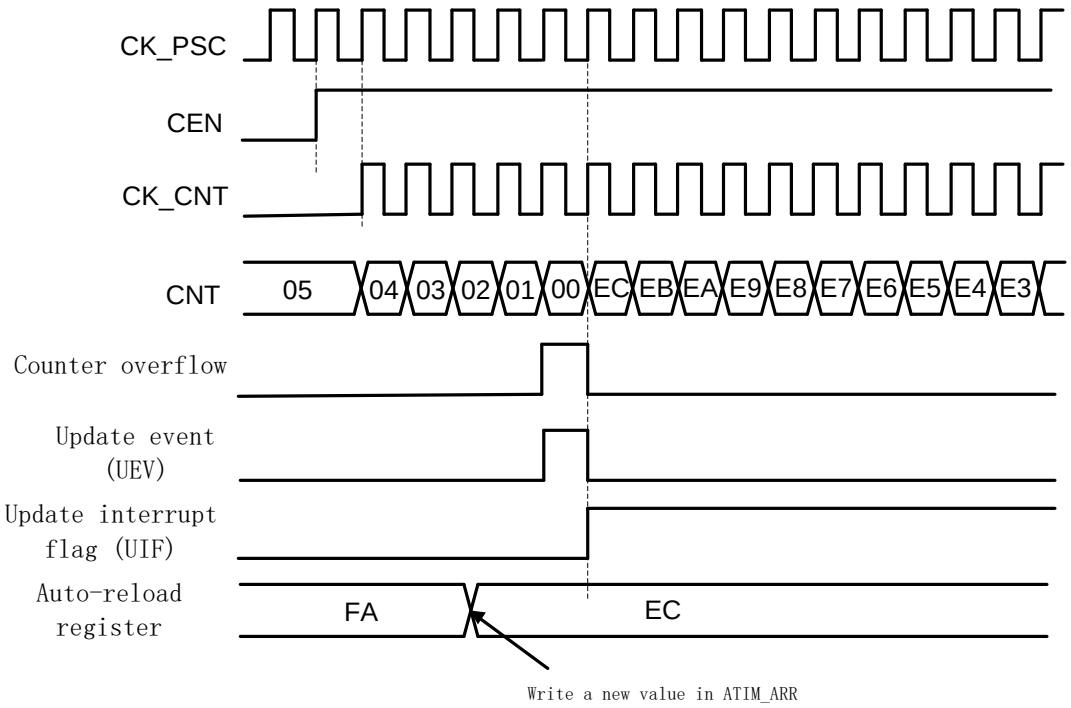


图 27-10 向下计数，内部时钟 2 分频

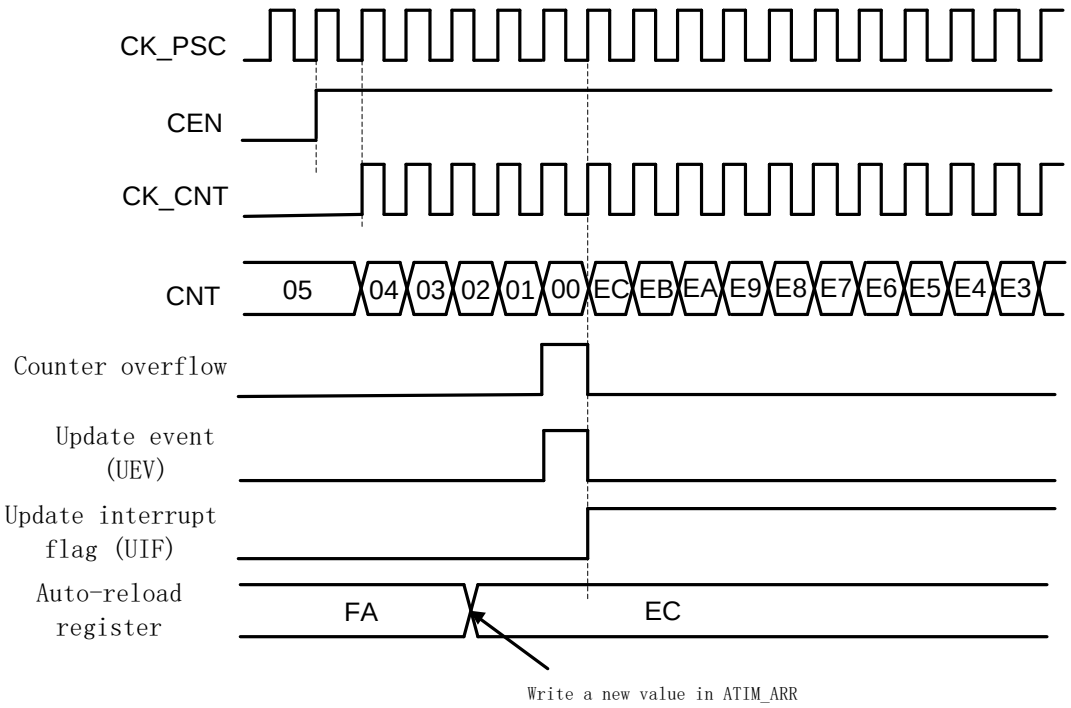


图 27-11 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从0开始向上计数，到ARR-1产生上溢出事件，然后从ARR开始向下计数到1，产生下溢出事件，再从0重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当CMS!=00时为中心对齐计数，当CMS=01时，输出比较功能仅在向下计数时有效，当CMS=10时，输出比较功能仅在向上计数时有效，当CMS=11时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在overflow和underflow的事件上都会更新 ARR、PSC和RCR的影子寄存器。

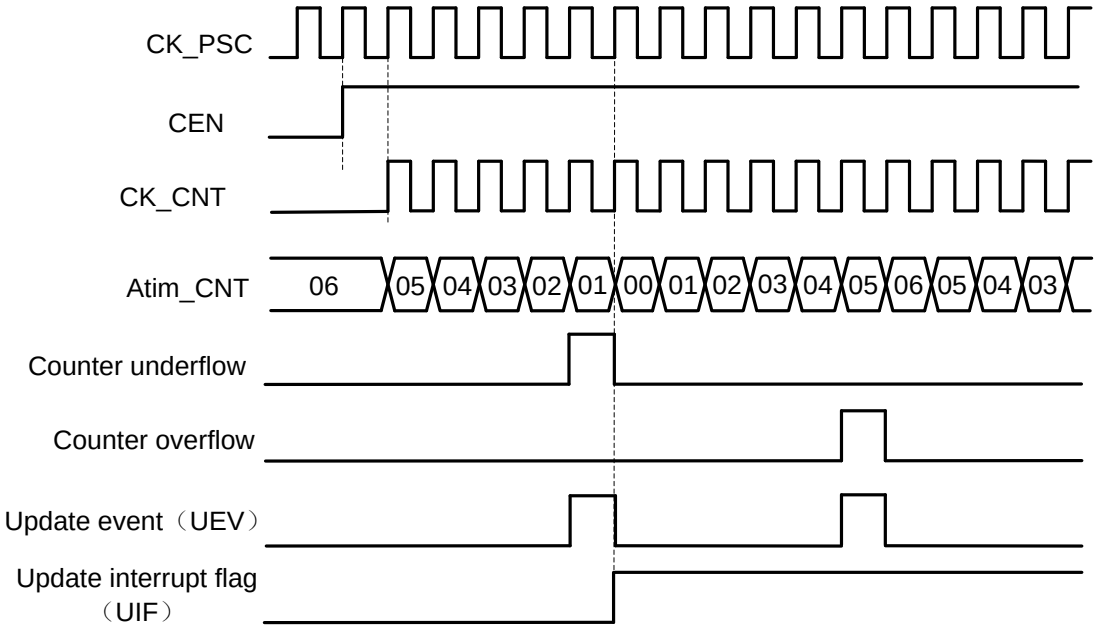


图 27-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

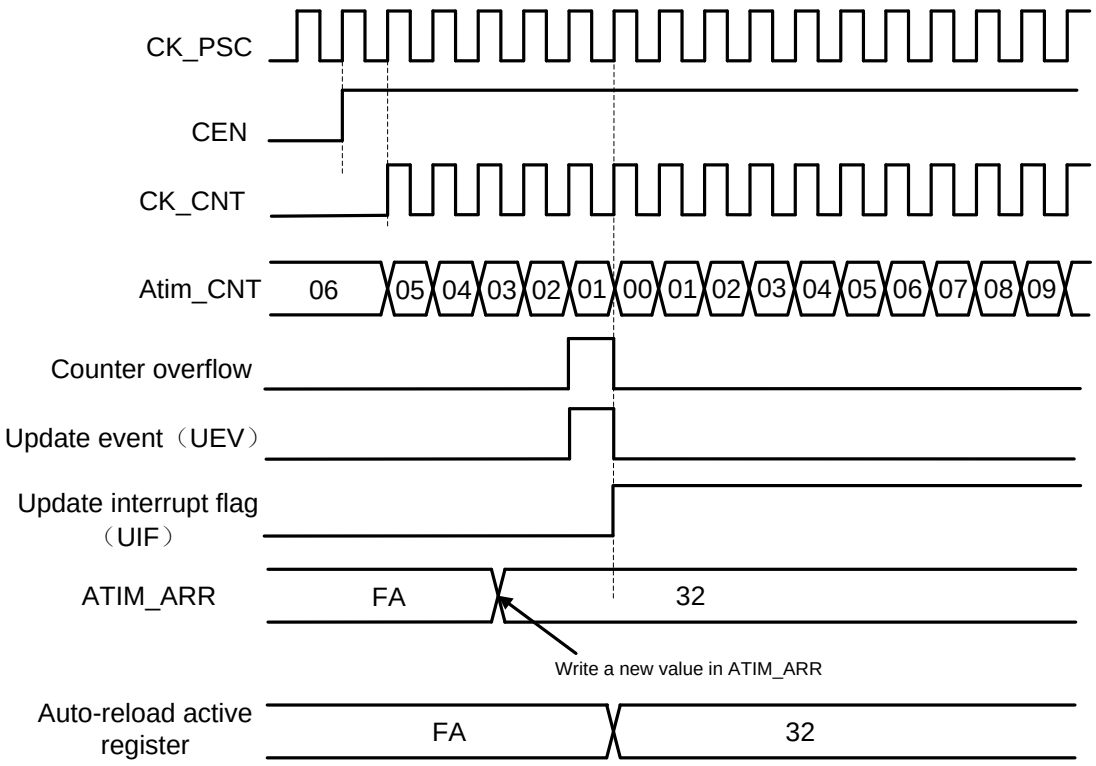


图 27-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

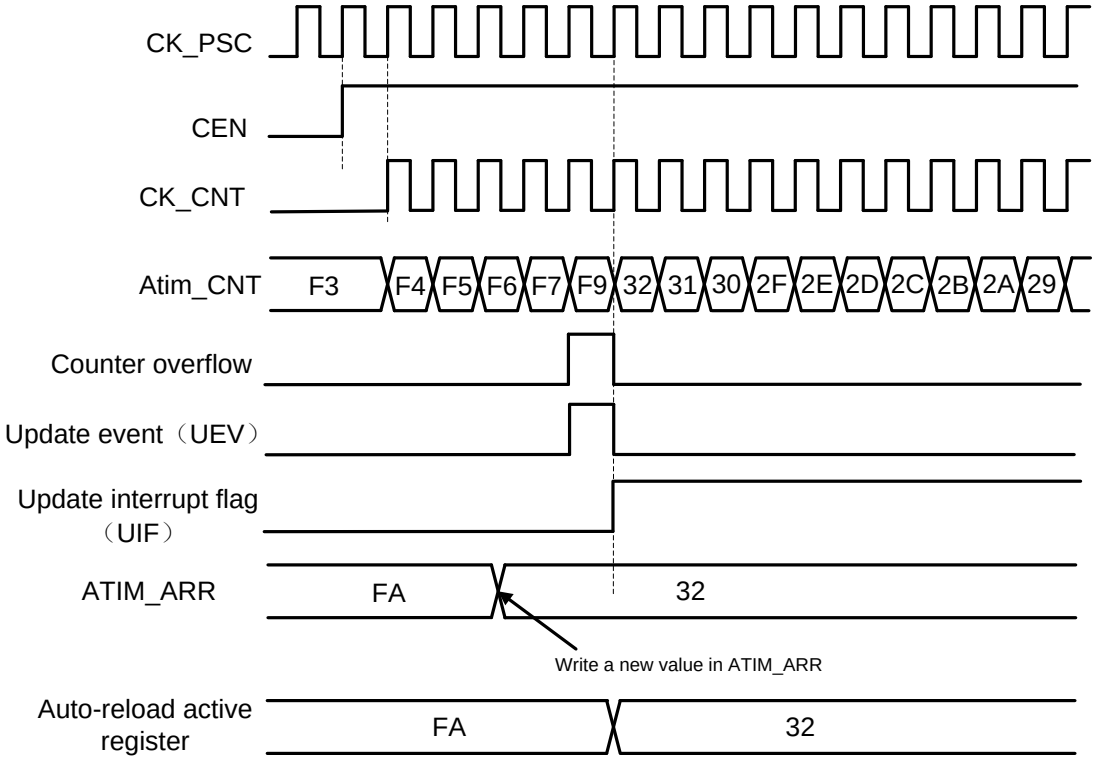


图 27-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

27.4.3 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（Tix）——外部时钟模式1
- 外部引脚触发输入（ETR）——外部时钟模式2
- 内部触发（ITRx）——使用一个timer的触发输出（TGO）作为计数时钟

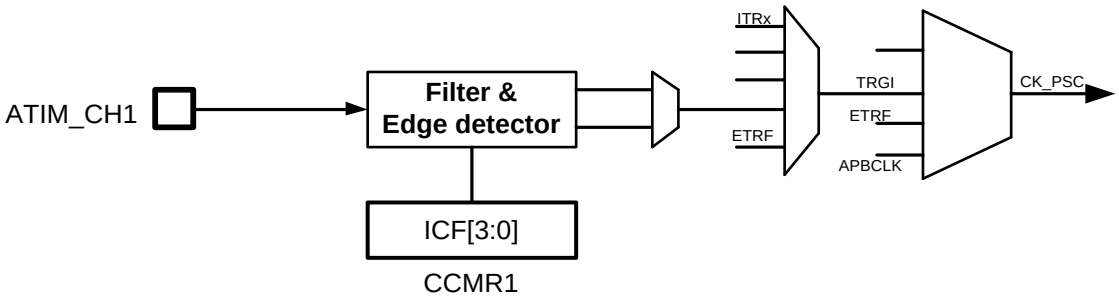


图 27-15 ATIM 时钟源框图

27.4.3.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

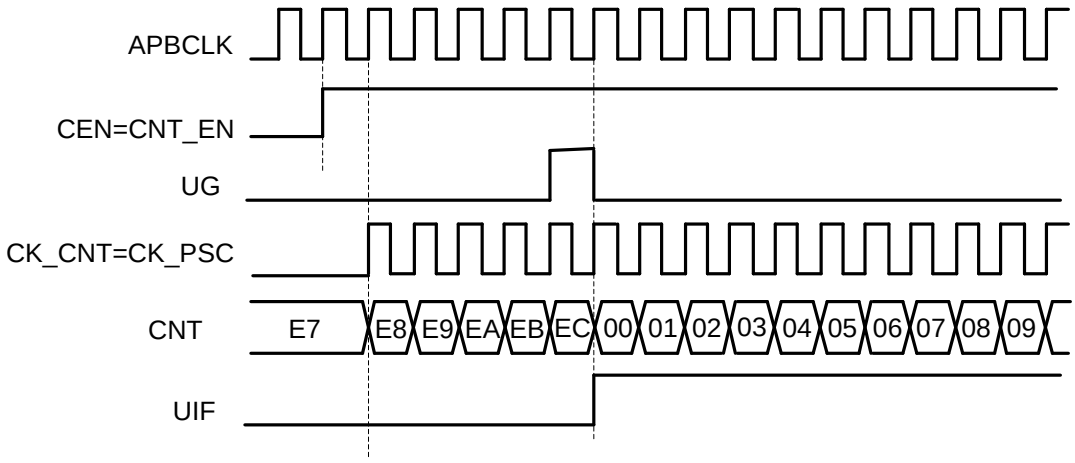


图 27-16 内部时钟源模式，时钟分频因子为 1

27.4.3.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置SMS=111，计数边沿可以配置为上升或下

降沿。

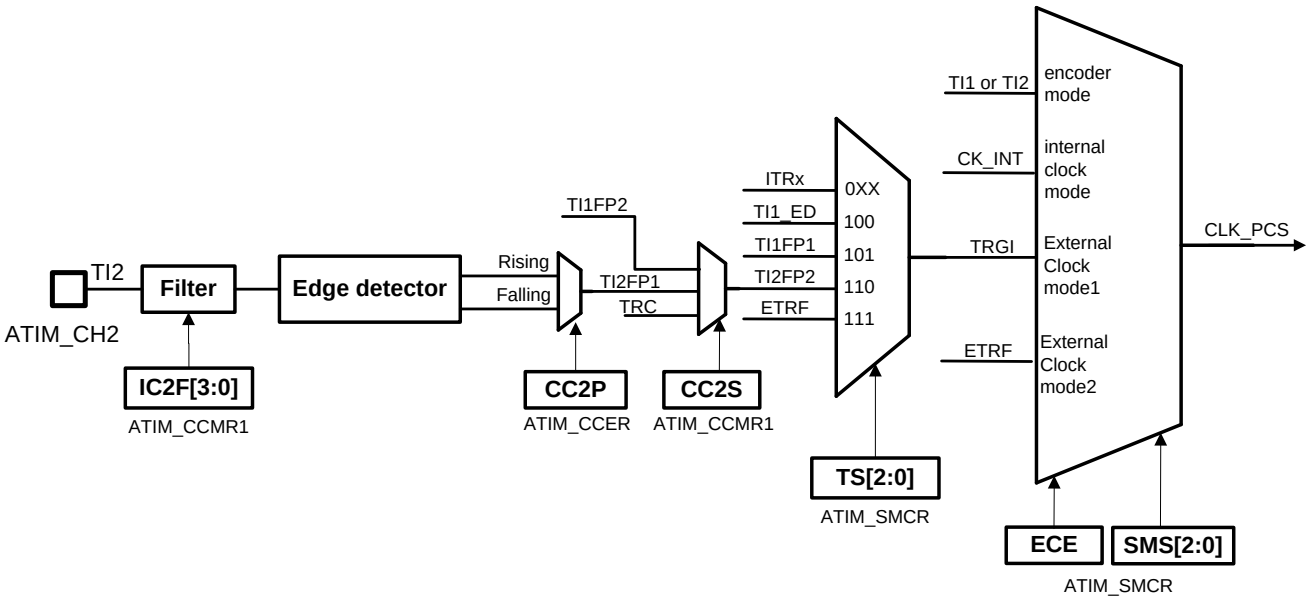


图 27-17TI2 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发TIF标志

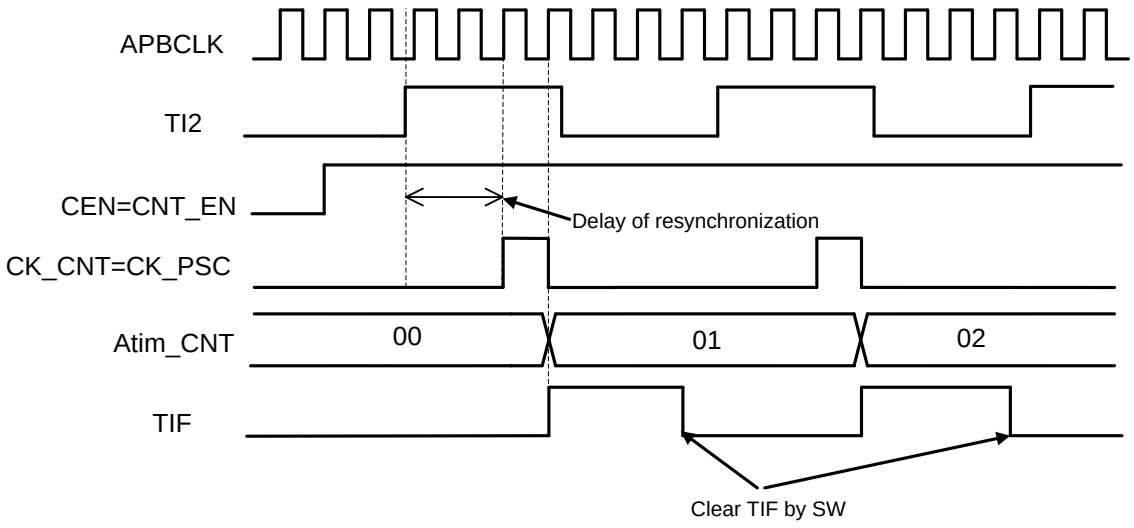


图 27-18 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能GPTIM的内部时钟（APBCLK），因为GPTIM要使用APB_CLK来对外部输入时钟进行同步和滤波。在外部时钟模式1下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的2级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于2

个APB_CLK周期。

此模式下只有通道1和2的输入可以用做时钟输入，所需配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置GPTIM_SMCR.SMCR=111
- 选择触发输入源，配置GPTIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置GPTIM_CCER.CC2E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

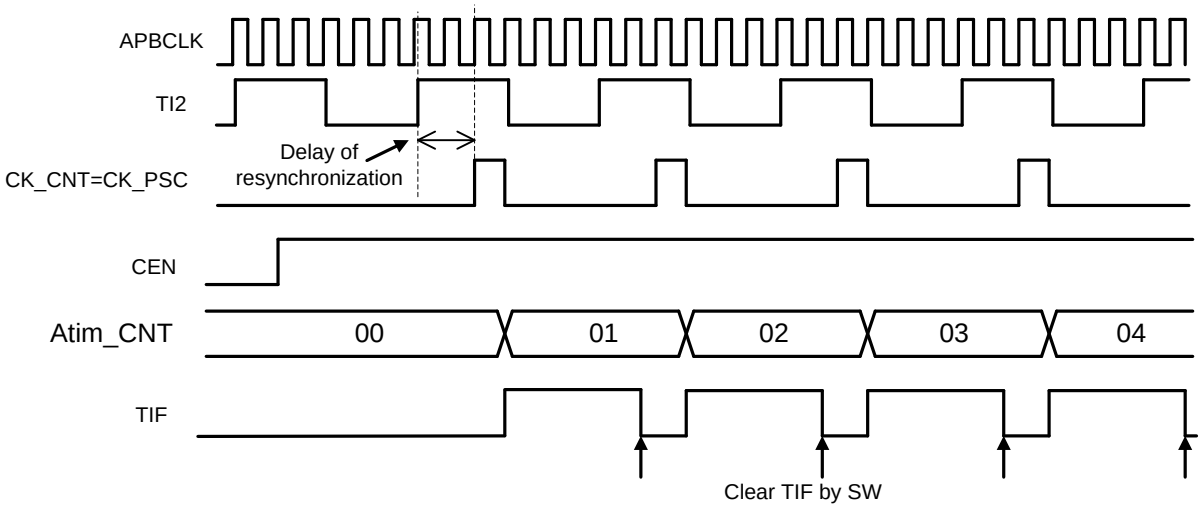


图 27-19 外部时钟模式 1 下的时序

27.4.3.3 外部时钟模式 2

此模式下使用GPTIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

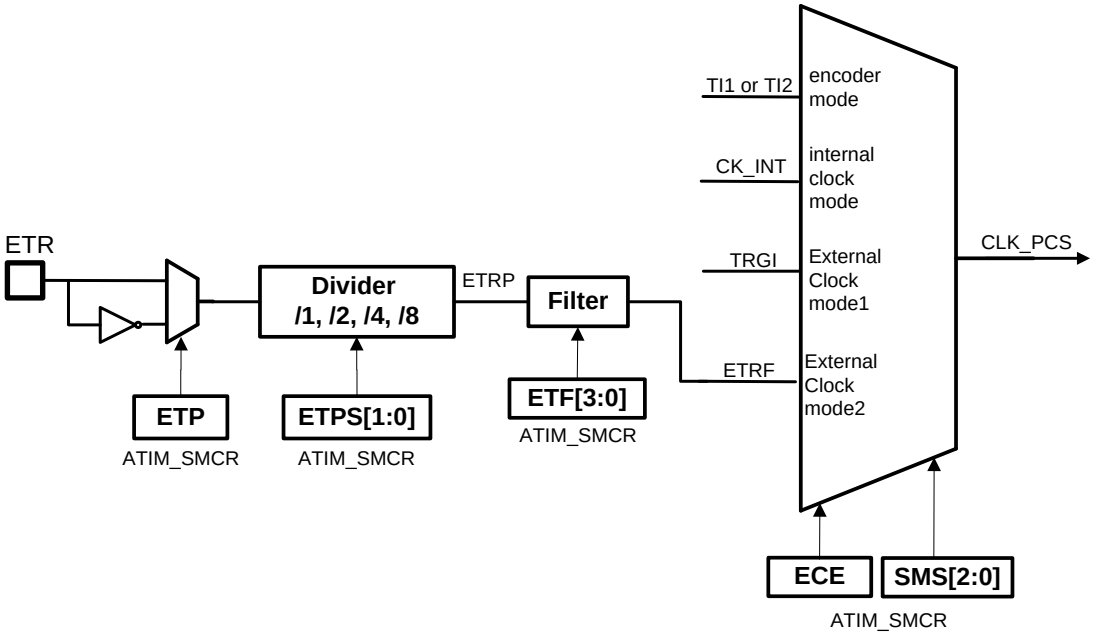


图 27-20 外部触发输入框图

下图是使用ETR二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于ETR输入上升沿。

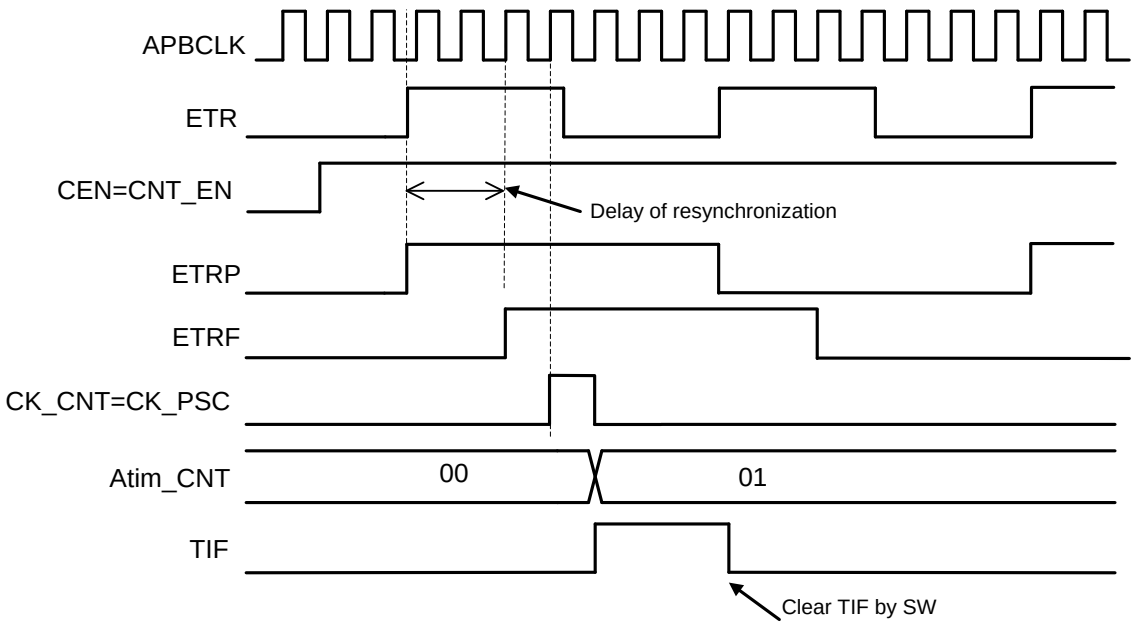


图 27-21 外部时钟模式 2 下的时序 1

与外部时钟模式1的主要差别是，ETR输入直接被分频后再进行滤波，产生CK_PSC时钟，这意味着可以支持ETR输入频率高于APB_CLK的应用场景，这种情况下，需要首先对ETR输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_ETR功能
- 设置ETP进行沿选择，GPTIM_SMCR.ETP=0
- 设置ETR分频比，配置GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,GPTIM_SMCR.ECE=1，GPTIM_SMCR.SMS=000
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

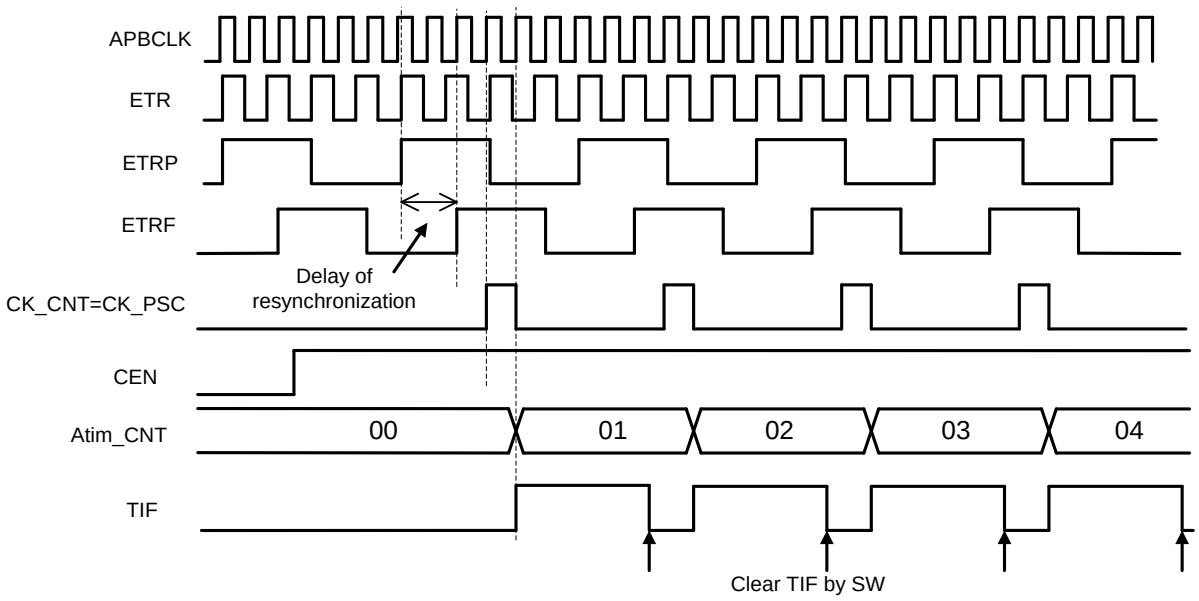


图27-22外部时钟模式2下的时序2

在使用外部时钟模式2时，仍可以将GPTIM配置为slave模式：比如使用ETR输入计数，同时使用另一个Timer的TRGO作为触发信号，当触发事件到来时，复位计数器重新开始计数。

27.4.3.4 内部触发模式

每个GPTIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当ITR选择为计数触发信号时，GPTIM计数器将在每个ITR信号的高电平期间计数，或者由ITR信号上升沿触发计数。通过内部触发模式可以实现Timer级联。

配置TIMx为master mode并周期性输出TRGO脉冲信号，TIMy配置为Slave mode并将TIMx的TRGO设置为ITR；当TIMx.TRGO脉冲到来时，TIMy计数一次。

基于内部触发模式的timer级联有如下要求：

- TRGO信号设计为APBCLK单周期脉冲
- TIMx和TIMy都工作在APBCLK时钟域
- TRGO对于接收方来说是一个同步脉冲
- Master和Slave的工作时钟都必须使能

内部触发模式可以使用的触发信号除了其他定时器输出外，还可以是ADC_EOC或者比较器模式输出。

27.4.4 内部触发信号（ITRx）的捕捉

每个GPTIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。通过这个方法，Timer可以捕捉各种芯片内部信号的周期或电平宽度。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

GPTIM0			Function
ITR0SEL	00	ATIM_TRGO	计数触发
	01	UART0_RX	宽度捕捉
	10	UART1_RX	宽度捕捉
	11	UART4_RX	宽度捕捉
ITR1SEL	00	GPTIM1_TRGO	计数触发
	01	XTHF	周期捕捉
	10	RCHF	周期捕捉
	11	LPUART0_RX	周期捕捉
ITR2SEL	00	BSTIM_TRGO	计数触发
	01	LPUART1_RX	宽度捕捉
	10	LPOSC	周期捕捉
	11	XTLF	周期捕捉
ITR3SEL	00	COMP1_TRGO	计数触发
	01	RCMF	周期捕捉
	10	COMP2_TRGO	计数触发
	11	LPT32_TRGO	计数触发
GPTIM1			**Function**
ITR0SEL	00	ATIM_TRGO	计数触发
	01	UART0_RX	宽度捕捉
	10	UART1_RX	宽度捕捉
	11	UART4_RX	宽度捕捉
ITR1SEL	00	GPTIM0_TRGO	计数触发
	01	XTHF	周期捕捉
	10	RCHF	周期捕捉
	11	ADC_EOC_TRGO	计数触发
ITR2SEL	00	BSTIM_TRGO	计数触发
	01	LSCLK	周期捕捉
	10	LPOSC	周期捕捉
	11	XTLF	周期捕捉
ITR3SEL	00	COMP1_TRGO	计数触发
	01	RCMF	周期捕捉
	10	COMP2_TRGO	计数触发
	11	LPT32_TRGO	计数触发

表 27-1 内部触发信号表

软件应保证选择正确的信号用于正确的功能，错误的配置将导致完全错误的结果。比如将ATIM_TRGO用于宽度捕捉，则结果没有意义。

27.4.5 捕捉/比较通道

GPTIM包含4个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样Tix输入并产生滤波后的信号TixF，然后边沿检测和极性选择产生对应的TixFPx信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

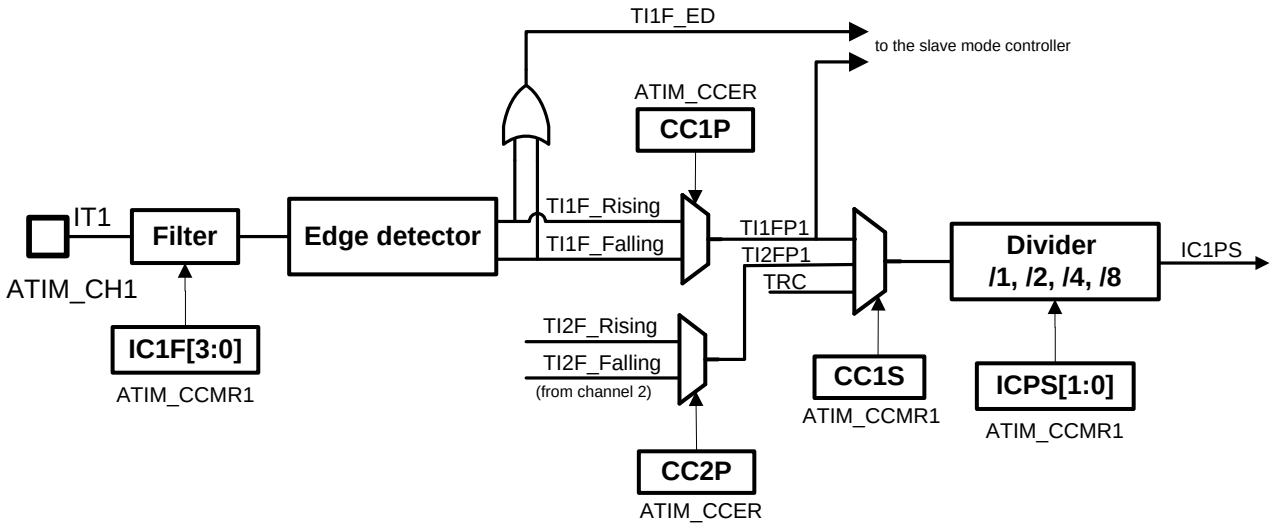


图27-23捕获/比较通道(通道1输入部分)

输出级电路会产生一个输出基准信号OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。GPTIM输出通道不支持互补输出。

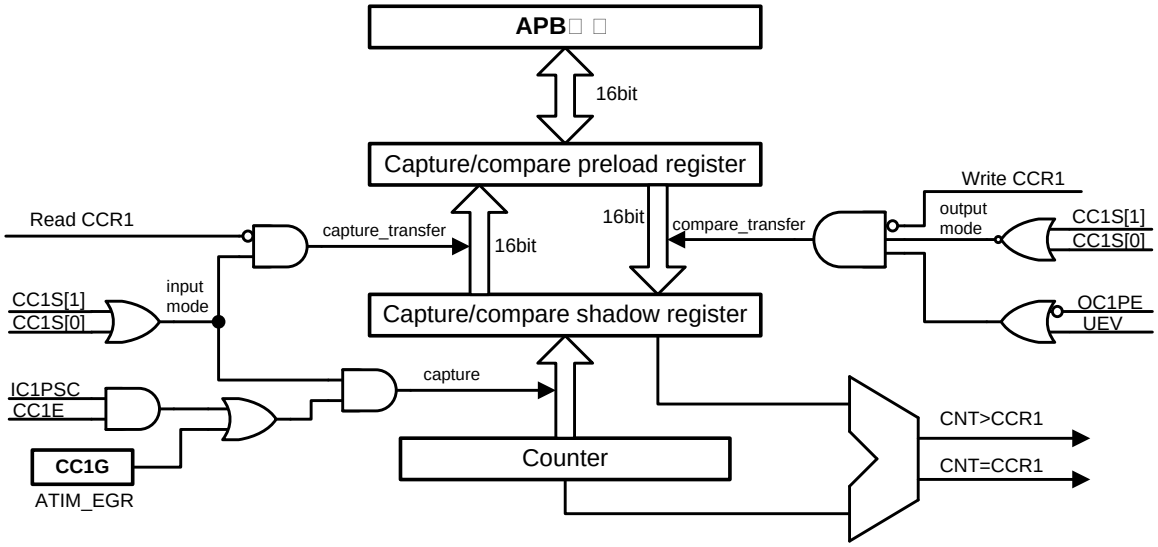


图27-24捕获/比较通道1的主电路

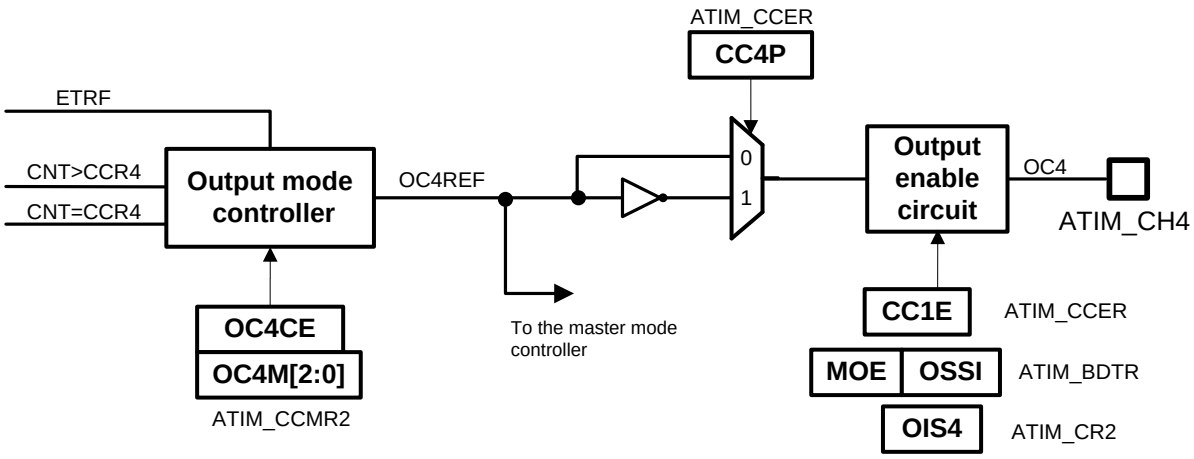


图27-25捕获/比较通道的输出部分

捕捉/比较寄存器（CCR）包含preload寄存器和shadow寄存器，软件读写总是访问preload寄存器。在捕捉模式下，捕捉值保存在shadow寄存器中并复制到preload寄存器。在比较模式下，preload寄存器的值被拷贝到shadow寄存器用来与计数器比较。

27.4.6 输入捕捉模式

当Icx信号上出现预期的电平变换，将触发一次capture，当前计数器值被锁存进CCR，与此同时，CcxIF中断标志置位，并且可以触发对应的中断或者DMA请求。如果一个捕捉事件在CcxIF为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR中上次捕捉值被覆盖）。CcxIF可以由软件清零，或者通过读取CCR寄存器自动清零。CcxOF标志通过软件写1清零。

通过两个或更多通道配合，可以实现PWM信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从TI1引脚输入，芯片内部将滤波后的信号取上升沿得到TI1FP1，将滤波后的信号取下降沿得到TI1FP2，将TI1FP1输入给捕捉通道1，将TI1FP2输入给捕捉通道2，即可实现通道1对输入信号上升沿捕捉，同时通道2对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过CCR1和CCR2寄存器的值，即可计算输入信号的周期和占空比。

实现在TI1输入的上升沿捕获计数器的值到GPTIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置GPTIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置GPTIM_CCER.CC1E=1

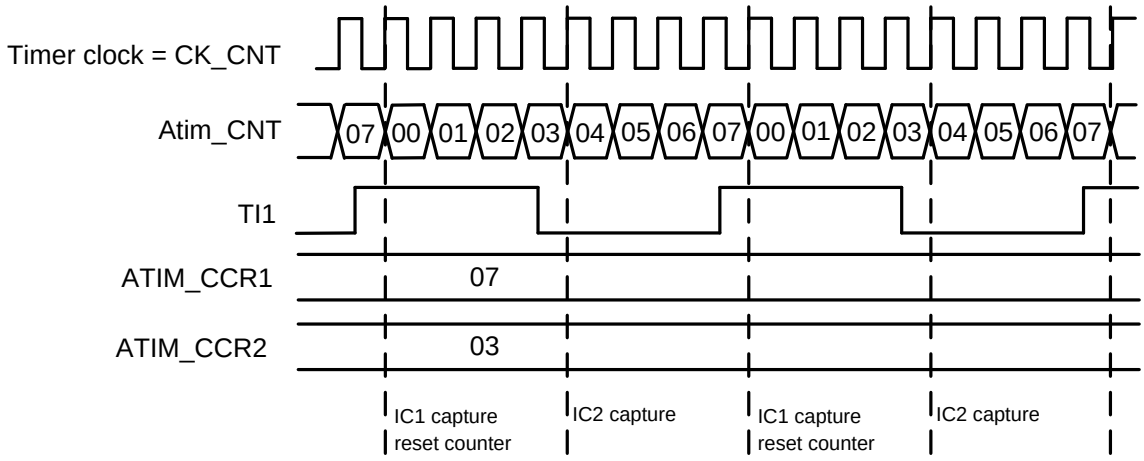


图27-26PWM输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0确保之后通道配置成功
- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置GPTIM_CCMR1.CC1S=01, GPTIM_CCMR1.CC2S=10
- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=1
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]，GPTIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0] ，GPTIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

27.4.7 软件 Force 输出

在比较输出模式下，软件可以直接将OCxREF force成特定电平，而独立于CCR和计数器的比较结果。

软件通过写OcxM=101寄存器，可以直接将OCxREF强制为有效（OCxREF固定为高有效），通过写OcxM=100可以直接将OCxREF强制为无效（低电平）。但是软件force操作不会取消比较过程，CCR和计数器的比较还会一直进行。

27.4.8 输出比较模式

输出比较模式下，当CCR与计数器值相等，OCxREF可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA请求可以发送。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

1、选择计数时钟（内部、外部、预分频等）

2、向ARR和CCR寄存器写入期望数据

3、根据需要设置中断使能和DMA使能

4、选择输出模式

5、使能计数器

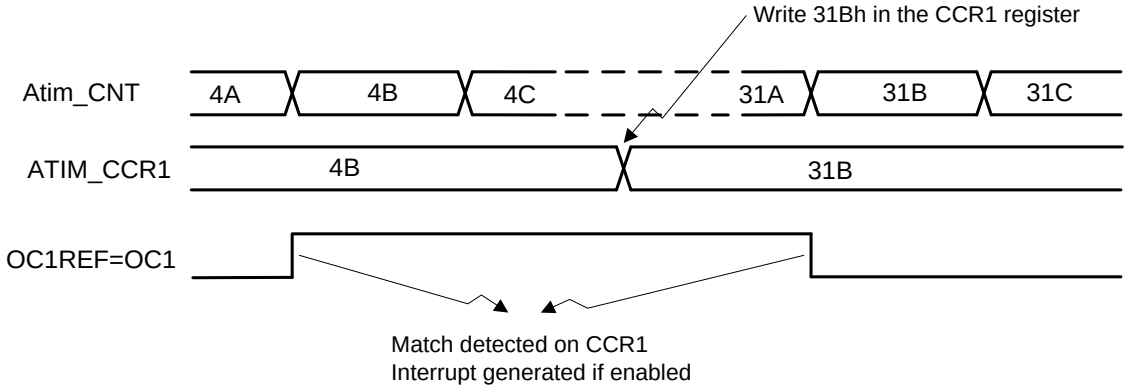


图27-27输出比较模式，翻转OC1

在不使能preload的情况下，软件可以随时改写CCR寄存器实现对输出波形的实时控制。如果使能了preload，则CCR shadow寄存器仅在下一次update event发生时更新为preload寄存器的内容。

27.4.9 PWM 输入模式

PWM模式可以输出脉宽调制信号，其周期由ARR寄存器决定，占空比由CCR寄存器决定。

输出信号的极性可以由CCxP寄存器配置。PWM模式工作中，CNT和CCR实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM输出也支持边缘对齐和中央对齐模式。

PWM边缘对齐模式

在向上计数的情况下，配置为PWM模式1时，OCxREF信号在CNT<CCR时为高电平，否则为低电平。如果CCR值大于ARR值，则OCxREF被固定为1；如果CCR为0则OCxREF被固定为0.

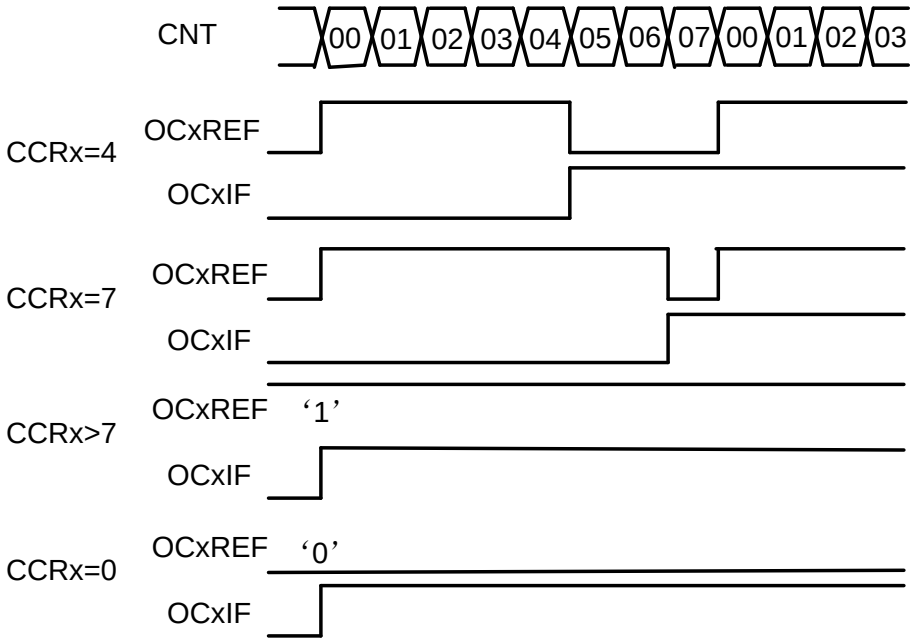


图27-28边沿对齐的PWM波形(ARR=7)

PWM中央对齐模式

OCxREF电平定义与边缘对齐模式相同。下图是一个示例：

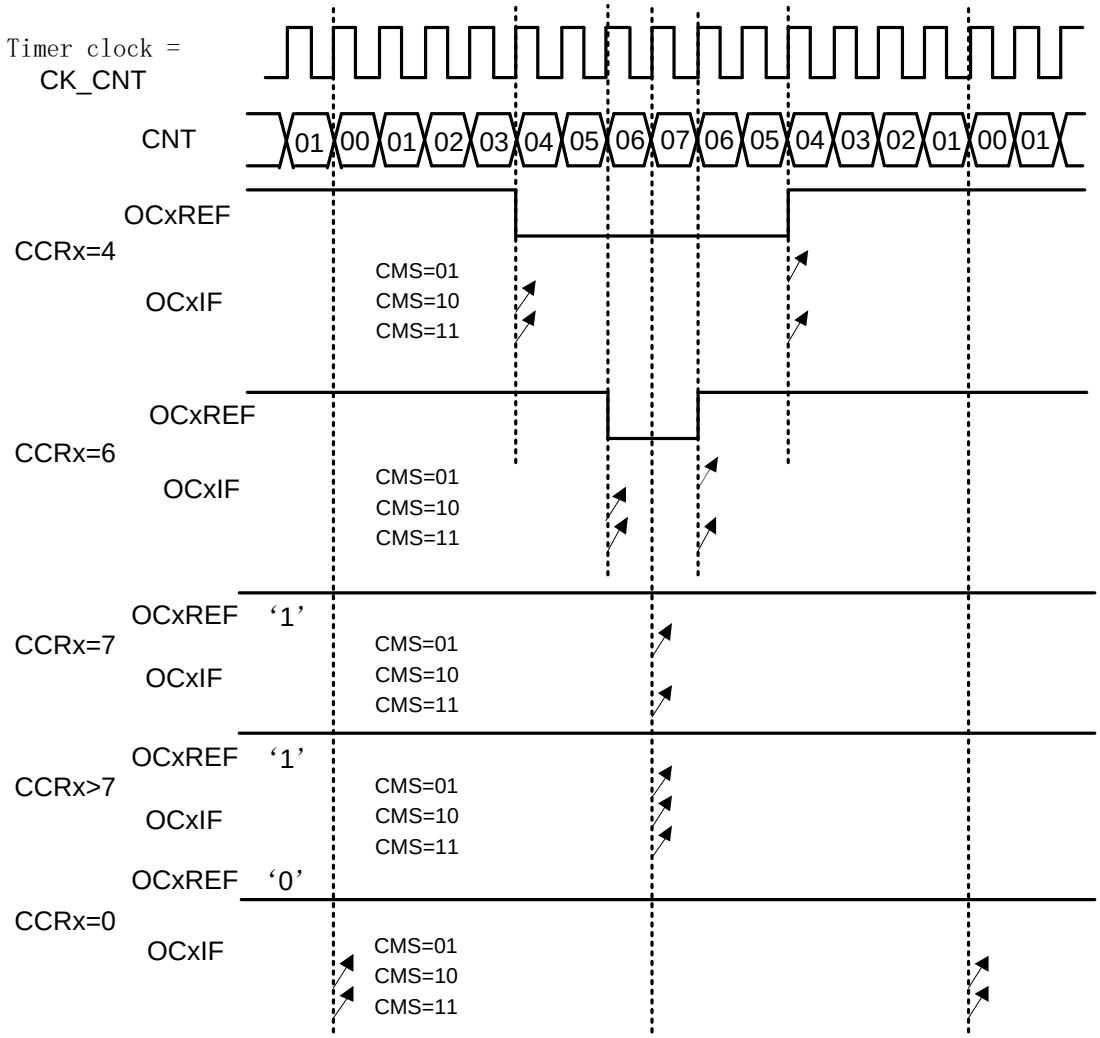


图27-29中央对齐的PWM波形(APR=7)

当启动中央对齐计数时，一开始的计数方向是由DIR寄存器决定的；随后在计数过程中，DIR寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过UG寄存器做一次update，并且在计数过程中不要改写计数器。

27.4.10 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次update event到来时，计数器会自动停止。只有当CCR和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求CNT<CCR<=ARR，在向下计数时，要求CNT>CCR

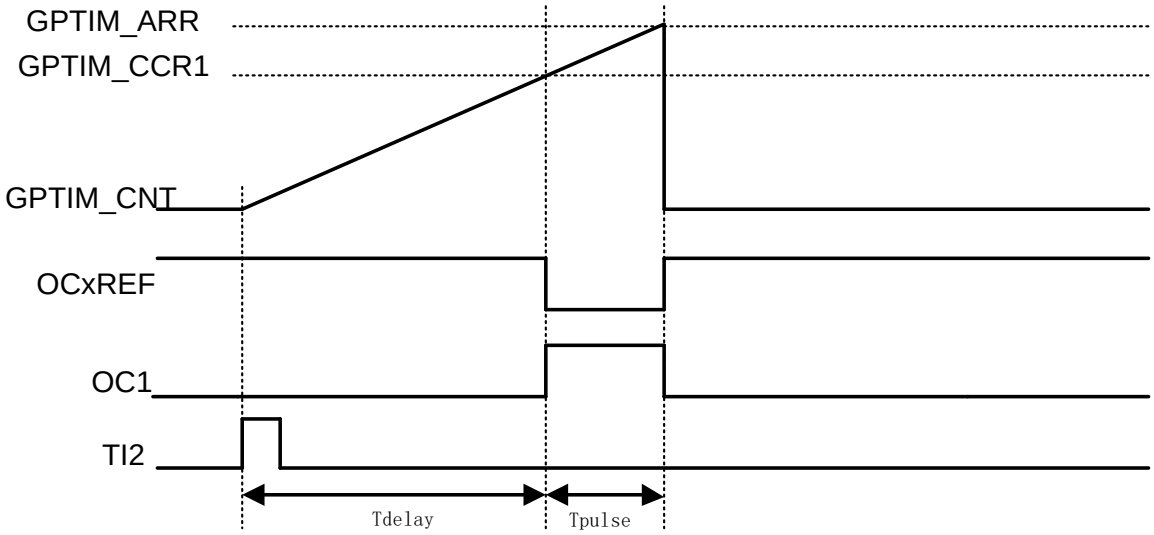


图27-30单脉冲模式的例子

上图是以TI2输入为计数器触发信号，计数值等于CCR后OCxREF输出低电平，计数到ARR后OCxREF回到高电平，并且计数器回滚到0，停止计数。

实现上述功能TI2作为输入触发的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置GPTIM_CCER.CC2E=1

实现上述功能OC1作为输出的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置GPTIM_CCMR1.CC1S=00
- 选择计数有效沿，配置GPTIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置GPTIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- GPTIM_CCR1的值决定了Tdelay
- GPTIM_ARR和GPTIM_CCR1的差值决定了Tpulse（GPTIM_ARR-GPTIM_CCR1）

- 设置为单脉冲模式，配置GPTIM_CR1.OPM=1

27.4.11 外部事件清除 OCxREF

OCxREF的有效状态未高电平，通过对外部ETR引脚施加高电平，可以直接拉低OCxREF，直到下一次update event。此功能仅在输出比较和PWM模式下有效，无法在软件force模式下起作用。使能此功能需要将OcxCE置1.

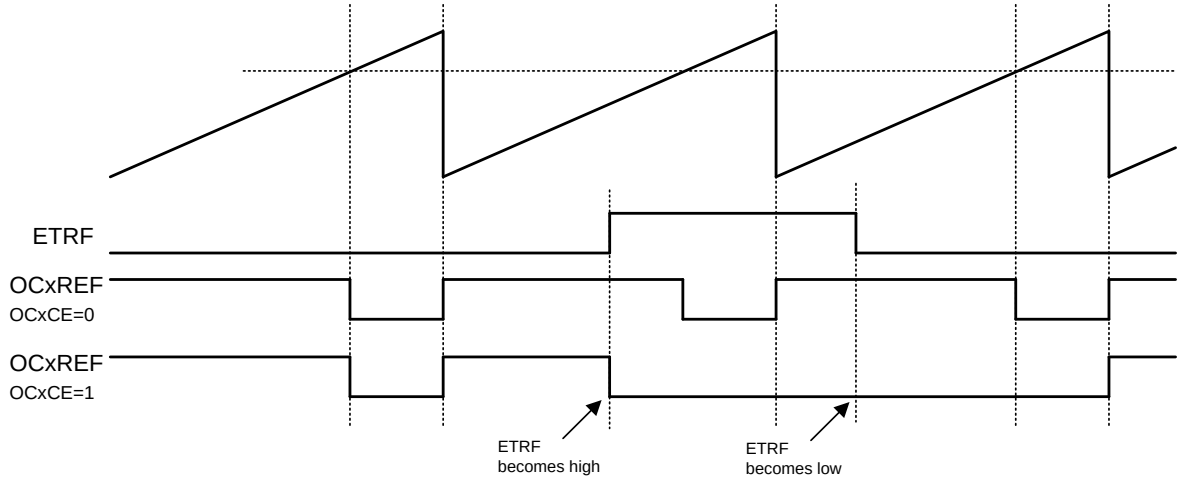


图27-31 ETR信号清除GPTIM的OCxREF

27.4.12 编码器接口模式（encoder interface）

编码器接口模式涉及到两个外部输入信号，GPTIM根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（TI1 对应TI2，TI2对应TI1）	TI1信号		TI2信号	
		上升	下降	上升	下降
仅在TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在TI1 和TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表27-2encoder interface计数方式

比如在计数器以TI1信号为时钟计数时，如果TI1上升沿采样到TI2为高电平，则计数器递减；如果TI1下降沿采样到TI2为高电平，则计数器递增。

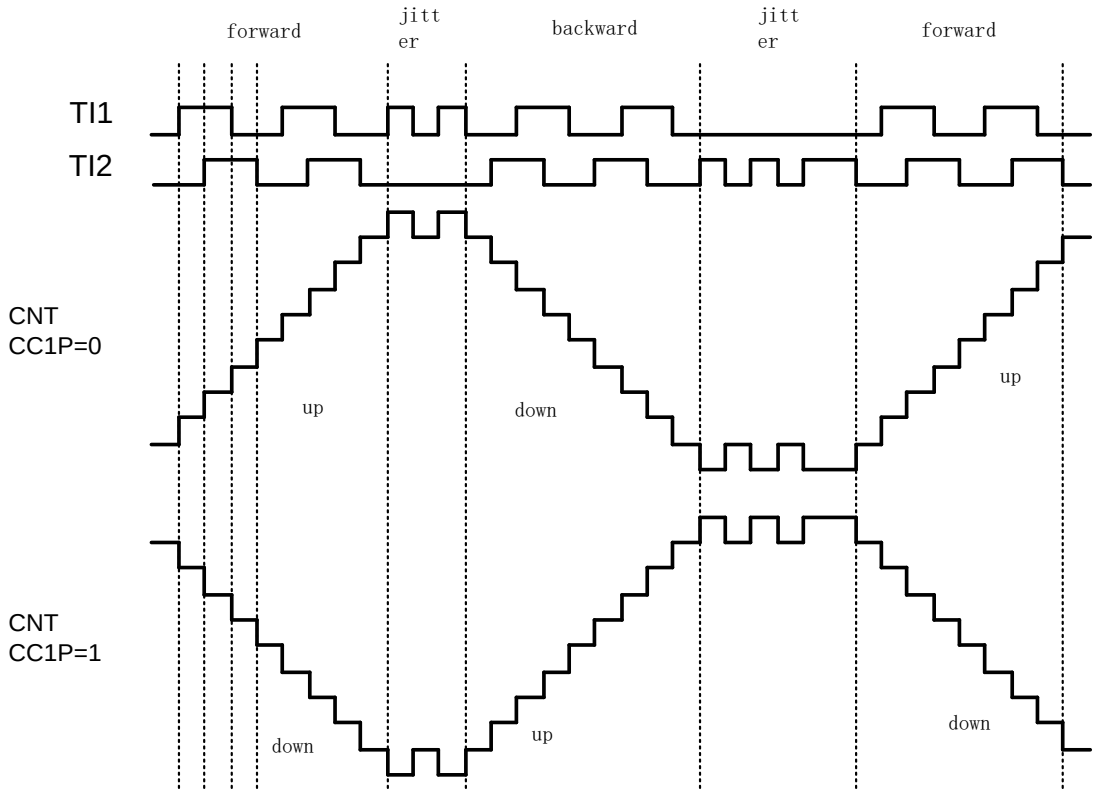


图27-32编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1，GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01，GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置GPTIM_SMCR.SMS[2:0]=011
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

27.4.13 GPTIM 从机模式

GPTIM作为slave时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致TIM内部所有preload寄存器重新初始化，CNT回到0开始计数。以下图为例，计数器正常计数，外部TI1输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

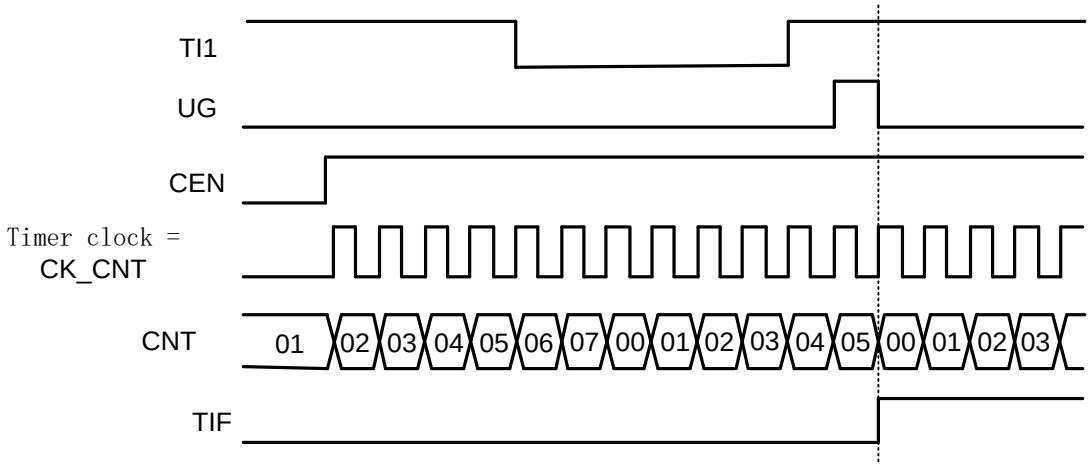


图27-33复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置GPTIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

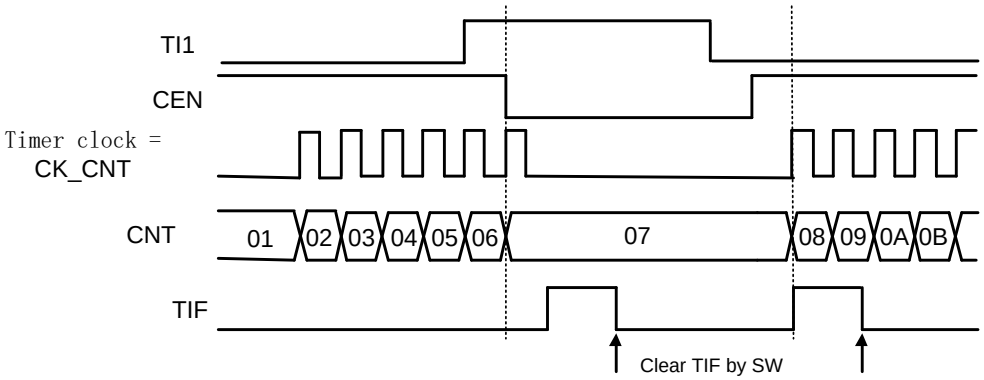


图27-34门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

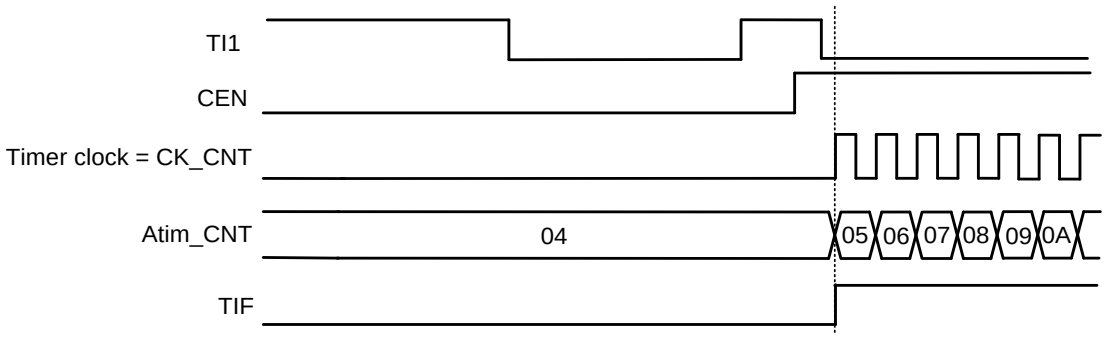


图27-35触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将ETR设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到TI1的上升沿之后，计数器开始以ETR输入的上升沿计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

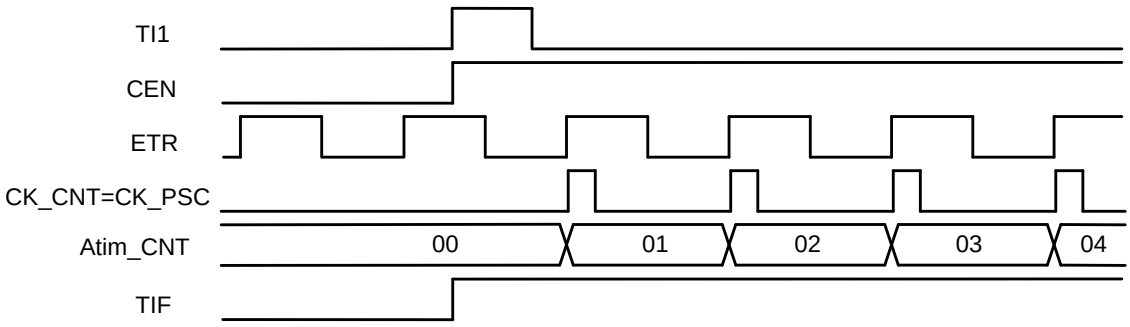


图27-36外部时钟模式2＋触发模式下的时序

27.4.14 DMA 访问

GPTIM支持6种DMA请求，分别为4个CC通道请求、外部触发请求和用户软件触发请求。

其中每个CC通道各自产生一个DMA请求，在捕捉模式下用于将CCRx中的内容传输给RAM，在比较模式下则用于将RAM中的数据写入CCRx；CC通道的DMA请求可以配置为单次传输或Burst传输（CCxBURSTEN），单次传输仅访问CCRx寄存器，Burst传输则根据DCR寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件和软件触发事件也可以产生DMA请求，当这两种请求发生时，会启动DMA Burst传输，向GPTIM内部1个或多个寄存器写入数据，或者从GPTIM读取1个或多个寄存器值。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
GTIMx_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	

表27-3DMA操作表

27.4.15 DMA Burst

DMA-Burst支持一个事件触发连续多次DMA请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA控制器需将外设目标地址指向一个虚拟寄存器GPTIM_DMAR。在特定的定时器事件发生时，GPTIM会连续发射多个DMA请求。每个DMA对GPTIM_DMAR的写操作都会被GPTIM重新定向到实际的功能寄存器上。

DBL寄存器用于设置DMA burst长度，DBA寄存器用于设置DMA访问GPTIM内部的基地址（相对于GPTIM_CR的offset）。

27.4.16 输入异或功能

通道1~3的输入信号可以被异或起来之后，接入到通道1的滤波和边沿电路输入，用于通道1的输入捕捉或者触发。

GPTIM_CR2寄存器的TI1S位用于选择通道1的输入是否来自于三个通道输入的异或。

27.4.17 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

27.5 寄存器

offset 地址	名称	符号
GPTIM0(模块起始地址:0x40013800)		
0x00000000	GPTIM0 控制寄存器 1 （GPTIM0 Control Register1）	GPTIM0_CR1
0x00000004	GPTIM0 控制寄存器 2 （GPTIM0 Control Register2）	GPTIM0_CR2
0x00000008	GPTIM0 从机模式控制寄存器 （GPTIM0 Slave Mode Control Register）	GPTIM0_SMCR
0x0000000C	GPTIM0 DMA 和中断使能寄存器 （GPTIM0 DMA and Interrupt Enable Register）	GPTIM0_DIER
0x00000010	GPTIM0 状态寄存器 （GPTIM0 Interrupt Status Register）	GPTIM0_ISR
0x00000014	GPTIM0 事件产生寄存器 （GPTIM0 Event Generation Register）	GPTIM0_EGR
0x00000018	GPTIM0 捕捉/比较模式寄存器 1 （GPTIM0 Capture/Compare Mode Register1）	GPTIM0_CCMR1
0x0000001C	GPTIM0 捕捉/比较模式寄存器 2 （GPTIM0 Capture/Compare Mode Register2）	GPTIM0_CCMR2
0x00000020	GPTIM0 捕捉/比较使能寄存器 （GPTIM0 Capture/Compare Enable Register）	GPTIM0_CCER
0x00000024	GPTIM0 计数器寄存器 （GPTIM0 Counter Register）	GPTIM0_CNT
0x00000028	GPTIM0 预分频寄存器 （GPTIM0 Prescaler Register）	GPTIM0_PSC
0x0000002C	GPTIM0 自动重载寄存器 （GPTIM0 Auto-Reload Register）	GPTIM0_ARR
0x00000034	GPTIM0 捕捉/比较寄存器 1 （GPTIM0 Capture/Compare Register1）	GPTIM0_CCR1
0x00000038	GPTIM0 捕捉/比较寄存器 2 （GPTIM0 Capture/Compare Register2）	GPTIM0_CCR2
0x0000003C	GPTIM0 捕捉/比较寄存器 3 （GPTIM0 Capture/Compare Register3）	GPTIM0_CCR3
0x00000040	GPTIM0 捕捉/比较寄存器 4 （GPTIM0 Capture/Compare Register4）	GPTIM0_CCR4
0x00000048	GPTIM0 DMA 控制寄存器 （GPTIM0 DMA Control Register）	GPTIM0_DCR
0x0000004C	GPTIM0 DMA 访问寄存器 （GPTIM0 DMA access Register）	GPTIM0_DMAR
0x00000060	GPTIM0 ITR 选择寄存器 （GPTIM0 Internal Trigger Select Register）	GPTIM0_ITRSEL
GPTIM1(寄存器起始地址:0x40013C00)		
0x00000000	GPTIM1 控制寄存器 1 （GPTIM1 Control Register1）	GPTIM1_CR1

offset 地址	名称	符号
0x00000004	GPTIM1 控制寄存器 2 （GPTIM1 Control Register2）	GPTIM1_CR2
0x00000008	GPTIM1 从机模式控制寄存器 （GPTIM1 Slave Mode Control Register）	GPTIM1_SMCR
0x0000000C	GPTIM1 DMA 和中断使能寄存器 （GPTIM1 DMA and Interrupt Enable Register）	GPTIM1_DIER
0x00000010	GPTIM1 状态寄存器 （GPTIM1 Interrupt Status Register）	GPTIM1_ISR
0x00000014	GPTIM1 事件产生寄存器 （GPTIM1 Event Generation Register）	GPTIM1_EGR
0x00000018	GPTIM1 捕捉/比较模式寄存器 1 （GPTIM1 Capture/Compare Mode Register1）	GPTIM1_CCMR1
0x0000001C	GPTIM1 捕捉/比较模式寄存器 2 （GPTIM1 Capture/Compare Mode Register2）	GPTIM1_CCMR2
0x00000020	GPTIM1 捕捉/比较使能寄存器 （GPTIM1 Capture/Compare Enable Register）	GPTIM1_CCER
0x00000024	GPTIM1 计数器寄存器 （GPTIM1 Counter Register）	GPTIM1_CNT
0x00000028	GPTIM1 预分频寄存器 （GPTIM1 Prescaler Register）	GPTIM1_PSC
0x0000002C	GPTIM1 自动重载寄存器 （GPTIM1 Auto-Reload Register）	GPTIM1_ARR
0x00000034	GPTIM1 捕捉/比较寄存器 1 （GPTIM1 Capture/Compare Register1）	GPTIM1_CCR1
0x00000038	GPTIM1 捕捉/比较寄存器 2 （GPTIM1 Capture/Compare Register2）	GPTIM1_CCR2
0x0000003C	GPTIM1 捕捉/比较寄存器 3 （GPTIM1 Capture/Compare Register3）	GPTIM1_CCR3
0x00000040	GPTIM1 捕捉/比较寄存器 4 （GPTIM1 Capture/Compare Register4）	GPTIM1_CCR4
0x00000048	GPTIM1 DMA 控制寄存器 （GPTIM1 DMA Control Register）	GPTIM1_DCR
0x0000004C	GPTIM1 DMA 访问寄存器 （GPTIM1 DMA access Register）	GPTIM1_DMAR
0x00000060	GPTIM1 ITR 选择寄存器 （GPTIM1 Internal Trigger Select Register）	GPTIM1_ITRSEL

27.5.1 GPTIMx 控制寄存器 1（GPTIMx_CR1）

名称	GPTIMx_CR1(x=0,1)							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CKD	
位权限	U-0						R/W-00	

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU，未实现，读为 0
9:8	CKD	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比）(Counter 503lock Divider) 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT 10：tDTS=4*tCK_INT 11：RFU，禁止使用
7	ARPE	Auto-reload 预装载使能(Auto-Reload Preload Enable) 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	CMS	计数器对齐模式选择(Counter Mode Selection) 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	计数方向寄存器(counter Direction) 0：向上计数 1：向下计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读
3	OPM	单脉冲输出模式(One Pulse Mode) 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择(Update Request Selection) 0：以下事件能够产生 update 中断 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断
1	UDIS	禁止 update(Update Disable) 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	计数器使能(Counter Enable) 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

27.5.2 GPTIMx 控制寄存器 2（GPTIMx_CR2）

名称	GPTIMx_CR2(x=0,1)							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	-		
位权限	R/W-0	R/W-000			R/W-0	U-0		

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	TI1S	通道 1 输入源选择(Timer Input 1 Selection) 0：GPTIMx_CH1 输入通道 1 1：GPTIMx_CH1, CH2, CH3 异或后输入通道 1
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源(Master Mode Selection) 000：GPTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	CCDS	捕捉/比较 DMA 选择(Capture/Compare DMA Selection) 0：捕捉/比较事件发生时发送 DMA 请求 1：Update Event 发生时发送 DMA 请求
2:0	-	RFU，未实现，读为 0

27.5.3 GPTIMx 从机模式控制寄存器（GPTIMx_SMCR）

名称	GPTIMx_SMCR(x=0,1)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
位权限	R/W-0	R/W-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	R/W-0	R/W-000			U-0	R/W-000		

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	**ETP**	外部触发信号极性配置(External Trigger Polarity) 0：高电平或上升沿有效 1：低电平或下降沿有效
14	**ECE**	外部时钟使能(External Clock Enable) 0：关闭外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	**ETPS**	外部触发信号预分频寄存器(External Trigger Prescaler) 外部触发信号 ETRP 的频率最多只能是 GPTIM 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8 分频
11:8	**ETF**	外部触发信号滤波时钟和长度选择(External Trigger Filter) 0000：无滤波 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	主/从模式(Master Slave Mode) 0：无动作 1：TRGI 触发的动作被延迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于单个外部事件对多个定时器进行同步的情况。
6:4	**TS**	触发选择，用于选择同步计数器的触发源(Trigger Source) 000：内部触发信号（ITR0） 001：内部触发信号（ITR1）

位号	助记符	功能描述
		010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2） 111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	-	RFU：未实现，读为 0
2:0	**SMS**	从机模式选择(Slave Mode Selection) 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

27.5.4 GPTIMx DMA 和中断使能寄存器（GPTIMx_DIER）

名称	GPTIMx_DIER(x=0,1)							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	TDE	-	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIE	-			CC2IE	CC1IE	UIE
位权限	U-0	R/W-0	U-0			R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 (CC4 Burst Enable) 0：Single 模式，仅访问 CCR

位号	助记符	功能描述
		1：Burst 模式，通过 DCR 配置访问的地址和长度
18	**CC3BURSTEN**	捕捉比较通道 3 的 DMA 模式配置 (CC3 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
17	**CC2BURSTEN**	捕捉比较通道 2 的 DMA 模式配置 (CC2 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
16	**CC1BURSTEN**	捕捉比较通道 1 的 DMA 模式配置 (CC1 Burst Enable) 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	**-**	RFU：未实现，读为 0
14	**TDE**	外部触发 DMA 请求使能(Triggered DMA Enable) 0：从机模式下，禁止外部触发事件产生 DMA 请求 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器）
13	**-**	RFU：未实现，读为 0
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 (CC4 DMA Enable) 0：禁止 CC4 DMA 请求 1：允许 CC4 DMA 请求
11	**CC3DE**	捕捉比较通道 3 的 DMA 请求使能 (CC3 DMA Enable) 0：禁止 CC3 DMA 请求 1：允许 CC3 DMA 请求
10	**CC2DE**	捕捉比较通道 2 的 DMA 请求使能 (CC2 DMA Enable) 0：禁止 CC2 DMA 请求 1：允许 CC2 DMA 请求
9	**CC1DE**	捕捉比较通道 1 的 DMA 请求使能 (CC1 DMA Enable) 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 (Update event DMA Enable) 0：Update Event 发生时，禁止产生 DMA 请求 1：Update Event 发生时，允许产生 DMA 请求
7	**-**	RFU：未实现，读为 0
6	**TIE**	触发事件中断使能 (Trigger event Interrupt Enable) 0：禁止触发事件中断 1：允许触发事件中断
5	**-**	RFU：未实现，读为 0
4	**CC3IE**	捕捉/比较通道 4 中断使能 (CC4 Interrupt Enable) 0：禁止捕捉/比较 4 中断 1：允许捕捉/比较 4 中断
3	**CC3IE**	捕捉/比较通道 3 中断使能 (CC3 Interrupt Enable) 0：禁止捕捉/比较 3 中断 1：允许捕捉/比较 3 中断
2	**CC2IE**	捕捉/比较通道 2 中断使能 (CC2 Interrupt Enable) 0：禁止捕捉/比较 2 中断 1：允许捕捉/比较 2 中断
1	**CC1IE**	捕捉/比较通道 1 中断使能 (CC1 Interrupt Enable) 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	Update 事件中断使能 (Update event Interrupt Enable)

位号	助记符	功能描述
		0：禁止 Update 事件中断 1：允许 Update 事件中断

27.5.5 GPTIMx 状态寄存器（GPTIMx_ISR）

名称	GPTIMx_ISR(x=0,1)							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIF	-	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	**-**	RFU：未实现，读为 0
12	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断(Over-Capture Interrupt Flag for CC4, write 1 to clear) 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断(Over-Capture Interrupt Flag for CC3, write 1 to clear) 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断(Over-Capture Interrupt Flag for CC2, write 1 to clear) 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断(Over-Capture Interrupt Flag for CC1, write 1 to clear) 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0：无 overcapture 事件 1：在 CC1IF 标志为 1 的情况下发生新的捕捉
8:7	**-**	RFU：未实现，读为 0
6	**TIF**	触发事件中断标志，硬件置位，软件写 1 清零 (Trigger event Interrupt Flag, write 1 to clear)
5	**-**	RFU：未实现，读为 0
4	**CC4IF**	捕捉/比较通道 4 中断标志 (CC4 Interrupt Flag, write 1 to clear) 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 (CC3 Interrupt Flag, write 1 to clear) 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 (CC2 Interrupt Flag, write 1 to clear) 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 (CC1 Interrupt Flag, write 1 to clear)

位号	助记符	功能描述
		如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果 CC1 通道配置为输入：发生捕捉事件时置位，软件写 1 清零，或者软件读 ATIM_CCR1 自动清零。
0	**UIF**	Update 事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to clear) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

27.5.6 GPTIMx 事件产生寄存器（GPTIMx_EGR）

名称	GPTIMx_EGR(x=0,1)							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TG	-			CC2G	CC1G	UG
位权限	U-0	W-0	U-0			W-0	W-0	W-0

位号	助记符	功能描述
31:7	**-**	RFU：未实现，读为 0
6	**TG**	软件触发，软件置位此寄存器产生触发事件，硬件自动清零 (Trigger Generate)
5:3	**-**	RFU：未实现，读为 0
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G (CC2 Generate)
1	**CC1G**	捕捉/比较通道 1 软件触发 (CC1 Generate) 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 (User Generate) 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

27.5.7 GPTIMx 捕捉/比较模式寄存器 1（GPTIMx_CCMR1）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	GPTIMx_CCMR1(x=0,1)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC		CC1S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE (OC2 Clear Enable)
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M (OC2 Mode)
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE (OC2 Preload Enable)
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE (OC2 Fast Enable)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 channel Selection) 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7	**OC1CE**	输出比较 1 清零使能(OC2 Clear Enable) 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M**	输出比较 1 模式配置,此寄存器定义 OC1REF 信号的行为 (OC1 Mode) 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低；在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC1REF 在 CNT<CCR1 时

位号	助记符	功能描述
		置低，否则置高；在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	**OC1PE**	输出比较 1 预装载使能 (OC1 Preload Enable) 0：CCR1 preload 寄存器无效，CCR1 可以直接写入 1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 (OC1 Fast Enable) 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC1S**	捕捉/比较 1 通道选择 (CC1 channel Selection) 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

输入捕捉模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:12	**IC2F**	输入捕捉 2 滤波 (IC2 Filter)
11:10	**IC2PSC**	输入捕捉 2 预分频 (IC2 Prescaler)
9:8	**CC2S**	捕捉/比较 2 通道选择 (Capture/Compare2 channel Selection) 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC3 映射到 TI2 10：CC2 通道配置为输入，IC3 映射到 TI1 11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	**IC1F**	输入捕捉 1 滤波 (IC1 Filter) 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	输入捕捉 1 预分频 (IC1 Prescaler)

位号	助记符	功能描述
		00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC1S**	捕捉/比较 1 通道选择 (Capture/Compare1 channel Selection) 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

27.5.8 GPTIMx 捕捉/比较模式寄存器 2（GPTIMx_CCMR2）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	GPTIMx_CCMR2(x=0,1)							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC2F				IC2PSC		CC4S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC		CC3S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15	**OC4CE**	输出比较 4 清零使能，参考 OC3CE (OC4 Clear Enable)
14:12	**OC4M**	输出比较 4 模式配置，参考 OC3M (OC4 Mode)
11	**OC4PE**	输出比较 4 预装载使能，参考 OC3PE (OC4 Preload Enable)
10	**OC4FE**	输出比较 4 快速使能，参考 OC3FE (OC4 Fast Enable)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7	**OC3CE**	输出比较 1 清零使能(OC3 Clear Enable) 0：OC1REF 不受 ETRF 影响

位号	助记符	功能描述
		1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC3M**	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 (OC3 Mode) 000：输出比较寄存器 CCR3 和计数器 CNT 的比较结果不会影响输出 001：CCR3=CNT 时，将 OC1REF 置高 010：CCR3=CNT 时，将 OC1REF 置低 011：CCR3=CNT 时，翻转 OC1REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低；在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC3REF 在 CNT<CCR3 时置低，否则置高；在向下计数时，OC3REF 在 CNT>CCR3 时置高，否则置低
3	**OC3PE**	输出比较 3 预装载使能 (OC3 Preload Enable) 0：CCR3 preload 寄存器无效，CCR3 可以直接写入 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC3FE**	输出比较 3 快速使能 (OC3 Fast Enable) 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC3S**	捕捉/比较 3 通道选择 (CC3 channel Selection) 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

输入捕捉模式

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:12	**IC4F**	输入捕捉 4 滤波 (IC4 Filter)
11:10	**IC4PSC**	输入捕捉 4 预分频 (IC4 Prescaler)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 3 滤波 (IC3 Filter) 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4

位号	助记符	功能描述
		0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频(IC3 Prescaler) 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC3S**	捕捉/比较 3 通道选择 (CC3 channel Selection) 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

27.5.9 GPTIMx 捕捉/比较使能寄存器（GPTIMx_CCER）

名称	GPTIMx_CCER(x=0,1)							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC4P	CC4E	-		CC3P	CC3E
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CC2P	CC2E	-		CC1P	CC1E
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0

位号	助记符	功能描述
31:14	**-**	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P (CC4 Polarity)
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E (CC4 output Enable)
11:10	**-**	RFU：未实现，读为 0

位号	助记符	功能描述
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P (CC3 Polarity)
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E (CC3 output Enable)
7:6	**-**	RFU：未实现，读为 0
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P (CC2 Polarity)
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E (CC2 output Enable)
3:2	**-**	RFU：未实现，读为 0
1	**CC1P**	捕捉/比较 1 输出极性 (CC1 Polarity) **CC1 通道配置为输出时：** 0：OC1 高有效 1：OC1 低有效 **CC1 通道配置为输入时：** CC1NP/CC1P 用于选择 TI1FP1 和 TI2FP1 的极性 00：非取反/上升沿 01：取反/下降沿 10：保留，不要使用 11：非取反，上下沿都有效
0	**CC1E**	捕捉/比较 1 输出使能 (CC1 output Enable) **CC1 通道配置为输出时** 0：OC1 输出关闭，Ocx=0，Ocx_EN=0 1：Ocx=OCxREF+极性选择，Ocx_EN=1 **CC1 通道配置为输入时** 0：关闭捕捉功能 1：使能捕捉功能

标准 Ocx 通道的输出控制位

CcxE 位	Ocx 输出状态
0	禁止输出（Ocx=0，Ocx_EN=0）
1	Ocx=OCxREF + 极性，Ocx_EN=1

27.5.10 GPTIMx 计数器寄存器（GPTIMx_CNT）

名称	GPTIMx_CNT(x=0,1)							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	CNT	计数器值(Counter)

27.5.11 GPTIMx 预分频寄存器（GPTIMx_PSC）

名称	GPTIMx_PSC(x=0,1)							
offset	0x00000028 + x*0x400							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	PSC	计数器时钟（CK_CNT）预分频值(Counter Clock Prescaler) $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

27.5.12 GPTIMx 自动重载寄存器（GPTIMx_ARR）

名称	GPTIMx_ARR(x=0,1)							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	ARR	计数溢出时的自动重载值(Auto-Reload Register)

位号	助记符	功能描述
		这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

27.5.13 GPTIMx 捕捉/比较寄存器 1（GPTIMx_CCR1）

名称	GPTIMx_CCR1(x=0,1)							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:0	**CCR1**	捕捉/比较通道 1 寄存器(Capture/Compare channel 1 Register) **如果通道 1 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC1 输出 **如果通道 1 配置为输入：** CCR1 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR1 为只读

27.5.14 GPTIMx 捕捉/比较寄存器 2（GPTIMx_CCR2）

名称	GPTIMx_CCR2(x=0,1)							
Offset	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**CCR2**	捕捉/比较通道 2 寄存器(Capture/Compare channel 2 Register) **如果通道 2 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 **如果通道 2 配置为输入：** CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

27.5.15 GPTIMx 捕捉/比较寄存器 3（GPTIMx_CCR3）

名称	GPTIMx_CCR3(x=0,1)							
Offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**CCR3**	捕捉/比较通道 3 寄存器(Capture/Compare channel 3 Register) **如果通道 3 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

27.5.16 GPTIMx 捕捉/比较寄存器 4（GPTIMx_CCR4）

名称	GPTIMx_CCR4(x=0,1)							
Offset	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	CCR4[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**CCR4**	捕捉/比较通道 4 寄存器(Capture/Compare channel 4 Register) **如果通道 4 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC4 输出 **如果通道 4 配置为输入：** CCR4 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR4 为只读

27.5.17 GPTIMx DMA 控制寄存器（GPTIMx_DCR）

名称	**GPTIMx_DCR(x=0,1)**							
Offset	0x00000048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			R/W-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 (DMA Burst Length) 对 GPTIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~18 00000：长度=1 00001：长度=2 00010：长度=3 00011：长度=4 00100：长度=5 00101：长度=6 00110：长度=7 00111：长度=8 01000：长度=9 01001：长度=10

位号	助记符	功能描述
		01010：长度=11 01011：长度=12 01100：长度=13 01101：长度=14 01110：长度=15 01111：长度=16 10000：长度=17 10001：长度=18 其他：无效值，禁止写入
7:5	-	RFU：未实现，读为 0
4:0	**DBA**	DMA 基地址，定义指向寄存器的偏移地址 (DMA Burst offset Address) 00000：GPTIM_CR1 00001：GPTIM_CR2 00010：GPTIM_SMCR …… 注意：当 DBA+DBL 超出了 GPTIM 寄存器地址范围，则实际 burst 传输到 GPTIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

27.5.18 GPTIMx DMA 访问寄存器（GPTIMx_DMAR）

名称	GPTIMx_DMAR(x=0,1)							
Offset	0x0000004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**DMAR**	DMA burst 访问寄存器 在使用 DMA burst 传输时，将 DMA 通道外设地址设置为 GPTIM_DMAR，GPTIM 会根据 DBL 的值产生多次 DMA 请求

27.5.19 GPTIMx ITR 选择寄存器（GPTIMx_ITRSEL）

名称	GPTIMx_ITRSEL(x=0,1)

Offset	0x00000060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ITR3SEL		ITR2SEL		ITR1SEL		ITR0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7:6	ITR3SEL	ITR 输入信号选择(Internal Trigger Source Selection) 内部触发信号（ITRx）的捕捉 详情参见 27.4.4 内部触发信号（ITRx）的捕捉
5:4	ITR2SEL	
3:2	ITR1SEL	
1:0	ITR0SEL	

28 基本定时器（BSTIM32）

28.1 概述

FM33LC0XX包含1个基本定时器。

基本定时器包含一个32bit自动重载计数器及一个可编程预分频器。

基本定时器主要用来产生系统时基，也可以产生触发事件来驱动ADC采样。

28.2 主要特性

- 32bit向上计数自动重载计数器
- 32bit可编程预分频器，支持实时调整计数时钟分频
- ADC定时触发功能
- 计数器溢出时产生中断

28.3 结构框图

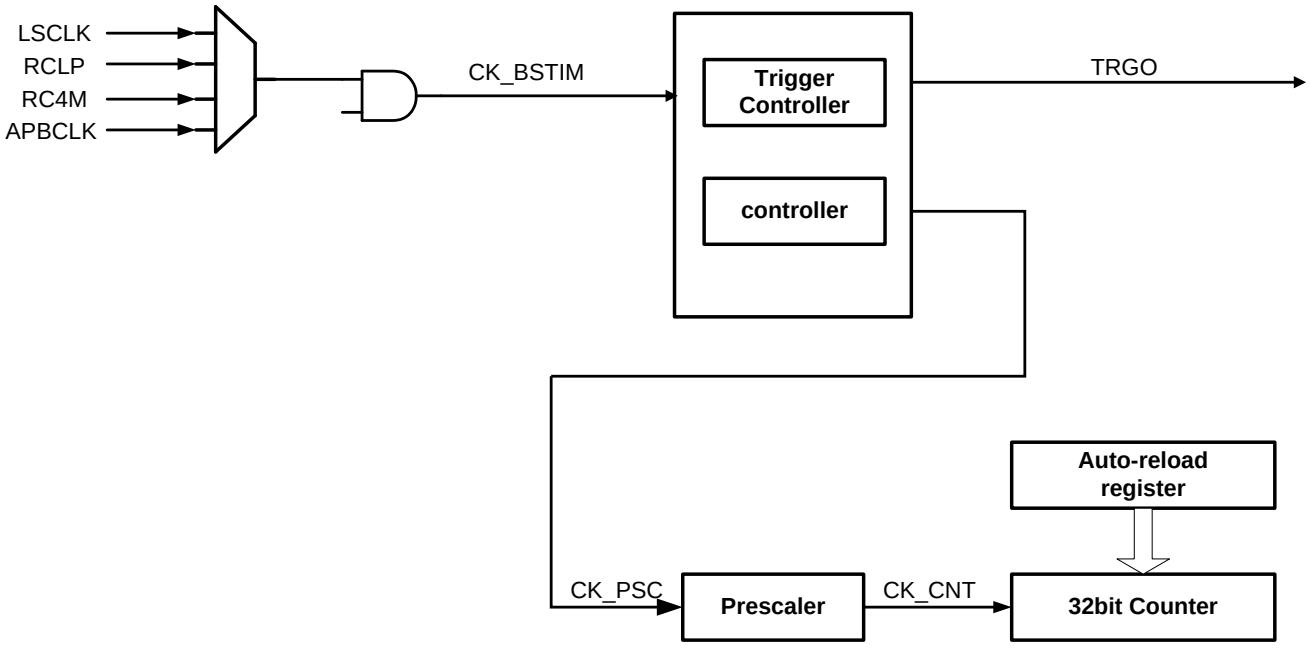


图28-1BSTIM32结构框图

28.4 功能描述

28.4.1 定时单元

基本定时器的定时单元由一个32位计数器和自动重载寄存器组成。计数器向上计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（BSTIM_CNT）
- 预分频寄存器（BSTIM_PSC）
- 自动重载寄存器（BSTIM_ARR）

ARR包含preload功能，软件读写ARR可以直接起效，或者只是访问其缓存，通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=1时，软件读写ARR都是访问其缓存寄存器，当update event（ATIM_CNT上溢出或者下溢出）发生时，会将缓存寄存器内的数据更新到ARR中。软件也可以通过寄存器操作主动触发ARR更新。

BSTIM_CNT工作时钟由BSTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

BSTIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际是改写缓存寄存器，只有当新的update event到来时，才会从缓存寄存器更新PSC。因此在CNT计数过程中，软件可以实时改写PSC。

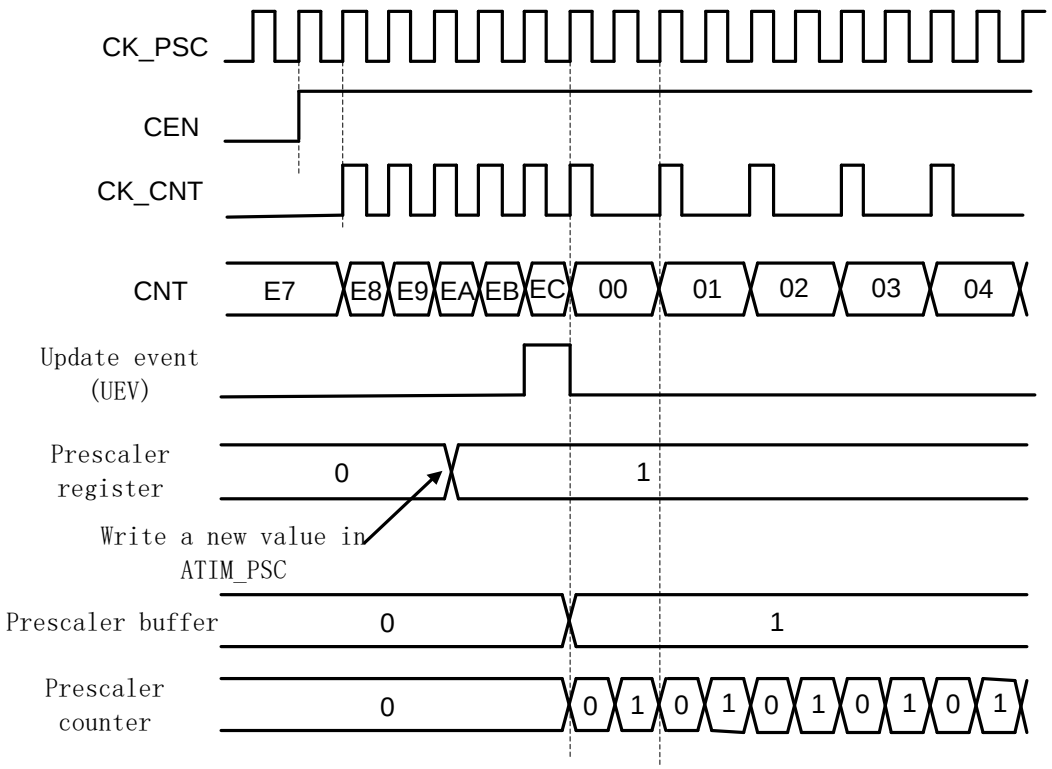


图 28-2 预分频从 1 变为 2 的波形

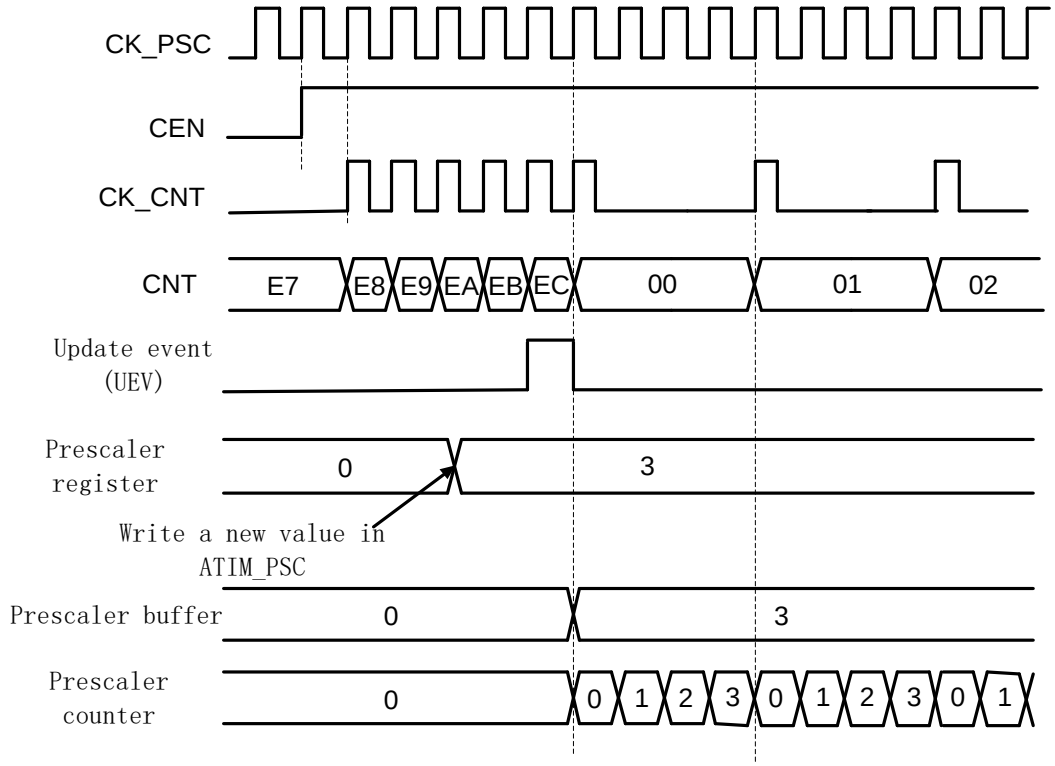


图 28-3 预分频从 1 变为 4 的波形

28.4.2 定时器工作模式

通用定时器只支持向上计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器不会触发UIF（Update Interrupt Flag）中断标志置位。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- BSTIM_RCR更新为缓存中的值
- BSTIM_ARR更新为缓存中的值
- BSTIM_PSC更新为缓存中的值

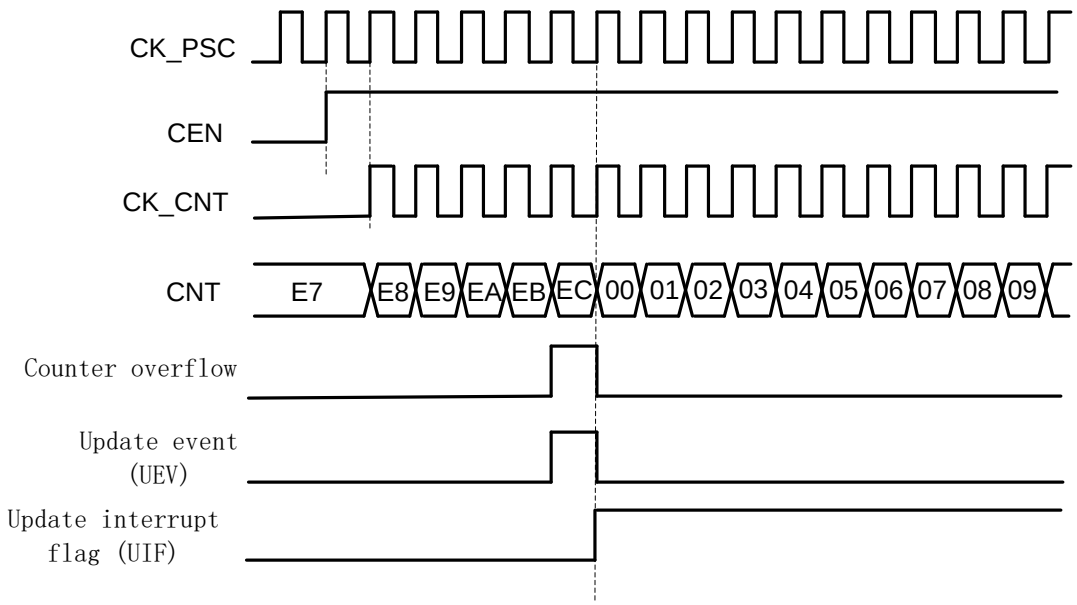


图 28-4 向上计数波形，内部时钟不分频

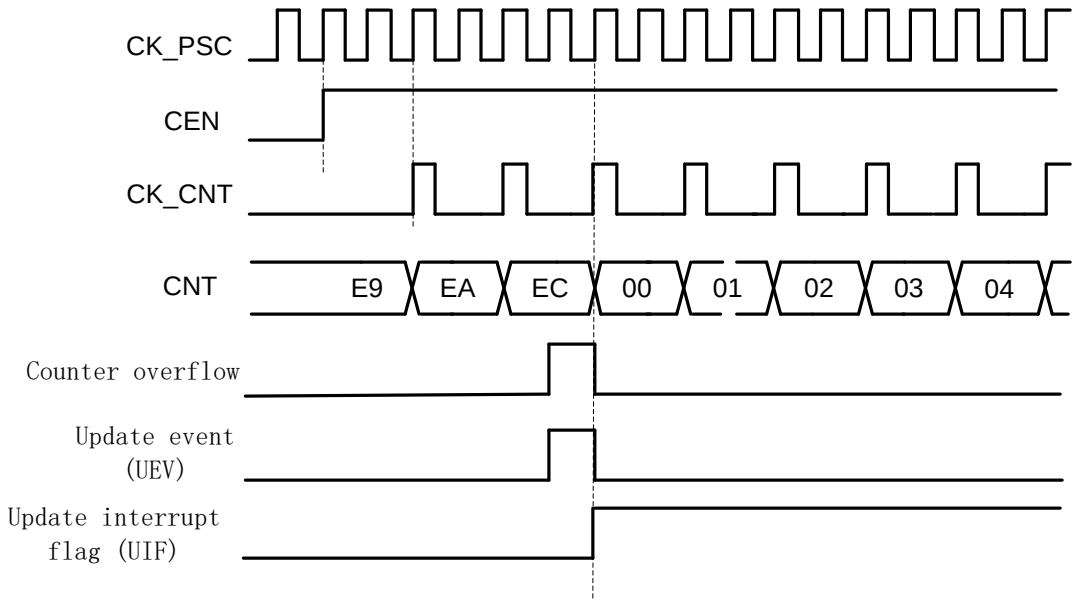


图 28-5 向上计数波形，内部时钟 2 分频

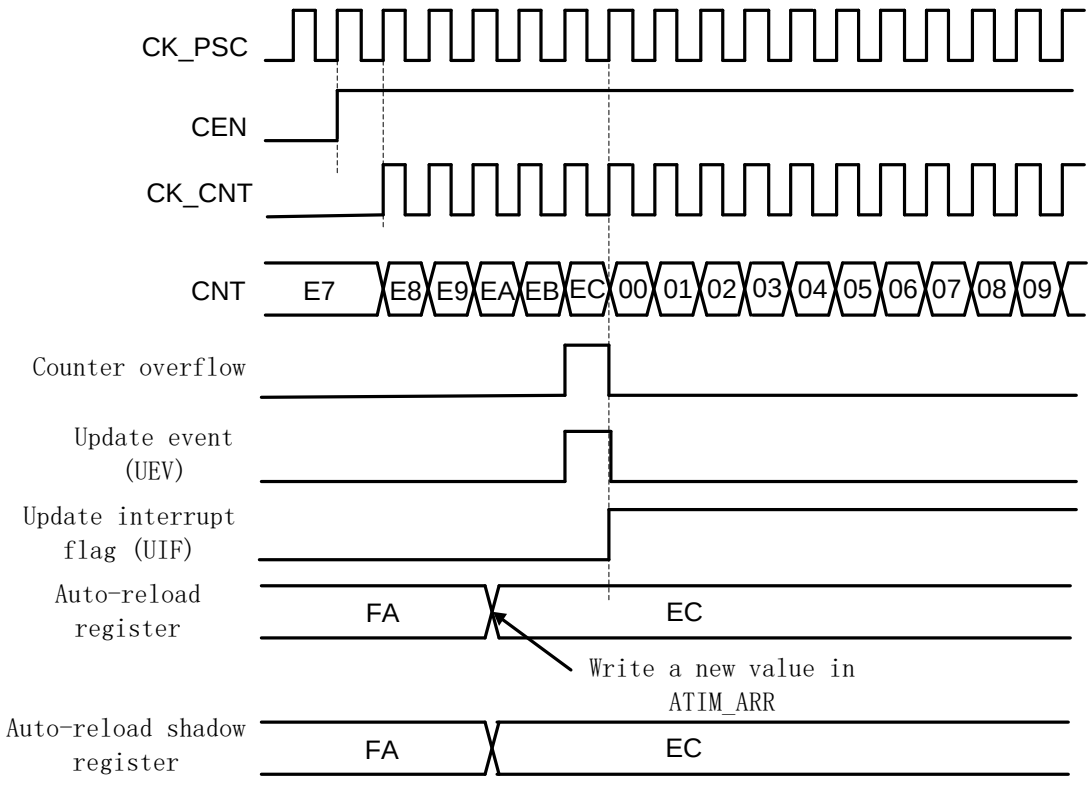


图 28-6ARPE=0（ARR 没有预装载）时的更新事件

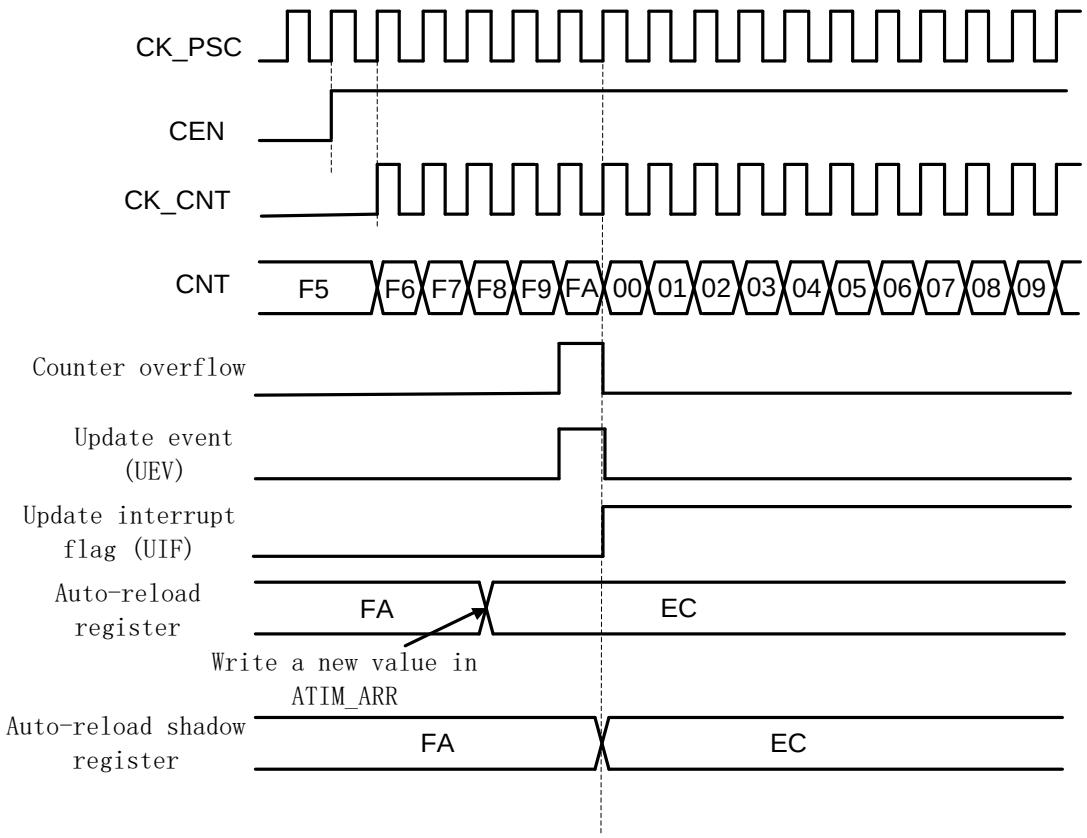


图 28-7ARPE=1（ARR 预装载）时的更新事件

28.4.3 计数器工作时钟

BSTIM使用内部时钟工作，CEN、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

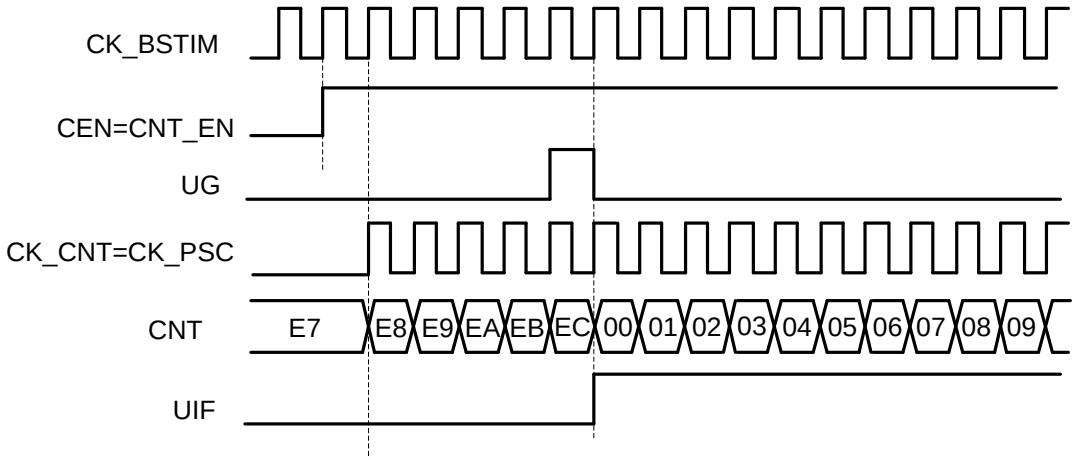


图 28-8 内部时钟源模式，时钟分频因子为 1

28.4.1 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DCU模块的DBG_TIMx_STOP寄存器定义。

28.5 寄存器

offset 地址	名称	符号
BSTIM32(模块起始地址:0x4001B400)		
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM_ARR

28.5.1 BSTIM 控制寄存器 1（BSTIM_CR1）

名称	BSTIM_CR1							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	-			OPM	URS	UDIS	CEN
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	ARPE	Auto-reload 预装载使能 (Auto-Reload Preload Enable) 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:4	-	RFU，未实现，读为 0
3	OPM	单脉冲输出模式 (One Pulse Mode) 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择 (Update Request Select) 0：以下事件能够产生 update 中断

位号	助记符	功能描述
		- 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	UDIS	禁止 update (Update Disable) 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	计数器使能 (Counter Enable) 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

28.5.2 BSTIM 控制寄存器 2（BSTIM_CR2）

名称	BSTIM_CR2							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

位号	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源(Master Mode Select) 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011/100/111：RFU 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3:0	-	RFU，未实现，读为 0

28.5.3 BSTIM 中断使能寄存器（BSTIM_IER）

名称	BSTIM_IER							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIE	Update 事件中断使能(Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

28.5.4 BSTIM 中断标志寄存器（BSTIM_ISR）

名称	BSTIM_ISR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。(Update event Interrupt Flag,write 1 to flag) 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器

位号	助记符	功能描述
		-URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

28.5.5 BSTIM 事件产生寄存器（BSTIM_EGR）

名称	BSTIM_EGR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UG
位权限	U-0							W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	UG	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零(User Generate) 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

28.5.6 BSTIM 计数器寄存器（BSTIM_CNT）

名称	BSTIM_CNT							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	CNT	计数器值(Counter)

28.5.7 BSTIM 预分频寄存器（BSTIM_PSC）

名称	BSTIM_PSC							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PSC[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PSC[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**PSC**	计数器时钟（CK_CNT）预分频值(Counter Clock Prescaler) $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

28.5.8 BSTIM 自动重载寄存器（BSTIM_ARR）

名称	BSTIM_ARR							
Offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**ARR**	计数溢出时的自动重载值(Auto-Reload Register) 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

29 低功耗定时器（LPTIM32）

29.1 概述

LPTIM32是32bits低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM32在在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM32甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。LPTIM32的主要特性有：

- 1 个独立的 32bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK、LPOSC、APBCLK、RCMF、ADC 转换结束信号
  - 外部时钟源：LPT_ETR（带有模拟滤波）
- 双通道 32bit 捕捉/比较寄存器
- 32bit 自动重载寄存器
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 32bit PWM 输出
- 32bit 输入信号捕捉，支持 DMA

29.2 结构框图

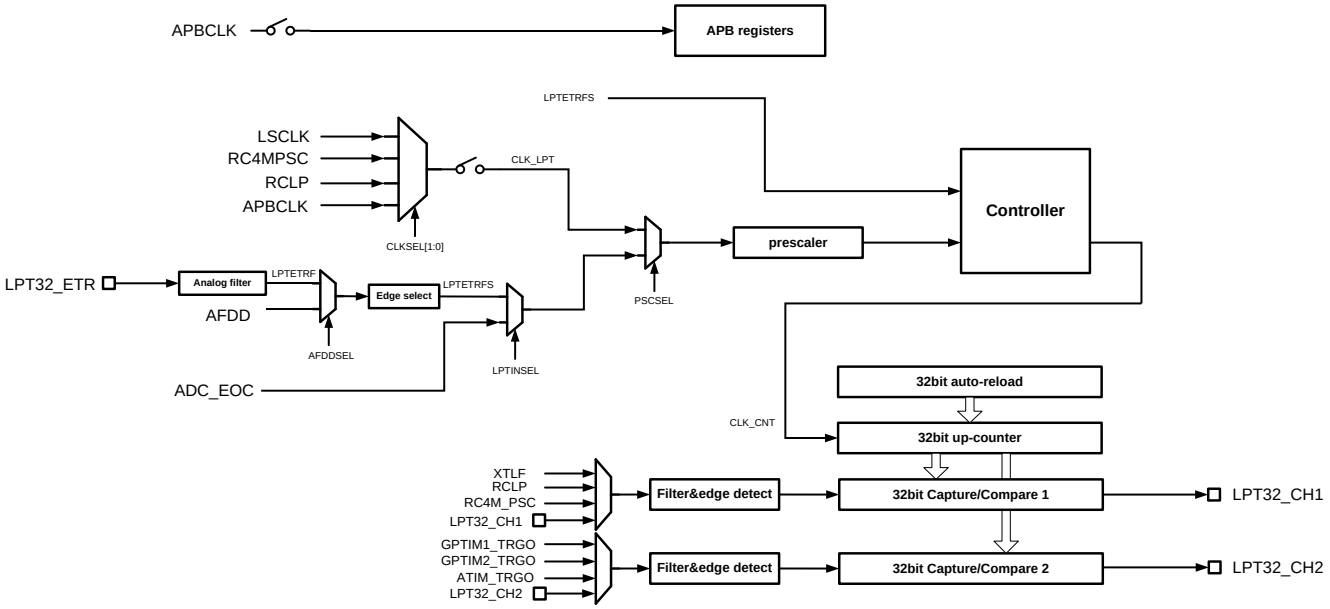


图 29-1LPTIM32 结构框图

29.3 定时器功能

LPTIM32支持4种定时器工作模式：普通定时器、外部脉冲触发计数、外部异步脉冲计数、Timeout模式。

29.3.1 普通定时器

当LPTCFG.TMODE=00时，LPTIM32为普通定时器工作模式

- 使用多路选择后的CLK_LPT时钟计数
- 需要配置CMU模块中的OPCCON2.LPTCKS寄存器，选择合适的计数时钟
- LPTEN使能置位后有两个计数时钟的同步过程
- 使能后定时器即开始向上计数，直到计数值等于LPTARR

单次计数和连续计数

LPTIM32有两种计数模式——单次计数和连续计数。

连续计数模式：计数器启动后保持运行，直到被关闭为止。计数器达到目标值（LPTARR）后回到0重新开始计数，并产生溢出中断。

单次计数模式：计数器被触发后计数到目标值（LPTARR）后回到0，并自动停止，产生溢出中断，同时硬件自动清除LPTEN。

29.3.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTCFG.TMODE=01）下，LPTIM32将LPT32_ETR引脚输入的信号作为触发信号使用。LPT32_ETR信号首先经过LPTIM32工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。由于需要使用CLK_LPT采样并识别LPT32_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM32对LPT32_ETR的哪个边沿计数。

下图举例说明了LPT32_ETR触发计数，上升沿有效的情况。

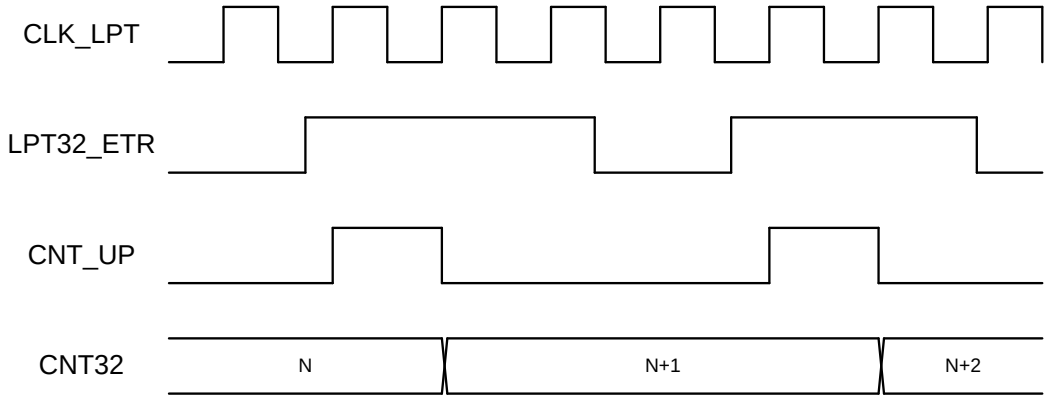


图 29-2 外部 ETR 脉冲上升沿触发计数

29.3.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTCFG.TMODE=10）下，LPTIM32可以将LPT32_ETR引脚输入的信号直接作为计数时钟使用（要配置PSCSEL=1并且LPTINSEL=0）。这种情况下，LPTIM32全异步工作，不需要使能任何内部时钟。软件可以通过LPTCFG.EDGESEL来选择定时器使用ETR上升沿还是下降沿计数。由于这种模式下LPT32_ETR引脚上的任何干扰信号都有可能引起定时器误动作，因此推荐使能ETR输入模拟滤波功能，能够滤除大约100ns以内的glitch信号。

下图举例说明了外部异步脉冲计数，下降沿有效的情况。

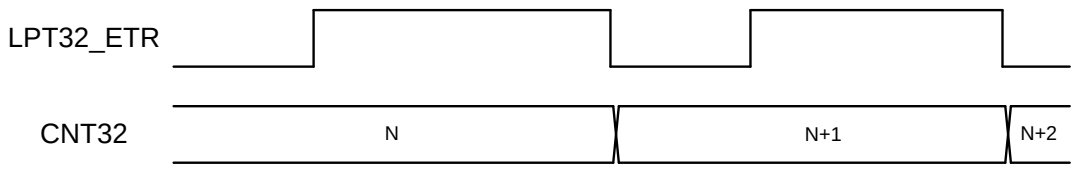


图 29-3 外部 ETR 脉冲异步计数

29.3.4 Timeout 模式

Timeout模式（LPTCFG.TMODE=11）下，LPTIM32将LPT32_ETR引脚输入的信号作为触发信号使用，定时器使用内部时钟CLK_LPT工作。Timeout模式下定时器启动后，不会立即开始计数，而是等待第一个LPT32_ETR信号的有效沿到来。当第一个有效沿到来后，触发定时器开始自由计数，

此后每个新的ETR有效沿都会清零计数器，并重新开始计数。根据外部输入ETR信号的实际频率，合理配置计数器工作时钟和溢出上限（LPTARR），可以保持定时器不会溢出。如果定时器出现溢出，则表示在规定时间间隔内没有预期的ETR事件到来，则定时器产生溢出中断，计数值回到0，并自动清除LPTEN结束计数过程。

LPT32_ETR信号首先经过LPTIM32工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发计数器清零重新计数。由于需要使用CLK_LPT采样并识别LPT32_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM32对LPT32_ETR的哪个边沿计数。

下图是timeout模式下使用LPT32_ETR上升沿清零，并最终溢出的例子。

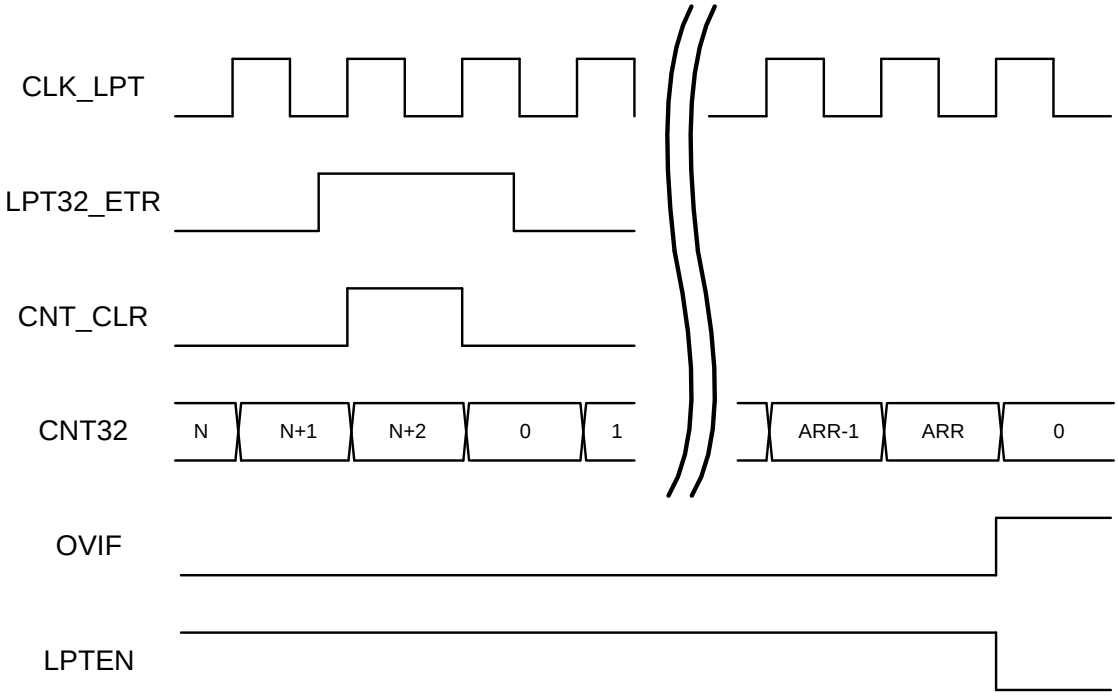


图 29-4TimeOut 模式

使用TimeOut模式，并使能LPTIM中断，在芯片休眠时可以实现外部信号触发的超时唤醒功能。此时只要LPT32_ETR管脚上有周期性信号输入，就能使芯片保持休眠，而一旦超过规定时间内没有新的触发信号到来，LPTIM超时溢出中断将唤醒芯片。

29.4 捕捉比较功能

LPTIM32带有两个独立的32bit捕捉比较通道，以32bit定时器为时基，结合CCRx寄存器，可以实现两路32bit PWM输出，或32bit输入捕捉功能。

29.4.1 32bit PWM

LPTIM32的两个独立捕捉/比较通道都可以输出32bit PWM波形。PWM功能需要将捕捉/比较通道配置为比较输出。

使能PWM功能后LPTIM32从0x0000_0000开始计数，计数值等于比较值（CCRx）时输出置高，计数值等于目标值寄存器（LPTARR）时输出变低；PWM周期由ARR寄存器决定，占空比由CCRx寄存器决定。LPTCFG.POLARITY寄存器可以配置输出波形的极性。

实现PWM输出功能，需要将LPTCFG.CCxS配置为10，此时LPT_CHx成为输出通道，相应的GPIO自动使能输出功能（软件需将GPIO配置为数字外设功能）。

下图是PWM输出，POLARITY=1的例子。

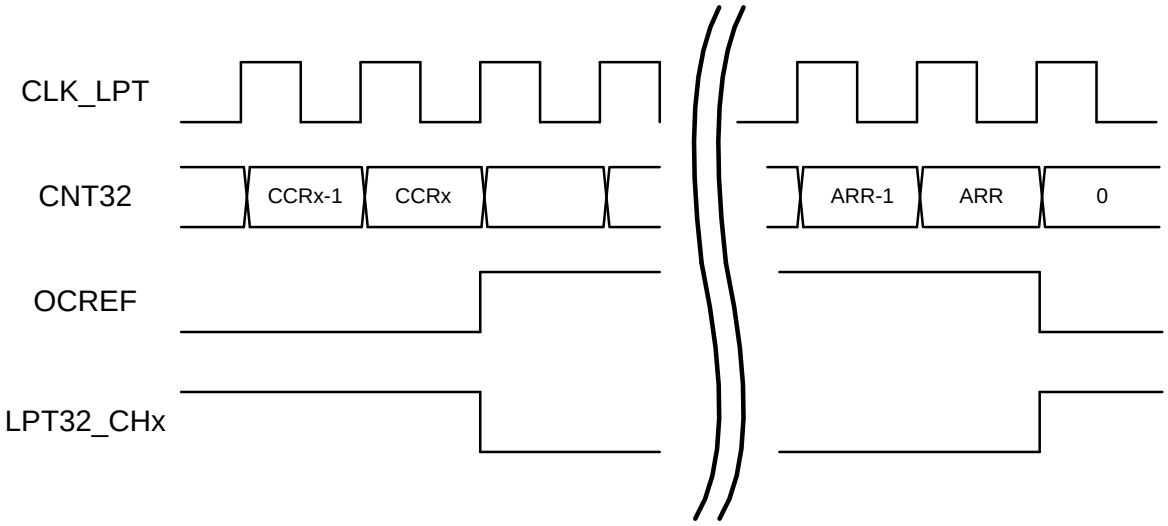


图 29-5PWM 输出

29.4.2 输入捕捉

LPTIM32的两个捕捉/比较通道可以实现两路独立的输入信号周期或电平宽度捕捉功能。输入信号捕捉功能可以配合DMA使用，实现多次连续捕捉结果的自动搬运。

输入捕捉可以配置为针对输入信号的上升沿、下降沿或上升下降沿进行捕捉。每次捕捉发生时，CAPxEDGE寄存器会指示当前捕捉到的是上升沿还是下降沿。

LPTIM32的通道1可以对外部引脚输入或者芯片内部时钟信号（XTLF、LPOSC、RCMF_PSC）进行捕捉，对内部时钟信号的周期捕捉可以用于软件配合的时钟频率校准；而通道2只能对外部引脚输入信号进行捕捉。

使能输入模式后，32bit计数器作为时基自由计数，当被捕捉信号的有效边沿到来后，当前计数值被锁存入CCRx寄存器，并产生捕捉中断；如果使能了DMA功能，CCRx还会同时被DMA读取并写入RAM指定地址。软件或DMA读取CCRx寄存器时，硬件都会自动清除捕捉中断，此外捕捉中断也可以由软件写1清零。当捕捉中断未被清除时，又有新的捕捉事件到来，会置位捕捉冲突中断标志

（CAPxOVR）。

下图是对输入信号上升下降沿进行捕捉的例子。

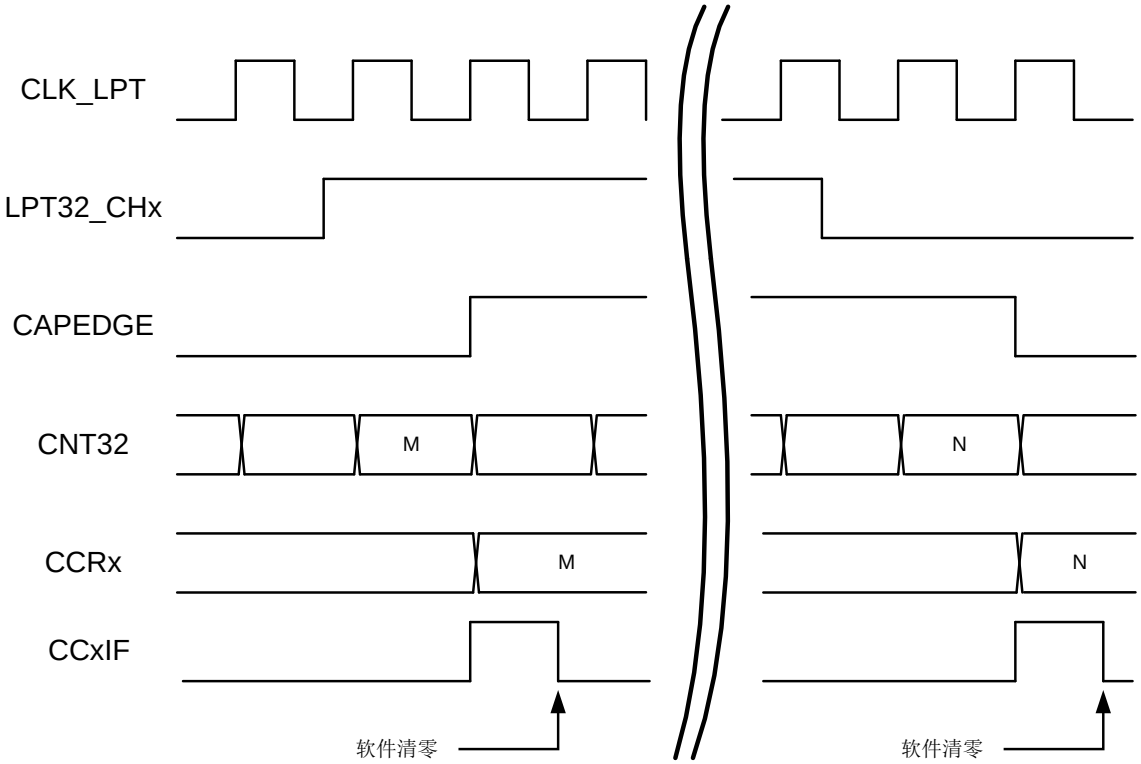


图 29-6PWM 输出

29.5 寄存器

offset 地址	名称	符号
LPTIM(模块起始地址:0x40013400)		
0x00000000	LPTIM 配置寄存器 （LPTIM Config Register）	LPTIM_CFGR
0x00000004	LPTIM 计数值寄存器 （LPTIM Counter Register）	LPTIM_CNT
0x00000008	LPTIM 捕捉比较控制和状态寄存器 （LPTIM Capture/Compare Control and Status Register）	LPTIM_CCSR
0x0000000C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM_ARR
0x00000010	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM_IER
0x00000014	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM_ISR
0x00000018	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM_CR
0x00000020	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM_CCR1
0x00000024	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM_CCR2

29.5.1 LPTIM 配置寄存器（LPTIM_CFGR）

名称	LPTIM_CFGR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	PSCSEL	LPTINSEL	DIVSEL			-	
位权限	U-0	R/W-0	R/W-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		-		ONST	TMOD	
位权限	R/W-0	R/W-00		U-0		R/W-0	R/W-00	

位号	助记符	功能描述
31:25	-	**未实现：读为0**
24	**ETR_AFEN**	ETR 输入模拟滤波使能 (External Trigger input Analog Filter Enable) 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns

位号	助记符	功能描述
23:15	-	**未实现：读为0**
14	**PSCSEL**	时钟预分频输入选择 (Prescaler input Select) 0：CLKSEL 选择的时钟 1：LPTINSEL 选择的信号
13	**LPTINSEL**	ETR 输入源选择 (External Trigger input source Select) 0：引脚输入 1：ADC_EOC
12:10	**DIVSEL**	计数时钟分频选择 (Counter Clock Divider Select) 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
9:8	-	**未实现：读为0**
7	**EDGESEL**	ETR 输入边沿选择 (ETR Clock Edge Select) 0：LPT_ETR 的上升沿计数 1：LPT_ETR 的下降沿计数
6:5	**TRIGCFG**	外部触发边沿选择（需使用内部时钟同步采样 LPT_ETR）(ETR trigger Configuration) 00：LPT_ETR 输入信号上升沿触发 01：LPT_ETR 输入信号下降沿触发 10/11：外部输入信号上升下降沿触发
4:3	-	未实现：读为0
2	**ONST**	单次计数模式使能 (One State Timerenable) 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。
1:0	**TMOD**	工作模式选择 (Timer operation Mode) 00：普通定时器模式 01：Trigger 脉冲触发计数模式 10：外部异步脉冲计数模式 11：Timeout 模式

29.5.2 LPTIM 计数值寄存器（LPTIM_CNT）

名称	LPTIM_CNT							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT32[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT32[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	CNT32[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT32[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	**CNT32**	32bit 计数器当前计数值(Counter 32bits-wide,read only)

29.5.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM_CCSR）

名称	**LPTIM_CCSR**							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CAP2SSEL		CAP1SSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CAP2EDGE	CAP1EDGE	CAPCFG2		CAPCFG1	
位权限	U-0		R-0	R-0	R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		POLAR2	POLAR1	CC2S		CC1S	
位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	

位号	助记符	功能描述
31:20	**-**	**未实现：读为0**
19:18	**CAP2SSEL**	通道 2 捕捉信号源选择 (Channel 2 Capture Source Select) 00：GPTIM0_TRGO 01：GPTIM1_TRGO 10：ATIM_TRGO 11：LPT32_CH2 输入
17:16	**CAP1SSEL**	通道 1 捕捉信号源选择 (Channel 1 Capture Source Select) 00：LPT32_CH1 输入 01：XTLF 10：LPOSC 11：RCMF_PSC
15:14	**-**	未实现：读为0
13	**CAP2EDGE**	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新 (Channel2 Captured Edge) 0：下降沿 1：上升沿
12	**CAP1EDGE**	通道 1 当前被捕捉的边沿，在 CC1IF 置位时更新 (Channel 1 Captured Edge) 0：下降沿 1：上升沿

位号	助记符	功能描述
11:10	**CAP2CFG**	通道 2 捕捉边沿选择 (Channel 2 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
9:8	**CAP1CFG**	通道 1 捕捉边沿选择 (Channel 1 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
7:6	**-**	未实现：读为0
5	**POLAR2**	通道2比较输出波形极性选择 (Channel 2 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
4	**POLAR1**	通道1比较输出波形极性选择 (Channel 1 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
3:2	**CC2S**	通道 2 捕捉/比较功能使能 (Channel 2 Capture/Compare Select) 00,11：禁止通道 2 捕捉/比较功能 01：使能通道 2 捕捉功能（LPT32_CH2 为输入） 10：使能通道2比较功能（LPT32_CH2为输出）
1:0	**CC1S**	通道 1 捕捉/比较功能使能 (Channel 1 Capture/Compare Select) 00,11：禁止通道 1 捕捉/比较功能 01：使能通道 1 捕捉功能（LPT32_CH1 为输入） 10：使能通道 1 比较功能（LPT32_CH1 为输出）

29.5.4 LPTIM 目标值寄存器（LPTIM_ARR）

名称	LPTIM_ARR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**ARR**	自动重载目标寄存器(Auto-Reload Register) 当计数器计数值等于 ARR 时，计数器回到初值重新开始向上计数

29.5.5 LPTIM 中断使能寄存器（LPTIM_IER）

名称	LPTIM_IER							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OVR2IE	OVR1IE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TRIGIE	OVIE	CC2IE	CC1IE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	**-**	**未实现：读为0**
9	**OVR2IE**	通道 2 捕捉溢出中断使能 (Channel 2 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
8	**OVR1IE**	通道 1 捕捉溢出中断使能 (Channel 1 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
7:4	**-**	未实现：读为0
3	**TRIGIE**	外部触发到来中断使能位 (External Trigger Interrupt Enable) 1：外部触发到来中断使能 0：外部触发到来中断禁止
2	**OVIE**	计数器溢出中断使能位 (Counter Over-Flow Interrupt Enable) 1：计数器溢出中断使能 0：计数器溢出中断禁止
1	**CC2IE**	捕捉/比较通道 2 中断使能位 (Capture/Compare channel 2 Interrupt Enable) 1：捕捉/比较通道 2 中断使能 0：捕捉/比较通道 2 中断禁止
0	**CC1IE**	捕捉/比较通道 1 中断使能位 (Capture/Compare channel 1 Interrupt Enable) 1：捕捉/比较通道 1 中断使能 0：捕捉/比较通道 1 中断禁止

29.5.6 LPTIM 中断标志寄存器（LPTIM_ISR）

名称	LPTIM_ISR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CAP2OVR	CAP1OVR
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TRIGIF	OVIF	CC2IF	CC1IF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	**未实现：读为0**
9	**CAP2OVR**	通道 2 捕捉溢出，硬件置位，软件写 1 清零 (Channel 2 Over-Capture Interrupt Flag,write 1 to clear) 1：输入捕捉模式下，CC2IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
8	**CAP1OVR**	通道 1 捕捉溢出，硬件置位，软件写 1 清零 (Channel 1 Over-Capture Interrupt Flag,write 1 to clear) 1：输入捕捉模式下，CC1IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
7:4	-	未实现：读为0
3	**TRIGIF**	外部触发到来中断标志位，写 1 清零 (External Trigger Interrupt Flag,write 1 to clear) 1：外部触发到来中断产生 0：无中断产生
2	**OVIF**	计数器溢出中断使能位，写 1 清零 (Counter Over-Flow Interrupt Flag,write 1 to clear) 1：计数器溢出中断产生 0：无中断产生
1	**CC2IF**	捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零 (Capture/Compare channel 2 Interrupt Flag,write 1 to clear) 1：计数器值和比较值 2 匹配，或者发生捕捉事件 0：无中断产生
0	**CC1IF**	捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零 (Capture/Compare channel 1 Interrupt Flag,write 1 to clear) 1：计数器值和比较值 1 匹配，或者发生捕捉事件 0：无中断产生

29.5.7 LPTIM 控制寄存器（LPTIM_CR）

名称	LPTIM_CR
Offset	0x00000018

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	**未实现：读为0**
0	**EN**	LPTIM 使能位 (LPTIM Enable) 1：使能计数器计数 0：禁止计数器计数

29.5.8 LPTIM 捕捉比较寄存器 1（LPTIM_CCR1）

名称	LPTIM_CCR1							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR1[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CCR1[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

29.5.9 LPTIM 捕捉比较寄存器 2（LPTIM_CCR2）

名称	LPTIM_CCR2							
offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CCR2[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	CCR2[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

30 实时时钟（RTC）

30.1 概述

实时时钟(RTC)模块可长时间维持精确计时，功耗极低，在所有功耗模式下都可以工作。

主要特性如下：

- BCD 时间，完整万年历（00~99 年）
- 周期唤醒中断
- 闹钟功能
- 可配置周期定时信号输出
- 数字调校，精度+/-0.477ppm
- 反馈电阻集成
- RTC 计时器部分不复位

30.2 结构框图

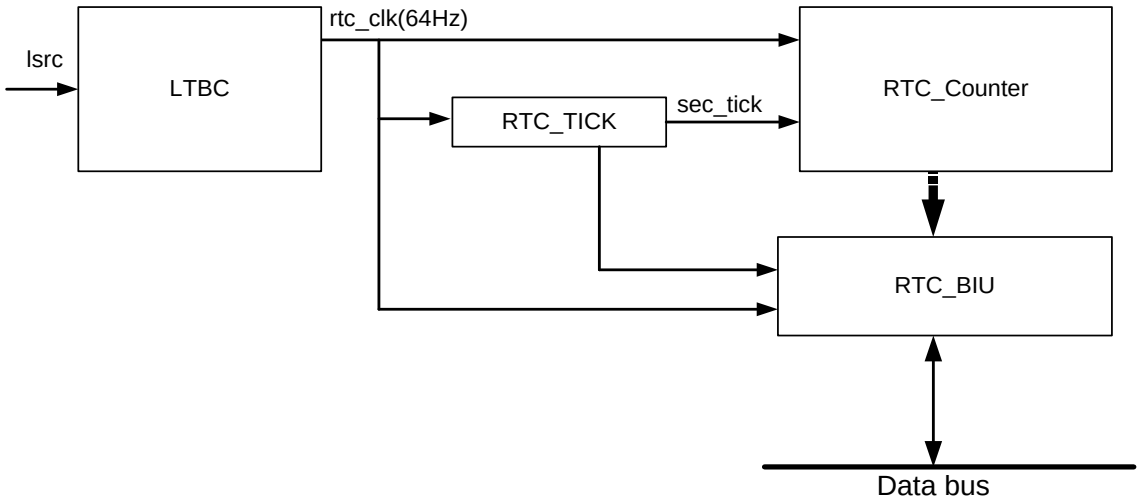


图 30-1RTC 结构框图

LTBC 模块为低功耗时基计数器模块，用于产生系统所需的低速工作时钟，具体描述见 29.3.1 节 LTBC 功能介绍。

RTC_TICK 模块主要用来产生每秒跳变的秒冲信号 sec_tick 以及产生中断需要的 2Hz，4Hz，8Hz，16Hz 等信号。Sec_tick 信号输出到 RTC_Counter 模块用来实现万年历的计数器同步。

RTC_Counter 模块是 RTC 的万年历实现模块，包括秒计数器，分钟计数器，小时计数器，天计数器，周天计数器，月计数器以及年计数器。模块可以实现闰年的自动识别。

30.3 工作原理

RTC 上电后不复位，因此正常工作前需要软件置入当前时间。走时时钟使用 32.768KHz 晶体振荡器。由于晶体振荡器有可能停振，为了保证可靠性，停振检测电路使能后不断检测 32.768KHz 振荡器输出，一旦发现停振，则产生报警中断。同时，软件可以配置 XTLF 停振时是否自动将 RTC 时钟切换到 LPOSC，如果使能这一功能，则 RTC 走时有一定误差，但是并不会停止；如果未使能自动切换，也可以由软件响应停振中断后进行相应处理。

30.3.1 时基计数器（LTBC）

低功耗时基计数器(LTBC)模块用于产生系统所需的低速工作时钟，功能包括：

- 通过对 LSCLK 的预分频得到 64Hz 的 RTC 与 WDT 工作时钟
- 可通过调整计数周期实现 RTC 时钟的数字调校，每 32s 调校一次可实现最小步长为 0.952ppm，调校后理论精度+/-0.477ppm
- PLL 虚拟调校可得到精确秒时标
- 可产生 1KHz、256Hz、64Hz、16Hz、4Hz、1Hz 周期中断，其中 1K 和 256Hz 是未经调校的，其他是经过数字调校的（如果使能了数字调校）
- 64Hz 预分频电路不受芯片复位影响
- 1/256s 精度授时

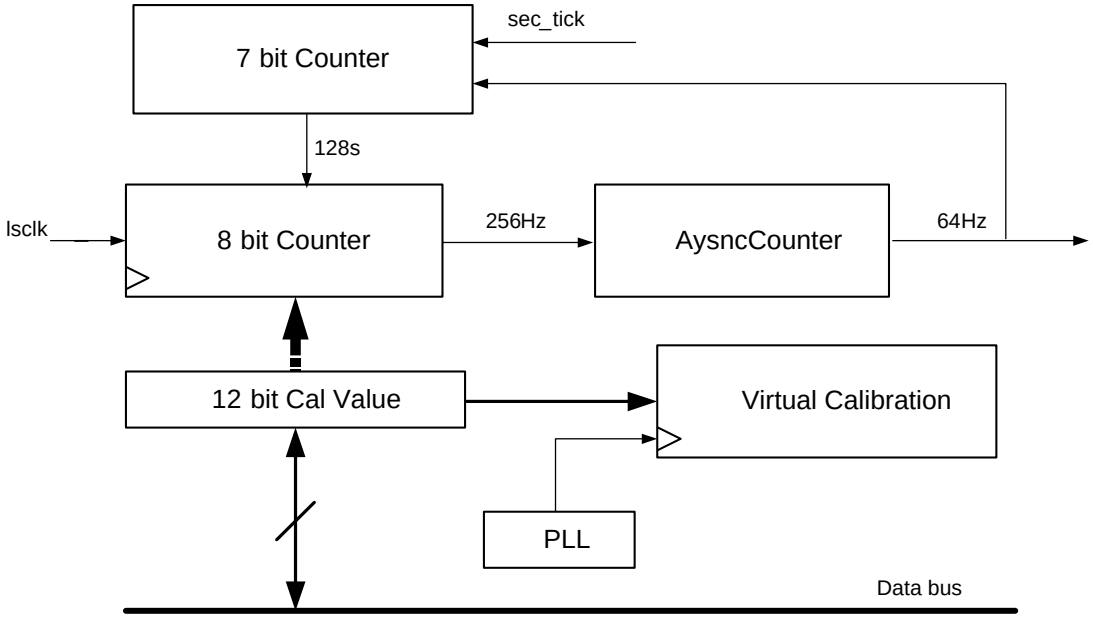


图 30-2LTBC 结构框图

30.3.2 LTBC 数字调校

LTBC主要由同步预分频计数器、异步分频计数器、时钟调校值寄存器、虚拟调校电路和控制寄存器

组成。

数字调校的目的是使RTC能够在较长周期内获得平均准确的计时。由于RTC的时钟源是32768Hz，因此数字调校的最小步长是30.5us，如果在1秒内调整一次，则最高精度只能达到30.517ppm。为了得到更高精度，必须在更长时间周期内进行调整。FM33LC0XX以32s为一个调校周期，每个周期内可以调整0~+/-511个32768Hz时钟周期，因此最高精度为30.5us/32s=0.952ppm，最大调校范围为+/-(511*30.517us/32s)=+/-487ppm，调校后平均最小时钟误差为+/-0.476ppm。

调校值由12bit寄存器组成，其中最高位为符号位，表示计数值增减，其余11bit表示增减的绝对值。为了提高每秒的平均精度，避免较大的秒间跃变，采取将128s调校值平均分配到每秒内的做法，其实现方法如下：

除了最高符号位，其余11bit可分为高4bit的公共值和7bit私有值，其中公共值表示32s内每秒都要调整的值，私有值表示128s内部分秒需要加减1。

Bit11	Bit[10:7]	Bit[6:0]
Sign	Common Value I	Differential Value (D)

调校值公式可表示为：Correction(ppm) = (C*32 + D)*30.517/32000000

假设只使时钟增加0.953ppm，即32s周期只增加一个30.5us，调校值写为0_0000_00001，所以公共值为0，私有值为1，只需要对32s内的一个秒周期加1即可；假设增加487ppm，即32s周期内增加511个30.5us，调校值写为0_1111_11111，公共值为15，私有值为31，表示32s中每秒都要加15，同时其中还有31s需要额外加1。

调校值举例：

ppm	ADJUST[1]	Common	Differential	Expression
0.953	0_0000_00001	0	1	1*30.517/32000000
-125.88	1_0100_00100	4	4	(4*32+4)*30.517/32000000
32.42	0_0001_00010	1	2	(1*32+2)*30.517/32000000
487.32	0_1111_11111	15	31	(15*32+31)*30.517/32000000

注：

[1] ADJUST: Clock Error Adjustment Register

为避免时序冲突，软件应在秒中断后更新ADJUST并启动时钟调校。

以ADJUST=0_0001_00000为例，在每一秒的最后添加1个32768hz周期。

30.3.3 BCD 时间

秒计时

秒计时仅需7bit，从0计数到59，其中bit[3:0]为1秒单位，计数范围0-9；bit[6:4]为10秒单位，计数范围0-5。当计数满60s后触发秒进位信号使分钟计数器加1。

Bit6-4	Bit3-0
0-5	0-9

分钟计时

分计时也仅需7bit，计数范围与秒相同，因此实现方法也相同。

Bit6-4	Bit3-0
0-5	0-9

小时计时

小时计数范围为 0-24，仅需 6bit：

Bit5-4	Bit3-0
0-2	0-9

天计时

天计数范围为 1-31，仅需 6bit，从 1 开始计数，根据月份以及闰年计数到 28/29/30/31，计满后触发天进位信号使月计数器加 1。

Bit5-4	Bit3-0
0-3	0-9

星期计时

星期计数范围为 0-6，仅需 3bit，从 0 到 6 循环计数。

Bit2-0
0-6

月计时

月计数范围为 1-12，仅需 5bit，从 1 开始计数到 12，计满后触发月进位信号使年计数器加 1。

Bit4	Bit3-0
0-1	0-9

年计时

年计数范围为 0-99，需 8bit，从 0 到 99 循环计数。

Bit7-4	Bit3-0
0-9	0-9

30.3.4 RTC 使能与停止

RTC 上电后自动工作，软件应在 32.768K 晶体振荡器完全起振后再设置当前时间；在晶体振荡器起振之前芯片使用内部环振计时，偏差较大。

如果软件关闭了 XTLF，并且不将 LSCLK 切换到 LPOSC，则 RTC 停止计时。

30.3.5 RTC 时间设置

软件可以在任意时刻直接设置 RTC 时间寄存器，通常建议在 XTLF 完成起振后再设置时间；由于设置时间寄存器的操作与 RTC 走时为异步操作关系，建议软件在秒中断事件之后进行时间设置，并且在置时后读出时间值校验。

注意，硬件并不检查时间合法性，软件须保证写入的 BCD 时间正确。

同时 FM33LC0XX 支持 ms 级授时，即可以设置时间到 3.9ms 级别精度（1/256s）。此外，当软件写入秒时间时，硬件自动清零 64Hz->1Hz 的秒内计数器，以便实现秒对齐。

为了提高抗干扰能力，FM33LC0XX 提供时间写保护功能，必须先对写保护寄存器写入 0xACACACAC，才能改写时间寄存器，置时完成后软件可以通过写入任意其他值来禁止时间寄存器的写入，恢复写保护。

30.3.6 RTC 时间读取

时间读取方式 1：

- 读当前时间寄存器值
- 再次读当前时间寄存器值
- 如果 2 次读取内容一致，则为正确的当前时间；如果两次读取内容不一致，则重复前两个步骤。

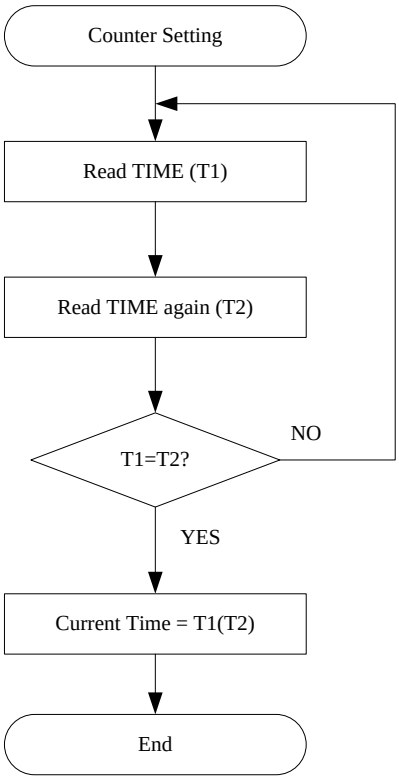


图 30-3RTC 时间读取流程图

时间读取方式 2：

软件在 1s 中断发生后立即读取时间寄存器，能保证读到正确的当前时间值。

30.3.7 闰年判断

FM33LC0XX 的 RTC 模块会自动判断闰年。

闰年的条件：(mod 400 == 0) or (mod 4 ==0 and mod 100 <> 0)

30.4 寄存器

offset 地址	名称	符号
RTC(模块起始地址:0x40011000)		
0x00000000	RTC 写使能寄存器 （RTC Write Enable Register）	RTC_WER
0x00000004	RTC 中断使能寄存器 （RTC Interrupt Enable Register）	RTC_IER
0x00000008	RTC 中断标志寄存器 （RTC Interrupt Status Register）	RTC_ISR
0x0000000C	BCD 时间秒寄存器 （BCD format time second registers）	RTC_BCDSEC
0x00000010	BCD 时间分钟寄存器 （BCD format time minute registers）	RTC_BCDMIN
0x00000014	BCD 时间小时寄存器 （BCD format time hour registers）	RTC_BCDHOUR
0x00000018	BCD 时间天寄存器 （BCD format time day registers）	RTC_BCDDAY
0x0000001C	BCD 时间星期寄存器 （BCD format time week registers）	RTC_BCDWEEK
0x00000020	BCD 时间月寄存器 （BCD format time month registers）	RTC_BCDMONTH
0x00000024	BCD 时间年寄存器 （BCD format time year registers）	RTC_BCDYEAR
0x00000028	闹钟寄存器 （RTC Alarm Register）	RTC_ALARM
0x0000002C	RTC 时间信号输出寄存器 （RTC Time Mark Select）	RTC_TMSEL
0x00000030	LTBC 数值调整寄存器 （RTC time Adjust Register）	RTC_ADJUST
0x00000034	LTBC 数值调整方向寄存器 （RTC time Adjust Sign Register）	RTC_ADSIGN
0x0000003C	毫秒计数值寄存器 （RTC Sub-Second Counter）	RTC_SBSCNT
0x00000070	RTC 备份寄存器组 0 （RTC Backup Registers 0）	RTC_BKR0
0x00000074	RTC 备份寄存器组 1 （RTC Backup Registers 1）	RTC_BKR1
0x00000078	RTC 备份寄存器组 2 （RTC Backup Registers 2）	RTC_BKR2
0x0000007C	RTC 备份寄存器组 3 （RTC Backup Registers 3）	RTC_BKR3
0x00000080	RTC 备份寄存器组 4 （RTC Backup Registers 4）	RTC_BKR4
0x00000084	RTC 备份寄存器组 5 （RTC Backup Registers 5）	RTC_BKR5
0x00000088	RTC 备份寄存器组 6 （RTC Backup Registers 6）	RTC_BKR6
0x0000008C	RTC 备份寄存器组 7 （RTC Backup Registers 7）	RTC_BKR7

30.4.1 RTC 写使能寄存器（RTC_WER）

名称	RTC_WER							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							WE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WE	RTC 写使能寄存器 (RTC Write Enable) 当 CPU 向 RTCWE 写入 0xACACACAC 时，允许 CPU 向 RTC 的 BCD 时间寄存器写入初值，这时 RTCWE 置 1；当 CPU 向 RTCWE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 RTCWE 清 0。

30.4.2 RTC 中断使能寄存器（RTC_IER）

名称	RTC_IER							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			ADJ_IE	ALARM_IE	1KHZ_IE	256HZ_IE	64HZ_IE
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IE	8HZ_IE	4HZ_IE	2HZ_IE	SEC_IE	MIN_IE	HOUR_IE	DAY_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IE	调校周期中断使能 (time Adjust Interrupt Enable) 1：中断使能打开 0：中断使能禁止

位号	助记符	功能描述
11	ALARM_IE	闹钟中断使能 (Alarm Interrupt Enable) 1：中断使能打开 0：中断使能禁止
10	1KHZ_IE	1khz 中断使能 (1Khz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
9	256HZ_IE	256hz 中断使能 (256hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
8	64HZ_IE	64hz 中断使能 (64hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
7	16HZ_IE	16hz 中断使能 (16hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
6	8HZ_IE	8hz 中断使能 (8hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
5	4HZ_IE	4hz 中断使能 (4hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
4	2HZ_IE	2hz 中断使能 (2hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
3	SEC_IE	秒中断使能 (1hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
2	MIN_IE	分中断使能 (Minute Interrupt Enable) 1：中断使能打开 0：中断使能禁止
1	HOUR_IE	小时中断使能 (Hour Interrupt Enable) 1：中断使能打开 0：中断使能禁止
0	DAY_IE	天中断使能 (Day Interrupt Enable) 1：中断使能打开 0：中断使能禁止

30.4.3 RTC 中断标志寄存器（RTC_ISR）

名称	RTC_ISR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-			ADJ_IF	ALARM_IF	1KHZ_IF	256HZ_IF	64HZ_IF
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IF	8HZ_IF	4HZ_IF	2HZ_IF	SEC_IF	MIN_IF	HOUR_IF	DAY_IF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IF	调校周期中断标志。写 1 清零 (time Adjust Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
11	ALARM_IF	闹钟中断标志。写 1 清零 (Alarm Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
10	1KHZ_IF	1khz 中断标志。写 1 清零 (1Khz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
9	256HZ_IF	256hz 中断标志。写 1 清零 (256hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
8	64HZ_IF	64hz 中断标志。写 1 清零 (64hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
7	16HZ_IF	16hz 中断标志。写 1 清零 (16hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
6	8HZ_IF	8hz 中断标志。写 1 清零 (8hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
5	4HZ_IF	4hz 中断标志。写 1 清零 (4hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
4	2HZ_IF	2hz 中断标志。写 1 清零 (2hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
3	SEC_IF	秒中断标志。写 1 清零 (1hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
2	MIN_IF	分中断标志。写 1 清零 (Minute Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生

位号	助记符	功能描述
1	HOUR_IF	小时中断标志。写 1 清零 (Hour Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生
0	DAY_IF	天中断标志。写 1 清零 (Day Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生

30.4.4 BCD 时间秒寄存器（RTC_BCDSEC）

名称	RTC_BCDSEC							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	SEC	秒时间数值，BCD 格式。(Binary-Coded Decimal format Seconds Register)

30.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）

名称	RTC_BCDMIN							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MIN						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	MIN	分钟时间数值，BCD 格式。(Binary-Coded Decimal format Minutes Register)

30.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）

名称	RTC_BCDHOUR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		HOUR					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	HOUR	小时数值，BCD 格式。(Binary-Coded Decimal format Hours Register)

30.4.7 BCD 时间天寄存器（RTC_BCDDAY）

名称	RTC_BCDDAY							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAY					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	DAY	天数值，BCD 格式。(Binary-Coded Decimal format Date Register)

30.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）

名称	RTC_BCDWEEK							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WEEK		
位权限	U-0					R/W-xxx		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	WEEK	周数值，BCD 格式。(Binary-Coded Decimal format Week Register)

30.4.9 BCD 时间月寄存器（RTC_BCDMONTH）

名称	RTC_BCDMONTH							
Offset	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			MONTH				
位权限	U-0			R/W-x xxxx				

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4:0	MONTH	月数值，BCD 格式。(Binary-Coded Decimal format Month Register)

30.4.10 BCD 时间年寄存器（RTC_BCDYEAR）

名称	RTC_BCDYEAR

Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	YEAR							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	YEAR	年数值，BCD 格式。(Binary-Coded Decimal format Year Register)

30.4.11 闹钟寄存器（RTC_ALARM）

名称	RTC_ALARM							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HOUR					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
位名	-	MIN						
位权限	U-0	R/W-000 0000						
位	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31:22	-	RFU：未实现，读为 0
21:16	HOUR	闹钟的小时数值。(Alarm Hour Register)
15	-	RFU：未实现，读为 0
14:8	MIN	闹钟的分数值。(Alarm Minute Register)
7	-	RFU：未实现，读为 0
6:0	SEC	闹钟的秒数值。(Alarm Second Register)

30.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）

名称	RTC_TMSEL

offset	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TMSEL			
位权限	U-0				R/W-0000			

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:0	TMSEL	频率输出选择信号：(Time Mark Select) 0000：RFU 0001：RFU 0010：输出秒计数器进位信号，高电平宽度 1s 0011：输出分计数器进位信号，高电平宽度 1s 0100：输出小时计数器进位信号，高电平宽度 1s 0101：输出天计数器进位信号，高电平宽度 1s 0110：输出闹钟匹配信号 0111：输出 32 秒方波信号 1000：RFU 1001：反向输出秒计数器进位信号 1010：反向输出分计数器进位信号 1011：反向输出小时计数器进位信号 1100：反向输出天计数器进位信号 1101：反向输出闹钟匹配信号 1110：RFU 1111：输出 RTC 内部秒时标方波

30.4.13 LTBC 数值调整寄存器（RTC_ADJUST）

名称	RTC_ADJUST							
Offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							ADJUST[8]
位权限	U-0							R/W-x
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJUST[7:0]							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8:0	ADJUST	LTBC 补偿调整数值 (Time Adjust)

30.4.14 LTBC 数值调整方向寄存器（RTC_ADSIGN）

名称	RTC_ADSIGN							
Offset	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							ADSIGN
位权限	U-0							R/W-x

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	ADSIGN	LTBC 补偿方向 (Adjust Sign) 0：表示增加计数初值 1：表示减少计数初值

30.4.15 毫秒计数值寄存器（RTC_SBSCNT）

名称	RTC_SBSCNT							
offset	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSCNT							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	MSCNT	毫秒计数器值，有效位 8bit，精度 3.9ms。(Milli-Second Counter)

30.4.16 RTC 备份寄存器组 x（RTC_BKRx）

名称	RTC_BKRx(x=0,1,2,3,4,5,6,7)							
Offset	0x00000070 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	BKP[31:24]							
位权限	R/W-xxxx xxxx							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BKP[23:16]							
位权限	R/W-xxxx xxxx							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BKP[15:8]							
位权限	R/W-xxxx xxxx							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BKP[7:0]							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:0	BKP	备份寄存器，可读写，无复位值 (RTC Backup Registers)

31 LCD 显示

31.1 概述

LCD 显示驱动模块用于驱动段码式液晶屏，能够支持 4、6、8COM，最大显示段数分别为 128 段（4COM）、180 段（6COM）和 224 段（8COM）。

主要特点：

- 最大支持 8×28、6×30、4×32 的显示段数
- 1/3bias、1/4bias
- 16 级灰度可调
- LCD 驱动支持片内电阻型
- 支持闪烁功能，且闪烁频率可调
- 支持间歇式点亮功能，点亮、熄灭时间可配置
- 支持全亮、全灭功能
- 低功耗，LCD 驱动可以在 Active 模式、Sleep 模式和 DeepSleep 模式下工作
- 支持 Type A 和 Type B 两种 LCD 驱动波形（可配置）
- 典型帧刷新频率 64Hz

31.2 结构框图

LCD 显示驱动模块结构框图如图 31-1 所示：

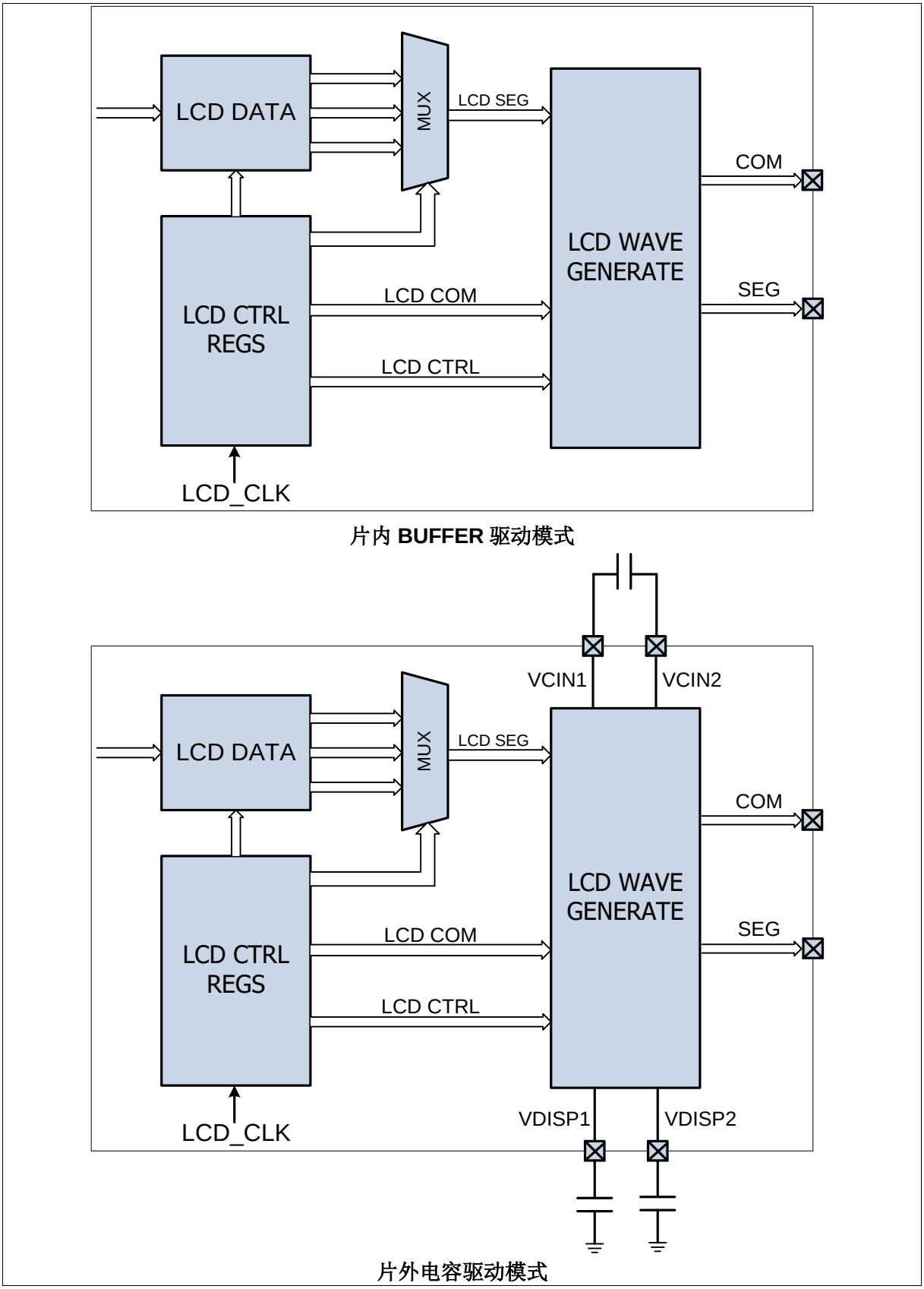


图 31-1 LCD 显示控制模块结构框图

31.3 IO 配置

LCD驱动电路工作时最多会占用36个GPIO，在使用LCD前，需要将用到的引脚设置为模拟功能（PxyFCR=11），并将相应的COMEN或SEGEN打开。如果在同一个引脚上复用了LCD之外的其他模拟功能，则必须保证其他模式外设不会使用这个引脚通道。

31.4 功能说明

31.4.1 工作时钟和显示帧频率

LCD驱动电路使用LSCLK工作，其典型频率在32KHz左右。通过配置DF寄存器，可以设置LCD显示的帧频率。帧频率的计算公式如下（注意DF不能为0）：

COM 数量	帧频率 Hz	
	A 类波形	B 类波形
4	显示电路工作频率 / (4 ×DF[7:0] × 2)	显示电路工作频率 / (4 ×DF[7:0] × 4)
6	显示电路工作频率 / (6 ×DF[7:0] × 2)	显示电路工作频率 / (6 ×DF[7:0] × 4)
8	显示电路工作频率 / (8 ×DF[7:0] × 2)	显示电路工作频率 / (8 ×DF[7:0] × 4)

表31-1帧频率计算公式

下表举例说明了DF寄存器的取值与帧频率之间的关系。通常情况下将帧频率设置为60Hz左右。

帧频率（Hz）	工作时钟（Hz）	4 公共端		6 公共端		8 公共端	
		A 类	B 类	A 类	B 类	A 类	B 类
50	32768	82	41	54	27	41	20
58	32768	70	35	47	24	35	17
64	32768	64	32	42	21	32	16
70	32768	58	29	39	20	29	14
75	32768	54	27	36	18	27	13

表31-2典型帧频率和DF的关系

31.4.2 LCD Type A 扫描波形

下图是1/4 duty，1/3bias，TypeA类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

4个公共端会依次有效，在某个公共端有效的时间内，SEG输出合适的电平，与COM电平共同施加在LCD面板上，压差大的段码将被显示，压差小的段码不显示。

1/3 bias表示LCD驱动电路能够输出4种驱动电平，对偏置电压平均分配得到。通过LCDBIAS寄存器可以配置VLCD偏置电压大小，最大不超过电源电压。

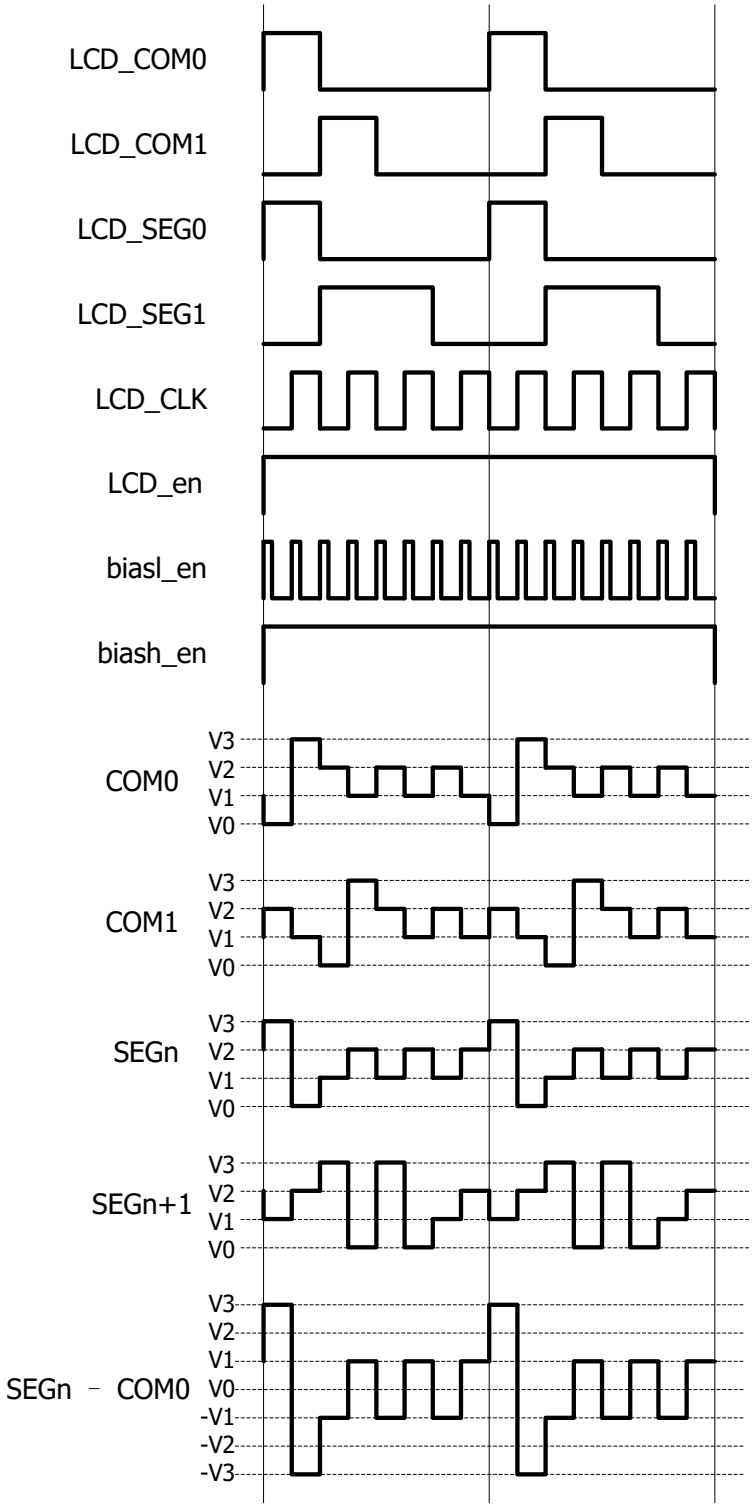


图 31-2LCD 驱动波形(1/4 duty, 1/3 bias, type A)

31.4.3 LCD Type B 扫描波形

下图是1/4 duty，1/3bias，TypeB类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

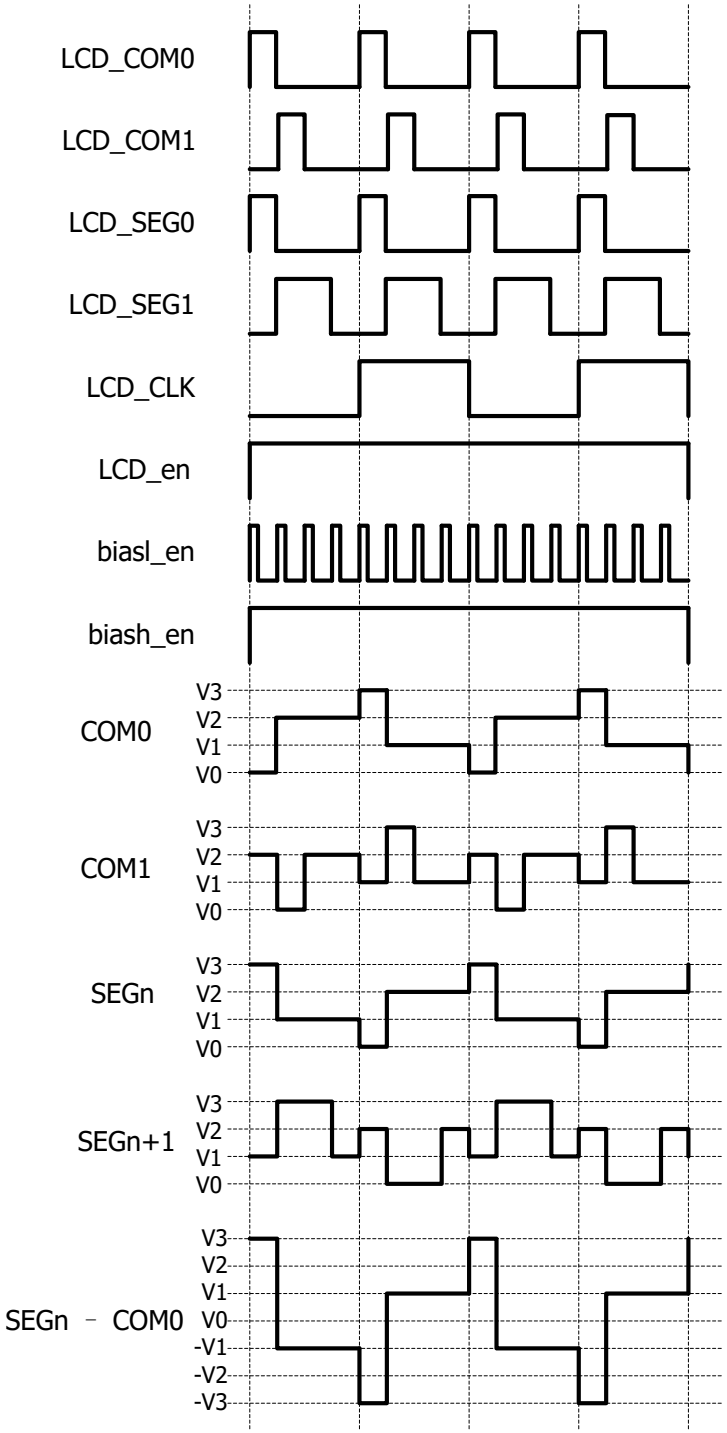


图 31-3 LCD 驱动波形(1/4 duty, 1/3 bias, type B)

31.4.4 片内 buffer 驱动模式

片内buffer驱动模式由电源电压通过分压电阻产生等分电压，分压输入到低功耗buffer以增强驱动能力，buffer输出连接至波形产生模块后产生COM和SEG信号，此模式无需片外器件，功耗较低。

其结构示意图如下：

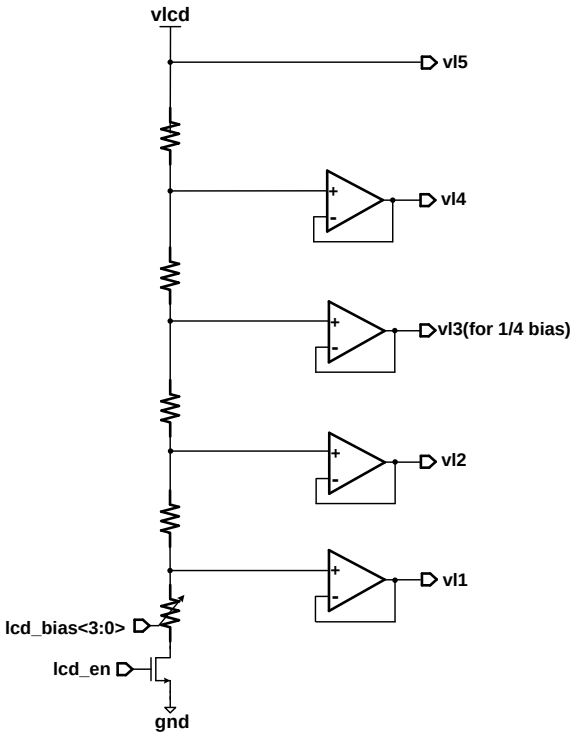


图 31-4LCD 片内电阻 buffer 型驱动电路

驱动Buffer的输出阻抗大约为5Kohm。

通过配置LCDBIAS寄存器，可以实现对LCD驱动电压的调整，具体参见30.4.7“偏置电压调整”章节。

31.4.5 显示闪烁功能

软件可以设置显示控制寄存器 DISPCTRL 中的 FLICKER 位为 1，来使能显示闪烁。FLICKER 使能后，根据 TON 和 TOFF 寄存器值确定闪烁频率。在使能 FLICKER 功能之前应先设置 TON/TOFF 并设置 MD 打开显示，若不设置 TON/TOFF，则其复位值为 0，显示会以 64Hz 闪烁。若不先打开显示，FLICKER 设置无效，不会有显示。

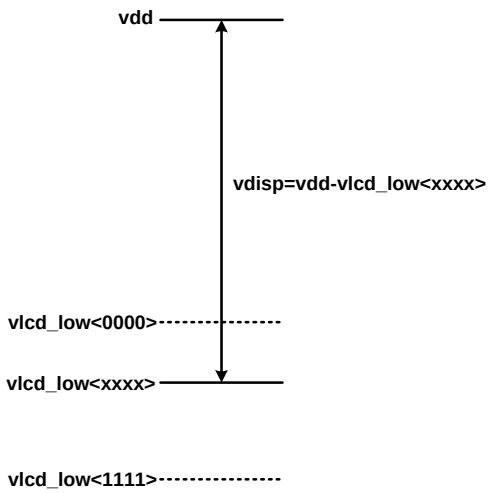
TON/TOFF 最小步长为 T_{step} = COM*DF[7:0]*2*16/32768Hz，实际 ON/OFF 时间为 TON/TOFF * T_{step}。显示和熄灭与帧扫描同步，即在一帧扫描完后熄灭，或在一帧开始时点亮，熄灭或点亮后给出相应中断。由于帧结束信号是 64HZ 的，因此 TON/TOFF 的计数值应为寄存器设置值 x16。

31.4.6 偏置电压调整

LCD 输出的显示电压范围可以调节以适应不同规格的液晶面板，输出电压范围可以表示为：

VDISP=VDD-VLCD_LOW

其中 VLCD_LOW 可以由 LCDBIAS[3:0]调节，LCDBIAS=0000 对应的 VLCD_LOW 电压最高，输出电压范围 VDISP=VDD-VLCD_LOW 最小；LCDBIAS=1111 对应的 VLCD_LOW 电压最低，输出电压范围 VDISP=VDD-VLCD_LOW 最大，如下图所示：



应用中应根据实际 LCD 面板特性，选择合适的 VDISP 电压，可参考 30.6.12 中的表格进行配置。

31.5 寄存器

offset 地址	名称	符号
LCD(模块起始地址:0x40010C00)		
0x00000000	显示控制寄存器 （LCD Control Register）	LCD_CR
0x00000004	显示测试控制寄存器 （LCD test Register）	LCD_TEST
0x00000008	显示频率控制寄存器 （LCD Frequency Control Register）	LCD_FCR
0x0000000C	闪烁时间寄存器 （LCD Flick Time Register）	LCD_FLKT
0x00000014	显示中断使能寄存器 （LCD Interrupt Enable Register）	LCD_IER
0x00000018	显示中断标志寄存器 （LCD Interrupt Status Register）	LCD_ISR
0x00000024	显示数据缓存寄存器 0 （LCD data buffer registers 0）	LCD_DATA0
0x00000028	显示数据缓存寄存器 1 （LCD data buffer registers 1）	LCD_DATA1
0x0000002C	显示数据缓存寄存器 2 （LCD data buffer registers 2）	LCD_DATA2
0x00000030	显示数据缓存寄存器 3 （LCD data buffer registers 3）	LCD_DATA3
0x00000034	显示数据缓存寄存器 4 （LCD data buffer registers 4）	LCD_DATA4
0x00000038	显示数据缓存寄存器 5 （LCD data buffer registers 5）	LCD_DATA5
0x0000003C	显示数据缓存寄存器 6 （LCD data buffer registers 6）	LCD_DATA6
0x00000040	显示数据缓存寄存器 7 （LCD data buffer registers 7）	LCD_DATA7
0x00000050	COM 使能控制寄存器 （LCD COM Enable Register）	LCD_COMEN
0x00000054	SEG 使能控制寄存器 0 （LCD SEG Enable Register0）	LCD_SEGEN0

31.5.1 显示控制寄存器（LCD_CR）

名称	LCD_CR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						IC_CTRL	
位权限	U-0						R/W-01	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ENMODE	FLICK	-		BIAS			

位权限	R/W-0	R/W-0	U-0		R/W-1110			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BIASMD	ANTIPOLAR	WFT	LMUX		EN
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

位号	助记符	功能描述
31:18	-	RFU：未实现，读为 0
17:16	**IC_CTRL**	偏置电路输入电流源大小控制 (Input bias Current Control) 00：电流最大 01：电流次大 10：电流次小 11：电流最小
15	**ENMODE**	驱动模式选择 (LCD Enabling Mode) 0：RFU 1：片内电阻型驱动
14	**FLICK**	显示闪烁使能位 (LCD Flick Enable) 1：显示闪烁，闪烁频率由 TON 和 TOFF 寄存器设置 0：关闭闪烁
13:12	**-**	RFU：未实现，读为 0
11:8	**BIAS**	LCD 偏置电平选择位，用于显示灰度控制 (LCD Bias Voltage Select)
7:6	**-**	RFU：未实现，读为 0
5	**BIASMD**	偏置类型选择 (Bias Mode) 1：1/3 Bias 0：1/4 Bias
4	**ANTIPOLAR**	防极化使能 (Anti-Polarization) 1：COM 和 SEG 在 LCD 关闭情况下接地 0：COM 和 SEG 在 LCD 关闭情况下浮空
3	**WFT**	驱动波形选择 (Waveform Format) 1：B类波形 0：A类波形
2:1	**LMUX**	COM 数量选择 (Segment Line Mux) 00：4COM 01：6COM 10/11：8COM
0	**EN**	LCD 显示使能位 (LCD Enable) 1：启动 LCD 显示 0：关闭 LCD 显示

LCDBIAS[3:0]	不同 VDD 下内部偏置电压(V)			
	5	4.5	3.6	3.0
0000	2.74	2.47	1.97	1.64
0001	2.83	2.54	2.03	1.69
0010	2.92	2.62	2.10	1.75
0011	3.01	2.71	2.17	1.81
0100	3.12	2.80	2.24	1.87
0101	3.23	2.90	2.32	1.94
0110	3.35	3.01	2.41	2.01
0111	3.47	3.13	2.50	2.08

LCDBIAS[3:0]	不同 VDD 下内部偏置电压(V)			
	5	4.5	3.6	3.0
1000	3.61	3.25	2.60	2.17
1001	3.76	3.39	2.71	2.26
1010	3.93	3.53	2.83	2.35
1011	4.10	3.69	2.95	2.46
1100	4.30	3.87	3.09	2.58
1101	4.51	4.06	3.25	2.71
1110	4.75	4.27	3.42	2.85
1111	5.00	4.50	3.60	3.00

31.5.2 显示测试控制寄存器（LCD_TEST）

名称	LCD_TEST							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LCCTRL	-						TESTEN
位权限	R/W-0	U-0						R/W-0

位号	助记符	功能描述
31:8	-	未实现，读为0
7	LCCTRL	LCD测试控制位，仅在测试模式下有效 (Line Constant Control) COM、SEG 输出电平由测试模式下的引脚输出数据寄存器决定。不同设置下 SEG 或 COM 输出的结果参见后文表格。
6:1	-	未实现，读为0
0	TESTEN	测试模式使能位 (Test mode Enable) 1：LCD 测试模式使能。在 LCD 测试模式下，LCD 引脚静态输出模拟直流电平，所有与动态扫描时间以及扫描波形相关寄存器设置无效 0：正常工作模式，测试模式无效，相关测试寄存器控制无效

测试模式下引脚输出电平：

LCCTRL	DISPDATA	COM 引脚输出电平	SEG 引脚输出电平
		1/3bias	1/3bias
0	0	V3	V2
0	1	V1	V4
1	0	V2	V3
1	1	V4	V1

31.5.3 测试模式下引脚输出数据寄存器

这组寄存器只在测试模式下有效，与相关显示寄存器共享存储空间。

名称	LCD 测试模式下引脚输出数据寄存器 TDISPDATA							
Offset								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TDISPDATA0	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
TDISPDATA1	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
TDISPDATA2	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
TDISPDATA3	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
TDISPDATA4	-	-	-	-	-	-	-	-
TDISPDATA5	-	-	-	-	-	-	-	-
TDISPDATA6	-	-	-	-	-	-	-	-
TDISPDATA7	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

31.5.4 显示频率控制寄存器（LCD_FCR）

名称	LCD_FCR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:8	**-**	未实现，读为0
7:0	**DF**	显示预分频寄存器 (Display Frequency)

31.5.5 闪烁时间寄存器（LCD_FLKT）

名称	LCD_FLKT
offset	0x0000000C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TOFF[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TON[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	**未实现，读为0**
15:8	**TOFF**	闪烁显示熄灭时间寄存器 (Display-Off Time) TOFF最小步长为T_{step} = COM*DF[7:0]*2*16/32768Hz，实际OFF时间为TOFF * T_{step}
7:0	**TON**	闪烁显示点亮时间寄存器 (Display-On Time) TON最小步长为T_{step} = COM*DF[7:0]*2*16/32768Hz，实际ON时间为TON * T_{step}

31.5.6 显示中断使能寄存器（LCD_IER）

名称	LCD_IER							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DONIE	DOFFIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	未实现，读为0
1	**DONIE**	显示点亮中断使能位 (Display-On Interrupt Enable) 1：显示点亮中断使能 0：显示点亮中断禁止
0	**DOFFIE**	显示熄灭中断使能位 (Display-OFF Interrupt Enable) 1：显示熄灭中断使能 0：显示熄灭中断禁止

31.5.7 显示中断标志寄存器（LCD_ISR）

名称	LCD_ISR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DONIF	DOFFIF
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:2	-	未实现，读为0
1	DONIF	显示点亮中断标志 (Display-On Interrupt Flag,write 1 to clear) 显示由灭变亮时硬件产生中断标志，硬件置位，软件清零
0	DOFFIF	显示熄灭中断标志 (Display-OFF Interrupt Flag,write 1 to clear) 显示由亮变灭时硬件产生中断标志，硬件置位，软件清零

31.5.8 显示数据缓存寄存器 x（LCD_DATAx）

LCD 显示模块内有 8 个 32 bit 的显示数据寄存器。均为可读可写，复位值为 0。

名称	LCD_DATAx(x=0,1,2,3,4,5,6,7)							
Offset	0x00000024 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DSDA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DSDA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DSDA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DSDA[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	DSDA	LCD 显示数据 (Display Data)

31.5.8.1 4COM 显示数据寄存器

名称	4com 显示数据寄存器							
Offset	0x00000024 ~ 0x00000038							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3

31.5.8.2 6COM 显示数据寄存器

名称	6com 显示数据寄存器							
Offset	0x00000024 ~ 0x00000040							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	6com 显示数据寄存器							
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM4	SEG28 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG29 COM5	SEG28 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5

31.5.8.3 8COM 显示数据寄存器

名称	8com 显示数据寄存器							
Offset	0x00000024 ~ 0x00000040							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1

名称	8com 显示数据寄存器							
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
DISPDATA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM6	SEG6 COM6	SEG5 COM6	SEG4 COM6	SEG3 COM6	SEG2 COM6	SEG1 COM6	SEG0 COM6
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM6	SEG14 COM6	SEG13 COM6	SEG12 COM6	SEG11 COM6	SEG10 COM6	SEG9 COM6	SEG8 COM6

名称	8com 显示数据寄存器							
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM6	SEG22 COM6	SEG21 COM6	SEG20 COM6	SEG19 COM6	SEG18 COM6	SEG17 COM6	SEG16 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6
DISPDATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM7	SEG6 COM7	SEG5 COM7	SEG4 COM7	SEG3 COM7	SEG2 COM7	SEG1 COM7	SEG0 COM7
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM7	SEG14 COM7	SEG13 COM7	SEG12 COM7	SEG11 COM7	SEG10 COM7	SEG9 COM7	SEG8 COM7
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM7	SEG22 COM7	SEG21 COM7	SEG20 COM7	SEG19 COM7	SEG18 COM7	SEG17 COM7	SEG16 COM7
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7

31.5.9 COM 使能控制寄存器（LCD_COMEN）

名称	LCD_COMEN							
Offset	0x00000050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				COMEN[3:0]			
位权限	U-0				R/W-0000			

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:0	COMEN	LCD COM 输出使能控制 (COM Enable) 1：COM 输出使能 0：COM 输出禁止

31.5.10 SEG 使能控制寄存器 0（LCD_SEGEN0）

名称	LCD_SEGEN0							
Offset	0x00000054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	SEG31_COM7_EN	SEG30_COM6_EN	SEG29_COM5_EN	SEG28_COM4_EN	SEGEN[27:24]			
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SEGEN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SEGEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGEN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	SEG31_COM7_EN	LCD SEG 和 COM 输出使能控制 (SEG31 or COM7 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
30	SEG30_COM6_EN	LCD SEG 和 COM 输出使能控制 (SEG30 or COM6 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
29	SEG29_COM5_EN	LCD SEG 和 COM 输出使能控制 (SEG29 or COM5 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
28	SEG28_COM4_EN	LCD SEG 和 COM 输出使能控制 (SEG28 or COM4 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
27:0	SEGEN	LCD SEG 输出使能控制 (SEG Enable) 每个 bit 对应一个特定的 SEG 1：SEG 输出使能 0：SEG 输出禁止

32 ADC

32.1 概述

FM33LC0XX 带有 1Msps 12bit SAR-ADC，可实现温度、电池电压或其他直流信号的测量功能。主要特点为：

- 工作电压 1.8～5.5V
- 输入信号幅度 0~VDDA
- 最高采样率 1Msps（F_{ADC}=16Mhz）
- 16 个单端输入通道，包含温度传感器、内部基准电压、运放输出 x2、12 个外部通道
- 8 个外部快速通道，8 个低速通道
- 可配置的采样保持时间
- 支持单次转换和连续转换
- 支持 DMA
- 支持过采样硬件平均，最高 16bit 输出（256 次平均）

32.2 结构框图

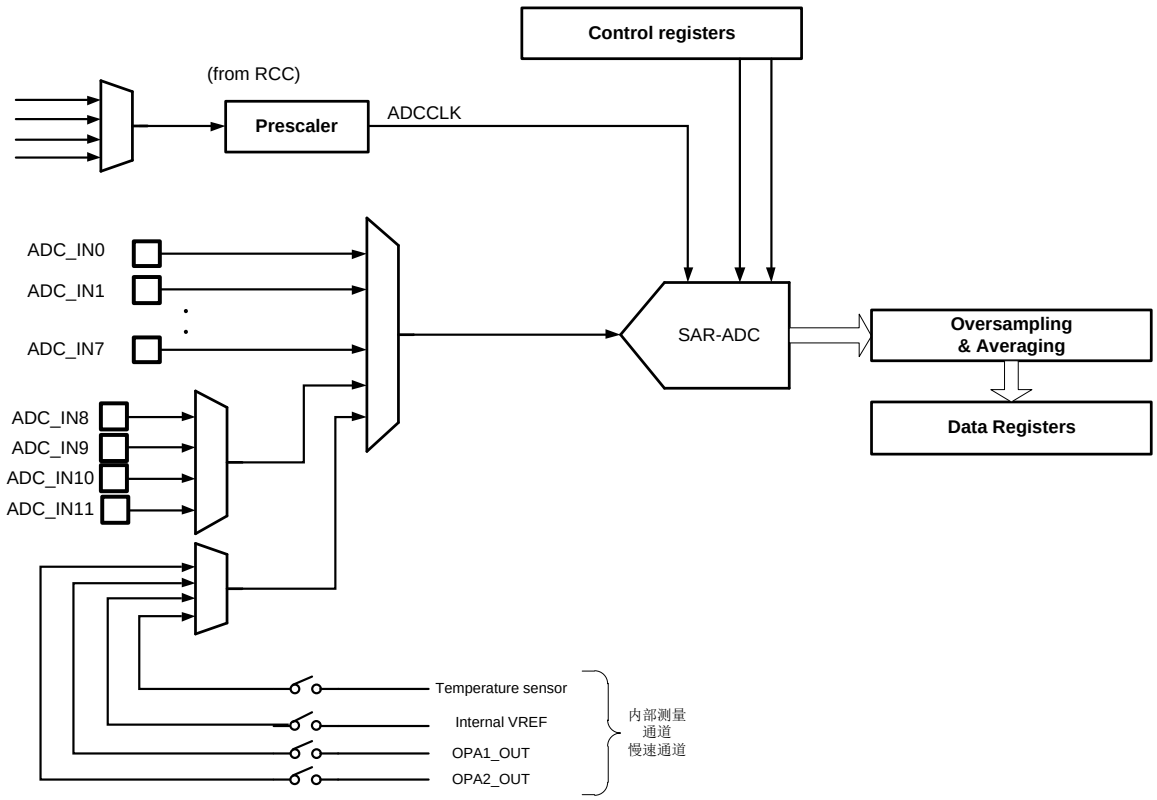


图 32-1 ADC 结构框图

ADC 本身有 10 个模拟输入通道，其中 0~7 直接接到 PAD 输入，作为高速通道使用；8 和 9 用于通道扩展，如上图，8 号通道扩展 4 个外部 PAD 低速输入通道，9 号通道扩展 4 个内部输入通道。

32.3 输入通道

ADC 支持 4 个内部通道和 12 个外部通道。

通道	IO	说明
ADC_IN0	PC9	外部快速通道，仅支持单端输入
ADC_IN1	PC10	
ADC_IN2	PD11	
ADC_IN3	PD0	
ADC_IN4	PD1	
ADC_IN5	PD2	
ADC_IN6	PA13	
ADC_IN7	PA14	
ADC_IN8	PC7	外部慢速通道，仅支持单端输入
ADC_IN9	PC8	
ADC_IN10	PA15	
ADC_IN11	PC6	
TS	N/A	温度传感器采样通道
VREFINT		内部基准源采样通道
OPA1		高速运放输出采样通道
OPA2		普通运放输出采样通道

表32-1ADC输入通道分配

32.4 功能描述

32.4.1 采样值与实际电压转换

ADC 一般使用电源电压作为基准电压，在电源电压发生变化时，特定输入信号电平对应的转换值也会发生变化。为了能够得到准确的绝对电压，解决方案如下：

- 芯片出厂时在 VDDA=3V 情况下，测量 VREFINT 的电压并保存在芯片 Flash 中
- 在以上条件下，使用 ADC 转换 VREFINT 输出，得到转换值 VREFINT_CAL 并保存在芯片中
- 芯片实际应用中，由于不知道当前 VDDA 电压，ADC 先测量 VREFINT 得到转换值 VREFINT_DATA；通过以下公式可以得到当前实际的 VDDA：

$$VDDA = \frac{VREFINT\_CAL}{VREFINT\_DATA} \times 3V$$

- 假设 ADC 对某个输入通道的采样值为 ADC_DATA，通过以下公式可以得到当前某个输入通道的实际电压（12bit 输出）

$$V_{CHANNEL} = \frac{VREFINT\_CAL \times ADC\_DATA}{VREFINT\_DATA \times 4095} \times 3V$$

- 采用这个方式，不需要知道每颗芯片 VREFINT 的实际电压值，仅需计算当前 VREFINT 采样值和出厂测试值的比例；

VREF1p2 采样的软件配置方法

软件使用 ADC 采样 VREF1p2 时，需要按照以下步骤：

- 置位 VREF_EN 寄存器，使能 VREF1p2 模块
- 置位 BUFFERCTRL.VREFBUFFER_EN，使能 VREF 输出 BUFFER
- 等待 VREF 建立，通过查询 VREF_RDY 寄存器，或通过 VREF_IF 中断
- 使能 ADC 的 REFCH 通道
- 使能 ADC 开始转换

32.4.2 温度传感器

ADC 使用内部通道测量 PTAT 输出电压，得到转换数据 TS_DATA，然后使用下式可以计算当前温度：

$$\text{Temperature} = \frac{TS\_DATA \times \frac{VREFINT\_CAL}{VREFINT\_DATA} - TS\_CAL30}{Slope} + 30$$

其中，TS_DATA 是 ADC 采样当前温度传感器输出的转换值；由于不知道当前 VDDA 的准确电平，因此这个转换值需要根据 VREFINT 的转换结果进行比例缩放；TS_CAL30 是芯片生产时在 30C+/-1C、VDDA=3.0V 的条件下进行温度定标的转换结果，这个数据保存在 flash 中。

Slope 表示温度传感器输出斜率，可以通过以下公式计算：

$$Slope = \frac{TS\_CAL85 - TS\_CAL30}{85 - 30}$$

其中 TS_CAL85 是 85C 下的温度定标值。

注意，并非所有芯片都有 85°C 和 30°C 两点定标，也可能只有 30°C 单点定标数据。这种情况下，可以使用温度传感器的典型斜率值来计算温度，典型斜率请参考芯片手册中的电气参数。

如果温度采样值只是用来做 RTC 温度补偿，则并不需要计算实际的温度值（-40~85 的十进制数），仅需要根据 $TS\_DATA \times \frac{VREFINT\_CAL}{VREFINT\_DATA}$ 的 12bit 结果（从物理原理来看这个结果就是 12bit），以 30C 为中心点进行地址查表即可。因为上式计算结果代表的是折算到 VDDA=3V 情况下温度传感器的 12 位输出结果，它与 TS_CAL30 的差值，即为偏离 30C 多少个 LSB，用这个信息作为地址对温度补偿校正表格查表，即可得到相应温度下的校正值。

使用温度传感器功能时，需要使能内部基准源的温度传感器输出，即同时置位 VREF_EN 和 PTAT_EN 寄存器，并且置位 VPTAT_BUFFER_EN 寄存器使能 PTAT 缓冲器，等待 5us 建立时间后，使能 ADC 对温度传感器通道进行采样。

温度传感器软件配置方法

软件使用温度传感器时，需按照如下步骤配置

- 置位 VREF_EN 寄存器，使能 VREF1p2 模块
- 置位 BUFFERCTRL.VREFPTAT_EN，使能 PTAT 输出
- 置位 BUFFERCTRL.VPTATBUFFER_EN，使能 PTAT 输出 BUFFER
- 等待 VREF 建立，通过查询 VREF_RDY 寄存器，或通过 VREF_IF 中断
- 使能 ADC 的 TSCH 通道
- 使能 ADC 开始转换
- 等待转换完成，读取结果，计算温度值

32.4.3 温度传感器的斜率和标定

温度传感器工作电压范围为 1.8~5.5V，温度测量范围不小于-40~+85C，跨度 125C。PTAT 输出斜率为 3.06mV/C，在 VDDA=5V 的情况下，分辨率为 1.22mV/LSB，即 2.51LSB/C；在 VDDA=3V 情况下，分辨率为 0.73mV/LSB，即 4.19 LSB/C。

电源电压	VPTAT slope	ADC mV/LSB	LSB/° C
1.8~5.5V	3.06mV/C	1.22@5V	2.51
		0.73@3V	4.19

举例来说，假设当前 ADC 采样温度传感器输出得到的转换值为 TS_DATA，根据以下步骤可以计算当前实际温度：

1) 由以下公式计算当前温度传感器输出的绝对电压值

$$\text{VPTAT} = \frac{VREFIN\_CAL \times TS\_DATA}{VREFINT\_DATA \times 4095} \times 3V$$

2) 由以下公式计算温度标定时（30° C）温度传感器输出的绝对电压值

$$\text{VPTAT\_30C} = \frac{TS\_CAL1}{4095} \times 3V$$

3) 根据温度传感器输出斜率计算当前绝对温度

$$\text{Temperature} = \frac{VPTAT - VPTAT\_30C}{3.06mV/C} + 30C$$

32.4.4 可编程采样时间

通过调整采样时间，可以适应不同输入信号源的内阻。通过 SMTS1 和 SMTS2 寄存器可以选择采样时间：

SMTSx	Sampling cycles
0000	4
0001	6
0010	9
0011	10
0100	16
0101	24
0110	32
0111	48
1000	96
1001	128
1010	192
1011	256
1100	384
others	Software control

表32-2ADC采样时间表

实际 ADC 的采样转换时间：tCONV = (Sampling Cycles + 12) * TADC_CLK

当 SMTS 寄存器配置为 1101/1110/1111 时，采样时间由软件控制。软件可以直接改写 sampt_b 寄存器，来自由控制任意长度的采样时间。软件对 sampt_b 的操作需要被同步到 ADCCLK 上。

软件控制采样时间时，ADC 在 START 置位后保持采样，直到软件对 SAMPT 寄存器写 1，4 个 ADCCLK 后 ADC 结束采样开始转换。SAMPT 寄存器在转换开始后自动清零。

ADC 采样时间主要由采样电容、被采样信号的输出阻抗、芯片内部输入通道阻抗和所需达到的采样精度共同决定。

下图是单端输入通道的电路结构示意：

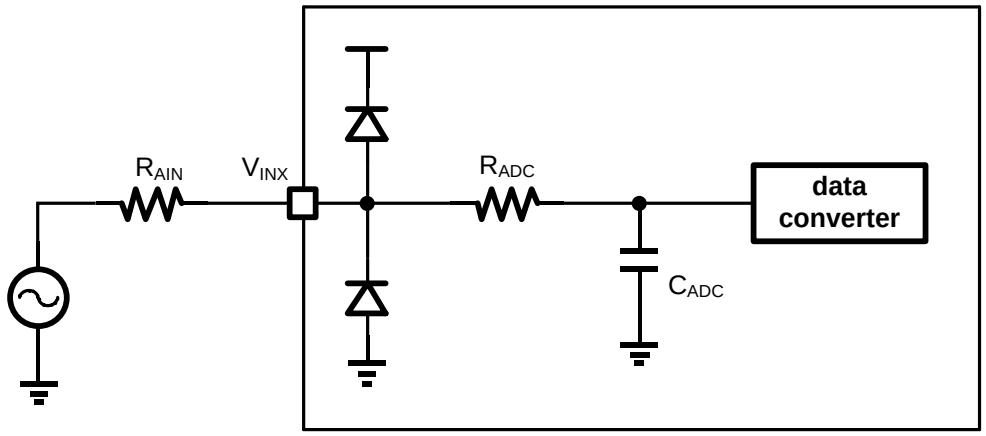


图 32-2 ADC 单端输入通道示意图

要求的采样时间可以根据下式估算：

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC}\right) \times C_{ADC}$$

其中，n=12，SA 表示容许的采样误差，比如 0.25 代表 1/4 LSB

应用中应根据芯片手册中的相关参数、以及系统参数，计算并确定可以接受的采样时间，并根据这个结果来配置 ADC 的工作时钟、采样周期等。

32.4.5 外部引脚控制的采样时间

除了寄存器控制采样时间之外，还可以通过外部引脚直接输入 sync_en 和 sampt_b 信号的方式来控制采样时间。此时 ADC 的 sync_en 和 sampt_b 信号直接来自于 GPIO 输入，内部寄存器控制无效。外部输入的控制信号会先经过 ADC 工作时钟同步。

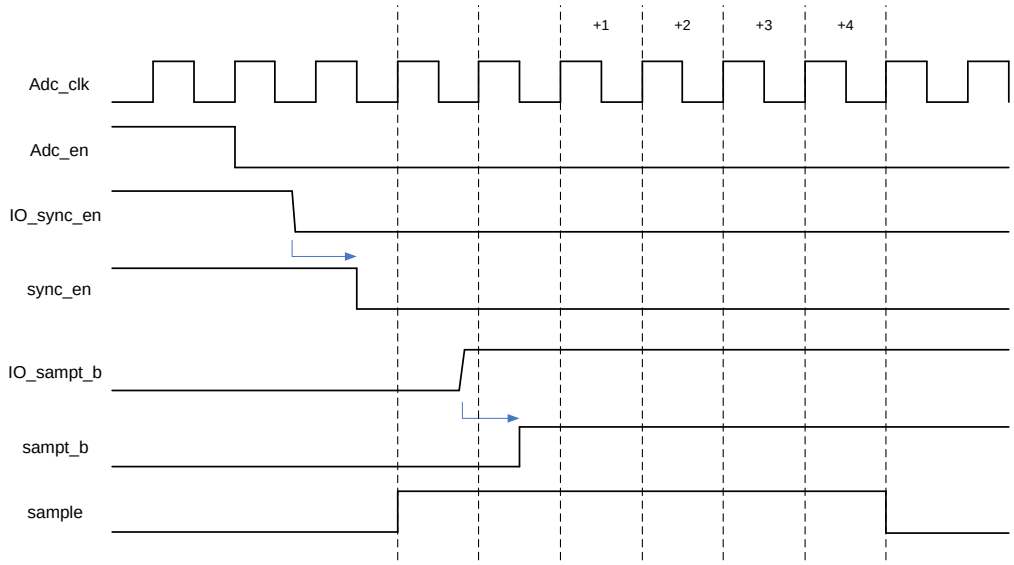


图 32-3 ADC 单端输入通道示意图

上图中外部输入的 IO_sampt_b 在 IO_sync_en 之后大约 2 个 ADC 时钟周期后拉高，所以实际内部采样时间被延长了 2cycle，为 6cycle。

注意：外部引脚触发控制采用 PB0 和 PB1 引脚。其中 PB0 的输入信号连接到 IO_sync_en，PB1 的输入信号连接到 IO_sampt_b。使用这一功能时，需要将 ADC_CR.EXSYNC 和 ADC_CR.EXSAMP 寄存器置位，并将 PB0 和 PB1 配置为 GPIO 输入。

32.4.6 转换模式

ADC 支持以下转换模式：

- 单次转换
  - 半自动触发（SEMI-AUTOMATIC）
  - 全自动触发（AUTOMATIC）
- 连续转换

转换启动可以由软件或事件触发，通过寄存器选择多个事件触发源。

单次转换模式下，有半自动触发和全自动触发两种模式。

全自动触发模式：软件或硬件触发事件启动 ADC 转换后，ADC 会顺序采样所有被使能的通道，单个通道采样完成后，EOC（End of Conversion）标志置位，所有通道采样完成后，EOS（End of Sequence）标志置位，本次转换结束。假设通道 0、3、5 被使能

- 1st 触发事件：通道 0、3、5 被顺序采样，过程中产生三次 EOC，最终产生 EOS
- 2nd 触发事件：重复上述过程

半自动触发模式：软件或硬件触发事件只会让 ADC 启动一次，转换一个使能通道。比如通道 0、3、5 被使能

- 1st 触发事件：通道 0 被采样，产生 EOC
- 2nd 触发事件：通道 3 被采样，产生 EOC
- 3rd 触发事件：通道 5 被采样，产生 EOC 和 EOS
- 4th 触发事件：通道 0 被采样，产生 EOC
- 5th 触发事件：通道 3 被采样，产生 EOC

  ……

连续转换模式：

触发事件到来后，所有使能通道被采样，并且 ADC 不会自动停止，而是循环采样，直到软件停止 ADC。

每个通道被采样后，数据保存在 ADC_DATA 寄存器中，软件要在下次转换前及时读走数据，或者通过 DMA 进行数据搬移。如果不能及时取走数据，将引起 Overrun，置位 overrun 标志，并可以发出中断。

32.4.7 转换触发

ADC 使能后，转换触发支持软件或硬件事件触发。

软件触发

软件通过置位 START 寄存器启动转换。

硬件触发

ADC 共有如下硬件触发源：RTC_TRGO、ATIM_TRGO、GPTIM0_TRGO、GPTIM1_TRGO、BSTIM_TRGO、比较器输出、及 2 个 GPIO 输入信号（PA8 和 PB9）；通过 IOTREN 寄存器，可以选择 IO 输入信号的上升沿、下降沿或上升下降沿触发转换。如果 ADC 正处于转换过程中，此时到来的触发信号会被忽略。

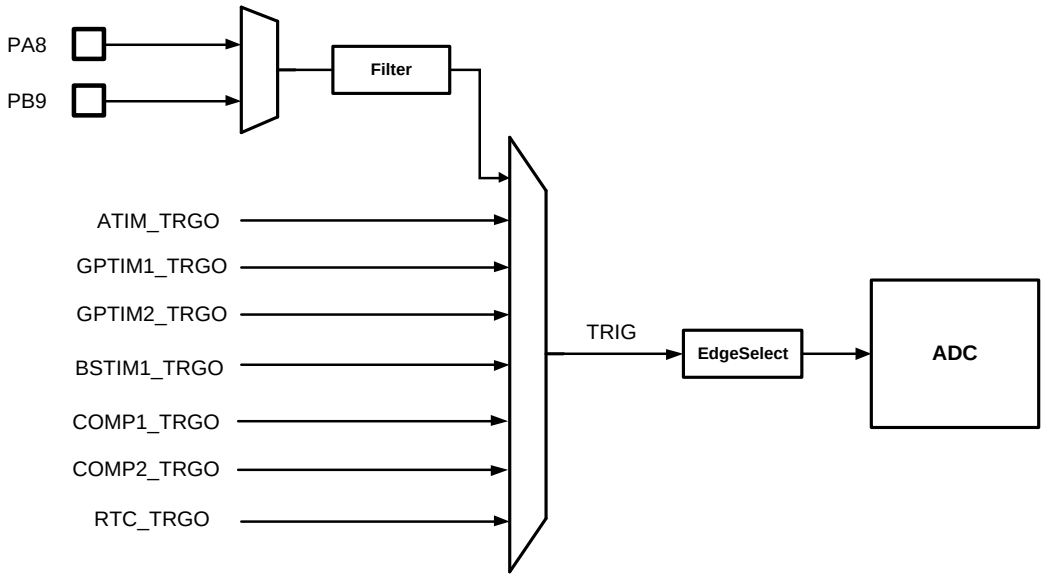


图 32-4 ADC 触发通道示意图

引脚输入的触发信号经过选择后输入到数字滤波和边沿检测模块。数字滤波实现原理与 IO 中断的数字滤波相同，即使用 APBCLK 连续采样三次输入信号，电平相同时才认为是合法信号。经过滤波的信号再经由边沿选择电路产生上升沿、下降沿或者双沿触发信号。

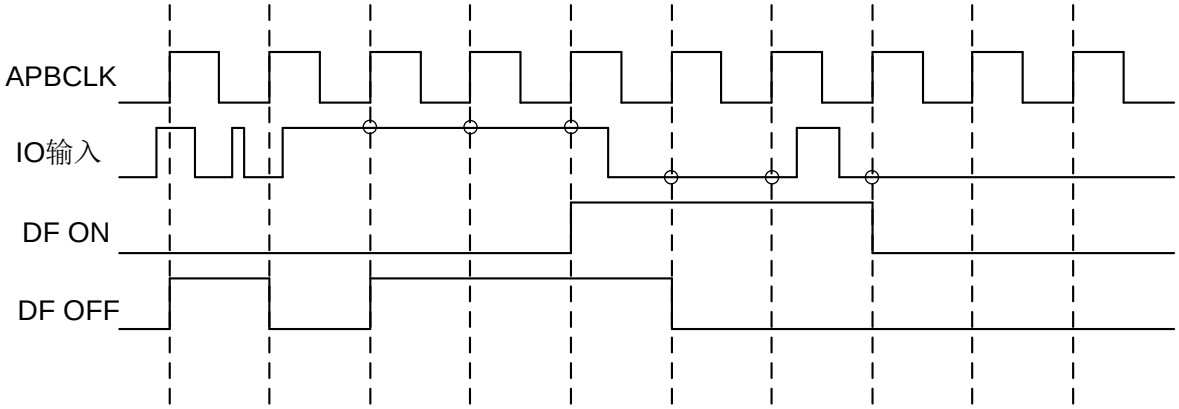


图 32-5 ADC 触发信号滤波

使用引脚输入信号触发 ADC 转换时，需进行如下配置：

- 将 PA8 或 PB9 配置为输入
- 设置 IOTRFEN 和 IOTREN 寄存器，配置滤波、触发边沿
- 配置 EXTS 寄存器，将触发源选为外部引脚输入
- 配置 ADC 工作时钟、采样时间、采样通道等
- 使能 ADC
- 指定 IO 上输入的特定电平变化将触发 ADC 转换

32.4.8 过采样和硬件平均

ADC 支持硬件过采样平均，可以在一定程度上提高分辨率。原理是对于低速输入信号，可以通过连续多次采样后求平均的方法提高 ENOB，过采样公式如下：

$$result = \frac{\sum_{n=1}^{N} CONVERSION_n}{M}$$

其中 N 是过采样倍数，可配置为 2/4/8/16/32/64/128/256，M 为结果右移位数，最大右移 8bit；由于每次转换结果为 12bit，最大 256 次累加得到的结果为 20bit，经过移位后可以得到 12~16bit 最终结果。ADC 输出结果最多只有 16bit，如果右移后结果超过 16bit，高位也会被丢弃。

在使能过采样的情况下，EOC 信号在 N 次连续采样后才置位，对于应用程序和 DMA 来说，感觉就好像只经过一次采样转换。

注意：当使能过采样时（OVSEN=1），必须使能自动等待模式（WAIT=1）。

32.4.9 ADC 工作时钟

ADC 采用双时钟结构，同时使用 APBCLK 和一个异步工作时钟 ADCCLK。

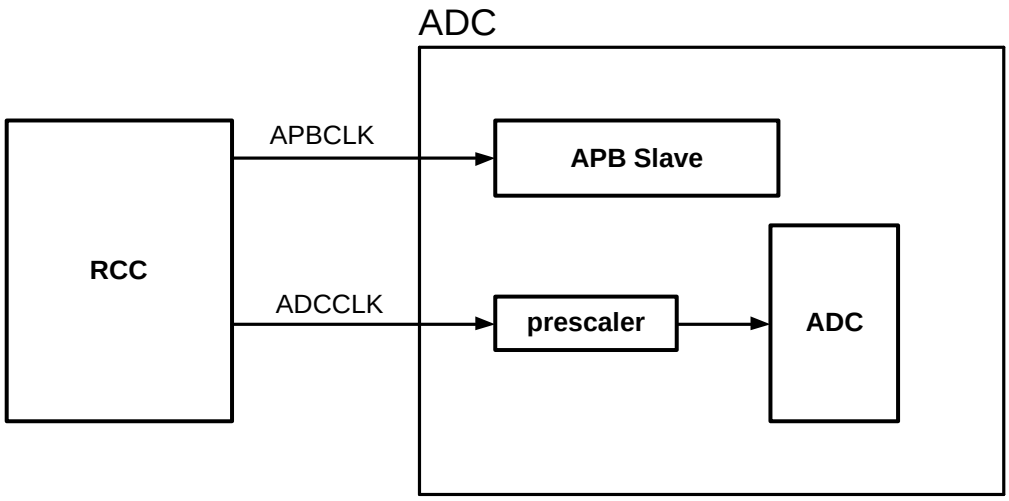


图 32-6 ADC 工作时钟

32.4.10 ADC 电源和基准电压

ADC 工作电源和基准电压为 VDDA。对于小封装型号，VDDA 和 VDD 合并打线，此时 ADC 的工作电源和基准电压即为 VDD。

需要注意的是，当系统方案中采用 5V 电源供电时，使用 ADC 前建议打开 ADC 电源检测功能，参见 13.8.1SVD 配置寄存器（SVD_CFGR）。

32.4.11 数据冲突和自动等待

每次转换完成后 EOC 标志会置位，软件或 DMA 读取 ADC_DATA 寄存器后会自动清除 EOC，也可以由软件写 1 清除。当 EOC 标志没有被清除的情况下，新的转换数据到来，就会导致 data overrun；有两种 overrun 模式：

OVRMOD=0：保持旧的数据，新数据丢弃

OVRMOD=1：写数据写入覆盖旧数据

注意：当使能过采样时（OVSEN=1），OVRMOD 寄存器无效，新的数据总是会覆盖旧数据。

当使用 DMA 时出现 overrun，则不会发起新的 DMA request，直到 OVR 标志被软件清零

ADC 控制器还支持自动等待，如果 WAIT 寄存器被软件置位，那么在 ADC_DATA 寄存器被读取之前，ADC 控制器不会发起新的转换；在等待状态中到来的硬件触发事件也会被忽略。WAIT 寄存器在 DMA 模式下也同样有效，即 DMA 没有读取上一次转换结果的情况下，ADC 控制器不会启动新

的转换。

下图是软件触发连续模式的情况下，使能了自动等待的示意图：

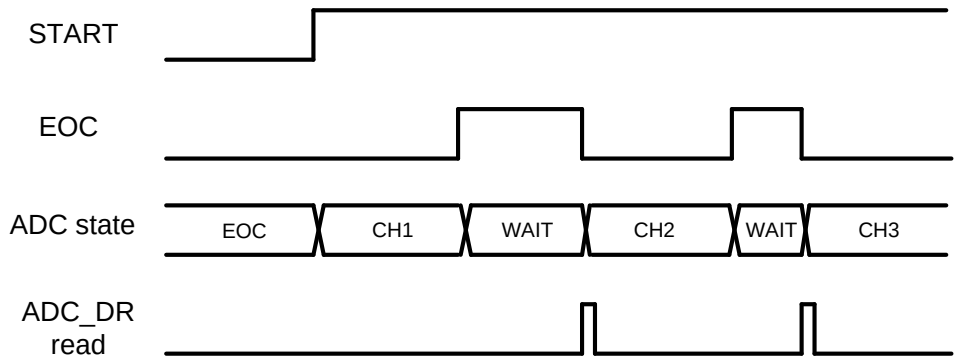


图 32-7 连续模式下的自动等待

32.4.12 DMA

在多通道转换或连续转换时，使用 DMA 进行转换结果搬移是高效的解决方案。在使能了 DMAEN 的情况下，当每次转换完成后（EOC），ADC controller 模块会产生一个 DMA 请求，通知 DMA 将数据寄存器中的结果搬运到指定的 SRAM 地址。ADC 的 DMA 接口支持单次模式和循环模式：

单次模式

转换完成后发起数据搬运，此过程会一直重复，直到软件配置的 DMA 传输长度完成，然后 ADC 控制器会自动停止转换（通过接收 DMA 的传输完成中断标志信号），关闭 ADC，不再向 DMA 发起请求。此模式主要用于对特定模拟信号进行一定长度的采样。

循环模式

与 DMA 的循环模式相配合，ADC 不断循环转换并发起 DMA 请求，直到软件停止转换。此模式可以用于处理连续不断模拟信号采样。ADC 转换完成信号可以被发送到 LPTIM 作为计数时钟，用于在循环模式下记录实际发生的转换次数。

在 DMA 使能情况下，如果发生 overrun，则 ADC 控制器不再发送 DMA 请求，直到 OVR 标志被清除。

注意，在单次和连续转换模式下，都可以支持 DMA 传输；DMA 传输长度以 EOC 的次数定义，而不是 EOS，即 DMA 只关心搬运多少次 ADC_DATA。

32.4.13 模拟窗口看门狗（AWD）

AWD 功能用于监视某个模拟输入通道或所有输入通道的输入信号电平是否处于寄存器设置的幅值范围之内。当 ADC 转换值高于 AWD_HT 或者低于 AWD_LT 时，都会置位中断标志寄存器。标志寄存器由软件写 1 清零。

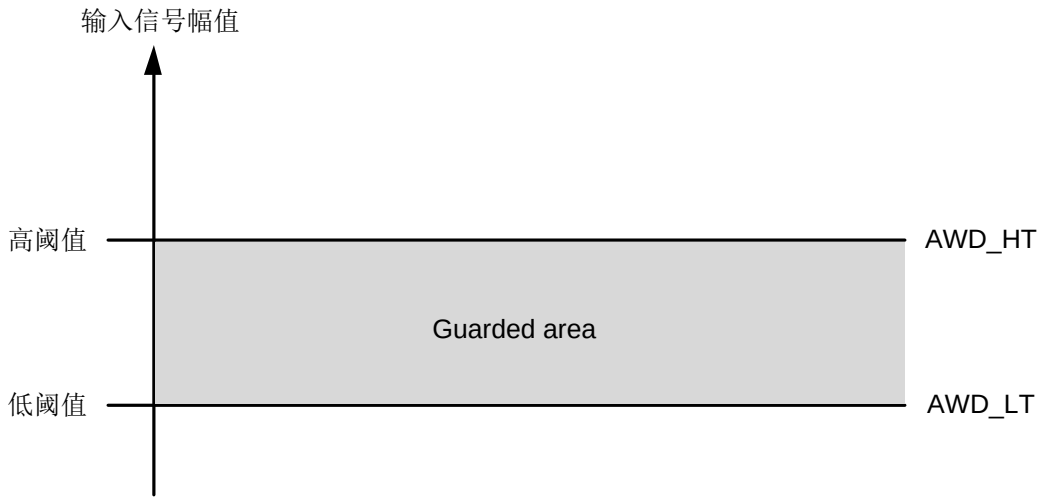


图 32-8 模拟看门狗

通过 AWDEN 寄存器使能模拟窗口看门狗功能，通过 AWDSC 寄存器配置单通道监视或全部通道监视。

32.5 低功耗模式

当芯片进入低功耗模式时，ADC 仍然允许工作。但是在低功耗模式下，芯片自动关闭了所有高速时钟源，所以 ADC 最高工作时钟仅为 RCMF，对应的最高采样率是 250Ksps。

32.6 寄存器

offset 地址	名称	符号
ADC(模块起始地址:0x4001AC00)		
0x00000000	ADC 中断和状态寄存器 （ADC Interrupt and Status Register）	ADC_ISR
0x00000004	ADC 中断使能寄存器 （ADC Interrupt Enable Register）	ADC_IER
0x00000008	ADC 控制寄存器 （ADC Control Register）	ADC_CR
0x0000000C	ADC 配置寄存器 （ADC Config Register）	ADC_CFGR
0x00000010	ADC 采样时间控制寄存器 （ADC Sampling Time Register）	ADC_SMTR
0x00000014	ADC 通道控制寄存器 （ADC Channel Enable Register）	ADC_CHER
0x00000018	ADC 数据寄存器 （ADC Data Register）	ADC_DR
0x0000001C	ADC 软件采样控制寄存器 （ADC Sampling Register）	ADC_SAMPT
0x00000020	ADC 模拟看门狗阈值寄存器 （ADC analog watchdog Threshold Register）	ADC_HLTR

32.6.1 ADC 中断和状态寄存器（ ADC_ISR）

名称	ADC_ISR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AH	AWD_UL	-	BUSY	OVR	EOS	EOC
位权限	U-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	AWD_AH	模拟看门狗超出上限中断标志 (Analog Watchdog Above High Threshold flag, write 1 to clear) 当采样值高于 AWD_HT 时，硬件置位，软件写 1 清零
5	AWD_UL	模拟看门狗低于下限中断标志 (Analog Watchdog Under Low Threshold flag, write 1 to clear) 当采样值低于 AWD_LT 时，硬件置位，软件写 1 清零
4	-	RFU：未实现，读为 0

位号	助记符	功能描述
3	BUSY	ADC 忙标志 (Busy flag,only read) 1：ADC 正在校准、采样或转换过程中 0：ADC 空闲
2	OVR	数据冲突标志，硬件置位，软件写 1 清零 (Over Run flag, write 1 to clear) 当 ADC_DATA 寄存器中的上一次转换结果还未被读取，新的转换结果又到来时，硬件置位 OVR 标志。 0：没有数据冲突 1：出现数据冲突
1	EOS	转换序列结束 (End Of Sequence flag, write 1 to clear) 所有使能通道都转换完成后，置位 EOS，软件写 1 清零。
0	EOC	单次转换结束 (End Of Conversion flag,write 1 to clear) 每个通道转换完成后，置位 EOC，软件写 1 清零。

32.6.2 ADC 中断使能寄存器（ ADC_IER）

名称	ADC_IER							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AHIE	AWD_ULIE	-		OVRIE	EOSIE	EOCIE
位权限	U-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6	AWD_AHIE	模拟看门狗采样值高于上限中断使能，1 有效 (Analog Watchdog Above High Threshold Interrupt Enable)
5	AWD_ULIE	模拟看门狗采样值低于上限中断使能，1 有效 (Analog Watchdog Under Low Threshold Interrupt Enable)
4:3	-	RFU：未实现，读为 0
2	OVRIE	数据冲突中断使能寄存器 (Over Run Interrupt Enable) 0：禁止数据冲突中断 1：允许数据冲突中断
1	EOSIE	转换序列结束中断使能寄存器 (End Of Sequence Interrupt Enable) 0：禁止 EOS 中断 1：允许 EOS 中断
0	EOCIE	单次转换结束中断使能寄存器 (End Of Conversion Interrupt Enable) 0：禁止 EOC 中断

位号	助记符	功能描述
		1：允许 EOC 中断

32.6.3 ADC 控制寄存器（ADC_CR）

名称	ADC_CR							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						EXSAMP	EXSYNC
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						START	ADEN
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	EXSAMP	外部引脚控制采样时间 (External Sample time control) 1：由 GPIO 输入信号来控制 ADC 采样时间 0：由寄存器控制 ADC 采样时间
8	EXSYNC	外部引脚控制采样使能 (External Synchronization enable) 1：由 GPIO 输入信号启动 ADC 采样 0：由 START 寄存器启动 ADC 采样
7:2	-	RFU：未实现，读为 0
1	START	ADC 启动转换寄存器，软件写 1 启动，硬件自动清零。
0	ADEN	ADC 使能寄存器。(ADC Enable) 在启动转换前先要置位 ADEN。 0：关闭 ADC 1：使能 ADC

32.6.4 ADC 配置寄存器（ADC_CFGR）

名称	ADC_CFGR							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		AWDCH				AWDSC	AWDEN
位权限	U-0		R/W-0000				R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OVSS				OVSR			OVSEN
位权限	R/W-0000				R/W-000			R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	IOTRFEN	TRGCFG		SEMI	WAIT	CONT	OVRM

位权限	U-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTS				-	SCANDIR	DMACFG	DMAEN
位权限	R/W-0000				U-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:30	-	RFU：未实现，读为 0
29:26	AWDCH	模拟窗口看门狗监视通道选择，仅在 AWDSC=1 时有效 (Analog Watchdog Channel Select) 0000：AWD 监视 ADC_IN0 0001：AWD 监视 ADC_IN1 0010：AWD 监视 ADC_IN2 0011：AWD 监视 ADC_IN3 0100：AWD 监视 ADC_IN4 0101：AWD 监视 ADC_IN5 0110：AWD 监视 ADC_IN6 0111：AWD 监视 ADC_IN7 1000：AWD 监视 ADC_IN8 1001：AWD 监视 ADC_IN9 1010：AWD 监视 ADC_IN10 1011：AWD 监视 ADC_IN11 其他：保留
25	AWDSC	模拟窗口看门狗单通道或全通道选择 (Analog Watchdog Single Channel mode) 0：AWD 监视所有被使能的外部输入通道 1：AWD 监视 AWDCH 指定的单个通道
24	AWDEN	模拟窗口看门狗使能寄存器 (Analog Watchdog Enable) 0：关闭 AWD 1：使能 AWD 仅能在 START=0 的情况下使能 AWD
23:20	OVSS	过采样移位控制寄存器 (Oversampling Shift) 0000：不移位 0001：右移 1bit 0010：右移 2bit 0011：右移 3bit 0100：右移 4bit 0101：右移 5bit 0110：右移 6bit 0111：右移 7bit 1000：右移 8bit Others：RFU
19:17	OVSR	过采样率控制 (Oversampling Ratio) 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x

位号	助记符	功能描述
16	OVSEN	过采样使能 (Oversampling Enable) 0：禁止过采样 1：使能过采样
15	-	RFU：未实现，读为 0
14	IOTRFEN	引脚触发信号数字滤波使能 (GPIO Trigger Filter Enable) 0：禁止数字滤波 1：使能数字滤波
13:12	TRGCFG	触发信号使能和极性选择 (Trigger Config) 00：禁止触发 01：上升沿触发 10：下降沿触发 11：上升、下降沿都触发
11	SEMI	单次转换半自动模式 (Semi-automatic)，仅在单次转换（CONT=0）时有效，参见“转换模式”章节 0：自动模式 1：半自动模式
10	WAIT	等待模式控制 (wait mode) 0：无等待，如果上次转换数据没有及时读取，则可能出现 Overrun 1：等待模式，在上次转换数据被读取前，不会启动下一次转换
9	CONT	连续转换模式使能 (Continuous mode) 0：单次转换 1：连续转换
8	OVRM	Overrun 模式控制 (Overrun mode) 0：当 overrun 发生时，保持上次数据，丢弃本次转换值 1：当 overrun 发生时，覆盖上次数据 *注：当 OVSEN=1 的情况下，OVRM 配置不起作用，过采样平均后的新数据总是会覆盖上次数据，软件应注意响应时间，避免 overrun*
7:4	EXTS	硬件触发源选择 (External trigger select) 0000：PA8 0001：PB9 0010：RFU 0011：ATIM_TRGO 0100：GPTIM0_TRGO 0101：GPTIM1_TRGO 0110：RFU 0111：RTC_TRGO 1000：BSTIM_TRGO 1001：RFU 1010：COMP1_TRGO 1011：COMP2_TRGO Others：RFU
3	-	ADC 低功耗模式使能，软件可以配置 (Low Power Mode) 1：使能 ADC 低功耗模式，最大工作时钟频率 4MHz，最高采样率 250Ksps 0：正常模式
2	SCANDIR	通道扫描顺序控制 (Scan Direction)（共 16 个通道，实际只会

位号	助记符	功能描述
		采样被使能的通道） 0：前向扫描，ADC_IN0->ADC_IN11->REF->TS->OPA1->OPA2 1：反向扫描，OPA2->OPA1->TS->REF->ADC_IN11->ADC_IN0
1	DMACFG	DMA 模式控制 (DMA Config) 0：单次模式 1：循环模式
0	DMAEN	DMA 使能 (DMA Enable) 0：禁止 DMA 1：使能 DMA

32.6.5 ADC 采样时间控制寄存器（ADC_SMTR）

名称	ADC_SMTR							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CHCG			
位权限	U-0				R/W-1000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SMTS2				SMTS1			
位权限	R/W-0000				R/W-0000			

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:8	CHCG	ADC 采样通道切换等待时间，在当前通道采样周期完成后，等待 CHCG 时间（CHCG*ADC 工作时钟周期），再切换到下一个采样通道(Channel Clock Gating) 0000, 0001, 0010：2*T_{ADCLK} 0011：3*T_{ADCLK} 0100：4*T_{ADCLK} 0101：5*T_{ADCLK} 0110：6*T_{ADCLK} 0111：7*T_{ADCLK} 1000：8*T_{ADCLK} 1001：9*T_{ADCLK} 1010：10*T_{ADCLK} 1011~1111：11*T_{ADCLK}
7:4	SMTS2	慢速通道采样时间控制（*ADC 工作时钟周期），用于配置 ADC_IN8/9/10/11 四个外部通道、以及 VREF1p2、TS、OPA 通道的采样时间 (Sampling Time Select 2) 0000：4 0001：6

位号	助记符	功能描述
		0010：9 0011：10 0100：16 0101：24 0110：32 0111：48 1000：96 1001：128 1010：192 1011：256 1100：384 Others：软件控制
3:0	SMTS1	快速通道采样时间控制（*ADC 工作时钟周期），用于配置 ADC_IN0~7 八个外部通道的采样时间 (Sampling Time Select 1) 0000：4 0001：6 0010：9 0011：10 0100：16 0101：24 0110：32 0111：48 1000：96 1001：128 1010：192 1011：256 1100：384 1101/1110/1111：软件控制

32.6.6 ADC 通道控制寄存器（ADC_CHER）

名称	ADC_CHER							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				OPA2CH	OPA1CH	REFCH	TSCH
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				ECH11	ECH10	ECH9	ECH8
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19	OPA2CH	OPA2 输出测量，写 1 使能 (OPA2 channel enable)
18	OPA1CH	OPA1 输出测量，写 1 使能 (OPA1 channel enable)
17	REFCH	内部基准电压测量通道，写 1 使能 (VREF channel enable)
16	TSCH	温度传感器测量通道，写 1 使能 (TempSensor channel enable)
15:12	-	RFU：未实现，读为 0
11	ECH11	ADC_IN11 测量通道，写 1 使能(External Channel Enable)
10	ECH10	ADC_IN10 测量通道，写 1 使能(External Channel Enable)
9	ECH9	ADC_IN9 测量通道，写 1 使能(External Channel Enable)
8	ECH8	ADC_IN8 测量通道，写 1 使能(External Channel Enable)
7	ECH7	ADC_IN7 测量通道，写 1 使能(External Channel Enable)
6	ECH6	ADC_IN6 测量通道，写 1 使能(External Channel Enable)
5	ECH5	ADC_IN5 测量通道，写 1 使能(External Channel Enable)
4	ECH4	ADC_IN4 测量通道，写 1 使能(External Channel Enable)
3	ECH3	ADC_IN3 测量通道，写 1 使能(External Channel Enable)
2	ECH2	ADC_IN2 测量通道，写 1 使能(External Channel Enable)
1	ECH1	ADC_IN1 测量通道，写 1 使能(External Channel Enable)
0	ECH0	ADC_IN0 测量通道，写 1 使能(External Channel Enable)

32.6.7 ADC 数据寄存器（ADC_DR）

名称	ADC_DR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DATA	ADC 转换结果(ADC conversion data) 在没有使能过采样平均的情况下，结果为低 12bit；在使能过采样平均的情况下，结果为 12~16bit

32.6.8 ADC 软件采样控制寄存器（ADC_SAMPT）

名称	ADC_SAMPT							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							SAMPT_S
位权限	U-0							0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	SAMPT_S	软件控制采样信号，仅在 SMTSx=1101/1110/1111 时有效 (Sample time software control) 0：ADC 采样 1：ADC 停止采样

32.6.9 ADC 模拟看门狗阈值寄存器（ADC_HLTR）

名称	ADC_HLTR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				AWD_HT[11:8]			
位权限	U-0				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AWD_HT[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				AWD_LT[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AWD_LT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:28	-	RFU：未实现，读为 0
27:16	AWD_HT	AWD 监视高阈值 (Analog Watchdog High Threshold)
15:12	-	RFU：未实现，读为 0
11:0	AWD_LT	AWD 监视低阈值 (Analog Watchdog Low Threshold)

33 USB 全速设备

33.1 概述

FM33LC0XX集成USB2.0 FS device，包含片上无晶振PHY和MAC控制器。

33.2 主要特性

- USB2.0 full-speed, crystal-less mode
- 支持1个双向控制端点，2个IN端点，2个OUT端点，类型可配置
- 96*35bits Packet RAM
- RxFIFO：48words，IN EP dedicated TxFIFO：16words*3
- CRC生成/检查，NRZI编解码，自动位填充
- 支持Suspend/Resume操作
- 集成D+D-上拉下拉电阻
- 支持软件断开连接

33.3 电源管理

FM33LC0XX的USB设备为self-powered架构，即不从VBUS直接取电工作。要保证USB正常工作，芯片电源必须是3.0~3.6V，否则可能导致主机枚举错误，无法正常通信。

33.4 系统总线架构

USB外设在系统总线上的架构如下图所示：

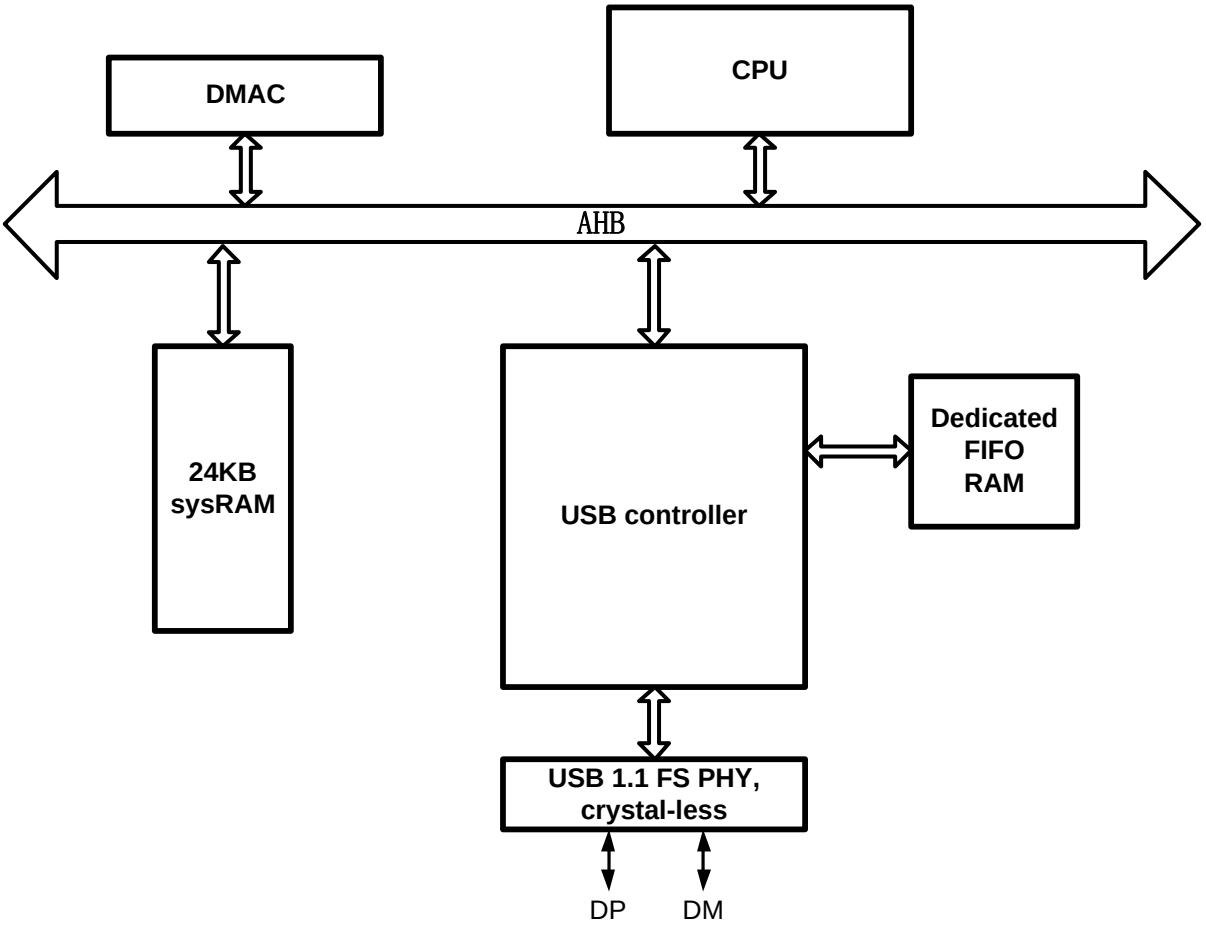


图 33-1USB 外设在总线上的位置

33.5 功能框图

33.6 USB Controller

33.6.1 概述

一些概念定义：

- Periodic Endpoint：isochronous或者interrupt传输定义为periodic传输，对应的端点称为periodic端点
- Non-Periodic Endpoint：bulk或control传输定义为non-periodic传输，对应的端点称为non-periodic端点
- 每个端点有专用的TxFIFO控制器
- 所有的端点共享一个RxFIFO控制器
- Periodic端点主要是要满足高带宽传输需求，并不代表non-periodic端点不能用于isochronou或interrupt传输

FM33LC0XX支持1个控制端点，2个IN端点（类型可配置）和2个OUT端点（类型可配置）。

FM33LC0XX可以支持BULK、Interrupt、Isochronous传输，Bulk和Interrupt两种传输类型的最大包长都是64字节，Isochronous最大包长1023字节。

33.6.1.1 Device 架构

FM33LC0XX的USB为device only架构，只支持USB设备角色，不支持Host或OTG。

Device内部地址offset按下图分配：

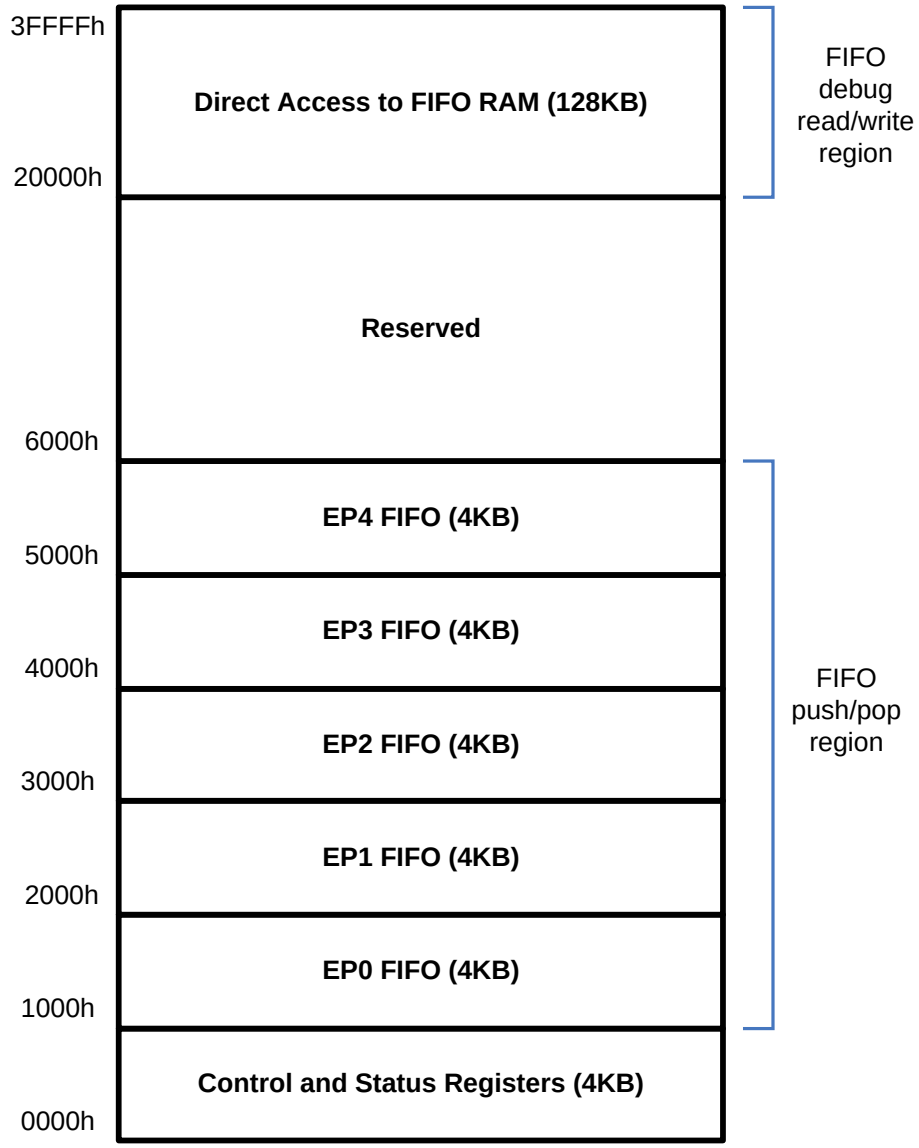


图 33-2USB 外设内部地址分配

每个IN端点有独立的TxFIFO分别映射到不同SRAM地址中，所有OUT端点的数据都由同一个RxFIFO映射到同一段SRAM地址中。其中每个IN端点TxFIFO深度是16words，所有OUT端点共享的RxFIFO深度是48words，因为总共实现了3个IN端点（Control*1，其他2个），所以总的FIFO存储深度是16*3+48=96words。

AHB对USB控制器所有寄存器（CSR）的访问都是32bit的。

Dedicated Transmit Data FIFO

每个IN端点实现了专用的TxFIFO，其Packet RAM空间也互相独立，不会互相影响，软件编程比较简单。AHB或者DMA通过对各个EP的TxFIFO地址段读写，实现对FIFO数据的PUSH和POP操作。写操作对应TxFIFO PUSH，读操作对应RxFIFO POP。

Receive Data FIFO

所有OUT端点共享一个接收FIFO。

- AHB slave或DMA接口单元从接收FIFO读取数据，MAC向发接收FIFO写入数据
- 这个FIFO可以保存多个属于不同端点的接收数据包
- SETUP包也写入这个接收FIFO，并且会预留空间
- 每个收到的数据包都带有一个状态字，包含数据包字节数、包状态、端点编号
- 接收FIFO是35bit宽
  - ✧ Bit[34:32]：RFU
  - ✧ Bit[31:0]：数据
  - ✧ Status字
    - ■ Bit[24:21]：frame number的最低4bit
    - ■ Bit[20:17]：包状态
    - ■ Bit[16:15]：PID
    - ■ Bit[14:4]：字节数
    - ■ Bit[3:0]：端点编号

33.7 USB device 的时钟和复位

33.7.1 系统时钟

USB device有如下工作时钟：

- HCLK：AHB总线时钟
- PHY_CLK：USB PHY输出的48Mhz时钟，供控制器使用

软件操作USB寄存器之前，必须置位CMU中的USB_PCE寄存器，打开USB寄存器操作时钟。

芯片能够通过解析USB总线上Host下发的数据帧，完成对PHY_CLK的自校准，当CLK_RDY寄存器置位时，表示自校准完成，PHY_CLK误差符合USB规范要求，并且能够实时跟踪1ms定时间隔的SoF包实现频率稳定。

33.7.2 系统复位

USB外设的复位包括：USB控制器复位、PHY复位、PHY内建工作时钟（Built-in Clock）复位。

通过清除RMU模块中的USBRST寄存器，能够撤销USB控制器复位。

通过置位PHY控制寄存器中的PHY_PONRST_B寄存器，撤销USB PHY的复位。

通过置位PHY控制寄存器中的NONCRY_RSTB寄存器，撤销PHY内建工作时钟的复位。

33.8 VBUS 接入唤醒

当MCU休眠时，VBUS接入事件需要唤醒MCU。这个唤醒功能可以通过VBUS直接接到5V tolerant IO上产生上升沿中断的方式实现。

VBUS接入唤醒使用PB12实现，此引脚设计为5V tolerant，能够接受高于电源电压的信号电平输入，并且具有WKUP唤醒功能。

33.9 中断层级

下表显示了USB中断产生和屏蔽的层级结构。

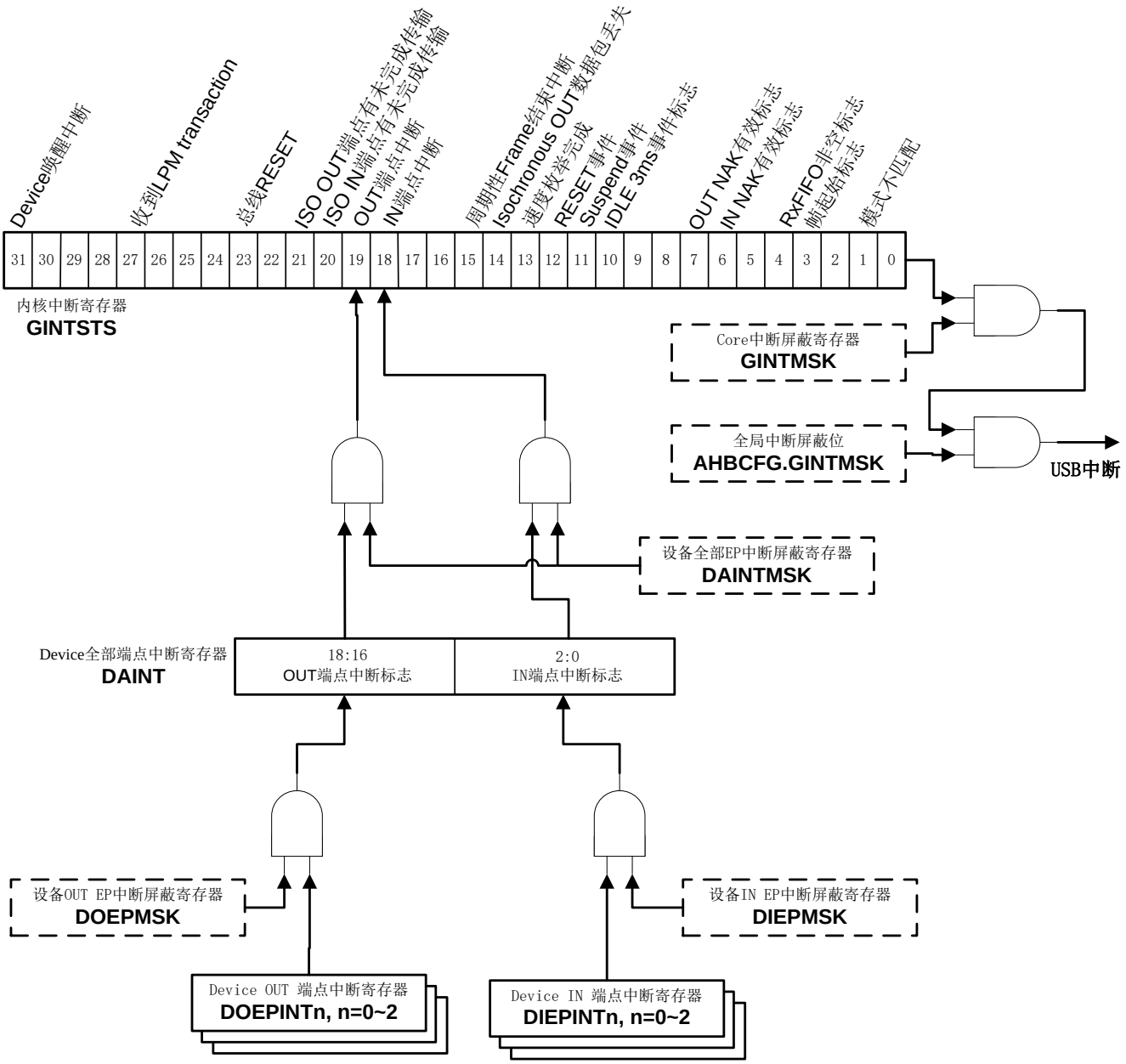


图 33-3USB 外设中断示意图

33.10 寄存器

offset 地址	名称	符号
USB 控制器寄存器(模块起始地址：0x50000000)		
0x00000008	AHB 配置寄存器 （USB AHB Global Config Register）	USB_GAHBCFG
0x0000000C	USB 全局配置寄存器 （USB Global Config Register）	USB_GUSBCFG
0x00000010	复位控制寄存器 （USB Global Reset Control Register）	USB_GRSTCTL
0x00000014	中断标志寄存器 （USB Global Interrupt Status Register）	USB_GINTSTS
0x00000018	中断屏蔽寄存器 （USB Global Interrupt Mask Register）	USB_GINTMSK
0x0000001C	接收状态 debug 读寄存器 （USB Receive Status Debug Read Register）	USB_GRXSTSR
0x00000020	接收状态读和 POP 寄存器 （USB Receive Status and Pop Register）	USB_GRXSTSP
0x00000024	RxFIFO size 寄存器 （USB Receive FIFO size Register）	USB_GRXFSIZ
0x00000028	Non-Periodic TxFIFO size 寄存器 （USB Non-Periodic Transmit FIFO size Register）	USB_GNPTXFSIZ
0x00000054	LPM 配置寄存器 （USB Low-Power-Mode config Register）	USB_GLPMCFG
0x00000800	Device 配置寄存器 （USB Device Config Register）	USB_DCFG
0x00000804	Device 控制寄存器 （USB Device Control Register）	USB_DCTL
0x00000808	Device 状态寄存器 （USB Device Status Register）	USB_DSTS
0x00000810	Device IN 端点通用中断屏蔽寄存器 （USB Device In Endpoint Interrupt Mask Register）	USB_DIEPMSK
0x00000814	Device OUT 端点通用中断屏蔽寄存器 （USB Device OUT Endpoint Interrupt Mask Registe）	USB_DOEPMSK
0x00000818	Device 全部端点中断寄存器 （USB Device All Endpoint Interrupt Register）	USB_DAINT
0x0000081C	Device 全部端点中断屏蔽寄存器 （USB Device All Endpoint Interrupt Mask Register）	USB_DAINTMSK
0x00000834	Device IN 端点 FIFO 空中断屏蔽寄存器 （USB Device IN endpoint FIFO empty interrupt Mask Register）	USB_DIEPEMPMSK
0x00000900	Device Control IN 端点 0 控制寄存器 （USB Device In Endpoint 0 Control Register）	USB_DIEPCTL0
0x00000908	Device Control IN 端点 0 中断寄存器 （USB Device In Endpoint 0 Interrupt Register）	USB_DIEPINT0
0x00000910	Device Control IN 端点 0 传输长度寄存器 （USB Device IN Endpoint 0 Transfer Size Register）	USB_DIEPTSIZ0

offset 地址	名称	符号
0x00000918	Device Control IN 端点 0 发送 FIFO 状态寄存器（USB Device In Endpoint 0 Tranmit FIFO Status Register）	USB_DTXFSTS0
0x00000920	Device IN 端点 1 控制寄存器（USB Device In Endpoint 1 Control Register）	USB_DIEPCTL1
0x00000928	Device IN 端点 1 中断寄存器（USB Device In Endpoint 1 Interrupt Register）	USB_DIEPINT1
0x00000930	Device IN 端点 1 传输长度寄存器（USB Device IN Endpoint 1 Transfer Size Register）	USB_DIEPTSIZ1
0x00000938	Device IN 端点 1 发送 FIFO 状态寄存器（USB Device In Endpoint 1 Tranmit FIFO Status Register）	USB_DTXFSTS1
0x00000940	Device IN 端点 2 控制寄存器（USB Device In Endpoint 2 Control Register）	USB_DIEPCTL2
0x00000948	Device IN 端点 2 中断寄存器（USB Device In Endpoint 2 Interrupt Register）	USB_DIEPINT2
0x00000950	Device IN 端点 2 传输长度寄存器（USB Device IN Endpoint 2 Transfer Size Register）	USB_DIEPTSIZ2
0x00000958	Device IN 端点 1 发送 FIFO 状态寄存器（USB Device In Endpoint 2 Tranmit FIFO Status Register）	USB_DTXFSTS2
0x00000B00	Device Control OUT 端点 0 控制寄存器（USB Device Out Endpoint 0 Control Register）	USB_DOEPCTL0
0x00000B08	Device Control OUT 端点 0 中断寄存器（USB Device Out Endpoint 0 Interrupt Register）	USB_DOEPINT0
0x00000B10	Device OUT 端点 0 传输长度寄存器（USB Device Out Endpoint 0 Transfer Size Register）	USB_DOEPTSIZ0
0x00000B20	Device OUT 端点 1 控制寄存器（USB Device Out Endpoint 1 Control Register）	USB_DOEPCTL1
0x00000B28	Device OUT 端点 1 中断寄存器（USB Device Out Endpoint 1 Interrupt Register）	USB_DOEPINT1
0x00000B30	Device OUT 端点 1 传输长度寄存器（USB Device Out Endpoint 1 Transfer Size Register）	USB_DOEPTSIZ1
0x00000B40	Device OUT 端点 2 控制寄存器（USB Device Out Endpoint 2 Control Register）	USB_DOEPCTL2
0x00000B48	Device OUT 端点 2 中断寄存器（USB Device Out Endpoint 2 Interrupt Register）	USB_DOEPINT2
0x00000B50	Device OUT 端点 2 传输长度寄存器（USB Device Out Endpoint 2 Transfer Size Register）	USB_DOEPTSIZ2
0x00000E00	功耗控制寄存器（USB Power Control Global Control Register）	USB_PCGCCTL

33.10.1 AHB 配置寄存器（USB_GAHBCFG）

名称	USB_GAHBCFG							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NPTxFEL	-						GINTMSK
位权限	R/W-0	U-0						R/W-0

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7	**NPTxFEL**	Non-Periodic TxFIFO 空水平线 (Non-Periodic TxFIFO Empty Level) 0：GINTSTS.NPTxFEmp 在 Non-Periodic 发送 FIFO 半空时置位 1：GINTSTS.NPTxFEmp 在 Non-Periodic 发送 FIFO 全空时置位
6:1	-	RFU：未实现，读为 0
0	**GINTMSK**	全局中断屏蔽位 (Global Interrupt Mask) 0：屏蔽所有中断 1：允许中断

33.10.2 USB 全局配置寄存器（USB_GUSBCFG）

名称	USB_GUSBCFG							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		USBTrT				-	
位权限	U-0		R/W-0101				U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					ToutCal		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31:14	-	RFU：未实现，读为 0
13:10	**USBTrT**	(USB turn-around time) 以 PHY 时钟数为单位设置总线周转时间。
9:3	-	RFU：未实现，读为 0
2:0	**ToutCal**	Full Speed 超时校准(Timeout Calibration) PHY 引入的额外延迟包括应用程序在该字段中设置的 PHY 时钟数，以及模块的全速数据包间超时间隔。不同 PHY 引入的延迟对数据线状态的影响是不同的。全速操作的 USB 标准超时值为 16 到 18 个位时间。应用程序必须根据枚举速度编程该字段。每个 PHY 时钟增加的位时间数为 0.25 个位时间。

33.10.3 复位控制寄存器（USB_GRSTCTL）

名称	USB_GRSTCTL							
Offset	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AHBIdle	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					TxFNum[4:2]		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TxFNum[1:0]		TxFFlsh	RxFFlsh	-		HSftRst	CSftRst
位权限	R/W-00		R/W-0	R/W-0	U-0		R/W-0	R/W-0

位号	助记符	功能描述
31	**AHBIdle**	AHB 主器件空闲 (AHB device Idle)
30:11	-	RFU：未实现，读为 0
10:6	**TxFNum**	指定被 TxFIFO Flush 寄存器清零的发送 FIFO 编号(TxFIFO Flush Number)
5	**TxFFlsh**	发送 FIFO 清零寄存器，软件写 1 清零 FIFO，清除完成后硬件自动清零此寄存器；软件应等待此寄存器被硬件清零后再执行其他操作。通常需要等待 8 个时钟周期。(TxFIFO Flush)
4	**RxFFlsh**	接收 FIFO 清零寄存器，软件写 1 清零 FIFO，清除完成后硬件自动清零此寄存器；软件应等待此寄存器被硬件清零后再执行其他操作。通常需要等待 8 个时钟周期。(RxFIFO Flush)
3:2	-	RFU：未实现，读为 0
1	**HSftRst**	AHBCLK 时钟域软复位 (AHB clock domain soft reset) FIFO 不会被清零，中断标志位不会被清零
0	**CSftRst**	PHY_CLK 时钟域软复位 (PHY clock domain soft reset) 按如下所述复位 HCLK 和 PHY 时钟域： ● 除下述各位外，清零各个中断和所有 CSR 寄存器位： —PCGCCTL.GATEHCLK 位 —PCGCCTL.STOPPCLK 位 —DCFG.DSPD 位

位号	助记符	功能描述
		● 将所有模块状态机（AHB 从器件除外）复位至空闲状态，并清空所有发送 FIFO 和接收 FIFO。 ● 在 AHB 传输的最后数据阶段结束后，尽快终止 AHB 主器件上的所有事务。立即终止 USB 上的所有事务。 应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。该位一旦清零，软件必须等待至少 3 个 PHY 时钟后才可以访问 PHY 域（同步延迟）。软件还必须在确定该寄存器中的位 31 置 1（AHB 主器件空闲）后方可开始运行。

33.10.4 中断标志寄存器（USB_GINTSTS）

名称	USB_GINTSTS							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WkUpINT	-			LPM_INT	-		
位权限	R/W-0	U-0			R/W-0	U-0		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ResetDet	-	incomplSOOUT	incomplSOIN	OEPINT	IEPINT	-	
位权限	R/W-0	U-0	R/W-0	R/W-0	R-0	R-0	U-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EOPF	ISOODrop	EnumDone	USBRst	USBSusp	ErlySusp	-	-
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GOUTNakEff	GINNakEff	-	RxFNEmp	SoF	-	ModeMis	CurMod
位权限	R-0	R-0	U-0	R-0	R/W-0	U-0	R/W-0	R-0

位号	助记符	功能描述
31	**WkUpINT**	Device 唤醒中断，当 Device 在 Suspend 模式下收到 Resume 信号时置位，软件写 1 清零 (Device Wakeup Interrupt flag, write 1 to clear)
30:28	**-**	RFU：未实现，读为 0
27	**LPM_INT**	Device 收到 LPM transaction 时置位，软件写 1 清零 (Device LPM transaction Interrupt flag, write 1 to clear)
26:24	**-**	RFU：未实现，读为 0
23	**ResetDet**	当 Device 在 Suspend 模式下检测到总线 RESET 时置位，软件写 1 清零 (Device Reset Detect flag, write 1 to clear)
22	**-**	RFU：未实现，读为 0
21	**incomplSOOUT**	当前 microframe 中至少有一个 ISO OUT 端点没有完成传输该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。 (ISO OUT transfer is incomplete flag)
20	**incomplSOIN**	当前 microframe 中至少有一个 ISO IN 端点没有完成传输该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触

位号	助记符	功能描述
		发。 (ISO IN transfer is incomplete flag)
19	**OEPINT**	OUT 端点中断 (OUT endpoint interrupt flag) 当某个 OUT 端点有中断待处理时，此寄存器置位 软件需要读取 DAINT 寄存器来判断出现中断的 OUT 端点，并读相应的 DOEPINTn 寄存器来判断中断原因。软件清除 DOEPINTn 中的中断标志寄存器后，此 bit 自动清零。
18	**IEPINT**	IN 端点中断 (IN endpoint interrupt flag) 当某个 IN 端点有中断待处理时，此寄存器置位 软件需要读取 DAINT 寄存器来判断出现中断的 IN 端点，并读相应的 DIEPINTn 寄存器来判断中断原因。软件清除 DIEPINTn 中的中断标志寄存器后，此 bit 自动清零。
17:16	**-**	RFU：未实现，读为 0
15	**EOPF**	周期性 Frame 结束中断，软件写 1 清零 (End of Periodic Frame flag, write 1 to clear) DCFG.PerFrInt 寄存器定义的周期 frame 间隔已经达到
14	**ISOODrop**	Isochronous OUT 数据包丢失中断，软件写 1 清零 (ISO OUT packet dropped flag, write 1 to clear) MAC 控制器试图向 RxFIFO 写入 isochronous OUT packet，但是 FIFO 没有足够空间容纳一个最大包长
13	**EnumDone**	总线速度枚举完成，应用程序必须读取 DSTS 寄存器来获取枚举速度。软件写 1 清零。 (Enumeration done flag, write 1 to clear)
12	**USBRst**	检测到 USB 总线 RESET 事件，软件写 1 清零 (USB bus Reset event flag, write 1 to clear)
11	**USBSusp**	检测到 USB 总线 Suspend 事件，软件写 1 清零 (USB bus Suspend event flag, write 1 to clear)
10	**ErlySusp**	检测到 USB 总线 IDLE 3ms，软件写 1 清零 (USB bus early suspend flag, write 1 to clear)
9:8	**-**	RFU：未实现，读为 0
7	**GOUTNakEff**	全局 OUT NAK 有效标志。软件置位 DCTL.SGOUTNak 后，等待 GOUTNakEff 置位表示此操作已经成功。软件通过写 DCTL.CGOUTNak 寄存器可以清除此中断标志。 (Global OUT NAK effective flag)
6	**GINNakEff**	全局非周期性 IN NAK 有效标志。软件置位 DCTL.SGNPInNak 后，等待 GINNakEff 置位表示此操作已经成功。软件通过写 DCTL.CGNPInNak 寄存器可以清除此中断标志。 (Global IN NAK effective flag)
5	**-**	RFU：未实现，读为 0
4	**RxFNEmp**	RxFIFO 非空标志，置位表示 RxFIFO 中至少有一个数据包等待读取 (RxFIFO not empty flag)
3	**SoF**	帧起始标志，软件写 1 清零。(Start of Frame flag, write 1 to clear) 当控制器接收到一个 SOF token 时，此中断标志置位。软件可以读取 Device Status 寄存器来获取当前帧编号。
2	**-**	RFU：未实现，读为 0
1	**ModeMis**	模式不匹配 (Mode Mismatch)
0	**CurMod**	当前工作模式，固定为 0，表示 Device (Current Mode)

33.10.5 中断屏蔽寄存器（USB_GINTMSK）

名称	USB_GINTMSK							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WkUpINTMsk	-			LPM_INTMsk	-		
位权限	R/W-0	U-0			R/W-0	U-0		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ResetDetMsk	-	incomplSOOUTMsk	incomplSOINMsk	OEPINTMsk	IEPINTMsk	EPMisMsk	-
位权限	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EOPFMsk	ISOODropMsk	EnumDoneMsk	USBRstMsk	USBSuspMsk	ErlySuspMsk	-	-
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GOUTNakEffMsk	GINNakEffMsk	NPTxFEmpMsk	RxFNEmpMsk	SoFMsk	-	ModeMisMsk	-
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0

位号	助记符	功能描述
31:0	**xMsk**	GINTSTS 中各中断标志寄存器的 Mask 位（??需要各个 mask 分开说明把） 0：屏蔽中断产生 1：允许中断产生

33.10.6 接收状态 debug 读寄存器（USB_GRXSTSR）

AHB对GRXSTSR的读操作将返回RxFIFO中顶部数据包的相应信息。

名称	USB_GRXSTSR							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							FN[3]
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FN[2:0]			PktSts				DPID[1]
位权限	R-000			R-0000				R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DPID[0]	BCnt[10:4]						
位权限	R-0	R-000 0000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BCnt[3:0]				EPNum			
位权限	R-0000				R-0000			

位号	助记符	功能描述
31:25	**-**	RFU：未实现，读为 0
24:21	**FN**	Frame Number 帧编号的低 4 位，仅在 isochronous OUT 端点的情况下有效

位号	助记符	功能描述
20:17	**PktSts**	包状态 (Packet Status) 0001：Global OUT NAK（触发中断） 0010：收到 OUT 数据包 0011：OUT 传输完成（触发中断） 0100：SETUP 事务完成（触发中断） 0110：收到 SETUP 数据包 其他：RFU
16:15	**DPID**	收到的 OUT 数据包的数据 PID (Data Packet ID) 00：DATA0 01：DATA1 10：DATA2 11：MDATA
14:4	**BCnt**	收到的数据包的字节计数 (Byte Count)
3:0	**EPNum**	当前接收到的数据包所属的端点编号 (End point number)

33.10.7 接收状态读和 POP 寄存器（USB_GRXSTSP）

AHB对USB_GRXSTSP的读操作将返回RxFIFO中顶部数据包的相应信息，同时完成一次POP。
地址：0x00000020

33.10.8 RxFIFO size 寄存器（USB_GRXFXIZ）

名称	USB_GRXFSIZ							
Offset	0x00000024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RxFDep[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RxFDep[7:0]							
位权限	R-0011 0000							

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:0	**RxFDep**	RxFIFO 深度，以 32 位字为单位，固定为 48，只读。(RxFIFO Depth)

33.10.9 Non-Periodic TxFIFO size 寄存器（USB_GNPTXFSIZ）

名称	USB_GNPTXFSIZ							
Offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	INEPTxF0Dep[15:8]							

位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	INEPTxF0Dep[7:0]							
位权限	R-0001 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	INEPTxF0StAddr[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	INEPTxF0StAddr[7:0]							
位权限	R-0011 0000							

位号	助记符	功能描述
31:16	**INEPTxF0Dep**	端点 0TxFIFO 深度，以 32 位字为单位，固定为 16，只读(IN Endpoint 0 TxFIFO depth)
15:0	**INEPTxF0StAddr**	端点 0 TxFIFO 起始地址，只读(IN Endpoint 0 TxFIFO start address)

33.10.10 LPM 配置寄存器（USB_GLPMCFG）

名称	**USB_GLPMCFG**							
Offset	0x00000054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							L1ResumeOK
位权限	U-0							R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SlpSts	L1Resp		HIRD_Thres				
位权限	R-0	R-00		R-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	bRmtWk	HIRD				AppL1Res	LPMCap
位权限	U-0	R-0	R-0000				R/W-0	R/W-0

位号	助记符	功能描述
31:17	**-**	RFU：未实现，读为 0
16	**L1ResumeOK**	进入 Sleep 状态后允许 Resume，仅在 LPM Sleep(L1)状态下有效。此寄存器在控制器进入 Sleep 模式 50us 后置位(L1 Resume is OK enable) 1：允许 resume 0：禁止 resume
15	**SlpSts**	Sleep 状态指示。当控制器进入 Sleep 模式后置位。(Sleep Status) 退出方式： ● 检测到 USB 总线活动 ● 软件置位 Remote Wakeup Signaling（DCTL.RmtWkUpSig） ● 软件复位控制器

位号	助记符	功能描述
14:13	**L1Resp**	通过这个寄存器查询 MAC 控制器对 LPM transaction 的响应(L1 Response) 00：ERROR 01：STALL 10：NYET 11：ACK
12:8	**HIRD_Thres**	在 HIRD_Thres[4] =1 且 HIRD[3:0] >= HIRD_Thres[3:0]时，L1 模式下控制器将 PHY 置于深度掉电模式(Host Initiated Resume Duration Threshold)
7	**-**	RFU：未实现，读为 0
6	**bRmtWk**	是否可以远程唤醒，只读。 (Remote Wakeup bit) 收到 LPM token 时依据 bmAttribute 更新
5:2	**HIRD**	Host 发起 resume 信号长度 (Host Initiated Resume Duration) 收到 LPM token HIRD bmAttribute 时更新 0000：50us 0001：125us 0010：200us 0011：275us 0100：350us 0101：425us 0110：500us 0111：575us 1000：650us 1001：725us 1010：800us 1011：875us 1100：950us 1101：1025us 1110：1100us 1111：1175us
1	**AppL1Res**	LPM 响应 (Application L1 Response) 如果 GLPMCFG.LPMCap=0，控制器总是响应 NYET 如果 GLPMCFG.LPMCap=1，控制器根据 AppL1Res 的值响应： 0：NYET 1：ACK（前提是完成了成功的 LPM transaction）
0	**LPMCap**	LPM 使能 (LPM enable) 0：不使能 LPM 支持 1：使能 LPM 支持

33.10.11 Device 配置寄存器（USB_DCFG）

名称	USB_DCFG							
Offset	0x00000800							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ResValid						-	
位权限	R/W-00 0010						U-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	EPMisCnt					-	
位权限	U-0	R/W-1 0000					U-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-			PerFrInt		DevAddr[6:4]		
位权限	U-0			R/W-00		R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DevAddr[3:0]				En32KS	NZStsOUTHShk	DevSpd	
位权限	R/W-0000				R/W-0	R/W-0	R/W-00	

位号	助记符	功能描述
31:26	**ResValid**	Resume 有效长度配置，Resume 信号维持长度大于此配置值时才认为有效。此寄存器仅在 DCFG.En32KS=1 时有效。(Resume Valid)
25:23	**-**	RFU：未实现，读为 0
22:18	**EPMisCnt**	当端点不匹配的数据包到达此寄存器规定的数量时，置位端点不匹配中断标志 GINTSTS.EPMis(Endpoint Mismatch packet count)
17:13	**-**	RFU：未实现，读为 0
12:11	**PerFrInt**	在一个帧周期中的特定位置产生中断 (Period of Frame Interrupt) 00：80%帧周期 01：85%帧周期 10：90%帧周期 11：95%帧周期
10:4	**DevAddr**	软件需要在每个 SetAddress 控制命令之后将设备地址写入此寄存器 (Device Address)
3	**En32KS**	置位此寄存器，在 suspend 模式下 PHY 接口电路将使用 32K 时钟 (Enable 32K clock in Suspend)
2	**NZStsOUTHShk**	用于配置控制传输 status stage 中收到非 0 长度数据包时回发的握手包类型 (Non-Zero Status OUT Handshake) 0：将收到的 OUT 包传递给软件，并且根据设备端点控制寄存器中的 NAK 和 STALL 位来产生合适的握手 1：回发 STALL 握手，并且不把收到的 OUT 包发给软件
1:0	**DevSpd**	配置设备速度类型，软件必须将此寄存器配置为 11（USB1.1 PHY，时钟 48M）(Device Speed)

33.10.12 Device 控制寄存器（USB_DCTL）

名称	**USB_DCTL**							
Offset	0x00000804							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							NakOnBble
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				PWROnPrgDone	CGOUTNak	SGOUTNak	CGNPInNak
位权限	U-0				R/W-0	W-0	W-0	W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	SGNPInNak	TstCtl	GOUTNakSts	GNPINNakSts	SftDisc	RmtWkUpSig
位权限	W-0	R/W-000	R-0	R-0	R/W-0	R/W-0

位号	助记符	功能描述
31:17	-	RFU：未实现，读为 0
16	**NakOnBble**	置位时，控制器在收到 babble 后自动回发 NAK(NAK on Bubble)
15:12	-	RFU：未实现，读为 0
11	**PWROnPrgDone**	从 power down 模式唤醒后，软件置位此寄存器表示寄存器配置完成 (Power-ON Programming Done)
10	**CGOUTNak**	对此寄存器写 1 清除 Global OUT NAK (Clear Global OUT NAK) (GINTSTS.GOUTNakEff)
9	**SGOUTNak**	对此寄存器写 1 将对 GlobalOUT NAK 置 1 (Set Global OUT NAK) （GINTSTS.GOUTNakEff）
8	**CGNPInNak**	对此寄存器写 1 清除 Global Non-Periodic IN NAK (Clear Global Non-Periodic IN NAK) （GINTSTS.GINNakEff）
7	**SGNPInNak**	对此寄存器写 1 将对 non-periodic INNAK 置 1 (Set Global Non-Periodic IN NAK) （GINTSTS.GINNakEff）
6:4	**TstCtl**	测试控制 (Test Control) 000：测试模式禁止 001：测试 J 模式输出 010：测试 K 模式输出 011：测试 SE0_NAK 模式输出 100：测试包模式 101：测试 Force 使能 其他：保留
3	**GOUTNakSts**	Global OUT NAK status 0：根据 RxFIFO 状态和 NAK、STALL 位设置决定握手形式 1：数据不会写入 RxFIFO，收到所有数据包都回发 NAK，除了 SETUP
2	**GNPINNakSts**	Global Non-Periodic IN NAK status 0：根据 TxFIFO 状态回发握手 1：在所有 Non-Periodic IN 端点上发送 NAK
1	**SftDisc**	Soft Disconnect 软件通过置位此寄存器可以模拟 USB 断开连接
0	**RmtWkUpSig**	软件置位此寄存器对 HOST 发起 remote wakeup；(Remote Wakeup Signaling) 根据 USB2.0 spec，软件必须在置位此寄存器后 1~15ms 内清零。 当使能了 LPM 并且控制器处于 L1 状态时，置位此寄存器将使 PHY 发送 L1 remote signaling 唤醒 Host，同时控制器将推出 sleep 模式。此时硬件会在 50us 后自动清除 RmtWkUpSig 寄存器。

33.10.13 Device 状态寄存器（USB_DSTS）

名称	USB_DSTS
Offset	0x00000808

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		SOFFN[13:8]					
位权限	U-0		R-00 0000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFFN[7:0]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				ErrticErr	EnumSpd		SuspSts
位权限	U-0				R-0	R-01		R-0

位号	助记符	功能描述
31:22	**-**	RFU：未实现，读为 0
21:8	**SOFFN**	接收到的 SoF 的 Frame Number (SoF Frame Number)
7:4	**-**	RFU：未实现，读为 0
3	**ErrticErr**	PHY 错误状态。(PHY Error Status flag) 控制器检测到 PHY 的错误状态后，将主动进入 Suspend，同时产生 Early Suspend 中断标志。 要从此错误状态中恢复，软件只有进行 soft disconnect 操作。
2:1	**EnumSpd**	Chirp Sequence 之后的控制器速度设置 (Enumeration Speed) 00：HS 01：FS，PHY_CLK 为 30 或 60M 10：LS 11：FS，PHY_CLK 为 48M
0	**SuspSts**	当检测到总线 Suspend 事件时置位 (Suspend Status flag)

33.10.14 Device IN 端点通用中断屏蔽寄存器（USB_DIEPMSK）

此寄存器用于mask所有IN端点的同类型中断，0表示屏蔽中断，1表示允许中断，复位默认值为屏蔽中断。

名称	USB_DIEPMSK							
Offset	0x00000810							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		NakMsk	-			BNAIntMsk	TxFUndrnMsk
位权限	U-0		R/W-0	U-0			R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	INEPNakEffMsk	INTknEPMisMsk	INTknTxFEmpMsk	TimeOutMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:14	-	RFU：未实现，必须保持复位值
13	**NakMsk**	NAK 中断屏蔽 (NAK interrupt mask enable)
12:10	-	RFU：未实现，必须保持复位值
9	**BNAIntMsk**	BNA 中断屏蔽 (BNA interrupt Mask enable)
8	**TxFUndrnMsk**	TxFifo Underrun 中断屏蔽 (TxFIFO under-run interrupt mask enable)
7	-	RFU：未实现，必须保持复位值
6	**INEPNakEffMsk**	IN 端点 NAK 有效中断屏蔽 (IN endpoint NAK effective interrupt mask enable)
5	**INTknEPMisMsk**	EP 不匹配时接收到 IN 令牌中断屏蔽 (IN token endpoint mismatch interrupt mask enable)
4	**INTknTxFEmpMsk**	TxFIFO 为空时接收到 IN 令牌中断屏蔽 (IN token TxFIFO Empty interrupt mask enable)
3	**TimeOutMsk**	超时中断屏蔽，Non-isochronous endpoints (Time Out interrupt mask enable)
2	**AHBErrMsk**	AHB error 中断屏蔽 (AHB error interrupt mask enable)
1	**EPDisbldMsk**	端点禁止中断屏蔽 (Endpoint disabled interrupt mask enable)
0	**XferComplMsk**	传输完成中断屏蔽 (Transfer Complete Interrupt Mask enable)

33.10.15 Device OUT 端点通用中断屏蔽寄存器（USB_DOEPMSK）

此寄存器用于mask所有OUT端点的同类型中断，0表示屏蔽中断，1表示允许中断，复位默认值为屏蔽中断。

名称	USB_DOEPMSK							
offset	0x00000814							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	NYETMsk	NakMsk	BbleErrMsk	-		BnaOutIntMsk	OutPktErrMsk
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	B2Bsetup	-	OutTknEPdisMsk	SetupMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk
位权限	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:15	-	RFU：未实现，必须保持复位值
14	**NYETMsk**	NYET 中断屏蔽 (NYET interrupt mask enable)
13	**NakMsk**	NAK 中断屏蔽 (NAK interrupt mask enable)
12	**BbleErrMsk**	Babble 错中断屏蔽 (Babble error interrupt mask enable)
11:10	-	RFU：未实现，必须保持复位值
9	**BnaOutIntMsk**	BNA 中断屏蔽 (BNA Out interrupt mask enable)
8	**OutPktErrMsk**	OUT 包错中断屏蔽 (OUT packet error interrupt mask enable)

位号	助记符	功能描述
7	-	RFU：未实现，必须保持复位值
6	**B2Bsetup**	收到 Back-to-Back SETUP 包中断屏蔽 (Back-to-Back Setup packet interrupt mask enable)
5	-	RFU：未实现，必须保持复位值
4	**OutTknEPdisMsk**	端点禁止时接收到 OUT 令牌中断屏蔽 (Out Token received when Eondpoint disabled interrupt mask enable)
3	**SetupMsk**	SETUP 阶段完成中断屏蔽，仅适用于 control OUT endpoints (Setup done interrupt mask enable)
2	**AHBErrMsk**	AHB error 中断屏蔽 (AHB error interrupt mask enable)
1	**EPDisbldMsk**	端点禁止中断屏蔽 (Endpoint disabled interrupt mask enable)
0	**XferComplMsk**	传输完成中断屏蔽 (Transfer complete interrupt mask enable)

33.10.16 Device 全部端点中断寄存器（USB_DAINT）

名称	USB_DAINT							
Offset	0x00000818							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	OutEPInt[15:8]							
位权限	R-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OutEPInt[7:0]							
位权限	R-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	InEPInt[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	InEPInt[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	**OutEPInt**	OUT 端点中断标志寄存器，每个 OUT 端点对应 1bit，最多 16 个 OUT 端点 (OUT endpoint interrupt)
15:0	**InEPInt**	IN 端点中断标志寄存器，每个 IN 端点对应 1bit，最多 16 个 IN 端点 (IN endpoint interrupt)

33.10.17 Device 全部端点中断屏蔽寄存器（USB_DAINTMSK）

名称	USB_DAINTMSK							
offset	0x0000081C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	OutEPMsk[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OutEPMsk[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	InEPMsk[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	InEPMsk[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	**OutEPMsk**	OUT 端点中断屏蔽寄存器，每个 OUT 端点对应 1bit，最多 16 个 OUT 端点；(OUT endpoint interrupt mask) 0 屏蔽中断，1 允许中断
15:0	**InEPMsk**	IN 端点中断屏蔽寄存器，每个 IN 端点对应 1bit，最多 16 个 IN 端点；(IN endpoint interrupt mask) 0 屏蔽中断，1 允许中断

33.10.18 Device IN 端点 FIFO 空中断屏蔽寄存器（USB_DIEPEMPMSK）

名称	**USB_DIEPEMPMSK**							
Offset	0x00000834							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					InEpTxfEmpMsk		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31:3	**-**	RFU：未实现，读为 0
2:0	**InEpTxfEmpMsk**	IN 端点发送 FIFO 空中断屏蔽寄存器，0 表示屏蔽中断，1 允许中断 (IN endpoint TxFIFO empty interrupt mask) Bit0：IN 端点 0 Bit1：IN 端点 1 Bit2：IN 端点 2

33.10.19 Device Control IN 端点 0 控制寄存器（USB_DIEPCTL0）

名称	**USB_DIEPCTL0**							
Offset	0x00000900							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EPEna	EPDis	-		SNAK	CNAK	TxFNum[3:2]	
位权限	R/W-0	R/W-0	U-0		W-0	W-0	R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	TxFNum[1:0]		Stall	-	EPType		NAKSts	-
位权限	R/W-00		R/W-0	U-0	R-00		R-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	USBActEP	NxtEP				-		
位权限	R-1	R/W-0000				U-0		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						MPS	
位权限	U-0						R/W-00	

位号	助记符	功能描述
31	**EPEna**	端点使能 (Endpoint Enable) 软件置位表示数据准备好在控制端点上传输；当传输完成或端点被关闭时，硬件自动清零。
30	**EPDis**	软件置位此寄存器可以停止端点数据传输，即使传输没有完成。当 MAC 控制器关闭控制端点后，硬件自动清零 EPDis 并产生端点关闭中断。(Endpoint Disable)
29:28	**-**	RFU：未实现，读为 0
27	**SNAK**	软件写 1 置位 NAK (Set NAK)
26	**CNAK**	软件写 1 清零 NAK (Clear NAK)
25:22	**TxFNum**	发送 FIFO 编号，固定为 0000 (TxFIFO Number)
21	**Stall**	当接收到 SETUP token 时，软件写 1 回发 STALL 握手包 (Send Stall packet)
20	**-**	RFU：未实现，读为 0
19:18	**EPType**	端点类型，固定为 00 (Endpoint type)
17	**NAKSts**	NAK 状态位 (NAK Status) 0：控制器根据 FIFO 状态来回发非 NAK 握手包 1：控制器回发 NAK 当软件置位此寄存器后，控制器停止发送数据，即使 TxFIFO 为非空。注意，控制器总是对 SETUP 数据包回发 ACK
16	**-**	RFU：未实现，读为 0
15	**USBActEP**	控制端点激活标志，固定为 1 (USB active Endpoint)
14:11	**NxtEP**	当前端点数据从 TxFIFO 读出后，下一笔数据所属的端点编号 (Next Endpoint)
10:2	**-**	RFU：未实现，读为 0
1:0	**MPS**	控制端点最大数据包长 (Max Packet Size) 00：64bytes 01：32bytes 10：16bytes 11：8bytes

33.10.20 Device Control OUT 端点 0 控制寄存器（USB_DOEPCTL0）

名称	USB_DOEPCTL0							
Offset	0x00000B00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EPEna	EPDis	-		SNAK	CNAK	-	
位权限	R/W-0	R/W-0	U-0		W-0	W-0	U-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-		Stall	Snp	EPType		NAKSts	-
位权限	U-0		R/W-0	R/W-0	R-00		R-0	U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	USBActEP	-						
位权限	R-1	U-0						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						MPS	
位权限	U-0						R-00	

位号	助记符	功能描述
31	**EPEna**	端点使能 (Endpoint Enable) 软件置位表示数据准备好在控制端点上传输；当传输完成或端点被关闭时，硬件自动清零。
30	**EPDis**	软件置位此寄存器可以停止端点数据传输，即使传输没有完成。当 MAC 控制器关闭控制端点后，硬件自动清零 EPDis 并产生端点关闭中断。(Endpoint Disable)
29:28	**-**	RFU：未实现，读为 0
27	**SNAK**	软件写 1 置位 NAK (Set NAK)
26	**CNAK**	软件写 1 清零 NAK (Clear NAK)
25:22	**-**	RFU：未实现，读为 0
21	**Stall**	当接收到 SETUP token 时，软件写 1 回发 STALL 握手包 (Send Stall packet)
20	**Snp**	SnoopMode 1：控制器不检查收到的 OUT 数据包的正确性 0：控制器检查 OUT 数据包正确性
19:18	**EPType**	端点类型，固定为 00 (Endpoint type)
17	**NAKSts**	NAK 状态位 (NAK Status) 0：控制器根据 FIFO 状态来回发非 NAK 握手包 1：控制器回发 NAK 当软件置位此寄存器后，控制器停止发送数据，即使 TxFIFO 为非空。注意，控制器总是对 SETUP 数据包回发 ACK
16	**-**	RFU：未实现，读为 0
15	**USBActEP**	控制端点激活标志，固定为 1 (USB active Endpoint)
14:2	**-**	RFU：未实现，读为 0
1:0	**MPS**	控制端点最大数据包长，与 DIEPCTL0.MPS 一致，不可改写 (Max Packet Size)

33.10.21 Device IN 端点 1 控制寄存器（USB_DIEPCTLx）

名称	USB_DIEPCTLx(x=1,2)							
Offset	0x00000900 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	TxFNum	
位权限	R/W-0	R/W-0	W-0	W-0	W-0	W-0	R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TxFNum		Stall	Snp	EPType		NAKSts	DPID

位权限	R/W-00		R/W/Dy-0	R/W-0	R/W-00		R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	USBActEP	-				MPS[10:8]		
位权限	R-1	U-0				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MPS[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**EPEna**	端点使能(Endpoint Enable) 软件置位表示数据准备好在控制端点上传输；当传输完成或端点被关闭时，硬件自动清零。
30	**EPDis**	软件置位此寄存器可以停止端点数据传输，即使传输没有完成。当 MAC 控制器关闭控制端点后，硬件自动清零 EPDis 并产生端点关闭中断。(Endpoint Disable)
29	**SetD1PIN**	针对 interrupt/bulk 类型端点有效，设置 DATA1 数据包 PID (Set Data1 PID)
28	**SetD0PIN**	针对 interrupt/bulk 类型端点有效，设置 DATA0 数据包 PID (Set Data0 PID)
27	**SNAK**	软件写 1 置位 NAK (Set NAK)
26	**CNAK**	软件写 1 清零 NAK (Clear NAK)
25:22	**TxFNum**	发送 FIFO 编号 (TxFIFO Number)
21	**Stall**	回发 STALL 握手包 (Send Stall Packet)
20	**Snp**	SnoopMode 1：控制器不检查收到的 OUT 数据包的正确性 0：控制器检查 OUT 数据包正确性
19:18	**EPType**	端点类型配置 (Endpoint type) 00：Control 01：Isochronous 10：Bulk 11：Interrupt
17	**NAKSts**	NAK 状态位 (NAK Status) 0：控制器根据 FIFO 状态来回发非 NAK 握手包 1：控制器回发 NAK 当软件置位此寄存器后，控制器停止发送数据，即使 TxFIFO 为非空。注意，控制器总是对 SETUP 数据包回发 ACK
16	**DPID**	(Data PID) 针对 interrupt/bulk 端点： 0：DATA0 1：DATA1 针对 isochronous 端点： 0：偶数帧 1：奇数帧
15	**USBActEP**	控制端点激活标志，收到 USB 总线 RESET 后清零；收到 SetConfiguration 或 SetInterface 命令后软件应置位此寄存器。(USB Active Endpoint)

位号	助记符	功能描述
14:11	-	RFU：未实现，必须保持复位值
10:0	**MPS**	端点最大数据包长 (Max Packet Size)

33.10.22 Device OUT 端点 1 控制寄存器（USB_DOEPCTLx）

名称	USB_DOEPCTLx(x=1,2)							
Offset	0x00000B00 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EPEna	EPDis	SetOddFr	SetEvenFr	SNAK	CNAK	TxFNum	
位权限	R/W-0	R/W-0	W-0	W-0	W-0	W-0	R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TxFNum		Stall	Snp	EPType		NAKSts	DPID
位权限	R/W-00		R/W/Dy-0	R/W-0	R/W-00		R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	USBActEP	-				MPS[10:8]		
位权限	R-1	U-0				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MPS[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**EPEna**	端点使能(Endpoint Enable) 软件置位表示数据准备好在控制端点上传输；当传输完成或端点被关闭时，硬件自动清零。
30	**EPDis**	软件置位此寄存器可以停止端点数据传输，即使传输没有完成。当 MAC 控制器关闭控制端点后，硬件自动清零 EPDis 并产生端点关闭中断。(Endpoint Disable)
29	**SetOddFr**	针对 isochronous 端点有效，设置奇数帧 (Set Odd Frame)
28	**SetEvenFr**	针对 isochronous 端点有效，设置偶数帧 (Set Even Frame)
27	**SNAK**	软件写 1 置位 NAK (Set NAK)
26	**CNAK**	软件写 1 清零 NAK (Clear NAK)
25:22	**TxFNum**	发送 FIFO 编号 (TxFIFO Number)
21	**Stall**	回发 STALL 握手包 (Send Stall Packet)
20	**Snp**	SnoopMode 1：控制器不检查收到的 OUT 数据包的正确性 0：控制器检查 OUT 数据包正确性
19:18	**EPType**	端点类型配置 (Endpoint type) 00：Control 01：Isochronous 10：Bulk 11：Interrupt

位号	助记符	功能描述
17	**NAKSts**	NAK 状态位 (NAK Status) 0：控制器根据 FIFO 状态来回发非 NAK 握手包 1：控制器回发 NAK 当软件置位此寄存器后，控制器停止发送数据，即使 TxFIFO 为非空。注意，控制器总是对 SETUP 数据包回发 ACK
16	**DPID**	(Data PID) 针对 interrupt/bulk 端点： 0：DATA0 1：DATA1 针对 isochronous 端点： 0：偶数帧 1：奇数帧
15	**USBActEP**	控制端点激活标志，收到 USB 总线 RESET 后清零；收到 SetConfiguration 或 SetInterface 命令后软件应置位此寄存器。(USB Active Endpoint)
14:11	**-**	RFU：未实现，必须保持复位值
10:0	**MPS**	端点最大数据包长 (Max Packet Size)

33.10.23 Device 端点中断寄存器（USB_DIEPINTx）

名称	USB_DIEPINTx(x=0,1,2)							
Offset	0x00000908 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	NYETInt	NAKInt	BbleErrInt	PktDrpSts	-	-	-
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TxFEmp	INEPNakEff	-	INTknTXFEmp	TimeOUT	-	EPDisbld	XferCompl
位权限	R-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:15	**-**	RFU：未实现，读为 0
14	**NYETInt**	当控制器在非 Isochronous OUT 端点上收到 NYET 响应时，置位此中断标志 (Not Yet Interrupt)
13	**NAKInt**	当控制器发送或接受 NAK 时，置位此中断标志 (NAK Interrupt) Isochronous IN 端点由于 TxFIFO 空导致发送 0 长度数据包时，此中断标志置位

位号	助记符	功能描述
12	**BbleErrInt**	控制器端点收到 babble 时置位此中断标志 (Babble Error Interrupt)
11	**PktDrpSts**	数据包丢失状态 (Packet Dropped Status) 当 ISO OUT 数据包丢失时置位标志，但是不会产生中断
10:8	**-**	RFU：未实现，读为 0
7	**TxFEmp**	仅 IN 端点有效，在 TxFIFO 半空或全空时置位 (TxFIFO Empty) 半空或全空状态选择由 GAHBCFG.NPTxFEmpLvl 寄存器决定
6	**INEPNakEff**	仅 Periodic IN 端点有效，当控制器采样到 NAK 位有效时置位 通过 DIEPCTLn.CNAK 寄存器清零 (IN Endpoint NAK Effective)
5	**-**	RFU：未实现，必须保持复位值
4	**INTknTxFEmp**	当接收到 IN token，但是对应的 TxFIFO 为空时置位 (In Token when TxFIFO empty)
3	**TimeOut**	最后一个 IN token 超时 (Time Out)
2	**-**	RFU：未实现，读为 0
1	**EPDisbld**	端点被软件关闭 (Endpoint disabled)
0	**XferCompl**	端点传输完成标志 (Transfer Complete)

33.10.24 Device 端点中断寄存器（USB_DOEPINTx）

名称	USB_DOEPINTx(x=0,1,2)							
Offset	0x00000B08 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	NYETInt	NAKInt	BbleErrInt	PktDrpSts	-	-	-
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TxFEmp	B2Bsetup	-	OUTTknEPDis	Setup	-	EPDisbld	XferCompl
位权限	R-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:15	**-**	RFU：未实现，读为 0
14	**NYETInt**	当控制器在非 Isochronous OUT 端点上收到 NYET 响应时，置位此中断标志 (Not Yet Interrupt)
13	**NAKInt**	当控制器发送或接受 NAK 时，置位此中断标志 (NAK Interrupt) Isochronous IN 端点由于 TxFIFO 空导致发送 0 长度数据包时，此中断标志置位
12	**BbleErrInt**	控制器端点收到 babble 时置位此中断标志 (Babble Error

位号	助记符	功能描述
		Interrupt)
11	**PktDrpSts**	数据包丢失状态 (Packet Dropped Status) 当 ISO OUT 数据包丢失时置位标志，但是不会产生中断
10:8	**-**	RFU：未实现，读为 0
7	**TxFEmp**	仅 IN 端点有效，在 TxFIFO 半空或全空时置位 (TxFIFO Empty) 半空或全空状态选择由 GAHBCFG.NPTxFEmpLvl 寄存器决定
6	**B2Bsetup**	仅控制 OUT 端点有效，当控制器连续收到超过 3 个 back-to-back SETUP 包时置位 (Back-to-Back setup)
5	**-**	RFU：未实现，必须保持复位值
4	**OUTTknEPDis**	当收到 OUT token，但是对应端点还未使能时置位 (Out Token when Endpoint disabled)
3	**Setup**	仅控制 OUT 端点有效，置位表示 SETUP phase 完成 (Setup done)
2	**-**	RFU：未实现，读为 0
1	**EPDisbld**	端点被软件关闭 (Endpoint disabled)
0	**XferCompl**	端点传输完成标志 (Transfer Complete)

33.10.25 Device IN 端点 0 传输长度寄存器（USB_DIEPTSIZ0）

名称	USB_DIEPTSIZ0							
Offset	0x00000910							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			PktCnt		-		
位权限	U-0			R/W-00		U-0		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	XferSize						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31:21	**-**	RFU：未实现，读为 0
20:19	**PktCnt**	设置 Control 传输包含的 packet 数量；每次控制器从 TxFIFO 读取一个 packet 后此寄存器递减 (Packet Count)
18:7	**-**	RFU：未实现，读为 0
6:0	**XferSize**	端点 0 传输字节数。可以设置为最大包长，每个包发送完成后产生中断。 (Transfer Size)

33.10.26 Device OUT 端点 0 传输长度寄存器（USB_DOEPTSIZ0）

名称	USB_DOEPTSIZ0

Offset	0x00000B10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	SupCnt		-				
位权限	U-0	R/W-00		U-0				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			PktCnt		-		
位权限	U-0			R/W-00		U-0		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	XferSize						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31	**-**	RFU：未实现，读为 0
30:29	**SupCnt**	设置端点可以接收的 back-to-back SETUP 包数量 (Setup packet Count) 01：1 packet 10：2 packet 11：3 packet
28:21	**-**	RFU：未实现，读为 0
20:19	**PktCnt**	每次一个包写入 RxFIFO 后此寄存器递减 (Packet Count)
18:7	**-**	RFU：未实现，读为 0
6:0	**XferSize**	端点 0 传输字节数。可以设置为最大包长，每个包发送完成后产生中断。每次读取 RxFIFO 时硬件递减此寄存器。 (Transfer Size)

33.10.27 Device IN 端点 1 传输长度寄存器（USB_DIEPTSIZx）

名称	USB_DIEPTSIZx(x=1,2)							
Offset	0x00000910 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	MC		PktCnt[9:5]				
位权限	U-0	R/W-00		R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PktCnt[4:0]					XferSize[18:16]		
位权限	R/W-0 0000					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XferSize[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XferSize[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**-**	RFU：未实现，读为 0
30:29	**MC**	仅 periodic IN 端点有效，指示每个 frame 中必须传输的包数。

位号	助记符	功能描述
		控制器使用这个寄存器来计算 ISO 端点的 PID (Multi Count) 01：1 packet 10：2 packets 11：3 packets
28:19	**PktCnt**	设置传输总包数，每次控制器从 TxFIFO 取出一个包时递减 (Packet Count)
18:0	**XferSize**	端点 0 传输字节数。可以设置为最大包长，每个包发送完成后产生中断。(Transfer Size)

33.10.28 Device OUT 端点 1 传输长度寄存器（USB_DOEPTSIZx）

名称	USB_DOEPTSIZx(x=1,2)							
offset	0x00000B10 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	RxDPID		PktCnt[9:5]				
位权限	U-0	R/W-00		R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PktCnt[4:0]					XferSize[18:16]		
位权限	R/W-0 0000					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XferSize[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XferSize[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**-**	RFU：未实现，读为 0
30:29	**RxDPID**	仅 ISO OUT 端点有效，指示控制器收到的最近一个数据包的 PID (Received Data PID) 00：DATA0 01：DATA2 10：DATA1 11：MDATA
28:19	**PktCnt**	设置传输总包数，每次控制器向 RxFIFO 写入 1 个包时递减 (Packet Count)
18:0	**XferSize**	端点 0 传输字节数。可以设置为最大包长，每个包接收完成后产生中断。每次从 RxFIFO 读出包时递减。(Transfer Size)

33.10.29 Device IN 端点发送 FIFO 状态寄存器（USB_DTXFSTSx）

名称	DTXFSTSx(x=0,1,2)							
Offset	0x00000918 + x*0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	INEPTxFSpcAvail[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	INEPTxFSpcAvail[7:0]							
位权限	R/W-0001 0000							

位号	助记符	功能描述
31:16	**-**	RFU：未实现，读为 0
15:0	**INEPTxFSpcAvail**	端点发送 FIFO 中的可用空间 (IN Endpoint TxFIFO Space Available) 0：TxFIFO 满 N：n words 可用

33.10.30 功耗控制寄存器（USB_PCGCCTL）

名称	USB_PCGCCTL							
Offset	0x00000E00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DpSlp	PhySlp	EN_L1Gating	-			GateHclk	StopPclk
位权限	R-0	R-0	R/W-0	U-0			R/W-0	R/W-0

位号	助记符	功能描述
31:8	**-**	RFU：未实现，读为 0
7	**DpSlp**	1 表示 L1 状态下 PHY 处于 deep sleep
6	**PhySlp**	1 表示 PHY 处于 sleep
5	**EN_L1Gating**	置位此寄存器时，L1 模式下关闭 PHY 时钟 (Enable L1 Clock Gating)
4:2	**-**	RFU：未实现，读为 0
1	**GateHclk**	Suspend 模式下关闭 HCLK (Gate AHB Clock)
0	**StopPclk**	Suspend 模式下关闭 PHY 时钟 (Stop PHY Clock)

33.11 编程模型

33.11.1 模块初始化

本节介绍了USB FS控制器上电后，应用程序必须执行的模块初始化序列。

上电后对所有模块全局寄存器进行初始化过程：

1、读取硬件配置寄存器(GHWCFG1, 2, 3, and 4)来判断控制器的配置参数。

2、配置AHB配置寄存器(GAHBCFG)的以下字段：

- 全局中断屏蔽位GINTMSK = 1
- RxFIFO 非空水位(GINTSTS.RXFLVL)
- 周期性TxFIFO 空水位(NPTxFEL)

3、配置USB全局配置寄存器(GUSBCFG)的以下字段：

- FS 超时校准字段
- USB 周转时间字段

4、软件取消对GINTMSK 寄存器中模式不匹配中断的屏蔽。

5、读取GINTSTS 中的CMOD 位，确定FS 控制器是工作在设备模式下。

33.11.2 设备初始化

上电后，模块初始化结束，确认进入设备模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化：

1、配置 DCFG 寄存器中以下字段：

- 设备速度
- 非零长度状态OUT 握手信号

2、编程GINTMSK 寄存器以开放以下中断：

- USB Reset
- Enumeration Done
- Early Suspend
- USB Suspend
- SOF

3、等待 GINTSTS 中的USBRST 中断。这表示已在 USB 上检测到复位信号并持续10 ms。

4、等待 GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。接收到此中断时，应用程序必须读取 DSTS 寄存器以确定枚举速度并执行枚举完成时的端点初始化中所列的步骤。

此时，设备已准备好接收 SOF 数据包并开始控制端点 0 上执行控制传输。

33.11.3 设备编程

33.11.3.1 端点 USB 复位初始化

1、 将所有 OUT 端点将 NAK 位置 1 ，DOEPCTLn.SNAK = 1

2、取消对以下中断位的屏蔽

- DAINTMSK.INEP0= 1（控制 0 IN 端点）
- DAINTMSK.OUTEP0 = 1（控制 0 OUT 端点）
- DOEPMSK.SETUP= 1
- DOEPMSK.XferCompl = 1
- DIEPMSK.XferCompl = 1
- DIEPMSK.TimeOut = 1

3、对端点相关寄存器中的以下字段进行编程，以使控制 OUT 端点 0 接收 SETUP 数据包

- DOEPTSIZ0.SetUP Count = 3

此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

33.11.3.2 端点枚举结束初始化

1、在枚举完成中断(GINTSTS. EnumDone)中，读取 DSTS 寄存器以确定设备的枚举速度。

2、对DIEPCTL0.MPS字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 上执行控制传输。

33.11.3.3 端点收到 SetAddress 命令时的初始化

1、将 SetAddress 命令中接收到的设备地址写入 DCFG 寄存器相应字段。

2、对模块进行编程以发出状态阶段的 IN 数据包。

33.11.3.4 端点收到 SetConfiguration/SetInterface 命令时的初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1、接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。

2、接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。

3、在新的配置中无效的端点必须停用。

4、修改 DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。

5、配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

33.11.3.5 1.14.3.5 端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1、配置DIEPCTLn或DOEPCTLn寄存器的以下字段，对所需端点的特性进行编程。

- 最大包大小
- USB 活动端点位设置为 1
- 端点初始数据同步位（对于中断和批量端点）
- 端点类型
- TxFIFO 编号

1、激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

33.11.3.6 端点停用

本节介绍停用现有端点所需的步骤。

1、在要停用的端点中，将DIEPCTLn或 DOEPCTLn寄存器中的 USB 活动端点位清零。

2、停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

33.11.4 操作模型

33.11.4.1 OUT 数据传输

本节描述在out传输阶段，内部数据流及应用程序操作步骤。

数据包读取

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1、捕获到RXFLVL中断(GINTSTS.RxFLvl)时，应用程序必须读取接收状态弹出寄存器 (GRXSTSP)。

2、应用程序可以通过写入GINTMSK.RxFLvl = 1’b0,来屏蔽 RXFLVL中断，直到数据包从接收 FIFO 中读取完毕。

3、如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出这些数据并存储在存储器中。

如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。

4、从接收 FIFO 读出的数据包状态有以下几种状态：

a) 全局 OUT NAK状态： PktSts = Global OUT NAK, BCnt = 11'h000, EPNum = I Care (4'h0), DPID = I Care (2'b00)。这些数据表示全局 OUT NAK 位已生效。

b) SETUP 数据包：PktSts = SETUP, BCnt = 11'h008, EPNum = Control EP Num, DPID = D0。这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。

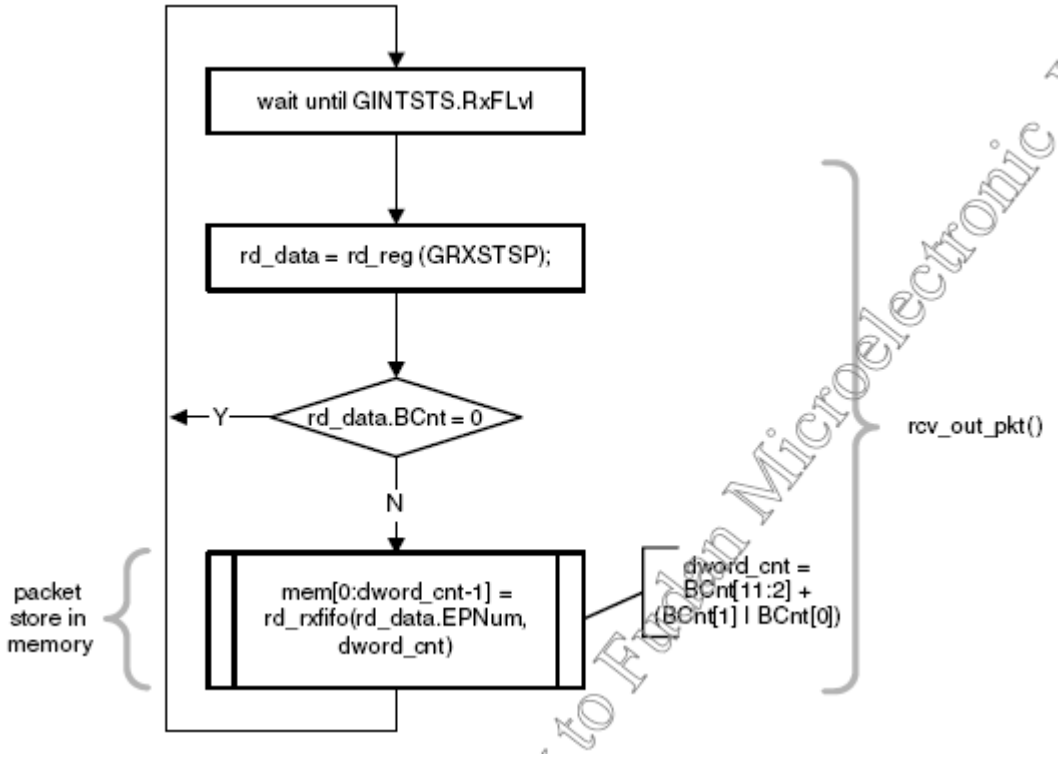
c) SETUP阶段完成： PktSts = Setup Stage Done, BCnt = 11'h0, EPNum = Control EP Num, DPID = Don't Care (2'b00)。这些数据表示指定端点的SETUP阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在该控制 OUT 端点上产生SETUP中断。

d) OUT 数据包： PktSts = DataOUT，BCnt =接收到的 OUT 数据包的大小(0 ≤ BCnt ≤ 1,024)，EPNum =收到数据包的端点编号，DPID =实际数据 PID。

e) 数据传输完成：PktSts = OUT数据传输完成， BCnt = 11'h0, EPNum =完成数据传输的OUT EP编号, DPID = I Care (2'b00)。这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发"传输完成"中断。

5、从接收 FIFO 中弹出数据后，必须取消对 GINTSTS.RxFLvl中断的屏蔽。

6、每次应用程序检测到 GINTSTS.RxFLvl 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。



Setup传输

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

●应用程序要求

1、要接收 SETUP 数据包，必须将控制 OUT 端点中的DOEPTSIZn.SUPCnt 字段编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 DOEPCTLn.EPEna位的设置。控制端点每收到一个 SETUP 数据包后，SUPCnt字段都会递减。如果在接收 SETUP 数据包之前，未将 SUPCnt字段编程为适当值，模块仍能接收 SETUP 数据包并使 SUPCnt字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包正确数量。

2、应用程序必须始终在接收数据 FIFO 中分配足够空间，以便能够在控制端点上接收连续的最多三个 SETUP 数据包。

3、应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个DWORDs。

4、应用程序必须从接收 FIFO 中读取并丢弃一个DWORD的“建立阶段完成”状态字。

●内部数据流

1、接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUT NAK 位置 1。

2、USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将SUPCnt字段递减 1。

— 第一个字包含由模块所使用的内部控制信息

— 第二个字包含 SETUP 命令的前 4 个字节

— 第三个字包含 SETUP 命令的最后 4 个字节

3、当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。

4、在 AHB 端，SETUP 数据包被应用程序读取。

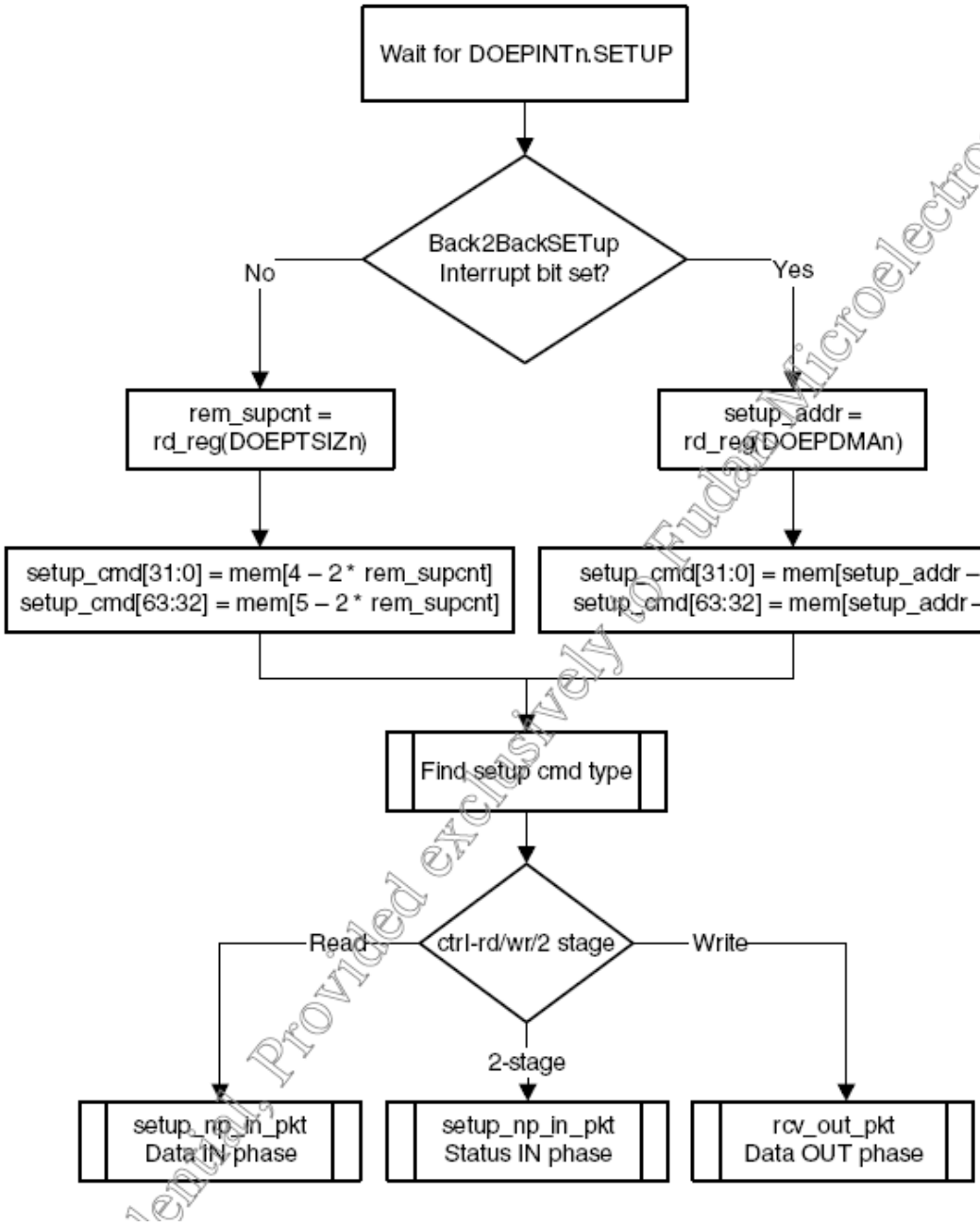
5、当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将触发 DOEPINTn.SETUP 中断，指示应用程序可以处理接收到的 SETUP 数据包。模块会将控制 OUT 端点的端点使能位清零。

●应用程序编程流程

1、设置字段DOEPTSIZn.SUPCnt = 3。

2、等待 GINTSTS.RxFLvl中断，并且从接收 FIFO 中进行数据包读取。

3、DOEPINTn.SETUP 中断触发表示 SETUP 数据传输成功完成。发生该中断时，应用程序必须读取DOEPTSIZn寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。



- 处理三个以上连续的 SETUP 数据包

根据 USB 2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB 2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。发生这种情况时，模块将生成中断DOEPINTn.Back2BackSETup。

将全局 OUT NAK 置 1

- 内部数据流：

1、如果应用程序将DCTL.SGOUTNak位置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复

NAK，而对同步 OUT 数据包直接忽略。

2、模块将全局 OUT NAK 模式DWORD写入接收 FIFO。应用程序必须为此留出足够空间。

3、当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GINTSTS.GOUTNakEff 中断置 1。

4、应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 DCTL.SGOUTNak 位清零来清除该中断。

- 应用程序编程顺序：

1、要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。DCTL.SGOUTNak = 1’b1

2、等待GINTSTS.GOUTNakEff中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。

3、在应用程序将 DCTL.SGOUTNak 位置 1之后，模块触发 GINTSTS.GOUTNakEff 中断之前，应用程序可以接收有效 OUT 数据包。

4、应用程序可通过对GINTMSK.GINNakEffMsk 位执行写操作来暂时屏蔽此中断。GINTMSK.GINNakEffMsk = 1’b0

5、当应用程序准备退出全局 OUT NAK 模式时，必须将DCTL.SGOUTNak 位清零。此操作还会清除 GINTSTS.GOUTNakEff中断。 DCTL.CGOUTNak = 1’b1

6、如果应用程序在之前已屏蔽此中断，则必须按以下方式取消对该中断的屏蔽：

GINTMSK.GINNakEffMsk = 1’b1

禁止 OUT 端点

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

- 应用程序编程顺序：

1、禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。

DCTL.SGOUTNak = 1’b1

2、等待 GINTSTS.GOUTNakEff 中断

3、通过编程以下字段来禁止 OUT 端点：

DOEPCTLn.EPDisable = 1’b1

DOEPCTLn.SNAK = 1’b1

1、等待 DOEPINTn.EPDisabled 中断，该中断表示已完全禁止OUT端点。引发 DOEPINTn.EPDisabled中断时，模块还会将以下位清零：

DOEPCTLn.EPDisable = 1’b0

DOEPCTLn.EPEnable = 1’b0

1、应用程序必须将全局 OUT NAK 位清零，以开始从其它未禁止的 OUT 端点接收数据。

DCTL.SGOUTNak = 1’b0

停止非同步 OUT 端点

本节介绍应用程序如何停止非同步端点。

1、将模块置于全局 OUT NAK模式。

2、禁止所需的端点，并设置STALL位

DOEPCTL.STALL = 1

3、当应用程序不再需要端点回复STALL握手信号时，必须将DOEPCTL.STALL位清零。

4、如果应用程序由于收到主机的SetFeature.Endpoint Halt或ClearFeature.Endpoint Halt命令来设置或清除端点的STALL状态，则必须在该控制端点进入状态阶段传输前，将 STALL位置1或清零。

通用非同步 OUT 数据传输

本节介绍一种常规非同步 OUT 数据传输（控制、批量或中断）。

- 应用程序要求：

1、建立OUT传输前，应用程序必须在存储器中分配一个缓冲区，以容纳要作为 OUT 传输的一部分而接收的所有数据。

2、对于OUT传输，端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数，且以字对齐。

- 传输大小[EPNUM] = n × (MPSIZ[EPNUM] + 4 – (MPSIZ[EPNUM] mod 4))
- 数据包个数[EPNUM] = n
- n > 0

3、发生OUT端点中断时，应用程序必须读取端点的传输大小寄存器以计算存储器中有效数据量。接收的有效数据量可能小于编程的传输大小。

- 存储器中的有效数据量 = 应用程序设置的初始传输量–模块更新后的剩余传输量
- 接收到 USB 数据包数 = 应用程序设置的初始数据包数-模块更新后的剩余数据包数

- 内部数据流：

1、应用程序必须在端点相关寄存器中设置传输大小和数据包计数字段，将 NAK 位清零，并使能端点来接收数据。

2、NAK 位清零后，只要接收 FIFO 中有空间，模块便开始接收数据并将数据写入接收FIFO。对于USB上接收的每个数据包，数据包及其状态都会写入接收FIFO。写入接收FIFO的每个数据包都会使该端点的数据包计数字段递减1。

- 收到的数据包若CRC无效，则自动被从接收FIFO中清除。
- 在USB上为数据包回复ACK后，模块将丢弃主机因无法检测到ACK而重新发送的非同步OUT数据包。应用程序不会在具有相同数据PID的相同端点上检测到多个连续的OUT数据包。在这种情况下，

数据包计数不会递减。

- 如果接收FIFO中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收FIFO。此外，非同步OUT令牌将会收到NAK握手应答。

- 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。

3、当数据包计数变为0或者在端点上接收到短数据包时，该端点的NAK位将置1。NAK位置1后，将忽略同步或非同步数据包并且不会将它们写入接收FIFO，同时非同步OUT令牌会收到NAK握手应答。

4、在数据写入接收FIFO后，应用程序将从接收FIFO中读取数据并将数据写入外部存储器，每个端点一次一个数据包。

5、在向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。

6、在以下情况时，OUT 端点的OUT数据传输完成状态将写入接收 FIFO：

- 传输大小为0并且数据包计数为 0

- 写入接收FIFO的最后一个OUT数据包是短数据包（0 =<数据包大小<最大数据包大小）

7、当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

- 应用程序编程流程：

1、使用传输大小和相应数据包个数对DOEPTSIZn寄存器进行编程。

2、使用端点特性对DOEPCTLn寄存器进行编程，设置端点使能位并清除NAK 位。

DOEPCTLn.EPEna = 1

DOEPCTLn.CNAK = 1

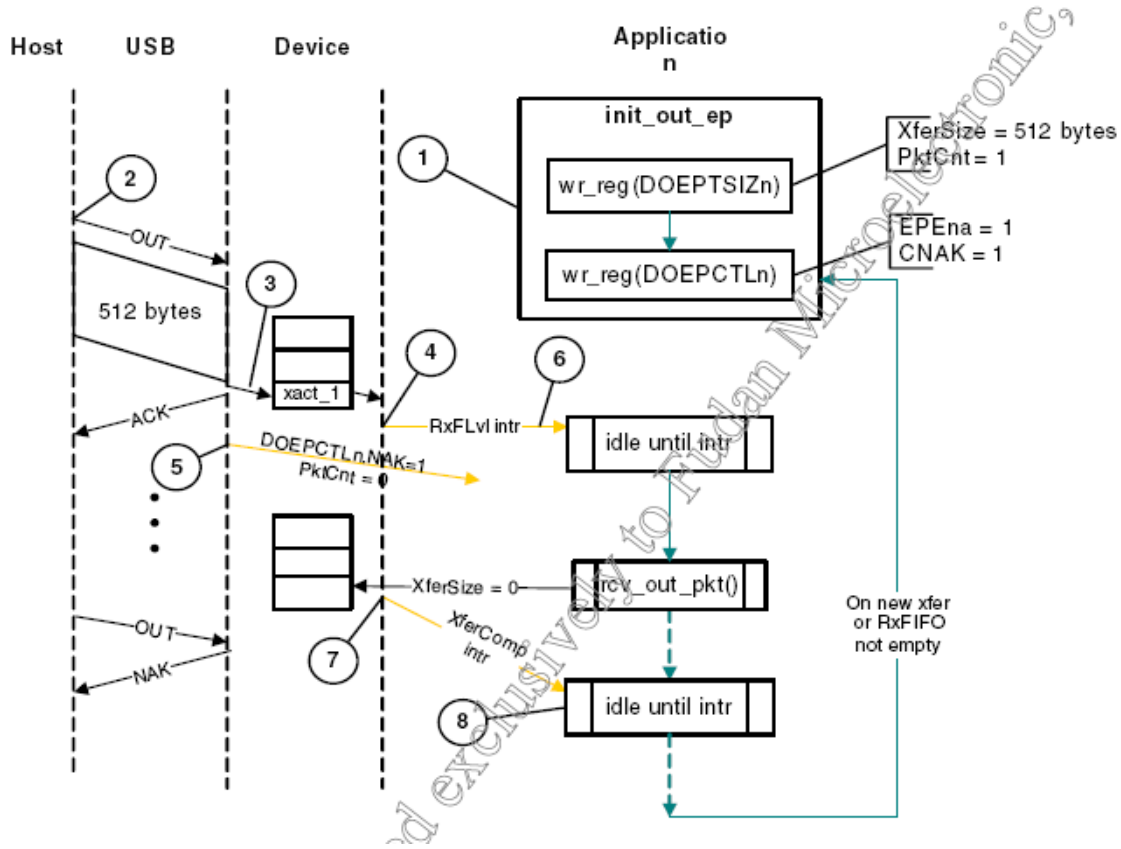
3、等待GINTSTS.RxStsQ中断并且从接收FIFO中读走数据包。此步骤可重复多次，具体取决于传输大小。

4、触发DOEPINTn.XferCompl中断，以表示非同步 OUT 数据传输成功完成。

5、读取DOEPTSIZn寄存器，以确定有效数据量。

示例：批量OUT传输

本节描述了将单个批量 OUT数据包从USB接收到AHB中的过程。



在接收到SetConfiguration/SetInterface命令后，应用程序将初始化所有OUT端点：设置DOEPCTLn寄存器中端点使能位并清除NAK 位，使用传输大小和相应数据包个数对DOEPTSIZn 寄存器进行编程。

1、主机尝试将数据（OUT 令牌）发送到端点。

2、当模块在USB上接收到OUT令牌时，模块会将数据包存储在RxFIFO中，只要其中有可用空间。

3、在RxFIFO中写入完整数据包后，模块随后会引发GINTSTS.RxFLvl中断。

4、接收到PKTCNT所指定数量的USB数据包后，模块在内部将该端点的NAK位置1，以避免其再接收任何数据包。

5、应用程序处理中断并从RxFIFO中读取数据。

6、应用程序读取完所有数据后（相当于 XferSize），模块将生成DOEPINTn.XferCompl中断。

7、应用程序处理中断并通过DOEPINTn.XferCompl中断的触发得知本次传输完成。

通用同步 OUT 数据传输

本节介绍常规的同步 OUT 数据传输。

- 应用程序要求

1、非同步OUT数据传输的所有应用程序要求均适用于同步OUT数据传输。

2、对于同步OUT数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的

最大数据包大小的数据包数目。同步类型的OUT数据传输事务必须在一个帧内完成。

1 ≤ packet count[epnum] ≤ 3

3、在周期性帧结束（GINTSTS.EOPF中断）之前，应用程序必须从接收FIFO中读取所有同步OUT数据包（数据条目和状态条目）。

4、要接收下一帧中的数据，必须在 GINTSTS.EOPF之后GINTSTS.SOF之前使能一个同步OUT端点。

- 内部数据流

1、同步OUT端点的内部数据流与非同步OUT端点的基本相同，但稍有差异。

2、同步OUT端点通过将端点使能位置1并将NAK位清零来使能时，必须相应地将偶数/奇数帧位置1。仅当符合以下条件时，模块才会在同步OUT端点上接收特定帧中的数据：

DOEPCTLn.Even/Odd microframe = DSTS.SOFFN[0]

4、 当应用程序从接收FIFO中完整地读取一个同步OUT数据包（数据和状态）时，模块会根据从接收FIFO中读取的最后一个同步OUT数据包的数据PID更新 DOEPTSIZn.ReceivedDPID字段。

- 应用程序编程顺序

1、使用传输大小和相应数据包计数对DOEPTSIZn寄存器进行编程

2、使用端点特性对DOEPCTLn寄存器进行编程，并将端点使能位、清除NAK位和奇数/偶数帧位置1。

- EPENA = 1
- CNAK = 1
- EONUM = 0（偶数）or 1（奇数）

3、等待GINTSTS.Rx StsQ中断并且从接收FIFO中读走数据包，此步骤可重复多次，具体取决于传输大小。

4、DOEPINTn.XferCompl中断表示同步OUT数据传输完成。该中断不一定意味着存储器中的数据是有效的。

5、对于同步OUT传输，应用程序可能并不总会检测到该中断。相反，应用程序可能检测到GINTSTS.incomplete Isochronous OUT中断。

6、读取DOEPTSIZn寄存器以确定接收的传输大小以及确定帧中接收的数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：

- DOEPTSIZn.RxDPID = D0，并且接收该有效数据的USB数据包数量 = 1
- DOEPTSIZn.RxDPID = D1，并且接收该有效数据的USB数据包数量 = 2

接收该有效数据的USB数据包数量 = 应用程序编程的初始数据包个数–模块更新后的剩余数据包个数，应用程序可将无效数据包丢弃。

不完整的同步 OUT 数据传输

本节介绍了同步 OUT 数据包出现丢包时应用程序编程流程。

- 内部数据流：

1、对于同步OUT端点，可能不会始终引发DOEPINTn.XferCompl中断。如果模块丢弃同步OUT数据包，则在以下情况下，应用程序可能无法检测到DOEPINTn.XferCompl中断：

-在接收FIFO无法容纳完整的ISO OUT数据包时，模块将丢弃接收到的ISO OUT数据

-接收到的同步OUT数据包存在CRC错误

-模块接收到的同步OUT令牌损坏

-应用程序从接收FIFO中读取数据的速度非常缓慢

2、如果模块在所有同步OUT端点的传输完成前检测到周期性帧结束，将触发未完成同步OUT数据中断（GINTSTS.incomplete Isochronous OUT data），指示至少有一个同步OUT端点上未触发DOEPINTn.XferCompl中断。此时，未完成传输的端点仍保持使能，但在USB的该端点上，没有进行中的有效传输。

- 应用程序编程流程：

1、硬件触发GINTSTS.incomplete Isochronous OUT data中断表示当前帧中至少有一个同步OUT端点具有未完成的传输。

2、如果因未从端点完全读取同步OUT数据而发生这种情况，应用程序必须确保首先从接收FIFO读取走所有同步OUT数据（包括数据条目和状态条目），然后再继续处理。

3、当应用程序接收到GINTSTS.incomplete Isochronous OUT data中断时，应用程序必须读取所有同步OUT端点的控制寄存器(DOEPCTLn)，以确定哪些端点在当前帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：

- DOEPCTLn.Even/Odd microframe bit = DSTS.SOFFN[0]
- DOEPCTLn.Endpoint Enable = 1

4、在检测到GINTSTS.SOF中断前，必须执行完成上一步操作，以确保当前帧编号未发生更改。

5、对于具有不完整传输的同步OUT端点，应用程序必须丢弃存储器中的数据，并通过将DOEPCTLn.Endpoint Disable bit位置1来禁止端点。

6、等待DOEPINTn.Endpoint Disabled中断，并且使能端点以在下一帧中接收新数据。由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

33.11.4.2 IN 数据传输

数据包写入

本节介绍应用程序如何将数据包写入端点FIFO。

1、 应用程序可以选择轮询模式或中断模式。

— 在轮询模式下，应用程序通过读取DTXFSTSn寄存器来监视端点发送数据FIFO的状态，从而确定数据FIFO中是否有足够空间。

— 在中断模式下，应用程序等待DIEPINTn.TxFEmp中断，然后读取DTXFSTSn寄存器以确定数据FIFO中是否有足够空间。

— 要写入单个非零长度的数据包，数据FIFO中必须有足够的空间来容纳整个数据包。

— 要写入零长度的数据包，应用程序一定不要查看FIFO空间。

1、 当应用程序确定有足够空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据FIFO。通常，应用程序必须对DIEPCTLn寄存器执行读改写操作，以避免在将端点使能位置1的同时修改寄存器中的其它内容。

如果有足够空间，应用程序可将同一端点的多个数据包写入发送FIFO。

将 IN 端点 NAK 置 1

- 内部数据流：

1、当应用程序将特定端点的IN NAK置1时，模块将停止端点上的数据发送，而不考虑端点发送FIFO中的数据是否可用。

2、非同步端点收到IN令牌，回复NAK握手应答。同步端点收到IN令牌，返回零长度数据包。

3、模块触发DIEPINTn.IN NAK Effective（IN 端点 NAK 有效）中断以响应DIEPCTL.Set NAK位。

4、应用程序检测到该中断后，便会认为端点处于IN NAK模式。应用程序可通过将DIEPCTLn. Clear NAK位置1来清除该中断。

- 应用程序编程流程：

1、 要在特定IN端点上停止发送任何数据，应用程序必须将IN NAK位置1。

DIEPCTLn.SetNAK = 1’b1

2、等待DIEPINTn.NAK Effective中断触发。该中断表示模块已在端点上停止发送数据。

3、在应用程序将NAK位置1，但DIEPINTn.NAK Effective中断尚未触发时，模块可以在端点上发送有效IN数据。

4、应用程序可通过写入DIEPMSK.NAK Effective位来临时屏蔽该中断。

DIEPMSK.NAK Effective = 1’b0

1、 要退出端点NAK模式，应用程序必须将DIEPCTLn.NAK状态位清零。此操作还会清除DIEPINTn.NAK Effective 中断。

DIEPCTLn.ClearNAK = 1’b1

1、如果应用程序已将该中断屏蔽，则必须按以下方式取消屏蔽：

DIEPMSK.NAK Effective = 1’b1

禁止 IN 端点

使用以下编程流程来禁止先前已使能的特定 IN 端点。

- 应用程序编程流程

1、应用程序必须先停止在AHB上写入数据，之后才能禁止IN端点。

2、应用程序必须将端点设置为NAK模式。

DIEPCTLn.SetNAK = 1’b1

3、等待DIEPINTn.NAK Effective中断。

4、将必须禁止的端点的 DIEPCTLn寄存器中的以下位置 1。

DIEPCTLn.Endpoint Disable = 1’b1

DIEPCTLn.SetNAK = 1’b1

1、 DIEPINTn.Endpoint Disabled中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：

DIEPCTLn.EPEnable = 1’b0

DIEPCTLn.EPDisable = 1’b0

6、对于周期性IN EP，应用程序必须读取DIEPTSIZn寄存器，以计算端点上有多少数据是在USB上发送的。

7、应用程序必须通过将GRSTCTL寄存器中的以下字段置 1，来清空端点发送FIFO中的数据：

GRSTCTL.TxFIFONum = Endpoint Transmit FIFO Number

GRSTCTL.TxFFlush = 1

应用程序必须轮询OTG_FS_GRSTCTL寄存器，直至模块将TXFFLSH位清零，这表示FIFO清空操作结束。要在该端点上发送新数据，应用程序可以在稍后重新使能该端点。

停止非同步 IN 端点

本节介绍应用程序如何才能停止非同步端点。

- 应用程序编程流程

1、禁止要停止的 IN 端点。同时将 STALL 位置 1。

2、当端点已使能时,设置

DIEPCTLn.Endpoint Disable = 1

DIEPCTLn.STALL = 1

STALL位总是比NAK位具有更高的优先级

3、模块触发DIEPINTn.Endpoint Disabled中断让应用程序知道指定端点已被禁止。

4、应用程序必须根据端点类型清空发送FIFO。对于非周期性端点，应用程序必须重新使能另一个

无需停止的非周期性端点来发送数据。

5、当应用程序准备好结束该端点的 STALL 握手信号时，必须将 DIEPCTLn.STALL 位清零。

6、如果应用程序因收到来自主机的 SetFeature.Endpoint Halt 命令或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态转换之前将 STALL 位置 1 或清零。

- 特例：停止控制IN/OUT 端点

如果在控制传输的数据阶段，主机发送的IN/OUT令牌数超过SETUP数据包指定的值，则模块必须对这些多余的IN/OUT令牌回复STALL。在这种情况下，应用程序必须在控制传输的数据阶段使能DIEPINTn.INTknTXFEmp中断和DOEPINTn.OUTTknEPdis中断（当模块已完成传输SETUP数据包指定的数据量后）。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的STALL位置1并清除此中断。

通用非周期性 IN 数据传输

- 应用程序要求：

1、建立IN传输前，应用程序必须确保组成一次IN传输的每个数据包都可以容纳在单个缓冲区中。

2、对于IN传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。

— 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：

传输大小[epnum] = n × MPSIZ[epnum] + sp (n >= 0，0 =< sp < MPSIZ[epnum])

如果 sp > 0，数据包计数[epnum] = n + 1；否则，数据包计数[epnum] = n

—要发送单个零长度数据包：

传输大小[epnum] = 0

数据包计数[epnum] = 1

—要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。

第一次传输：传输大小[epnum] = n × MPSIZ[epnum]；数据包计数 = n；

第二次传输：传输大小[epnum] = 0；数据包计数[epnum] = 1；

3、使能某个端点进行数据传输后，模块会更新传输大小寄存器。在IN传输结束时（Endpoint Disabled中断），应用程序必须读取传输大小寄存器，以确定送入发送FIFO中的数据已有多少通过USB发送出去。

4、送入发送FIFO中的数据量 = 应用程序编程的初始传输大小–模块更新后的最终传输大小

在USB上传输的数据量 = （应用程序编程的初始传送包计数–模块更新后的最终传送包计数）×MPSIZ[epnum]

要通过USB发送的剩余数据量 =应用程序编程的初始传输大小–在USB上传输的数据量

- 内部数据流：

1、应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。

2、应用程序还必须向该端点的发送FIFO写入需发送的数据。

3、应用程序每向发送FIFO写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送FIFO，直到该端点的传输大小变为0。向FIFO写入数据后，“FIFO中的数据包数”计数会递增（这是一个3位计数，由模块在内部进行维护，每个IN端点发送FIFO对应一个。在IN端点FIFO中，模块所维护的最大数据包数始终为8个）。对于零长度数据包，每个FIFO均另有一个单独的标志，FIFO中没有任何数据。

4、当数据写入发送FIFO后，模块会在接收到IN令牌时将这些数据送出。每个数据包发送出去并收到回复的ACK握手信号后，该端点的数据包计数都会递减1，直到数据包计数变0为止。发生超时时，数据包计数不会递减。

5、对于零长度数据包（由内部零长度标志指示），模块会在收到IN令牌后发出一个零长度数据包，并递减数据包计数字段的值。

6、如果接收到IN令牌的端点对应的FIFO中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个IN Tkn Rcvd When FIFO Empty中断（前提是该端点的NAK位未置1）。模块在该非同步端点上回复NAK握手信号。

7、模块会在内部使FIFO指针重新返回到开头，并且不会生成超时中断。

8、当传输大小为0且数据包计数为0时，将生成该端点的传输完成中断，同时将端点使能清零。

- 应用程序编程流程：

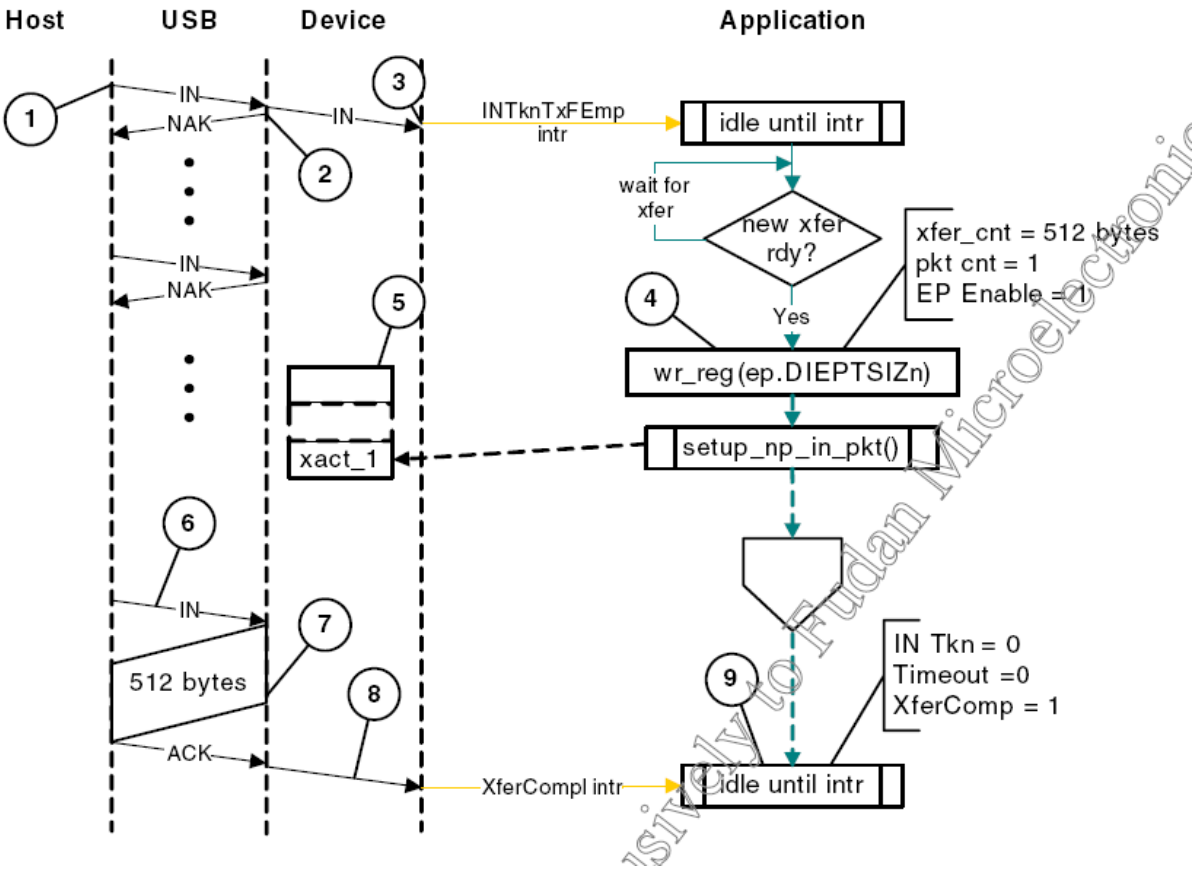
1、使用传输大小和相应数据包计数对DIEPTSIZn寄存器进行设置。

2、使用端点特性对DIEPCTLn寄存器进行设置，并将CNAK和EPENA位置1。

3、发送非零长度数据包时，应用程序必须轮询DTXFSTSn寄存器（其中x为与该端点相关联的FIFO编号）以确定数据FIFO中是否有足够的空间。写入数据前，应用程序也可选用DIEPINTn.TxFEmp位。

示例：批量IN传输

本节描述了将单个批量IN数据包从USB发送到主机的过程。



1、主机发送IN token，试图从端点读取数据。

2、在USB上接收到IN token后，由于发送FIFO中没有待发送的数据，模块返回NAK握手信号。

3、模块产生DIEPINTn.IN Token Rcvd When TxFIFO Empty中断，告知应用程序FIFO中没有数据可发送。

4、发送数据准备好之后，在DIEPTSIZn寄存器中设置传输大小和数据包计数字段。

5、应用程序将长度小于或等于MPSIZ[epnum]的数据包写入TxFIFO。

6、主机重新发送IN token。

7、由于数据已经准备，模块将数据包发出并接受主机的ACK握手。

8、由于XferSize为0，发送结束，设备产生DIEPINTn.XferCompl中断。

9、应用程序相应中断，并通过中断设置判读发送结束。

通用周期性 IN 数据传输

本节介绍典型的周期性 IN 数据传输。

- 应用程序要求

1、通用非周期性 IN 数据传输的应用程序要求 1、2、3、4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。

— 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束

时的一个短数据包。要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

传输大小[epnum] = n × MPSIZ[epnum] + sp（其中n >= 0，且 0 =< sp < MPSIZ[epnum]）

如果 (sp > 0)，数据包计数[epnum] = n + 1；否则，数据包计数[epnum] = n；mc[epnum] = 数据包计数[EPNUM]

— 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。要发送单个零长度数据包：

传输大小[EPNUM] = 0

数据包计数[EPNUM] = 1

mc[EPNUM] = 数据包计数[EPNUM]

1、应用程序一次只能安排一帧的数据传输。

(DIEPTSIZn.MC-1)× DIEPCTLn.MPS =< DIEPTSIZn.XferSiz =< DIEPTSIZn.MC × DIEPCTLn.MPS

DIEPTSIZn.PktCnt = DIEPTSIZn.MC

如果DIEPTSIZn.XferSiz < DIEPTSIZn.MC * DIEPCTLn.MPS，，传输的最后一个数据包为短数据包

3、接收到IN令牌前，应用程序必须将要在帧中发送的完整数据写入到发送FIFO中。在接收到IN令牌时，即使发送FIFO中该帧要发送的数据只差1个双字未写进来，模块也会执行FIFO为空时的操作。

4、当发送FIFO为空时，同步端点上将回复零长度数据包，中断端点上将回复NAK握手信号。

- 内部数据流：

1、应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。

2、应用程序还必须向与该端点相关联的发送FIFO写入必需的数据。

3、应用程序每向发送FIFO写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送FIFO，直到该端点的传输大小变为0。

4、当周期性端点接收到IN令牌时，模块将开始发送FIFO中的数据（如果FIFO中有数据）。如果FIFO中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个IN Tkn Rcvd When TxF Empty中断。同步端点上将回复零长度数据包，中断端点上将回复NAK握手信号。

5、端点的数据包计数会在下列情况下递减 1：

— 对于同步端点，发送一个零长度或非零长度的数据包时

— 对于中断端点，在发送ACK握手信号时递减

— 当传输大小和数据包计数均为0时，将生成该端点的传输完成中断，同时将端点使能位清零。

1、在“周期性帧间隔”（DCFG.PerFrint位控制）内，当模块发现任何在当前帧内应为空的同步IN端点FIFO中的数据还未发送完成时，都会生成一个GINTSTS.incompISOIN中断。

- 应用程序编程顺序：

1、使用端点特性对 DIEPTSIZn寄存器进行编程，并将 CNAK 和 EPENA 位置 1。

2、将需要在下一帧中发送的数据写入发送 FIFO。

3、硬件触发 DIEPINTn.In Token Rcvd When TxF Empty中断表示应用程序尚未将需要发送的全部数据写入发送 FIFO。

4、如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。

5、硬件触发 DIEPINTn.XferCompl 中断时如果 DIEPINTn.In Tkn Rcvd 中断，则表示成功完成同步 IN 传输。读取 DIEPTSIZn 寄存器时得到传输大小 = 0 且数据包计数 = 0，则表示所有数据都已通过 USB 发送完毕。

6、硬件触发 DIEPINTn.XferCompl 中断时无论是否产生DIEPINTn.In Tkn Rcvd中断，只要有TxF Empty中断，都表示成功完成中断 IN 传输。读取 DIEPTSIZn 寄存器时得到传输大小 = 0 且数据包计数 = 0，则表示所有数据都已通过 USB 发送完毕。

7、触发GINTSTS.incomplete Isochronous IN Transfer 中断时如果未产生任何前述中断，则表示在当前帧中模块至少未收到 1 个周期性的 IN 令牌。

不完整的同步 IN 数据传输

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

- 内部数据流

1、符合下列条件之一时，即认为同步 IN 传输未完成：

—模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到GINTSTS.incomplete Isochronous IN Transfer中断。

—应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序检测到DIEPINTn.IN Tkn Rcvd When TxFIFO Empty中断。应用程序可忽略此中断，因为最终这将在周期性帧结束时产生一个未完成同步 IN 传输中断（GINTSTS.incomplete Isochronous IN Transfer）。模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。

2、应用程序必须尽快停止向发送 FIFO 写入数据。

3、应用程序必须将端点的 NAK 位和禁止位置 1。

4、模块会禁止该端点，将禁止位清零并触发端点的Endpoint Disable中断。

- 应用程序编程顺序

1、应用程序可以在任何同步 IN 端点上忽略 DIEPINTn.IN Tkn Rcvd When TxFIFO empty中断，因为最终这将产生一个GINTSTS.incomplete Isochronous IN Transfer中断。

2、硬件触发GINTSTS.incomplete Isochronous IN Transfer中断表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。

3、应用程序必须读取所有同步 IN 端点的DIEPCTLn寄存器来检测存在未完成 IN 数据传输的端点。

4、应用程序必须停止向与这些端点相关联的FIFO写入数据。

5、对 DIEPCTLn寄存器中的下列字段进行编程以禁止端点：

—DIEPCTLn.SetNAK = 1

—DIEPCTLn.Endpoint Disable = 1

1、硬件触发 DIEPINTn.Endpoint Disabled中断表示模块已禁止该端点。此时，应用程序必须清空相关联的发送 FIFO 中的数据。要刷新数据，应用程序必须使用 GRSTCTL 寄存器。

33.11.4.3 Control 传输

本节描述不同类型的控制传输。

控制写传输(SETUP, Data OUT, Status IN)

- 应用程序流程

1、DOEPINTn.SETUP 中断表明收到有效的SETUP 包，在Setup 阶段结束前，必须将DOEPTSIZn.SUPCnt字段写为3以便接收之后的SETUP包。参考“Setup 传输”一节。

2、在SETUP中断前收到最终的SETUP包表明进入 Data OUT 阶段，将模块设置为控制OUT传输。参考“通用非同步 OUT数据传输”一节。

3、控制端点0的一次传输，应用程序可收到64字节。如果应用程序预期收到的字节数大于64，必须重复使能端点来接收另外的的64字节，直到收取到所有数据。

4、最后一帧数据包传输完成后DOEPINTn.Transfer Compl中断置起表明控制传输中的数据阶段完成。

5、Data OUT 阶段完成后，为执行接收到的SETUP指令，应用程序要编程所需的内部模块寄存器。这一步是可选的，取决于收到的SETUP指令。

6、在Status IN 阶段,应用程序要把模块按“通用非周期IN传输”描述进行配置，来实现Status IN 阶段的传输。参考“通用周期性 IN 数据传输”一节

7、硬件置起 DIEPINTn.Transfer Compl中断表明Status IN 阶段的传输结束。

8.重复之前描述的步骤，直到完成全部控制写传输。

控制读传输 (SETUP, Data IN, Status OUT)

1、DOEPINTn.SETUP 中断表明收到有效的SETUP 包，在Setup 阶段结束前，必须将DOEPTSIZn.SUPCnt字段写为3以便接收之后的SETUP包。参考“Setup 传输”一节。

2、在SETUP中断前收到最终的SETUP包表明进入 Data IN 阶段，将模块设置为控制 IN 传输。参考“通用非同步 IN 数据传输”一节。

3、控制端点0的一次传输，应用程序可发送16字节。如果应用程序预期发送的字节数大于64，必须重复使能端点来发送另外的的64字节，直到发送完所有数据。

4、重复以上步骤直到DIEPINTn.Transfer Compl中断置起，该中断标志着控制传输Data IN 阶段结束。

5、为了完成Status OUT阶段的数据OUT传输，应用程序要把模块按“通用非同步 OUT数据传输”描述进行配置。应用程序必须设置DCFG.NZStsOUTHShk 握手字段以适应状态阶段的OUT传输。

6、硬件置起DOEPINTn.Transfer Compl中断表明控制传输的Status OUT阶段结束，这标志着控制读传输完成。

两级控制传输(SETUP/Status IN)

1、DOEPINTn.SETUP中断表明收到有效的SETUP包，在Setup阶段结束前，必须将DOEPTSIZn.SUPCnt字段写为3以便接收之后的SETUP包。参考“Setup 传输”一节。

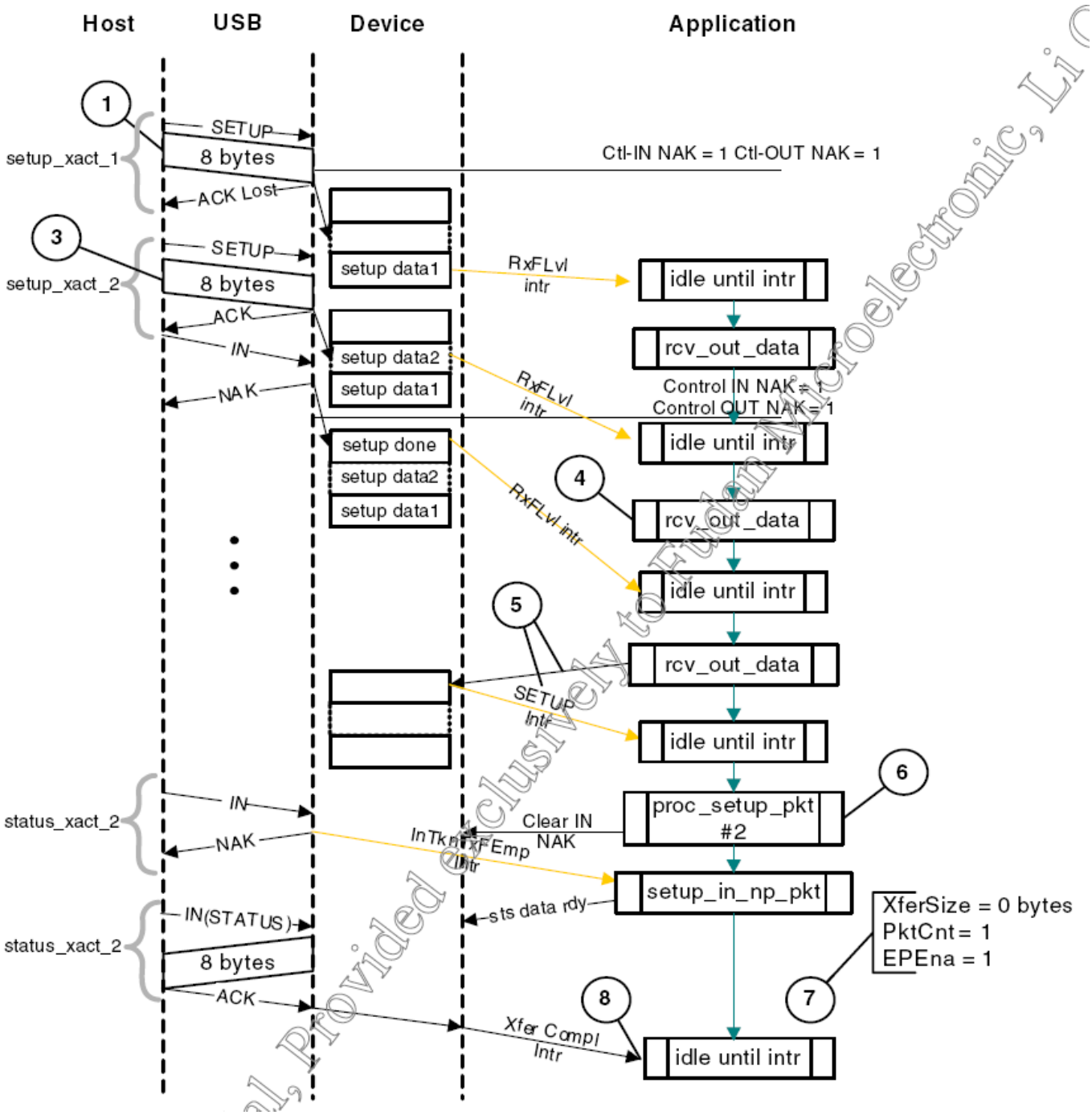
2、解码在SETUP中断前收到最终的SETUP包，如果数据解析指示为两级控制传输，应用程序依据接收到的Setup指令设置模块内的寄存器来执行Setup指令。

3、在Status IN阶段,应用程序要把模块按“通用非周期IN传输”描述进行配置，来实现Status IN 阶段的传输。参考“通用周期性 IN 数据传输”一节

4、硬件置起 DIEPINTn.Transfer Compl中断表明Status IN 阶段的传输结束。

5、重复之前描述的步骤，直到完成全部两级控制传输。

示例：两级控制传输



1、在USB上收到SETUP包#1并写入到接收FIFO，模块应答ACK，HOST未收到ACK检测到超时。

2、接收FIFO中的SETUP包触发GINTSTS.RxFLvl中断，应用程序清空接收FIFO。

3、在USB上收到SETUP包#2并被写入到接收FIFO，模块应答ACK。

4、接收FIFO中的SETUP包触发GINTSTS.RxFLvl中断，应用程序清空接收FIFO。

5、第二个SETUP包之后，主机发送控制IN令牌指示进入Status IN阶段。模块应答NAK并将SETUP阶段结束状态写入接收FIFO，这个写入触发GINTSTS.RxFLvl中断，并清空接收FIFO。模块在读出SETUP阶段结束的DWORD后触发DOEPINTn.SetUp packet中断。

6、在DOEPINTn.SetUp packet中断处理程序中，应用程序解码SETUP包#2为两级控制传输，清除

控制IN EP的 NAK 位（DIEPCTL0.CNAK = 1）。

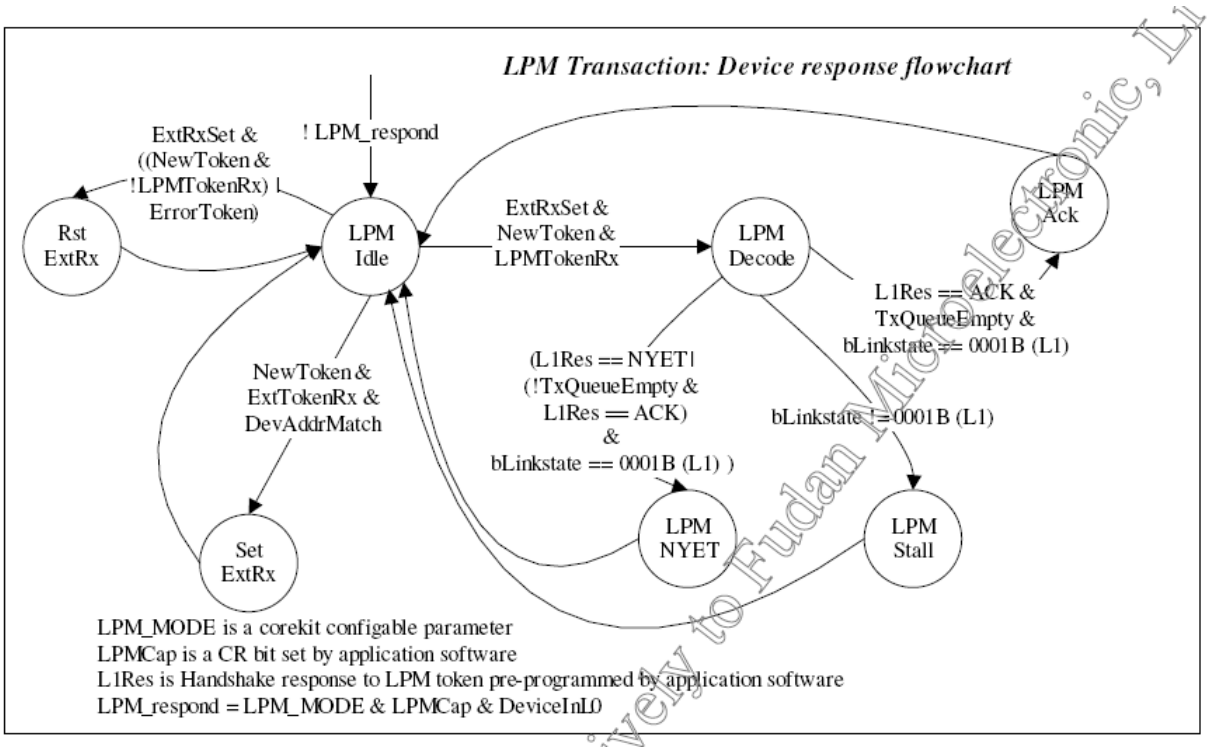
7、IN NAK清除后，模块触发DIEPINTn.INTknTXFEmp中断，在该中断处理程序中使能控制 IN EP，设置DIEPTSIZn.XferSize = 0，DIEPTSIZn.PktCnt = 1。这使模块发送一个0长度的数据包。

8、在Status IN 阶段结束时，模块触发DIEPINTn.XferCompl中断通知应用程序。

33.11.5 LPM 编程

1、要使能LMP功能，需设置GLPMCFG.LPMCap = 1。之后可以设置一些可选的低功耗选项，GLPMCFG.EnblSlpM（使能SLEEP模式），PCGCCTL.Enbl_L1Gating（使能PHY的时钟门控），GLPMCFG.HIRD_Thres（使能浅低功耗模式）。

1、修改GLPMCFG .AppL1Res字段来配置对主机的LPM事务指令的返回握手，可配置为ACK或NYET。



3、LPM事务包含以下3个包：

- 主机发出带有EXT PID 0000B的令牌包
- 主机发出带有SubPID 0011B (LPM token)的令牌包
- 器件发出握手包

4、如果存在如下状态，模块不会在USB上回复主机（ERROR response）也不会通知应用程序。

- 令牌包有PID error 或 CRC5 error
- GLPMCFG.LPMCap位没有置1
- 器件在Suspend mode下（L2）
- 器件还没有 reset 或没有枚举结束（L3）
- 器件已经在Sleep mode下（L1）

5、没有以上情况模块会通知收到新的Token，并且对EXT PID令牌和 LPM 令牌做出响应。

6、器件收到EXT PID令牌包，会将EXT PID和之前接收到的PID作比较，并对接收到的地址域和枚举时接收的器件ID做比较。

- 如果地址和PID都匹配，并且ExtRxSet = 0，则设置ExtRxSet = 1，且等待下一帧token。
- 如果地址和PID不匹配，设置ExtRxSet = 0，忽略当前令牌包，不会在USB上回复主机（ERROR response）也不会通知应用程序。

7、只有ExtRxSet = 1时，器件才会对SubPID 0011B (LPM token)做出响应。ExtRxSet = 1表明成功接收到EXT PID令牌包，之后没有接收到其他令牌包，并且没有错误状态。

- 如果EXT PID令牌包之后收到重复的EXT PID令牌包，器件按上一条处理。
- 如果EXT PID令牌包之后收到其它令牌包，设置ExtRxSet = 0，按正常流程回应收到的令牌包。
- 如果收到正常的LPM token，设置LPMTokenRx =1，解码LPM token，并相应的回复STALL,NYET或ACK。

8、STALL握手，在没有ERROR发生，接收到的LPM token的bLinkState不是SLEEP(L1) ，即该字段不是0001B，器件会回复STALL握手。之后会更新如下状态寄存器：

- GLPMCFG.CoreL1Res字段设置为STALL, GLPMCFG.SlpSts = 0
- GLPMCFG.HIRD 和 GLPMCFG.bRemoteWake用接收到的LPM token里的数据做做更新。
- 不论GINTMSK.LPM_IntMsk为何值，GINTSTS.LPM_Int = 1，表明收到LPM事务。
- 如果GINTMSK.LPM_IntMsk = 0，应用程序将相应该中断。

模块回复了STALL握手之后，LPM事务响应单元回复到IDLE状态，等待接收新的LPM事务。

9、NYET握手，在没有ERROR，也没有回复STALL的时候，器件在如下情况下回复NYET握手：

- AppL1Res字段设置为NYET。
- AppL1Res字段设置为 ACK，但是还有一条或更多的传输队列没有完成。

回复NYET握手后，状态寄存器更新如上节STALL握手，除了GLPMCFG.CoreL1Res字段设置为NYET而不是STALL。

LPM事务响应单元回复到IDLE状态，等待接收新的LPM事务。

1、 ACK握手，在没有ERROR，也没有回复STALL和NYET的时候，AppL1Res字段的设置为 ACK，器件回复ACK。LPM事务响应单元触发L0-to-L1事务单元，LPM事务响应单元回复到IDLE状态，等待接收新的LPM事务。

11、L0-to-L1状态转换

- 等待 Token 重试时间，TL1TokenRetry = 8 µs + 0.5 µs（8 µs是LPM指定的值，0.5 µs是冗余的等待时间）。如果在TL1TokenRetry时间内收到主机发送的任何传输都放弃继续向L1状态转移
- 初始SLEEP状态更新，TL1TokenRetry时间结束后，更新如下状态寄存器：

a) GLPMCFG.CoreL1Res = ACK and GLPMCFG.SlpSts = 1

b) GLPMCFG.HIRD 和 GLPMCFG.bRemoteWake用接收到的LPM token里的数据做做更新。

c) 不论GINTMSK.LPM_IntMsk为何值，GINTSTS.LPM_Int = 1，表明收到LPM事务。

d) 如果GINTMSK.LPM_IntMsk = 0，应用程序将相应该中断。

- 转换到SLEEP状态

1) 切换UTMI PHY的控制信号到 L1 sleep状态。

2) PCGCCTL.Enbl_L1Gating =1则模块内部关闭UTMI PHY的时钟。

3)状态转换时间不要超过TL1TransitionDev = 1µs（临界0.5 µs = 10 × (8 + 1 + 0.5)超过LPM ECN的指定值）

12、 L1状态事件

- 进入L1状态后，参数可配置的减计数器启动（TL1Residency = 50 µs）。计数结束后L1ResumeOK状态置位。TL1Residency是预期的PHY信号状态稳定时间，这个延时时间后信号状态稳定，模块开始检测主机发起的Resume信号。
- 在GLPMCFG.L1ResumeOK 置位后，器件开始检测USB上信号状态，收到RESET信号或Resume信号后，器件必须退出L1状态。

13、器件发起的L1退出

- 只要在L1 状态，器件就可以通过设置DCTL.RmtWkUpSig = 1发起L1退出。在设置DCTL.RmtWkUpSig字段发起L1退出前必须先读取判断GLPMCFG.bRemoteWake字段和GLPMCFG. .L1ResumeOK 字段状态，若 GLPMCFG.bRemoteWake =1 且GLPMCFG. .L1ResumeOK = 1，程序先清除GLPMCFG.HIRD_Thres[4]，GLPMCFG.EnblSlp和PCGCCTL.Enbl_L1Gating的设置，再设置DCTL.RmtWkUpSig = 1发起L1退出。
- 之后模块立即改变UTMI PHY的控制信号来引导从SLEEP状态的退出。
- 模块设置GLPMCFG.CoreL1Res = ERROR，GLPMCFG.L1ResumeOK=0。
- 模块设置GLPMCFG.HIRD 和 GLPMCFG.bRemoteWake为复位值。

- 最后，模块清除GLPMCFG.SlpSts位，置位GINTSTS.WkUpInt中断标志，如果GINTMSK.WkUpIntMsk = 0，模块发起中断请求。

14、主机/HUB发起的L1退出

进入L1状态后，TL1Residency =_50 μs的计时结束（状态位GLPMCFG.L1ResumeOK = 1），器件开始监听线上状态。

如果在Jresume期间有有效的J-to-K的信号状态转换，器件即开始L1退出流程，退出方式和器件发起的L1退出流程相同。

15、RESET信号发起的L1退出

进入L1状态后，甚至在TL1Residency =_50 μs的计时期间，器件如果监测到Jreset (2.5 μs)时间的J-to-SE0信号变化，GINTSTS.USBRst置位发起L1退出。如果GINTMSK.USBRstMsk = 0，软件将响应改中断。

34 I/O 端口

34.1 概述

I/O 端口的主要功能特性：

- GPIO 引脚最高耐 5.5V 电压
- GPIO 数字输入具有施密特特性
- 部分 GPIO 输入支持模拟滤波
- 部分 GPIO 输入支持数字滤波
- GPIO 可配置为上拉、开漏输出
- Sleep/DeepSleep 模式下保持状态

34.2 引脚类型

FM33LC0XX 主要有三种类型的 GPIO 引脚，其中大部分引脚支持输入输出、数字外设功能、模拟外设通道、可控上拉电阻、可控开漏输出功能；强驱动引脚除了以上功能外，具有增强的推挽输出驱动能力；而真开漏引脚只有 NMOS 驱动，没有 PMOS 驱动，无法对外驱动逻辑高电平。

PB12 是 5V tolerant 引脚，可以承受高于芯片电源电压的输入信号。

34.2.1 GPIO，输入输出使能，可控上拉电阻，可控开漏输出

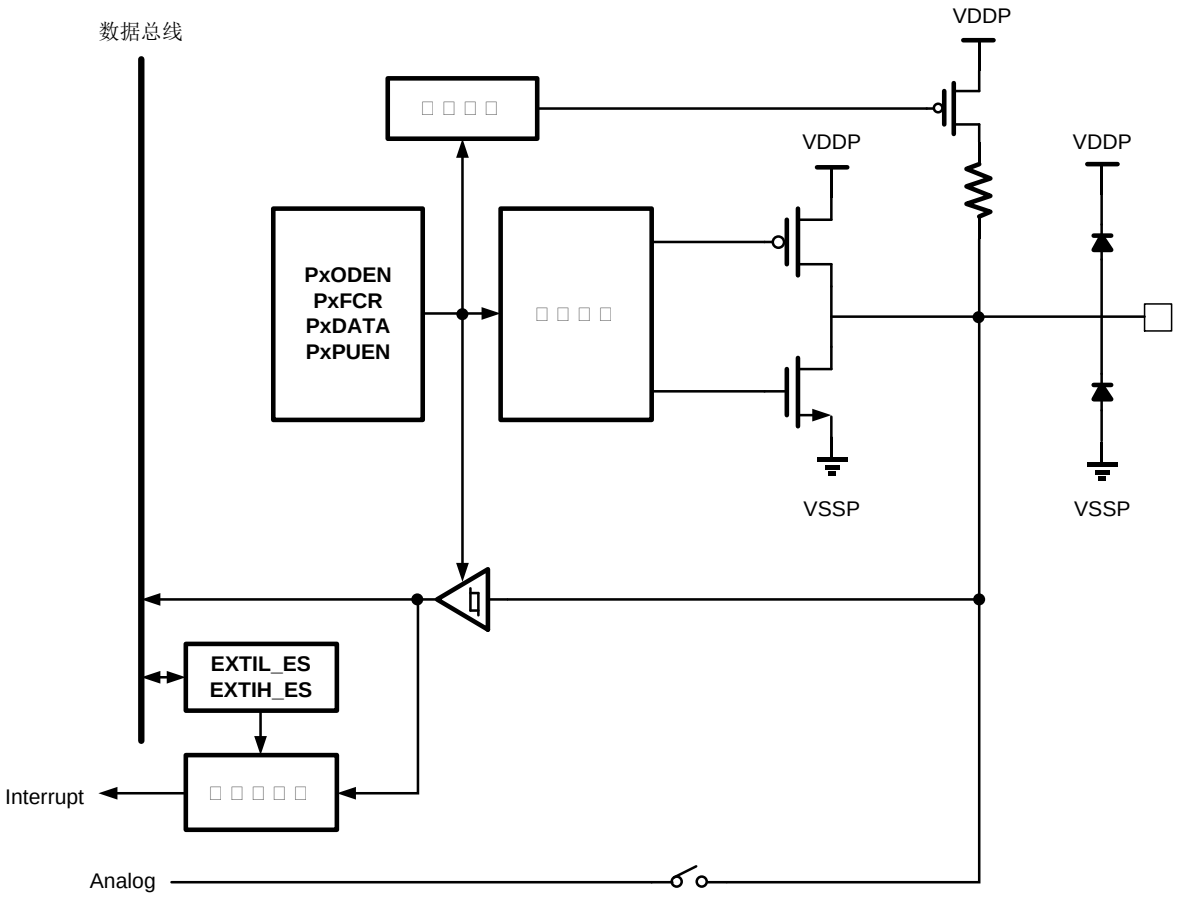


图 34-1 普通 GPIO 结构框图

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表34-1GPIO功能逻辑定义表

34.2.2 GPIO，输入输出使能，真开漏输出（PA11、PA12）

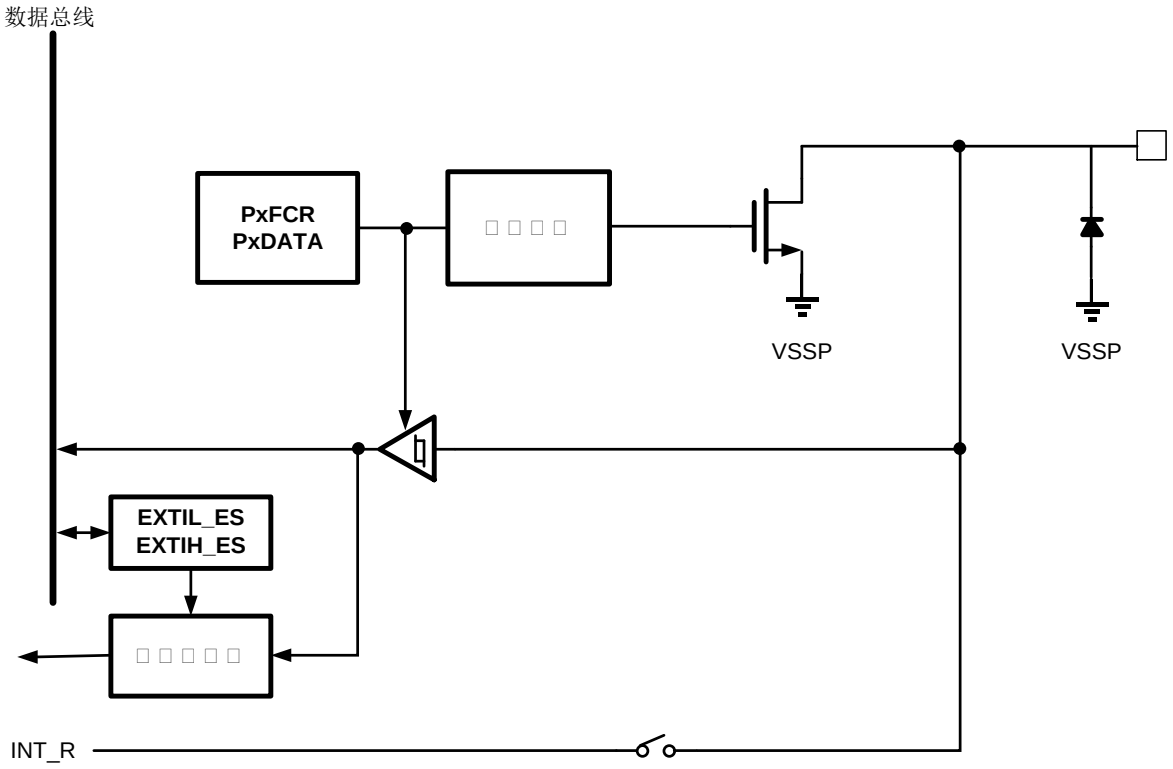


图 34-2 真开漏 GPIO 结构框图

上述IO的控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	x	0/1	0	0	1	0/1
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	x		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表34-2真开漏IO功能逻辑定义表

34.2.3 GPIO，输入输出使能，2 个可控上拉电阻，可控开漏输出（仅 7816 数据口）

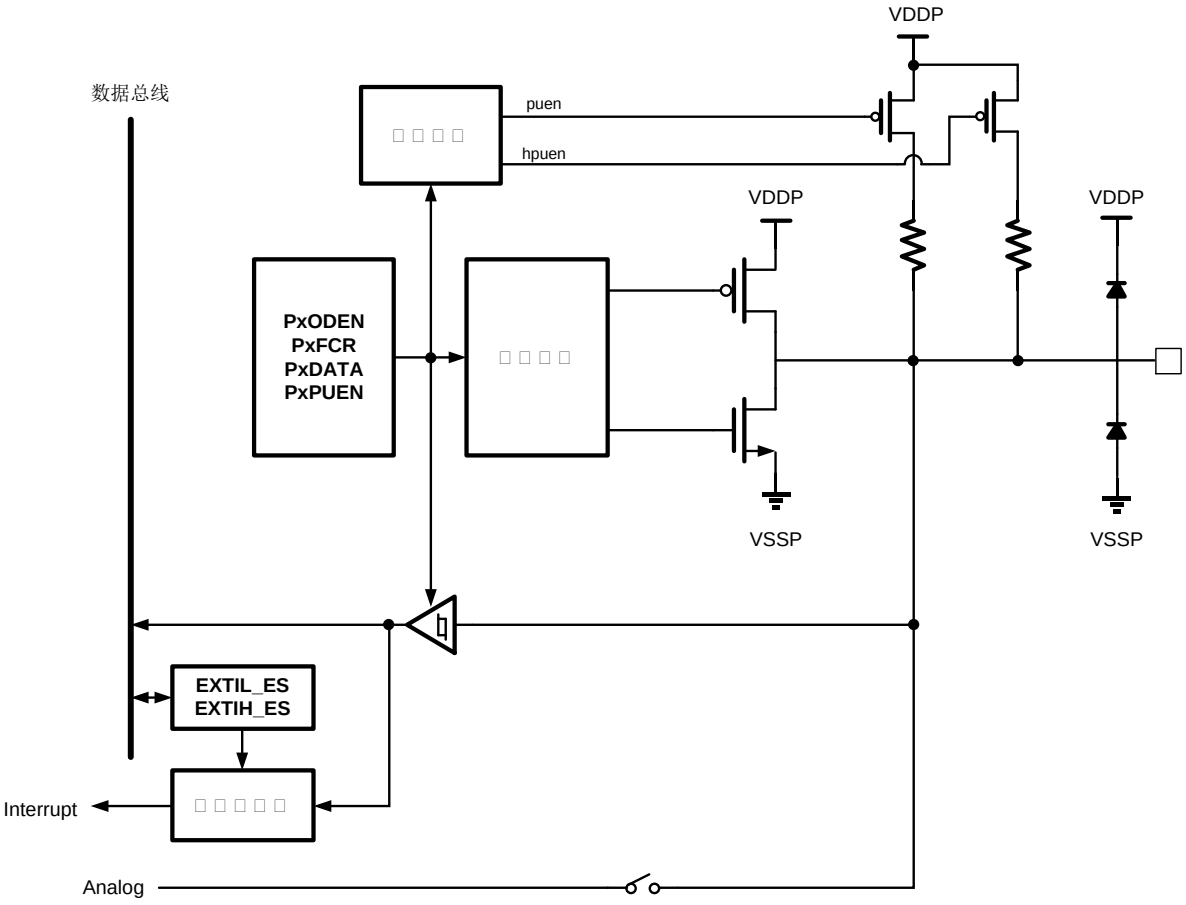


图 34-3 普通 GPIO（两路上拉）结构框图

上述IO的寄存器控制逻辑与其他GPIO相同，并联上拉控制（强上拉）仅由7816模块自动控制。

34.2.4 GPIO，输入输出使能，可控上拉电阻，可控开漏输出，HV tolerant（PB12）

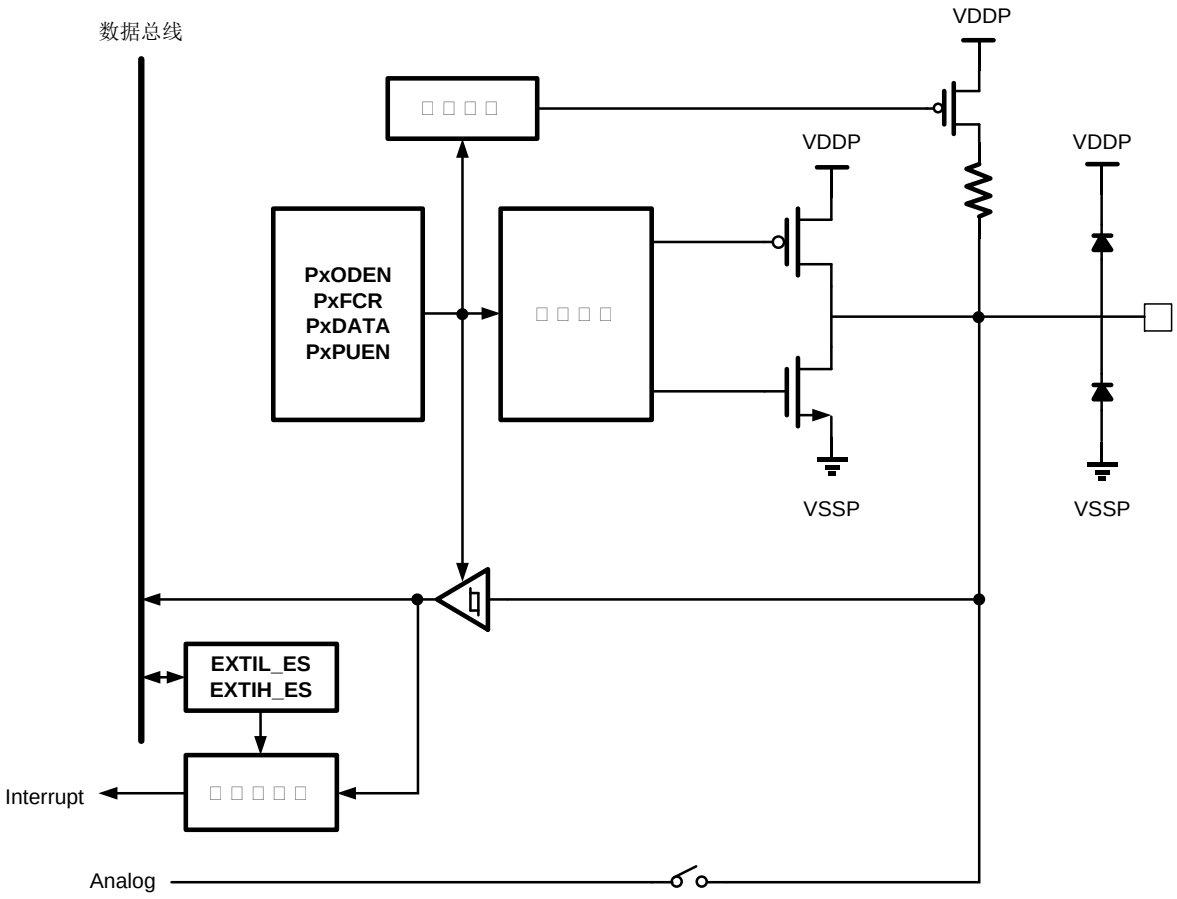


图 34-45V-tolerant GPIO 结构框图

此种GPIO可以承受高于电源电压的输入，比如电源为3V时，可以承受5.5V输入，而不会引起漏电。此IO的主要用途是用作VBUS接入唤醒。

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	

				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表34-3GPIO功能逻辑定义表

34.3 IO 端口功能定义

芯片大部分引脚为数模混合IO，每个通用GPIO都有4bit控制寄存器：FCR[1:0]、PUEN、ODEN，其中FCR用于选择IO引脚功能，定义如下：

FCR: Function Control Register	PAD function
00	GPIO input
01	GPIO output
10	Digital Function（数字外设功能）
11	Analog

表34-4FCR定义表

34.3.1 GPIO 输入

当某个 GPIO 被配置成输入功能，并且对应的输入使能寄存器被置位时：

- 输出驱动缓冲器被关闭
- 施密特触发器使能
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- PxDIN 寄存器直接反应 IO 上的电平状态

34.3.2 GPIO 输出

当某个 GPIO 被配置成输出功能，并且对应的输出使能寄存器被置位时：

- 输出驱动缓冲器使能
  - 开漏输出模式（PxODEN=1）：输出 0 时 IO 驱动低电平，输出 1 时 IO 关闭驱动缓冲器
  - 推挽输出模式（PxODEN=0）：输出 0 时 IO 驱动低电平，输出 1 时 IO 驱动高电平
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 软件读取 PxDIN 寄存器能够获得 IO 上的电平状态
- 软件读取 PxDO 寄存器获得上次写入的值

34.3.3 数字外设功能

当某个 GPIO 被配置成数字外设功能：

- IO 的输入或输出方向由所连接的外设功能决定
- 由 PxODEN 控制输出时是开漏输出还是推挽输出
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 软件读取 PxDIN 寄存器能够获得 IO 上的电平状态

部分引脚支持多个数字外设功能，则还需要额外的控制寄存器（ADT_AFSELx）来区分。

支持多个数字外设功能的引脚有：

GPIO	数字功能 1 PxDFS[x]=0	数字功能 2 PxDFS[x]=1	Additional AFSEL
PA8	SPI1_SSN	LPTI	PADFS[8]
PA9	SPI1_SCK	LPTO	PADFS[9]
PA13	UART0_RX	LPUART0_RX	PADFS[13]
PA14	UART0_TX	LPUART0_TX	PADFS[14]
PB0	SPI1_MISO	UART1_RX	PBDFS[0]
PB1	SPI1_MOSI	UART1_TX	PBDFS[1]
PB2	UART2_RX	ATIM_CH1N	PBDFS[2]
PB3	UART2_TX	ATIM_CH2N	PBDFS[3]
PB8	SPI1_SSN	ATIM_CH3N	PBDFS[8]
PB9	SPI1_SCK	GPT0_ETR	PBDFS[9]
PB10	SPI1_MISO	GPT0_CH1	PBDFS[10]
PB11	SPI1_MOSI	GPT0_CH2	PBDFS[11]
PB12	FOUT1	ATIM_ETR	PBDFS[12]
PB13	UART1_RX	LPUART1_RX	PBDFS[13]
PB14	UART1_TX	LPUART1_TX	PBDFS[14]
PC2	U7816_CLK	GPT0_CH1	PCDFS[2]
PC3	U7816_IO	GPT0_CH2	PCDFS[3]
PD11	FOUT0	ATIM_BKR	PDDFS[11]

表34-5多个数字外设功能选择表

34.3.4 模拟功能

当某个 GPIO 被配置成模拟功能：

- 输出缓冲器关闭
- 数字输入功能关闭

- 上拉电阻关闭
- 软件读取 PxDIN 返回 0
- IO 模拟通道被连接到特定的模拟外设上
- 如果一个 IO 同时连接到多个模拟外设，则多个模拟外设在同一时刻只能使能其中一个

34.3.5 使用外部晶体引脚

FM33LC0XX 支持外接 32768Hz 晶体和 4~32MHz 高频晶体。

其中 PC11 和 PC12 默认为 GPIO，配置为模拟功能之后作为 XTHFIN 和 XTHFOUT 外接高频晶体。

PD9 和 PD10 默认为模拟功能，外接 32768Hz 晶体；在关闭 XTLF 功能的情况下，也可以配置为 GPIO 使用。

如果要使用外部 32K 时钟输入，则仅需保留 PD10 的模拟功能（XT32KI），并从 XT32KI 灌入 32K 时钟，此时 PD9 可以配置为 GPIO 使用。

34.4 NRST 引脚

NRST 引脚用于产生芯片复位，带有数字滤波，滤波采用 LPOSC 时钟工作。当休眠模式下芯片关闭 LPOSC 时，NRST 输入低电平将强制使能 LPOSC，如果滤波有效则复位芯片，如果滤波无效则在滤波结束后自动关闭 LPOSC。有效滤波长度是 2~3 个 LPOSC 周期之间，即 60~90us 的典型范围。

如果芯片处于低功耗模式，NRST 有效也会使芯片退出低功耗模式。

34.5 WKUPx 引脚

FM33LC0XX 有 8 个 WKUP 引脚，能够将芯片从 Sleep/DeepSleep 模式下唤醒，即使片上振荡器都停止工作，WKUP 仍能唤醒芯片。

WKUPx 引脚输入上升沿或者下降沿（软件配置）能够将芯片从休眠模式下唤醒。为了使能此功能，需将对应引脚配置为 GPIO 输入功能，并且相应的 PINWKEN 置位，注意 PAD 内部带有上拉电阻，如果配置为上升沿唤醒，则必须关闭上拉电阻。

每个支持 WKUP 功能的 IO 都带有大约 100ns 的片内模拟滤波，能够滤除输入信号上的毛刺，避免误触发。

Sleep/DeepSleep 模式下，使能了的 WKUPx 引脚上任何大于 100ns 的脉冲都会触发芯片唤醒。

WKUPx 功能使用时需要注意外部引脚输入的初始状态。在使能 WKUP 时，可能由于初态的关系导致虚假的唤醒事件，软件应注意识别并处理。

使用 WKUP 功能时，必须将对应引脚的 FCR 寄存器配置为 00（GPIO 输入），按需要设置唤醒边沿（PINWKSELx）并使能 PINWKENx 寄存器。当某个 WKUPx 引脚上产生唤醒事件后，PMU 模块内部的唤醒源标志查询寄存器内对应的 bit 位将会自动置位。

34.6 外部引脚中断（EXTI）

34.6.1 功能说明

FM33LC0XX 的 4 组 GPIO（A~D）最多可以产生 16 个 EXTI 中断，每组 GPIO 分别可以产生 4 个 EXTI 中断标志，最终所有的 EXTI 中断汇总到 NVIC 的#46 入口。

中断标志和引脚对应关系如下表：

GPIO	EXTI输入选择	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]
PA8~PA11	EXTI_ASEL[5:4]	EXTI[2]
PA12~PA15	EXTI_ASEL[7:6]	EXTI[3]
PB0~PB3	EXTI_BSEL[1:0]	EXTI[4]
PB4~PB7	EXTI_BSEL[3:2]	EXTI[5]
PB8~PB11	EXTI_BSEL[5:4]	EXTI[6]
PB12~PB15	EXTI_BSEL[7:6]	EXTI[7]
PC0~PC3	EXTI_CSEL[1:0]	EXTI[8]
PC4~PC7	EXTI_CSEL[3:2]	EXTI[9]
PC8~PC11	EXTI_CSEL[5:4]	EXTI[10]
PC12	-	EXTI[11]
PD0~PD3	EXTI_DSEL[1:0]	EXTI[12]
PD4~PD7	EXTI_DSEL[3:2]	EXTI[13]
PD8~PD11	EXTI_DSEL[5:4]	EXTI[14]
PD12	-	EXTI[15]

表34-6外部引脚中断配置

EXTI_xSEL 寄存器用于选择某个 IO 接入 EXTI 通道，EXTI 模块可以配置是否对输入信号进行数字滤波。

数字滤波的实现方法是由IO采样时钟连续采样到3次相同电平才认为是合法电平输入，如下图所示。

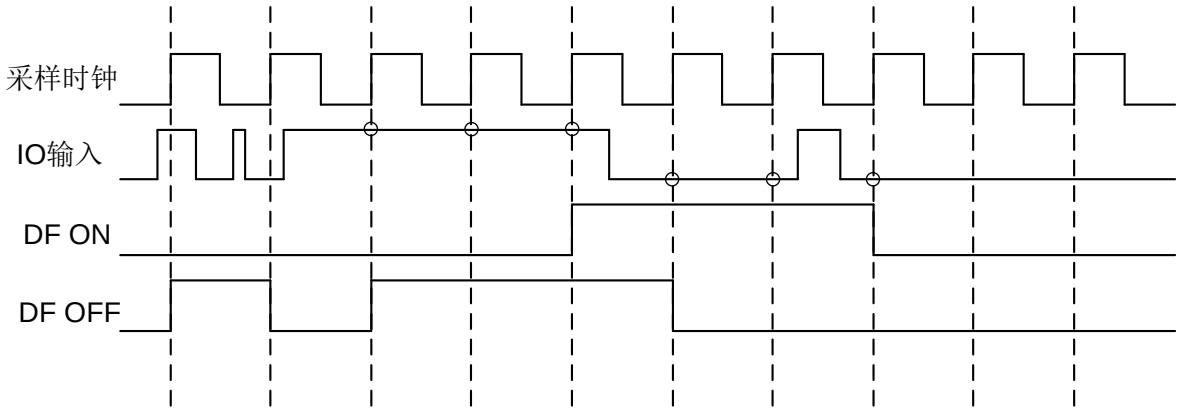


图 34-5 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

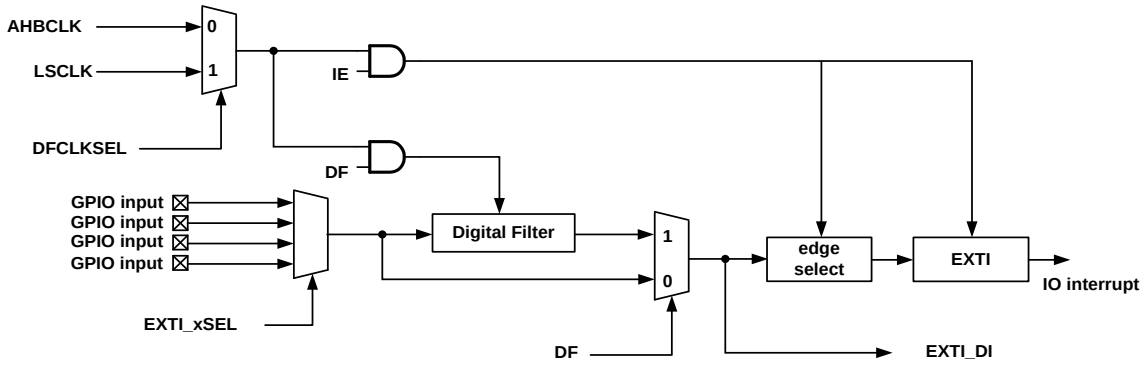


图 34-6 EXTI 信号输入示意图

用户应根据引脚功能需要使能或禁止数字滤波功能，使能数字滤波后，将根据 AHBCLK 频率不同，对 IO 输入信号引入不同的采样延迟。经过数字滤波后的输出信号，软件也可以在 EXTI_DI 寄存器读到。

EXTI还可以配置输入信号的有效边沿，支持上升沿、下降沿、上升下降沿触发中断，或者禁止EXTI中断触发，由EXTI_EDS寄存器配置。

34.6.2 应用指南

如需在 Sleep/DeepSleep 模式下启动 EXTI 中断唤醒功能，推荐按照如下步骤进行操作：

- 关闭所有 EXTI 使能
- 配置 SYSCLKSEL 寄存器（0x0x4000020C）的 EXTICKSEL 位为 1，选择 LSCLK 进行 EXTI 采样
- 根据需要打开或关闭 EXTI 数字滤波使能
- 配置相应 GPIO 为输入
- 配置 EXTI_SEL 寄存器选择对应的 IO
- 置位 OPCCON1.EXTICKE，打开 EXTI 工作时钟使能

- 等待至少 4 个 LSCLK 周期
- 配置 EXTI_EDS 触发边沿选择，使能所需的 EXTI 中断
- 正常进入 Sleep 模式

芯片上电后默认关闭所有 EXTI，同时默认的引脚中断采样时钟是系统时钟 APBCLK。如果用户使用系统时钟产生 EXTI，推荐流程如下：

- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 OPCCON1.EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 APBCLK 周期
- 配置 EXTI_EDS 触发边沿选择，使能所需的 EXTI 中断

如果希望使用低速的 LSCLK 来产生 EXTI，推荐流程如下：

- 将 EXTI 采样时钟配置为 LSCLK
- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 OPCCON1.EXTICKE，打开 EXTI 采样时钟使能
- 等待至少 4 个 LSCLK 时钟周期
- 配置 EXTI_EDS 触发边沿，使能所需的 EXTI 中断

34.7 快速 GPIO 输出

FM33LC0XX 可以通过 set-reset 功能快速改变每个 GPIO 的输出数据（bitwise operation），从而提高 IO 输出效率，特别是可以提高 read-modify-write 操作的效率和可靠性（atomic）。方法是每个 GPIO 组的输出数据寄存器都有 2 组 set-reset 映射虚拟地址，对 set 寄存器特定 bit 写 1 可以置位对应的数据寄存器的 bit 位，对 reset 寄存器特定地址写 1 可以清除对应的数据寄存器的 bit 位。

34.8 寄存器

offset 地址	名称	符号
GPIOA(模块起始地址:0X40000C00)		
0x00000000	GPIOA 输入使能寄存器 （GPIOA Input Enable Register）	GPIOA_INEN
0x00000004	GPIOA 上拉使能寄存器 （GPIOA Pull-Up Enable Register）	GPIOA_PUEN
0x00000008	GPIOA 开漏使能寄存器 （GPIOA Open-Drain Enable Register）	GPIOA_ODEN
0x0000000C	GPIOA 功能选择寄存器 （GPIOA Function Control Register）	GPIOA_FCR
0x00000010	GPIOA 输出数据寄存器 （GPIOA Data Output Register）	GPIOA_DO
0x00000014	GPIOA 输出数据置位寄存器 （GPIOA Data Set Register）	GPIOA_DSET
0x00000018	GPIOA 输出数据复位寄存器 （GPIOA Data Reset Register）	GPIOA_DRST
0x0000001C	GPIOA 输入数据寄存器 （GPIOA Data Input Register）	GPIOA_DIN
0x00000020	GPIOA 额外数字功能寄存器 （GPIOA Digital Function Select）	GPIOA_DFS
0x00000028	GPIOA 模拟开关使能寄存器 （GPIOAAnalog channel Enable Register）	GPIOA_ANEN
GPIOB(模块起始地址:0X40000C40)		
0x00000000	GPIOB 输入使能寄存器 （GPIOB Input Enable Register）	GPIOB_INEN
0x00000004	GPIOB 上拉使能寄存器 （GPIOB Pull-Up Enable Register）	GPIOB_PUEN
0x00000008	GPIOB 开漏使能寄存器 （GPIOB Opeb-Drain Enable Register）	GPIOB_ODEN
0x0000000C	GPIOB 功能选择寄存器 （GPIOB Function Control Register）	GPIOB_FCR
0x00000010	GPIOB 输出数据寄存器 （GPIOB Data Output Register）	GPIOB_DO
0x00000014	GPIOB 输出数据置位寄存器 （GPIOB Data Set Register）	GPIOB_DSET
0x00000018	GPIOB 输出数据复位寄存器 （GPIOB Data Reset Register）	GPIOB_DRST
0x0000001C	GPIOB 输入数据寄存器 （GPIOBData Input Register）	GPIOB_DIN
0x00000020	GPIOB 额外数字功能寄存器 （GPIOB Digital Function Select）	GPIOB_DFS
0x00000028	GPIOB 模拟开关使能寄存器 （GPIOB Analog channel Enable Register）	GPIOB_ANEN
GPIOC(模块起始地址:0X40000C80)		
0x00000000	GPIOC 输入使能寄存器 （GPIOC Input Enable Register）	GPIOC_INEN
0x00000004	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUEN
0x00000008	GPIOC 开漏使能寄存器 （GPIOC Opeb-Drain Enable Register）	GPIOC_ODEN

offset 地址	名称	符号
0x0000000C	GPIOC 功能选择寄存器 （GPIOC Function Control Register）	GPIOC_FCR
0x00000010	GPIOC 输出数据寄存器 （GPIOC Data Output Register）	GPIOC_DO
0x00000014	GPIOC 输出数据置位寄存器 （GPIOC Data Set Register）	GPIOC_DSET
0x00000018	GPIOC 输出数据复位寄存器 （GPIOC Data Reset Register）	GPIOC_DRST
0x0000001C	GPIOC 输入数据寄存器 （GPIOC Data Input Register）	GPIOC_DIN
0x00000020	GPIOC 额外数字功能寄存器 （GPIOC Digital Function Select）	GPIOC_DFS
0x00000028	GPIOC 模拟开关使能寄存器 （GPIOC Analog channel Enable Register）	GPIOC_ANEN
GPIOD(模块起始地址:0X40000CC0)		
0x00000000	GPIOD 输入使能寄存器 （GPIOD Input Enable Register）	GPIOD_INEN
0x00000004	GPIOD 上拉使能寄存器 （GPIOD Pull-Up Enable Register）	GPIOD_PUEN
0x00000008	GPIOD 开漏使能寄存器 （GPIOD Opeb-Drain Enable Register）	GPIOD_ODEN
0x0000000C	GPIOD 功能选择寄存器 （GPIOD Function Control Register）	GPIOD_FCR
0x00000010	GPIOD 输出数据寄存器 （GPIOD Data Output Register）	GPIOD_DO
0x00000014	GPIOD 输出数据置位寄存器 （GPIOD Data Set Register）	GPIOD_DSET
0x00000018	GPIOD 输出数据复位寄存器 （GPIOD Data Reset Register）	GPIOD_DRST
0x0000001C	GPIOD 输入数据寄存器 （GPIOD Data Input Register）	GPIOD_DIN
0x00000020	GPIOD 额外数字功能寄存器 （GPIOD Digital Function Select）	GPIOD_DFS
0x00000028	GPIOD 模拟开关使能寄存器 （GPIOD Analog channel Enable Register）	GPIOD_ANEN
GPIO (模块起始地址:0X40000D00)		
0x00000000	EXTI 输入选择寄存器 （External Interrupt input Select Register）	GPIO_EXTISEL
0x00000004	EXTI 边沿选择和使能寄存器 （External Interrupt Edge Select and Enable Register）	GPIO_EXTIEDS
0x00000008	EXTI 数字滤波控制寄存器 （External Interrupt Digital Filter Register）	GPIO_EXTIDF
0x0000000C	EXTI 中断标志寄存器 （External Interrupt and Status Register）	GPIO_EXTIISR
0x00000010	EXTI 输入信号寄存器 （External Interrupt Data Input Register）	GPIO_EXTIDI
0x00000100	FOUT 配置寄存器 （Frequency Output Select Register）	GPIO_FOUTSEL
0x00000200	WKUP 使能寄存器	GPIO_PINWKEN

offset 地址	名称	符号
	（Wakeup Enable Register）	

34.8.1 GPIOx 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000000 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	INEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	INEN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	INEN	GPIO 输入使能控制 (Portx Input Enable) 0：关闭输入使能 1：打开输入使能

34.8.2 GPIOx 上拉使能寄存器（GPIOx_PUEN）

名称	GPIOx_PUEN(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000004 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PUEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PUEN[7:0]							

位权限	R/W-0000 0000

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	PUEN	GPIO 上拉控制 (Portx Pull-Up Enable) 0：关闭上拉 1：使能上拉

34.8.3 GPIOx 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000008 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ODEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ODEN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	ODEN	GPIO 开漏输出使能 (Portx Open-Drain Enable) 0：关闭开漏输出 1：使能开漏输出

34.8.4 GPIOx 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x0000000C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15FCR		Px14FCR		Px13FCR		Px12FCR	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Px11FCR		Px10FCR		Px9FCR		Px8FCR	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	Px7FCR		Px6FCR		Px5FCR		Px4FCR	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Px3FCR		Px2FCR		Px1FCR		Px0FCR	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

位号	助记符	功能描述
31:30	Px15FCR	Px[15]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
29:28	Px14FCR	Px[14]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
27:26	Px13FCR	Px[13]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
25:24	Px12FCR	Px[12]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
23:22	Px11FCR	Px[11]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
21:20	Px10FCR	Px[10]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
19:18	Px9FCR	Px[9]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
17:16	Px8FCR	Px[8]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
15:14	Px7FCR	Px[7]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出

位号	助记符	功能描述
		10：Digital function 11：Analog function
13:12	Px6FCR	Px[6]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
11:10	Px5FCR	Px[5]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
9:8	Px4FCR	Px[4]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
7:6	Px3FCR	Px[3]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
5:4	Px2FCR	Px[2]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
3:2	Px1FCR	Px[1]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
1:0	Px0FCR	Px[0]引脚功能选择 (Portx Function Control Register) 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function

34.8.5 GPIOx 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000010 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DO[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DO[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DO	GPIO output data register

34.8.6 GPIOx 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000014 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DSET[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DSET[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DSET	GPIO output data set register 举例：向 GPIOA_DSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。 GPIOA _DSET/GPIOB_DSET 为 16 位； GPIOC_DSET/GPIOD_DSET 为 13 位

34.8.7 GPIOx 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST(x=A,B,C,D)

Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000018 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DRESET[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DRESET[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DRESET	GPIO output data reset register 举例：向 GPIOA_DRST 写 0x0000_8000，则 PADO[15]清零，其余位保持不变 GPIOA _DRESET/GPIOB_DRESET 为 16 位； GPIOC_DRESET/GPIOD_DRESET 为 13 位

34.8.8 GPIOx 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN(x=A,B,C,D)							
offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x0000001C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIN[15:8]							
位权限	R-xxxx xxxx							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIN[7:0]							
位权限	R-xxxx xxxx							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DIN	Portx input data register 此寄存器仅占用地址空间，无物理实现。软件读此寄存器直接返回引脚输入信号，芯片并不对引脚输入进行锁存

34.8.9 GPIOx 额外数字功能寄存器（GPIOx_DFS）

名称	GPIOx_DFS(x=A,B,C,D)							
Offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000020 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DFS[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DFS[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DFS	Portx Digital Function Select 对于具有多个数字外设功能的引脚，通过 PxDFS 寄存器可以选择使用哪个外设功能。 注意，对于不同的 IO 分组，有效的寄存器位置是不一样的，详细定义请参考**表 34-5**

34.8.10 GPIOx 模拟开关使能寄存器（GPIOx_ANEN）

名称	GPIOx_ANEN(x=A,B,C,D)							
offset	PA,y=0 PB,y=1 PC,y=2 PD,y=3 0x00000028 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ANEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ANEN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	ANEN	PortX 模拟开关使能 (Portx Analog channel Enable) 1：使能 IO 模拟开关 0：关闭 IO 模拟开关 *注：支持模拟开关的 IO 有* *PA6/PA7/PA13/PA14/PA15* *PB0/PB1/PB13/PB14/PB15* *PC0/PC1/PC6/PC7/PC8/PC9/PC10* *PD0/PD1/PD2/PD11/PD12* *对应以上 IO 的 PxANEN 寄存器有效；其余寄存器无意义。*

34.8.11 EXTI 输入选择寄存器（GPIO_EXTISEL）

名称	GPIO_EXTISEL							
Offset	0x00000100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		DSEL					
位权限	U-0		R/W-00 0000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		CSEL					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BSEL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ASEL							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:30	-	RFU：未实现，读为 0
29:24	DSEL	PortD EXTI 中断输入选择 (External Interrupt PortD Select) EXTI[14]：EXTI_DSEL[5:4] – 00：PD8 01：PD9 10：PD10 11：PD11 EXTI[13]：EXTI_DSEL[3:2] – 00：PD4 01：PD5 10：PD6 11：PD7

位号	助记符	功能描述
		EXTI[12]：EXTI_DSEL[1:0] – 00：PD0 01：PD1 10：PD2 11：PD3
23:22	-	RFU：未实现，读为 0
21:16	CSEL	PortC EXTI 中断输入选择 (External Interrupt PortC Select) EXTI[10]：EXTI_CSEL[5:4] – 00：PC8 01：PC9 10：PC10 11：PC11 EXTI[9]：EXTI_CSEL[3:2] – 00：PC4 01：PC5 10：PC6 11：PC7 EXTI[8]：EXTI_CSEL[1:0] – 00：PC0 01：PC1 10：PC2 11：PC3
15:8	BSEL	PortB EXTI 中断输入选择 (External Interrupt PortB Select) EXTI[7]：EXTI_BSEL[7:6] – 00：PB12 01：PB13 10：PB14 11：PB15 EXTI[6]：EXTI_BSEL[5:4] – 00：PB8 01：PB9 10：PB10 11：PB11 EXTI[5]：EXTI_BSEL[3:2] – 00：PB4 01：PB5 10：PB6 11：PB7 EXTI[4]：EXTI_BSEL[1:0] – 00：PB0 01：PB1 10：PB2 11：PB3
7:0	ASEL	PortA EXTI 中断输入选择 (External Interrupt PortA Select) EXTI[3]：EXTI_ASEL[7:6] – 00：PA12 01：PA13 10：PA14 11：PA15

位号	助记符	功能描述
		EXTI[2]：EXTI_ASEL[5:4] – 00：PA8 01：PA9 10：PA10 11：PA11 EXTI[1]：EXTI_ASEL[3:2] – 00：PA4 01：PA5 10：PA6 11：PA7 EXTI[0]： EXTI_ASEL[1:0] – 00：PA0 01：PA1 10：PA2 11：PA3

34.8.12 EXTI 边沿选择和使能寄存器（GPIO_EXTIEDS）

名称	GPIO_EXTIEDS							
Offset	0x00000104							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI15_EDS		EXTI14_EDS		EXTI13_EDS		EXTI12_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI11_EDS		EXTI10_EDS		EXTI9_EDS		EXTI8_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI7_EDS		EXTI6_EDS		EXTI5_EDS		EXTI4_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI3_EDS		EXTI2_EDS		EXTI1_EDS		EXTI0_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	

位号	助记符	功能描述
31:30	EXTI15_EDS	EXTI[15]边缘触发选择 (External Interrupt 15 Edge Select) 00：rising 01：falling 10：both 11：disable
29:28	EXTI14_EDS	EXTI[14]边缘触发选择 (External Interrupt 14 Edge Select) 00：rising 01：falling 10：both 11：disable
27:26	EXTI13_EDS	EXTI[13]边缘触发选择 (External Interrupt 13 Edge Select) 00：rising

位号	助记符	功能描述
		01：falling 10：both 11：disable
25:24	EXTI12_EDS	EXTI[12]边缘触发选择 (External Interrupt 12 Edge Select) 00：rising 01：falling 10：both 11：disable
23:22	EXTI11_EDS	EXTI[11]边缘触发选择 (External Interrupt 11 Edge Select) 00：rising 01：falling 10：both 11：disable
21:20	EXTI10_EDS	EXTI[10]边缘触发选择 (External Interrupt 10 Edge Select) 00：rising 01：falling 10：both 11：disable
19:18	EXTI9_EDS	EXTI[9]边缘触发选择 (External Interrupt 9 Edge Select) 00：rising 01：falling 10：both 11：disable
17:16	EXTI8_EDS	EXTI[8]边缘触发选择 (External Interrupt 8 Edge Select) 00：rising 01：falling 10：both 11：disable
15:14	EXTI7_EDS	EXTI[7]边缘触发选择 (External Interrupt 7 Edge Select) 00：rising 01：falling 10：both 11：disable
13:12	EXTI6_EDS	EXTI[6]边缘触发选择 (External Interrupt 6 Edge Select) 00：rising 01：falling 10：both 11：disable
11:10	EXTI5_EDS	EXTI[5]边缘触发选择 (External Interrupt 5 Edge Select) 00：rising 01：falling 10：both 11：disable
9:8	EXTI4_EDS	EXTI[4]边缘触发选择 (External Interrupt 4 Edge Select) 00：rising 01：falling 10：both 11：disable
7:6	EXTI3_EDS	EXTI[3]边缘触发选择 (External Interrupt 3 Edge Select)

位号	助记符	功能描述
		00：rising 01：falling 10：both 11：disable
5:4	EXTI2_EDS	EXTI[2]边缘触发选择 (External Interrupt 2 Edge Select) 00：rising 01：falling 10：both 11：disable
3:2	EXTI1_EDS	EXTI[1]边缘触发选择 (External Interrupt 1 Edge Select) 00：rising 01：falling 10：both 11：disable
1:0	EXTI0_EDS	EXTI1[0] 边缘触发选择 (External Interrupt 0 Edge Select) 00：rising 01：falling 10：both 11：disable

34.8.13 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）

名称	GPIO_EXTIDF							
Offset	0x00000108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DF[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DF	EXTI[0~15]输入数字滤波功能使能 (External Interrupt Digital Filter Enable) 0：关闭 EXTI 数字滤波 1：使能 EXTI 数字滤波

34.8.14 EXTI 中断标志寄存器（GPIO_EXTIISR）

名称	GPIO_EXTIISR
Offset	0x0000010C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IF[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	IF	EXTI[0~15]外部引脚中断标志寄存器，共可以产生 16 个引脚中断 (External Interrupt Flags) 硬件置位，软件写 1 清零

34.8.15 EXTI 输入信号寄存器（GPIO_EXTIDI）

名称	GPIO_EXTIDI							
Offset	0x00000110							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DI[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DI[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	DI	EXTI[0~15]输入信号只读寄存器，软件读取此寄存器可以观察 EXTI 的 16 个输入信号的当前状态 (External Interrupt Data Input) *注：当使能了数字滤波后，软件可以从这个寄存器读取到某个 IO 输入信号滤波后的状态。*

34.8.16 FOUT 配置寄存器（GPIO_FOUTSEL）

名称	GPIO_FOUTSEL

offset	0x00000200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FOUT1SEL				FOUT0SEL			
位权限	R/W-0000				R/W-0000			

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:4	FOUT1SEL	PB12 输出选择(Frequency Output Select for PB12) 0000：XTLF 0001：LPOSC 0010：RCHF/64 0011：LSCLK 0100：AHBCLK/64 0101：RTCTM 0110：PLLO/64 0111：RTCCLK64Hz 1000：APBCLK/64 1001：PLLO 1010：RCMFPSC 1011：RCHF 1100：XTHF/64 1101：ADCCLK/64 1110：CLK8K 1111：COMP2O
3:0	FOUT0SEL	PD11 输出选择(Frequency Output Select for PD11) 0000：XTLF 0001：LPOSC 0010：RCHF/64 0011：LSCLK 0100：AHBCLK/64 0101：RTCTM 0110：PLLO/64 0111：RTCCLK64Hz 1000：APBCLK/64 1001：PLLO 1010：RCMFPSC 1011：RCHF 1100：XTHF/64 1101：COMP1O 1110：CLK8K

位号	助记符	功能描述
		1111：ADC_CLK

34.8.17 WKUP 使能寄存器（GPIO_PINWKEN）

名称	GPIO_PINWKEN							
Offset	0x00000300							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKISEL	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SEL[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	WKISEL	WKUPx 中断入口选择 (WKUP interrupt entry select) 0：NMI 中断 1：#38 入口
30:16	-	RFU：未实现，读为 0
15:8	SEL	WKUP 边沿选择 (Wakeup edge Select) 1：对应的 WKUP 引脚为上升沿唤醒 0：对应的 WKUP 引脚为下降沿唤醒
7:0	EN	WKUP 引脚使能信号 (Wakeup Enable) 1：对应的 WKUP 引脚功能有效 0：对应的 WKUP 引脚功能无效 PINWKEN[x]控制 WKUPx 引脚的使能

35 专用编程接口

35.1 概述

FM33LC0XX芯片可使用复旦微电子所提供的专用编程器，或者通过Bootloader下载用户程序。编程器通过专用编程接口(SWD)与芯片通信，完成程序下载，并可对Flash全空间内容进行Checksum校验。

35.2 编程器使用

编程器的使用方法请参考应用手册，或联系复旦微电子公司。

36 调试支持

36.1 概述

FM33LC0XX芯片基于ARM Cortex-M0处理器构建，并支持相应的debug特性。通过硬件断点（breakpoint）和数据观察点（watchpoint），调试器可以在特定指令取指和数据访问时停止CPU内核运行，检视内核寄存器和系统外设状态，并根据需要恢复内核运行。

仿真调试主机通过SWD接口与FM33LC0xx芯片互联，并实现仿真调试。

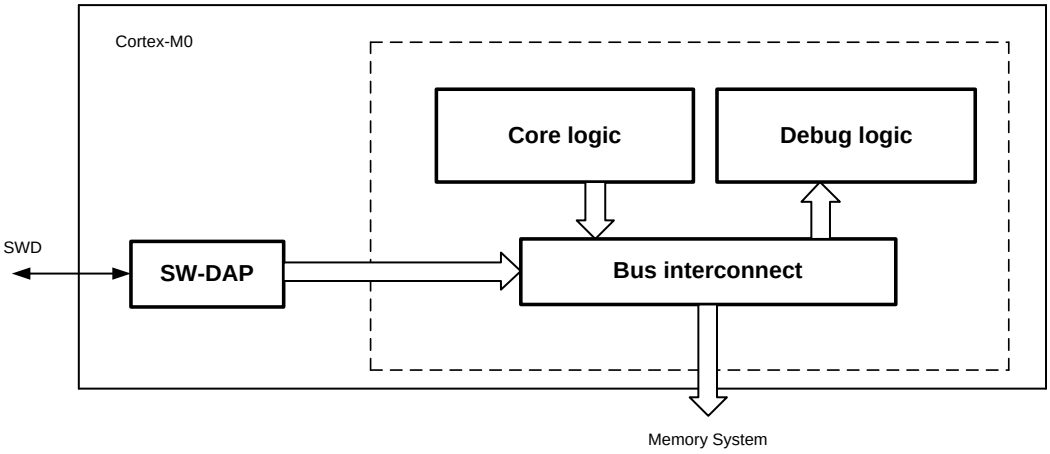


图 36-1 Cortex-M0 调试系统示意图

关于Cortex-M0内核的debug特性，请参考ARM公司的Cortex-M0技术参考手册。

36.2 Debug 引脚

36.2.1 SWD 引脚

FM33LC0xx系列MCU的SWD引脚位置如下表：

SWD pins	Debug功能	引脚定义
SWDIO	SWD数据输入/输出	PD8
SWCLK	SWD时钟输入	PD7

注意：芯片复位后PD7和PD8都默认为输入状态，与大部分GPIO不同。

36.2.2 上拉电阻

芯片复位后，SWDIO引脚默认使能内部上拉（约100K欧姆），SWCLK引脚默认不使能内部上拉电阻，因此用户需要在PCB上外接上拉电阻，以防止输入引脚浮空导致漏电增加。

36.3 SWD 接口协议

36.3.1 协议简介

SWD协议采用LSB-first进行数据收发。通过SWD接口，调试主机可以读写DPACC和APACC寄存器组。

SWIO每次切换数据方向时，总线上需插入turn-around时间，这段时间内主机和从机都不会驱动SWIO。在两次传输之间，主机必须将线驱动为低电平进入idle状态，或继续发送一次新传输的起始位继续传输，在一次数据包传输之后，主机也可以空闲，使线保持为高电平或由上拉电阻上拉。SWD协议没有明确的复位信号，在没有看到预期的信号时，主机或目标机将对复位进行检测。通过保持线为高电平持续50个时钟周期之后跟随一个读ID的请求，可以确保在检测到错误或复位之后重新同步成功。

36.3.2 传输序列

SWD每个通信传输序列包含三个部分：

1、包请求（8bits），由主机发送

2、ACK响应（3bits），由从机回发

3、数据传输（33bits），由主机或从机发送

其中包请求字节定义如下：

Bit	Name	描述
0	Start	起始位，必须是1
1	ApnDP	AP/DP选择 0：DP访问 1：AP访问
2	RnW	读写选择 0：写请求 1：读请求
4:3	A[3:2]	DP/AP寄存器的地址域
5	Parity	Bit0~Bit4数据的校验位
6	Stop	0
7	Park	主机不驱动，通过总线上拉，从机读为1

包请求发送后，总线上总是有1bit的turn-around时间。

ACK响应定义如下：

Bit	Name	描述
0:2	ACK	001：FAULT

		010：WAIT 100：OK

如果主机发起读操作，或者ACK为WAIT或FAULT，则ACK之后必须插入turn-around时间。

数据传输格式如下：

Bit	Name	描述
0:31	Data	读出或写入的数据
32	Parity	32bit 数据的校验位

36.3.3 SW-DP ID code

Cortex-M0的SW-DP有一个固定的ID code：0x0BB11477

SW-DP处于非活跃状态，直到主机读取ID code。

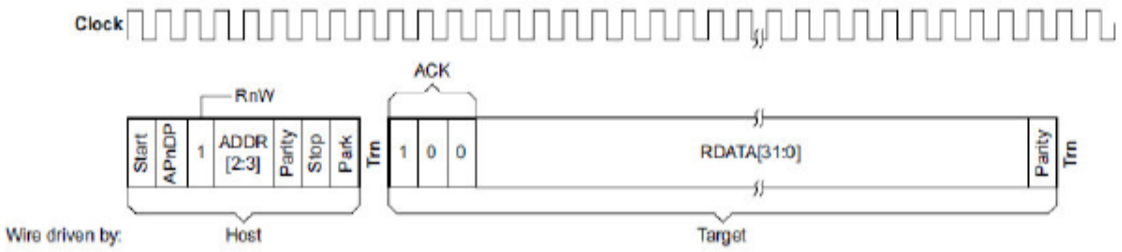
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

36.3.4 主机读操作

一次成功的读操作由以下三个阶段组成

- 一个8位的读数据包请求（request），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，WAIT响应为010，FAULT响应为001.
- 一个33位的数据读阶段（payload），从主机到目标。

默认情况下，在第一和第二阶段之间以及第三阶段之后有一个时钟的掉转周期，一次成功的读操作如下图。

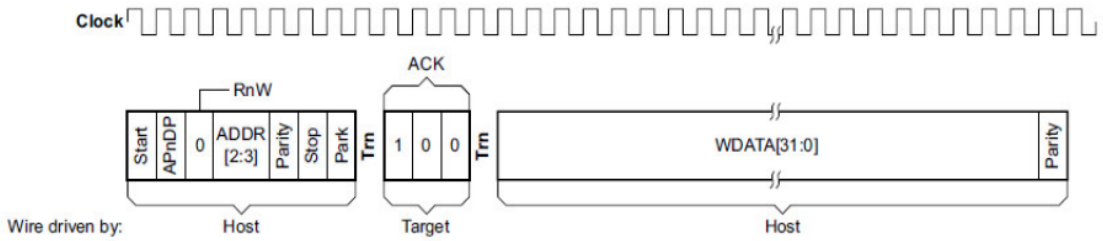


36.3.5 主机写操作

一次写操作由以下三个阶段组成

- 一个8位的写数据包请求（header），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，FAULT响应为001.
- 一个33位的数据写阶段（payload），从主机到目标。

默认情况下，每两个阶段之间都有一个时钟的掉转周期，一次成功的写操作如下图。



36.4 SWD-DP 寄存器

36.4.1 寄存器列表

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
		ABORT	WO
01	0x0	CTRL/STAT	RW
	0x1	DLCR	RW
10	x	RESEND	RO
		SELECT	WO
11	x	RDBUFF	RO

关于寄存器的详细说明，请参考Cortex-M0 Technical Reference Manual.

36.5 Core debug 寄存器

通过操作core debug寄存器可以实现内核调试。主机通过SW-DP访问以下内核调试寄存器。

Address	Name	Type	Function
0xE000EDF0	DHCSR	RW	Debug Halting Control and Status Register
0xE000EDF4	DCRSR	WO	Debug Core Register Selector Register
0xE000EDF8	DCRDR	RW	Debug Core Register Data Register
0xE000EDFC	DEMCR	RW	Debug Exception and Monitor Control Register
0xE000EE00 to 0xE000EEFF	-	-	Reserved for Debug Extension

上述debug寄存器不被系统复位影响，仅受上电复位影响。通过以下方式可以实现CPU复位后立即halt：

- 置位DEMCR寄存器的bit0 (VC_CORRESET)
- 置位DHCSR寄存器的bit0 (C_DEBUGEN)
- 执行系统复位

36.6 Debug 相关的配置项

通过配置DBG_CR寄存器，可以设置在调试状态下，芯片内部的定时器、看门狗电路是否继续工作。详情请参见6.5.1DEBUG配置寄存器（DBG_CR）。

37 器件签名信息

每一颗FM33LC0系列MCU都有自己的器件签名，包括存储器容量信息和唯一器件ID号。

37.1 寄存器

起始地址	字节大小（单位：Byte）	模块名称
0x40000000	256	SCU

地址	名称	符号
0x40000000	存储器容量查询寄存器 （Memory Capacity Query Register）	SCU_CQR

37.1.1 存储器容量查询寄存器（SCU_CQR）

通过查询SYSCON寄存器的bit2，可以获得器件Flash容量信息。

名称	SCU_CQR							
地址	0x40000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	VERIFAIL	RDPROTFAIL	-					
位权限	R-0	R-0	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					FLSCFG	-	
位权限	U-0					R-0	U-0	

位号	助记符	功能描述
31	VERIFAIL	Flash 配置信息，即 LDT0 区，校验错误标志(Verification code checksum Fail) 1：某项配置信息校验出现错误，错误项将保持默认值 0：校验正确
30	RDPROTFAIL	OPTBYTE/ACLOCK，即 LDT1 区，校验错误标志(Read-out Protection checksum Fail) 1：校验失败，Debug 接口及 ACLOCK 保护相关寄存器置 1 0：校验通过
29:3	-	RFU：未实现，读为 0
2	FLSCFG	Flash 大小配置(Flash size configuration)

位号	助记符	功能描述
		0：256KB 1：128KB （RAM 都是 24KB）
1:0	-	RFU：未实现，读为 0

37.2 器件 UID

FM33LC0系列每颗MCU的器件UID都是全球唯一的，由原厂写入，出厂后不可改写。

UID共128 bits，保存在Flash特殊扇区，软件运行时可以读取此UID，用于实现代码保护或安全启动类应用。

UID访问地址是0x1FFF_FE80~0x1FFF_FE8F

版本列表

版本号	发布日期	页数	章节或图表	更改说明
1.0	2019.08	676		首次发布
1.1	2019.09	687		增加 36 调试支持章节
1.2	2019.11	695		完善电参数章节内容 更新极限参数和引脚类型
1.3	2019.12	696		更新温度传感器绝对温度计算方法
1.4	2020.01	696		完善 CDM 和 LU 数据
1.5	2020.02	699	2.1.3	增加 QFN32 封装
1.6	2020.02	700	1	修改综述中的部分典型参数 更新引脚说明
1.7	2020.02	701	3.4.11	更新 OPA 参数
1.8	2020.04	700	3.4.4	修改 VREF buffer 输出建立时间
1.9	2020.04	701	2.1.3	增加 FM33LC0x5N 的 LQFP48 封装
2.0	2020.05	700	1.2	删除芯片结构图中的 Beeper 模块
2.1	2020.05	700	2.1.5	增加 TSSOP20 封装
2.2	2020.05	702	2.1.5	增加 FM33LC0x3U QFN32 封装

上海复旦微电子集团股份有限公司销售及服务网点

上海复旦微电子集团股份有限公司
地址：上海市国泰路 127 号 4 号楼
邮编：200433
电话：(86-021) 6565 5050
传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司
地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室
电话：(852) 2116 3288　2116 3338
传真：(852) 2116 0882

北京办事处
地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室
邮编：100007
电话：（86-10）8418 6608
传真：（86-10）8418 6211

深圳办事处
地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室
邮编：518028
电话：(86-0755) 8335 0911　8335 1011　83352011 83350611
传真：(86-0755) 8335 9011

台湾办事处
地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室
电话：(886-2) 7721 1889
传真：(886-2) 7722 3888

新加坡办事处
地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929
电话：(65) 6472 3688
传真：(65) 6472 3669

北美办事处
地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA
电话：(480) 857-6500 ext 18

公司网址：http://www.fmsh.com/

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for fudan ***manufacturer:***

Other Similar products are found below :

FM25Q04A-SO-U-G FM33LC046N FM1702Q FM1701 FM17520-QNA-T-G FM33LC023N FM25Q16A-SO-U-G FM25Q04-DN-T-G FM33L025 FM15160 508-03 FM1702SL FM33G026 FM25Q08A-DN-T-G FM24C04D-SO-T-G FM33L026 FM17522 FM25W32-SO-T-G FM17550 FM33G023 FM33L013 FM25F005-TS-T-G FM25F005-SO-T-G FM24C04D-TS-T-G FM24C256A-TS-T FM24C08D-SO-T-G FM24C32D-SO-T-G FM25Q16A-S0-T-G FM93C46A-SO-T-G FM1735Q FM24C02B-DN-T-G FM33G048 FM11NC08S FM24C256A-SO-T-G FM33G045 FM25F01-DN-T-G FM24C32A-SO-T-G FM25Q16A-DN-T-G FM34C04D-DN-T-G FM24C32A-TS-T-G FM93C56A-SO-T-G FM24C64D-TS-T-G FM24C32D-TS-T-G FM24C256E-TS-T-G FM24C128D-DN-T FM24C64D-SO-T-G FM24C02B-TS-T-G FM24C128D-SO-T-G-AX FM24C02C-SO-T-G