

GigaDevice Semiconductor Inc.

GD32F103xx **ARM[®] 32-bit Cortex[™]-M3 CPU**

用户手册

本文档由北京锐鑫同创科技有限公司依据 GD32F103xx User Manual V1.0 翻译整理完成，已经进行了全面的校对，由于我们的水平有限以及文档篇幅的庞大，翻译的过程中难免会有错误和遗漏的地方，希望广大读者们能够及时向我们反馈您在阅读期间所发现的错误和问题，我们会尽快在下一个版本中更正。您可以发邮件到 gd32@realsenes.com.cn 向我们提出您的意见和建议，谢谢。

版本：1.0

发布日期：2013年9月1日

目 录

第 1 章	系统及存储器架构	1
1	ARM Cortex-M3 处理器	1
2	系统架构	2
3	存储器映射	4
4	引导配置	7
5	CRC 计算单元	7
第 2 章	电源控制 (PWR)	11
1	简介	11
2	主要特点	11
3	功能描述	11
4	PWR 寄存器	14
第 3 章	复位及时钟控制单元 (RCCU)	20
1	复位控制单元 (RCU)	20
2	时钟控制单元 (CCU)	21
3	RCCU 寄存器	25
第 4 章	通用 I/O (GPIO) 及备选功能 I/O (AFIO)	42
1	GPIO	42
2	GPIO 寄存器	50
3	AFIO	57
4	AFIO 寄存器	62
第 5 章	中断及事件	68
1	简介	68
2	主要特点	68
3	功能描述	68
4	中断及 EXTI 寄存器	71
第 6 章	DMA 控制器	75
1	简介	75
2	主要特点	75
3	功能描述	75
4	DMA 寄存器	80
第 7 章	定时器 (TMx)	85
1	高级定时器 (TM1)	85
2	TM1 寄存器	114
3	通用定时器 (TM2~TM4)	132
4	TM2~TM4 寄存器	155
第 8 章	实时时钟 RTC	172
1	简介	172
2	主要特点	172
3	功能描述	172

4	RTC 寄存器	175
第 9 章	看门狗 (WDG)	180
1	独立式看门狗 (IWDG)	180
2	IWDG 寄存器	182
3	窗口式看门狗 (WWDG)	184
4	WWDG 寄存器	186
第 10 章	模数转换器 (ADC)	189
1	简介	189
2	主要特点	189
3	功能描述	189
4	ADC 寄存器	202
第 11 章	I ² C 接口	212
1	简介	212
2	主要特点	212
3	功能描述	212
4	I ² C 寄存器	226
第 12 章	SPI 接口	234
1	简介	234
2	主要特点	234
3	功能描述	234
4	SPI 寄存器	244
第 13 章	通用同步/异步收发器 (USART)	250
1	简介	250
2	主要特点	250
3	功能描述	251
4	USART 寄存器	261
第 14 章	通用串行总线全速接口 (USB2.0 FS)	269
1	简介	269
2	主要特点	269
3	功能描述	269
4	USB 寄存器	274
第 15 章	控制器局域网 (bxCAN)	282
1	简介	282
2	主要特点	282
3	功能描述	283
4	bxCAN 寄存器	294
第 16 章	外部存储器接口 (EXMC)	312
1	简介	312
2	主要特点	312
3	功能描述	312
4	EXMC 寄存器	328

第 17 章 文档修订记录 333

图

图 1	Cortex™-M3 内部框图.....	2
图 2	GD32F103xx 系列微控制器系统架构.....	3
图 3	GD32F103xx 系列微控制器存储器映射表.....	5
图 4	CRC 计算单元框图.....	8
图 5	电源概览.....	12
图 6	系统复位电路框图.....	20
图 7	时钟树.....	22
图 8	HSE 时钟源.....	23
图 9	标准 I/O 管脚的基本结构.....	43
图 10	标准 I/O 管脚与可耐受 5V 的 I/O 管脚.....	44
图 11	浮空输入/上拉输入/下拉输入的配置.....	45
图 12	输出配置.....	46
图 13	高阻模拟输入配置.....	47
图 14	备选功能配置.....	48
图 15	EXTI 框图.....	71
图 16	DMA 请求映射.....	78
图 17	高级定时器内部框图.....	86
图 18	分频系数从 1 改为 2 的计数器时序图.....	87
图 19	分频系数从 1 改为 4 的计数器时序图.....	87
图 20	分频系数为 1 的计数器时序图.....	88
图 21	分频系数为 2 的计数器时序图.....	88
图 22	分频系数为 4 的计数器时序图.....	89
图 23	分频系数为 N 的计数器时序图.....	89
图 24	当 ARPE=0 时计数器更新事件的计数器时序图.....	90
图 25	当 ARPE=1 时计数器更新事件的计数器时序图.....	90
图 26	分频系数为 1 的计数器时序图.....	91
图 27	分频系数为 2 的计数器时序图.....	92
图 28	分频系数为 4 的计数器时序图.....	92
图 29	分频系数为 N 的计数器时序图.....	93
图 30	当计数器未使用时，计数器更新事件的计数器时序图.....	93
图 31	分频系数为 1、TMx_ARR = 0x5 的计数器时序图.....	94
图 32	分频系数为 2、TMx_ARR = 0x5 的计数器时序图.....	95
图 33	分频系数为 4、TMx_ARR = 0x63 的计数器时序图.....	95
图 34	分频系数为 N 时的计数器时序图.....	96
图 35	APRE = 1 时计数器更新事件的计数器时序图.....	96
图 36	APRE = 1 时计数器更新事件的计数器时序图.....	97
图 37	更新速率示例，取决于工作模式以及 TMx_RCR.....	98
图 38	标准模式下的控制电路，内部时钟分频系数为 1.....	98
图 39	捕获/比较通道（示例：通道 1 输入级）.....	100

图 40	捕获/比较通道 1 主电路.....	100
图 41	捕获/比较通道 1~3 输出级.....	101
图 42	捕获/比较通道 4 输出级.....	101
图 43	输出比较模式, OC1 的翻转.....	102
图 44	带有死区时间插入的互补输出.....	103
图 45	当延迟大于负脉冲时的死区时间波形.....	103
图 46	当延迟大于正脉冲时的死区时间波形.....	103
图 47	发生紧急停止事件时的输出表现.....	104
图 48	单脉冲模式.....	105
图 49	编码器接口模式下的计数器运行情况.....	106
图 50	TF1FP1 极性反转后, 编码器接口模式下的计数器运行情况.....	107
图 51	重启模式下的控制电路.....	108
图 52	暂停模式下的控制电路.....	108
图 53	触发模式下的控制电路.....	109
图 54	定时器 1 主机/从机模式定时器示例.....	109
图 55	用定时器 2 的使能输出触发定时器 1.....	110
图 56	用定时器 2 的定时器更新事件触发定时器 1.....	111
图 57	用定时器 2 的使能信号来驱动定时器 1.....	112
图 58	用定时器 2 的 OC1REF 信号来驱动定时器 1.....	112
图 59	用定时器 1 的 TI1 信号来触发定时器 1 和定时器 2.....	113
图 60	通用定时器内部框图.....	133
图 61	分频系数从 1 改为 2 的计数器时序图.....	135
图 62	分频系数从 1 改为 4 的计数器时序图.....	135
图 63	分频系数为 1 的计数器时序图.....	136
图 64	分频系数为 2 的计数器时序图.....	137
图 65	分频系数为 4 的计数器时序图.....	137
图 66	分频系数为 N 的计数器时序图.....	138
图 67	当 ARPE=0 时计数器更新事件的计数器时序图.....	138
图 68	当 ARPE=1 时计数器更新事件的计数器时序图.....	139
图 69	分频系数为 1 的计数器时序图.....	140
图 70	分频系数为 2 的计数器时序图.....	140
图 71	分频系数为 4 的计数器时序图.....	141
图 72	分频系数为 N 的计数器时序图.....	141
图 73	当计数器未使用时, 计数器更新事件的计数器时序图.....	142
图 74	分频系数为 1、TMx_ARR = 0x5 的计数器时序图.....	143
图 75	分频系数为 2 的计数器时序图.....	143
图 76	分频系数为 4、TMx_ARR = 0x63 的计数器时序图.....	144
图 77	分频系数为 N 时的计数器时序图.....	144
图 78	APRE = 1 时计数器更新事件的计数器时序图.....	145
图 79	APRE = 1 时计数器更新事件 (计数器上溢) 的计数器时序图.....	145
图 80	标准模式下的控制电路, 内部时钟分频系数为 1.....	146

图 81	捕获/比较通道（示例：通道 1 输入级）	147
图 82	捕获/比较通道 1 主电路	148
图 83	捕获/比较通道 4 输出级	148
图 84	输出比较模式，OC1 的翻转	149
图 85	单脉冲模式	150
图 86	编码器接口模式下的计数器运行情况	151
图 87	TF1FP1 极性反转后，编码器接口模式下的计数器运行情况	152
图 88	重启模式下的控制电路	153
图 89	暂停模式下的控制电路	153
图 90	触发模式下的控制电路	154
图 91	定时器 1 主机/从机模式定时器示例	154
图 92	RTC 内部框图	173
图 93	RTC 秒信号及闹钟信号的波形（PR = 3, ALARM = 2）	174
图 93	RTC 秒信号及溢出信号的波形（PR = 3）	175
图 95	独立式看门狗内部框图	181
图 96	窗口看门狗内部框图	185
图 97	窗口式看门狗时序图	186
图 98	ADC 模块内部框图	190
图 99	单次转换模式	191
图 100	连续转换模式，禁用扫描	192
图 101	连续转换模式，使能扫描	192
图 102	扫描转换模式	193
图 103	间断转换模式	194
图 104	数据对齐	195
图 105	双 ADC 模式下的功能框图	197
图 106	常规组并发模式	198
图 107	插入组并发模式	198
图 108	交替触发模式，禁用插入间断模式	199
图 109	交替触发模式，使能插入间断模式	199
图 110	高速延迟模式，使能连续转换模式	200
图 111	低速延迟模式	200
图 112	常规组并发+交替触发模式	201
图 113	插入组并发+延迟模式	201
图 114	I ² C 模块内部框图	213
图 115	I ² C 数据有效性	214
图 116	起始位与停止位	214
图 117	时钟同步	215
图 118	SDA 线仲裁	215
图 119	采用 7 位地址的 I ² C 通信流	216
图 120	采用 10 位地址的 I ² C 通信流	216
图 121	从机发送的程序流模型	218

图 122	从机接收的程序流模型	219
图 123	主机发送的程序流模型	221
图 124	主机接收的程序流模型	223
图 125	单主/单从应用框图.....	235
图 126	SPI 数据时钟时序图.....	236
图 127	通过 DMA 发送	242
图 128	通过 DMA 接收	244
图 129	USART 模块内部框图.....	252
图 130	USART 字符帧 (9 数据位+1 停止位)	253
图 131	LIN 模式下的帧格式错误检测以及暂停帧检测.....	255
图 132	同步模式下的 USART 接线示例.....	256
图 133	8 位格式的 USART 同步通信波形 (LBCL=1)	256
图 134	IrDA SIR ENDEC 模块	258
图 135	IrDA 数据调制	258
图 136	两个 USART 设备之间的硬件流控.....	258
图 137	硬件流控	259
图 138	USART 中断映射框图.....	261
图 139	缓冲描述符表的用法示例 (USB_BTABLE=0)	270
图 140	发送寄存器	285
图 141	发送邮箱的基本工作状态	285
图 142	接收寄存器	287
图 143	32 位过滤.....	288
图 144	16 位过滤.....	288
图 145	32 位掩码过滤.....	288
图 146	32 位掩码过滤.....	289
图 147	位时间的组成	291
图 148	EXMC 内部框图	313
图 149	外部存储器设备地址映射	313
图 150	示例 1 模式 1 写操作时序.....	316
图 151	示例 1 模式 1 读操作时序.....	316
图 152	示例 2 模式 A 写操作时序.....	317
图 153	示例 2 模式 A 读操作时序.....	318
图 154	示例 3 模式 B 写操作时序.....	319
图 155	示例 3 模式 2 写操作时序.....	319
图 156	示例 3 模式 2/B 读操作时序	321
图 157	示例 4 模式 C 写操作时序.....	322
图 158	示例 4 模式 C 读操作时序.....	322
图 159	示例 5 模式 D 写操作时序.....	323
图 160	示例 5 模式 D 读操作时序.....	324
图 161	示例 6 复用模式写操作时序	326
图 162	示例 6 复用模式读操作时序	326

图 163	示例 7 异步等待信号作用时的写操作时序	327
图 164	示例 7 异步等待信号作用时的读操作时序	327

表

表 1	Flash 模块的组织	6
表 2	引导模式	7
表 3	节电模式小结	14
表 4	CKOUT 的时钟源选择	25
表 5	GPIO 端口配置位	43
表 6	高级定时器 TM1	48
表 7	通用定时器 TM2/3/4	48
表 8	USART	49
表 9	SPI	49
表 10	I ² C	49
表 11	bxCAN	50
表 12	USB	50
表 13	ADC	50
表 14	其它 I/O	50
表 15	调试接口信号	57
表 16	调试接口映射	58
表 17	ADC1 外部触发插入式转换备选功能重映射	58
表 18	ADC1 外部触发标准转换备选功能重映射	58
表 19	ADC2 外部触发插入式转换备选功能重映射	58
表 20	ADC2 外部触发标准转换备选功能重映射	58
表 21	TM1 备选功能重映射	59
表 22	TM2 备选功能重映射	59
表 23	TM3 备选功能重映射	59
表 24	TM4 备选功能重映射	60
表 25	USART1 备选功能重映射	60
表 26	USART2 备选功能重映射	60
表 27	USART3 备选功能重映射	60
表 28	I ² C1 备选功能重映射	61
表 29	SPI1 备选功能重映射	61
表 30	CAN 备选功能重映射	61
表 31	OSC32 管脚配置	62
表 32	OSC 管脚配置	62
表 33	Cortex™-M3 中的 NVIC 异常类型	69
表 34	GD32F103xx 的中断向量表	69
表 35	DMA 在源与地址间的访问动作	77
表 36	各 DMA 通道的可用请求	79
表 37	编码器信号以及计数方向	106
表 38	编码器信号以及计数方向	151
表 39	40kHz (LSI) 频率下的 IWDG 超时范围	181

表 40	36MHz (f_{PCLK1}) 频率下的 WWDG 超时范围	186
表 41	ADC 管脚定义	190
表 42	常规通道的外部触发选择	195
表 43	插入通道的外部触发选择	196
表 44	I ² C 总线相关术语定义	213
表 45	事件状态标志	225
表 46	I ² C 错误标志	225
表 47	SPI 中断请求	244
表 48	USART 重要管脚描述	251
表 49	停止位配置	253
表 50	USART 中断请求	260
表 51	双缓冲的用法	273
表 52	接收状态编码	279
表 53	节点类型编码	279
表 54	节点类别含义	279
表 55	发送状态编码	279
表 56	32 位列表模式过滤条目编号	289
表 57	匹配索引	290
表 58	NOR Flash 接口信号功能	314
表 59	PSRAM 接口信号功能	314
表 60	PSRAM 接口信号功能	315
表 61	示例 1 模式 1 相关寄存器配置	315
表 62	示例 1 模式 A 相关寄存器配置	317
表 63	示例 3 模式 2/B 相关寄存器配置	318
表 64	示例 4 模式 C 相关寄存器配置	321
表 65	示例 5 模式 D 相关寄存器配置	323
表 66	示例 6 复用模式相关寄存器配置	324
表 67	修订历史	333

第1章 系统及存储器架构

GD32F103xx 系列微控制器的系统架构将在本章进行详细介绍，包括 ARM[®] Cortex[™]-M3 处理器内核、总线架构以及存储器构成。Cortex[™]-M3 处理器是一种面向新时代的内核，实现了许多新颖的特性。高度集成化以及各种先进功能，使 Cortex[™]-M3 处理器非常适用于那些要求微控制器兼顾高性能、低功耗的市场和产品。简单说来，Cortex[™]-M3 处理器包含三组 AHB 总线，分别称为 ICode 总线、DCode 总线以及系统总线。Cortex[™]-M3 处理器的所有存储器访问都是在这三组总线上执行的，根据访问功能以及目的存储空间予以选择。存储器部分采用哈佛型总线架构，预定义存储器映射，且最高可提供 4GB 的寻址空间，充分保障了系统的灵活性和可扩展性。

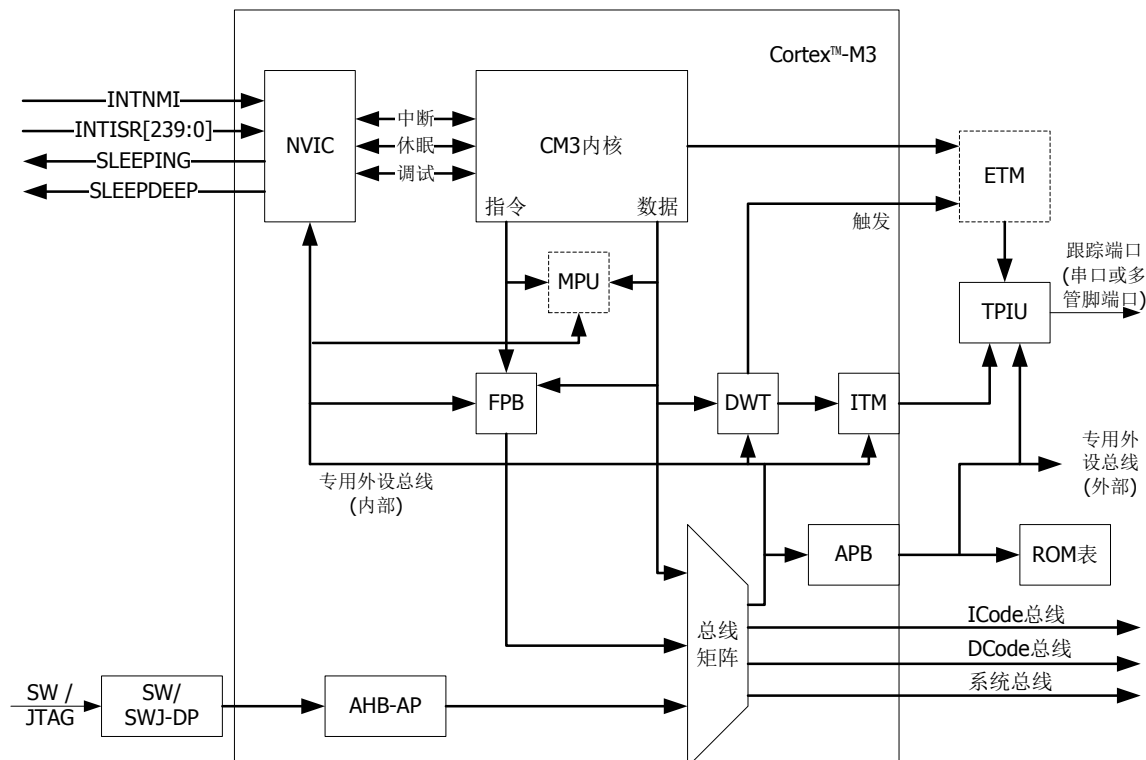
1 ARM Cortex-M3 处理器

Cortex[™]-M3 处理器是一种通用的 32 位处理器内核，尤其适用于兼顾高性能、低功耗的微控制器设计。该处理器具有许多新颖的特性，例如 Thumb2 指令集、硬件除法器、低延迟中断响应、原子级位绑定访问以及多总线同时工作。Cortex[™]-M3 处理器是基于 ARMv7 架构的，既支持 Thumb 指令集也支持 Thumb2 指令集。此外 Cortex[™]-M3 处理器提供了以下的系统级外设：

- 内部总线矩阵，用于实现 ICode 总线、DCode 总线、系统总线、专用外设总线（PPB）以及调试专用总线（AHB-AP）的互联
- 嵌套式向量型中断控制器（Nested Vectored Interrupt Controller，简称为 NVIC）
- 闪存地址重载及断点单元（Flash Patch and Breakpoint，简称为 FPB）
- 数据观测点及跟踪单元（Data Watchpoint and Trace，简称为 DWT）
- 跟踪仪器宏单元（Instrument Trace Macrocell，简称为 ITM）
- 串行线 JTAG 调试接口（Serial Wire JTAG Debug Port，简称为 SWJ-DP）
- 跟踪端口接口单元（Trace Port Interface Unit，简称为 TPIU）

下图为 Cortex[™]-M3 处理器的内部框图。如果您想了解关于 Cortex[™]-M3 更详尽的信息，请参阅《ARM[®] Cortex[™]-M3 技术参考手册》。

图 1 Cortex™-M3 内部框图



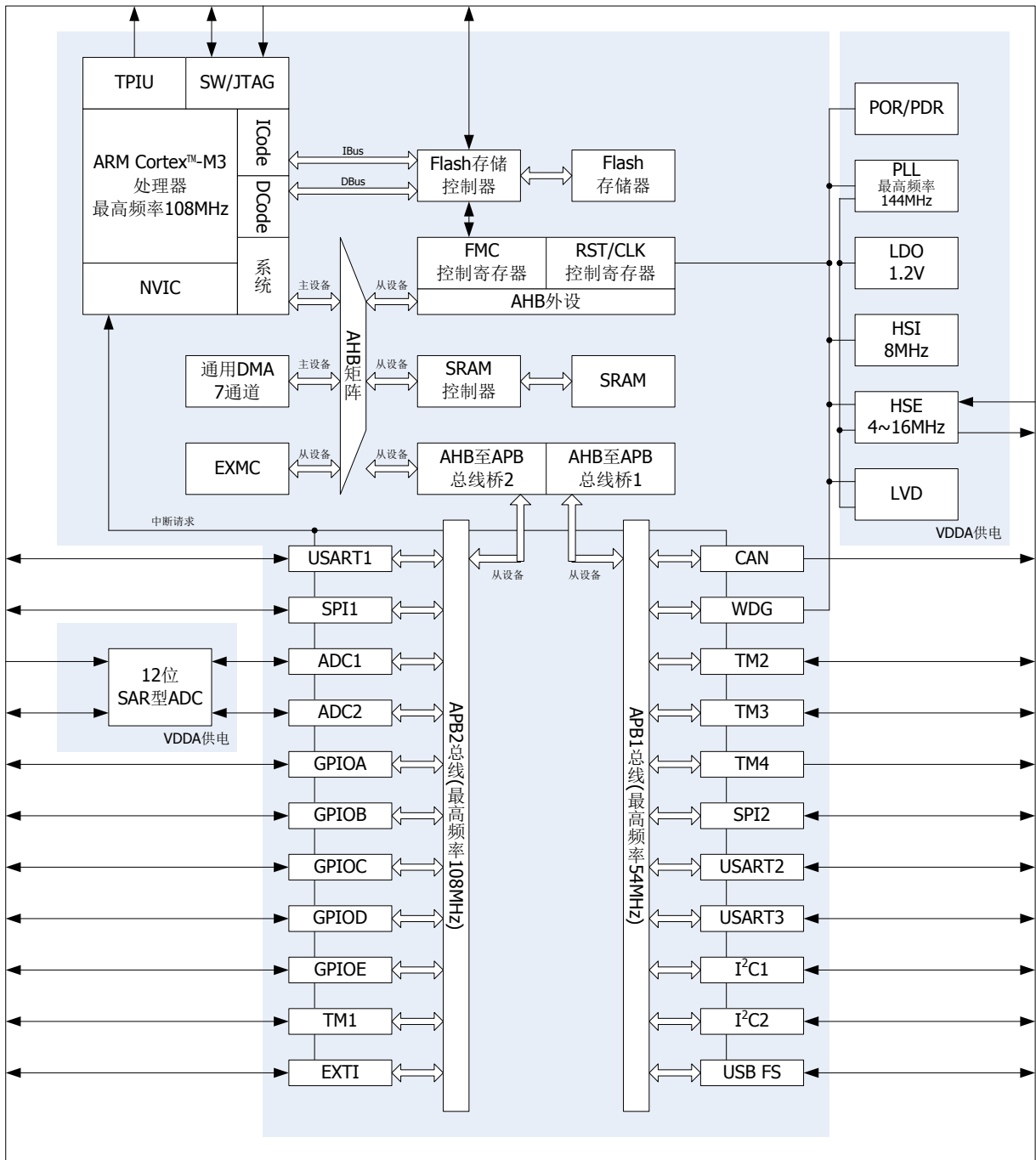
2 系统架构

GD32F103xx 系列微控制器的系统架构如下图所示。AHB 总线矩阵是基于 AMBA 3.0 AHB-LITE 技术的，这是一种多层级的 AHB，可确保系统中多个主设备和从设备之间可建立并行的访问路径。AHB 总线矩阵中有四个主设备：Cortex™-M3 内核的 ICode 总线、DCode 总线、系统总线，以及 DMA 控制器。ICode 总线即指令总线，同时也可以从代码存储区（0x0000 0000~0x1FFF FFFF）取出向量送至 Cortex™-M3 内核。DCode 总线即数据总线，主要用于加载/贮存数据，同时也可用于代码存储区的调试访问。与之类似，系统总线主要用于取值/取向量、加载/贮存数据以及调试访问系统存储区。系统存储区可划分为片内 SRAM 区、外部存储映射区以及外设映射区。AHB 总线矩阵中有五个从设备：Flash 存储控制器的 ICode、DCode 接口，片内 SRAM，外部存储控制器以及系统 AHB 总线。

系统 AHB 总线与所有的 AHB 外设相连，此外还包含两条 AHB-APB 总线桥，这样可以在系统 AHB 总线以及两个 APB 总线之间实现完全同步连接。两条 APB 总线则与所有的 APB 外设相连。APB1 总线的最高速度限制为 54MHz，APB2 总线则可全速运行（依器件型号而定，最高可达 108MHz）。

以上所述设备通过多层级 AHB 总线架构相互连接，如图 2 所示：

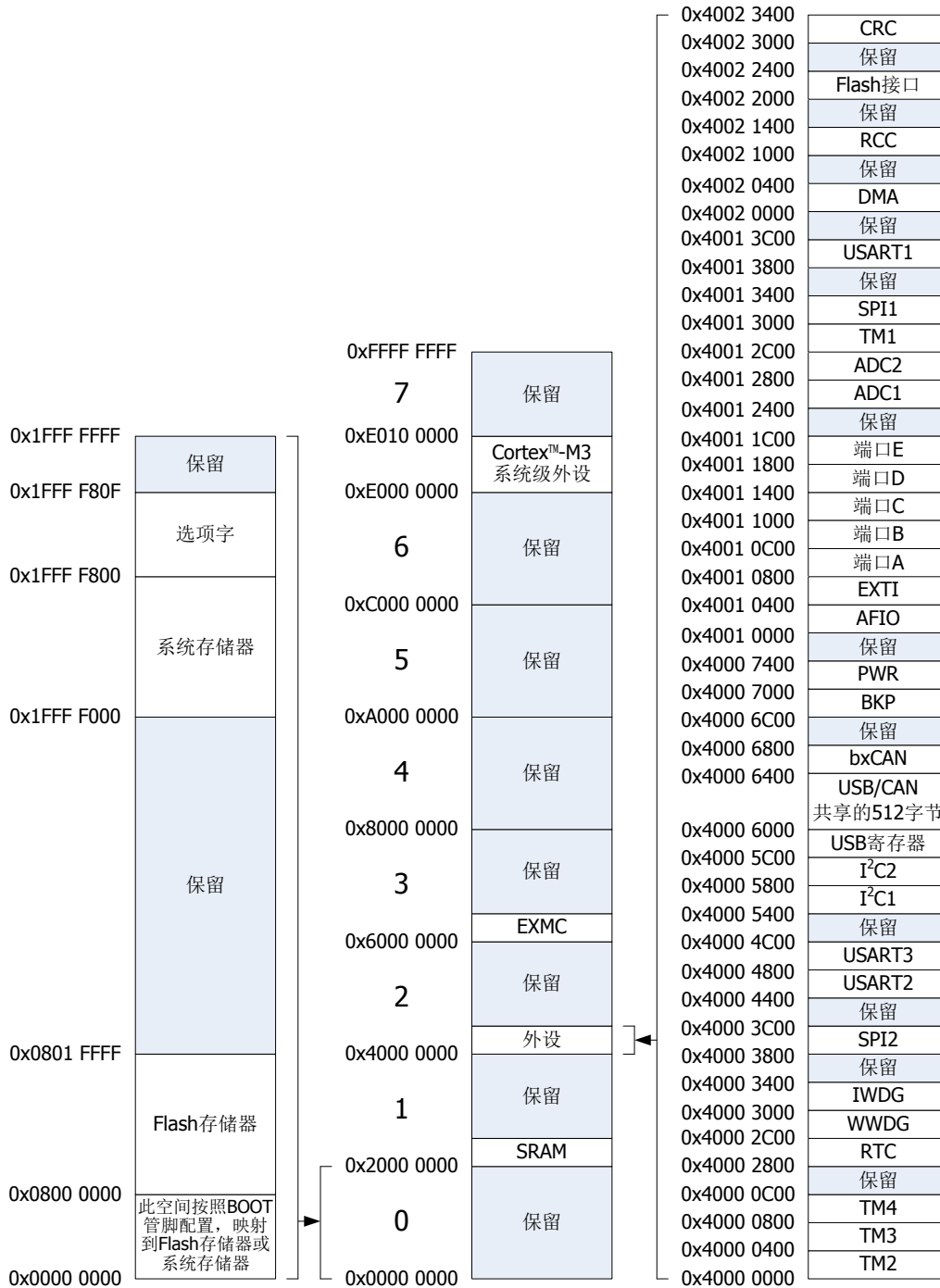
图 2 GD32F103xx 系列微控制器系统架构



3 存储器映射

ARM® Cortex™-M3 处理器采用哈佛型架构，取指和加载/贮存数据各自采用不同的总线。指令代码和数据从逻辑上位于同一寻址空间内，但分别划分在不同的地址段内。程序存储器、数据存储器、寄存器以及 I/O 端口均整合于同一个 4GB 线性寻址空间内（Cortex™-M3 内核总线宽度为 32 位，因此最大寻址空间为 4GB）。此外，Cortex™-M3 处理器提供预定义的存储器映射，这样对于用户开发而言，可以降低不同器件供应商芯片的软件差异程度，减少重复性的基础工作。不过，有些存储段是保留用于 Cortex™-M3 处理器系统级外设的。下图是 GD32F103xx 系列微控制器的存储器映射，包括代码、SRAM、外设以及其它预定义存储段。每种类型的每个外设统一分配 1kB 空间，这样能够极大简化针对单个外设的地址译码。APB1 外设位于 0x4000 0000~0x4000 FFFF 存储段内，APB2 外设位于 0x4001 0000~0x4001 7FFF 存储段内。AHB 外设位于 0x4001 8000~0x4002 FFFF 存储段内。

图 3 GD32F103xx 系列微控制器存储器映射表



3.1 位绑定

为减少“读-修改-写”操作所消耗的时间，Cortex™-M3 处理器提供了位绑定功能（bit-banding），可按位为单位进行原子级的访问。存储器映射表中包含了两个位绑定段，分别位于 SRAM 和外设

中。在位绑定段中，位别名区的某个字会映射到位绑定区的某个位上。

根据如下的映射计算公式，即可体现出位别名区的每个字如何对应于位绑定区的每个位。计算公式为：

$$\text{位别名地址} = \text{位绑定基地址} + (\text{字节偏移} \times 32) + (\text{位偏移} \times 4)$$

其中：

- 位别名地址：位别名区中，对应于指定位的单个字的映射地址
- 位绑定基地址：位别名区的起始地址
- 字节偏移：包含目标位的字节在位绑定区中的字节偏移
- 位偏移：目标位在该字节中的偏移

举例来说，要想对 0x2000 0200 的第 7 位进行原子级操作，其对应的位别名地址为：

$$\text{位别名地址} = 0x2200\ 0000 + (0x200 * 32) + (7 * 4) = 0x2200\ 401C$$

如果对 0x2200 401C 进行写操作，那么 0x2000 0200 的第 7 位将会相应变化；如果对 0x2200 401C 进行读操作，那么视 0x2000 0200 的第 7 位状态将会分别返回 0x0000 0001 或 0x0000 0000。

3.2 片上 SRAM

GD32F103xx 系列微控制器提供最大 20kB 的片上 SRAM，起始地址为 0x2000 0000。片上 SRAM 支持按字节（8 位对齐）访问、按半字（16 位对齐）访问、按字（32 位对齐）访问。

3.3 片上 Flash

GD32F103xx 系列微控制器提供最大 128kB 的片上 Flash，支持单周期 32 位的读操作，无任何等待延时。片上 Flash 支持按字节（8 位对齐）访问、按半字（16 位对齐）访问、按字（32 位对齐）访问。Flash 存储器可以单次编程一个半字（16 位）或一个字（32 位）。Flash 存储器的每个页都可以分别擦除。Flash 存储器可以单次执行全片擦除（除信息块之外）。

Flash 存储器最大 128kB，可分为 128 页，每页 1kB。此外片上还包含一个 2kB 的信息块，用于存储引导加载程序（Boot Loader）。详见下表。

表 1 Flash 模块的组织

块	名称	地址范围	大小
主 Flash 块	第 0 页	0x0800 0000 ~ 0x0800 03FF	1kB
	第 1 页	0x0800 0400 ~ 0x0800 07FF	1kB
	第 2 页	0x0800 0800 ~ 0x0800 0BFF	1kB
	
	
	第 127 页	0x0801 FC00 ~ 0x0801 FFFF	1kB
信息块	系统存储器	0x1FFF F000 ~ 0x1FFF F7FF	2kB
	选项字节	0x1FFF F800 ~ 0x1FFF F80F	16B

4 引导配置

GD32F103xx 系列微控制器提供三种引导源，可通过 BOOT1、BOOT0 管脚来配置。BOOT 管脚的状态是在复位后的第 4 个 SYSCLK 上升沿锁存的，因此用户应确保上电复位或系统复位之后 BOOT1、BOOT0 管脚处于正确的状态，才能确保正确选择引导源。详见下表：

表 2 引导模式

引导来源	引导模式选择脚	
	BOOT1	BOOT0
主 Flash 存储器	x	0
系统存储器	0	1
片上 SRAM	1	1

执行完上电序列后、或系统复位后，ARM® Cortex™-M3 处理器将会依次从 0x0000 0000 地址取出栈顶值，从 0x0000 0004 取出引导代码起始地址，之后按照引导代码起始地址开始执行。

根据所选的引导模式，主 Flash 存储器（实际地址 0x0800 0000 开始）或系统存储器（实际地址 0x1FFF F000 开始）将被映射到引导空间内，即从 0x0000 0000 开始。如果用户选用片上 SRAM（实际地址 0x2000 0000 开始）作为引导源，那么在应用程序的初始化代码中，用户必须通过修改 NVIC 异常向量表以及偏移寄存器，将向量表重置到 SRAM 中。

系统存储器中包含内嵌的引导加载程序，可用于对 Flash 存储器编程。在 GD32F103xx 系列微控制器中，引导加载程序是通过 USART1 接口激活工作的。

5 CRC 计算单元

CRC（Cyclic Redundancy Check，循环冗余校验）是一种常用的错误校验机制，多用于数字化网络以及存储设备中，可校验原始数据是否发生意外改变。

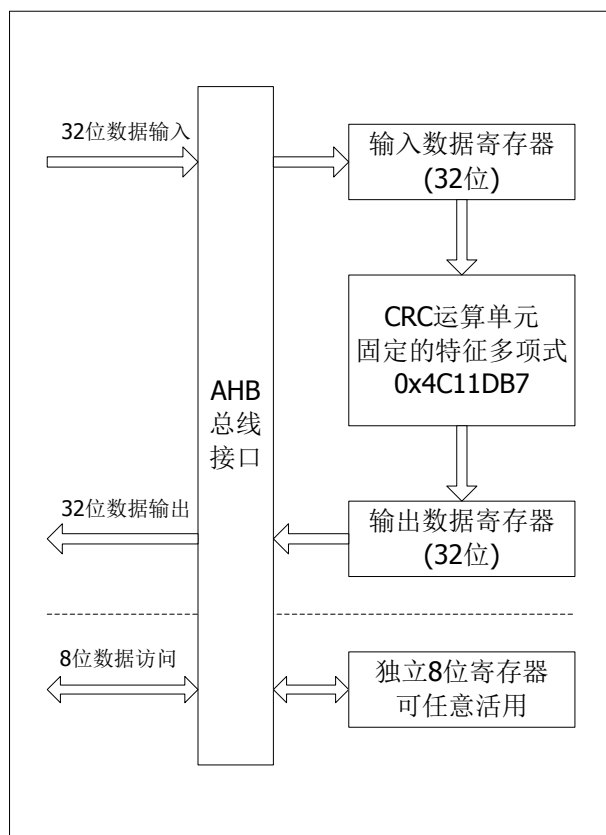
本器件中的 CRC 计算单元仅可计算特征多项式固定、32 位的 CRC 值。

CRC 计算单元具有以下主要特性：

- 32 位数据输入、32 位数据输出。从输入数据到计算结果就绪，所需计算时间为 4 个 AHB 时钟周期。
- 另提供一个与计算无关的通用 8 位寄存器，用户可自行用于任何目的。
- 固定的特征多项式：**0x4C11DB7**

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

图 4 CRC 计算单元框图



CRC 计算单元具有以下功能：

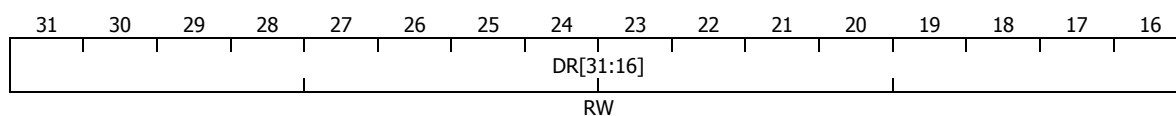
- 用于计算 32 位原始数据的 CRC 校验值。CRC_DR 寄存器接收原始数据，同时也保存计算结果。假如没有事先清除 CRC_DR 寄存器的内容（软件写 CRC_CR 寄存器），那么新输入的 32 位原始数据将会基于 CRC_DR 的原有值继续计算。CRC 计算要消耗 4 个 AHB 总线时钟周期，在此期间 AHB 总线将被 CRC 计算单元挂起，这样可以实现背靠背的数据写入操作、或背靠背的写入-读出操作。
- 本模块提供一个通用 8 位寄存器 CRC_IDR，供用户自由使用。CRC_IDR 与 CRC 计算过程毫无关联，在此寄存器中暂存的值可以随时读出。

5.1 CRC 寄存器

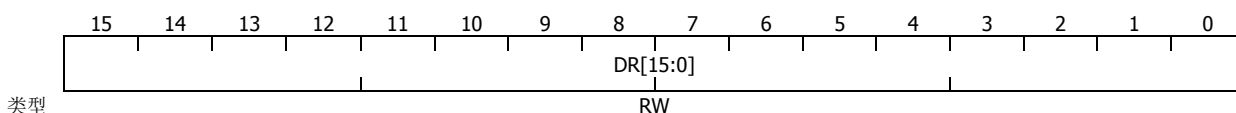
CRC 数据寄存器（CRC_DR）

地址偏移：0x00

复位值：0xFFFF FFFF



类型



类型

位/位域	名称	描述
31:0	DR[31:0]	CRC 计算结果 软件可读可写。 注意：写入的值是无法回读的，因为回读到的值是计算后的结果。

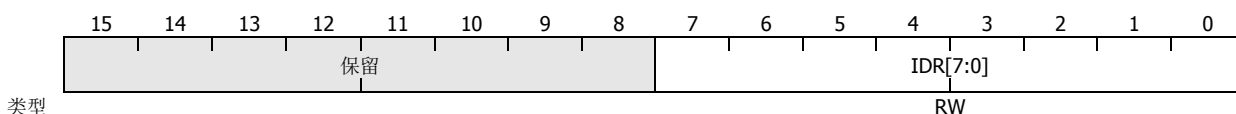
CRC 独立数据寄存器 (CRC_IDR)

地址偏移: 0x04

复位值 : 0x0000 0000



类型



类型

位/位域	名称	描述
31:8	保留	请保持其复位状态
7:0	IDR[7:0]	通用数据寄存器 软件可读可写。 本位域的值与 CRC 计算无关，因此本字节可由任意外设（或软件）用于任何用途。CRC_CR 寄存器对此字节无任何影响。

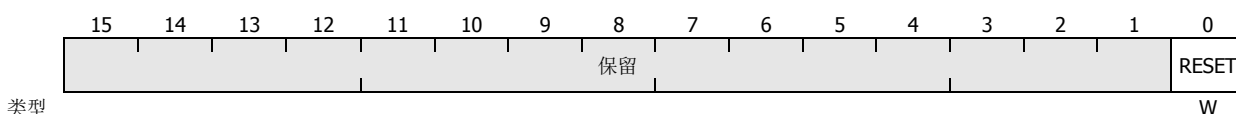
CRC 控制寄存器 (CRC_CR)

地址偏移: 0x08

复位值 : 0x0000 0000



类型



类型

位/位域	名称	描述
31:1	保留	请保持其复位状态
0	RESET	复位 CRC 计算单元 软件对本位写 1，即可复位 CRC 计算单元并将数据寄存器置为 0xFFFF FFFF。 本位只能写 1，由硬件自动清零。

第2章 电源控制（PWR）

1 简介

GD32F103xx 系列微控制器一直将功耗视为最重要的设计方向之一，因此在电源控制单元(PWR)中提供三种降低功耗的工作模式，分别是休眠模式、熟睡模式以及待机模式。这些模式可以降低器件功耗，使应用在 CPU 运行时间、运行速度以及能耗这几个相互冲突的指标之间达到最佳的平衡。芯片内电源部分可划分为三个电源域： V_{DD} 域、1.2V 域以及备电域。 V_{DD} 域的电源来自 V_{DD} 管脚。 V_{DD} 域中的内置 LDO 调压后给 1.2V 域供电。备电域有一个电源开关，当 V_{DD} 电源关闭后从 V_{BAT} 供电。

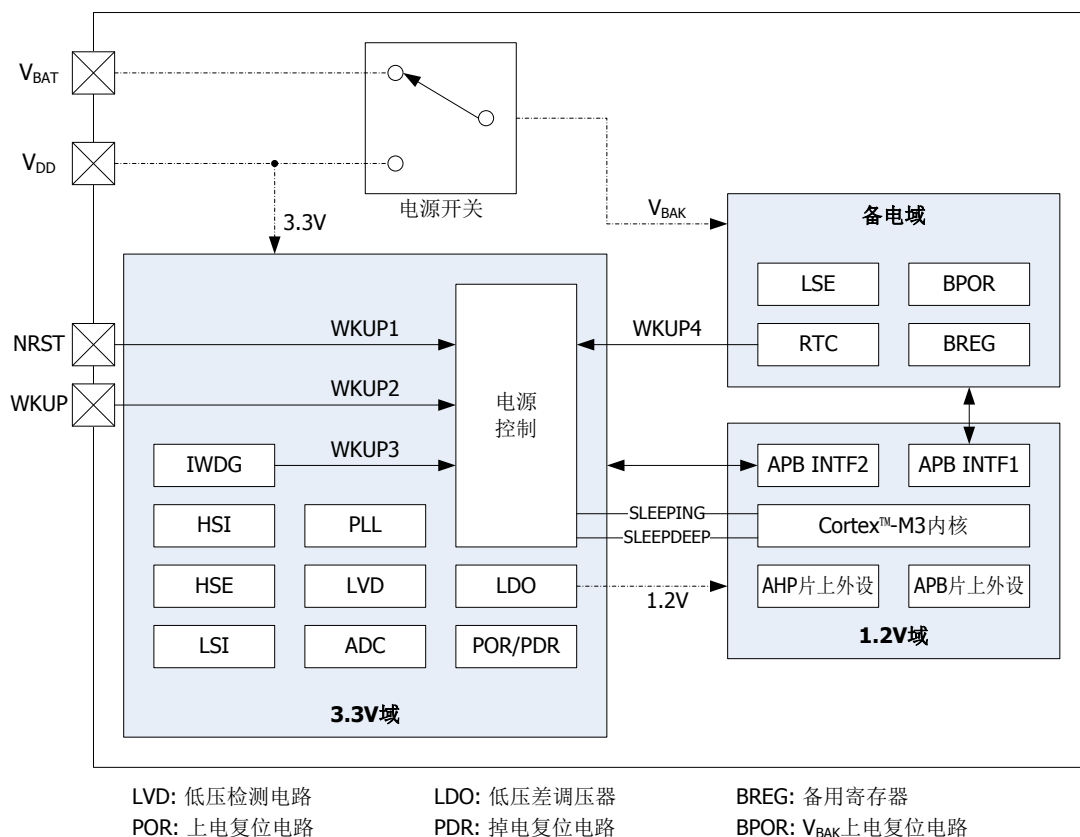
2 主要特点

- 三个电源域： V_{DD} 域、1.2V 域以及备电域
- 三种低功耗工作模式：休眠模式、熟睡模式以及待机模式
- 内部调压器提供 1.2V 电源
- 提供 84 字节由 V_{BAK} 供电的备用寄存器，在待机模式下可保存用户的应用数据
- 当供电低于预设的门限时，低压检测电路可产生中断或唤醒事件
- 当 V_{DD} 电源关闭时，由电池 (V_{BAT}) 为备用电源域供电
- 提供校正寄存器，保存 RTC 校正值

3 功能描述

下图详细描绘出 PWR 模块以及相关电源域的内部配置。

图 5 电源概览



3.1 备电域

备电域根据内部电源开关的选择，可从 V_{DD} 或电池 (V_{BAT}) 供电。为了在 V_{DD} 关断后保持备用寄存器的内容并保持 RTC 的供电，可将 V_{BAT} 管脚连接到备用的供电电源（电池或其它电源）。内部电源开关是由 V_{DD} 域的掉电复位电路控制的。假如应用设计不采用外部电池，那么推荐将 V_{BAT} 通过一个 100nF 的外部陶瓷退耦电容连接到 V_{DD} 上。

备电域的复位源包括备电域上电复位 (BPOR) 以及备电域软件复位。BPOR 信号可将设备保持在复位状态，直至 V_{BAK} 完全上电。应用软件也可以触发备电域软件复位，方法是对 RCC_BDCR 寄存器中的 BKPRST 写 1。

实时时钟 (Real Time Clock, 简称为 RTC) 单元既可以从内部低速 RC 振荡电路 (LSI) 获取时钟，也可以从外部低速晶体振荡器 (LSE) 获取时钟。当 V_{DD} 掉电后，RTC 单元只能采用 LSE 作为时钟源。在执行 WFI/WFE 指令进入节电模式之前，Cortex™-M3 需要先将预期的唤醒时间写入比较寄存器，并且使能唤醒功能以产生 RTC 定时器唤醒事件。这样在进入节电模式一定时间后，当比较匹配事件产生时，便会产生 RTC 闹钟事件唤醒器件。关于 RTC 的配置及操作，详见 RTC 的相关章节。

备电域提供了 42 个 16 位寄存器、总计 84 字节，用于存储用户的应用数据。这些寄存器由 V_{BAK} 供电，当 V_{DD} 电源关闭后可持续供电。只有当备电域上电复位或备电域软件复位时，备用寄存器的内容才会复位。

3.2 V_{DD} 电源域

所有的模拟模块都处于 V_{DD} 电源域内，包括 LDO（电压整流器）、LVD（低压检测模块）、POR/PDR（上电复位/掉电复位模块）、ADC（模数转换器）、HSI（内部高速 RC 振荡电路）、HSE（外部高速晶体振荡电路）、LSI（内部低速 RC 振荡电路）、PLL（锁相环电路）、IWDG（独立式看门狗）以及所有管脚。LDO 为 1.2V 电源域供电，提供三种工作模式：运行模式（最大功耗）、熟睡模式（低功耗）、待机模式（关断）。LVD 用于检测 V_{DD} 是否低于预设值，一旦低于预设值则将产生中断告知 Cortex™-M3 内核。HSI 是默认的系统时钟源，其频率校正到 8MHz 左右。复位后，系统时钟将由 HSI 直接驱动。

3.3 1.2V 电源域

Cortex™-M3 内核逻辑部分、AHB/APB 外设、与备电域/ V_{DD} 电源域的 APB 接口等主要功能模块均位于此电源域内。当 1.2V 上电后，POR 单元将在 1.2V 电源域产生复位序列。复位完成后，如果要进入指定的节电模式，必须先配置相关的控制位，之后一旦执行 WFI（Wait For Interrupt，等待中断）指令或 WFE（Wait For Event，等待事件）指令，器件便可进入该节电模式。关于这方面的详细内容，将在后续章节予以说明。

3.4 节电模式

系统复位或上电复位之后，GD32F103xx 系列微控制器默认开启所有功能，且所有电源域全部通电。若用户希望降低功耗，可降低系统时钟（HCLK、PCLK1、PCLK2），或关闭未用外设的时钟。此外为了进一步降低功耗，器件提供三种节电模式，分别为休眠模式、熟睡模式、待机模式。

休眠模式

休眠模式对应于 Cortex™-M3 的 SLEEPING 工作模式。在休眠模式下，只关闭 Cortex™-M3 内核的时钟。要进入休眠模式，只需将 Cortex™-M3 系统控制寄存器的 SLEEPDEEP 位清零，之后执行 WFI 指令或 WFE 指令。若通过 WFI 指令进入休眠模式，则任意中断都可以唤醒系统。若通过 WFE 指令进入休眠模式，则任意唤醒事件都可以唤醒系统。这种节电模式的唤醒时间最短，因为无需浪费时间在中断入口或出口。

熟睡模式

熟睡模式对应于 Cortex™-M3 的 SLEEPDEEP 工作模式。在熟睡模式下，1.2V 电源域的所有时钟全部关闭，HSI、HSE 以及 PLL 也全部禁用。此时的 LDO 可以工作在正常模式或低功耗模式下，取决于 PWR_CTL 寄存器中 LDO_LP 位的设置。要进入熟睡模式之前，需要将 Cortex™-M3 系统

控制寄存器的 SLEEPDEEP 位置位，并将 PWR_CTL 寄存器的 SDBM 位清零。之后执行 WFI 指令或 WFE 指令后即可进入熟睡模式。任何来自 EXTI 的中断或者唤醒事件都能将系统从熟睡模式中唤醒。当退出熟睡模式时，HSI 将被选作系统时钟。请注意，假如熟睡时 LDO 处于低功耗模式，那么唤醒时还将需要额外的延时。

待机模式

待机模式也是基于 Cortex™-M3 的 SLEEPDEEP 工作模式的。在待机模式下，整个 1.2V 电源域全部停止供电，LDO 关断，HSI、HSE、PLL 全部禁用。要进入待机模式，应将 Cortex™-M3 系统控制寄存器中的 SLEEPDEEP 位置位，并且将 PWR_CTL 寄存器中的 DSPD 置位。之后执行 WFI 指令或 WFE 指令后，系统即进入待机模式。待机模式有四个唤醒源，分别是：NRST 管脚的外部复位信号、RTC 闹钟信号、IWDG 复位信号、WKUP 管脚的上升沿。借助待机模式可以获得最低的功耗，但相应的唤醒时间也是最长的。此外，在待机模式下所有 SRAM 和寄存器（除备用寄存器外）的内容将全部丢失。退出待机模式时将会产生上电复位序列，Cortex™-M3 内核会从 0x0000 0000 开始重新执行指令。

表 3 节电模式小结

模式名称	睡眠模式	深度睡眠模式	待机模式
描述	仅 CPU 时钟被关闭	<ul style="list-style-type: none"> ■ 1.2V 电源域的所有时钟全部关闭 ■ HSI、HSE 以及 PLL 均被禁用 	<ul style="list-style-type: none"> ■ 1.2V 电源域停止供电 ■ HSI、HSE 以及 PLL 均被禁用
LDO 状态	开启	开启或低功耗模式	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 SDBM = 0	SLEEPDEEP = 1 SDBM = 1
入口	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若从 WFI 进入，则任何中断均可唤醒 若从 WFE 进入，则任何事件均可唤醒	来自 EXTI 线的任何中断或事件	<ul style="list-style-type: none"> ■ NRST 管脚 ■ WKUP 管脚 ■ RTC 闹钟 ■ IWDG 复位信号
唤醒延时	无	HSI 唤醒时间 LDO 唤醒时间（假如 LDO 处于低功耗模式）	上电序列

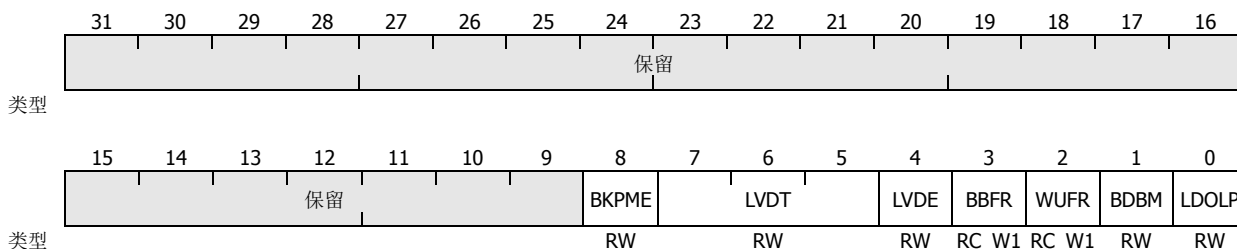
4 PWR 寄存器

外设寄存器可按半字（16 位）或字（32 位）访问。

4.1 电源控制寄存器 (PWR_CTL)

地址偏移: 0x00

复位值 : 0x0000 0000 (从待机模式唤醒后复位)

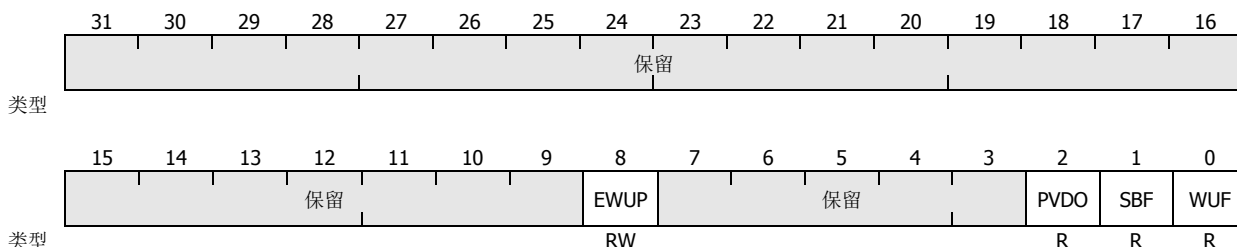


位/位域	名称	描述
31:9	保留	请保持其复位状态
8	BKPWE	备电域写入使能 0: 禁用对备用寄存器的写操作 1: 使能对备用寄存器的写操作 复位后, 所有对备用寄存器的写操作均无效。要想对这些寄存器进行写操作, 必须先将此标志位置位。
7:5	LVDT[2:0]	低压检测电路门限 000: 2.2V 001: 2.3V 010: 2.4V 011: 2.5V 100: 2.6V 101: 2.7V 110: 2.8V 111: 2.9V
4	LVDE	低压检测电路使能 0: 禁用低压检测电路 1: 使能低压检测电路
3	SBFR	待机标志复位 0: 无实际效果 1: 清零待机标志位 本位始终回读为 0。
2	WUFR	唤醒标志复位 0: 无实际效果 1: 清零唤醒标志位 本位始终回读为 0。
1	SDBM	待机模式 0: 当 Cortex™-M3 进入 DEEPSLEEP 模式时, 器件进入熟睡模式 1: 当 Cortex™-M3 进入 DEEPSLEEP 模式时, 器件进入待机模式
0	LDOLP	LDO 低功耗模式 0: 熟睡模式下, LDO 正常工作 1: 熟睡模式下, LDO 按低功耗方式工作

4.2 电源状态寄存器 (PWR_ST)

地址偏移: 0x04

复位值 : 0x0000 0000 (从待机模式唤醒后不复位)

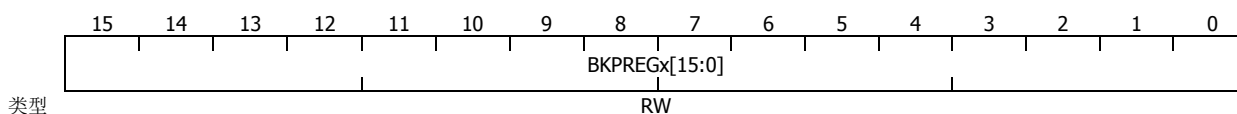


位/位域	名称	描述
31:9	保留	请保持其复位状态
8	WUPE	WAKEUP 管脚使能 0: 禁用 WAKEUP 管脚功能 1: 使能 WAKEUP 管脚功能 假如在进入节电模式之前将 WUPE 置位, 则 WAKEUP 管脚的上升沿可将系统从节电模式中唤醒。 注意: 由于 WAKEUP 管脚是高电平有效的, 因此本位置位时管脚将会打开内部下拉。
7:3	保留	请保持其复位状态
2	LVDF	低压检测电路状态标志 0: 未产生低压事件 (V_{DD} 始终高于预设的门限值) 1: 检测到低压事件 (V_{DD} 低于或等于预设的门限值)
1	SBF	待机标志 0: 系统未处于待机模式中 1: 系统处于待机模式中 仅当发生上电复位/掉电复位、或者将 PWR_CTL 寄存器的 SBFR 置位时, 本标志位才会清零。
0	WUF	唤醒标志 0: 未发生过唤醒事件 1: 检测到来自 WKUP 管脚或 RTC 闹钟的唤醒事件 仅当发生上电复位/掉电复位、或者将 PWR_CTL 寄存器的 SBFR 置位时, 本标志位才会清零。

4.3 备用数据寄存器 x (BKP_REGx) (x = 1~42)

地址偏移: 0x04~0x28, 0x40~0xBC

复位值 : 0x0000

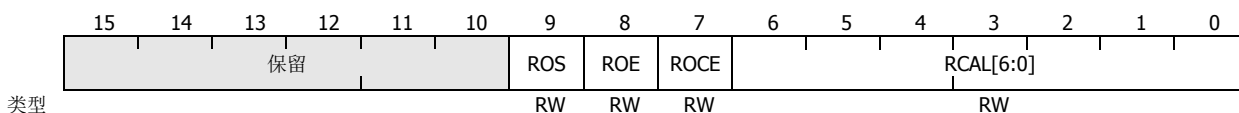


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	BKPREGx	备用寄存器 x (x = 1~42) 这些寄存器用于通用数据存储。即使 1.2V 电源域停止供电, BKPREGx 寄存器的内容也不会丢失。

4.4 RTC 时钟寄存器 (BKP_RTCC)

地址偏移: 0x2C

复位值 : 0x0000

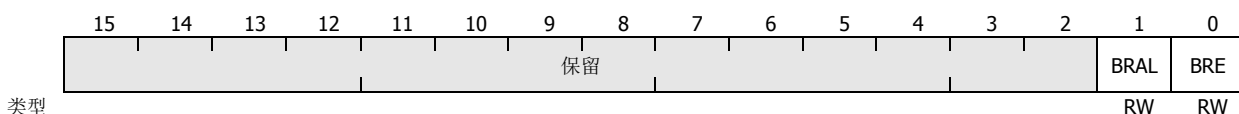


位/位域	名称	描述
31:10	保留	请保持其复位状态
9	ROS	RTC 输出选择 0: 选择 RTC 闹钟输出信号作为 RTC 的输出信号 1: 选择 RTC 秒输出信号作为 RTC 的输出信号 只有当备电域复位时, 本标志位才会复位。
8	ROE	RTC 输出使能 0: 禁用 RTC 输出 1: 使能 RTC 输出 使能之后, RTC 输出信号将通过 BR 管脚输出。 只有当备电域复位时, 本标志位才会复位。
7	RCOE	RTC 时钟输出使能 0: 禁用 RTC 时钟输出 1: 使能 RTC 时钟输出 使能之后, RTC 时钟信号将通过 BR 管脚输出。ROE 的优先级高于 RCOE, 若 ROE 置位, 则不管 RCOE 的值为何, BR 管脚输出的都是 RTC 输出信号。 仅当上电复位/掉电复位后, 本标志位才会复位。
6:0	RCAL[6:0]	RTC 时钟校正 本位域的数值代表每 2^{20} 个时钟脉冲将忽略多少个时钟脉冲。

4.5 备用域复位控制寄存器 (BKP_BRC)

地址偏移: 0x30

复位值 : 0x0000

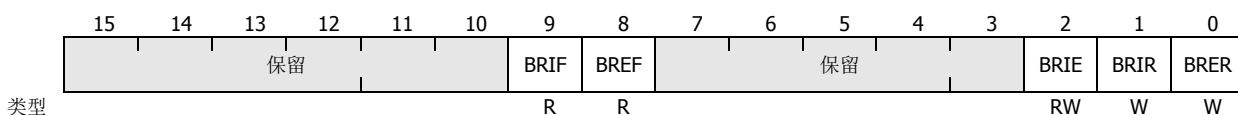


位/位域	名称	描述
31:16	保留	请保持其复位状态
1	BRAL	BR 管脚有效电平 0: 高电平有效 1: 低电平有效
0	BRE	备电域复位使能 0: BR 管脚用作 GPIO 1: BR 管脚专用于备电域复位功能。当 BR 管脚检测到有效电平后，将清除所有 BKPREGx 寄存器的内容。

4.6 备用域复位中断寄存器 (BKP_BRI)

地址偏移: 0x34

复位值 : 0x0000



位/位域	名称	描述
31:10	保留	请保持其复位状态
9	BRIF	备电域复位中断标志 0: 未检测到 BR 中断 1: 检测到 BR 中断 向 BRIR 写 1, 或者当 BRIE 为 0 时, 本标志位将复位。
8	BREF	备电域复位事件标志 0: 未检测到 BR 事件 1: 检测到 BR 事件 向 BRER 写 1 后, 本标志位将复位。
7:3	保留	请保持其复位状态
2	BRIE	备电域复位中断使能 0: 禁用备电域复位中断 1: 使能备电域复位中断 仅当发生系统复位并且从待机模式唤醒后, 本标志位才会复位。
1	BRIR	备电域复位中断复位 0: 无意义 1: 复位 BRIF 标志 本标志位始终回读为 0。
0	BRER	备电域复位事件复位 0: 无意义 1: 复位 BREF 标志 本标志位始终回读为 0。

第3章 复位及时钟控制单元（RCCU）

1 复位控制单元（RCU）

1.1 简介

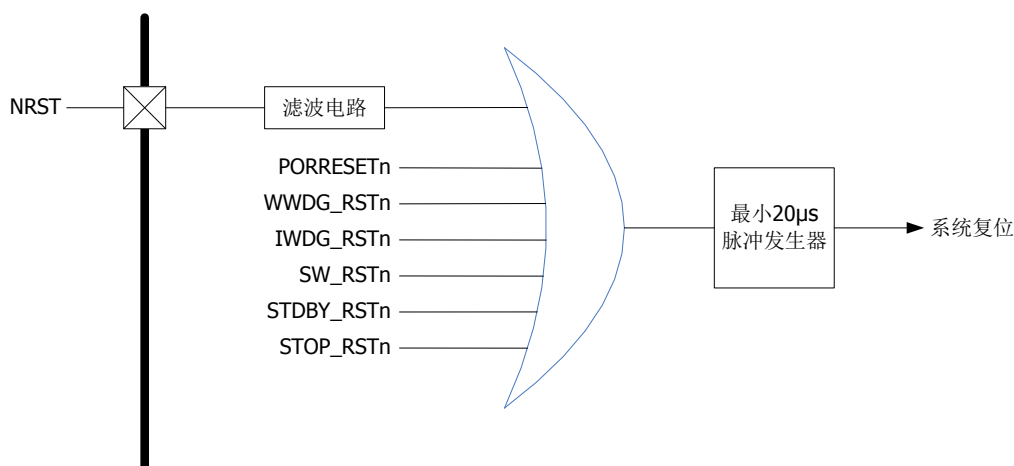
GD32F103xx 系列微控制器的复位控制单元负责控制三种类型的复位：上电复位、系统复位以及备电域复位。上电复位通常又称为冷复位，能够在上电过程中复位除备电域外的整个系统。系统复位可复位处理器内核以及外设 IP 部分，除 SW-DP 控制器及备电域以外。备电域复位则专门用于复位备电域。复位可通过外部信号触发，可通过内部事件触发，还能通过复位发生器触发。本章将对这些内容进行详细说明。

1.2 功能描述

电源复位

电源复位可通过外部复位信号产生（例如上电复位/掉电复位），也可以通过内部复位发生器产生（退出待机模式时）。电源复位将所有寄存器（除备用域外）恢复到初始值。当内部 LDO 调压器已准备好提供 1.2V 电源后，低电平有效的电源复位信号将会转为失效状态。RESET 异常服务子程序的向量固定位于存储器映射表的 0x0000 0004 处。

图 6 系统复位电路框图



系统复位

系统复位可以由下列事件产生：

- 上电复位（PORRESETn）

- 外部管脚复位 (NRST)
- 窗口式看门狗定时器复位 (WWDG_RSTn)
- 独立式看门狗定时器复位 (IWDG_RSTn)
- Cortex™-M3 应用中断及复位控制寄存器中的 SYSRESETREQ 置位 (SW_RSTn)
- 若用户选项字的 nRST_STDBY 被改写, 则每当试图进入待机模式时都将产生系统复位 (STOP_RSTn)
- 若用户选项字的 nRST_STOP 被改写, 则每当试图进入停机模式时都将产生系统复位 (STOP_RSTn)

对任意复位源 (不论是外部复位还是内部复位), 系统复位脉冲发生器可保障产生最短 20 μ s 的低电平脉冲。

备电域复位

将备电域控制寄存器中的 BKPRST 置位, 或者产生备电域上电复位 (假如 V_{DD} 和 V_{BAT} 之前都已经掉电, 则 V_{DD} 或 V_{BAT} 上电即产生备电域上电复位), 即可将备电域复位。

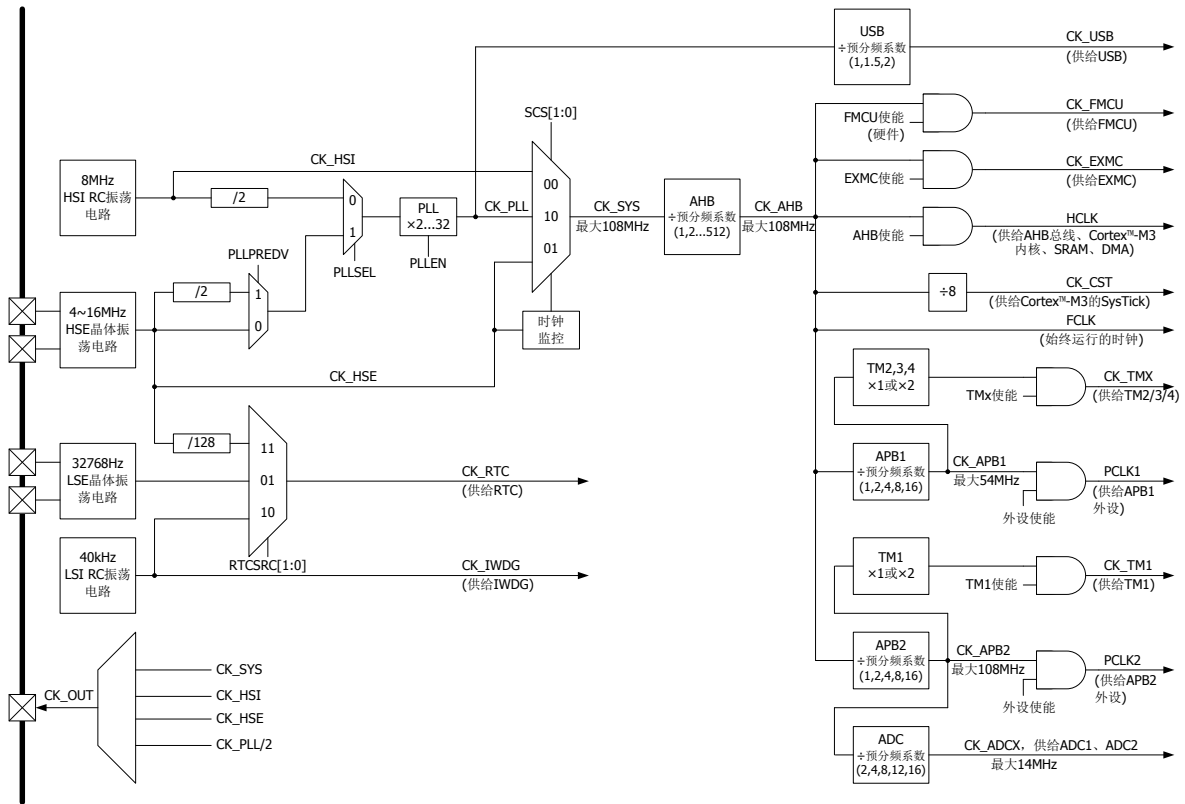
2 时钟控制单元 (CCU)

2.1 简介

时钟控制单元提供各种频率控制及时钟控制功能。这包括一路内部高速 RC 振荡电路 (HSI)、一路外部高速晶体振荡电路 (HSE)、一路内部低速 RC 振荡电路 (LSI)、一路外部低速晶体振荡电路 (LSE)、一路锁相环 (PLL)、一个 HSE 时钟监控电路、时钟预分频器、时钟多路复用开关以及时钟整形电路。

AHB 总线、APB 总线以及 Cortex™-M3 内核的时钟都是从系统时钟 (CK_SYS) 分频得到的。系统时钟可以从 HSI、HSE 或 PLL 获得。系统时钟 (CK_SYS) 的最高频率可达 108MHz。独立式看门狗定时器以及实时时钟 (RTC) 则采用 LSI 或 LSE 作为时钟源。

图 7 时钟树



AHB 频率、高速 APB (APB2) 以及低速 APB (APB1) 时钟域频率，可分别通过预分频器进行配置。AHB 和 APB2 时钟域的最高频率可达 108MHz，APB1 时钟域的最高频率可达 54MHz。Cortex™-M3 系统定时器 (SysTick) 的外部时钟是由 AHB 时钟 (HCLK) 8 分频后得到的。SysTick 模块要么采用这个时钟，要么直接采用 AHB 时钟 (HCLK)，这可以通过系统定时器控制及状态寄存器进行设置。模数转换器都是由高速时钟域 (APB2) 提供时钟的，分频系数可从 2、4、6、8、12 或 16 中选择。

假如 APB 预分频系数为 1，那么定时器时钟频率就是 AHB 频率除以 1，否则就是 AHB 频率除以 APB 预分频系数的一半。

2.2 主要特点

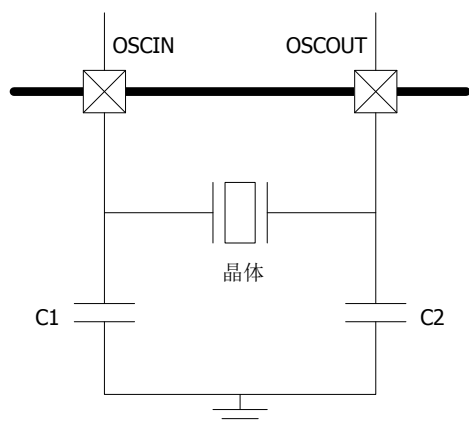
- 4~16MHz 外部高速晶体振荡电路 (HSE)
- 8MHz 内部高速 RC 振荡电路 (HSI)
- 32.768kHz 外部低速晶体振荡电路 (LSE)
- 40kHz 内部低速 RC 振荡电路 (LSI)
- PLL 时钟源可以来自 HSE 或 HSI
- HSE 时钟监控

2.3 功能描述

备用域复位

外部高速晶体振荡电路（HSE）支持 4~16MHz 的外部晶体，可产生高精度的时钟源用作系统时钟。工作时，必须将特定频率的晶体连接到两根 HSE 管脚之间，并且晶体与管脚的距离应尽可能短。为确保振荡可靠性，还需要连接特定的外部电阻和电容元件。

图 8 HSE 时钟源



通过全局时钟控制寄存器 GCC_GCCR 的 HSEEN 位，可以打开或关闭 HSE 晶体振荡电路。查询全局时钟控制寄存器 RCC_GCCR 的 HSESTB 标志位，可表明外部高速晶体的振荡是否稳定。HSE 上电时并不会立即投入使用，直到 HSESTB 标志由硬件置位后才行。这个延迟时间就是众所周知的“晶体起振时间”。当 HSE 工作稳定之后，若全局时钟控制寄存器 RCC_GCCR 的 HSESTBIE=1，则会产生中断。此时 HSE 时钟才能够用作系统时钟源或用作 PLL 的输入时钟。

通过设置全局时钟控制寄存器 RCC_GCCR 的 HSEBPS 位以及 HSEEN 位，即可选中外部时钟旁通模式。CK_HSE 等于驱动 OSC_IN 管脚的外部时钟。

HSI 时钟

内部高速 RC 振荡电路 HSI，具有 8MHz 的固定频率，是器件上电后的默认时钟源。由于无需外接元件，HSI 振荡电路提供了一种低成本的时钟源。通过设置全局时钟控制寄存器 RCC_GCCR 的 HSIEN 位，可开启或关闭 HSI 的 RC 振荡电路。查询全局时钟控制寄存器 RCC_GCCR 的 HSIRSTB 标志，可表明内部 RC 振荡电路是否已经工作稳定。HSI 振荡电路的起振时间要比 HSE 晶体振荡电路的起振时间短。当 HSI 工作稳定之后，若全局时钟控制寄存器 RCC_GCCR 中的 HSISTBIE=1，则会产生中断。HSI 时钟也可以用作 PLL 的时钟输入。

HSI 的频率精度可以由制造商校正，但其运行频率仍不及 HSE 精准。选择何种振荡类型，需结合应用设计要求、工作环境以及成本要求进行综合考量。

若 HSE 或 PLL 作为系统时钟源，为了尽量减少系统从停机模式中恢复所需的时间，当系统刚刚唤醒时会强制使用 HSI 作为系统时钟。

PLL

内部锁相环 PLL 可提供 16~108MHz 的时钟输出，具体输出频率为基频(4~16MHz)的 2~32 倍。

通过全局时钟控制寄存器 RCC_GCCR 的 PLEN 位，可选择开启或关闭 PLL。查询全局时钟控制寄存器 RCC_GCCR 的 PLLSTB 标志，可表明 PLL 时钟是否已经稳定。当 PLL 工作稳定后，若全局时钟控制寄存器 RCC_GCCR 的 PLLSTBIE=1，则会产生中断。

LSE 时钟

32.768kHz 外部低速晶体（或陶瓷谐振器）振荡电路，为实时时钟模块提供了低功耗、高精度的时钟源。通过备电域控制寄存器 RCC_BDCR 的 LSEEN 位，可选择开启或关闭 LSE 振荡电路。查询备用域控制寄存器 RCC_BDCR 的 LSESTB 标志，可表明 LSE 时钟是否已经稳定。当 LSE 工作稳定后，若备用域控制寄存器 RCC_BDCR 中的 LSESTBIE=1，则将产生中断。

通过设置备用域控制寄存器 RCC_BDCR 中的 LSEBPS 位以及 LSEEN 位，可选择外部时钟的旁通模式。CK_LSE 等于驱动 OSC32_IN 管脚的外部时钟频率。

LSI 时钟

内部低速 RC 振荡电路的频率大约为 40kHz，是供给实时时钟电路或独立式看门狗定时器的低功耗时钟源。由于无需外接元件，LSI 提供了低成本的时钟源。通过全局时钟控制寄存器 RCC_GCCR 的 LSIEN 位，可选择开启或关闭 LSI 的 RC 振荡电路。查询全局时钟控制寄存器 RCC_GCCR 的 LSISTB 标志，可表明 LSI 时钟是否已经工作稳定。当 LSI 工作稳定后，若全局时钟控制寄存器 RCC_GCCR 中的 LSISTBIE=1，则将产生中断。

系统时钟（SYS_CLK）的选择

系统复位后，默认的 CK_SYS 时钟源为 HSI，也可以通过修改全局时钟配置寄存器 RCC_GCFGR 中的系统时钟切换 SCS 位域切换为 HSE 或 PLL。当改变 SCS 位域的值时，CK_SYS 仍将采用之前的时钟源进行工作，直到目标时钟源稳定后才会切换。如果某个时钟源是直接连接 CK_SYS 或 PLL 的，那么该时钟源是不可停止的。

HSE 时钟监控器（CKM）

通过全局时钟控制寄存器 RCC_GCCR 的 CKMEN（HSE 时钟监控使能位），可使能 HSE 时钟监控功能。此功能应在 HSE 起振延时之后开启，且应在 HSE 停止振荡之后关闭。一旦检测到 HSE 停振，HSE 将会自动被禁用，此时全局时钟控制寄存器 RCC_GCCR 中的 CKMF（HSE 时钟阻塞标志）将会置位，同时还会产生一个 HSE 故障事件。该故障事件的中断连接在 Cortex™-M3 的不可屏蔽中断 NMI 上。假如 HSE 被选作 CK_SYS 或 PLL 的时钟源，那么 HSE 故障将会强制将 CK_SYS 切换到 HSI，并且自动关闭 PLL。

时钟输出能力

时钟输出的有效频率范围为 32kHz 到 54MHz。通过全局时钟配置寄存器 RCC_GCFGR 的 CKOUTSRC (CKOUT 时钟源选择位域), 可选择数种时钟信号。请注意, 要输出所选的时钟信号, 相应的 GPIO 管脚必须配置为适当的备选功能 I/O (AFIO) 模式。

表 4 CKOUT 的时钟源选择

CKOUTSRC	所选时钟源
0xx	无时钟
100	CK_SYS
101	CK_HSI
110	CK_HSE
111	CK_PLL/2

3 RCCU 寄存器

3.1 全局时钟控制寄存器 (RCC_GCCR)

地址偏移: 0x00

复位值 : 0x0000 XX83 (其中 X 代表未定义)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLL STB	PLL EN	保留				CKM EN	HSE BPS	HSE STB	HSE EN
						R	RW					R	RW	R	RW
类型															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSIADJ[4:0]				保留	HSI STB	HSI EN	
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW		R	RW
类型															

位/位域	名称	描述
31:26	保留	请保持其复位状态
25	PLLSTB	PLL 时钟稳定标志 由硬件置位, 表明 PLL 输出时钟已工作稳定, 可供使用。 0: PLL 未稳定 1: PLL 已稳定
24	PLLEN	PLL 使能 由软件置位及清零。假如 PLL 用作系统时钟源, 则本位不得清零。当进入停机或待机模式时, 硬件自动将本位清零。 0: PLL 被关闭 1: PLL 被打开
23:20	保留	请保持其复位状态

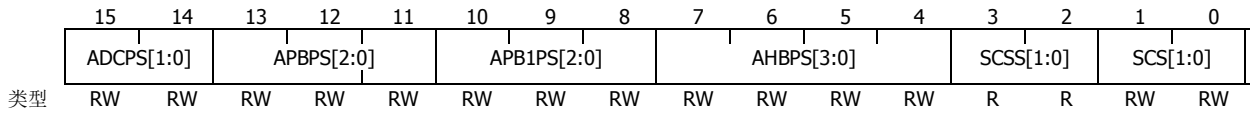
位/位域	名称	描述
19	CKMEN	HSE 时钟监控使能 0: 禁用对外部 4~16MHz 晶体振荡电路 (HSE) 的监控功能 1: 使能对外部 4~16MHz 晶体振荡电路 (HSE) 的监控功能 当硬件检测到 HSE 时钟发生阻塞、始终保持低电平或高电平时, 内部硬件将系统时钟切换到内部高速 RC 振荡电路 (HSI)。要恢复初始系统时钟源, 要么通过外部复位, 要么通过上电复位, 或者软件清零 CKSF。 请注意: 当使能 HSE 时钟监控功能后, 不管 HSIEN 的状态如何, 系统都会自动使能 HSI。
18	HSEBPS	外部晶体振荡电路 (HSE) 时钟旁通模式使能 仅当 HSEEN 为 0 时, 本位才可以执行写操作。 0: 禁用 HSE 旁通模式 1: 使能 HSE 旁通模式
17	HSESTB	外部晶体振荡电路 (HSE) 时钟稳定标志 由硬件置位, 表明 HSE 振荡电路已工作稳定, 可供使用。 0: HSE 振荡电路未稳定 1: HSE 振荡电路已稳定
16	HSEEN	外部高速晶体振荡电路 (HSE) 使能 由软件置位及清零。假如 HSE 用作系统时钟源或 PLL 时钟源, 则本位不得清零。当进入停机或待机模式时, 硬件自动将本位清零。 0: 禁用外部 4~16MHz 晶体振荡电路 1: 使能外部 4~16MHz 晶体振荡电路
15:8	HSICAL[7:0]	内部高速 RC 振荡电路校准值 此数值将在上电时自动加载。
7:3	HSIADJ[4:0]	内部高速 RC 振荡电路时钟调整值 此位域由软件进行设定。调整值将会加入校准值中, 作为 RC 振荡电路的实际校准值。用户应借助调整值将 HSI 调整至 8MHz±1% 范围内。
2	保留	请保持其复位状态
1	HSISTB	内部高速 RC 振荡电路 (HSI) 时钟稳定标志 由硬件置位, 表明 HSI 振荡电路已工作稳定, 可供使用。 0: HSI 振荡电路未稳定 1: HSI 振荡电路已稳定
0	HSIEN	内部高速 RC 振荡电路 (HSI) 使能 由软件置位及清零。假如 HSI 用作系统时钟源, 则本位不得清零。当进入停机或待机模式时, 硬件自动将本位置位。此外, 若 HSECKM 置位, 则当 HSE 时钟发生阻塞、始终保持低电平或高电平时, 硬件也会自动将本位置位。 0: 禁用内部 8MHz RC 振荡电路 1: 使能内部 8MHz RC 振荡电路

3.2 全局时钟配置寄存器 (RCC_GCFGR)

地址偏移: 0x04

复位值 : 0x0000 0000

类型	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		ADC PS[2]	ADC PS[2]	CKOUTSRC[2:0]			保留	USBPS	PLLMF[3:0]			PLL PREDV	PLL SEL		
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW



位/位域	名称	描述
31:29	保留	请保持其复位状态
28	ADCPS[2]	ADCPS 寄存器的第 2 位。 详见后述。
27	PLLMF[4]	PLLMF 寄存器的第 4 位。 详见后述。
26:24	CKOUTSRC	CKOUT 时钟源选择 由软件置位或清零。 0xx: 未选择时钟源 100: 选用系统时钟源 101: 选用内部高速 RC 振荡电路 (HSI) 作为时钟源 110: 选用外部高速晶体振荡电路 (HSE) 作为时钟源 111: 选用 CK_PLL/2 作为时钟源
23:22	USBPS	USB 时钟预分频系数选择 用于配置 USB 的时钟预分频系数, 由软件置位或清零。USB 模块的时钟必须为 48MHz。如果 USB 时钟已被使能, 那么此位域不得清零。 00: CK_PLL/1.5 01: CK_PLL 10: CK_PLL/2.5 11: CK_PLL/2

位/位域	名称	描述
21:18	PLLMF[3:0]	<p>PLL 倍频系数</p> <p>结合前面的 PLLMF[4]组成 5 位的位域，由软件配置以定义 PLL 的倍频系数。</p> <p>00000: 输入时钟×2</p> <p>00001: 输入时钟×3</p> <p>00010: 输入时钟×4</p> <p>00011: 输入时钟×5</p> <p>00100: 输入时钟×6</p> <p>00101: 输入时钟×7</p> <p>00110: 输入时钟×8</p> <p>00111: 输入时钟×9</p> <p>01000: 输入时钟×10</p> <p>01001: 输入时钟×11</p> <p>01010: 输入时钟×12</p> <p>01011: 输入时钟×13</p> <p>01100: 输入时钟×14</p> <p>01101: 输入时钟×15</p> <p>01110: 输入时钟×16</p> <p>01111: 输入时钟×16</p> <p>10000: 输入时钟×17</p> <p>10001: 输入时钟×18</p> <p>10010: 输入时钟×19</p> <p>10011: 输入时钟×20</p> <p>10100: 输入时钟×21</p> <p>10101: 输入时钟×22</p> <p>10110: 输入时钟×23</p> <p>10111: 输入时钟×24</p> <p>11000: 输入时钟×25</p> <p>11001: 输入时钟×26</p> <p>11010: 输入时钟×27</p> <p>11011: 输入时钟×28</p> <p>11100: 输入时钟×29</p> <p>11101: 输入时钟×30</p> <p>11110: 输入时钟×31</p> <p>11111: 输入时钟×32</p> <p>请注意： PLL 输出频率绝对不得超过 108MHz。</p>
17	PLLPREDV	<p>HSE 作为 PLL 时钟源的预分频选择</p> <p>当 HSE 作为 PLL 的时钟源时，本位决定 HSE 是否需要分频后再输入 PLL。由软件置位或清零。</p> <p>0: HSE 时钟直接输入 PLL</p> <p>1: CK_HSE/2 输入 PLL</p>
16	PLLSEL	<p>PLL 时钟源选择</p> <p>由软件置位或清零，决定 PLL 的时钟源。</p> <p>0: CK_HSI/2 作为 PLL 的时钟源</p> <p>1: 选用 HSE 作为 PLL 的时钟源</p>
15:14	ADCPS[1:0]	<p>ADC 时钟预分频选择</p> <p>结合前面的 ADCPS[2]组成 3 位的位域，由软件配置 ADC 时钟频率。</p> <p>000: CK_APB2/2</p> <p>001: CK_APB2/4</p> <p>010: CK_APB2/6</p> <p>011: CK_APB2/8</p> <p>100: CK_APB2/2</p> <p>101: CK_APB2/12</p> <p>110: CK_APB2/8</p> <p>111: CK_APB2/16</p>

位/位域	名称	描述
13:11	APB2PS[2:0]	APB2 时钟预分频系数选择 由软件配置 APB2 时钟预分频系数。 0xx: 直接使用 CK_AHB 100: CK_AHB/2 101: CK_AHB/4 110: CK_AHB/8 111: CK_AHB/16
10:8	APB1PS[2:0]	APB1 时钟预分频系数选择 由软件配置 APB1 时钟预分频系数。 0xx: 直接使用 CK_AHB 100: CK_AHB/2 101: CK_AHB/4 110: CK_AHB/8 111: CK_AHB/16 <i>请注意: CK_APB2 输出时钟频率绝对不得超过 54MHz。</i>
7:4	AHBPS[3:0]	AHB 时钟预分频系数选择 由软件配置 AHB 时钟的分频系数。 0xxx: 直接使用 CK_SYS 1000: CK_SYS/2 1001: CK_SYS/4 1010: CK_SYS/8 1011: CK_SYS/16 1100: CK_SYS/64 1101: CK_SYS/128 1110: CK_SYS/256 1111: CK_SYS/512
3:2	SCSS[1:0]	系统时钟切换状态 由硬件设置, 表明系统时钟的当前时钟源。 00: CK_HSI 01: CK_HSE 10: CK_PLL 11: 保留
1:0	SCS[1:0]	系统时钟切换 由软件选择系统时钟的时钟源。由于改变 CK_SYS 是有原生延迟的, 因此软件必须读取 SCSS 位域来判断时钟切换过程是否已经完成。当退出待机模式或待机模式时, 系统自动将时钟切换至 HSI; 当选用 HSE 作为 CK_SYS 或 PLL 时钟源时, 若 HSE 时钟监控模块检测到时钟阻塞, 系统也会自动将时钟切换至 HSI。 00: CK_HSI 01: CK_HSE 10: CK_PLL 11: 保留

3.3 全局时钟中断寄存器 (RCC_GCIR)

地址偏移: 0x08

复位值 : 0x0000 0000

访问 : 无等待态, 可按字 (32 位)、半字 (16 位)、字节 (8 位) 访问

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								CKMC	保留		PLL STBR	HSE STBR	HSI STBR	LSE STBR	LSI STBR
类型									W			W	W	W	W	W
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		PLL STBIE	HSE STBIE	HSI STBIE	LSE STBIE	LSI STBIE	CKMF	保留		PLL STBF	HSE STBF	HSI STBF	LSE STBF	LSI STBF	
类型			RW	RW	RW	RW	RW	R			R	R	R	R	R	

位/位域	名称	描述
31:24	保留	请保持其复位状态
23	CKMR	HSE 时钟阻塞中断复位 软件对本位写 1，即可清零 CKMF 标志。 0: 无意义 1: 清零 CKMF 标志
22:21	保留	请保持其复位状态
20	PLLSTBR	PLL 稳定中断复位 软件对本位写 1，即可清零 PLLSTBF 标志。 0: 无意义 1: 清零 PLLSTBF 标志
19	HSESTBR	HSE 稳定中断复位 软件对本位写 1，即可清零 HSESTBF 标志。 0: 无意义 1: 清零 HSESTBF 标志
18	HSISTBR	HSI 稳定中断复位 软件对本位写 1，即可清零 HSISTBF 标志。 0: 无意义 1: 清零 HSISTBF 标志
17	LSESTBR	LSE 稳定中断复位 软件对本位写 1，即可清零 LSESTBF 标志。 0: 无意义 1: 清零 LSESTBF 标志
16	LSISTBR	LSI 稳定中断复位 软件对本位写 1，即可清零 LSISTBF 标志。 0: 无意义 1: 清零 LSISTBF 标志
15:13	保留	请保持其复位状态
12	PLLSTBIE	PLL 稳定中断使能 由软件置位或清零，即可使能/禁用 PLL 稳定中断。 0: 禁用 PLL 稳定中断 1: 使能 PLL 稳定中断
11	HSESTBIE	HSE 稳定中断使能 由软件置位或清零，即可使能/禁用 HSE 稳定中断。 0: 禁用 HSE 稳定中断 1: 使能 HSE 稳定中断

位/位域	名称	描述
10	HSISTBIE	HSI 稳定中断使能 由软件置位或清零，即可使能/禁用 HSI 稳定中断。 0: 禁用 HSI 稳定中断 1: 使能 HSI 稳定中断
9	LSESTBIE	LSE 稳定中断使能 由软件置位或清零，即可使能/禁用 LSE 稳定中断。 0: 禁用 LSE 稳定中断 1: 使能 LSE 稳定中断
8	LSISTBIE	LSI 稳定中断使能 由软件置位或清零，即可使能/禁用 LSI 稳定中断。 0: 禁用 LSI 稳定中断 1: 使能 LSI 稳定中断
7	CKMF	HSE 时钟阻塞中断标志 当 HSE 时钟发生阻塞时，硬件将本位置位。当软件对 CKMR 写 1 时，本位即清零。 0: HSE 时钟工作正常 1: HSE 时钟阻塞
6:5	保留	请保持其复位状态
4	PLLSTBF	PLL 稳定中断标志 当 PLL 时钟稳定、并且 PLLSTBIE=1 时，硬件将本位置位。当软件对 PLLSTBR 写 1 时，本位即清零。 0: 未产生 PLL 稳定中断 1: 已产生 PLL 稳定中断
3	HSESTBF	HSE 稳定中断标志 当外部 4~16MHz 晶体振荡时钟稳定、并且 HSESTBIE=1 时，硬件将本位置位。当软件对 HSESTBR 写 1 时，本位即清零。 0: 未产生 HSE 稳定中断 1: 已产生 HSE 稳定中断
2	HSISTBF	HSI 稳定中断标志 当内部 8MHz RC 振荡时钟稳定、并且 HSISTBIE=1 时，硬件将本位置位。当软件对 HSISTBR 写 1 时，本位即清零。 0: 未产生 HSI 稳定中断 1: 已产生 HSI 稳定中断
1	LSESTBF	LSE 稳定中断标志 当外部 32768Hz 时钟稳定、并且 LSESTBIE=1 时，硬件将本位置位。当软件对 LSESTBR 写 1 时，本位即清零。 0: 未产生 LSE 稳定中断 1: 已产生 LSE 稳定中断
0	LSISTBF	LSI 稳定中断标志 当内部 32kHz 时钟稳定、并且 LSISTBIE=1 时，硬件将本位置位。当软件对 LSISTBR 写 1 时，本位即清零。 0: 未产生 LSI 稳定中断 1: 已产生 LSI 稳定中断

3.4 APB2 复位控制寄存器 (RCC_APB2RCR)

地址偏移: 0x0C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART1 RST	保留	SPI1 RST	TM1 RST	ADC2 RST	ADC1 RST	保留		PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST
	RW		RW	RW	RW	RW			RW	RW	RW	RW	RW		RW

类型

类型

位/位域	名称	描述
31:15	保留	请保持其复位状态
14	USART1RST	USART1 复位 由软件置位或清零。 0: 未复位 1: 复位 USART1
13	保留	请保持其复位状态
12	SPI1RST	SPI1 复位 由软件置位或清零。 0: 未复位 1: 复位 SPI1
11	TM1RST	定时器 1 复位 由软件置位或清零。 0: 未复位 1: 复位定时器 1
10	ADC2RST	ADC2 复位 由软件置位或清零。 0: 未复位 1: 复位 ADC2
9	ADC1RST	ADC1 复位 由软件置位或清零。 0: 未复位 1: 复位 ADC1
8:7	保留	请保持其复位状态
6	PERST	GPIO 端口 E 复位 由软件置位或清零。 0: 未复位 1: 复位 GPIO 端口 E
5	PDRST	GPIO 端口 D 复位 由软件置位或清零。 0: 未复位 1: 复位 GPIO 端口 D
4	PCRST	GPIO 端口 C 复位 由软件置位或清零。 0: 未复位 1: 复位 GPIO 端口 C

位/位域	名称	描述
3	PBRST	GPIO 端口 B 复位 由软件置位或清零。 0: 未复位 1: 复位 GPIO 端口 B
2	PARST	GPIO 端口 A 复位 由软件置位或清零。 0: 未复位 1: 复位 GPIO 端口 A
1	保留	请保持其复位状态
0	AFRST	备选功能 I/O 复位 由软件置位或清零。 0: 未复位 1: 复位备选功能 I/O

3.5 APB1 复位控制寄存器 (RCC_APB1RCR)

地址偏移: 0x10

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			PWR RST	BKP RST	保留	CAN RST	保留	USB RST	I2C2 RST	I2C1 RST	保留	USART 3 RST	USART 2 RST	保留	
			RW	RW		RW		RW	RW	RW		RW	RW		
类型															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI2 RST	保留		WWD GRST	保留							TM4 RST	TM3 RST	TM2 RST	
	RW			RW								RW	RW	RW	
类型															

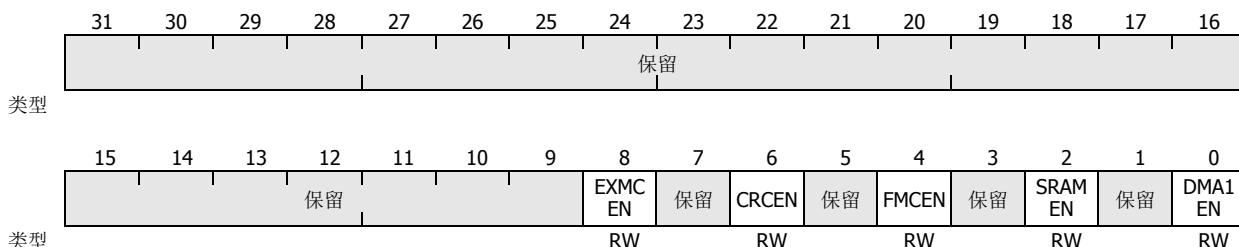
位/位域	名称	描述
31:29	保留	请保持其复位状态
28	PWRRST	电源控制复位 由软件置位或清零。 0: 未复位 1: 复位电源控制单元
27	BKPRST	备电域接口复位 由软件置位或清零。 0: 未复位 1: 复位备电域接口
26	保留	请保持其复位状态
25	CANRST	CAN 复位 由软件置位或清零。 0: 未复位 1: 复位 CAN
24	保留	请保持其复位状态

位/位域	名称	描述
23	USBRST	USB 复位 由软件置位或清零。 0: 未复位 1: 复位 USB
22	I2C2RST	I ² C2 复位 由软件置位或清零。 0: 未复位 1: 复位 I ² C2
21	I2C1RST	I ² C1 复位 由软件置位或清零。 0: 未复位 1: 复位 I ² C1
20:19	保留	请保持其复位状态
18	USART3RST	USART3 复位 由软件置位或清零。 0: 未复位 1: 复位 USART3
17	USART2RST	USART2 复位 由软件置位或清零。 0: 未复位 1: 复位 USART2
16:15	保留	请保持其复位状态
14	SPI2RST	SPI2 复位 由软件置位或清零。 0: 未复位 1: 复位 SPI2
13:12	保留	请保持其复位状态
11	WWDGRST	窗口式看门狗复位 由软件置位或清零。 0: 未复位 1: 复位窗口式看门狗
10:3	保留	请保持其复位状态
2	TM4RST	定时器 4 复位 由软件置位或清零。 0: 未复位 1: 复位定时器 4
1	TM3RST	定时器 3 复位 由软件置位或清零。 0: 未复位 1: 复位定时器 3
0	TM2RST	定时器 2 复位 由软件置位或清零。 0: 未复位 1: 复位定时器 2

3.6 AHB 时钟控制寄存器 (RCC_AHBCCR)

地址偏移: 0x14

复位值 : 0x0000 0014



位/位域	名称	描述
31:9	保留	请保持其复位状态
8	EXMCEN	EXMC 时钟使能 由软件置位或清零。 0: 禁用 EXMC 时钟 1: 使能 EXMC 时钟
7	保留	请保持其复位状态
6	CRCEN	CRC 时钟使能 由软件置位或清零。 0: 禁用 CRC 时钟 1: 使能 CRC 时钟
5	保留	请保持其复位状态
4	FMCEN	FLITF 时钟使能 由软件置位或清零, 使能/禁用睡眠模式下的 FLITF 时钟。 0: 睡眠模式下, 禁用 FLITF 时钟 1: 睡眠模式下, 使能 FLITF 时钟
3	保留	请保持其复位状态
2	SRAMEN	SRAM 接口时钟使能 由软件置位或清零, 使能/禁用睡眠模式下的 SRAM 接口时钟。 0: 睡眠模式下, 禁用 SRAM 时钟 1: 睡眠模式下, 使能 SRAM 时钟
1	保留	请保持其复位状态
0	DMA1EN	DMA1 时钟使能 由软件置位或清零。 0: 禁用 DMA1 时钟 1: 使能 DMA1 时钟

3.7 APB2 时钟控制寄存器 (RCC_APB2CCR)

地址偏移: 0x18

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART 1 EN	保留	SPI1 EN	TM1EN	ADC2 EN	ADC1 EN	保留	PEEN	PDEN	PCEN	PBEN	PAEN	保留	AFEN	
	RW		RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	

类型

类型

位/位域	名称	描述
31:15	保留	请保持其复位状态
14	USART1EN	USART1 时钟使能 由软件置位或清零。 0: 禁用 USART1 时钟 1: 使能 USART1 时钟
13	保留	请保持其复位状态
12	SPI1EN	SPI1 时钟使能 由软件置位或清零。 0: 禁用 SPI1 时钟 1: 使能 SPI1 时钟
11	TM1EN	定时器 1 时钟使能 由软件置位或清零。 0: 禁用定时器 1 时钟 1: 使能定时器 1 时钟
10	ADC2EN	ADC2 接口时钟使能 由软件置位或清零。 0: 禁用 ADC2 接口时钟 1: 使能 ADC2 接口时钟
9	ADC1EN	ADC1 接口时钟使能 由软件置位或清零。 0: 禁用 ADC1 接口时钟 1: 使能 ADC1 接口时钟
8:7	保留	请保持其复位状态
6	PEEN	GPIO 端口 E 时钟使能 由软件置位或清零。 0: 禁用 GPIO 端口 E 时钟 1: 使能 GPIO 端口 E 时钟
5	PDEN	GPIO 端口 D 时钟使能 由软件置位或清零。 0: 禁用 GPIO 端口 D 时钟 1: 使能 GPIO 端口 D 时钟
4	PCEN	GPIO 端口 C 时钟使能 由软件置位或清零。 0: 禁用 GPIO 端口 C 时钟 1: 使能 GPIO 端口 C 时钟

位/位域	名称	描述
3	PBEN	GPIO 端口 B 时钟使能 由软件置位或清零。 0: 禁用 GPIO 端口 B 时钟 1: 使能 GPIO 端口 B 时钟
2	PAEN	GPIO 端口 A 时钟使能 由软件置位或清零。 0: 禁用 GPIO 端口 A 时钟 1: 使能 GPIO 端口 A 时钟
1	保留	请保持其复位状态
0	AFEN	备选功能 I/O 时钟使能 由软件置位或清零。 0: 禁用备选功能 I/O 时钟 1: 使能备选功能 I/O 时钟

3.8 APB1 时钟控制寄存器 (RCC_APB1CCR)

地址偏移: 0x1C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			PWREN	BKPEN	保留	CANEN	保留	USBEN	I2C2 EN	I2C1 EN	保留		USART 3 EN	USART 2 EN	保留
			RW	RW		RW		RW	RW	RW			RW	RW	
类型															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI2 EN	保留		WWD GEN	保留							TM4EN	TM3EN	TM2EN	
	RW			RW								RW	RW	RW	
类型															

位/位域	名称	描述
31:29	保留	请保持其复位状态
28	PWREN	电源接口时钟使能 由软件置位或清零。 0: 禁用电源接口时钟 1: 使能电源接口时钟
27	BKPEN	备用域接口时钟使能 由软件置位或清零。 0: 禁用备用域接口时钟 1: 使能备用域接口时钟
26	保留	请保持其复位状态
25	CANEN	CAN 时钟使能 由软件置位或清零。 0: 禁用 CAN 时钟 1: 使能 CAN 时钟
24	保留	请保持其复位状态

位/位域	名称	描述
23	USBEN	USB 时钟使能 由软件置位或清零。 0: 禁用 USB 时钟 1: 使能 USB 时钟
22	I2C2EN	I ² C2 时钟使能 由软件置位或清零。 0: 禁用 I ² C2 时钟 1: 使能 I ² C2 时钟
21	I2C1EN	I ² C1 时钟使能 由软件置位或清零。 0: 禁用 I ² C1 时钟 1: 使能 I ² C1 时钟
20:19	保留	请保持其复位状态
18	USART3EN	USART3 时钟使能 由软件置位或清零。 0: 禁用 USART3 时钟 1: 使能 USART3 时钟
17	USART2EN	USART2 时钟使能 由软件置位或清零。 0: 禁用 USART2 时钟 1: 使能 USART2 时钟
16:15	保留	请保持其复位状态
14	SPI2EN	SPI2 时钟使能 由软件置位或清零。 0: 禁用 SPI2 时钟 1: 使能 SPI2 时钟
13:12	保留	请保持其复位状态
11	WWDGEN	窗口式看门狗时钟使能 由软件置位或清零。 0: 禁用窗口式看门狗时钟 1: 使能窗口式看门狗时钟
10:3	保留	请保持其复位状态
2	TM4EN	定时器 4 时钟使能 由软件置位或清零。 0: 禁用定时器 4 时钟 1: 使能定时器 4 时钟
1	TM3EN	定时器 3 时钟使能 由软件置位或清零。 0: 禁用定时器 3 时钟 1: 使能定时器 3 时钟
0	TM2EN	定时器 2 时钟使能 由软件置位或清零。 0: 禁用定时器 2 时钟 1: 使能定时器 2 时钟

3.9 备电域控制寄存器 (RCC_BDCR)

地址偏移: 0x20

复位值 : 0x0000 0000 (由备电域复位命令复位)

请注意: 备电域控制寄存器 (RCC_BDCR) 的 LSEEN、LSEBPS、RTC SRC、RTCEN 位仅在执行备电域复位命令之后才会清零。只有当电源控制寄存器 (PWR_CR) 的 DBP 置位的情况下, 这些位才可以修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															BKP RST
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	保留						RTCSRC[1:0]	保留				LSE BPS	LSE STB	LSEEN	
RW							RW	RW					RW	RW	RW

位/位域	名称	描述
31:17	保留	请保持其复位状态
16	BKPRST	备电域复位 由软件置位或清零。 0: 无意义 1: 复位备电域
15	RTCEN	RTC 时钟使能 由软件置位或清零。 0: 禁用 RTC 时钟 1: 使能 RTC 时钟
14:10	保留	请保持其复位状态
9:8	RTCSRC[1:0]	RTC 时钟源选择 由软件配置 RTC 的时钟源。 00: 无时钟输入 01: CK_LSE 10: CK_LSI 11: CK_HSE/128
7:3	保留	请保持其复位状态
2	LSEBPS	LSE 旁通模式使能 由软件置位或清零。 0: 禁用 LSE 旁通模式 1: 使能 LSE 旁通模式
1	LSESTB	外部低速晶体振荡电路稳定 由硬件置位, 表明 LSE 振荡电路已工作稳定, 可供使用。 0: LSE 振荡电路未稳定 1: LSE 振荡电路已稳定
0	LSEEN	LSE 使能 由软件置位或清零。 0: 禁用 LSE 1: 使能 LSE

3.10 全局控制/状态寄存器 (RCC_GCSR)

地址偏移: 0x24

复位值 : 0x0C00 0000 (其中复位类标志由电源复位清零, 其它标志由系统复位清零)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LP RSTF	WWDG RSTF	IWDG RSTF	SW RSTF	PWR RSTF	EP RSTF	保留	RSTFC	保留							
类型	RW	RW	RW	RW	RW	RW		RW								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														LSI STB	LSIEN
类型															R	RW

位/位域	名称	描述
31	LPRSTF	低功耗复位标志 当停机模式/待机模式下产生复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 低功耗模式下未产生复位 1: 低功耗模式下产生了复位
30	WWDGRSTF	窗口式看门狗定时器复位标志 当窗口式看门狗定时器产生复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 未产生窗口式看门狗定时器复位 1: 已产生窗口式看门狗定时器复位
29	IWDGRSTF	独立式看门狗定时器复位标志 当独立式看门狗定时器产生复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 未产生独立式看门狗定时器复位 1: 已产生独立式看门狗定时器复位
28	SWRSTF	软件复位标志 当产生软件复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 未产生软件复位 1: 已产生软件复位
27	PWRGRSTF	电源复位标志 当产生电源复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 未产生电源复位 1: 已产生电源复位
26	EPRSTF	外部管脚复位标志 当外部管脚产生复位时, 由硬件置位。当软件向 RSTFC 写 1 时, 本标志位清零。 0: 未产生外部管脚复位 1: 已产生外部管脚复位
25	保留	请保持其复位状态

位/位域	名称	描述
24	RSTFC	复位标志清除 软件对本位写 1，即可清零所有复位标志。 0: 无意义 1: 清零所有复位标志
23:2	保留	请保持其复位状态
1	LSISTB	内部低速 RC 振荡电路稳定 由硬件置位，表明 LSI 振荡电路已工作稳定，可供使用。 0: LSI 振荡电路未稳定 1: LSI 振荡电路已稳定
0	LSIEN	LSI 使能 由软件置位或清零。 0: 禁用 LSI 1: 使能 LSI

第4章 通用 I/O（GPIO）及备选功能 I/O（AFIO）

1 GPIO

1.1 简介

通用 I/O 接口包含 5 个通用的输入输出端口。每个端口包含 16 个独立的通用管脚，具备输入功能以及输出功能；因此，通用 I/O 接口最多可支持 80（16×5）根管脚。每个 GPIO 端口都有相关的控制寄存器和配置寄存器，以满足不同应用的要求。至于 GPIO 管脚的外部中断，在外部中断控制单元（External Interrupt Control Unit，简称为 EXTI）中有相关的控制寄存器和配置寄存器。

GPIO 端口与其它的可选功能（Alternative Function，简称为 AF）是管脚复用的，这样可以保证封装管脚运用的最大灵活性。通过配置相应的寄存器，可将 GPIO 管脚用作备选功能，无需考虑备选功能是输入还是输出。

每个 GPIO 管脚可通过软件配置为输出（推挽输出或开漏输出）功能、输入（有无上/下拉）功能或者外设的可选功能。大部分 GPIO 管脚是与数字备选功能或模拟备选功能复用的。除模拟输入脚外，其余的 GPIO 都具备大电流驱动能力。

1.2 主要特点

- 输入/输出方向控制
- 输入管脚可配置弱上拉/下拉
- 输出管脚可配置为推挽或开漏
- 输出管脚的置位/复位控制
- 可用作外部中断，触发沿可设置（需结合 EXTI 配置寄存器）
- 可配置为模拟输入/输出
- 可配置为备选功能输入/输出
- 端口配置锁定

1.3 功能描述

每个 GPIO 管脚都可通过软件单独配置为输入或输出，并能在多种工作模式中进行选择。配置是通过两个 32 位配置寄存器（GPIOx_CR1、GPIOx_CR2）以及两个 32 位数据寄存器（GPIOx_DIR、GPIOx_DOR）。详见下表：

表 5 GPIO 端口配置位

配置模式		CMF1	CMF0	MODE1	MODE0	PxODR 寄存器
输入	模拟输入	0	0	00		无关
	浮空输入		1			无关
	弱下拉输入	1	0			0
	弱上拉输入					1
通用输出	推挽输出	0	0	00: 保留	0 或 1	
	开漏输出		1	01: 速度最高 10MHz	0 或 1	
备选功能输出	推挽输出	1	0	10: 速度最高 2MHz	无关	
	开漏输出		1	11: 速度最高 50MHz	无关	

下图是 I/O 管脚的基本结构。

图 9 标准 I/O 管脚的基本结构

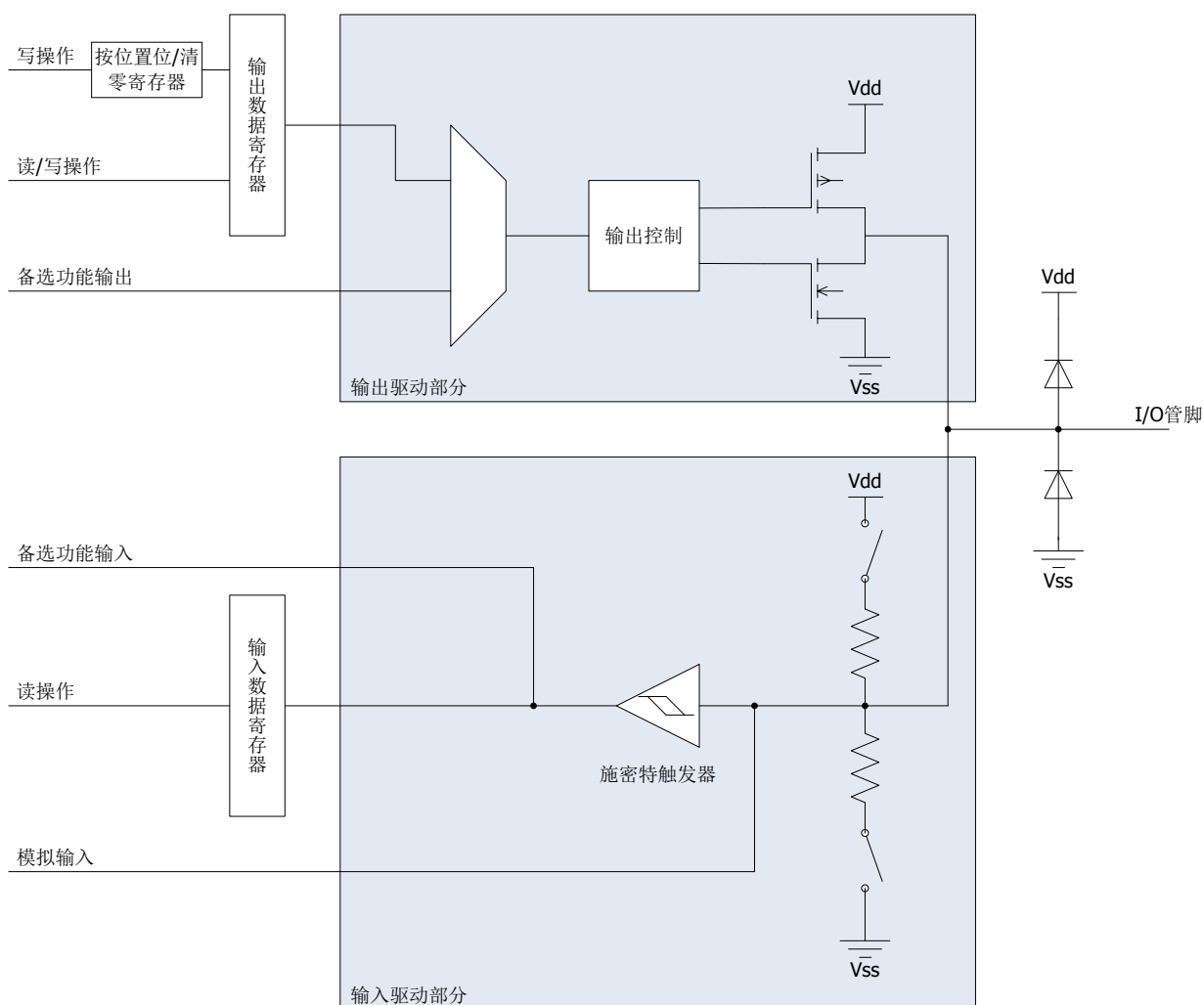
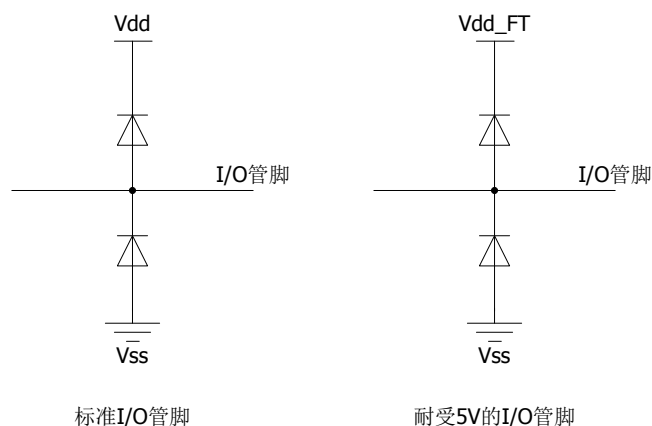


图 10 标准 I/O 管脚与可耐受 5V 的 I/O 管脚



GPIO 管脚的默认配置

在复位期间和复位之后，备选功能并未激活，I/O 口皆配置为浮空输入模式。不过，与调试功能复用的管脚（PA15、PA14、PA13、PB4）在复位后是激活的：

PA15: JTDI 功能，带上拉

PA14: JTCK 功能，带下拉

PA13: JTMS 功能，带上拉

PB4: NJTRST 功能，带上拉

GPIO 管脚可以配置为输入管脚或输出管脚。当 GPIO 管脚配置为输入管脚时，外部管脚上的数据（电平）每个 APB2 时钟周期都会装载到数据输入寄存器（GPIO_DIR）。所有 GPIO 管脚都具有内部弱上拉和弱下拉的选项。

当 GPIO 管脚配置为输出管脚时，用户可以配置该管脚的最大翻转速度，并且选择输出驱动的模式：推挽模式或开漏模式。数据输出寄存器（GPIO_DOR）的值将会输出到 I/O 管脚上。

所有管脚均具有外部中断能力。要使用外部中断，则管脚必须配置为输入模式。当需要对 GPIO_DOR 寄存器进行按位写操作时，不需要禁用中断，用户可以对位操作寄存器（GPIO BOR，或专用于清零的 GPIO_BCR）的单个/多个位写 ‘1’，即可修改单个/多个管脚的输出，并且此过程仅需单个原子级的 APB2 写操作。此时其它位将不受影响。

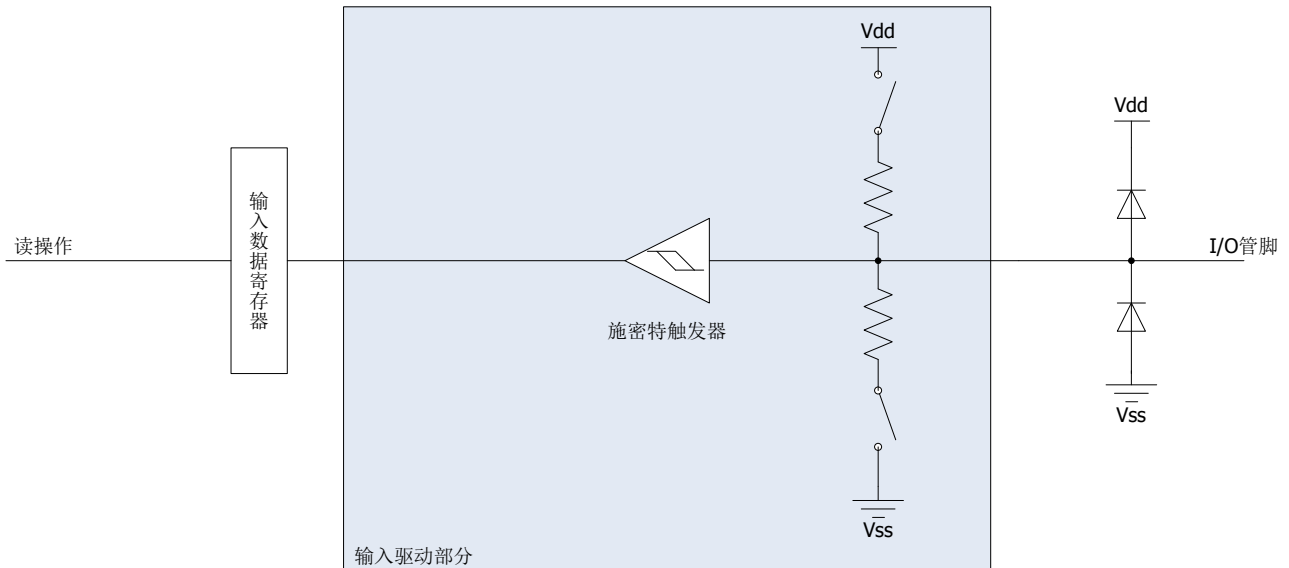
输入配置

当 GPIO 管脚配置为输入管脚时：

- 输入施密特触发器将被激活
- 可选择弱上拉以及弱下拉
- I/O 管脚上的数据（电平）每个 APB2 时钟周期都被采样并送入数据输入寄存器
- 输出缓冲被禁用

下图是 I/O 管脚配置为输入时的状态：

图 11 浮空输入/上拉输入/下拉输入的配置



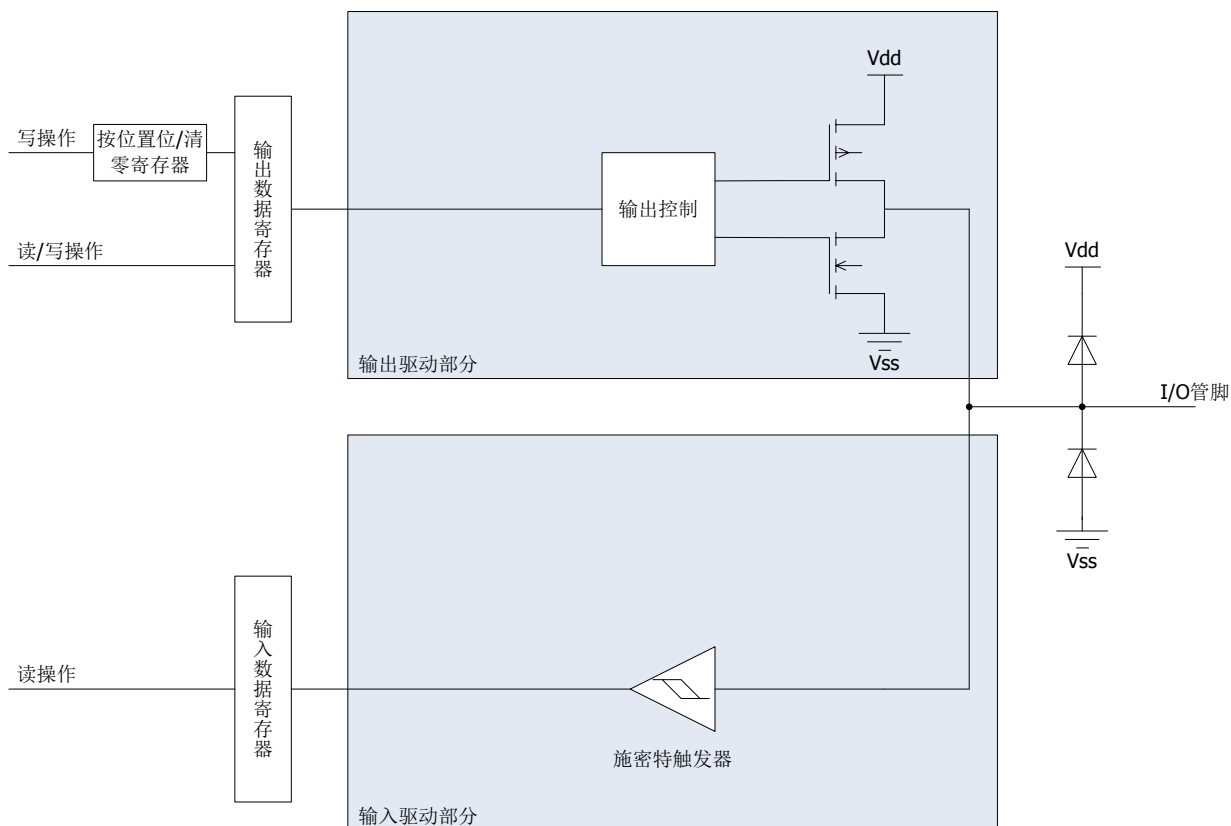
输出配置

当 GPIO 管脚配置为输出管脚时：

- 输入施密特触发器将被激活
- 弱上拉以及弱下拉电阻被禁用
- 输出缓冲器被使能：
 - 开漏模式：输出寄存器的“0”将使 NMOS 管导通，输出寄存器的“1”将使得管脚处于高阻状态。
 - 推挽模式：输出寄存器的“0”将使 NMOS 管导通，输出寄存器的“1”将使 PMOS 管导通。
- 在推挽模式下，对数据输出寄存器的读操作，将返回其上次写入的值
- 在开漏模式下，对数据输入寄存器的读操作，将返回其 I/O 状态

下图为 I/O 管脚配置为输出时的状态：

图 12 输出配置



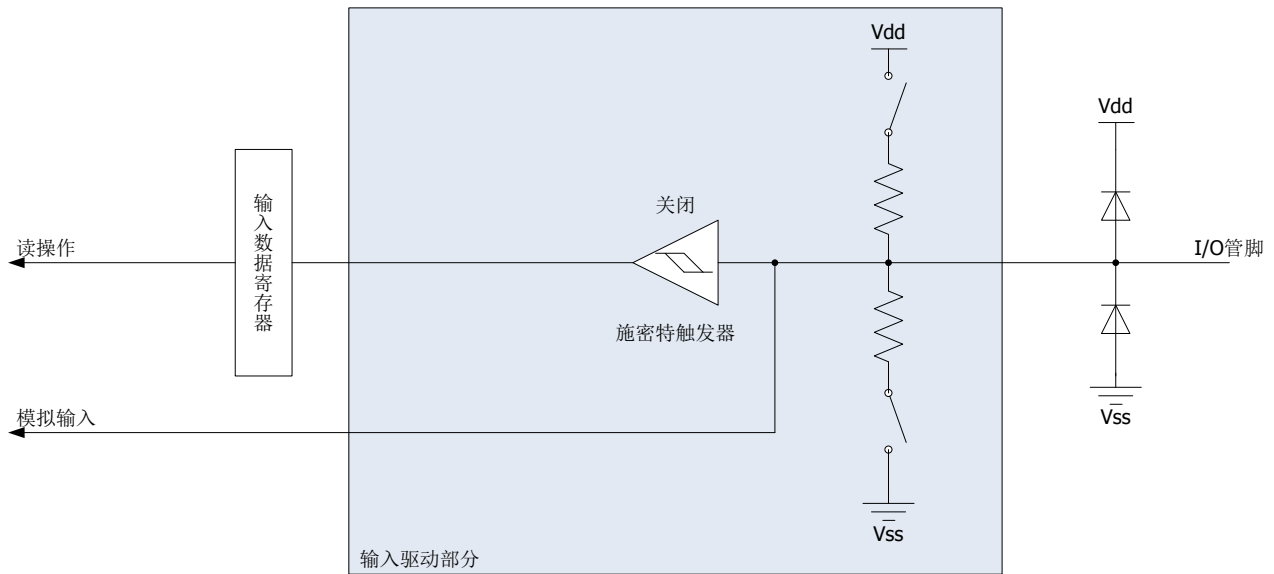
模拟输入配置

当 GPIO 管脚配置为模拟输入管脚时：

- 弱上拉以及弱下拉电阻被禁用
- 输出缓冲器被禁用
- 输入的施密特触发器被禁用
- 对数据输入寄存器的读操作，始终返回“0”

下图为 I/O 管脚配置为高阻模拟输入时的状态：

图 13 高阻模拟输入配置



备选功能（AF）配置

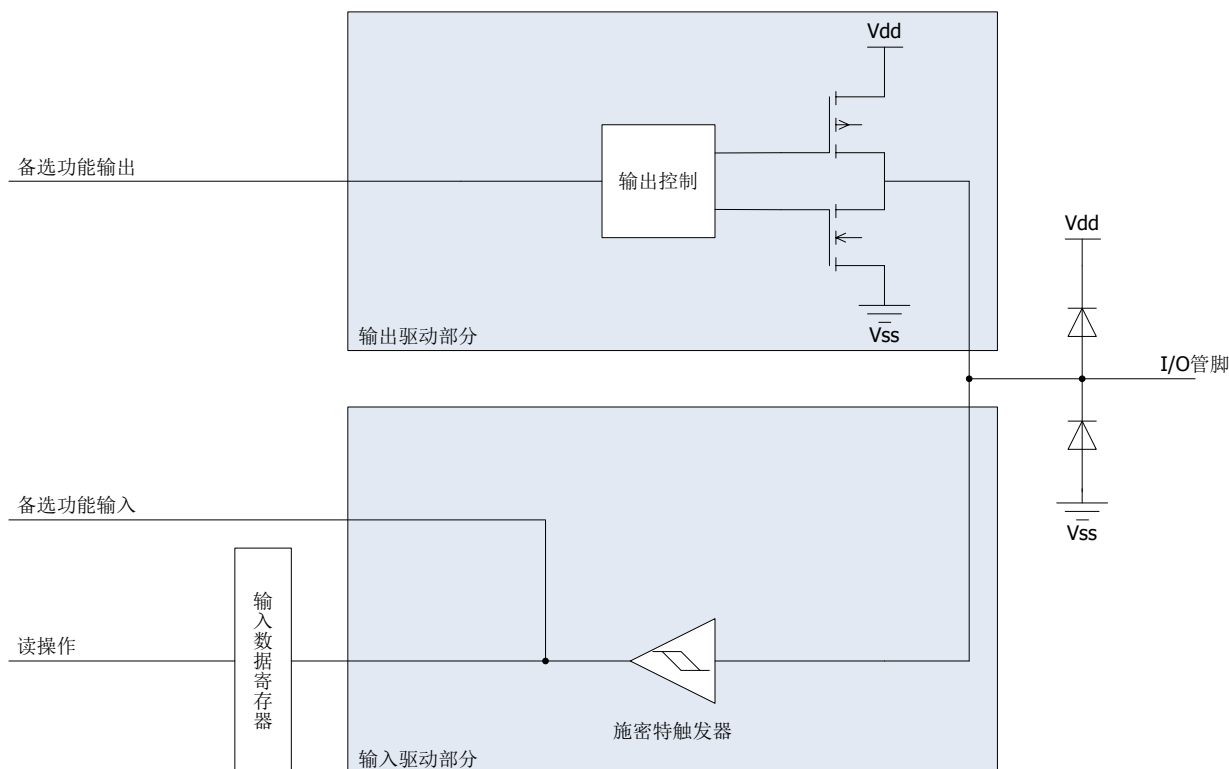
为了适合多种不同的器件封装，GPIO 支持由软件将一些备选功能映射到部分其它管脚上。对于双向的备选功能，相应的端口必须配置为备选功能输出模式。在这种情况下，输入驱动部分将配置为浮空输入模式。

当 GPIO 管脚配置为备选功能时：

- 输出缓冲器将开启为开漏或推挽模式
- 输出缓冲器由外设模块驱动
- 输入施密特触发器启用
- 弱上拉以及弱下拉电阻被禁用
- 每个 APB2 时钟周期采样 I/O 管脚上的数据（电平）并送入数据输入寄存器
- 在开漏模式下，对数据输入寄存器进行读操作，将返回 I/O 状态
- 在推挽模式下，对数据输出寄存器进行读操作，将返回上一次写入的值

下图是 I/O 管脚配置为备选功能时的状态：

图 14 备选功能配置



针对器件外设的 GPIO 配置

以下诸表给出了针对器件各种外设的 GPIO 配置。

表 6 高级定时器 TM1

TIM1/TIM8 管脚配置	功能配置	GPIO 配置
TM1_CH[4:1]	输入捕获通道[4:1]	浮空输入
	输出捕获通道[4:1]	备选功能, 推挽模式
TM1_CH[3:1]N	互补输出通道[3:1]	备选功能, 推挽模式
TM1_BKIN	紧急制动输入	浮空输入
TM1_ETR	外部触发定时器输入	浮空输入

表 7 通用定时器 TM2/3/4

TM2/3/4 管脚配置	功能配置	GPIO 配置
TM2/3/4_CH[4:1]	输入捕获通道[4:1]	浮空输入
	输出捕获通道[4:1]	备选功能, 推挽模式
TM2/3/4_ETR	外部触发定时器输入	浮空输入

表 8 USART

USART 管脚配置	功能配置	GPIO 配置
USART[3:1]_TX	全双工	备选功能, 推挽模式
	半双工, 同步模式	备选功能, 推挽模式
USART[3:1]_RX	全双工	浮空输入/上拉输入
	半双工, 同步模式	未使用。可用作 GPIO
USART[3:1]_CK	同步模式	备选功能, 推挽模式
USART[3:1]_RTS	硬件流控	备选功能, 推挽模式
USART[3:1]_CTS	硬件流控	浮空输入/上拉输入

表 9 SPI

SPI 管脚配置	功能配置	GPIO 配置
SPIx_SCK	主机	备选功能, 推挽模式
	从机	浮空输入
SPIx_MOSI	全双工/主机	备选功能, 推挽模式
	全双工/从机	浮空输入/上拉输入
	单工双向数据口线/主机	备选功能, 推挽模式
	单工双向数据口线/从机	未使用。可用作 GPIO
SPIx_MISO	全双工/主机	浮空输入/上拉输入
	全双工/从机 (点对点)	备选功能, 推挽模式
	全双工/从机 (多点广播)	备选功能, 开漏模式
	单工双向数据口线/主机	用作 GPIO。
	单工双向数据口线/从机 (点对点)	备选功能, 推挽模式
	单工双向数据口线/从机 (多点广播)	备选功能, 开漏模式
SPIx_NSS	硬件主机/从机	浮空输入/上拉输入/下拉输入
	硬件主机/NSS 输出使能	备选功能, 推挽模式
	软件	用作 GPIO

表 10 I²C

I ² C 管脚配置	功能配置	GPIO 配置
I2Cx_SCL	I ² C 时钟	备选功能, 开漏模式
I2Cx_SDA	I ² C 数据	备选功能, 开漏模式

表 11 bxCAN

bxCAN 管脚配置	GPIO 配置
CAN_TX (数据发送口线)	备选功能, 推挽模式
CAN_RX (数据接收口线)	浮空输入/上拉输入

表 12 USB

USB 管脚配置	GPIO 配置
USB_DM/USB_DP	当 USB 使能后, 这两根管脚将从内部连接 USB 收发器

表 13 ADC

ADC 管脚配置	GPIO 配置
ADC	模拟输入

表 14 其它 I/O

管脚配置	功能配置	GPIO 配置
BR-RTC 管脚	RTC 输出	硬件强制, 当配置 BKP_CR 以及 BKP_RTCCR 寄存器后
	欠压事件输入	
CK_OUT	时钟输出	备选功能, 推挽模式
EXTI 输入口线	外部中断输入	浮空输入/上拉输入/下拉输入

GPIO 锁定机制

锁定机制可保护 I/O 口的配置。当 LOCK 置位后, 该端口的位将不得修改, 直到下次复位为止。

2 GPIO 寄存器

2.1 GPIO 端口控制寄存器 1 (GPIOx_CR1) (x = A~E)

地址偏移: 0x00

复位值 : 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMF7[1:0]	MODE7[1:0]	CMF6[1:0]	MODE6[1:0]	CMF5[1:0]	MODE5[1:0]	CMF4[1:0]	MODE4[1:0]								

类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMF3[1:0]		MODE3[1:0]		CMF2[1:0]		MODE2[1:0]		CMF1[1:0]		MODE1[1:0]		CMF0[1:0]		MODE0[1:0]	
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:30	CMF7[1:0]	本端口第 7 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
29:28	MODE7[1:0]	本端口第 7 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
27:26	CMF6[1:0]	本端口第 6 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
25:24	MODE6[1:0]	本端口第 6 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
23:22	CMF5[1:0]	本端口第 5 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
21:20	MODE5[1:0]	本端口第 5 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
19:18	CMF4[1:0]	本端口第 4 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
17:16	MODE4[1:0]	本端口第 4 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
15:14	CMF3[1:0]	本端口第 3 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
13:12	MODE3[1:0]	本端口第 3 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
11:10	CMF2[1:0]	本端口第 2 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
9:8	MODE2[1:0]	本端口第 2 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。
7:6	CMF1[1:0]	本端口第 1 脚配置位 由软件配置端口，详见 CMF0[1:0]之说明。
5:4	MODE1[1:0]	本端口第 1 脚模式位 由软件配置端口，详见 MODE0[1:0]之说明。

位/位域	名称	描述
3:2	CMF0[1:0]	本端口第 0 脚配置位 由软件配置端口。 输入模式 (MODE[1:0]=00) 下 00: 模拟模式 01: 浮空输入 10: 带上拉/下拉输入 11: 保留 输出模式 (MODE[1:0]>00) 下 00: 推挽模式的 GPIO 输出 01: 开漏模式的 GPIO 输出 10: 推挽模式的 AFIO 输出 11: 开漏模式的 AFIO 输出
1:0	MODE0[1:0]	本端口第 0 脚模式位 由软件配置端口。 00: 输入模式 (复位后为此状态) 01: 输出模式, 最大支持 10MHz 10: 输出模式, 最大支持 2MHz 11: 输出模式, 最大支持 50MHz

2.2 GPIO 端口控制寄存器 2 (GPIOx_CR2) (x = A~E)

地址偏移: 0x04

复位值 : 0x4444 4444

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMF15[1:0]		MODE15[1:0]		CMF14[1:0]		MODE14[1:0]		CMF13[1:0]		MODE13[1:0]		CMF12[1:0]		MODE12[1:0]	
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMF11[1:0]		MODE11[1:0]		CMF10[1:0]		MODE10[1:0]		CMF9[1:0]		MODE9[1:0]		CMF8[1:0]		MODE8[1:0]	
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

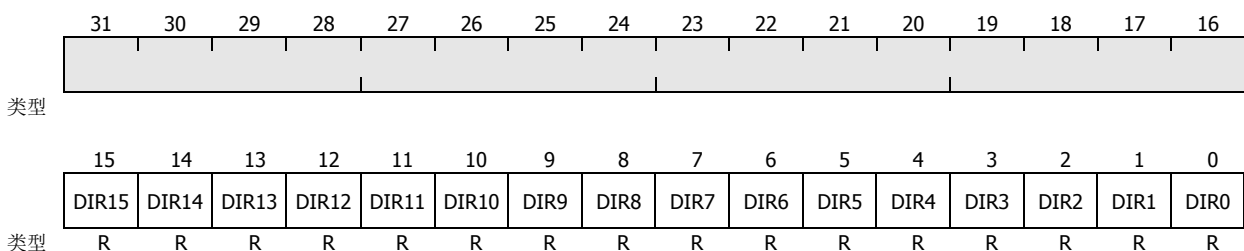
位/位域	名称	描述
31:30	CMF15[1:0]	本端口第 15 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
29:28	MODE15[1:0]	本端口第 15 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
27:26	CMF14[1:0]	本端口第 14 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
25:24	MODE14[1:0]	本端口第 14 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
23:22	CMF13[1:0]	本端口第 13 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。

位/位域	名称	描述
21:20	MODE13[1:0]	本端口第 13 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
19:18	CMF12[1:0]	本端口第 12 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
17:16	MODE12[1:0]	本端口第 12 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
15:14	CMF11[1:0]	本端口第 11 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
13:12	MODE11[1:0]	本端口第 11 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
11:10	CMF10[1:0]	本端口第 10 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
9:8	MODE10[1:0]	本端口第 10 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
7:6	CMF9[1:0]	本端口第 9 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
5:4	MODE9[1:0]	本端口第 9 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。
3:2	CMF8[1:0]	本端口第 8 脚配置位 由软件配置端口, 详见 CMF0[1:0]之说明。
1:0	MODE8[1:0]	本端口第 8 脚模式位 由软件配置端口, 详见 MODE0[1:0]之说明。

2.3 GPIO 端口数据输入寄存器 (GPIOx_DIR) (x = A~E)

地址偏移: 0x08

复位值 : 0x0000 XXXX

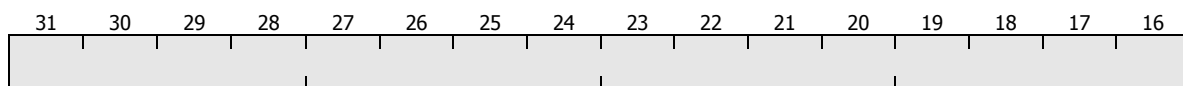


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	DIR[15:0]	本端口各管脚输入数据 这些位由软件置位和清零。 0: 输入信号为低电平 1: 输入信号为高电平

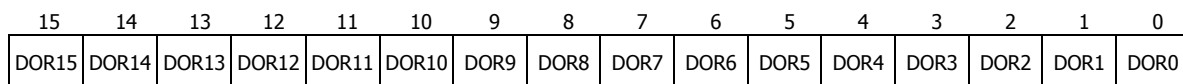
2.4 GPIO 端口数据输出寄存器 (GPIOx_DOR) (x = A~E)

地址偏移: 0x0C

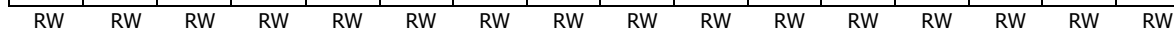
复位值 : 0x0000 0000



类型



类型

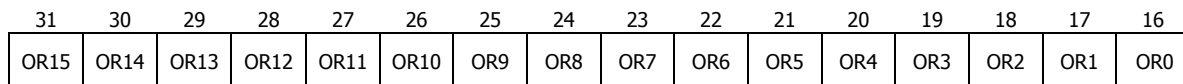


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	DOR[15:0]	本端口各管脚输出数据 这些位由软件置位和清零。 0: 管脚输出低电平 1: 管脚输出高电平

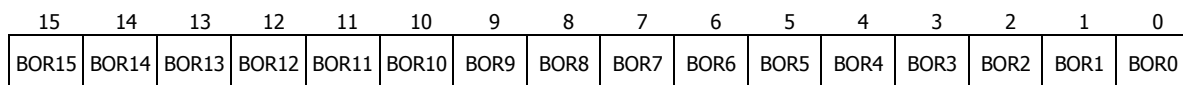
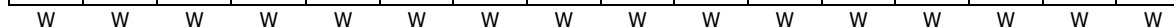
2.5 GPIO 端口按位操作寄存器 (GPIOx_BOR) (x = A~E)

地址偏移: 0x10

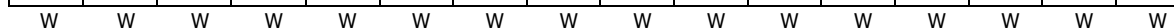
复位值 : 0x0000 0000



类型



类型

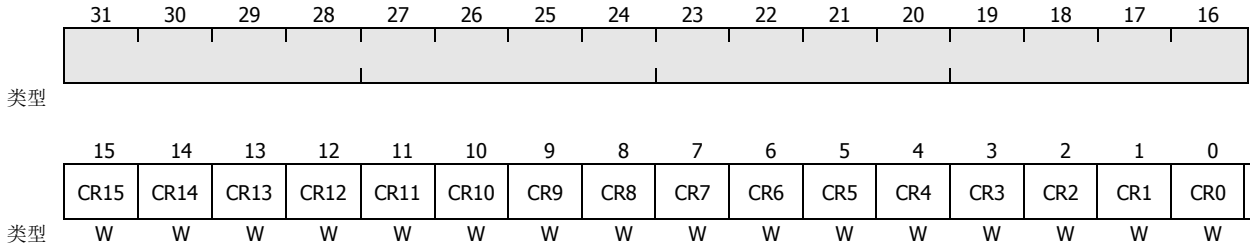


位/位域	名称	描述
31:16	OR[15:0]	本端口各管脚清零位 这些位由软件置位和清零。 0: 相应的 ODRx 位无变化 1: 将相应的 ODRx 位清零
15:0	BOR[15:0]	本端口各管脚置位位 这些位由软件置位和清零。 0: 相应的 ODRx 位无变化 1: 将相应的 ODRx 位置位

2.6 GPIO 端口按位清零寄存器 (GPIOx_BCR) (x = A~E)

地址偏移: 0x14

复位值 : 0x0000 0000

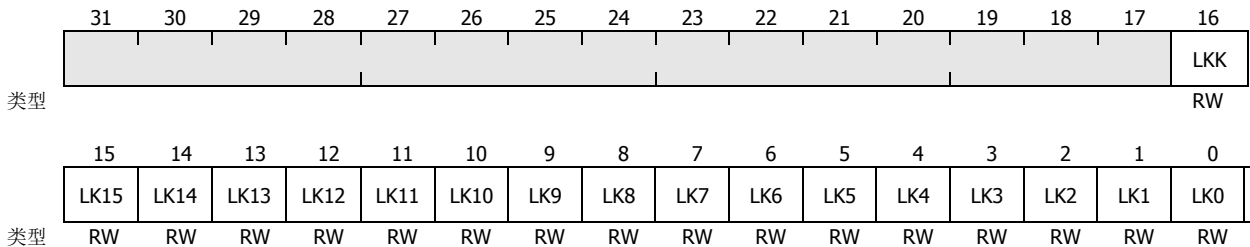


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	CR[15:0]	本端口各管脚清零 这些位由软件置位和清零。 0: 相应的 ODRx 位无变化 1: 将相应的 ODRx 位清零

2.7 GPIO 端口配置锁定寄存器 (GPIOx_LOCKR) (x = A~E)

地址偏移: 0x18

复位值 : 0x0000 0000



位/位域	名称	描述
31:17	保留	请保持其复位状态
16	LKK	锁定密钥位 本位始终可读, 但是只能通过锁定密钥写入序列才能置位。 0: 端口配置锁定密钥未激活 1: 端口配置锁定密钥已激活 锁定密钥写入序列 写入 1 → 写入 0 → 写入 1 → 读出为 0 → 读出为 1 请注意: 在执行锁定密钥写入序列期间, 不得变更 LCK[15:0]的内容。
15:0	LK[15:0]	本端口各管脚锁定位 这些位由软件置位和清零。 0: 端口配置未锁定 1: 端口配置已锁定

3 AFIO

3.1 简介

为了扩充 GPIO 的灵活使用，或便于实现外设功能，每个 I/O 管脚都可以通过 AFIO 端口配置寄存器 (AFIO_PCFR1) 配置为最多三种不同的功能。通过外设 IO 重映射功能，可以为应用设计出最恰当的管脚分布。此外，各个 GPIO 管脚可通过设置相关的 EXTI 来源选择寄存器 (AFIO_ESSR) 配置为 EXTI 中断口线，以触发中断或者事件。

3.2 主要特点

- 访问寄存器时作为 APB 从接口
- EXTI 来源选择
- 每个管脚具有最多四种可配置的备选功能

3.3 功能描述

JTAG/SWD 备选功能重映射

调试接口信号在 GPIO 端口上的映射如下表所示。

表 15 调试接口信号

备选功能	GPIO 管脚
JTMS/SWDIO	PA13
JTCK/SWCLK	PA14
JTDI	PA15
JTDO/TRACESWO	PB3
NJTRST	PB4
TRACECK	PE2
TRACECK0	PE3
TRACECK1	PE4
TRACECK2	PE5
TRACECK3	PE6

为了减少调试接口所占用的 GPIO，用户可以将 AFIO_PCFR1 寄存器的 SWJ_CFG[2:0]位域配置为其它数值，参见下表。

表 16 调试接口映射

SW_CFG [2:0]	可用的调试接口	需要用到的 SWJ I/O 管脚				
		PA13/ JTMS/ SWDIO	PA14/ JTCK/ SWCLK	PA15/ JTDI	PB3/ JTDO/ TRACE SWO	PB4/ NJTRST
000	完整 SWJ (JTAG-DP + SW-DP) (复位后的状态)	X	X	X	X	X
001	完整 SWJ (JTAG-DP + SW-DP) 但不包含 NJTRST	X	X	X	X	
010	禁用 JTAG-DP, 使能 SW-DP	X	X			
100	禁用 JTAG-DP, 禁用 SW-DP					
其它值	禁止这样设置					

ADC 备选功能重映射

参见 AFIO 端口配置寄存器 (AFIO_PCFR1)。

表 17 ADC1 外部触发插入式转换备选功能重映射

备选功能	ADC1_ETRGINJ_REMAP=0	ADC1_ETRGINJ_REMAP=1
ADC1 外部触发插入式转换	ADC1 外部触发插入式转换连接到 EXTI15	ADC1 外部触发插入式转换连接到 TIM8_CH4

表 18 ADC1 外部触发标准转换备选功能重映射

备选功能	ADC1_ETRGREG_REMAP=0	ADC1_ETRGREG_REMAP=1
ADC1 外部触发标准转换	ADC1 外部触发标准转换连接到 EXTI11	ADC1 外部触发标准转换连接到 TIM8_TRGO

表 19 ADC2 外部触发插入式转换备选功能重映射

备选功能	ADC2_ETRGINJ_REMAP=0	ADC2_ETRGINJ_REMAP=1
ADC2 外部触发插入式转换	ADC2 外部触发插入式转换连接到 EXTI15	ADC2 外部触发插入式转换连接到 TIM8_CH4

表 20 ADC2 外部触发标准转换备选功能重映射

备选功能	ADC2_ETRGREG_REMAP=0	ADC2_ETRGREG_REMAP=1
ADC2 外部触发标准转换	ADC2 外部触发标准转换连接到 EXTI11	ADC2 外部触发标准转换连接到 TIM8_TRGO

定时器备选功能重映射

GD32F103xx 系列微控制器包含一个高级控制定时器（TM1）以及 3 个通用定时器（TM2~4）。参见 AFIO 端口配置寄存器（AFIO_PCFR1）。

表 21 TM1 备选功能重映射

备选功能	TIM1_REMAP[1:0] =00(无重映射)	TIM1_REMAP[1:0] =10(部分重映射)	TIM1_REMAP[1:0] =11(完全重映射)
TM1_ETR	PA12		PE7
TM1_CH1	PA8		PE9
TM1_CH2	PA9		PE11
TM1_CH3	PA10		PE13
TM1_CH4	PA11		PE14
TM1_BKIN	PB12	PA6	PE15
TM1_CH1N	PB13	PA7	PE8
TM1_CH2N	PB14	PB0	PE10
TM1_CH3N	PB15	PB1	PE12

表 22 TM2 备选功能重映射

备选功能	TIM2_REMAP [1:0]=00 (无重映射)	TIM2_REMAP [1:0]=01 (部分重映射)	TIM2_REMAP [1:0]=10 (部分重映射)	TIM1_REMAP [1:0]=11 (完全重映射)
TM2_CH1/TM2_ETR	PA0	PA15	PA0	PA15
TM2_CH2	PA1	PB3	PA1	PB3
TM2_CH3	PA2		PB10	
TM2_CH4	PA3		PB11	

表 23 TM3 备选功能重映射

备选功能	TIM3_REMAP[1:0] =00(无重映射)	TIM3_REMAP[1:0] =10(部分重映射)	TIM3_REMAP[1:0] =11(完全重映射)
TM3_CH1	PA6	PB4	PC6
TM3_CH2	PA7	PB5	PC7
TM3_CH3	PB0		PC8
TM3_CH4	PB1		PC9

表 24 TM4 备选功能重映射

备选功能	TIM4_REMAP=0	TIM4_REMAP=1
TM4_CH1	PB6	PD12
TM4_CH2	PB7	PD13
TM4_CH3	PB8	PD14
TM4_CH4	PB9	PD15

USART 备选功能重映射

参见 AFIO 端口配置寄存器 (AFIO_PCFR1)。

表 25 USART1 备选功能重映射

备选功能	USART1_REMAP=0	USART1_REMAP=1
USART1_TX	PA9	PB6
USART1_RX	PA10	PB7

表 26 USART2 备选功能重映射

备选功能	USART2_REMAP=0	USART2_REMAP=1
USART2_CTS	PA0	PD3
USART2_RTS	PA1	PD4
USART2_TX	PA2	PD5
USART2_RX	PA3	PD6
USART2_CK	PA4	PD7

表 27 USART3 备选功能重映射

备选功能	USART3_REMAP [1:0]=00 (无重映射)	USART3_REMAP [1:0]=10 (部分重映射)	USART3_REMAP [1:0]=11 (完全重映射)
USART3_TX	PB10	PC10	PD8
USART3_RX	PB11	PC11	PD9
USART3_CK	PB12	PC12	PD10
USART3_CTS	PB13		PD11
USART3_RTS	PB14		PD12

I²C1 备选功能重映射

参见 AFIO 端口配置寄存器 (AFIO_PCFR1)。

表 28 I²C1 备选功能重映射

备选功能	I2C1_REMAP=0	I2C1_REMAP=1
I2C1_SCL	PB6	PB8
I2C1_SDA	PB7	PB9

SPI1 备选功能重映射

参见 AFIO 端口配置寄存器 (AFIO_PCFR1)。

表 29 SPI1 备选功能重映射

备选功能	SPI1_REMAP=0	SPI1_REMAP=1
SPI1_NSS	PA4	PA15
SPI1_SCK	PA5	PB3
SPI1_MISO	PA6	PB4
SPI1_MOSI	PA7	PB5

CAN 备选功能重映射

CAN 信号可以分别映射到端口 A、端口 B 或者端口 D，如下表所示。不过，映射到端口 D 对 36 脚、48 脚以及 64 脚封装无实际意义（并未引出相应管脚）。

表 30 CAN 备选功能重映射

备选功能	CAN_REMAP [1:0]=00	CAN_REMAP [1:0]=10 ¹	CAN_REMAP [1:0]=11 ²
CAN_RX	PA11	PB8	PD0
CAN_TX	PA12	PB9	PD1

注 1：这种重映射在 36 脚封装上不可用。

注 2：这种重映射仅对 100 脚封装可用，并且前提是 PD0 和 PD1 并未映射到 OSCIN 以及 OSCOUT。

CLK 管脚备选功能重映射

当 LSE 振荡电路关闭时，LSE 振荡电路管脚 OSC32_IN 以及 OSC32_OUT 分别可以作为通用 I/O 口 PC14、PC15 使用。对这两个管脚而言，LSE 功能要优于 GPIO 功能。但是当 1.2V 电源域停止供电（进入待机模式）、或当备电域由 V_{BAT} 供电（V_{DD} 停止供电）时，PC14/PC15 的 GPIO 功能将失效，依然工作于模拟模式。

表 31 OSC32 管脚配置

备选功能	LSE = ON	LSE = OFF
PC14	OSC32_IN	PC14
PC15	OSC32_OUT	PC15

HSE 振荡电路管脚 OSC_IN 以及 OSC_OUT 分别可以当做通用 I/O 口 PD0、PD1 使用。

表 32 OSC 管脚配置

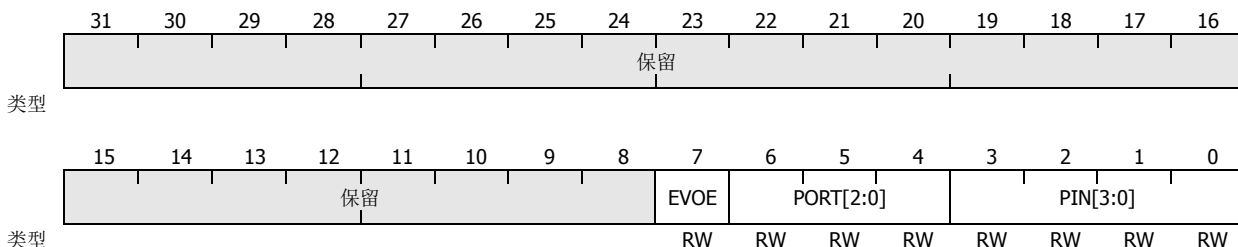
备选功能	HSE = ON	HSE = OFF
PD0	OSC_IN	PD0
PD1	OSC_OUT	PD1

4 AFIO 寄存器

4.1 事件控制寄存器 (AFIO_ECR)

地址偏移: 0x00

复位值 : 0x0000 0000



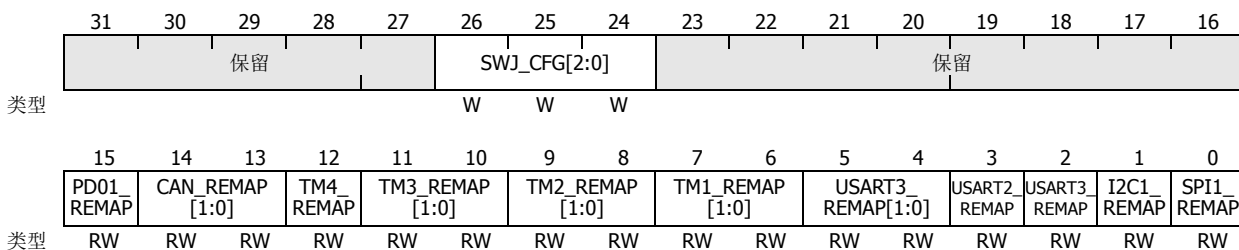
位/位域	名称	描述
31:8	保留	请保持其复位状态
7	EVOE	事件输出使能 本位可由软件置位和清零。 0: 禁用事件输出 1: 使能事件输出
6:4	PORT[2:0]	事件输出端口选择 本位域可由软件置位和清零。 000: 端口 A 001: 端口 B 010: 端口 C 011: 端口 D 100: 端口 E

位/位域	名称	描述
3:0	PIN[3:0]	事件输出管脚选择 这些位由软件置位和清零。 0000: 第 0 管脚 0001: 第 1 管脚 0010: 第 2 管脚 1111: 第 15 管脚

4.2 AFIO 端口配置寄存器 1 (AFIO_PCFR1)

地址偏移: 0x04

复位值 : 0x0000 0000



位/位域	名称	描述
31:27	保留	请保持其复位状态
26:24	SWJ_CFG[2:0]	串行线 JTAG 配置 本位域可由软件置位和清零。 000: 完整 SWJ (JTAG-DP+SW-DP)。复位后即为此状态 001: 完整 SWJ (JTAG-DP+SW-DP), 但不包含 NJTRST 010: 禁用 JTAG-DP, 使能 SW-DP 100: 禁用 JTAG-DP, 禁用 SW-DP 其它: 未定义
23:16	保留	请保持其复位状态
15	PD01_REMAP	PD0/PD1 对 OSC_IN/OSC_OUT 的重映射 本位可由软件置位和清零。 0: 无重映射 1: PD0 重映射到 OSC_IN 上, PD1 重映射到 OSC_OUT 上
14:13	CAN_REMAP[1:0]	CAN 接口的重映射 本位域可由软件置位和清零。 00: CAN_RX 重映射到 PA11 上, CAN_TX 重映射到 PA12 上 01: 未定义 10: CAN_RX 重映射到 PB8 上, CAN_TX 重映射到 PB9 上 11: CAN_RX 重映射到 PD0 上, CAN_TX 重映射到 PD1 上
12	TM4_REMAP	TM4 重映射 本位可由软件置位和清零。 0: 无重映射 1: 重映射

位/位域	名称	描述
11:10	TM3_REMAP[1:0]	TM3 重映射 本位域可由软件置位和清零。 00: 无重映射 01: 未定义 10: 部分重映射 11: 完全重映射
9:8	TM2_REMAP[1:0]	TM2 重映射 本位域可由软件置位和清零。 00: 无重映射 01: 未定义 10: 部分重映射 11: 完全重映射
7:6	TM1_REMAP[1:0]	TM1 重映射 本位域可由软件置位和清零。 00: 无重映射 01: 未定义 10: 部分重映射 11: 完全重映射
5:4	USART3_REMAP[1:0]	USART3 重映射 本位域可由软件置位和清零。 00: 无重映射 01: 未定义 10: 部分重映射 11: 完全重映射
3	USART2_REMAP	USART2 重映射 本位可由软件置位和清零。 0: 无重映射 1: 重映射
2	USART1_REMAP	USART1 重映射 本位可由软件置位和清零。 0: 无重映射 1: 重映射
1	I2C1_REMAP	I ² C1 重映射 本位可由软件置位和清零。 0: 无重映射 1: 重映射
0	SPI1_REMAP	SPI1 重映射 本位可由软件置位和清零。 0: 无重映射 1: 重映射

4.3 EXTI 来源选择寄存器 1 (AFIO_ESSR1)

地址偏移: 0x08

复位值 : 0x0000 0000



类型

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域	名称	描述
31:16	保留	请保持其复位状态
15:12	EXTI3[3:0]	EXTI3 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
11:8	EXTI2[3:0]	EXTI2 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
7:4	EXTI1[3:0]	EXTI1 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
3:0	EXTI0[3:0]	EXTI0 管脚配置 本位域可由软件置位和清零。 0000: 选中 PA[0]脚作为 EXTI0 的中断源信号 0001: 选中 PB[0]脚作为 EXTI0 的中断源信号 0010: 选中 PC[0]脚作为 EXTI0 的中断源信号 0011: 选中 PD[0]脚作为 EXTI0 的中断源信号 0100: 选中 PE[0]脚作为 EXTI0 的中断源信号

4.4 EXTI 来源选择寄存器 2 (AFIO_ESSR2)

地址偏移: 0x0C

复位值 : 0x0000 0000

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

类型

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

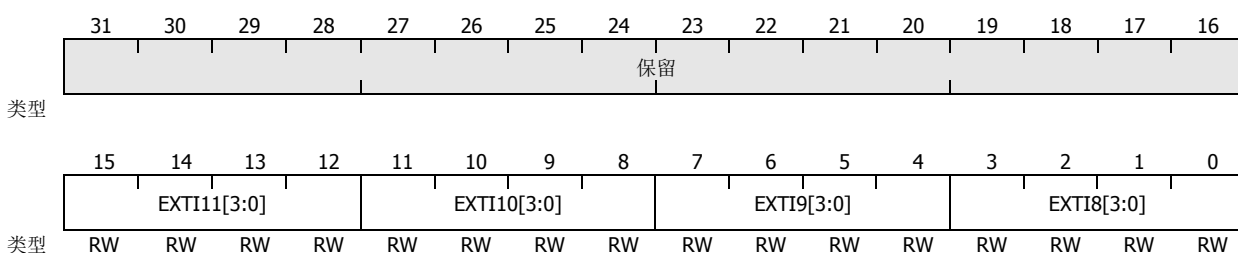
位/位域	名称	描述
31:16	保留	请保持其复位状态
15:12	EXTI7[3:0]	EXTI7 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。

位/位域	名称	描述
11:8	EXTI6[3:0]	EXTI6 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
7:4	EXTI5[3:0]	EXTI5 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
3:0	EXTI4[3:0]	EXTI4 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。

4.5 EXTI 来源选择寄存器 3 (AFIO_ESSR3)

地址偏移: 0x10

复位值 : 0x0000 0000

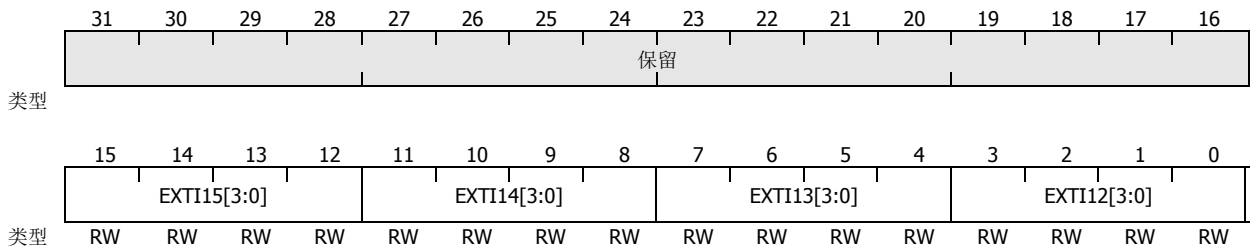


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:12	EXTI11[3:0]	EXTI11 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
11:8	EXTI10[3:0]	EXTI10 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
7:4	EXTI9[3:0]	EXTI9 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。
3:0	EXTI8[3:0]	EXTI8 管脚配置 本位域可由软件置位和清零。 参见 EXTI0[3:0]的描述。

4.6 EXTI 来源选择寄存器 4 (AFIO_ESSR4)

地址偏移: 0x14

复位值 : 0x0000 0000



位/位域	名称	描述
31:16	保留	请保持其复位状态
15:12	EXTII15[3:0]	EXTII15 管脚配置 本位域可由软件置位和清零。 参见 EXTIO[3:0]的描述。
11:8	EXTII14[3:0]	EXTII14 管脚配置 本位域可由软件置位和清零。 参见 EXTIO[3:0]的描述。
7:4	EXTII13[3:0]	EXTII13 管脚配置 本位域可由软件置位和清零。 参见 EXTIO[3:0]的描述。
3:0	EXTII12[3:0]	EXTII12 管脚配置 本位域可由软件置位和清零。 参见 EXTIO[3:0]的描述。

第5章 中断及事件

1 简介

Cortex™-M3 集成了嵌套式矢量型中断控制器（Nested Vectored Interrupt Controller，简称为 NVIC），实现高效的异常/中断处理。NVIC 机构实现了超低延迟的异常/中断处理，并辅助控制电源管理。它与内核是紧密耦合的。用户可以阅读《Cortex™-M3 技术参考手册》以获取关于 NVIC 的更详细说明。

GD32F103xx 系列微控制器还提供了一个外部的中断/事件控制器，称为 EXTI，其中包含 19 个相互独立的边沿检测电路，可向处理器内核产生中断请求或唤醒事件。EXTI 有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI 中的每个边沿检测电路都能分别进行配置和屏蔽。

2 主要特点

- Cortex™-M3 系统异常
- 43 种可屏蔽的外设中断
- 可设置 4 位优先级——总共提供 16 种优先级
- 高效的中断处理
- 支持异常抢占以及咬尾中断
- 用于低功耗模式的唤醒系统
- 在 EXTI 中 19 个相互独立的边沿检测电路
- 3 种触发类型：上升沿触发、下降沿触发和任意沿触发
- 软件中断或事件触发
- 可配置的触发来源

3 功能描述

3.1 NVIC 以及异常/中断的处理

ARM® Cortex™-M3 处理器和嵌套式向量型中断控制器（NVIC）在处理（Handler）模式下对所有异常进行优先级区分以及处理。当产生异常时，系统自动将处理器当前工作态压栈，在执行完中断服务子程序（ISR）后自动将其出栈。

取中断向量是与处理器工作态压栈并行执行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现中断的背靠背执行，大大削减了工作态反复切换所带来的开销。下表列出了所有的异常类型。对于这些异常中的 7 种（系统处理）以及 43 种中断，可通过软件设置 16 个优先级（如下表

所示)。

表 33 Cortex™-M3 中的 NVIC 异常类型

异常类型	向量编号	优先级	向量地址	描述
-	0	-	0x0000 0000	保留
复位	1	-3	0x0000 0004	复位
NMI	2	-2	0x0000 0008	不可屏蔽中断
硬件故障	3	-1	0x0000 000C	各种硬件级别的故障
存储器管理	4	可编程设置	0x0000 0010	存储器管理
总线故障	5	可编程设置	0x0000 0014	预取指故障，存储器访问故障
用法故障	6	可编程设置	0x0000 0018	未定义的指令，或非法的处理器工作态
-	7~10	-	0x0000 001C~ 0x0000 0028	保留
服务调用	11	可编程设置	0x0000 002C	通过 SWI 指令实现的系统服务调用
调试监控	12	可编程设置	0x0000 0030	调试监控器
-	13	-	0x0000 0034	保留
挂起服务	14	可编程设置	0x0000 0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000 003C	系统节拍定时器
中断	16~58	可编程设置	0x0000 0040~ 0x0000 00E8	外设中断（详见下表）

表 34 GD32F103xx 的中断向量表

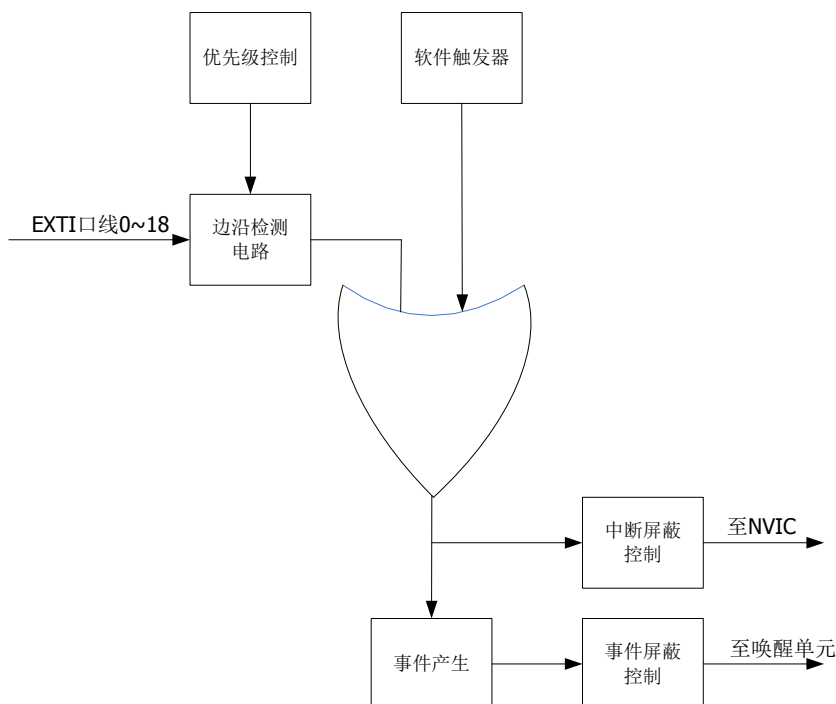
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口式看门狗中断	0x0000 0040
IRQ 1	17	通过 EXTI 口线检测的 PVD 中断	0x0000 0044
IRQ 2	18	侵入中断	0x0000 0048
IRQ 3	19	RTC 全局中断	0x0000 004C
IRQ 4	20	Flash 全局中断	0x0000 0050
IRQ 5	21	RCC 全局中断	0x0000 0054
IRQ 6	22	EXTI 口线 0 中断	0x0000 0058
IRQ 7	23	EXTI 口线 1 中断	0x0000 005C
IRQ 8	24	EXTI 口线 2 中断	0x0000 0060
IRQ 9	25	EXTI 口线 3 中断	0x0000 0064
IRQ 10	26	EXTI 口线 4 中断	0x0000 0068
IRQ 11	27	DMA1 通道 1 全局中断	0x0000 006C
IRQ 12	28	DMA1 通道 2 全局中断	0x0000 0070

中断编号	向量编号	外设中断描述	向量地址
IRQ 13	29	DMA1 通道 3 全局中断	0x0000 0074
IRQ 14	30	DMA1 通道 4 全局中断	0x0000 0078
IRQ 15	31	DMA1 通道 5 全局中断	0x0000 007C
IRQ 16	32	DMA1 通道 6 全局中断	0x0000 0080
IRQ 17	33	DMA1 通道 7 全局中断	0x0000 0084
IRQ 18	34	ADC1 及 ADC2 全局中断	0x0000 0088
IRQ 19	35	USB 高优先级中断, 或 CAN TX 中断	0x0000 008C
IRQ 20	36	USB 低优先级中断, 或 CAN RX 中断	0x0000 0090
IRQ 21	37	CAN RX1 中断	0x0000 0094
IRQ 22	38	CAN SCE 中断	0x0000 0098
IRQ 23	39	EXTI 口线[9:5]中断	0x0000 009C
IRQ 24	40	TIM1 紧急停止中断	0x0000 00A0
IRQ 25	41	TIM1 更新中断	0x0000 00A4
IRQ 26	42	TIM1 触发及通信中断	0x0000 00A8
IRQ 27	43	TIM1 捕获比较中断	0x0000 00AC
IRQ 28	44	TIM2 全局中断	0x0000 00B0
IRQ 29	45	TIM3 全局中断	0x0000 00B4
IRQ 30	46	TIM4 全局中断	0x0000 00B8
IRQ 31	47	I ² C1 事件中断	0x0000 00BC
IRQ 32	48	I ² C1 错误中断	0x0000 00C0
IRQ 33	49	I ² C2 事件中断	0x0000 00C4
IRQ 34	50	I ² C2 错误中断	0x0000 00C8
IRQ 35	51	SPI1 全局中断	0x0000 00CC
IRQ 36	52	SPI2 全局中断	0x0000 00D0
IRQ 37	53	USART1 全局中断	0x0000 00D4
IRQ 38	54	USART2 全局中断	0x0000 00D8
IRQ 39	55	USART3 全局中断	0x0000 00DC
IRQ 40	56	EXTI 口线[15:10]中断	0x0000 00E0
IRQ 41	57	通过 EXTI 口线产生的 RTC 闹钟中断	0x0000 00E4
IRQ 42	58	通过 EXTI 口线产生的 USB 从挂起唤醒	0x0000 00E8

3.2 外部中断及事件 (EXTI)

EXTI 包含 19 个相互独立的边沿检测电路, 并且可以向处理器产生中断请求或唤醒事件。EXTI 有 3 种触发类型: 上升沿触发、下降沿触发和任意沿触发。EXTI 中的每个边沿检测电路都可以分别予以配置或屏蔽。下图是 EXTI 的内部框图。

图 15 EXTI 框图



EXTI 触发源包括来自 GPIO 管脚的 16 根口线以及来自内部模块（包括 PVD、RTC 闹钟以及 USB 唤醒）的 3 根口线。通过配置 GPFIO 模块内的 AFIO_EXTICRx 寄存器，所有 GPIO 管脚都可能选作 EXTI 的触发源，参见 GPFIO 一章。

除了中断，EXTI 还可以向处理器提供事件信号。Cortex™-M3 处理器完全支持 WFI（等待中断）指令、WFE（等待事件）指令、SEV（发送事件）指令。GD32F103xx 系列微控制器提供一个唤醒中断控制器（WIC），于是用户可以放心地让处理器与 NVIC 进入功耗极低的休眠模式，由 WIC 来识别中断/事件并区分其优先级。当某些预期的事件发生时（例如特定的 GPIO 管脚电平翻转、或者 RTC 闹钟动作），EXTI 可唤醒处理器和整个系统。

4 中断及 EXTI 寄存器

4.1 中断屏蔽寄存器（EXTI_IMR）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													MR18	MR17	MR16
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:19	保留	请保持其复位状态
18:0	MRx	中断屏蔽控制 0: 第 x 口线的中断被屏蔽 1: 第 x 口线的中断未被屏蔽

4.2 事件屏蔽寄存器 (EXTI_EMR)

地址偏移: 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													MR18	MR17	MR16
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:19	保留	请保持其复位状态
18:0	MRx	事件屏蔽控制 0: 第 x 口线的事件被屏蔽 1: 第 x 口线的事件未被屏蔽

4.3 上升沿选择寄存器 (EXTI_RTSTR)

地址偏移: 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TR18	TR17	TR16
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:19	保留	请保持其复位状态

位/位域	名称	描述
18:0	TRx	触发边沿控制 0: 第 x 口线的上升沿无特殊作用 1: 第 x 口线的上升沿将视为中断/事件的请求

4.4 下降沿选择寄存器 (EXTI_FTSTR)

地址偏移: 0x0C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TR18	TR17	TR16
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:19	保留	请保持其复位状态
18:0	TRx	触发跳变沿控制 0: 第 x 口线的下降沿无特殊作用 1: 第 x 口线的下降沿将视为中断/事件的请求

4.5 软件中断事件寄存器 (EXTI_SWIER)

地址偏移: 0x10

复位值 : 0x0000 0000

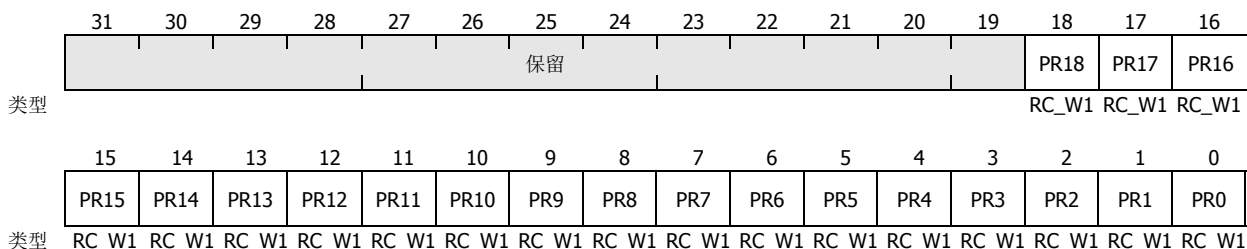
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													SWIER 18	SWIER 17	SWIER 16
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0
RW	RW	RW	RW	RW	RW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

位/位域	名称	描述
31:19	保留	请保持其复位状态
18:0	SWIERx	中断/事件软件触发 0: 禁用 EXTIx 的软件中断/事件请求 1: 激活 EXTIx 的软件中断/事件请求

4.6 挂起寄存器 (EXTI_PR)

地址偏移: 0x14

复位值 : 未定义



位/位域	名称	描述
31:19	保留	请保持其复位状态
18:0	PRx	中断/事件状态 对这些位写 1, 即可将其清零。 0: EXTI 第 x 口线未被触发 1: EXTI 第 x 口线已触发

第6章 DMA 控制器

1 简介

直接存储器访问（Direct Memory Access，简称为 DMA）模块实现了一种无需 CPU 参与、完全依靠硬件的在外设及/或存储器之间传递数据的工作方式，从而极大地解放了 CPU，用户可以将 CPU 用在其它系统功能上。通过 DMA，系统可以在外设与存储器之间、在存储器与存储器之间快速传输数据，无需 CPU 的任何干涉。

DMA 控制器总共有 7 个通道，每个通道专用于管理来自一个或多个外设的存储器访问请求。当同时出现多个 DMA 请求时，DMA 控制器凭仲裁机制来判断和区分优先级。

2 主要特点

- 数据块长度可编程设置，最大 65536
- 7 个通道，每个通道都可以编程配置
- AHB 及 APB 外设、Flash、SRAM，都可以作为 DMA 操作的来源或目的
- 每个通道连接到固定的硬件 DMA 请求
- DMA 通道请求的优先级由软件配置以及硬件通道号共同决定
- 支持 4 种类型的访问方式：外设到外设、外设到存储器、存储器到外设、存储器到存储器
- 每个通道都具有三种类型的事件标志和一个中断请求
- 在单个通道内，来源和目的的传输单元尺寸可软件配置

3 功能描述

DMA 与 Cortex™-M3 内核共享系统总线，可直接进行存储器访问。如果 CPU 和 DMA 都准备访问同一块目标地址（不管是存储器还是外设）时，DMA 请求可能在一定总线周期内遏止 CPU 对系统总线的访问。不过，总线矩阵是采用轮盘式调度的，因此可确保系统总线（无论是到存储器还是到外设）至少有一半带宽会保留给 CPU。

3.1 优先级以及仲裁机制

当同一时刻出现多个 DMA 请求时，其执行顺序将由 DMA 仲裁机构决定。仲裁依据则既有软件因素又有硬件因素：

- 首要考虑因素是软件优先级。DMA 的软件优先级分为四级：
 1. 最高优先级
 2. 高优先级

3. 低优先级
4. 最低优先级

- 次要考虑因素是硬件优先级。硬件优先级的规律是：通道号越小，则优先级越高。当然，按硬件因素判定的前提是两个 DMA 通道的软件优先级相同。

3.2 递增模式

DMA_CCRx 寄存器的 PSIZE 位域和 MSIZE 位域分别用于配置外设和存储器传输数据块的大小。

DMA_CCRx 寄存器的 PINC 位和 MINC 位用于配置递增工作模式。

若使能此工作模式，则下次传输地址将自动加 1、加 2 或者加 4（取决于传输数据单元大小）。

3.3 循环模式

DMA_CCRx 寄存器的 CIRC 位用于使能循环模式。若使能此工作模式，待传输的数据个数将自动重载为最初写入的配置值。

此工作模式适用于处理连续不断的数据流，例如 ADC 转换结果等等。请注意：若使能 MEM2MEM 模式，则循环模式无效。

3.4 存储器到存储器模式

DMA 允许在两个存储器块之间传输数据，这种模式称为存储器到存储器模式（MEM2MEM）。要使用这种模式，用户必须将 DMA_CCRx 寄存器的 MEM2MEM 置位。在这种工作模式下，只要使能某个通道，数据就会自动从来源地址传递到目的地址，直到 DMA_CNDTRx 寄存器递减到 0 为止。请注意，这种模式下循环模式将无效（否则会构成死循环）。

3.5 中断请求

在 DMA_CCRx 寄存器中可以配置三种类型事件的标志，可产生相应的中断：

- TEIF：传输错误
- TCIF：传输完成
- HTIF：半传输完成

同样，在 DMA_CCRx 寄存器中，可以配置三种对应的中断使能位：

- TEIE：传输错误中断使能
- TCIE：传输完成中断使能
- HTIE：半传输完成中断使能

假如在 DMA 通道运行过程中产生了错误，那么该通道将会立即关闭，并且将 TEIF 置位；如果 TEIE=1，那么就会产生一个中断。

3.6 DMA 通道配置流程

配置 DMA 通道应按照下列顺序进行：

1. 配置 DMA_CPARx 寄存器，设定外设地址。
2. 配置 DMA_CMARx 寄存器，设定存储器地址。
3. 配置 DMA_CNDTRx 寄存器，设定传输数据单元总数。
4. 配置 DMA_CCRx 寄存器，设定通道软件优先级、传输方向、工作模式、数据大小以及中断类型。
5. 配置 DMA_CCRx 寄存器，启动通道。

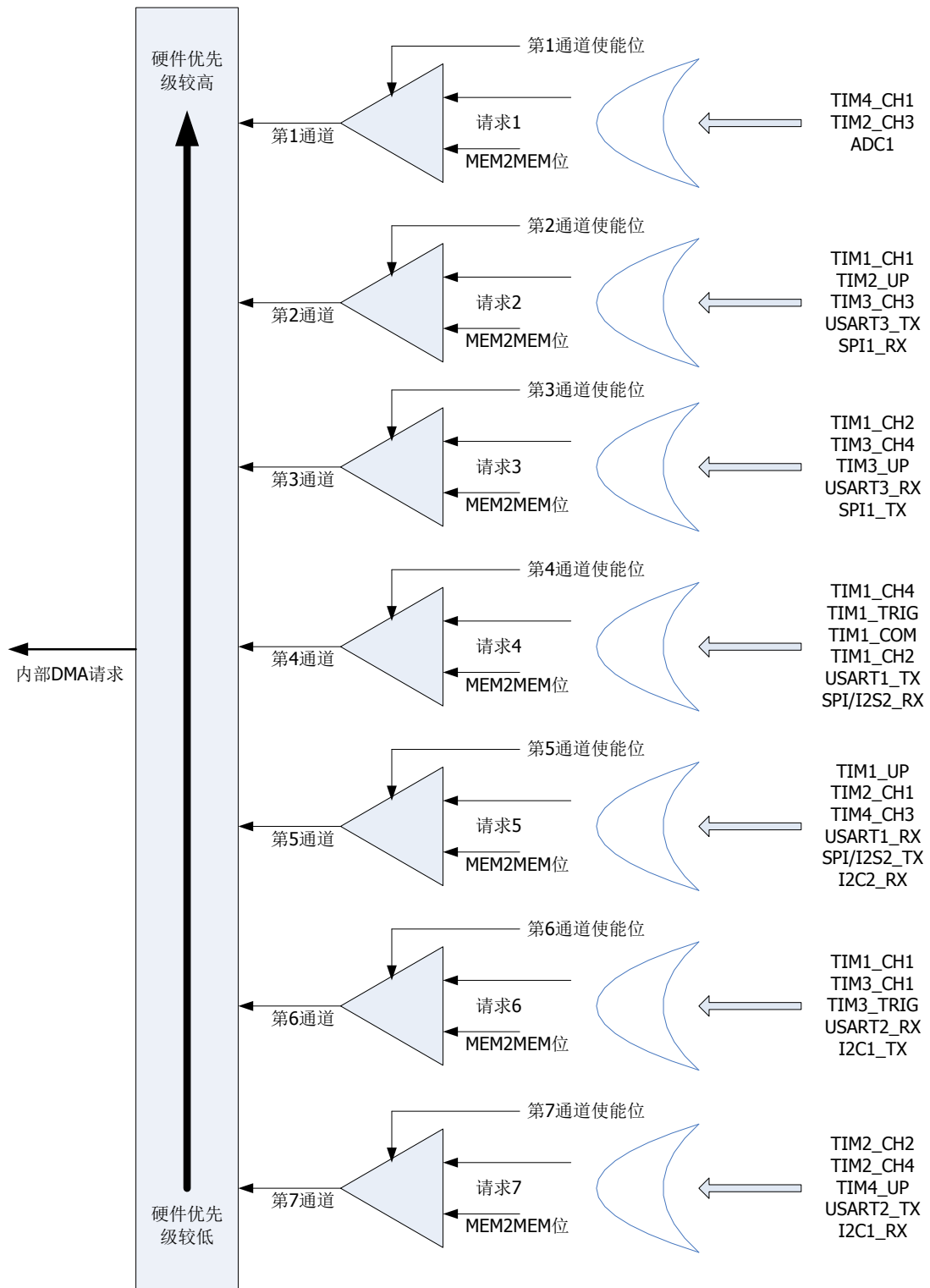
表 35 DMA 在源与地址间的访问动作

来源数据宽度	目的数据宽度	NDT 值	传输操作	
			来源	目的
32	32	4	1: 读操作, A3A2A1A0[31:0]@0x0 2: 读操作, A7A6A5A4[31:0]@0x4 3: 读操作, ABAAA9A8[31:0]@0x8 4: 读操作, AFAEADAC[31:0]@0xC	1: 写操作, A3A2A1A0[31:0]@0x0 2: 写操作, A7A6A5A4[31:0]@0x4 3: 写操作, ABAAA9A8[31:0]@0x8 4: 写操作, AFAEADAC[31:0]@0xC
	16		1: 读操作, A3A2A1A0[31:0]@0x0 2: 读操作, A7A6A5A4[31:0]@0x4 3: 读操作, ABAAA9A8[31:0]@0x8 4: 读操作, AFAEADAC[31:0]@0xC	1: 写操作, A1A0[15:0]@0x0 2: 写操作, A5A4[15:0]@0x2 3: 写操作, A9A8[15:0]@0x4 4: 写操作, ADAC[15:0]@0x6
	8		1: 读操作, A3A2A1A0[31:0]@0x0 2: 读操作, A7A6A5A4[31:0]@0x4 3: 读操作, ABAAA9A8[31:0]@0x8 4: 读操作, AFAEADAC[31:0]@0xC	1: 写操作, A0[7:0]@0x0 2: 写操作, A4[7:0]@0x1 3: 写操作, A8[7:0]@0x2 4: 写操作, AC[7:0]@0x3
16	32		1: 读操作, A1A0[15:0]@0x0 2: 读操作, A3A2[15:0]@0x2 3: 读操作, A5A4[15:0]@0x4 4: 读操作, A7A6[15:0]@0x6	1: 写操作, 0000A1A0[31:0]@0x0 2: 写操作, 0000A3A2[31:0]@0x4 3: 写操作, 0000A5A4[31:0]@0x8 4: 写操作, 0000A7A6[31:0]@0xC
	16		1: 读操作, A1A0[15:0]@0x0 2: 读操作, A3A2[15:0]@0x2 3: 读操作, A5A4[15:0]@0x4 4: 读操作, A7A6[15:0]@0x6	1: 写操作, A1A0[15:0]@0x0 2: 写操作, A3A2[15:0]@0x2 3: 写操作, A5A4[15:0]@0x4 4: 写操作, A7A6[15:0]@0x6
	8		1: 读操作, A1A0[15:0]@0x0 2: 读操作, A3A2[15:0]@0x2 3: 读操作, A5A4[15:0]@0x4 4: 读操作, A7A6[15:0]@0x6	1: 写操作, A0[7:0]@0x0 2: 写操作, A2[7:0]@0x1 3: 写操作, A4[7:0]@0x2 4: 写操作, A6[7:0]@0x3
8	32		1: 读操作, A0[7:0]@0x0 2: 读操作, A1[7:0]@0x1 3: 读操作, A2[7:0]@0x2 4: 读操作, A3[7:0]@0x3	1: 写操作, 000000A0[31:0]@0x0 2: 写操作, 000000A1[31:0]@0x4 3: 写操作, 000000A2[31:0]@0x8 4: 写操作, 000000A3[31:0]@0xC
	16		1: 读操作, A0[7:0]@0x0 2: 读操作, A1[7:0]@0x1 3: 读操作, A2[7:0]@0x2 4: 读操作, A3[7:0]@0x3	1: 写操作, 00A0[15:0]@0x0 2: 写操作, 00A1[15:0]@0x2 3: 写操作, 00A2[15:0]@0x4 4: 写操作, 00A3[15:0]@0x6
	8		1: 读操作, A0[7:0]@0x0 2: 读操作, A1[7:0]@0x1 3: 读操作, A2[7:0]@0x2 4: 读操作, A3[7:0]@0x3	1: 写操作, A0[7:0]@0x0 2: 写操作, A1[7:0]@0x1 3: 写操作, A2[7:0]@0x2 4: 写操作, A3[7:0]@0x3

3.7 外设 DMA 请求连接表

总共有 7 个外设可以产生 DMA 请求，但同一时刻只能有一个外设请求 DMA 操作。每个外设的 DMA 请求可以激活或者禁用。

图 16 DMA 请求映射



下表列出了每个通道所支持的外设请求。

表 36 各 DMA 通道的可用请求

外设	第 1 通道	第 2 通道	第 3 通道	第 4 通道	第 5 通道	第 6 通道	第 7 通道
----	--------	--------	--------	--------	--------	--------	--------

外设	第 1 通道	第 2 通道	第 3 通道	第 4 通道	第 5 通道	第 6 通道	第 7 通道
TM1	●	TM1_CH1	TM1_CH2	TM1_CH4 TM1_TRIG TM1_COM	TM1_UP	TM1_CH3	●
TM2	TM2_CH3	TM2_UP	●	●	TM2_CH1	●	TM2_CH2 TM2_CH4
TM3	●	TM3_CH3	TM3_CH4 TM3_UP	●	●	TM3_CH1 TM3_TRIG	●
TM4	TM4_CH1	●	●	TM4_CH2	TM4_CH3	●	TM4_UP
ADC1	ADC1	●	●	●	●	●	●
SPI	●	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	●	●
USART	●	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_TX	USART2_RX
I ² C	●	●	●	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX

4 DMA 寄存器

4.1 DMA 中断状态寄存器 (DMA_ISR)

地址偏移: 0x00

复位值 : 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
类型					R	R	R	R	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
类型	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位/位域	名称	描述
31:28	保留	请保持其复位状态
27/23/19/ 15/11/7/3	TEIFn	第 n 通道 (n=1~7) 的错误标志 这些位由硬件置位。软件可通过 DMA_IFCR 寄存器将其清零。 0: 第 n 通道未发生错误 1: 第 n 通道发生错误
26/22/18/ 14/10/6/2	HTIFn	第 n 通道 (n=1~7) 的半传输完成标志 这些位由硬件置位。软件可通过 DMA_IFCR 寄存器将其清零。 0: 第 n 通道尚未传输完成一半 1: 第 n 通道已传输完成一半

位/位域	名称	描述
25/21/17/ 13/9/5/1	TCIFn	第 n 通道 (n=1~7) 的传输完成标志 这些位由硬件置位。软件可通过 DMA_IFCR 寄存器将其清零。 0: 第 n 通道尚未全部传输完成 1: 第 n 通道已全部传输完成
24/20/16/ 12/8/4/0	GIFn	第 n 通道 (n=1~7) 的全局中断标志 这些位由硬件置位。软件可通过 DMA_IFCR 寄存器将其清零。 0: 第 n 通道尚未发生 TEIF、HTIF 或 TCIF 1: 第 n 通道已发生 TEIF、HTIF 或 TCIF 中至少一种

4.2 DMA 中断标志清除寄存器 (DMA_IFCR)

地址偏移: 0x04

复位值 : 0x0000 0000

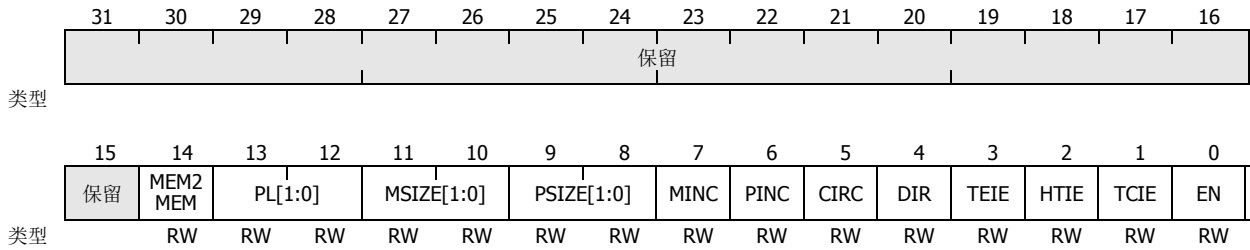
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
类型				W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
类型															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

位/位域	名称	描述
31:28	保留	请保持其复位状态
27/23/19/ 15/11/7/3	CTEIFn	清除第 n 通道 (n=1~7) 的错误标志 软件将本位置位, 即可将相应的标志位清零。 0: 无意义 1: 将 DMA_ISR 中第 n 通道的 TEIF 位清零
26/22/18/ 14/10/6/2	CHTIFn	清除第 n 通道 (n=1~7) 的半传输完成标志 软件将本位置位, 即可将相应的标志位清零。 0: 无意义 1: 将 DMA_ISR 中第 n 通道的 HTIF 位清零
25/21/17/ 13/9/5/1	CTCIFn	清除第 n 通道 (n=1~7) 的传输完成标志 软件将本位置位, 即可将相应的标志位清零。 0: 无意义 1: 将 DMA_ISR 中第 n 通道的 TCIF 位清零
24/20/16/ 12/8/4/0	CGIFn	清除第 n 通道 (n=1~7) 的全局中断标志 软件将本位置位, 即可将相应的标志位清零。 0: 无意义 1: 将 DMA_ISR 中第 n 通道的 GIF 位清零

4.3 DMA 通道 x 配置寄存器 (DMA_CCRx) (x 代表通道号, x = 1~7)

地址偏移: $0x08 + 0d20 \times (x - 1)$

复位值 : 0x0000 0000



位/位域	名称	描述
31:15	保留	请保持其复位状态
14	MEM2MEM	存储器到存储器模式 由软件置位和清零。 0: 禁用存储器到存储器模式 1: 使能存储器到存储器模式
13:12	PL[1:0]	本通道的优先级 本位域可由软件置位和清零。 00: 最低优先级 01: 低优先级 10: 高优先级 11: 最高优先级
11:10	MSIZE[1:0]	存储器传输数据单元的大小 本位域可由软件置位和清零。 00: 8 位 01: 16 位 10: 32 位 11: 保留
9:8	PSIZE[1:0]	外设传输数据单元的大小 本位域可由软件置位和清零。 00: 8 位 01: 16 位 10: 32 位 11: 保留
7	MINC	存储器递增模式 由软件置位和清零。 0: 使能存储器递增模式 1: 禁用存储器递增模式
6	PINC	外设递增模式 由软件置位和清零。 0: 使能外设递增模式 1: 禁用外设递增模式
5	CIRC	循环模式 (若 MEM2MEM 置位, 则此位设置无效) 由软件置位和清零。 0: 禁用循环模式 1: 使能循环模式
4	DIR	本通道的数据传输方向 由软件置位和清零。 0: 外设作为来源设备 1: 存储器作为来源设备

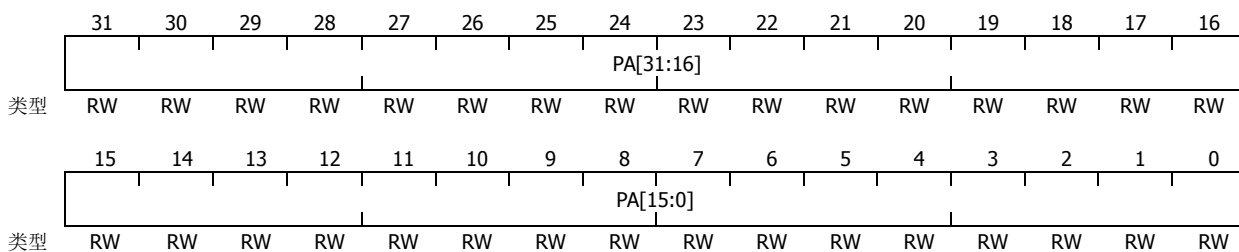
位/位域	名称	描述
3	TEIE	通道传输错误中断使能位 由软件置位和清零。 0: 禁用 TE 中断 1: 使能 TE 中断
2	HTIE	通道半传输完成中断使能位 由软件置位和清零。 0: 禁用 HT 中断 1: 使能 HT 中断
1	TCIE	通道传输完成中断使能位 由软件置位和清零。 0: 禁用 TC 中断 1: 使能 TC 中断
0	EN	激活通道 由软件置位和清零。 0: 禁用本通道 1: 使能本通道

4.4 DMA 通道 x 外设地址寄存器 (DMA_CPARx) (x 代表通道号, x = 1~7)

地址偏移: $0x10 + 0d20 \times (x - 1)$

复位值 : 0x0000 0000

注意: 当通道已经使能后, 不得修改此寄存器。



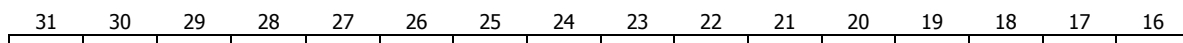
位/位域	名称	描述
31:0	PA[31:0]	由软件配置 DMA 外设的基地址 若 PSIZE 配置为 16 位, 则自动按半字对齐。 若 PSIZE 配置为 32 位, 则自动按字对齐。

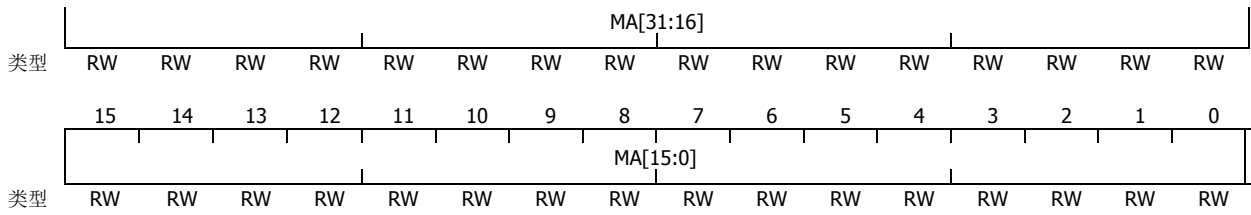
4.5 DMA 通道 x 存储器地址寄存器 (DMA_CMARx) (x 代表通道号, x = 1~7)

地址偏移: $0x14 + 0d20 \times (x - 1)$

复位值 : 0x0000 0000

注意: 当通道已经使能后, 不得修改此寄存器。





位/位域	名称	描述
31:0	MA[31:0]	由软件配置 DMA 存储器的基地址 若 MSIZE 配置为 16 位，则自动按半字对齐。 若 MSIZE 配置为 32 位，则自动按字对齐。

第7章 定时器 (TMx)

1 高级定时器 (TM1)

1.1 简介

高级定时器即 TM1，可用于各种高级控制目的。TM1 包含一个 16 位的向上/向下计数器，四个 16 位比较/捕获寄存器 (TM1_CHCC)，一个 16 位自动重载寄存器 (TM1_CARL) 以及多个控制寄存器。结合这些寄存器，可实现各种各样的功能，包括通用定时器、输入信号脉宽测量或输出特定波形（例如单脉冲输出或 PWM 输出）。TM1 使用一个两输入的解码器，支持编码器接口。

高级定时器 (TM1) 以及通用定时器 (TMx) 是相互完全独立的，并不共享任何资源，但是可以同步工作。

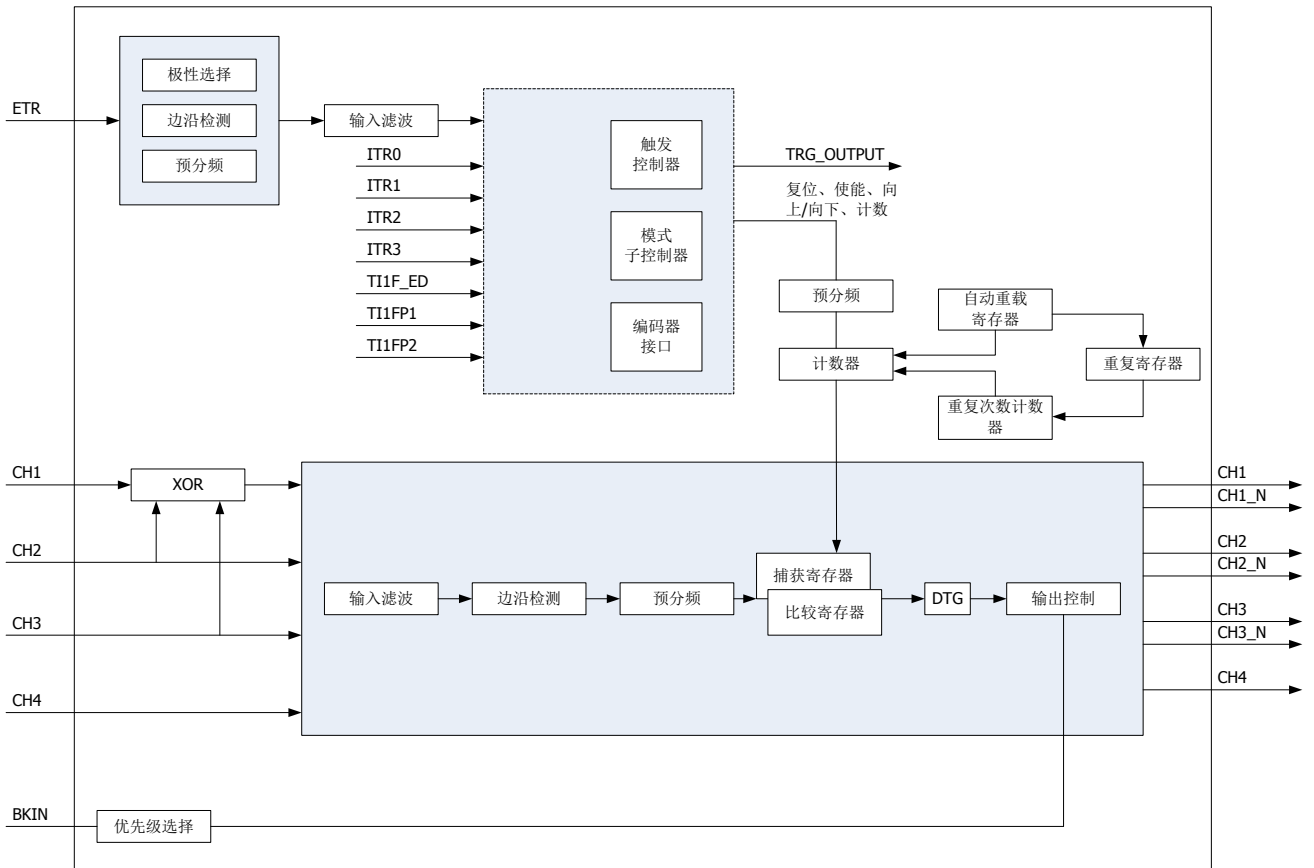
1.2 主要特点

- 16 位向下、向上、下上自动重载计数器
- 16 位可编程预分频器，计数器时钟频率可按照 1~65536 之间的任意系数分频
- 最大 4 个独立通道，支持多种功能，包括捕获、比较匹配输出、产生 PWM 波形（边沿对齐模式或中心对齐模式）以及单脉冲模式输出
- 重复次数计数器，仅在给定的计数器周期后才更新定时器寄存器
- 紧急停止输入，可触发定时器将输出信号恢复到一个可知的状态
- 根据更新、触发事件、输入捕获事件、输出比较匹配事件或紧急停动输入，产生中断或 DMA 请求
- 提供可编程的死区时间，用于实现输出匹配
- 提供同步电路，控制 TM1 按照外部信号工作，或实现多个定时器的级联
- 编码接口控制器，将两输入通过正交解码工作
- TM1 主/从模式控制器

1.3 功能描述

下图描绘了高级定时器的内部配置详情。

图 17 高级定时器内部框图



预分频计数器

预分频器可将定时器时钟 (CK_PSC) 分频得到计数器时钟 (CK_CNT)，分频系数可在 1~65536 之间任意设置。分频系数可通过预分频系数寄存器 (TM1_PSC) 更改，而且在运行时也能更改，但要等到下一次更新事件后才会生效。

图 18 分频系数从 1 改为 2 的计数器时序图

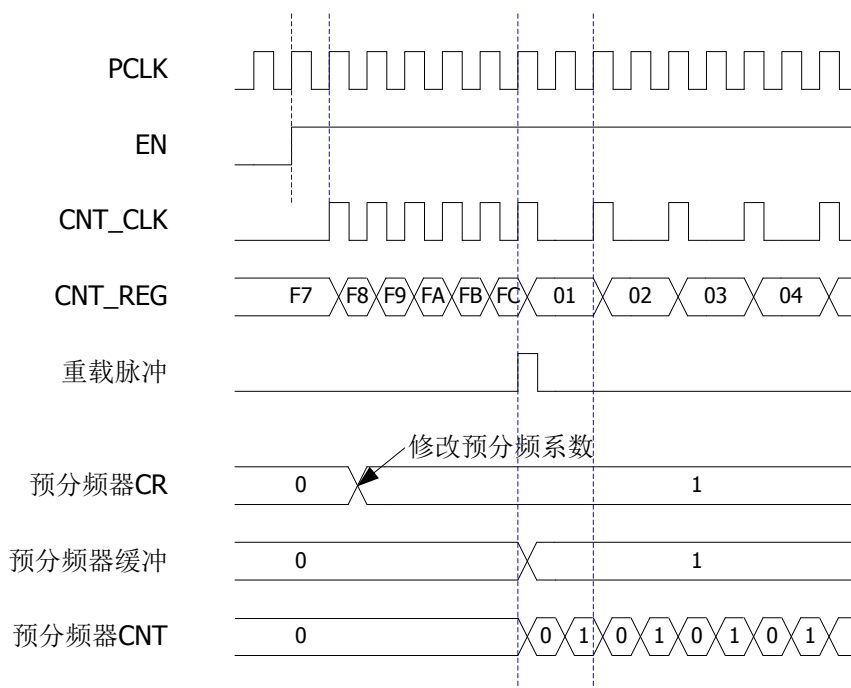
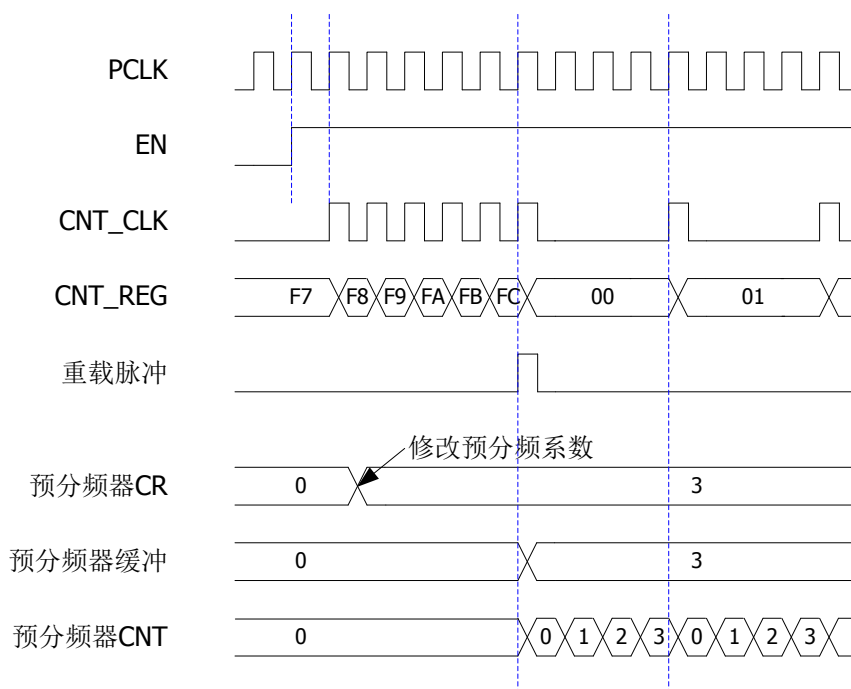


图 19 分频系数从 1 改为 4 的计数器时序图



向上计数模式

在这种工作模式下，计数器的值从 0 开始累加，一直累加到计数器重载值（在 `TM1_CARL` 寄存器

中设置)。一旦计数器值达到了重载值，寄存器将从 0 开始重新计数。假如设置过重复次数计数器，那么在计数值溢出若干次后将产生更新事件；否则每当计数值溢出时都会产生更新事件。对于向上计数模式，TM1_CTRL1 寄存器中的计数方向 DIR 位必须设置为 0。

当 TM1_EVG 寄存器中的 UPG 置位时，表明发生了更新事件，此时计数值将初始化为 0，并产生一个更新事件。

假如将 TM1_CTRL1 寄存器中的 UPDIS 置位，那么更新事件将被禁用。

当更新事件发生时，所有寄存器的内容（包括重复次数计数器、重载值寄存器、预分频系数寄存器）都将更新。

下面通过一些示例来演示一下不同的时钟频率下计数器的表现（TMx_ARR = 0x63）。

图 20 分频系数为 1 的计数器时序图

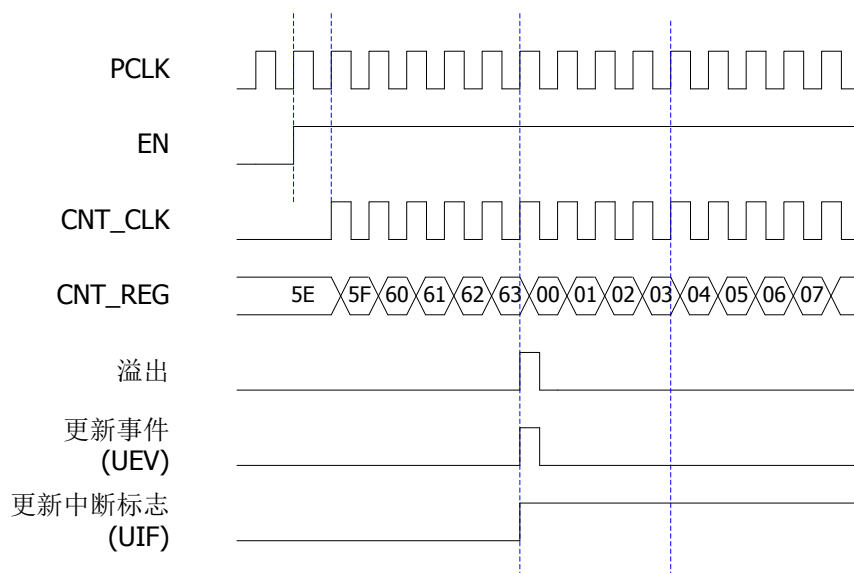


图 21 分频系数为 2 的计数器时序图

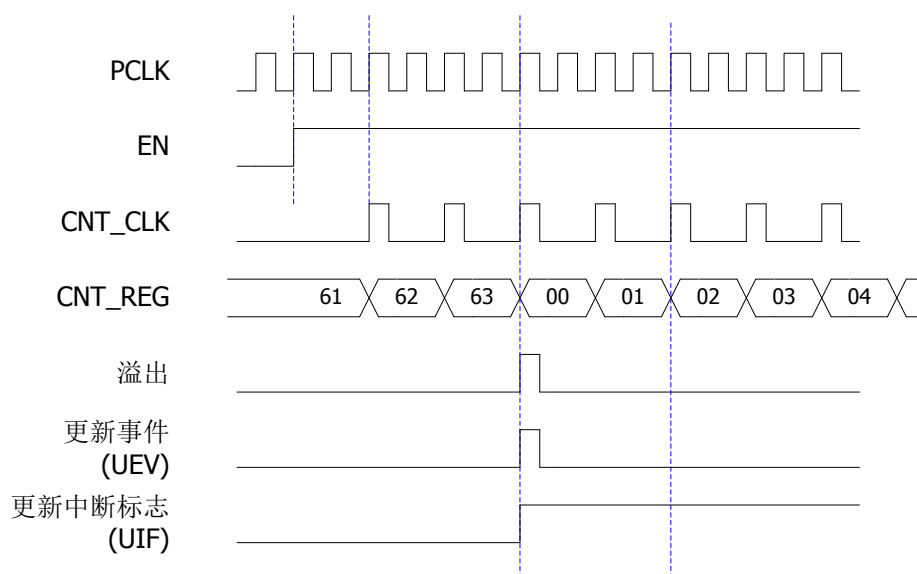


图 22 分频系数为 4 的计数器时序图

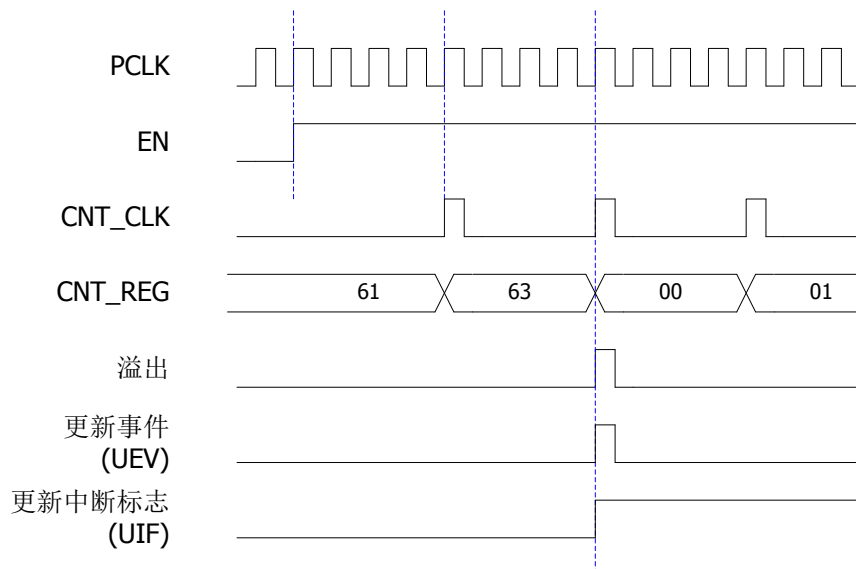


图 23 分频系数为 N 的计数器时序图

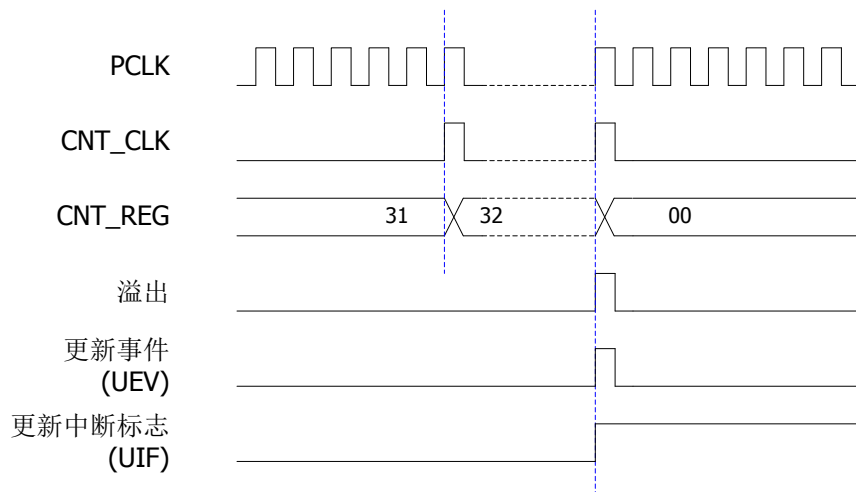


图 24 当 ARPE=0 时计数器更新事件的计数器时序图

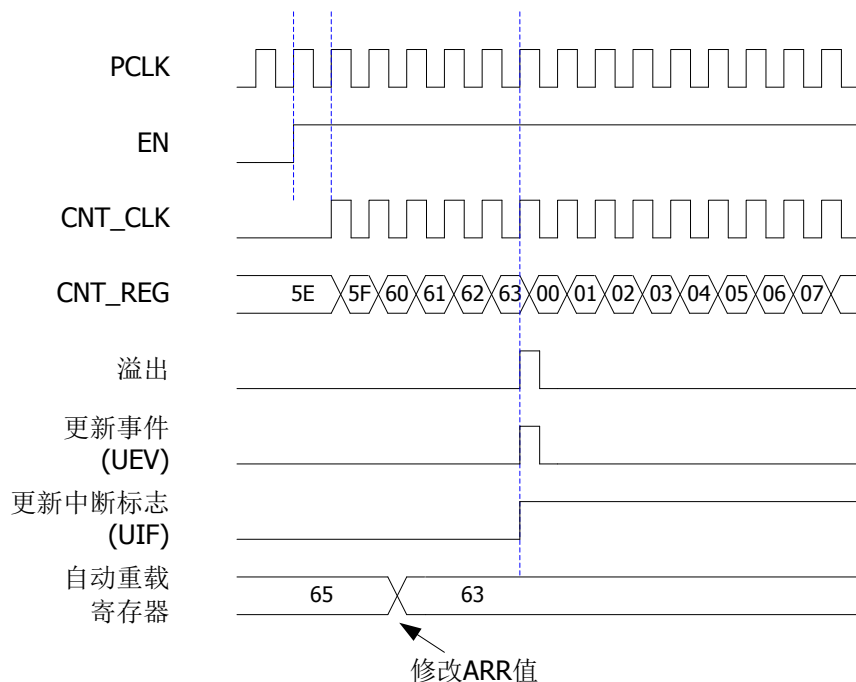
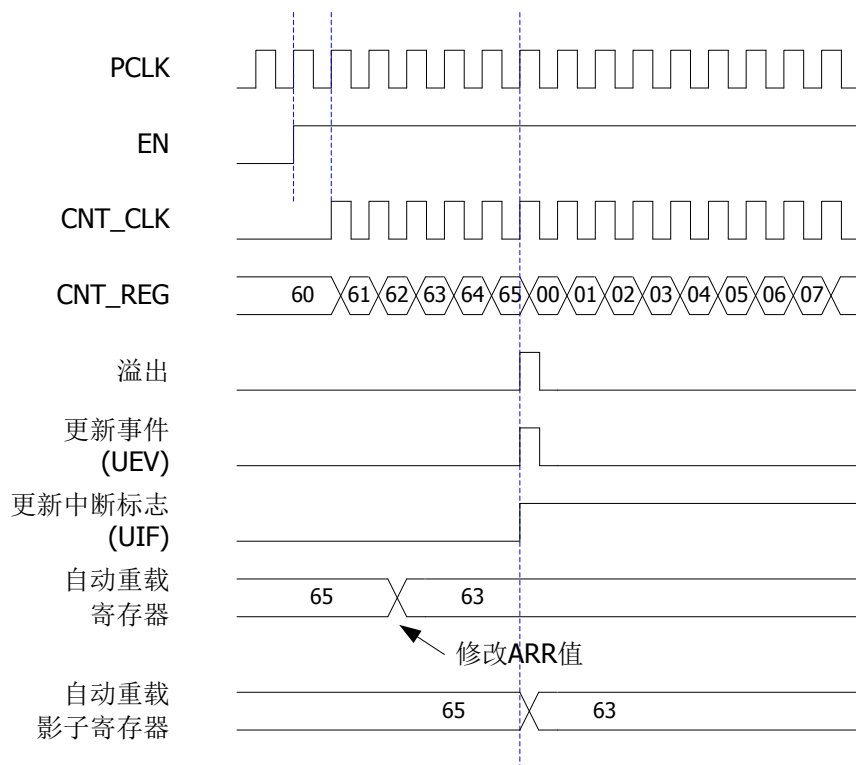


图 25 当 ARPE=1 时计数器更新事件的计数器时序图



向下计数模式

在这种工作模式下，计数器的值从重载值（在 `TM1_CARL` 寄存器中设置）开始递减，一直递减到 0。一旦计数器值达到了 0，计数器将重新从重载值开始向下计数。假如设置过重复次数计数器，那么在计数值下溢一定次数后便会产生更新事件；否则每次计数器下溢后都会产生更新事件。在向下计数模式下，`TM1_CTRL1` 寄存器中的计数方向 `DIR` 位必须设置为 1。

当 `TM1_EVG` 寄存器中的 `UPG` 置位时，表明发生了更新事件，计数值将初始化为重载值并产生一个更新事件。

假如将 `TM1_CTRL1` 寄存器中的 `UPDIS` 置位，那么更新事件将被禁用。

当更新事件发生时，所有寄存器（包括重复次数计数器、重载值寄存器、预分频系数寄存器）的内容都将更新。

下面通过一些示例来演示一下不同的时钟频率下计数器的表现（`TMx_ARR = 0x63`）。

图 26 分频系数为 1 的计数器时序图

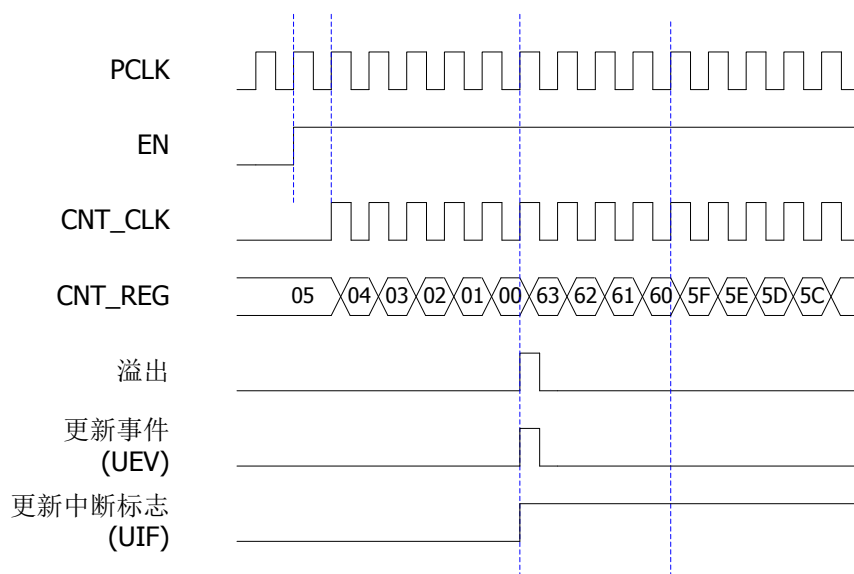


图 27 分频系数为 2 的计数器时序图

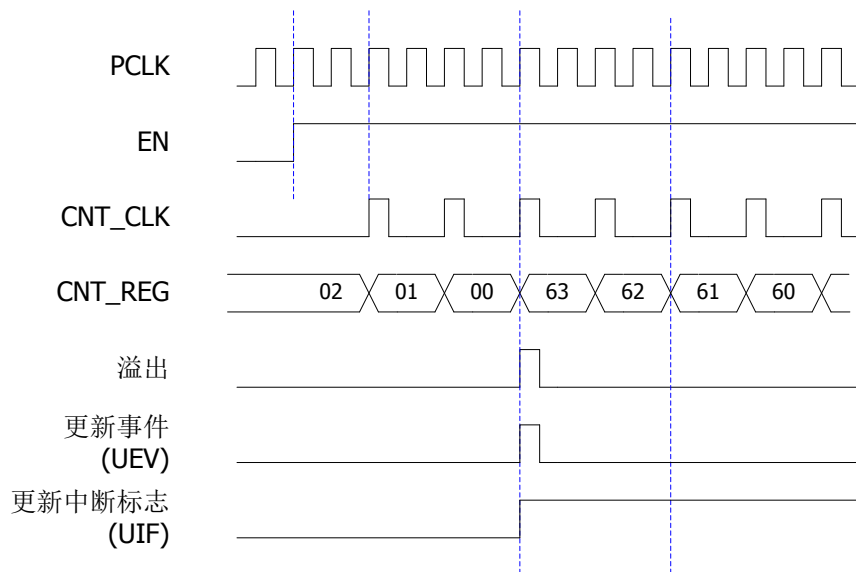


图 28 分频系数为 4 的计数器时序图

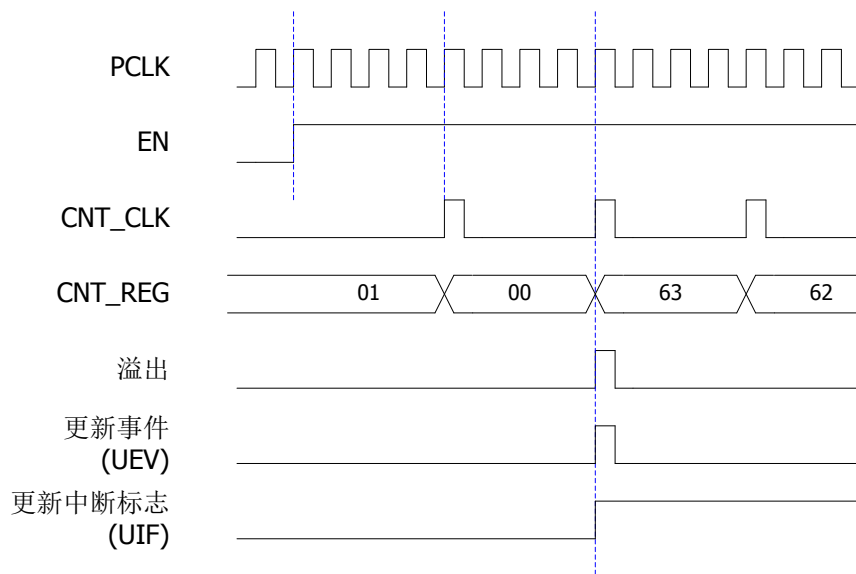


图 29 分频系数为 N 的计数器时序图

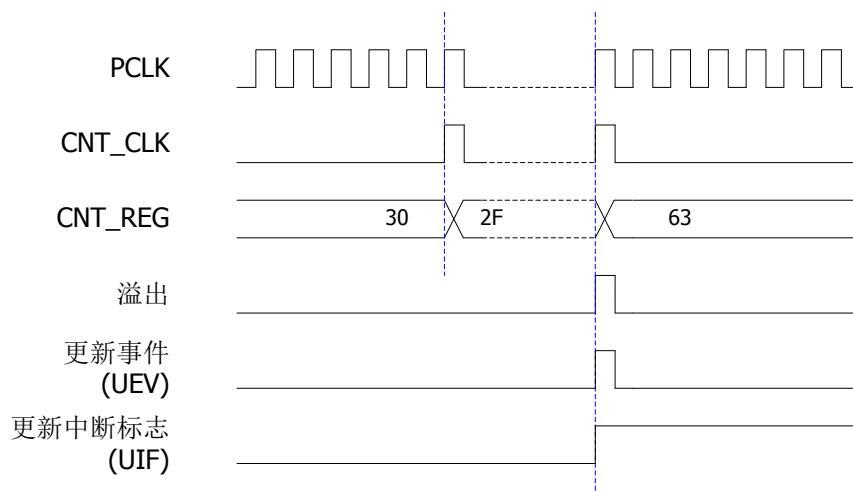
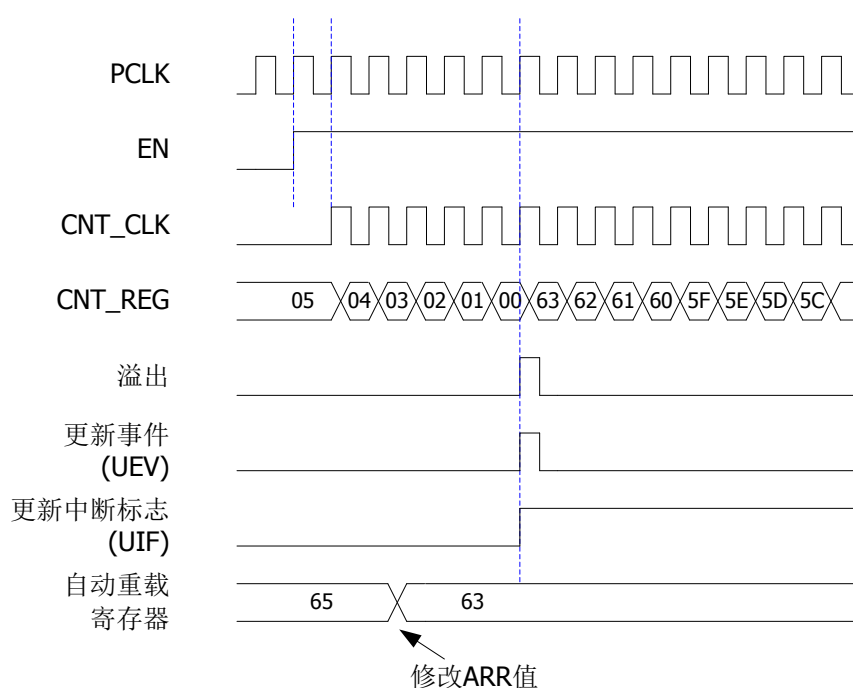


图 30 当计数器未使用时，计数器更新事件的计数器时序图



中心对齐计数模式

在中心对齐计数模式下，计数器的值从 0 开始递增到重载值，然后从重载值开始递减到 0，如此反复。定时器模块在计数值递增到重载值减 1 时会产生一个溢出事件，在计数值递减到 1 时产生一个下溢事件。在中心对齐模式下，TM1_CTRL1 寄存器中的计数方向 DIR 位只读，用于表明当前的计数方向。该位将由硬件自动更新。

在中心对齐模式下，将 TM1_EVG 寄存器中的 UPG 置位，即可将计数值清零（不管当前计数器是

向上计数还是向下计数) 并产生一个更新事件。

在向下计数 (TM1_CTRL1 寄存器中的 CAM 位域为 01) 模式下发生下溢事件、在向上计数 (TM1_CTRL1 寄存器中的 CAM 位域为 10) 模式下发生溢出事件、或在中心对齐 (TM1_CTRL1 寄存器中的 CAM 位域为 11) 模式下发生下溢或溢出事件时, TM1_EVG 寄存器中的 UPIF 位都会置位。

若将 TM1_CTRL1 寄存器中的 UPDIS 置位, 那么更新事件将被禁用。

当更新事件发生时, 所有寄存器 (包括重复次数计数器、重载值寄存器、预分频系数寄存器) 的内容都将更新。

下面通过一些示例来演示一下不同的时钟频率下计数器的表现 (TMx_ARR = 0x5)。

图 31 分频系数为 1、TMx_ARR = 0x5 的计数器时序图

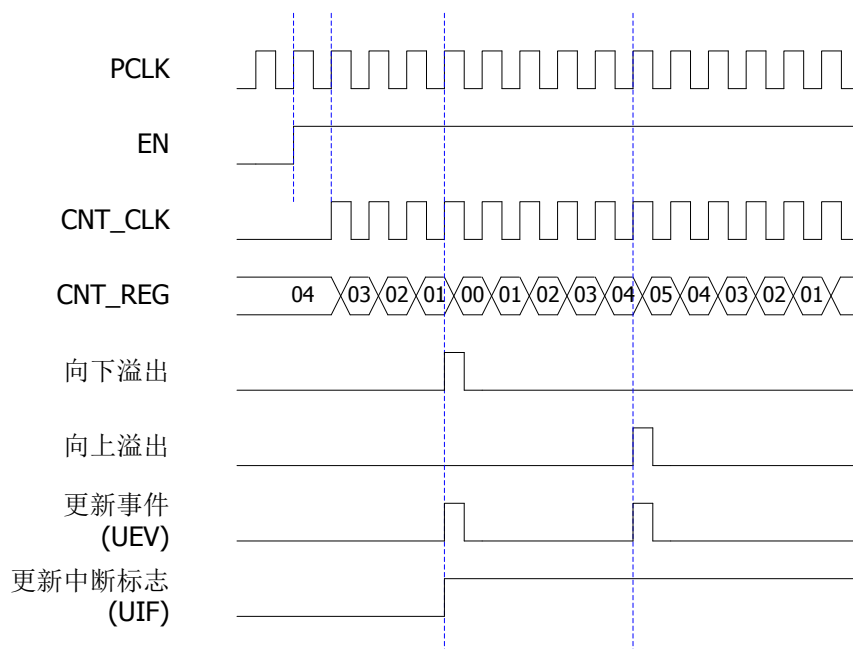


图 32 分频系数为 2、TMx_ARR = 0x5 的计数器时序图

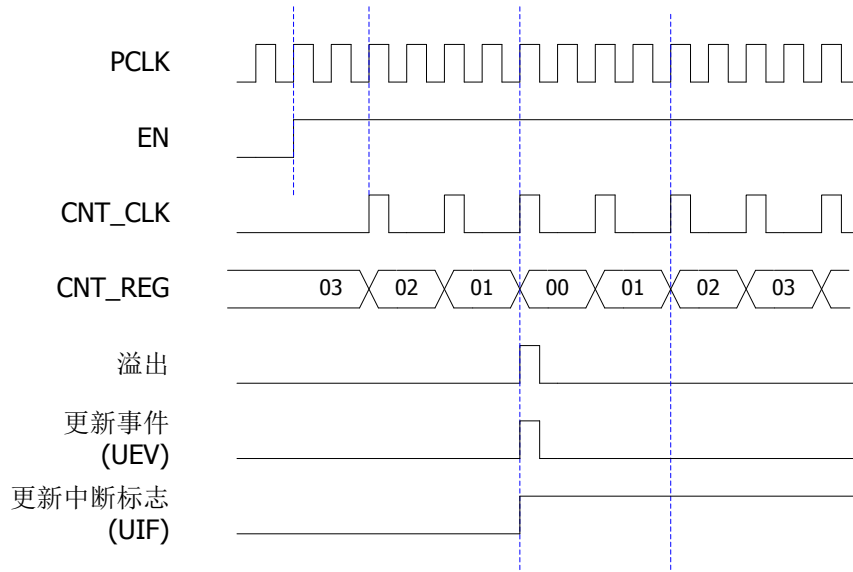


图 33 分频系数为 4、TMx_ARR = 0x63 的计数器时序图

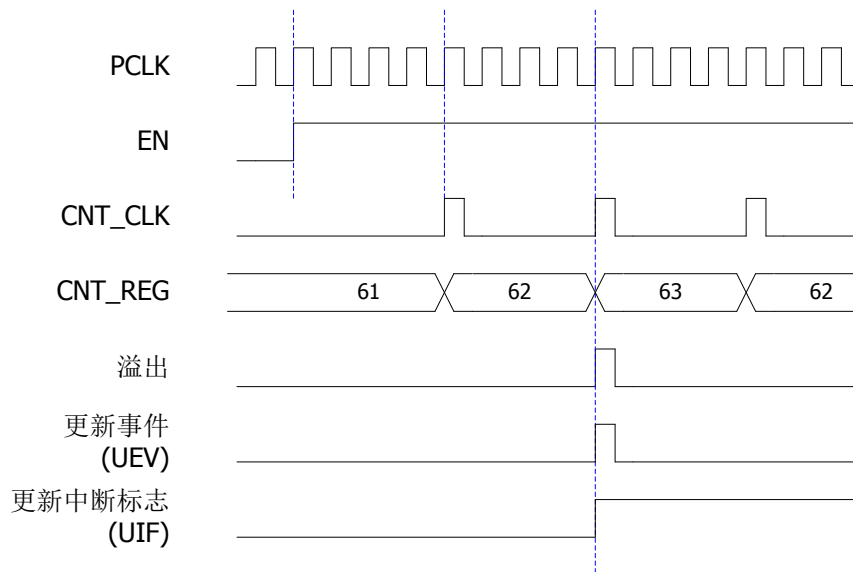


图 34 分频系数为 N 时的计数器时序图

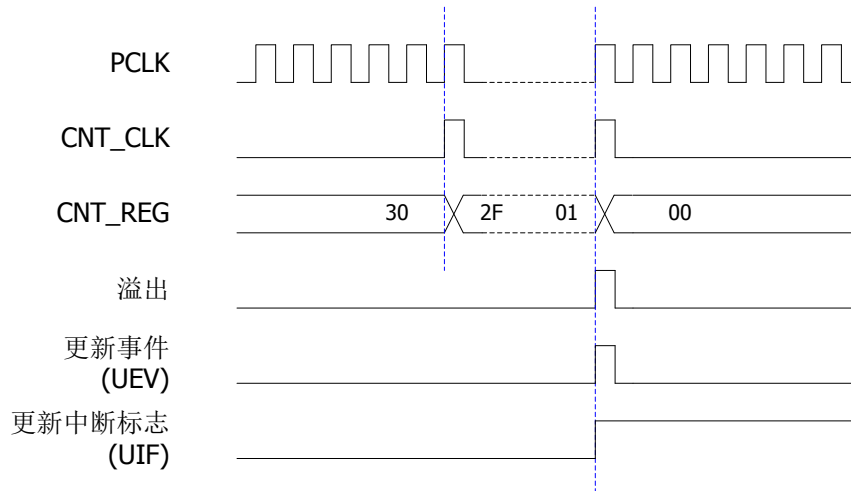


图 35 APRE = 1 时计数器更新事件的计数器时序图

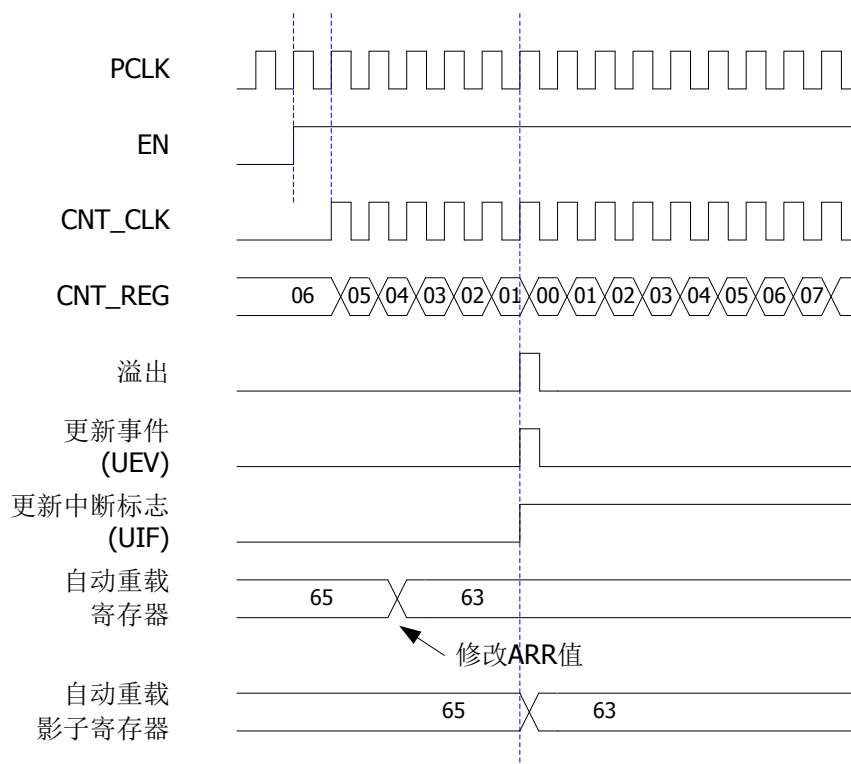
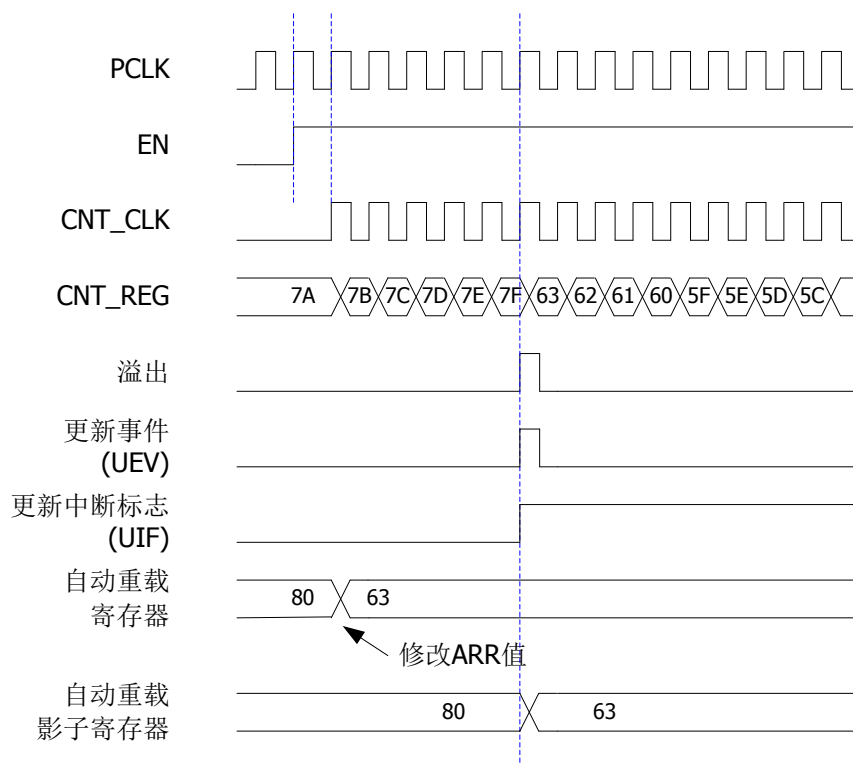


图 36 APRE = 1 时计数器更新事件的计数器时序图



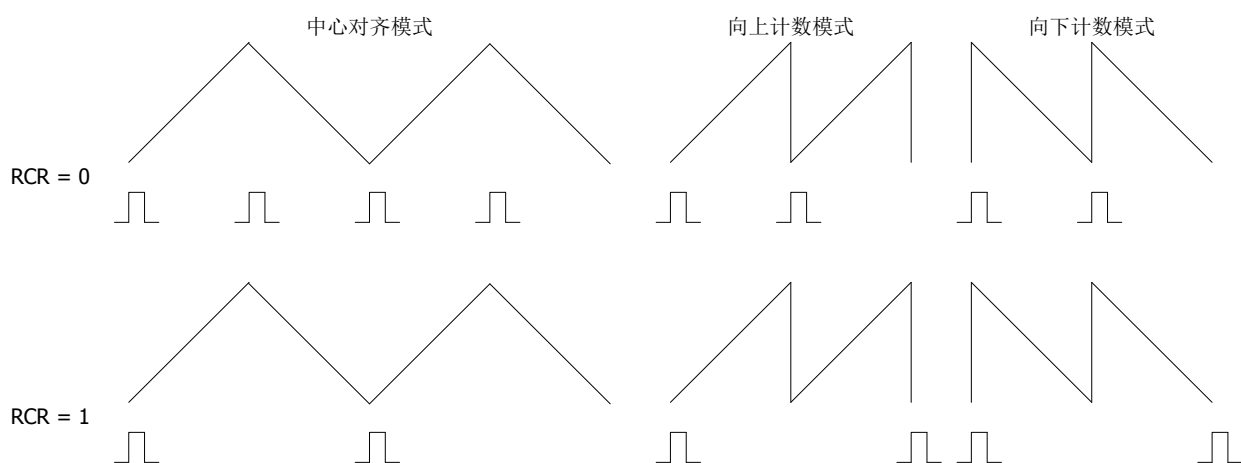
重复次数计数器

借助重复次数计数功能，可以在计数值溢出/下溢了指定的 N 个周期之后 (N 通过 **TM1_CREP** 寄存器中的 **CREP** 位域设置)，才会产生计数器更新事件、更新定时器各寄存器的内容。向上计数模式下的每次计数值溢出、向下计数模式下的每次计数值下溢、或中心对齐模式下的每次计数值溢出/下溢，都会使重复次数计数值递减。

将 **TM1_EVG** 寄存器中的 **UPG** 置位，即可重载 **TM1_CREP** 寄存器中的 **CREP** 位域，并产生一个更新事件。

在中心对齐模式下，若 **RCR** 的值为奇数，在启动计数器之前写 **CREP** 位域后发生溢出，即可产生一个更新事件；在启动计数器之后写 **CREP** 位域后发生下溢，即可产生一个更新事件。

图 37 更新速率示例，取决于工作模式以及 TMx_RCR



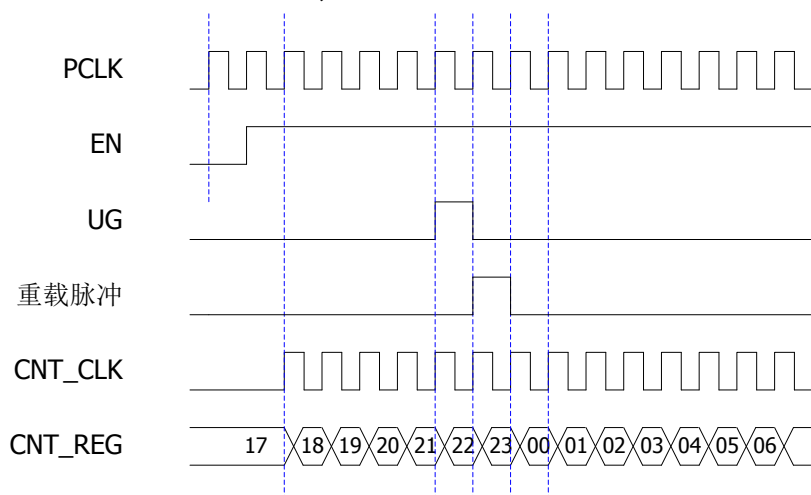
时钟选择

下面介绍用于决定内部预分频计数时钟源的时钟控制器。

■ 内部定时器时钟 CK_TM1

默认的内部时钟源是 APB2 时钟 CK_APB2，禁用从机模式后即被用于驱动预分频计数器。若将 TM1_SMCR 寄存器中的 SMC 位域设置为 0x1、0x2、0x3、0x6、0x7 中的某个值，那么从模式控制器将被使能，且预分频器将由其它时钟源（通过 TM1_SMCR 寄存器中的 TRGS 位域选择）供给，详见后述。当从模式选择位域 SMC 设置为 0x4 或 0x5，则内部时钟 CK_TM1 被选作预分频计数器的驱动时钟源。

图 38 标准模式下的控制电路，内部时钟分频系数为 1



■ 正交解码器

要选择正交解码器模式，TM1_SMCR 寄存器中的 SMC 位域必须为 0x1、0x2 或 0x3 中的某个值。正交编码器功能采用 TM1_CH0、TM1_CH1 两管脚的输入状态产生时钟脉冲，以该脉冲驱动预分频计数器。计数方向位 DIR 则由硬件在输入源信号的每次跳变沿自动选择。输入源信号可以只从 TM1_CH0 管脚获取，只从 TM1_CH1 管脚获取，或同时从 TM1_CH0、TM1_CH1 获取。

■ 内部触发器输入 (ITI)

预分频计数器可以在 ITI 信号的每个上升沿或下降沿计数。要选择这种模式，应将 TM1_SMCR 寄存器中的 SMC 位域设置为 0x6；此时计数器的表现如同一个事件计数器。输入事件（也就是这里的 ITI）可通过设置 TRGS 位域进行选择。若选择 ITI 信号作为时钟源，那么内部边沿检测电路将在 ITI 信号的每个上升沿或下降沿产生一个时钟脉冲，用以驱动预分频计数器。

■ 外部输入管脚 (Tix)

预分频计数器可以由外部管脚 TM1_Tix 的每个上升沿或者下降沿驱动计数。要选择这种模式，应将 TM1_SMCR 寄存器中的 SMC 位域设置为 0x7，并将 TRGS 位域设置为 0x4、0x5 或 0x6。请注意：Tix 是从 TM1_Tix 经过一个数字滤波器后采样得到的。

■ 外部触发器输入 (ETIF)

预分频计数器可以由外部管脚 TM1_ETI 的每个上升沿或者下降沿驱动计数。要选择这种模式，应将 TM1_SMCR 寄存器中的 ECM2E 置位。另一种选择 ETIF 信号作为时钟源的方法是将 SMC 位域设为 0x6，并将 TRGS 位域设为 0x7。请注意：ETIF 信号是从 TM1_ETI 管脚经过一个数字滤波器后采样得到的。若选择 ETIF 信号作为时钟源，那么包含边沿检测电路的触发控制器将在 ETIF 信号的每个上升沿产生一个时钟脉冲，用以驱动预分频计数器。

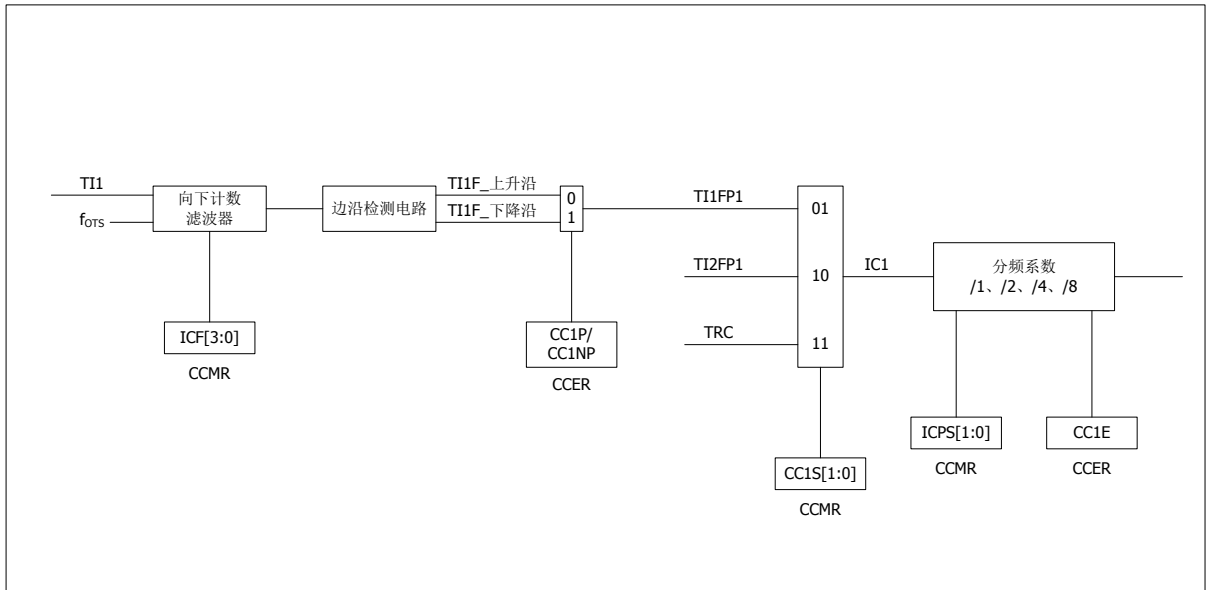
捕获/比较通道

TM1 有 4 个独立的通道，可用于捕获输入或者比较匹配输出。每个通道都是围绕着一个通道捕获/比较寄存器设计的，由一个输入级、通道控制器以及一个输出级组成。

■ 输入捕获级

输入捕获级由数字滤波器、通道极性选择、边沿检测电路以及通道预分频器组成。通道 0 的输入信号 (TI0) 可以选择来自 TM1_CH0 管脚，或者 TM1_CH0、TM1_CH1、TM1_CH2 管脚的异或运算结果。通道输入信号 (Tix) 经由数字滤波器采样后产生输入信号 TixF，之后通道极性选择以及边沿检测电路可产生 TixS0ED 或 TixS1ED 信号，实现输入捕获功能。有效的输入事件号可通过通道输入预分频寄存器 (CHx_ICP) 设置。

图 39 捕获/比较通道 (示例: 通道 1 输入级)



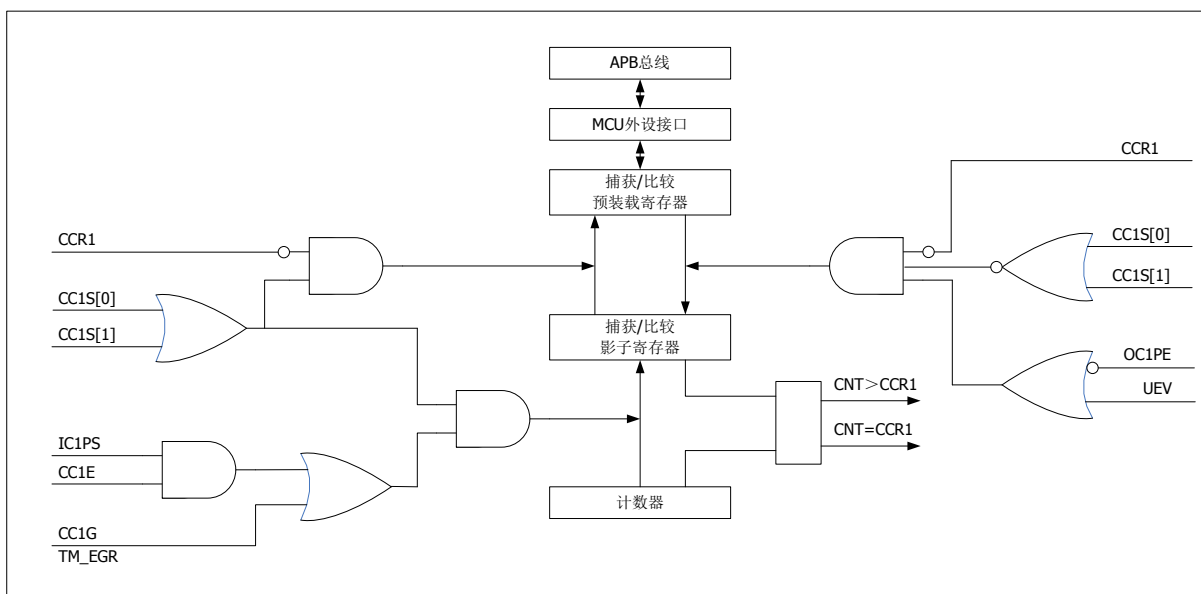
■ 通道控制器

高级定时器有 4 个独立的通道，可用于捕获输入或比较匹配输出。

当用于输入捕获模式时，当产生捕获事件后，计数值先被捕获到 $TM1_CHCCx$ 影子寄存器中，之后传递到 $TM1_CHCCx$ 预装载寄存器中。

当用于比较匹配输出模式时， $TM1_CHCCx$ 预装载寄存器的值将拷贝到对应的影子寄存器中，随后将计数值与寄存器值进行比较。

图 40 捕获/比较通道 1 主电路



■ 输出级

TM1 有 4 个通道，用于比较匹配、单脉冲输出或 PWM 输出。

图 41 捕获/比较通道 1~3 输出级

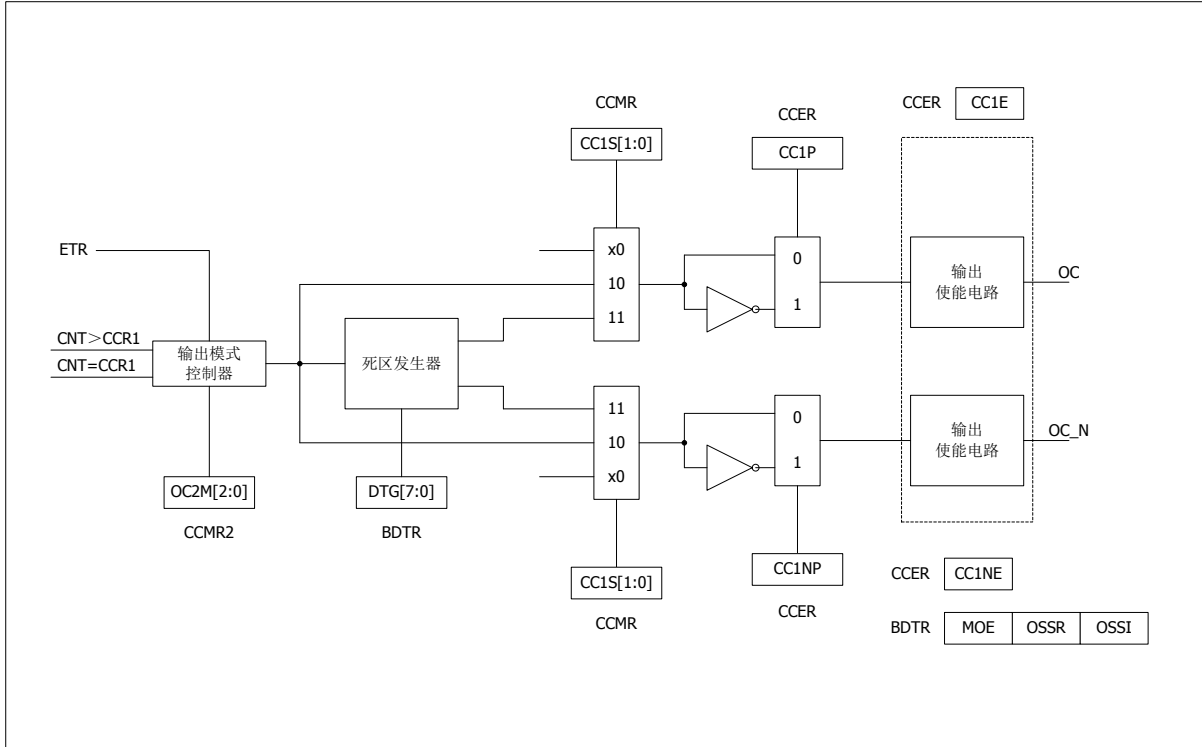


图 42 捕获/比较通道 4 输出级

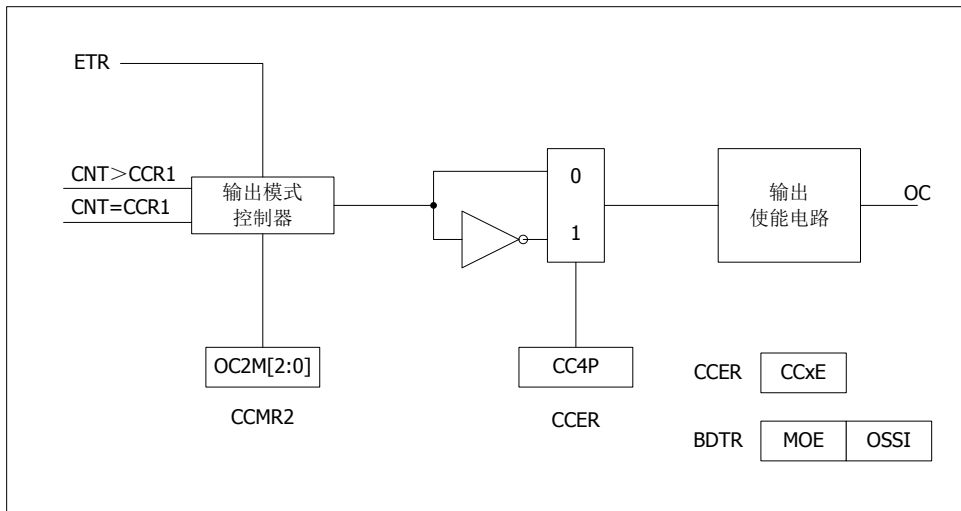
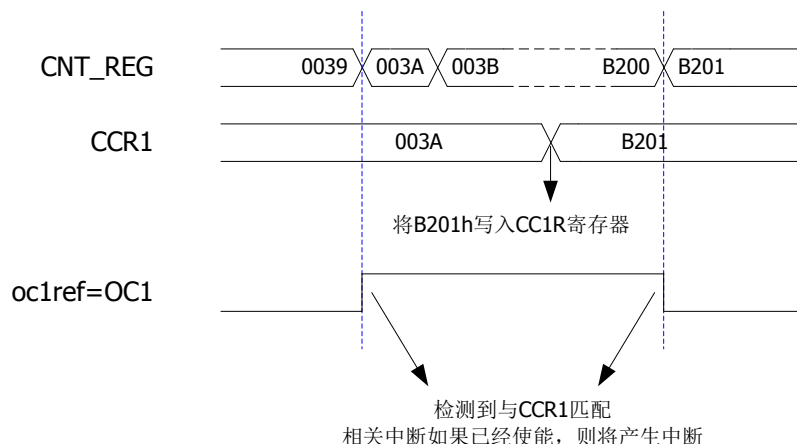


图 43 输出比较模式，OC1 的翻转



当通道用于捕获输入时，一旦输入信号发生有效跳变沿，计数值将被捕获到通道捕获/比较寄存器 (TM1_CHCCx) 中。捕获事件产生后，TM1_STA 寄存器中的 CHxIF 标志将会置位。假如 CHxIF 标志位已经置位（即该标志之前未被软件清零），并且本通道又产生了新的捕获事件，那么相应通道的过捕获标志位 CHxOF 将会置位。捕获事件产生后，取决于 CHxDE 位的状态将可能产生 DMA 请求，取决于 CHxIE 位的状态将可能产生中断。

输入捕获模式也可用于对 TM1_CHx 管脚的信号 (TIx) 进行脉宽测量。

通道输出参考信号

当 TM1 作为比较匹配输出模式时，OCxREF 信号（第 x 通道输出参考信号）将通过配置 CHxOM 位域进行定义。当计数值与 TM1_CHCCx 寄存器的内容匹配时，OCxREF 信号可以有多种类型的输出功能，包括：将 CHxOM 位域设为 0x00 即可保持原始电平，将 CHxOM 位域设为 0x01 即可置 1，将 CHxOM 位域设为 0x02 即可置 0，将 CHxOM 位域设为 0x03 即可翻转信号。

PWM 模式 1 输出以及 PWM 模式 2 输出也是另一种类型的 OCxREF 输出，分别通过将 CHxOM 位域置为 0x6/0x7 来实现。在这两种模式中，OCxREF 信号电平将根据计数方向以及计数值与 TM1_CHCCx 寄存器内容之间的关系进行变化。详见相关寄存器的说明。

OCxREF 信号的另一种特殊功能就是强制输出，可通过将 CHxOM 位域设为 0x4/0x5 来实现。此时输出可强制为失效电平/生效电平，不管计数器值与 TM1_CHCCx 寄存器内容之间的关系如何。

当 ETIF 信号是从外部的 TM1_ETI 管脚获取的，可强制 OCxREF 信号为 0；若将 TM1_CHCTRL1 寄存器中的 CHxOCE 置位，也可以强制 OCxREF 信号为高电平。在下一个计数器更新事件发生前，OCxREF 信号不会返回至生效电平状态。

互补输出以及死区时间

TM1 能输出一对互补信号。通过组合多个控制位，即可激活互补信号 OCx 和 OCxN：TM1_CHE 寄存器中的 CHxE 以及 CHxNE、TM1_BKDT 以及 TM1_CTRL2 寄存器中的 POE、ISOx、ISOxN、

IOS、ROS。输出极性由 TM1_CHE 寄存器中的 CHxP 以及 CHxNP 决定。

假如 CHxE、CHxNE、POE 都置位，那么就应当插入死区时间。OCx 输出信号的上升沿将相较于 OCxREF 信号上升沿有一定的延迟，OCxN 输出信号的上升沿将相较于 OCxREF 信号下降沿有一定的延迟。延迟时间是由 10 位的死区时间计数器（TM1_BKDT 寄存器的 DT 位域）决定的。假如延迟时间大于有效输出（OCx 或 OCxN）宽度，那就不会产生相应的脉冲。

图 44 带有死区时间插入的互补输出

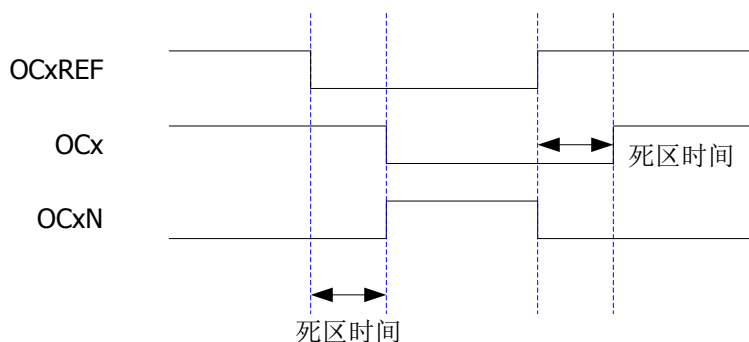


图 45 当延迟大于负脉冲时的死区时间波形

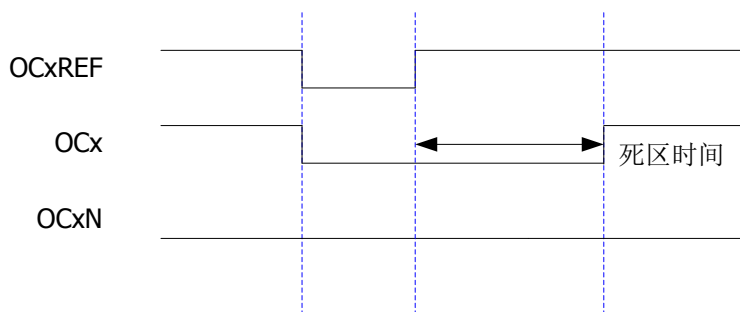
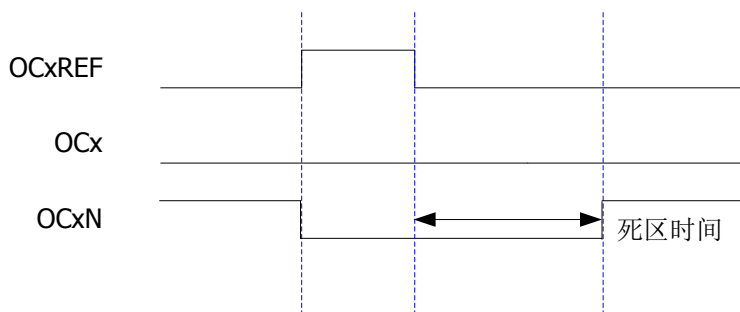


图 46 当延迟大于正脉冲时的死区时间波形



紧急停止功能

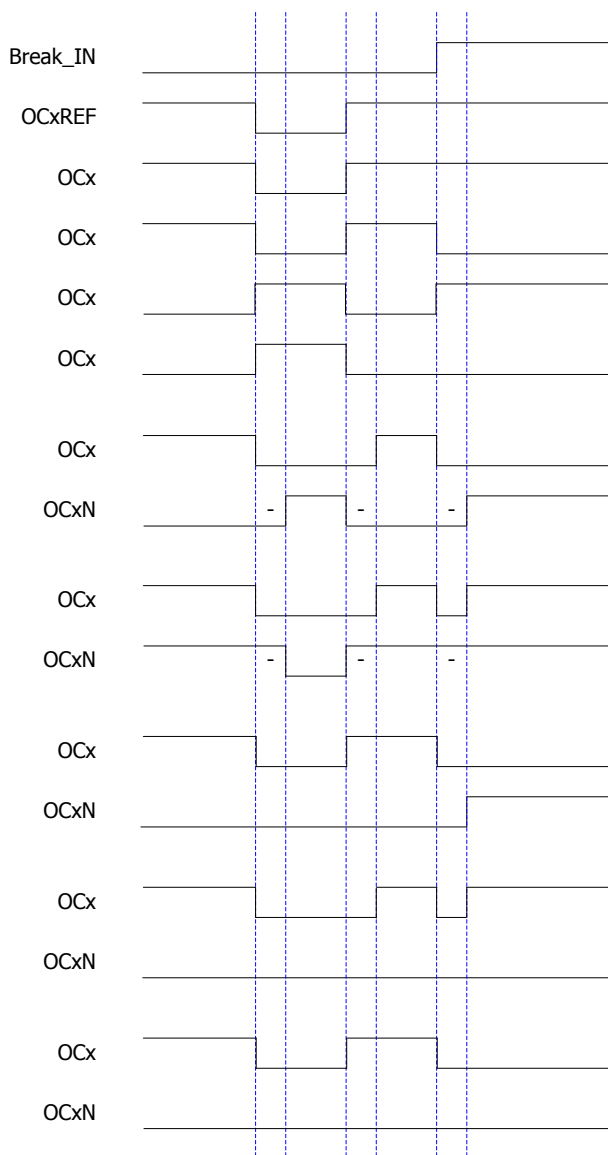
在紧急停止功能中，OCx 以及 OCxN 的电平是由 TM1_BKDT 寄存器中的 POE、IOS 以及 ROS 位、

TM1_CTRL2 寄存器中的 ISOx、ISOxN 位决定的，但绝对不能同时置为生效电平。紧急停止信号源是紧急停止输入脚，或 RCCU 模块中由时钟监视器 (CKM) 产生的 HSE 阻塞事件。紧急停止功能由 TM1_BKDT 寄存器中的 BRKE 位来使能。紧急停止输入信号的极性是由 TM1_BKDT 寄存器中的 BRKP 位设置的。

当紧急停止事件发生时，POE 位将被异步清零。一旦 POE 为 0，输出管脚 OCx 和 OCxN 将按照 TM1_CR2 寄存器中 ISOx 位的设置进行驱动。若 IOS=0，那么定时器释放使能输出，否则使能输出保持为高电平。互补输出首先置为复位状态，之后将重新激活死区时间发生器，在死区时间之后按 ISOx 和 ISOxN 位域的设置驱动输出电平。

紧急停动事件发生后，TM1_STAT 寄存器中的 BKIF 将置位。若 BKIE=1，则会产生一个中断。若 BKDE=1，则会发送一个 DMA 请求。

图 47 发生紧急停止事件时的输出表现

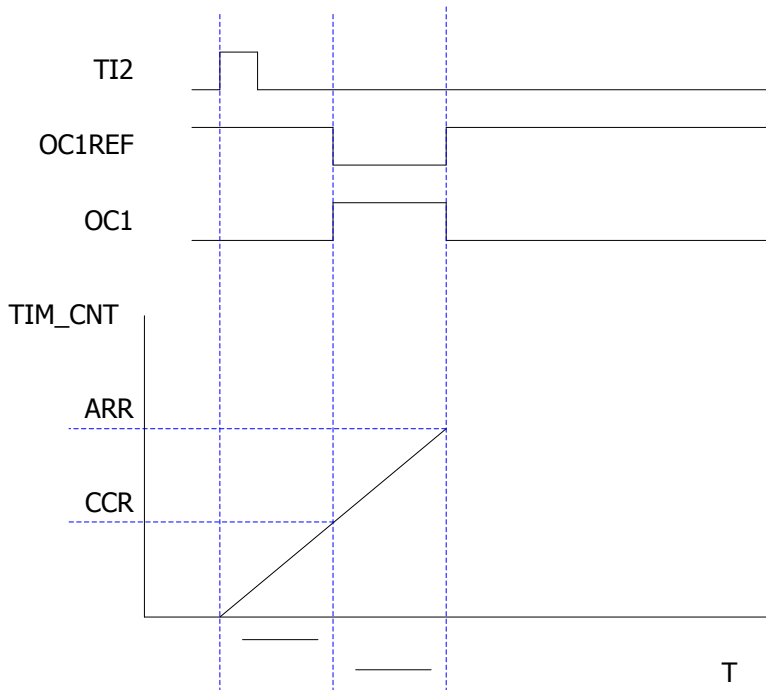


单脉冲模式

一旦定时器设置为工作在单脉冲模式下,不需要刻意将 `TM1_CTRL1` 寄存器的定时器使能位 `CNTE` 置为 1 来使能计数器。产生脉冲的触发来源多样,可以源自触发信号边沿,或通过软件将 `CNTE` 置位。将 `CNTE` 置位或来自触发信号边沿,都可以产生一个脉冲,之后 `CNTE` 位即保持在高电平状态,直到发生更新事件或被软件写 0。假如 `CNTE` 位被软件清零,则计数器将停止计数,其值保持不变。假如 `CNTE` 位被硬件的更新事件自动清零,那么计数器将会重新初始化。

在单脉冲模式下,将 `CNTE` 置为 1 的触发有效沿能够自动使能计数器。不过,由于要将计数值与 `TM1_CHCCx` 寄存器的内容进行比较,这个过程会存在多个时钟的延迟。为尽量减少这个延迟,用户可将 `TM1_CHCTRL1` 寄存器的 `CHxOEF` 置位。在单脉冲模式下,当触发信号产生上升沿后,`OCxREF` 信号将立即被强制拉到发生比较匹配事件而不考虑比较结果的状态。仅当输出通道配置为工作在 `PWM1` 或 `PWM2` 输出模式下,并且触发来源为触发信号时,`CHxOEF` 位才可用。

图 48 单脉冲模式



正交解码器

正交解码器功能采用分别来自 `TM1_CH1` 以及 `TM1_CH2` 管脚的两个正交输入信号 `TI1` 和 `TI2`,根据二者互相关联产生计数值。`DIR` 位在每个输入源信号的跳变沿由硬件自动改变。通过将 `SMC` 位域设为 `0x1`、`0x2`、`0x3`,可选择输入源是仅 `TI1`、仅 `TI2`,还是 `TI1` 和 `TI2`。计数方向的改变机制如下表所示。正交解码器可以当做带有方向选择的外部时钟,这就意味着计数器将会在间隔为

0 到计数器重载值之间持续计数。因此，用户必须在计数器开始计数之前完成 TM1_CARL 寄存器的配置。

表 37 编码器信号以及计数方向

计数模式	电平	TI1FP1		TI2FP2	
		上升沿	下降沿	上升沿	下降沿
仅 TI1 计数	TI2FP = 高	向下	向上	-	-
	TI2FP = 低	向上	向下	-	-
仅 TI2 计数	TI1FP = 高	-	-	向上	向下
	TI1FP = 低	-	-	向下	向上
TI1 及 TI2 计数	TI2FP = 高	向下	向上	X	X
	TI2FP = 低	向上	向下	X	X
	TI1FP = 高	X	X	向上	向下
	TI1FP = 低	X	X	向下	向上

请注意：“-”代表不计数；“X”代表不可能出现的情况。

图 49 编码器接口模式下的计数器运行情况

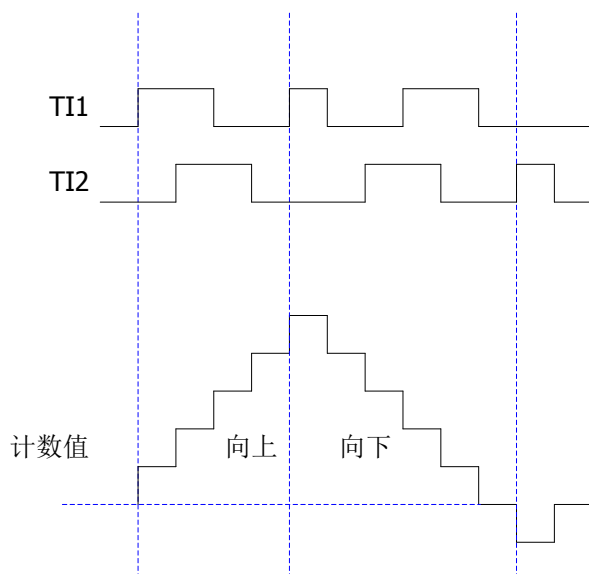
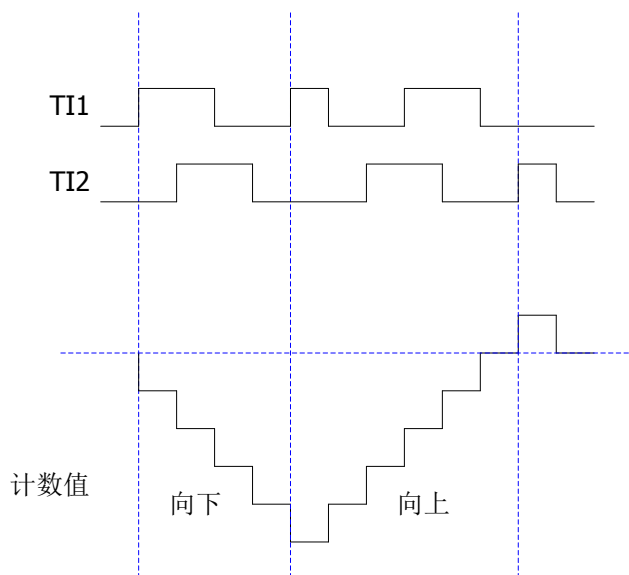


图 50 TF1FP1 极性反转后，编码器接口模式下的计数器运行情况



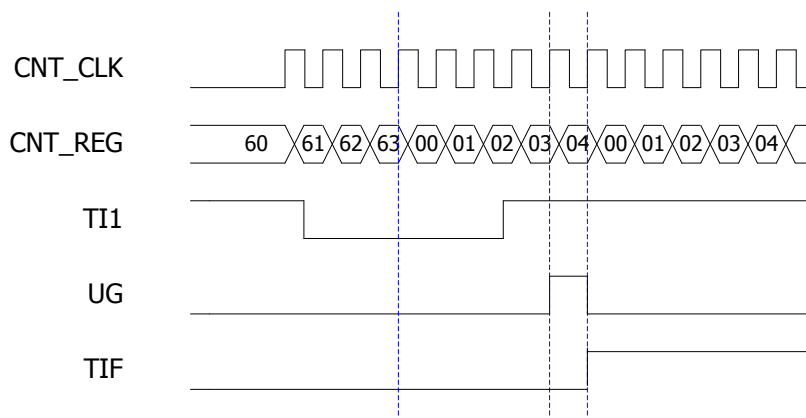
从机控制器

TM1 可以以多种模式与外部触发信号同步，包括重启模式、暂停模式以及触发模式，可通过 TM1_SMC 寄存器中的 SMC 位域进行选择。这些模式的触发输入均来自 STI 信号，可通过 TM1_SMC 寄存器中的 TRGS 位域进行选择。下面我们将详细介绍从机控制器的运行模式。

■ 重启模式

计数器及其预分频器可以 STI 信号的上升沿被重新初始化。当检测到 STI 上升沿时，更新事件软件产生位 UPG 将被硬件自动置位，于是触发事件标志也会被置位。之后计数器和预分频器将会重新初始化。尽管 UPG 位被硬件置位，但更新事件是否发生还取决于计数器更新事件禁用控制位 UPDIS 的设置。若 UPDIS=1（禁止产生更新事件），那么就不会产生更新事件，不过计数器和预分频器仍然会在检测到 STI 上升沿时重新初始化。若 UPDIS=0（使能产生更新事件），那么更新事件将会与 STI 上升沿一道产生，之后所有预装载寄存器都将被更新。

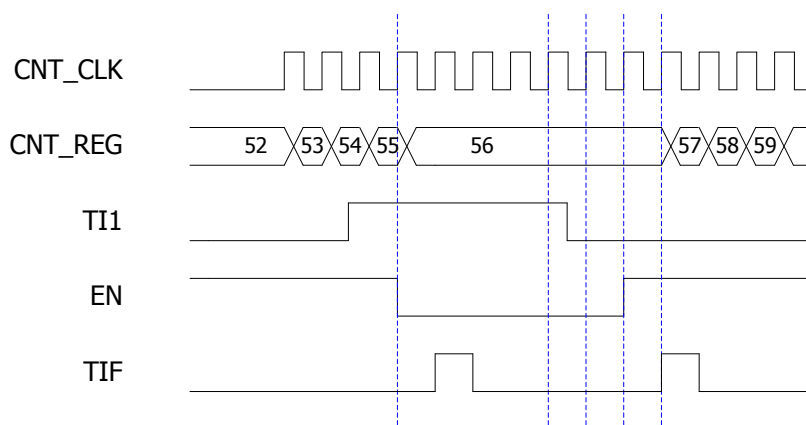
图 51 重启模式下的控制电路



■ 暂停模式

在暂停模式下，选中的 **STI** 输入信号电平被当做计数器开始运行/停止运行的控制信号。当选中的 **STI** 输入信号为高电平时，计数器开始计数；当选中的 **STI** 输入信号为低电平时，计数器停止计数（计数器保持其当前值，不会复位）。

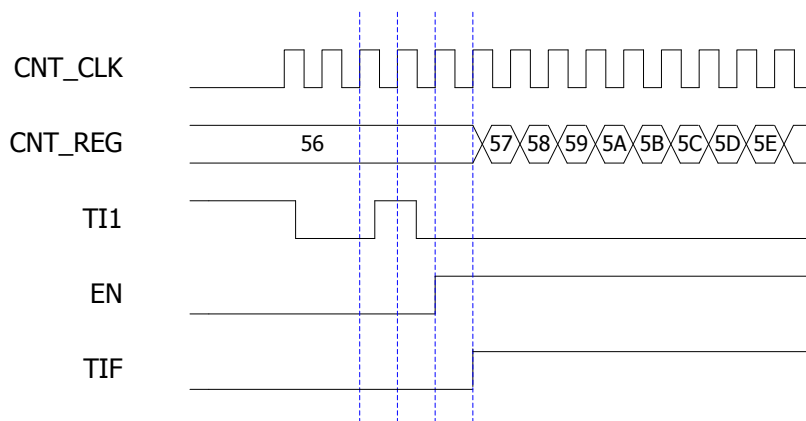
图 52 暂停模式下的控制电路



■ 触发模式

当计数器被禁用之后，计数器仍然可以在 **STI** 信号的上升沿继续计数。当检测到 **STI** 上升沿时，计数器将从当前值开始继续计数。请注意：**STI** 信号仅用于触发计数器使之继续计数，无法控制计数器的停止计数。

图 53 触发模式下的控制电路

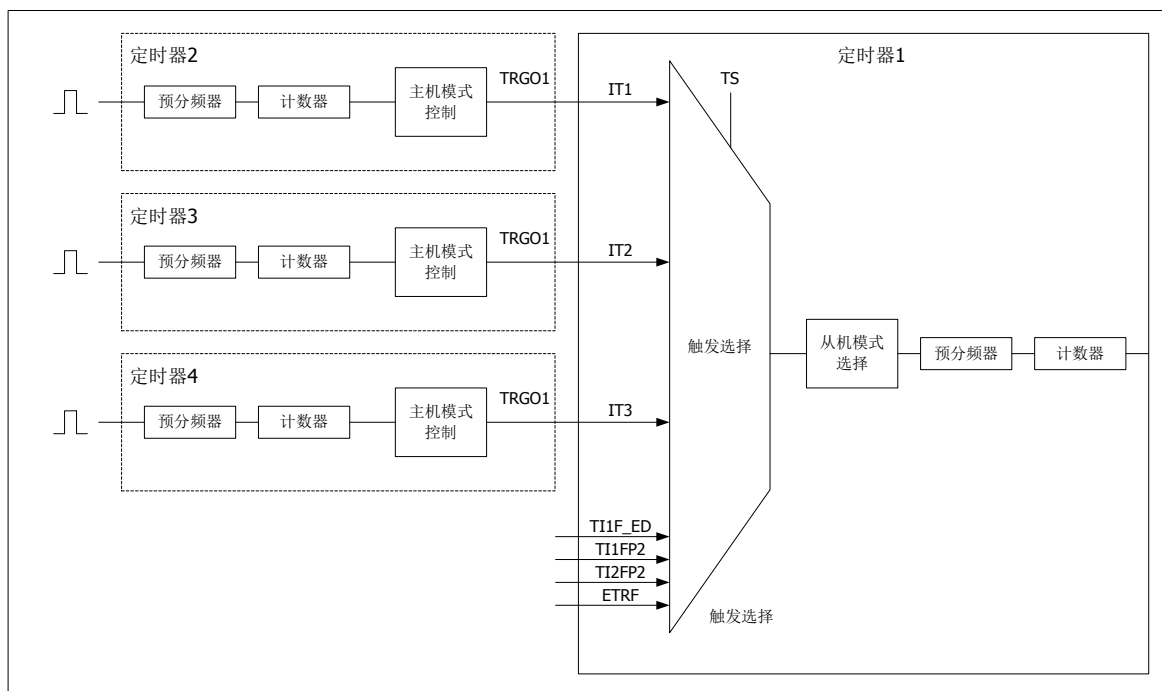


定时器互联

定时器可以内部互联形成定时器链或者实现同步。要实现这种工作方式，应将一个定时器配置为工作于主机模式下，同时将另一个定时器配置为工作于从机模式下。下面给出几种分别处于主机模式和从机模式的示例。

下图演示出定时器 1 配置为从机模式时的触发选择。

图 54 定时器 1 主机/从机模式定时器示例



其它互联示例：

■ 定时器 2 作为定时器 1 的预分频器

可以将定时器 2 配置为定时器 1 的预分频器，其连接情况如下图所示。按下述流程配置：

1. 配置定时器 2 为主机模式，选择其更新事件（UEV）为触发输出（TM2_CR2 寄存器中的 MMS=010）。之后，定时器 2 将在每次计数值溢出时驱动一个周期性的信号。
2. 配置定时器 2 周期（TM2_ARR 寄存器）。
3. 选择定时器 1 的输入触发源为定时器 2（TM1_SMCR 寄存器中的 TS=001）。
4. 配置定时器 1 为外部时钟模式 1（TM1_SMCR 寄存器中的 SMS=111）。
5. 将 CEN（TM1_CR1 寄存器）写 1，启动定时器 1。
6. 将 CEN（TM2_CR1 寄存器）写 1，启动定时器 2。

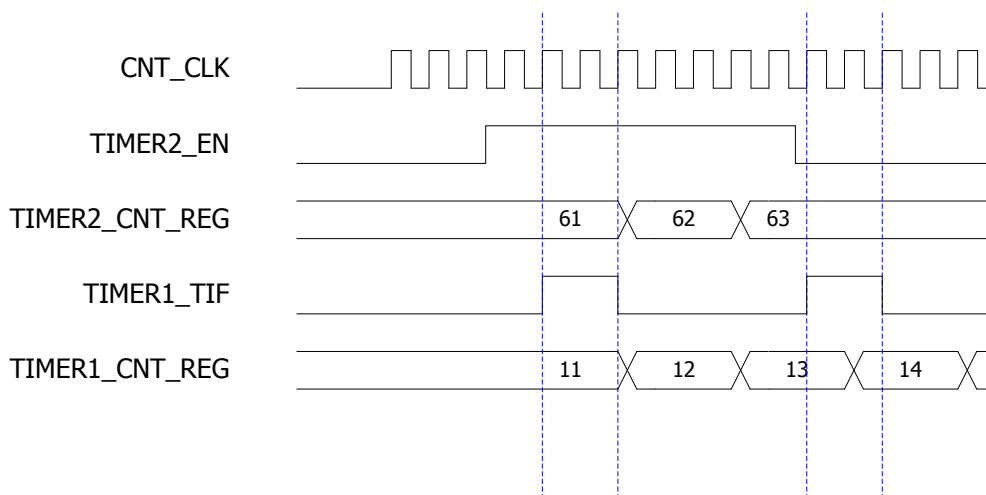
■ 随定时器 2 的使能/更新信号来启动定时器 1

首先，我们使能定时器 1，其使能信号来自定时器 2，如下图所示。定时器 1 经定时器 2 触发使能，从其当前值开始按照获取的内部时钟进行计数。

当定时器 1 收到触发信号后，其 CEN 位将自动置位，计数器开始计数，直到用户关闭定时器 1 为止。两个定时器的计数器时钟频率都是 CK_INT 的 3 分频（ $f_{CK_CNT} = f_{CK_INT}/3$ ）。按下述流程配置：

1. 配置定时器 2 为主机模式，且发送使能信号作为触发输出（TM2_CR2 寄存器中的 MMS=010）。
2. 配置定时器 1 的输入触发源来自定时器 2（TM1_SMCR 寄存器中的 TS=001）。
3. 配置定时器 1 为触发模式（TM1_SMCR 寄存器中的 SMS=110）。
4. 将 CEN（TM2_CR1 寄存器）写 1，启动定时器 2。

图 55 用定时器 2 的使能输出触发定时器 1



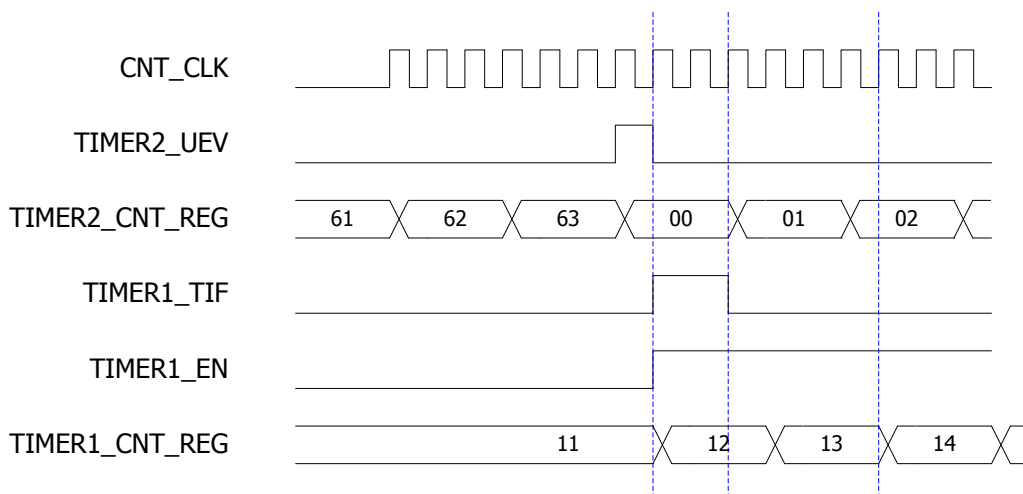
除了使能信号之外，同样也可以用更新事件来作为触发源，如下图所示。按下述流程配置：

1. 配置定时器 2 为主机模式，且发送其更新事件（UEV）作为触发输出（TM2_CR2 寄存器中的

MMS=010)。

2. 配置定时器 2 周期 (TM2_ARR 寄存器)。
3. 配置定时器 1 的输入触发源来自定时器 2 (TM1_SMCR 寄存器中的 TS=001)。
4. 配置定时器 1 为触发模式 (TM1_SMCR 寄存器中的 SMS=110)。
5. 将 CEN 写 1 (TM2_CR1 寄存器), 启动定时器 2。

图 56 用定时器 2 的定时器更新事件触发定时器 1

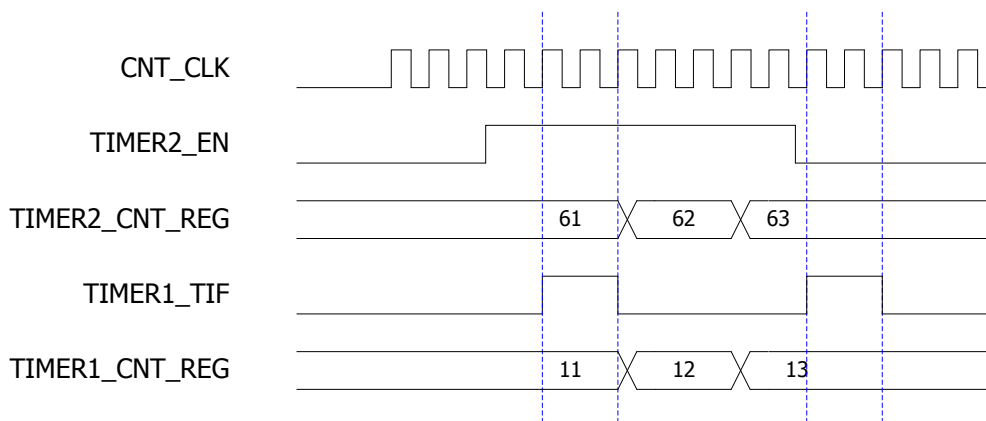


■ 随定时器 2 的使能/OC1 参考信号来使能定时器 1 计数

在这个示例中, 我们用定时器 2 的使能输出控制定时器 1 的使能, 如下图所示。仅当定时器 2 被使能后, 定时器 1 才能按内部时钟计数。两个定时器的计数器时钟频率都是 CK_INT 的 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。按下述流程配置:

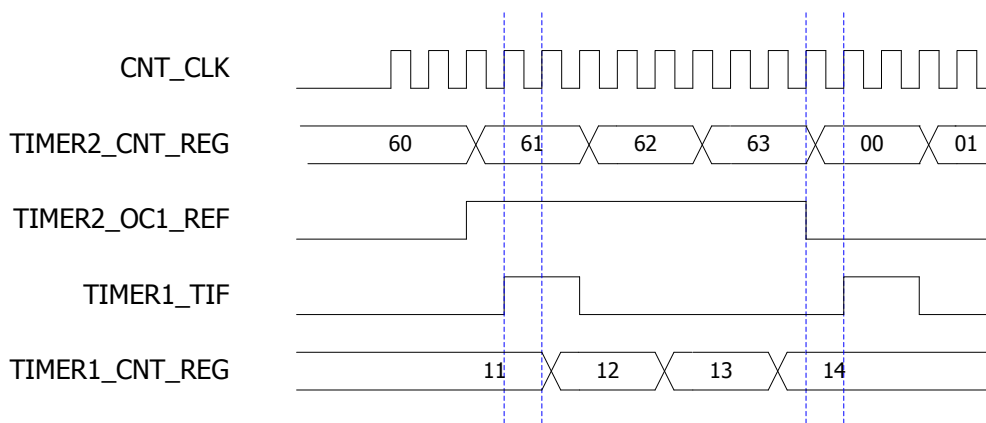
1. 配置定时器 2 为主机模式, 且将其输出使能信号作为触发输出 (TM2_CR2 寄存器中的 MMS=001)。
2. 配置定时器 1 的输入触发源来自定时器 2 (TM1_SMCR 寄存器中的 TS=001)。
3. 配置定时器 1 为边沿触发模式 (TM1_SMCR 寄存器中的 SMS=101)。
4. 将 CEN 写 1 (TM1_CR1 寄存器), 启动定时器 1。
5. 将 CEN 写 1 (TM2_CR1 寄存器), 启动定时器 2。
6. 将 CEN 写 0 (TM2_CR1 寄存器), 停止定时器 2。

图 57 用定时器 2 的使能信号来驱动定时器 1



而在下面的示例中，除了使能信号输出之外，我们也可以使用 `OCx_REF` 信号作为触发源。按下述流程配置：

1. 配置定时器 2 为主机模式，且将其输出比较 1 参考信号 (`OC1REF`) 作为触发输出 (`TM2_CR2` 寄存器中的 `MMS=100`)。
2. 配置定时器 2 的 `OC1REF` 波形 (`TM2_CCMR1` 寄存器)。
3. 配置定时器 1 的输入触发源来自定时器 2 (`TM1_SMCR` 寄存器中的 `TS=001`)。
4. 配置定时器 1 为边沿触发模式 (`TM1_SMCR` 寄存器中的 `SMS=101`)。
5. 将 `CEN` 写 1 (`TM1_CR1` 寄存器)，启动定时器 1。
6. 将 `CEN` 写 1 (`TM2_CR1` 寄存器)，启动定时器 2。

 图 58 用定时器 2 的 `OC1REF` 信号来驱动定时器 1


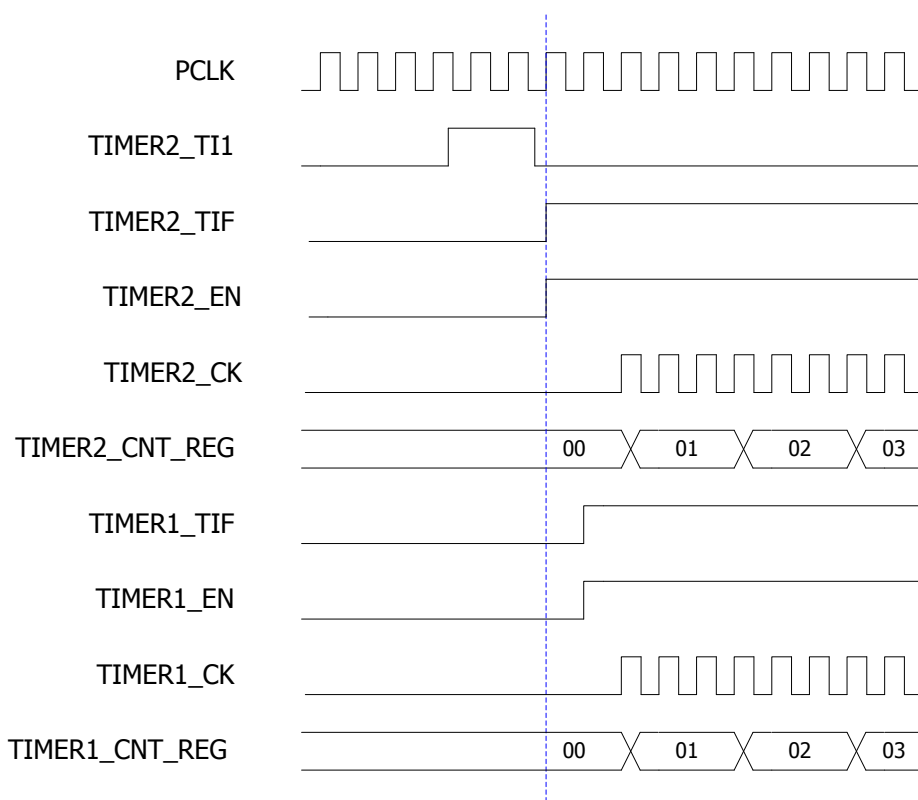
■ 使用外部触发信号来同步启动两个定时器

我们将定时器 1 配置为按定时器 2 的使能输出信号来触发启动，并将定时器 2 配置为根据其 `TI1` 输入信号的上升沿触发。为确保两个定时器同步工作，定时器 2 必须配置为主机/从机模式。按下述流程配置：

1. 配置定时器 2 为从机模式，其输入触发源来自 TI1（TM2_SMCR 寄存器中的 TS=100）。
2. 配置定时器 2 为触发模式（TM2_SMCR 寄存器中的 SMS=110）。
3. 配置定时器 2 为主机/从机模式（TM2_SMCR 寄存器中的 MSM=1）。
4. 配置定时器 1 的输入触发源来自定时器 2（TM1_SMCR 寄存器中的 TS=001）。
5. 配置定时器 1 为触发模式（TM1_SMCR 寄存器中的 SMS=110）。

当定时器 2 的 TI1 输出上升沿时，两个定时器将按照内部时钟同步开始计数，并且两个定时器的 TIF 标志都会置位。

图 59 用定时器 1 的 TI1 信号来触发定时器 1 和定时器 2



■ 定时器调试模式

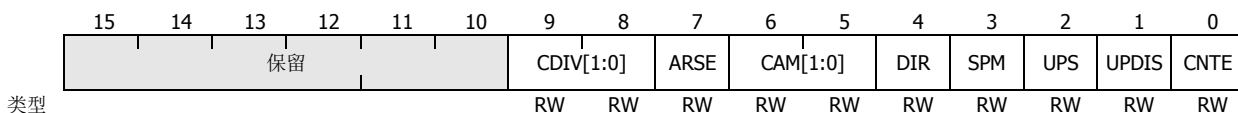
当 Cortex™-M3 内核暂停时，若 DBGMCU 模块中的 DBG_TMx_STOP 设置为 1，则 TMx 定时器将停止计数。

2 TM1 寄存器

2.1 TM1 控制寄存器 1 (TM1_CTRL1)

地址偏移: 0x00

复位值 : 0x0000



位/位域	名称	描述
15:10	保留	请保持其复位状态
9:8	CDIV	时钟分频系数 CDIV 位域由软件配置, 用于指定定时器时钟 (CK_INT) 与死区及采样时钟 (DTS) 之间的分频系数。DTS 时钟用于死区时间发生器以及数字滤波器。 00: $f_{DTS} = f_{CK_INT}$ 01: $f_{DTS} = f_{CK_INT}/2$ 10: $f_{DTS} = f_{CK_INT}/4$ 11: 保留
7	ARSE	自动重载影子寄存器使能 0: 禁用 TMx_ARLD 寄存器的影子寄存器 1: 使能 TMx_ARLD 寄存器的影子寄存器
6:5	CAM	中心对齐模式选择 00: 非中心对齐模式。计数器的计数方向由 DIR 位指定 01: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置), 仅当计数器向下计数时才置位 10: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置), 仅当计数器向上计数时才置位 11: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置), 当计数器向上或者向下计数时都会置位 当从边沿对齐模式切换到中心对齐模式时, 应当禁用计数器。
4	DIR	计数方向 0: 向上计数 1: 向下计数 当定时器配置为中心对齐模式或编码器模式时, 此位只可读。
3	SPM	单脉冲模式 0: 当产生计数器更新事件后, 计数器仍继续计数 1: 当产生计数器更新事件后, 硬件清零 CEN 位, 计数器停止计数
2	UPS	更新来源 本位用于设置更新事件的来源。 0: 下述任何事件都可以触发产生更新中断以及 DMA 请求: <ul style="list-style-type: none"> - UPG 置位 - 计数器产生溢出事件或下溢事件 - 从机模式控制器产生了更新事件 1: 仅计数器溢出/下溢事件才会触发产生更新中断以及 DMA 请求

位/位域	名称	描述
1	UPDIS	更新禁止 本位用于使能或禁用更新事件的产生。 0: 使能更新事件。当产生下述任何事件时，都会产生更新事件，并且缓冲的寄存器会装载其重载值： - UPG 置位 - 计数器产生溢出事件或下溢事件 - 从机模式控制器产生了更新事件 1: 禁用计数器更新事件。假如 UG 置位或者从模式控制器产生硬件复位事件，则缓冲的寄存器保持其当前值，计数器与预分频器重新初始化，
0	CNTE	计数器使能 0: 禁用计数器 1: 使能计数器 当定时器 1 工作于外部时钟，边沿触发模式以及编码器模式下，CEN 位必须由软件置位。在触发模式下，硬件会自动将 CNTE 置位。

2.2 TM1 控制寄存器 2 (TM1_CTRL2)

地址偏移: 0x04

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
类型	保留	ISO4	ISO3N	ISO3	OIS2N	ISO2	ISO1N	ISO1	TI1S	MMC[2:0]		DMAS	CCUC	保留	保留	CCSE
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW

位/位域	名称	描述
15	保留	请保持其复位状态
14	ISO4	通道 4 输出的默认状态 参见 ISO1 的说明。
13	ISO3N	通道 3 互补输出的默认状态 参见 ISO1N 的说明。
12	ISO3	通道 3 输出的默认状态 参见 ISO1 的说明。
11	ISO2N	通道 2 互补输出的默认状态 参见 ISO1N 的说明。
10	ISO2	通道 2 输出的默认状态 参见 ISO1 的说明。
9	ISO1N	通道 1 互补输出的默认状态 0: 当 POE 位复位时，OC1N 置为低电平 1: 当 POE 位复位时，OC1N 置为高电平 仅当 TMx_BKDT 寄存器的 LK[1:0]位域为 00 时，本位才可修改。
8	ISO1	通道 1 输出的默认状态 0: 当 POE 位复位时，OC1 置为低电平 1: 当 POE 位复位时，OC1 置为高电平 假如有 OC1N 互补输出，则 OC1 输出在死区时间后翻转。仅当 TMx_BKDT 寄存器的 LK[1:0]位域为 00 时，本位才可修改。

位/位域	名称	描述
7	TI1S	通道 1 触发输入选择 0: 选择 TMx_CH1 管脚输入信号作为通道 1 的触发输入 1: 选择 TMx_CH1、CH2、CH3 的异或结果作为通道 1 的触发输入
6:4	MMC	主模式控制 本位域用于选择在主模式下发送给从定时器的 TRGO 信号，以便实现特定的同步功能。 000: 复位。当 TMx_EVTG 寄存器的 UG 置位时，或由从模式控制器产生复位时，TRGO 脉冲将会产生。在后一种情况下，TRGO 信号相较于实际的复位信号存在一定延迟。 001: 使能。若需要同时启动多个定时器，或者要实现在一定的时间窗口内启动从定时器，那么这种模式将非常有用。在这种模式下，主模式控制器将选择计数器使能信号 CNT_EN 作为 TRGO。当 CNTE 控制位被置位，或当边沿触发模式下的触发输入为高电平时，计数器使能信号将会置位。边沿触发模式下的触发器输入信号与 TRGO 输出之间存在一定的延迟，除非选择了主-从模式。 010: 更新。在这种模式下，主模式控制器以计数器更新事件作为 TRGO。 011: 捕获/比较。在这种模式下，主模式控制器在产生捕获事件或者比较匹配事件后产生 TRGO 脉冲。 100: 比较。在这种模式下，主模式控制器选择 OC1REF 信号作为 TRGO。 101: 比较。在这种模式下，主模式控制器选择 OC2REF 信号作为 TRGO。 110: 比较。在这种模式下，主模式控制器选择 OC3REF 信号作为 TRGO。 111: 比较。在这种模式下，主模式控制器选择 OC4REF 信号作为 TRGO。
3	DMAS	DMA 请求来源选择 0: 当通道 x 事件产生时，发送通道 x 的 DMA 请求 1: 当更新事件产生时，发送通道 x 的 DMA 请求
2	CCUC	捕获/比较控制影子寄存器更新控制 当捕获/比较控制影子寄存器（对 CHxE、OCxNE 以及 CMPxM 位域）被使能（CCSE=1），本位控制这些影子寄存器何时被更新。 0: 当 CCUG 置位时，更新影子寄存器 1: 当 CCUG 置位、或者检测到 TRGI 上升沿时，更新影子寄存器 如果通道不具有互补输出，那么这个位无意义。
1	保留	请保持其复位状态
0	CCSE	捕获/比较控制影子寄存器使能 0: 禁用 CHxE、OCxNE、CMPxM 位域的影子寄存器 1: 使能 CHxE、OCxNE、CMPxM 位域的影子寄存器。这些寄存器写入后，仅当 CCUG 置位后这些寄存器才会更新内容。 如果通道并未具有互补输出，那么这个位无意义。

2.3 TM1 从模式控制寄存器 (TIM1_SMC)

地址偏移: 0x08

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETPL	ECM2E	ETPSC[1:0]		ETFC[3:0]			MSM	TRGS[2:0]		保留	SMC[2:0]				
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域	名称	描述
------	----	----

位/位域	名称	描述
15	ETPL	外部触发信号极性 本位用于指定 ETR 信号的极性。 0: ETR 信号为高电平有效或者上升沿有效 1: ETR 信号为低电平有效或者下降沿有效
14	ECM2E	外部时钟模式 2 使能 在外部时钟模式 2 下，计数器由 ETRF 信号的任一有效跳变沿触发。 0: 禁用外部时钟模式 2 1: 使能外部时钟模式 2 将 ECM2E 置位，与选择外部时钟模式 1 且 TRGF 连接到 ETRF (SMC=111 且 TRGS=111) 具有相同的效果。 在复位模式、边沿驱动模式或触发模式下，可以同时使用外部时钟模式 2。不过此时 TRGS 位域绝对不能为 111。 假如外部时钟模式 1 与外部时钟模式 2 同时被使能，则外部时钟输入将是 ETRF。
13:12	ETPSC	外部触发预分频器 外部触发信号 ETRP 的频率绝对不得超过 TMx_CLK 频率的 1/4。如果外部触发信号是某种高速时钟，那么需要借助预分频器将 ETRP 信号降频。 00: 禁用预分频器 01: ETRP 信号将被 2 分频 10: ETRP 信号将被 4 分频 11: ETRP 信号将被 8 分频
11:8	ETFC	外部触发滤波控制 数字滤波器中用到了事件计数器，每发生 N 个输入事件后才会改变输出。本位域指定用于采样 ETRP 信号的频率以及用在 ETRP 上的数字滤波器长度。 0000: 禁用数字滤波。 $f_{SAMP} = f_{CK_INT}$, $N = 1$ 0001: $f_{SAMP} = f_{CK_INT}$, $N = 2$ 0010: $f_{SAMP} = f_{CK_INT}$, $N = 4$ 0011: $f_{SAMP} = f_{CK_INT}$, $N = 8$ 0100: $f_{SAMP} = f_{DTS}/2$, $N = 6$ 0101: $f_{SAMP} = f_{DTS}/2$, $N = 8$ 0110: $f_{SAMP} = f_{DTS}/4$, $N = 6$ 0111: $f_{SAMP} = f_{DTS}/4$, $N = 8$ 1000: $f_{SAMP} = f_{DTS}/8$, $N = 6$ 1001: $f_{SAMP} = f_{DTS}/8$, $N = 8$ 1010: $f_{SAMP} = f_{DTS}/16$, $N = 5$ 1011: $f_{SAMP} = f_{DTS}/16$, $N = 6$ 1100: $f_{SAMP} = f_{DTS}/16$, $N = 8$ 1101: $f_{SAMP} = f_{DTS}/32$, $N = 5$ 1110: $f_{SAMP} = f_{DTS}/32$, $N = 6$ 1111: $f_{SAMP} = f_{DTS}/32$, $N = 8$
7	MSM	主-从模式 在这种模式下，触发输入事件会延迟一点儿才生效，以实现当前定时器通过 TRGO 与其从定时器之间的完美同步。假如需要将多个定时器同步到一个外部事件上，那么可以使用这种模式。 0: 禁用主-从模式 1: 使能主-从模式

位/位域	名称	描述
6:4	TRGS	触发源选择 本位域指定哪种信号选作同步计数器的触发输入。 000: 内部触发输入 0 (ITR0) 保留 001: 内部触发输入 1 (ITR1) TM2 010: 内部触发输入 2 (ITR2) TM3 011: 内部触发输入 3 (ITR3) TM4 100: TI1 边沿标志 (TI1F_ED) 101: 经过滤波的通道 1 输入 (TI1FP1) 110: 经过滤波的通道 2 输入 (TI2FP2) 111: 外部触发输入 (ETRF) 若使能从模式, 则本位域绝对不得修改
3	保留	请保持其复位状态
2:0	SMC	从模式控制 000: 禁用模式。当 CNTE 置位后, 预分频器直接由内部时钟供给。 001: 正交解码模式 1。计数器按照 TI2FP2 的边沿计数, 计数方向取决于 TI1FP1 的电平。 010: 正交解码模式 2。计数器按照 TI1FP1 的边沿计数, 计数方向取决于 TI2FP2 的电平。 011: 正交解码模式 3。计数器按照 TI1FP1 和 TI2FP2 的边沿计数, 计数方向则取决于彼此。 100: 重启模式。当选定的触发输入信号出现上升沿时, 计数器将重新初始化, 所有影子寄存器将更新。 101: 暂停模式。触发输入为高电平时, 使能计数器时钟; 触发输入为低电平时, 禁用计数器时钟。 110: 触发模式。触发输入信号的上升沿将使能计数器。此时计数器是不能被从模式控制器禁用的。 111: 外部时钟模式 1。计数器按照指定的触发输入信号的上升沿计数。 由于每当 TI1F 跳变时 TI1F_ED 都会输出 1 个脉冲, 并且边沿驱动模式会检查触发信号的电平状态, 因此当 TI1F_ED 被选作触发输入信号时, 不得使用边沿驱动模式。

2.4 TM1 DMA 及中断使能寄存器 (TM1_DIE)

地址偏移: 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDE	CCUDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
15	保留	请保持其复位状态
14	TRGDE	触发中断使能 0: 禁用触发中断 1: 使能触发中断
13	CCUDE	通道控制计数器更新 DMA 请求使能 0: 禁用通道控制计数器更新 DMA 请求 1: 使能通道控制计数器更新 DMA 请求

位/位域	名称	描述
12	CH4DE	通道 4 DMA 请求使能 0: 禁用通道 4 DMA 请求 1: 使能通道 4 DMA 请求
11	CH3DE	通道 3 DMA 请求使能 0: 禁用通道 3 DMA 请求 1: 使能通道 3 DMA 请求
10	CH2DE	通道 2 DMA 请求使能 0: 禁用通道 2 DMA 请求 1: 使能通道 2 DMA 请求
9	CH1DE	通道 1 DMA 请求使能 0: 禁用通道 1 DMA 请求 1: 使能通道 1 DMA 请求
8	UPDE	更新 DMA 请求使能 0: 禁用更新 DMA 请求 1: 使能更新 DMA 请求
7	BKIE	紧急停止中断使能 0: 禁用紧急停止中断 1: 使能紧急停止中断
6	TRGIE	触发中断使能 0: 禁用触发中断 1: 使能触发中断
5	CCUIE	通道控制计数器更新中断使能 0: 禁用通道控制计数器更新中断 1: 使能通道控制计数器更新中断
4	CH4IE	通道 4 中断使能 0: 禁用通道 4 中断 1: 使能通道 4 中断
3	CH3IE	通道 3 中断使能 0: 禁用通道 3 中断 1: 使能通道 3 中断
2	CH2IE	通道 2 中断使能 0: 禁用通道 2 中断 1: 使能通道 2 中断
1	CH1IE	通道 1 中断使能 0: 禁用通道 1 中断 1: 使能通道 1 中断
0	UPIE	更新中断使能 0: 禁用更新中断 1: 使能更新中断

2.5 TM1 状态寄存器 (TM1_SR)

地址偏移: 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH4OF	CH3OF	CH2OF	CH1OF	保留	BKIF	TRGIF	CCUIF	CH4IF	CH3IF	CH2IF	CH1IF	UPIF
			RC_W0	RC_W0	RC_W0	RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

类型

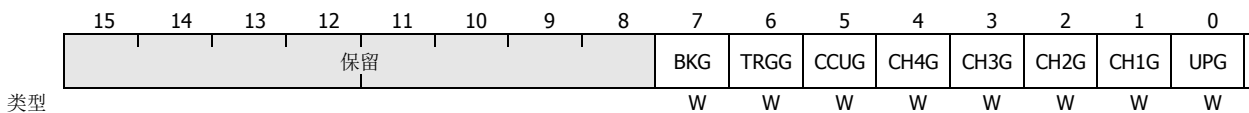
位/位域	名称	描述
15:13	保留	请保持其复位状态
12	CH4OF	通道 4 过捕获标志 参见 CH1OF 的描述。
11	CH3OF	通道 3 过捕获标志 参见 CH1OF 的描述。
10	CH2OF	通道 2 过捕获标志 参见 CH1OF 的描述。
9	CH1OF	通道 1 过捕获标志 若通道 1 配置为输入模式, 当发生捕获事件且 CH1IF 标志已经置位的情况下, 本标志将由硬件置位。本标志由软件清零。 0: 未发生过捕获中断 1: 发生过捕获中断
8	保留	请保持其复位状态
7	BKIF	紧急停止中断标志 当紧急停止输入信号变为有效, 本标志将由硬件置位。假如紧急停止输入信号不再有效, 应由软件清零。 0: 紧急停止输入信号未检测到有效电平 1: 紧急停止输入信号检测到有效电平
6	TRGIF	触发输入中断 当触发事件产生时, 本标志将由硬件置位。本标志由软件清零。在除边沿驱动之外的所有模式下, 当从模式控制器使能后, TRGI 输入的有效跳变沿将产生一个触发事件。在边沿驱动模式下, 当从模式控制器使能后, TRGI 输入的任何跳变沿都将产生触发事件。 0: 未发生触发事件 1: 已产生触发中断
5	CCUIF	通道控制计数器更新中断标志 当通道控制计数器更新事件产生时, 本标志将由硬件置位。本标志可由软件清零。 0: 未发生通道控制计数器更新中断 1: 已发生通道控制计数器更新中断
4	CH4IE	通道 4 中断使能 参见 CH1IE 的描述。
3	CH3IE	通道 3 中断使能 参见 CH1IE 的描述。
2	CH2IE	通道 2 中断使能 参见 CH1IE 的描述。

位/位域	名称	描述
1	CH1IE	通道 1 中断使能 本标志由硬件置位，由软件清零。若通道 1 处于输入模式，则当产生捕获时间时，本标志将置位。若通道 1 处于输出模式，则当产生比较匹配事件时，本标志将置位。 0: 未产生通道 1 中断 1: 已产生通道 1 中断
0	UPIF	更新中断标志 当发生更新事件时，本标志将由硬件置位。本标志由软件清零。 0: 未发生更新中断 1: 已发生更新中断

2.6 TM1 事件产生寄存器 (TM1_EVG)

地址偏移: 0x14

复位值 : 0x0000



位/位域	名称	描述
15:8	保留	请保持其复位状态
7	BKG	紧急停止事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，POE 将清零，BKIF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。 0: 不产生紧急停止事件 1: 产生紧急停止事件
6	TRGG	触发事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，TMx_STA 寄存器的 TRGIF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。 0: 不产生触发事件 1: 产生触发事件
5	CCUG	通道控制更新事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，具有互补输出的通道的通道控制寄存器 (CHxE、OCxNE 以及 CMPxM 位域) 将会得到更新。 0: 不产生控制更新事件 1: 产生控制更新事件
4	CH4G	通道 4 捕获/比较事件产生 参见 CH1G 的描述。
3	CH3G	通道 3 捕获/比较事件产生 参见 CH1G 的描述。
2	CH2G	通道 2 捕获/比较事件产生 参见 CH1G 的描述。

位/位域	名称	描述
1	CH1G	通道 1 捕获/比较事件产生 本位由软件置位，用以在通道 1 产生一个捕获/比较事件；本位由硬件自动清零。当对本位写入 1 时，CC1IF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。此外，假如通道 1 配置为输入模式，则计数器的当前值将被捕获到 TM1_CHCC1 寄存器中。若 CH1IF 标志已经置位，还会将 CH1OF 标志置位。 0: 不产生通道 1 捕获/比较事件 1: 产生通道 1 捕获/比较事件
0	UPG	更新事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，若选择的工作模式是中心对齐模式或向上计数模式，则计数器将被清零；否则（向下计数模式）计数器将会装载重载值。同时，预分频计数器也会被清零。 0: 不产生更新事件 1: 产生更新事件

2.7 TM1 通道控制寄存器 1 (TM1_CHCTRL1)

地址偏移: 0x18

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2 OCE	CH2OM[2:0]			CH2 OSE	CH2 OFE	CH2M[1:0]		CH1 OCE	CH1OM[2:0]			OC1 OSE	CH1 OFE	CH1M[1:0]	
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

输出比较模式:

位/位域	名称	描述
15	CH2OCE	通道 2 输出比较清除使能 参见 CH1OCE 的描述。
14:12	CH2OM	通道 2 输出比较模式 参见 CH1OM 的描述。
11	CH2OSE	通道 2 输出比较影子寄存器使能 参见 CH1OSE 的描述。
10	CH2OFE	通道 2 输出比较快速使能 参见 CH1OFE 的描述。
9:8	CH2M	通道 2 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH2E 位为 0）时，本位域才可写。 00: 通道 2 配置为输出 01: 通道 2 配置为输入，IC2 映射到 TI2 10: 通道 2 配置为输入，IC2 映射到 TI1 11: 通道 2 配置为输入，IC2 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

位/位域	名称	描述
7	CH1OCE	<p>通道 1 输出比较清除使能</p> <p>当本位置位时，当 ETRF 输入检测到高电平时将会清除 OC1REF 信号。</p> <p>0: 禁用通道 1 输出比较清除</p> <p>1: 使能通道 1 输出比较清除</p>
6:4	CH1OM	<p>通道 1 输出比较模式</p> <p>本位域指定输出用于驱动 OC1 和 OC1N 的参考信号 OC1REF 的行为方式。OC1REF 是高电平有效的，而 OC1 和 OC1N 的有效电平则取决于 CH1P 和 CH1NP 位域的设置。</p> <p>000: 冻结。OC1REF 信号保持固定电平，不管输出比较寄存器 TM1_CHCC1 以及计数值的比较结果如何。</p> <p>001: 匹配时拉高。当计数值与输出比较寄存器 TM1_CHCC1 匹配时，OC1REF 信号将强制拉高。</p> <p>010: 匹配时拉低。当计数值与输出比较寄存器 TM1_CHCC1 匹配时，OC1REF 信号将强制拉低。</p> <p>011: 匹配时翻转。当计数值与输出比较寄存器 TM1_CHCC1 匹配时，OC1REF 信号将翻转。</p> <p>100: 强制拉低。OC1REF 信号始终为低电平。</p> <p>101: 强制拉高。OC1REF 信号始终为高电平。</p> <p>110: PWM 模式 1。当向上计数时，只要计数值小于 TM1_CHCC1，OC1REF 为高电平，否则为低电平。当向下计数时，只要计数值大于 TM1_CHCC1，OC1REF 为低电平，否则为高电平。</p> <p>111: PWM 模式 2。当向上计数时，只要计数值小于 TM1_CHCC1，OC1REF 为低电平，否则为高电平。当向下计数时，只要计数值大于 TM1_CHCC1，OC1REF 为高电平，否则为低电平。</p> <p>当配置为 PWM 模式时，仅当输出比较模式从“冻结”切换到“PWM”模式，或当比较结果发生变化时，OCREF 电平才会变化。</p> <p>若 TMx_BKDT 寄存器的 LK[1:0]位域为 11、且 CH1M 位域为 00，本位域不得更改。</p>
3	CH1OSE	<p>通道 1 输出比较影子寄存器使能</p> <p>当本位置位时，TM1_CHCC1 寄存器的影子寄存器将会使能，并在每个更新事件时得到更新。</p> <p>0: 禁用通道 1 输出比较影子寄存器</p> <p>1: 使能通道 1 输出比较影子寄存器</p> <p>仅在单脉冲模式下（TM1_CTRL1 寄存器的 OPM 置位），PWM 模式无需使能影子寄存器也可使用。</p> <p>若 TMx_BKDT 寄存器的 LK[1:0]位域为 11、且 CH1M 位域为 00，本位域不得更改。</p>
2	CH1OFE	<p>通道 1 快速比较输出使能</p> <p>当本位置位时，假如通道配置为 PWM1 或 PWM2 模式，则触发输入事件在 CC 输出上的效果会被加速。输出通道会将触发输入信号的有效沿视为比较匹配，OC 将会变成与比较结果无关的电平。</p> <p>0: 禁用通道 1 快速比较输出。触发输入的边沿到 CC1 有效输出的最短延时为 5 个时钟周期。</p> <p>1: 使能通道 1 快速比较输出。触发输入的边沿到 CC1 有效输出的最短延时为 3 个时钟周期。</p>
1:0	CH1M	<p>通道 1 模式选择</p> <p>本位域指定通道的方向，以及选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH1E 位为 0）时，本位域才可写。</p> <p>00: 通道 1 配置为输出</p> <p>01: 通道 1 配置为输入，IC1 映射到 TI1</p> <p>10: 通道 1 配置为输入，IC1 映射到 TI2</p> <p>11: 通道 1 配置为输入，IC1 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。</p>

输入捕获模式:

位/位域	名称	描述
15:12	CH2ICF	通道 2 输入捕获滤波控制 参见 CH1ICF 的描述。
11:10	CH2ICP	通道 2 输入捕获预分频系数 参见 CH1ICP 的描述。
9:8	CH2M	通道 2 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH2E 位为 0）时，本位域才可写。 00: 通道 2 配置为输出 01: 通道 2 配置为输入，IC2 映射到 TI2 10: 通道 2 配置为输入，IC2 映射到 TI1 11: 通道 2 配置为输入，IC2 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。
7:4	CH1ICF	通道 1 输入捕获滤波控制 数字滤波器中采用了一个事件计数器，经过 N 次输入事件后，输出才会产生跳变。本位域用于指定用于采样 TI1 输入信号的频率，以及用于 TI1 的数字滤波器的长度。 0000: 禁用数字滤波。 $f_{SAMP} = f_{CK_INT}$, $N = 1$ 0001: $f_{SAMP} = f_{CK_INT}$, $N = 2$ 0010: $f_{SAMP} = f_{CK_INT}$, $N = 4$ 0011: $f_{SAMP} = f_{CK_INT}$, $N = 8$ 0100: $f_{SAMP} = f_{DTS}/2$, $N = 6$ 0101: $f_{SAMP} = f_{DTS}/2$, $N = 8$ 0110: $f_{SAMP} = f_{DTS}/4$, $N = 6$ 0111: $f_{SAMP} = f_{DTS}/4$, $N = 8$ 1000: $f_{SAMP} = f_{DTS}/8$, $N = 6$ 1001: $f_{SAMP} = f_{DTS}/8$, $N = 8$ 1010: $f_{SAMP} = f_{DTS}/16$, $N = 5$ 1011: $f_{SAMP} = f_{DTS}/16$, $N = 6$ 1100: $f_{SAMP} = f_{DTS}/16$, $N = 8$ 1101: $f_{SAMP} = f_{DTS}/32$, $N = 5$ 1110: $f_{SAMP} = f_{DTS}/32$, $N = 6$ 1111: $f_{SAMP} = f_{DTS}/32$, $N = 8$
3:2	CH1ICP	通道 1 输入捕获预分频系数 本位域指定通道 1 输入的预分频系数。若 TM1_CHE 寄存器的 CH1E 位被清零，则预分频器将会复位。 00: 禁用预分频器，在通道输入信号每个跳变沿时捕获 01: 通道输入信号每 2 个跳变沿时捕获 10: 通道输入信号每 4 个跳变沿时捕获 11: 通道输入信号每 8 个跳变沿时捕获
1:0	CH1M	通道 1 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH1E 位为 0）时，本位域才可写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入，IC1 映射到 TI1 10: 通道 1 配置为输入，IC1 映射到 TI2 11: 通道 1 配置为输入，IC1 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

2.8 TM1 通道控制寄存器 2 (TM1_CHCTRL2)

地址偏移: 0x1C

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH4 OCE	CH4OM[2:0]			CH4 OSE	CH4 OFE	CH4M[1:0]		CH3 OCE	CH3OM[2:0]			OC3 OSE	CH3 OFE	CH3M[1:0]	
类型	CH4ICF[3:0]			CH4ICP[1:0]				CH3ICF[3:0]			CH3ICP[1:0]					
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

输出比较模式:

位/位域	名称	描述
15	CH4OCE	通道 4 输出比较清除使能 参见 CH1OCE 的描述。
14:12	CH4OM	通道 4 输出比较模式 参见 CH1OM 的描述。
11	CH4OSE	通道 4 输出比较影子寄存器使能 参见 CH1OSE 的描述。
10	CH4OFE	通道 4 输出比较快速使能 参见 CH1OFE 的描述。
9:8	CH4M	通道 4 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH4E 位为 0）时，本位域才可写。 00: 通道 4 配置为输出 01: 通道 4 配置为输入，IC4 映射到 TI4 10: 通道 4 配置为输入，IC4 映射到 TI3 11: 通道 4 配置为输入，IC4 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。
7	CH3OCE	通道 3 输出比较清除使能 参见 CH1OCE 的描述。
6:4	CH3OM	通道 3 输出比较模式 参见 CH1OM 的描述。
3	CH3OSE	通道 3 输出比较影子寄存器使能 参见 CH1OSE 的描述。
2	CH3OFE	通道 3 输出比较快速使能 参见 CH1OFE 的描述。
1:0	CH3M	通道 3 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH3E 位为 0）时，本位域才可写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入，IC3 映射到 TI3 10: 通道 3 配置为输入，IC3 映射到 TI4 11: 通道 3 配置为输入，IC3 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

输入捕获模式:

位/位域	名称	描述
------	----	----

位/位域	名称	描述
15:12	CH4ICF	通道 4 输入捕获滤波控制 参见 CH1ICF 的描述。
11:10	CH4ICP	通道 4 输入捕获预分频系数 参见 CH1ICP 的描述。
9:8	CH4M	通道 4 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH4E 位为 0），本位域才可写。 00: 通道 4 配置为输出 01: 通道 4 配置为输入，IC4 映射到 TI4 10: 通道 4 配置为输入，IC4 映射到 TI3 11: 通道 4 配置为输入，IC4 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。
7:4	CH3ICF	通道 3 输入捕获滤波控制 参见 CH1ICF 的描述。
3:2	CH3ICP	通道 3 输入捕获预分频系数 参见 CH1ICP 的描述。
1:0	CH3M	通道 3 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TM1_CHE 寄存器的 CH3E 位为 0）时，本位域才可写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入，IC3 映射到 TI3 10: 通道 3 配置为输入，IC3 映射到 TI4 11: 通道 3 配置为输入，IC3 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

2.9 TM1 通道使能寄存器 (TM1_CHE)

地址偏移: 0x20

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH4P	CH4E	CH3NP	CH3NE	CH3P	CH3E	CH2NP	CH2NE	CH2P	CH2E	CH1NP	CH1NE	CH1P	CH1E	
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

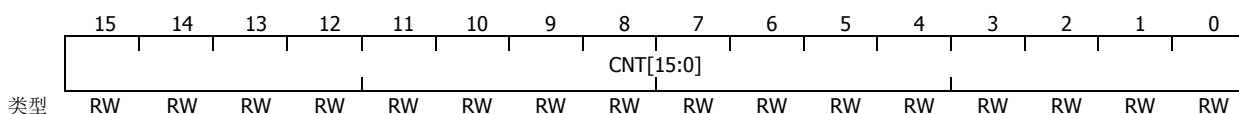
位/位域	名称	描述
15:14	保留	请保持其复位状态
13	CH4P	通道 4 极性 参见 CH1P 的描述。
12	CH4E	通道 4 使能 参见 CH1E 的描述。
11	CH3NP	通道 3 互补输出极性 参见 CH1NP 的描述。

位/位域	名称	描述
10	CH3NE	通道 3 互补输出使能 参见 CH1NE 的描述。
9	CH3P	通道 3 极性 参见 CH1P 的描述。
8	CH3E	通道 3 使能 参见 CH1E 的描述。
7	CH2NP	通道 2 互补输出极性 参见 CH1NP 的描述。
6	CH2NE	通道 2 互补输出使能 参见 CH1NE 的描述。
5	CH2P	通道 2 极性 参见 CH1P 的描述。
4	CH2E	通道 2 使能 参见 CH1E 的描述。
3	CH1NP	通道 1 互补输出极性 当通道 1 配置为输出模式时，本位指定互补输出信号的极性。 0: 通道 1 高电平有效 1: 通道 1 低电平有效 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11 或 10，本位域不得更改。
2	CH1NE	通道 1 互补输出使能 当通道 1 配置为输出模式时，对本位写 1 即可使能通道 1 的互补输出。 0: 禁用通道 1 互补输出 1: 使能通道 1 互补输出
1	CH1P	通道 1 极性 当通道 1 配置为输入模式时，本位指定 IC1 信号的极性。当通道 1 配置为输出模式时，本位指定输出信号的极性。 0: 通道 1 高电平有效 1: 通道 1 低电平有效 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11 或 10，本位域不得更改。
0	CH1E	通道 1 使能 当通道 1 配置为输入模式时，对本位写 1 即可将 OC1 信号置为有效状态。当通道 1 配置为输出模式时，对本位写 1 即可使能通道 1 的捕获事件。 0: 禁用通道 1 1: 使能通道 1

2.10 TM1 计数寄存器 (TM1_CNT)

地址偏移: 0x24

复位值 : 0x0000



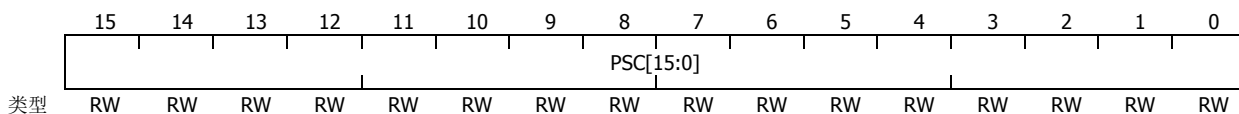
位/位域	名称	描述
------	----	----

位/位域	名称	描述
15:0	CNT	本位域代表当前计数值。对本位域执行写操作，即可更改计数值。

2.11 TM1 预分频寄存器 (TM1_PSC)

地址偏移: 0x28

复位值 : 0x0000

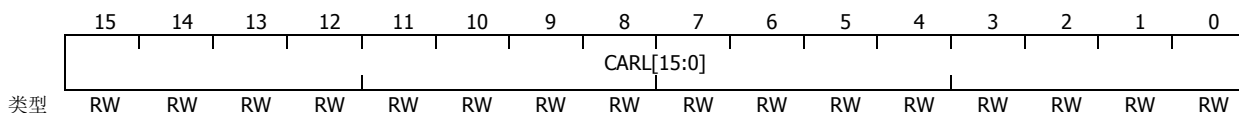


位/位域	名称	描述
15:0	PSC	计数器时钟的预分频系数 PSC 时钟将除以(PSC+1)以获得计数器时钟。本位域的值将在每次产生更新事件时装载到相应的影子寄存器中。

2.12 TM1 计数器自动重载寄存器 (TM1_CARL)

地址偏移: 0x2C

复位值 : 0x0000

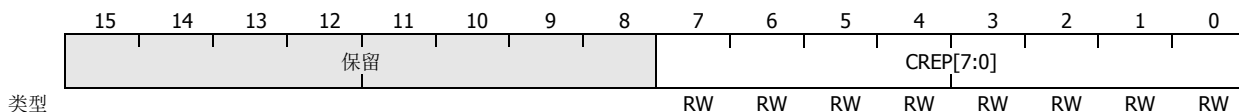


位/位域	名称	描述
15:0	CARL	计数器自动重载值 本位域指定计数器的自动重载值。

2.13 TM1 计数器重复次数寄存器 (TM1_CREP)

地址偏移: 0x30

复位值 : 0x0000



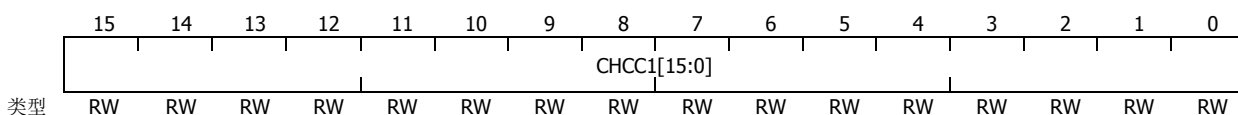
位/位域	名称	描述
------	----	----

位/位域	名称	描述
15:8	保留	请保持其复位状态
7:0	CREP	计数器重复次数 本位域指定计数器更新事件的产生频率。每当重复次数计数器递减到 0 时，就会产生更新事件。若使能影子寄存器，则影子寄存器的更新频率也会受到本位域的影响。

2.14 TM1 通道 1 捕获/比较寄存器 (TM1_CHCC1)

地址偏移: 0x34

复位值 : 0x0000

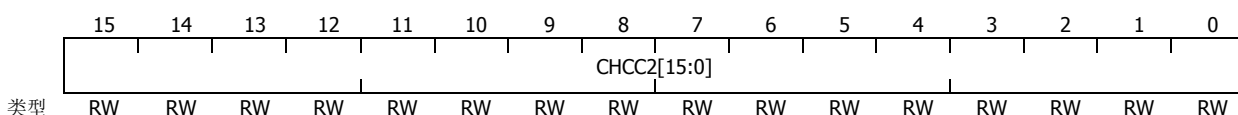


位/位域	名称	描述
15:0	CHCC1	通道 1 的捕获/比较值 当通道 1 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 1 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

2.15 TM1 通道 2 捕获/比较寄存器 (TM1_CHCC2)

地址偏移: 0x38

复位值 : 0x0000

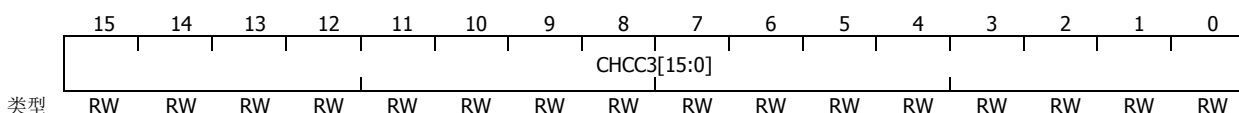


位/位域	名称	描述
15:0	CHCC2	通道 2 的捕获/比较值 当通道 2 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 2 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

2.16 TM1 通道 3 捕获/比较寄存器 (TM1_CHCC3)

地址偏移: 0x3C

复位值 : 0x0000



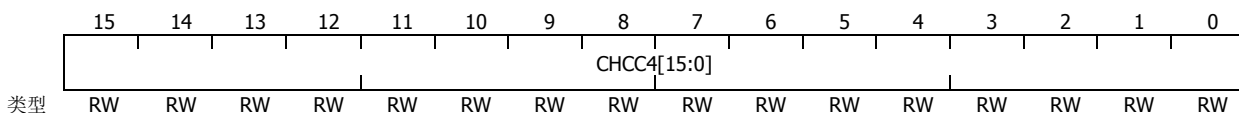
类型

位/位域	名称	描述
15:0	CHCC3	通道 3 的捕获/比较值 当通道 3 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 3 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

2.17 TM1 通道 4 捕获/比较寄存器 (TM1_CHCC4)

地址偏移: 0x40

复位值 : 0x0000



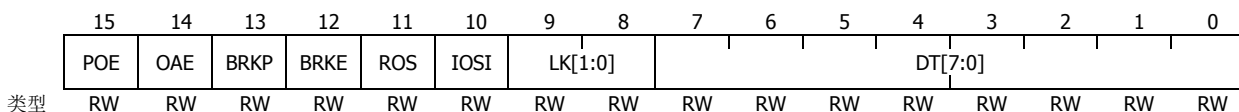
类型

位/位域	名称	描述
15:0	CHCC4	通道 4 的捕获/比较值 当通道 4 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 4 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

2.18 TM1 紧急停止及死区时间寄存器 (TM1_BKDT)

地址偏移: 0x44

复位值 : 0x0000



类型

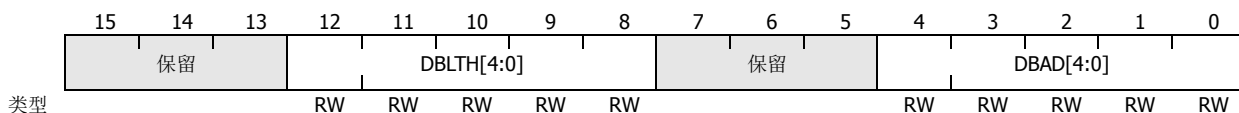
位/位域	名称	描述
15	POE	主输出使能 取决于 OAE 位的状态，本位由软件置位，由硬件自动清零。当紧急停止输入有效时，本位将由硬件异步清零。当通道配置为输出模式时，若相应的使能位 (TMx_CHE 寄存器的 CHxE、CHxNE) 已经置位，则对本位写 1 即可使能通道输出 (OC 与 OCN)。 0: 禁用通道输出，或强制为无效状态 1: 使能通道输出

位/位域	名称	描述
14	OAE	<p>输出自动使能 本位控制 POE 能否被硬件自动使能。 0: 硬件不得将 MOE 置位 1: 假如紧急停止输入并未有效, 那么硬件可以在下一次更新事件时自动将 MOE 置位。 仅当 TMx_BKDT 寄存器的 LK[1:0]位域为 00 时, 本位才能修改。</p>
13	BRKP	<p>紧急停止极性 本位用于指定 BRK 输入信号的极性。 0: BRK 输入信号为低电平有效 1: BRK 输入信号为高电平有效</p>
12	BRKE	<p>紧急停止使能 若将本位写 1, 则将使能 BRK 输入以及 CCS 时钟阻塞事件输入。 0: 禁用紧急停止输入 1: 使能紧急停止输入 仅当 TMx_BKDT 寄存器的 LK[1:0]位域为 00 时, 本位才能修改。</p>
11	ROS	<p>运行模式关断状态配置 当 POE 置位时, 本位用于指定通道的输出状态, 具有互补输出的通道, 并且已经配置为输出模式。 0: 当 POE 置位时, 禁用通道输出信号 (OC/OCN) 1: 当 POE 置位时, 使能通道输出信号 (OC/OCN)。这与 TM1_CHE 寄存器的 CHxE/CHxNE 设置有关。 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11 或 10 时, 本位域不得更改。</p>
10	IOS	<p>停止模式关断状态配置 当 POE 清零时, 本位用于指定通道的输入状态, 已经配置为输入模式。 0: 当 POE 清零时, 禁用通道输出信号 (OC/OCN) 1: 当 POE 清零时, 使能通道输出信号 (OC/OCN)。这与 TM1_CHE 寄存器的 CHxE/CHxNE 设置有关。 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11 或 10 时, 本位域不得更改。</p>
9:8	LK	<p>锁定控制 本位域指定相关寄存器的写保护。 00: 禁用锁定功能。寄存器没有写保护。 01: 锁定模式 1。TM1_CTRL2 寄存器的 ISOx/ISOxN 以及 TM1_BKDT 寄存器的 BRKE/BRKP/OAE/DT 都将被写保护。 10: 锁定模式 2。除了锁定模式 1 中被保护的位域之外, TMx_CHE 寄存器的 CHxP/CHxNP (假如相应通道配置为输出模式) 以及 TM1_BKDT 寄存器的 ROS/IOS 也被写保护。 11: 锁定模式 3。除了锁定模式 2 中倍保护的位域之外, TM1_CHCTRLx 寄存器的 CHxOM/CHxOSE (假如相应通道配置为输出模式) 也被写保护。 本位域只允许在复位后写入一次。一旦写入 TMx_BKDT 寄存器, 那么本位域将被写保护。</p>
7:0	DT	<p>死区时间值 本位域用于指定死区时间, 死区时间将被插入到输出跳变之前。DT 值与死区时间的对应关系为: DT[7:5]=0xx 时, 死区时间=DT[7:0]×t_{DT}, 且 t_{DT}=t_{DTS} DT[7:5]=10x 时, 死区时间=(64+DT[5:0])×t_{DT}, 且 t_{DT}=t_{DTS}*2 DT[7:5]=110 时, 死区时间=(32+DT[4:0])×t_{DT}, 且 t_{DT}=t_{DTS}*8 DT[7:5]=111 时, 死区时间=(32+DT[4:0])×t_{DT}, 且 t_{DT}=t_{DTS}*16 仅当 TMx_BKDT 寄存器的 LK[1:0]位域为 00 时, 本位才能修改。</p>

2.19 TM1 DMA 控制寄存器 (TM1_DCTRL)

地址偏移: 0x48

复位值 : 0x0000

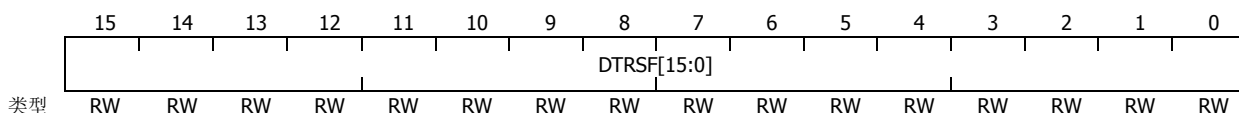


位/位域	名称	描述
15:13	保留	请保持其复位状态
12:8	DBLTH	DMA 访问突发长度 当寄存器访问是通过 TMx_DTRSF 地址实现的, 则本位域指定突发传输的次数。
7:5	保留	请保持其复位状态
4:0	DBAD	DMA 访问基地址 当寄存器访问是通过 TMx_DTRSF 地址实现的, 则本位域指定相较于 TM1_CTRL1 寄存器的地址偏移量。

2.20 TM1 DMA 传输寄存器 (TM1_DTRSF)

地址偏移: 0x4C

复位值 : 0x0000



位/位域	名称	描述
15:0	DTRSF	DMA 传输 当对此地址进行读写操作时, 实际将会访问如下地址: (DBAD+突发计数器)×4 突发计数器由硬件计算, 其范围为 0~DBLTH。

3 通用定时器 (TM2~TM4)

3.1 简介

通用定时器 (TM2、TM3 及 TM4) 包括: 一个 16 位的向上/向下计数器, 四个 16 位比较/捕获寄存器 (TMx_CHCC), 一个 16 位自动重载寄存器 (TMx_CARL) 以及多个控制寄存器。这些寄存器可以用于各种功能, 包括通用定时器、输入信号脉宽测量或输出特定波形 (例如单脉冲输出或 PWM 输出)。TMx 使用一个两输入的解码器, 支持编码器接口。

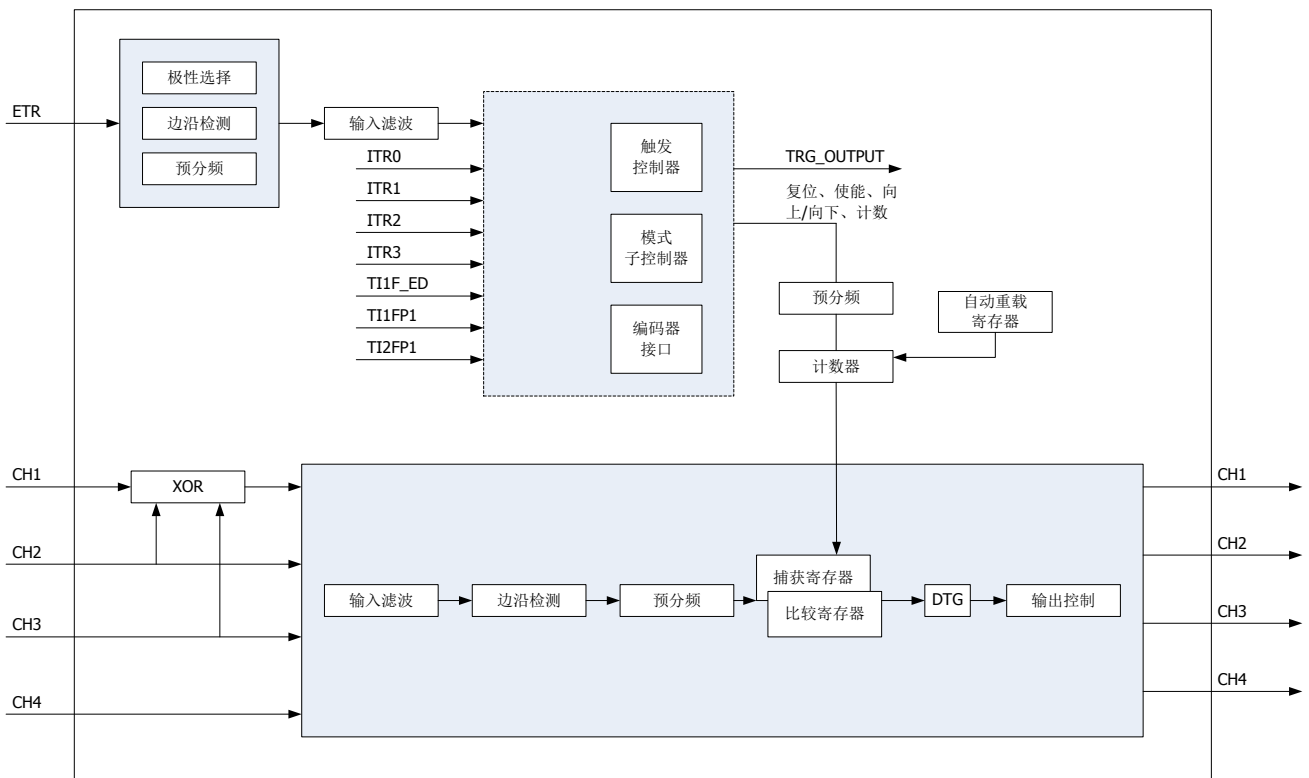
3.2 主要特点

- 16 位向下、向上、下上自动重载计数器
- 16 位可编程预分频器，计数器时钟频率可按照 1~65536 之间的任意系数分频
- 最大 4 个独立通道，支持多种功能，包括捕获、比较匹配输出、产生 PWM 波形（边沿对齐模式或中心对齐模式）以及单脉冲模式输出
- 根据更新、触发事件、输入捕获事件、输出比较匹配事件或紧急停动输入，产生中断或 DMA 请求
- 提供同步电路，控制 TMx 按照外部信号工作，或实现多个定时器的级联
- 编码接口控制器，将两输入通过正交解码工作
- TMx 主/从模式控制器

3.3 功能描述

下图描绘了通用定时器的内部配置详情。

图 60 通用定时器内部框图



预分频计数器

预分频器可将定时器时钟（CK_PSC）分频得到计数器时钟（CK_CNT），分频系数可在 1~65536 之间任意设置。分频系数可通过预分频系数寄存器（TMx_PSC）更改，而且在运行时也能更改，

但要等到下一次更新事件后才会生效。

图 61 分频系数从 1 改为 2 的计数器时序图

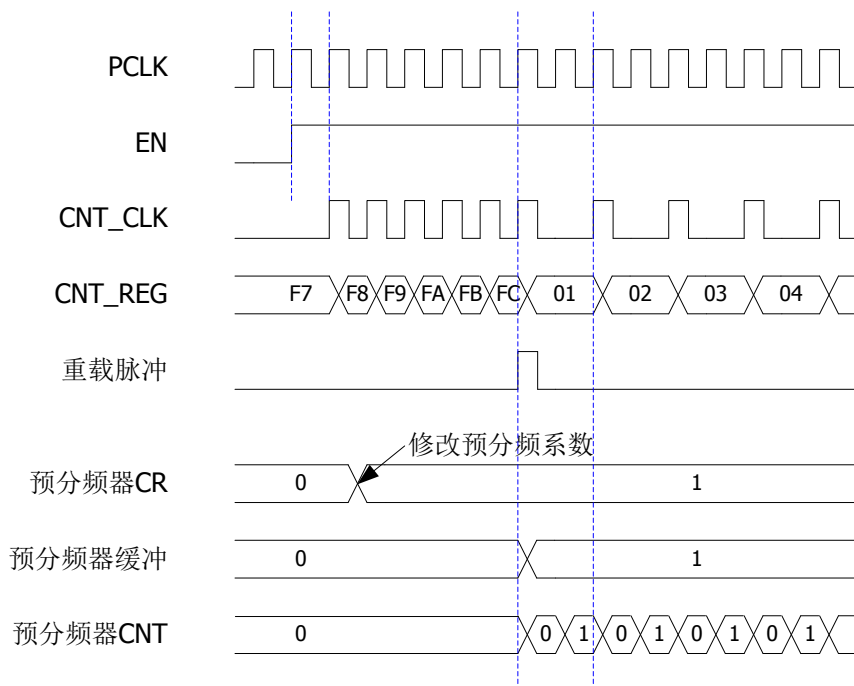
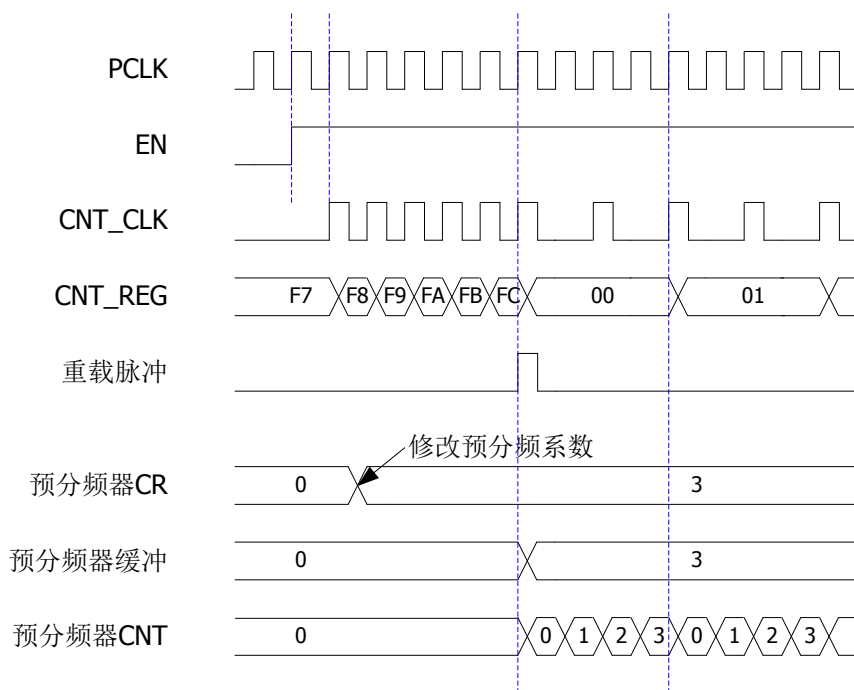


图 62 分频系数从 1 改为 4 的计数器时序图



向上计数模式

在这种工作模式下，计数器的值从 0 开始累加，一直累加到计数器重载值（在 `TMx_CARL` 寄存器中设置）。一旦计数器值达到了重载值，寄存器将从 0 开始重新计数。假如设置过重复次数计数器，那么在计数值溢出若干次后将产生更新事件；否则每当计数值溢出时都会产生更新事件。对于向上计数模式，`TMx_CTRL1` 寄存器中的计数方向 `DIR` 位必须设置为 0。

当 `TMx_EVG` 寄存器中的 `UPG` 置位时，表明发生了更新事件，此时计数值将初始化为 0，并产生一个更新事件。

假如将 `TMx_CTRL1` 寄存器中的 `UPDIS` 置位，那么更新事件将被禁用。

当更新事件发生时，所有寄存器的内容（包括重复次数计数器、重载值寄存器、预分频系数寄存器）都将更新。

图 63 分频系数为 1 的计数器时序图

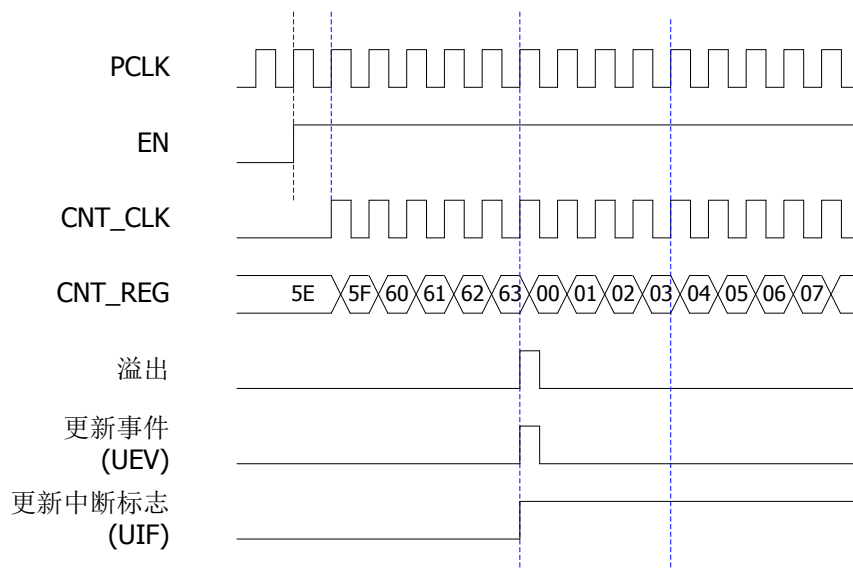


图 64 分频系数为 2 的计数器时序图

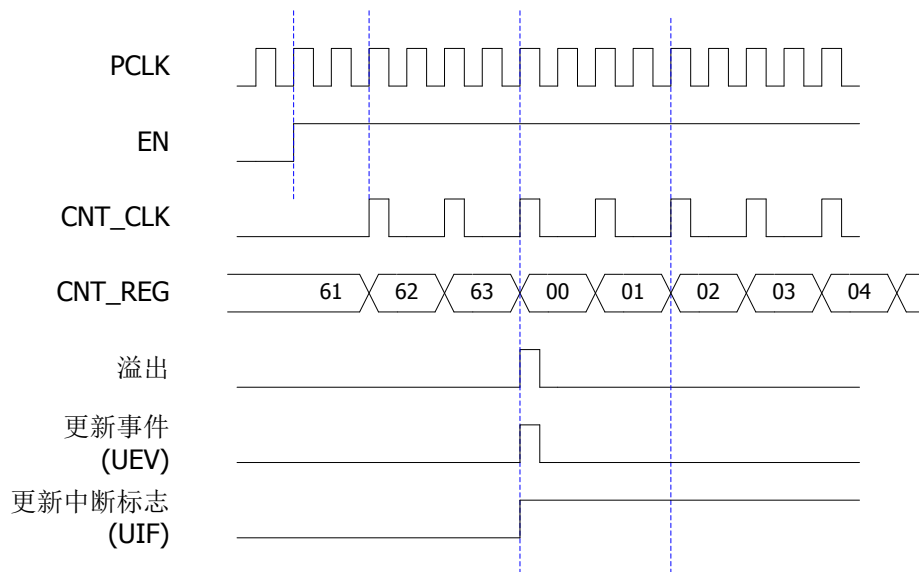


图 65 分频系数为 4 的计数器时序图

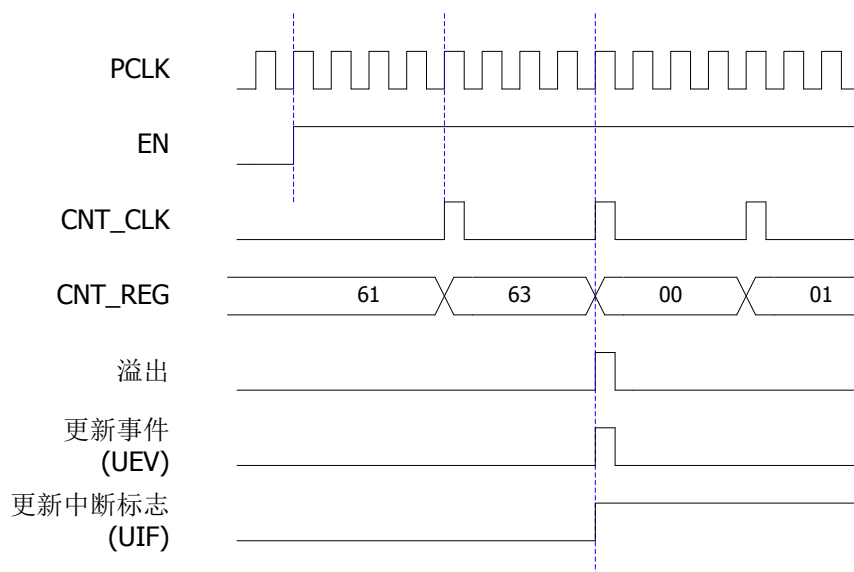


图 66 分频系数为 N 的计数器时序图

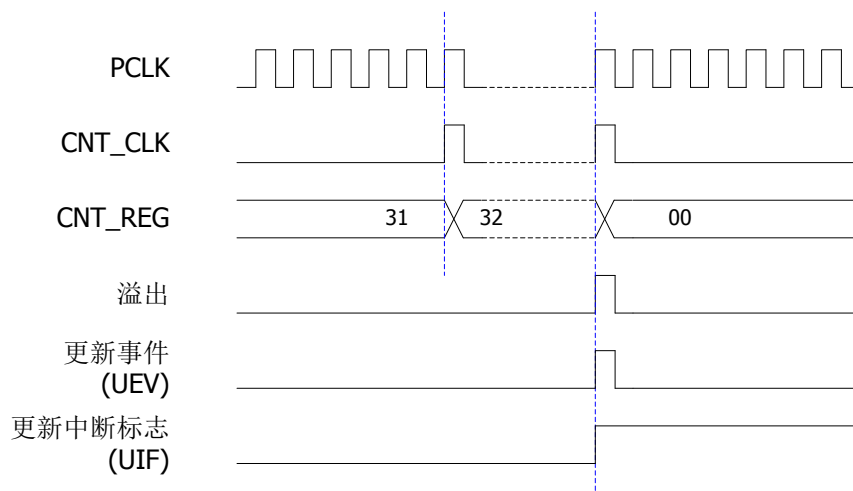


图 67 当 ARPE=0 时计数器更新事件的计数器时序图

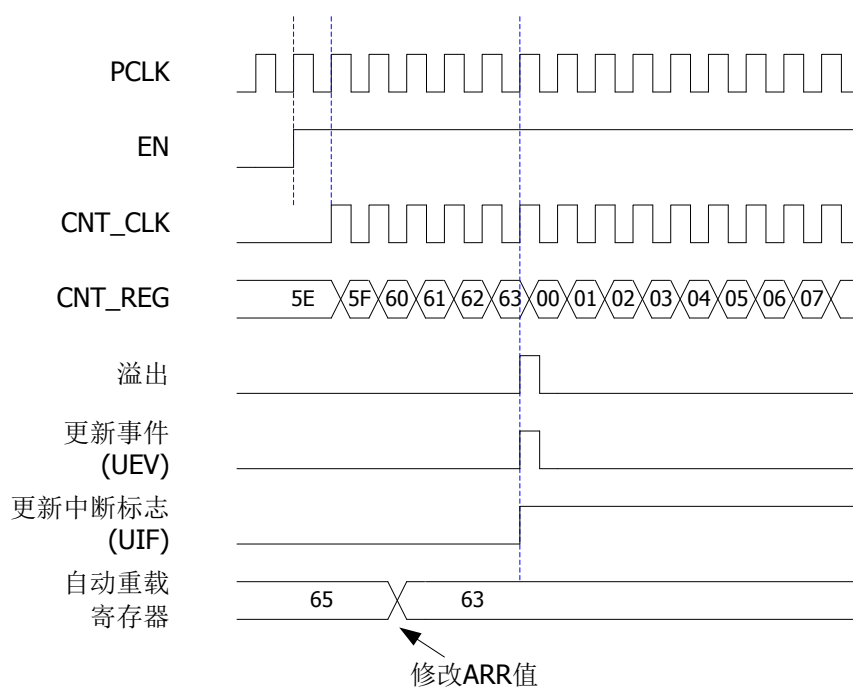
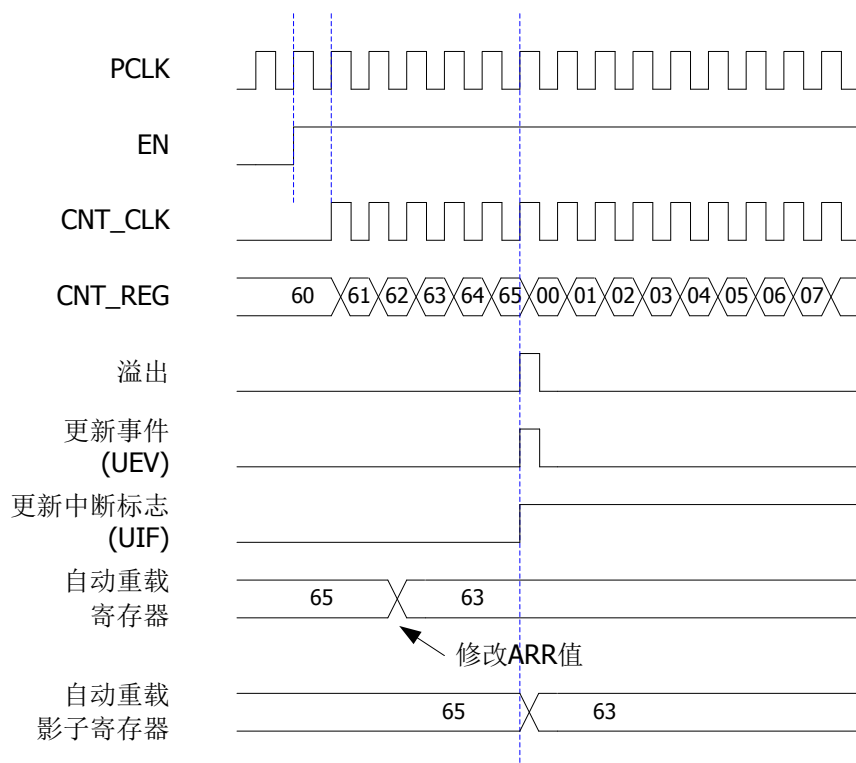


图 68 当 ARPE=1 时计数器更新事件的计数器时序图



向下计数模式

在这种工作模式下，计数器的值从重载值（在 `TMx_CARL` 寄存器中设置）开始递减，一直递减到 0。一旦计数器值达到了 0，计数器将重新从重载值开始向下计数。假如设置过重复次数计数器，那么在计数值下溢一定次数后便会产生更新事件；否则每次计数器下溢后都会产生更新事件。在向下计数模式下，`TMx_CTRL1` 寄存器中的计数方向 `DIR` 位必须设置为 1。

当 `TMx_EVG` 寄存器中的 `UPG` 置位时，表明发生了更新事件，计数值将初始化为重载值并产生一个更新事件。

假如将 `TMx_CTRL1` 寄存器中的 `UPDIS` 置位，那么更新事件将被禁用。

当更新事件发生时，所有寄存器（包括重复次数计数器、重载值寄存器、预分频系数寄存器）的内容都将更新。

图 69 分频系数为 1 的计数器时序图

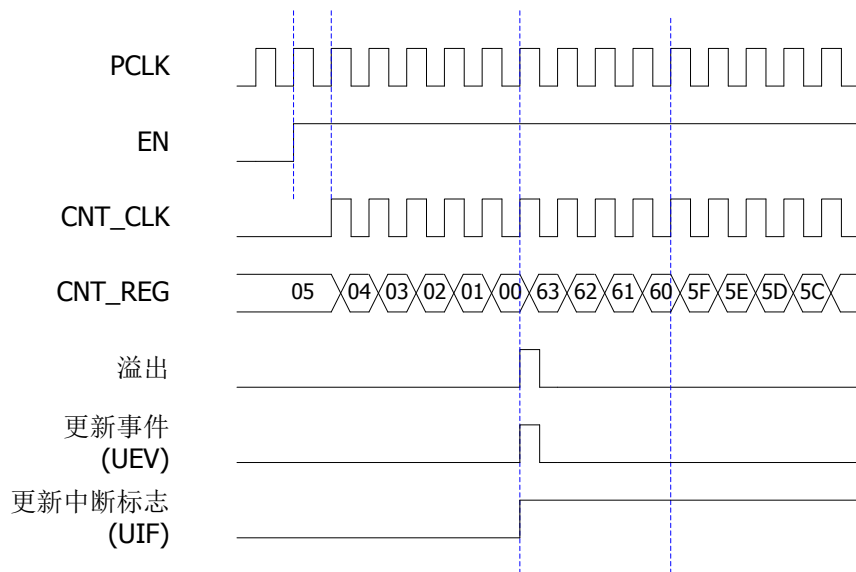


图 70 分频系数为 2 的计数器时序图

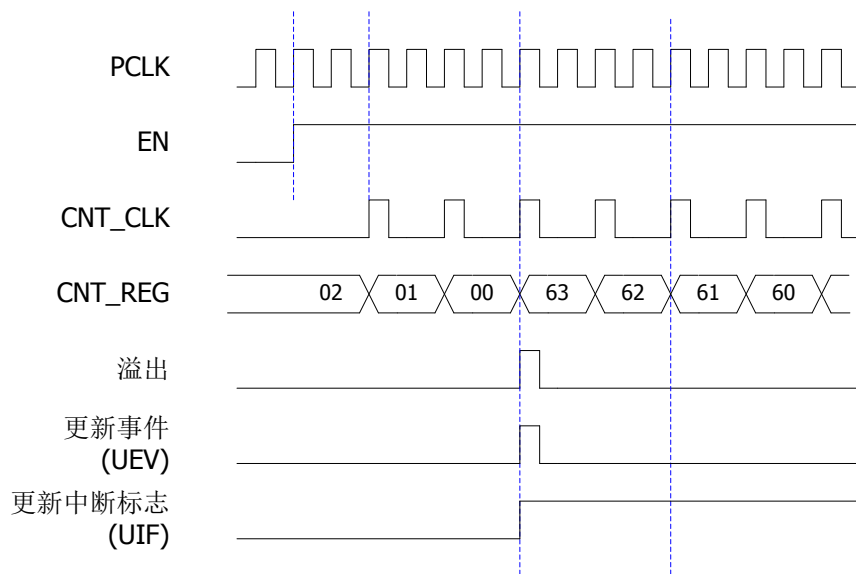


图 71 分频系数为 4 的计数器时序图

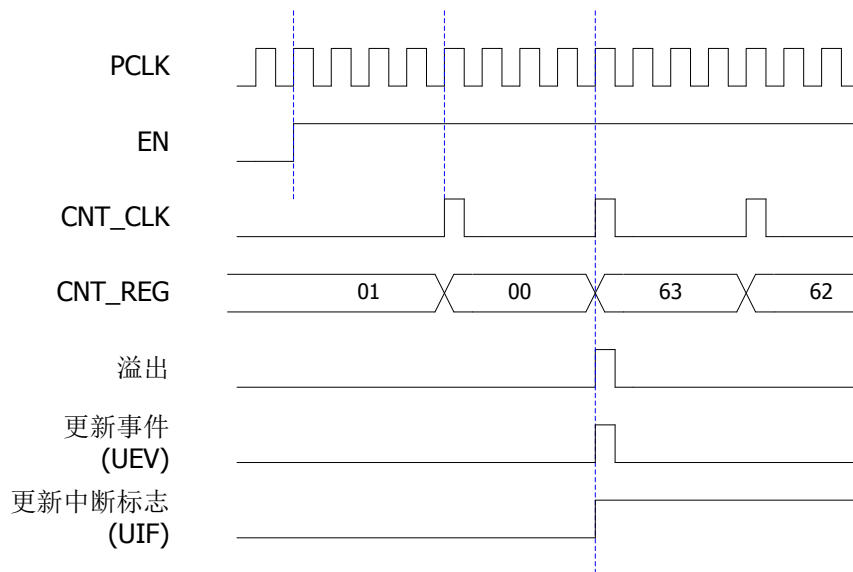


图 72 分频系数为 N 的计数器时序图

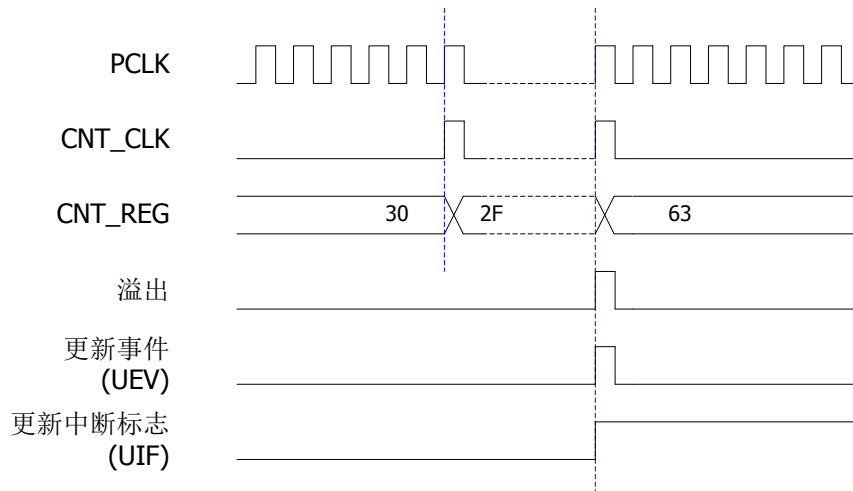
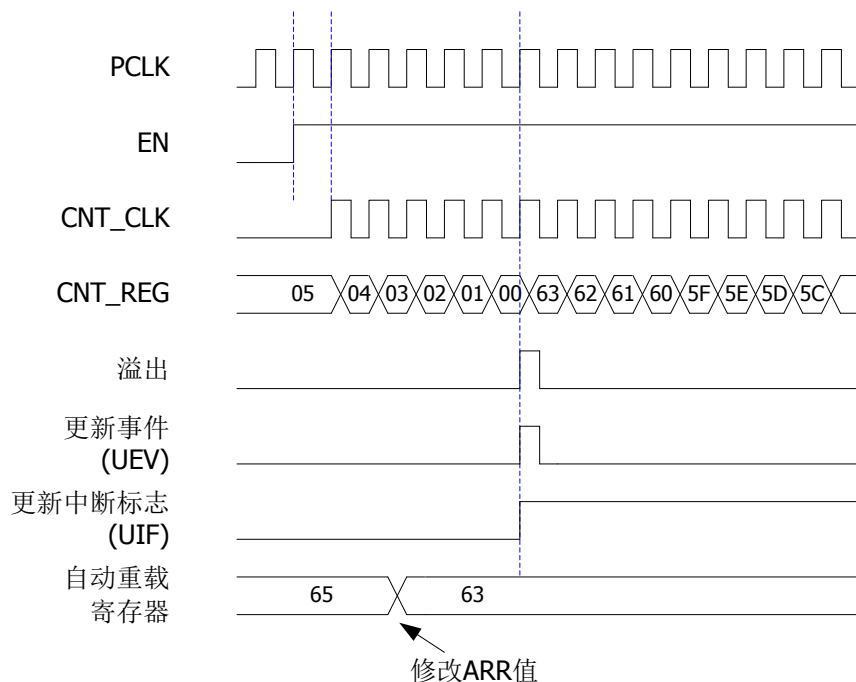


图 73 当计数器未使用时，计数器更新事件的计数器时序图



中心对齐计数模式

在中心对齐计数模式下，计数器的值从 0 开始递增到重载值，然后从重载值开始递减到 0，如此反复。定时器模块在计数值递增到重载值减 1 时会产生一个溢出事件，在计数值递减到 1 时产生一个下溢事件。在中心对齐模式下，`TMx_CTRL1` 寄存器中的计数方向 `DIR` 位只读，用于表明当前的计数方向。该位将由硬件自动更新。

在中心对齐模式下，将 `TMx_EVG` 寄存器中的 `UPG` 置位，即可将计数值清零（不管当前计数器是向上计数还是向下计数）并产生一个更新事件。

在向下计数（`TMx_CTRL1` 寄存器中的 `CAM` 位域为 01）模式下发生下溢事件、在向上计数（`TMx_CTRL1` 寄存器中的 `CAM` 位域为 10）模式下发生溢出事件、或在中心对齐（`TMx_CTRL1` 寄存器中的 `CAM` 位域为 11）模式下发生下溢或溢出事件时，`TMx_EVG` 寄存器中的 `UPIF` 位都会置位。

若将 `TMx_CTRL1` 寄存器中的 `UPDIS` 置位，那么更新事件将被禁用。

当更新事件发生时，所有寄存器（包括重复次数计数器、重载值寄存器、预分频系数寄存器）的内容都将更新。

图 74 分频系数为 1、TMx_ARR = 0x5 的计数器时序图

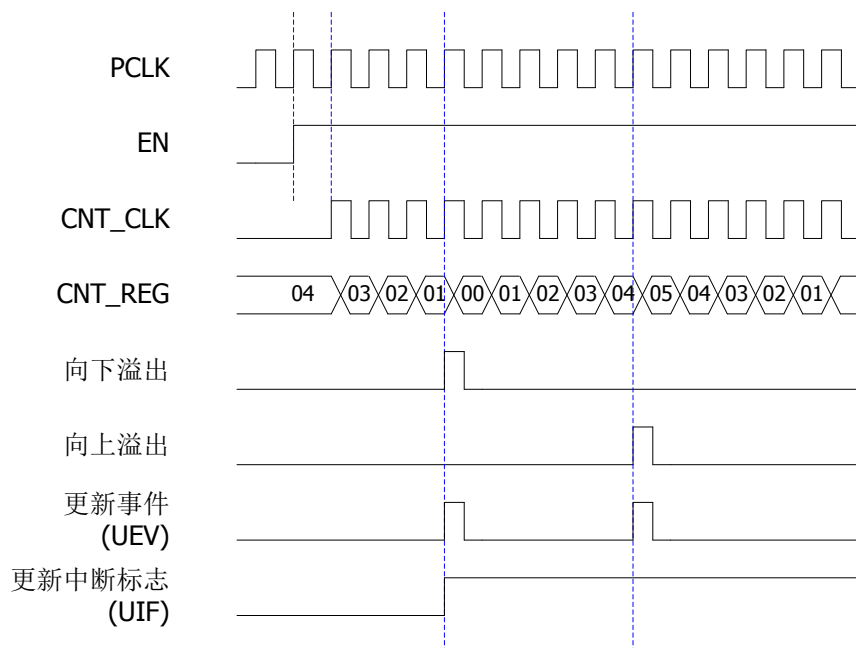


图 75 分频系数为 2 的计数器时序图

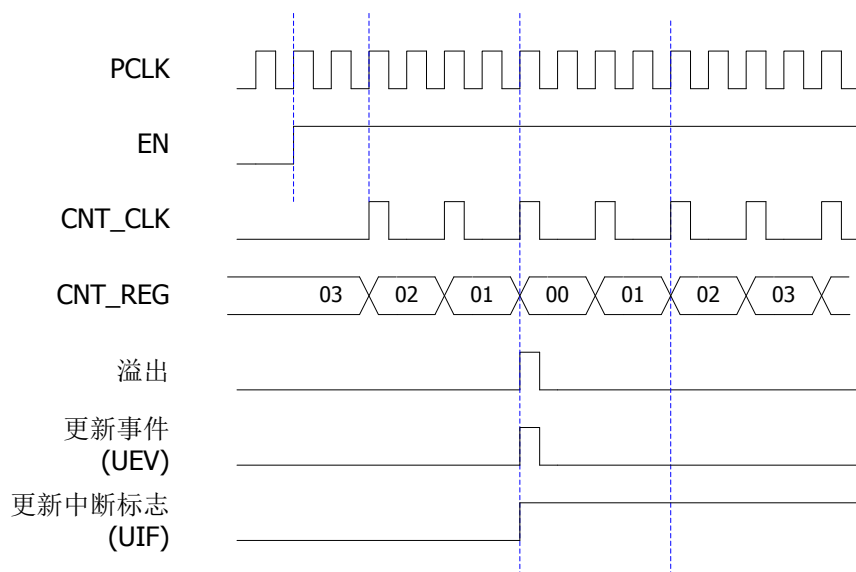


图 76 分频系数为 4、TMx_ARR = 0x63 的计数器时序图

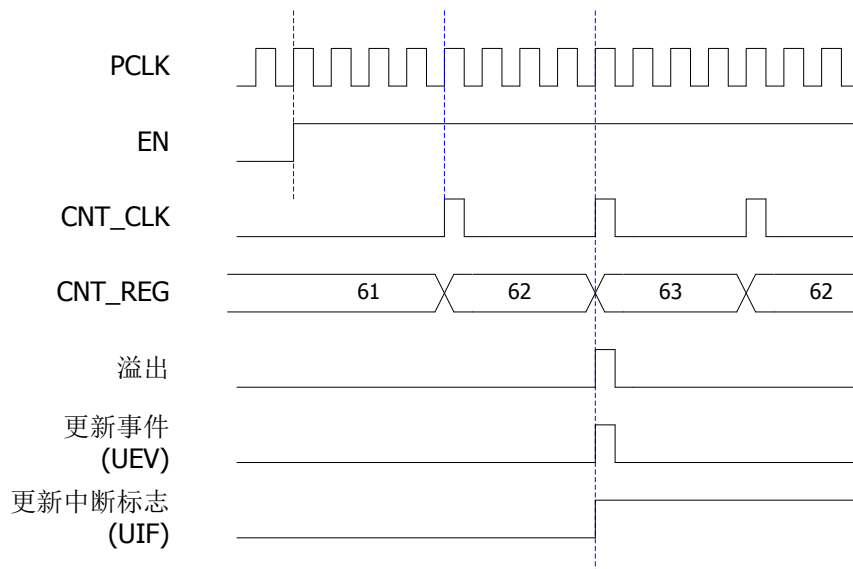


图 77 分频系数为 N 时的计数器时序图

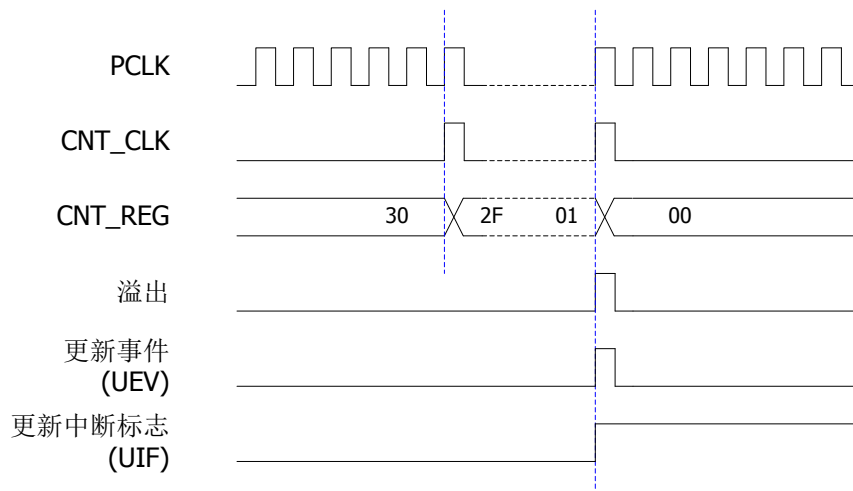


图 78 APRE = 1 时计数器更新事件的计数器时序图

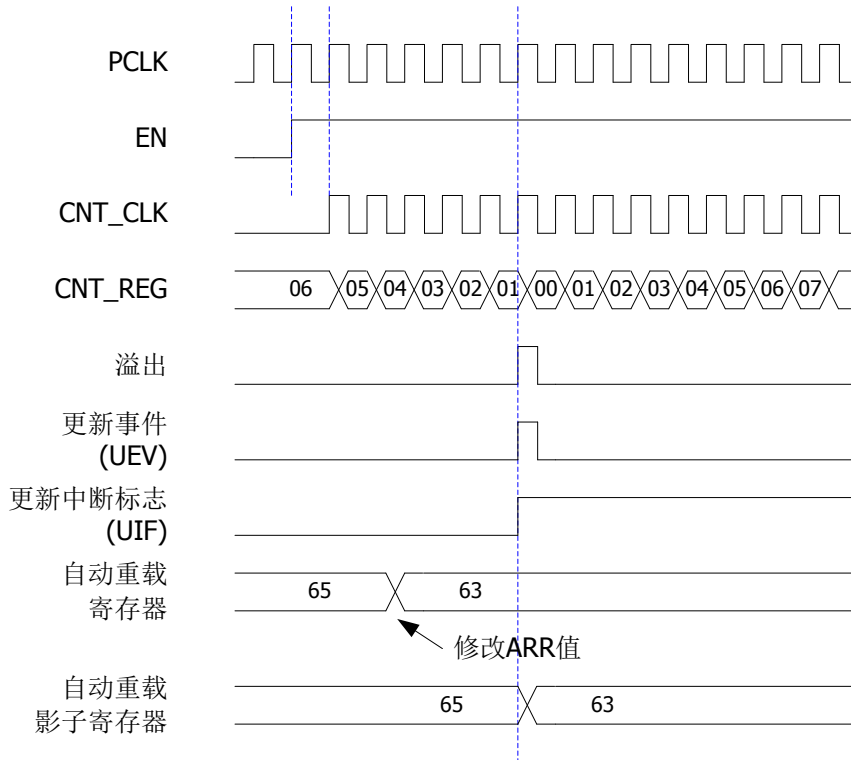
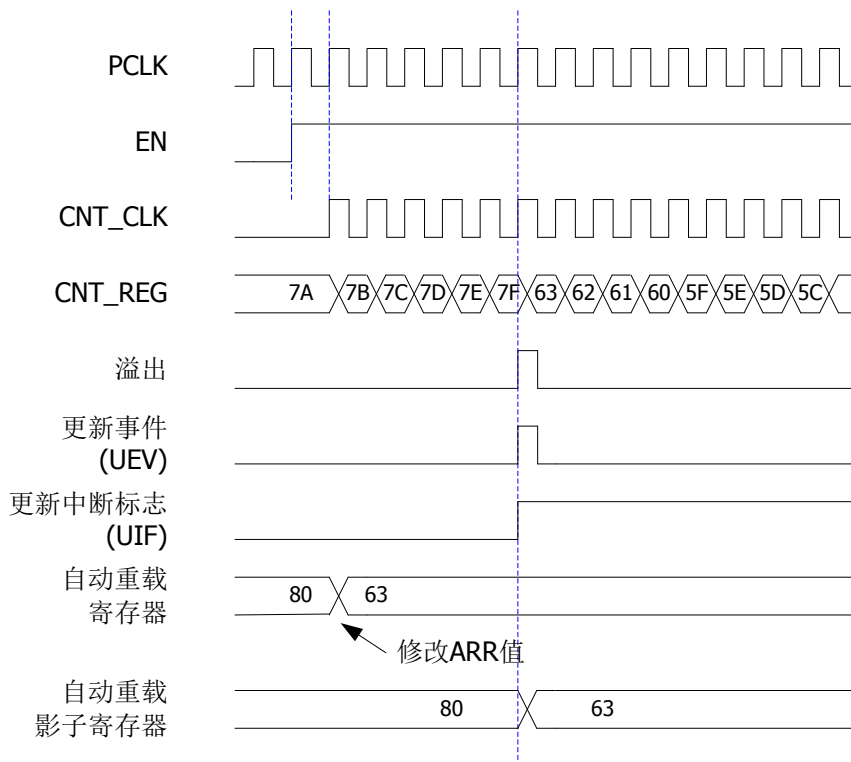


图 79 APRE = 1 时计数器更新事件（计数器上溢）的计数器时序图



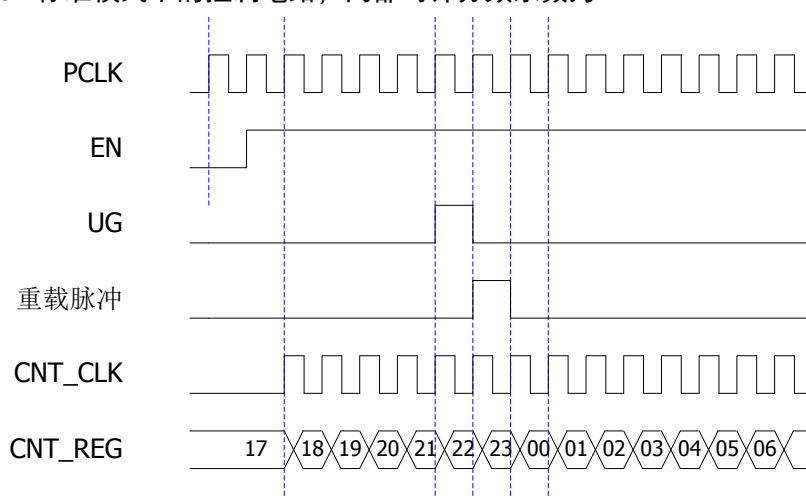
时钟选择

下面描述定时器模块时钟控制器，决定内部预分频计数器的时钟源。

■ 内部定时器时钟 CK_TMx

默认的内部时钟源是 APB2 时钟 CK_APB2，禁用从机模式后即被用于驱动预分频计数器。若将 TMx_SMCR 寄存器中的 SMC 位域设置为 0x1、0x2、0x3、0x6、0x7 中的某个值，那么从模式控制器将被使能，且预分频器将由其它时钟源（通过 TMx_SMCR 寄存器中的 TRGS 位域选择）供给，详见后述。当从模式选择位域 SMC 设置为 0x4 或 0x5，则内部时钟 CK_TM1 被选作预分频计数器的驱动时钟源。

图 80 标准模式下的控制电路，内部时钟分频系数为 1



■ 正交解码器

要选择正交解码器模式，TMx_SMCR 寄存器中的 SMC 位域必须为 0x1、0x2 或 0x3 中的某个值。正交编码器功能采用 TMx_CH0、TMx_CH1 两管脚的输入状态产生时钟脉冲，以该脉冲驱动预分频计数器。计数方向位 DIR 则由硬件在输入源信号的每次跳变沿自动选择。输入源信号可以只从 TMx_CH0 管脚获取，只从 TMx_CH1 管脚获取，或同时从 TMx_CH0、TMx_CH1 获取。

■ 内部触发器输入 (ITI)

预分频计数器可以在 ITI 信号的每个上升沿或下降沿计数。要选择这种模式，应将 TMx_SMCR 寄存器中的 SMC 位域设置为 0x6；此时计数器的表现如同一个事件计数器。输入事件（也就是这里的 ITI）可通过设置 TRGS 位域进行选择。若选择 ITI 信号作为时钟源，那么内部边沿检测电路将在 ITI 信号的每个上升沿或下降沿产生一个时钟脉冲，用以驱动预分频计数器。

■ 外部输入管脚 (TIx)

预分频计数器可以由外部管脚 TMx_TIx 的每个上升沿或者下降沿驱动计数。要选择这种模式，应将 TMx_SMCR 寄存器中的 SMC 位域设置为 0x7，并将 TRGS 位域设置为 0x4、0x5 或 0x6。请注意：TIx 是从 TMx_TIx 经过一个数字滤波器后采样得到的。

■ 外部触发器输入 (ETIF)

预分频计数器可以由外部管脚 TMx_ETI 的每个上升沿或者下降沿驱动计数。要选择这种模式，应将 TMx_SMCR 寄存器中的 $ECM2E$ 置位。另一种选择 $ETIF$ 信号作为时钟源的方法是将 SMC 位域设为 $0x6$ ，并将 $TRGS$ 位域设为 $0x7$ 。请注意： $ETIF$ 信号是从 TMx_ETI 管脚经过一个数字滤波器后采样得到的。若选择 $ETIF$ 信号作为时钟源，那么包含边沿检测电路的触发控制器将在 $ETIF$ 信号的每个上升沿产生一个时钟脉冲，用以驱动预分频计数器。

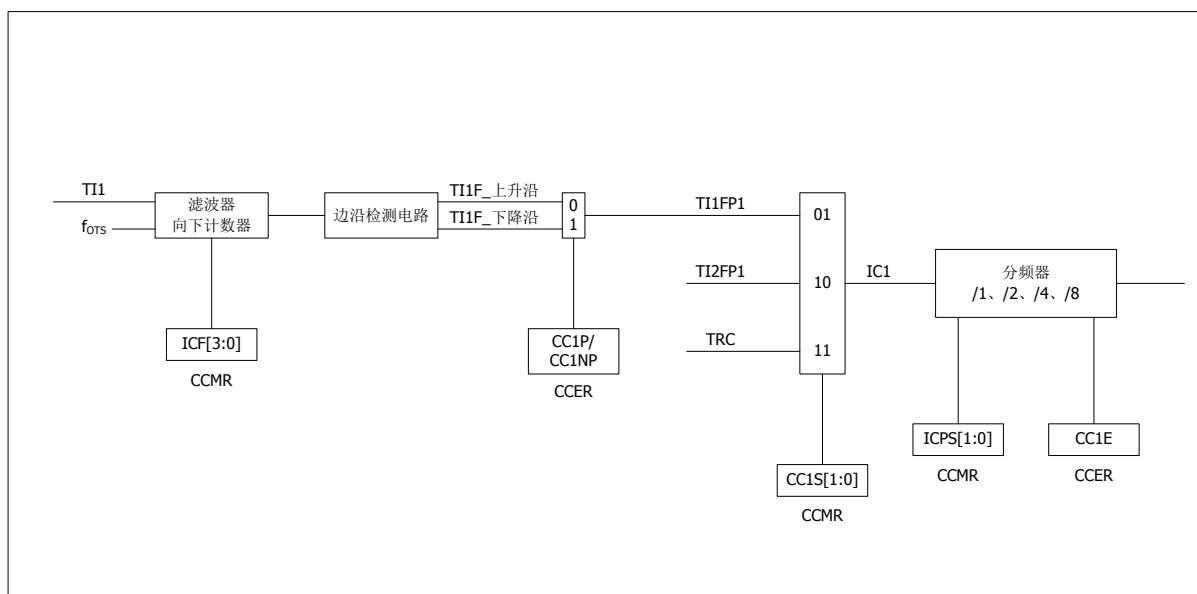
捕获/比较通道

TMx 有 4 个独立的通道，可用于捕获输入或者比较匹配输出。每个通道都是围绕着一个通道捕获/比较寄存器设计的，由一个输入级、通道控制器以及一个输出级组成。

■ 输入捕获级

输入捕获级由数字滤波器、通道极性选择、边沿检测电路以及通道预分频器组成。通道 0 的输入信号 ($TI0$) 可以选择来自 TMx_CH0 管脚，或者 TMx_CH0 、 TMx_CH1 、 TMx_CH2 管脚的异或运算结果。通道输入信号 (TIx) 经由数字滤波器采样后产生输入信号 $TIxF$ ，之后通道极性选择以及边沿检测电路可产生 $TIxS0ED$ 或 $TIxS1ED$ 信号，实现输入捕获功能。有效的输入事件号可通过通道输入预分频寄存器 (CHx_ICP) 设置。

图 81 捕获/比较通道 (示例：通道 1 输入级)



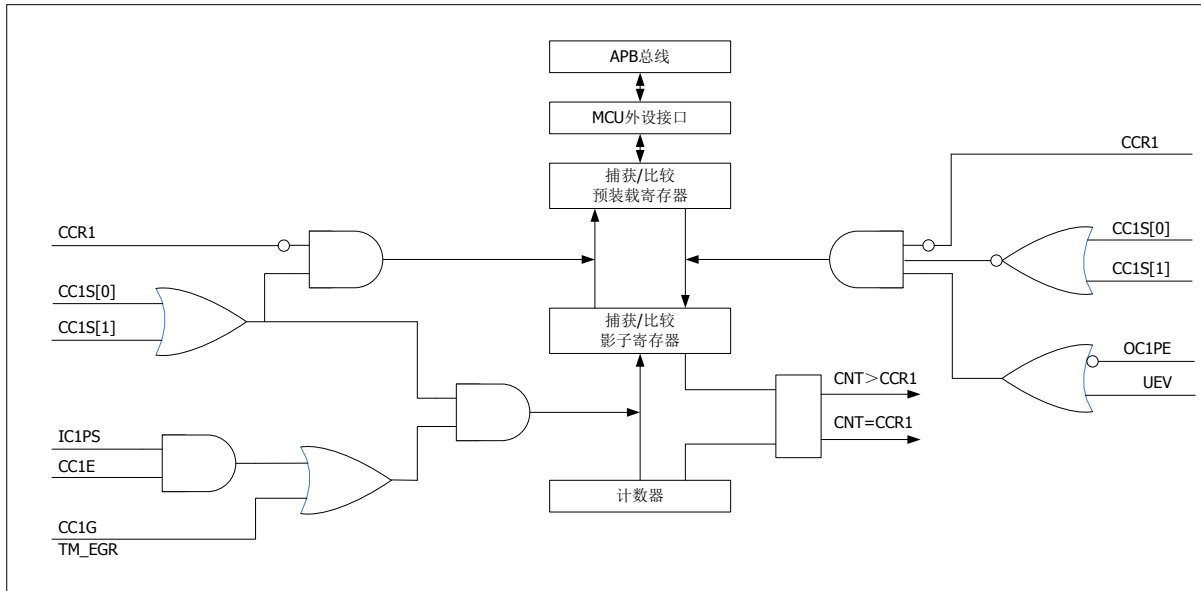
■ 通道控制器

通用定时器有 4 个独立的通道，可用于捕获输入或比较匹配输出。

当用于输入捕获模式时，当产生捕获事件后，计数值先被捕获到 TMx_CHCCx 影子寄存器中，之后传递到 TMx_CHCCx 预装载寄存器中。

当用于比较匹配输出模式时，TMx_CHCCx 预装载寄存器的值将拷贝到对应的影子寄存器中，随后计数器的值与寄存器值进行比较。

图 82 捕获/比较通道 1 主电路



■ 输出级

TMx 有 4 个通道，用于比较匹配，单脉冲输出或 PWM 输出。

图 83 捕获/比较通道 4 输出级

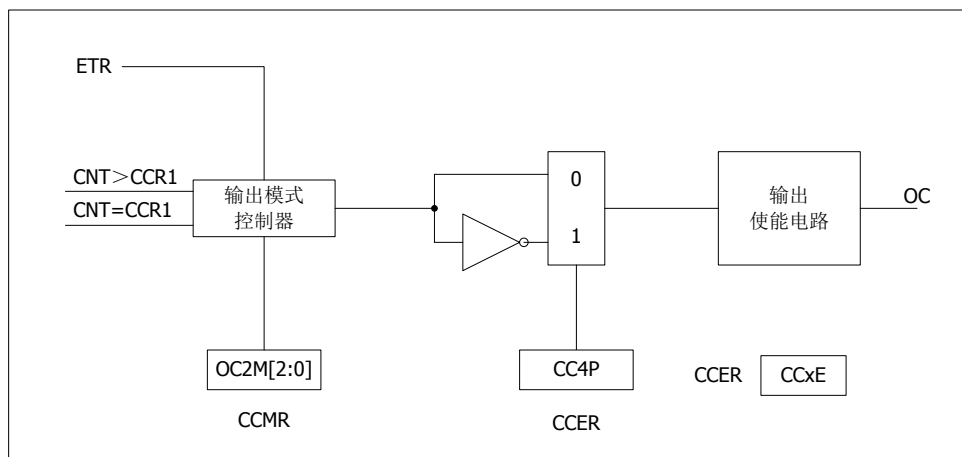
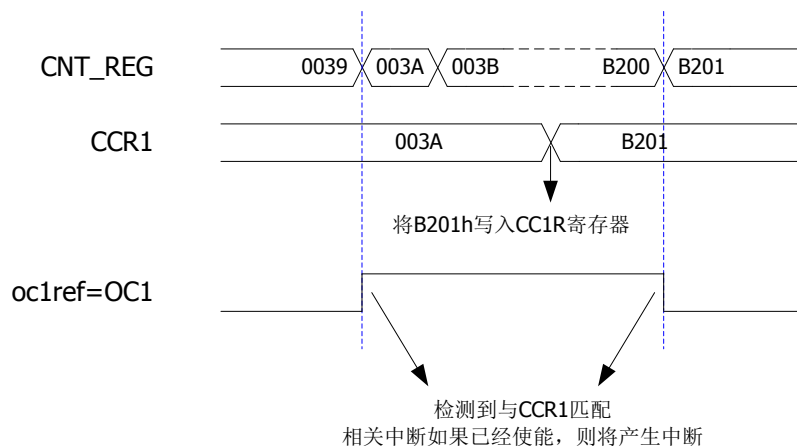


图 84 输出比较模式，OC1 的翻转



当通道用于捕获输入时，一旦输入信号发生有效跳变沿，计数值将被捕获到通道捕获/比较寄存器（`TMx_CHCCx`）中。捕获事件产生后，`TMx_STA` 寄存器中的 `CHxIF` 标志将会置位。假如 `CHxIF` 标志位已经置位（即该标志之前未被软件清零），并且本通道又产生了新的捕获事件，那么相应通道的过捕获标志位 `CHxOF` 将会置位。捕获事件产生后，取决于 `CHxDE` 位的状态将可能产生 DMA 请求，取决于 `CHxIE` 位的状态将可能产生中断。

输入捕获模式也可用于对 `TMx_CHx` 管脚的信号（`TIx`）进行脉宽测量。

通道输出参考信号

当 `TMx` 作为比较匹配输出模式时，`OCxREF` 信号（第 `x` 通道输出参考信号）将通过配置 `CHxOM` 位域进行定义。当计数值与 `TMx_CHCCx` 寄存器的内容匹配时，`OCxREF` 信号可以有多种类型的输出功能，包括：将 `CHxOM` 位域设为 `0x00` 即可保持原始电平，将 `CHxOM` 位域设为 `0x01` 即可置 1，将 `CHxOM` 位域设为 `0x02` 即可置 0，将 `CHxOM` 位域设为 `0x03` 即可翻转信号。

PWM 模式 1 输出以及 PWM 模式 2 输出也是另一种类型的 `OCxREF` 输出，分别通过将 `CHxOM` 位域置为 `0x6/0x7` 来实现。在这两种模式中，`OCxREF` 信号电平将根据计数方向以及计数值与 `TMx_CHCCx` 寄存器内容之间的关系进行变化。详见相关寄存器的说明。

`OCxREF` 信号的另一种特殊功能就是强制输出，可通过将 `CHxOM` 位域设为 `0x4/0x5` 来实现。此时输出可强制为失效电平/生效电平，不管计数器值与 `TMx_CHCCx` 寄存器内容之间的关系如何。

当 `ETIF` 信号是从外部的 `TMx_ETI` 管脚获取的，可强制 `OCxREF` 信号为 0；若将 `TMx_CHCTRL1` 寄存器中的 `CHxOCE` 置位，也可以强制 `OCxREF` 信号为高电平。在下一个计数器更新事件发生前，`OCxREF` 信号不会返回至生效电平状态。

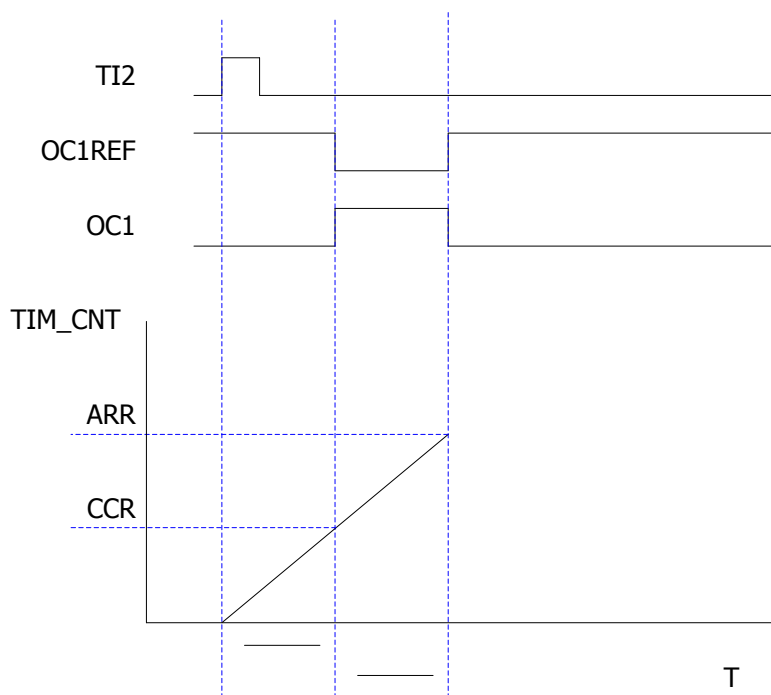
单脉冲模式

一旦定时器设置为工作在单脉冲模式下，不需要刻意将 `TMx_CTRL1` 寄存器的定时器使能位 `CNTE` 置为 1 来使能计数器。产生脉冲的触发来源多样，可以源自触发信号边沿，或通过软件将 `CNTE` 置位。将 `CNTE` 置位或来自触发信号边沿，都可以产生一个脉冲，之后 `CNTE` 位即保持在高电平

状态，直到发生更新事件或被软件写 0。假如 CNTE 位被软件清零，则计数器将停止计数，其值保持不变。假如 CNTE 位被硬件的更新事件自动清零，那么计数器将会重新初始化。

在单脉冲模式下，将 CNTE 置为 1 的触发有效沿能够自动使能计数器。不过，由于要将计数值与 TMx_CHCCx 寄存器的内容进行比较，这个过程会存在多个时钟的延迟。为尽量减少这个延迟，用户可将 $TMx_CHCTRL1$ 寄存器的 CHxOEF 置位。在单脉冲模式下，当触发信号产生上升沿后，OCxREF 信号将立即被强制拉到发生比较匹配事件而不考虑比较结果的状态。仅当输出通道配置为工作在 PWM1 或 PWM2 输出模式下，并且触发来源为触发信号时，CHxOEF 位才可用。

图 85 单脉冲模式



正交解码器

正交解码器功能采用分别来自 TMx_CH1 以及 TMx_CH2 管脚的两个正交输入信号 TI1 和 TI2，根据二者互相关联产生计数值。DIR 位在每个输入源信号的跳变沿由硬件自动改变。通过将 SMC 位域设为 0x1、0x2、0x3，可选择输入源是仅 TI1、仅 TI2，还是 TI1 和 TI2。计数方向的改变机制如下表所示。正交解码器可以当做带有方向选择的外部时钟，这就意味着计数器将会在间隔为 0 到计数器重载值之间持续计数。因此，用户必须在计数器开始计数之前完成 TMx_CARL 寄存器的配置。

表 38 编码器信号以及计数方向

计数模式	电平	TI1FP1		TI2FP2	
		上升沿	下降沿	上升沿	下降沿
仅 TI1 计数	TI2FP = 高	向下	向上	-	-
	TI2FP = 低	向上	向下	-	-
仅 TI2 计数	TI1FP = 高	-	-	向上	向下
	TI1FP = 低	-	-	向下	向上
TI1 及 TI2 计数	TI2FP = 高	向下	向上	X	X
	TI2FP = 低	向上	向下	X	X
	TI1FP = 高	X	X	向上	向下
	TI1FP = 低	X	X	向下	向上

请注意：“-”代表不计数；“X”代表不可能出现的情况。

图 86 编码器接口模式下的计数器运行情况

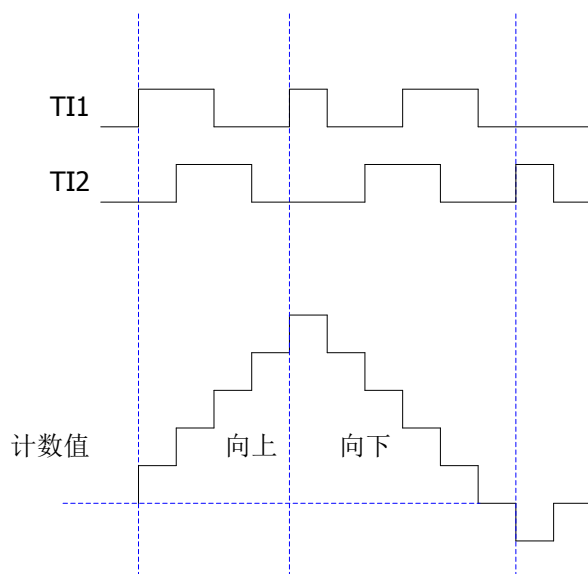
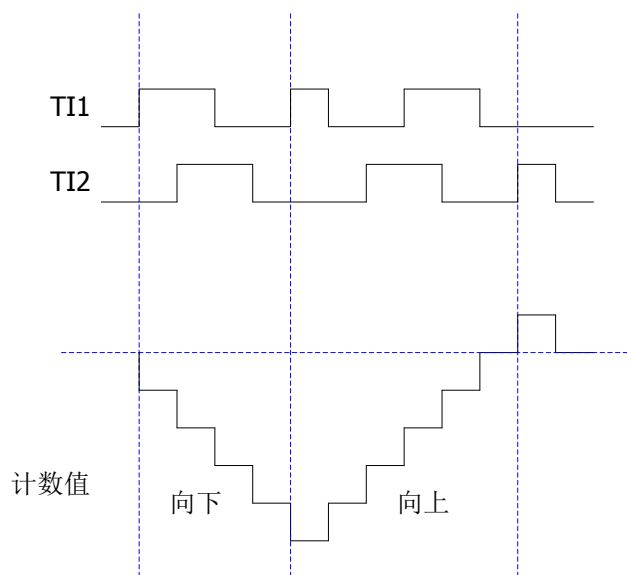


图 87 TF1FP1 极性反转后，编码器接口模式下的计数器运行情况



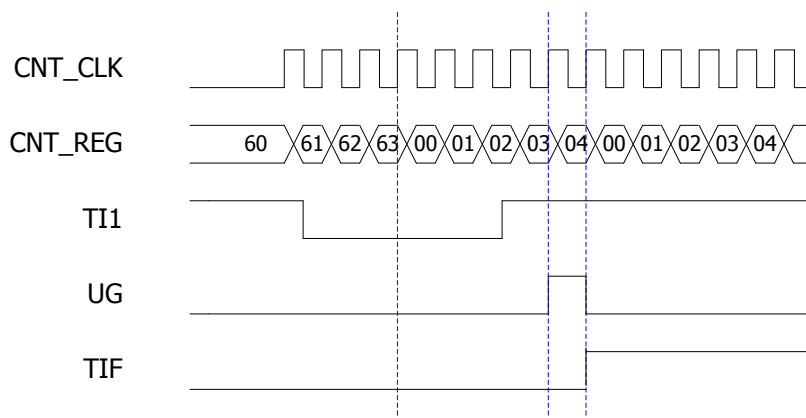
从机控制器

TMx 可以以多种模式与外部触发信号同步，包括重启模式、暂停模式以及触发模式，可通过 TMx_SMC 寄存器中的 SMC 位域进行选择。这些模式的触发输入均来自 STI 信号，可通过 TMx_SMC 寄存器中的 TRGS 位域进行选择。下面我们将详细介绍从机控制器的运行模式。

■ 重启模式

计数器及其预分频器可以 STI 信号的上升沿被重新初始化。当检测到 STI 上升沿时，更新事件软件产生位 UPG 将被硬件自动置位，于是触发事件标志也会被置位。之后计数器和预分频器将会重新初始化。尽管 UPG 位被硬件置位，但更新事件是否发生还取决于计数器更新事件禁用控制位 UPDIS 的设置。若 UPDIS=1（禁止产生更新事件），那么就不会产生更新事件，不过计数器和预分频器仍然会在检测到 STI 上升沿时重新初始化。若 UPDIS=0（使能产生更新事件），那么更新事件将会与 STI 上升沿一道产生，之后所有预装载寄存器都将被更新。

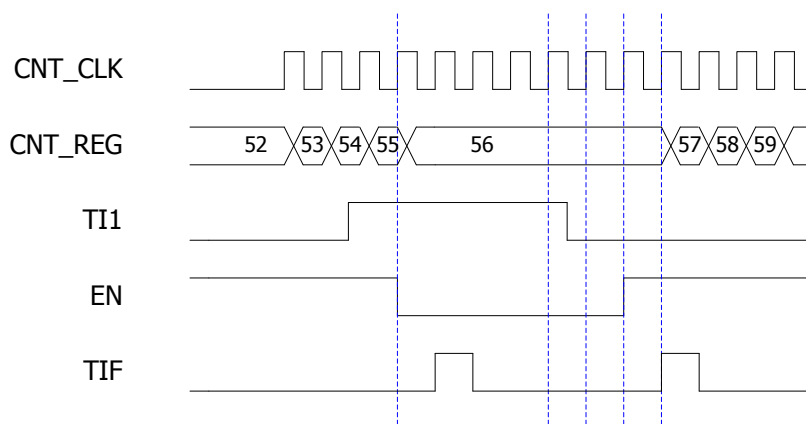
图 88 重启模式下的控制电路



■ 暂停模式

在暂停模式下，选中的 **STI** 输入信号电平被当做计数器开始运行/停止运行的控制信号。当选中的 **STI** 输入信号为高电平时，计数器开始计数；当选中的 **STI** 输入信号为低电平时，计数器停止计数（计数器保持其当前值，不会复位）。

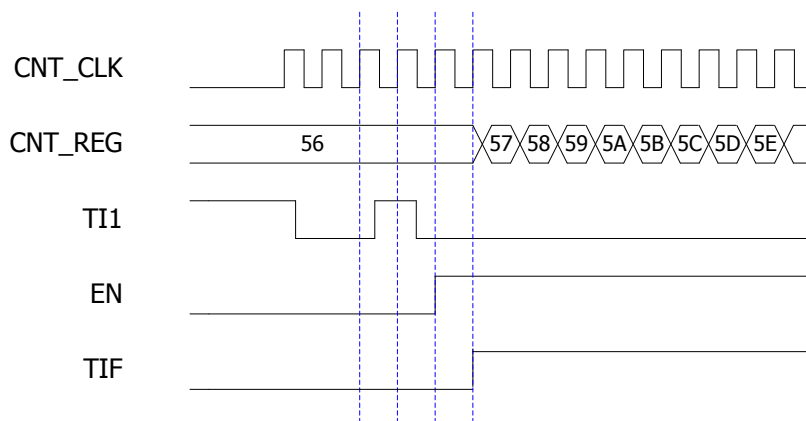
图 89 暂停模式下的控制电路



■ 触发模式

当计数器被禁用之后，计数器仍然可以在 **STI** 信号的上升沿继续计数。当检测到 **STI** 上升沿时，计数器将从当前值开始继续计数。请注意：**STI** 信号仅用于触发计数器使之继续计数，无法控制计数器的停止计数。

图 90 触发模式下的控制电路

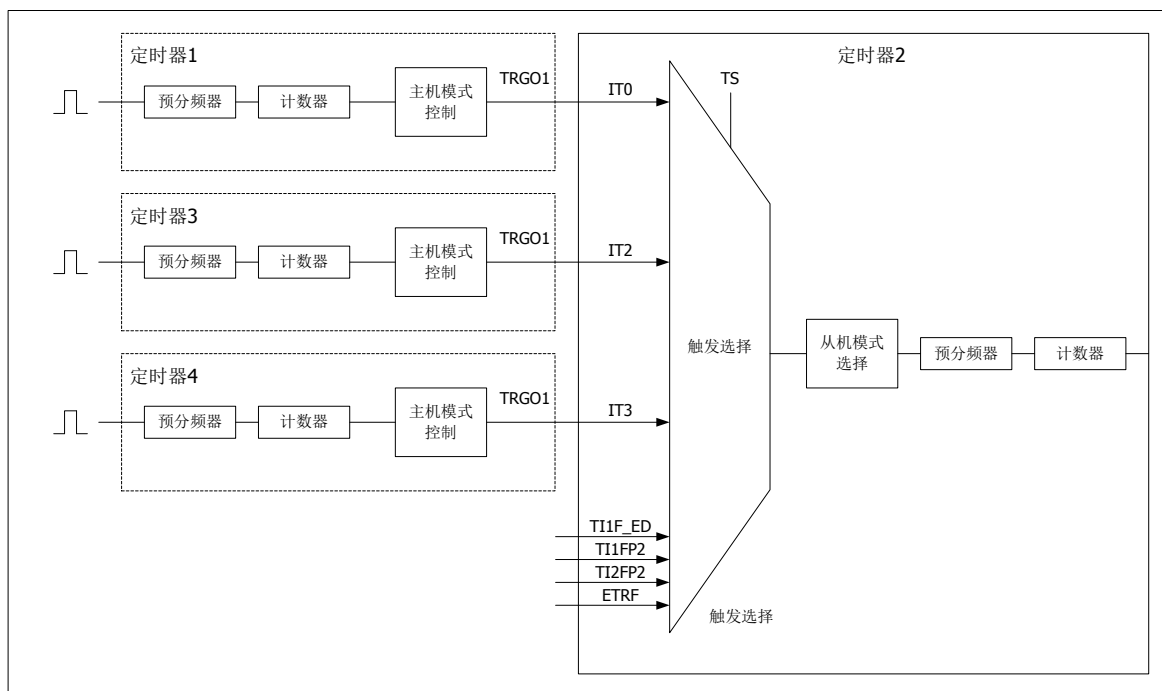


定时器互联

定时器可以内部互联形成定时器链或者实现同步。要实现这种工作方式，应将一个定时器配置为工作于主机模式下，同时将另一个定时器配置为工作于从机模式下。下面给出几种分别处于主机模式和从机模式的示例。

下图演示出定时器 1 配置为从机模式时的触发选择。

图 91 定时器 1 主机/从机模式定时器示例



■ 定时器调试模式

当 Cortex™-M3 内核暂停时，若 DBGMCU 模块中的 DBG_TMx_STOP 设置为 1，则 TMx 定时器将停止计数。

4 TM2~TM4 寄存器

4.1 TMx 控制寄存器 1 (TMx_CTRL1)

地址偏移：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CDIV[1:0]	ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CNTE	
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

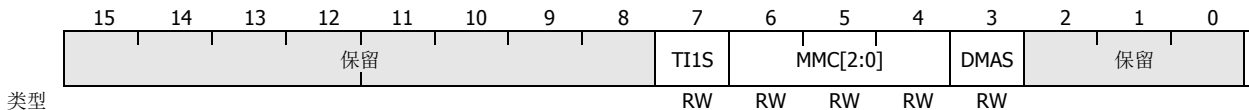
位/位域	名称	描述
15:10	保留	请保持其复位状态
9:8	CDIV	时钟分频系数 CDIV 位域由软件配置，用于指定定时器时钟 (CK_INT) 与死区及采样时钟 (DTS) 之间的分频系数。DTS 时钟用于死区时间发生器以及数字滤波器。 00: $f_{DTS} = f_{CK_INT}$ 01: $f_{DTS} = f_{CK_INT}/2$ 10: $f_{DTS} = f_{CK_INT}/4$ 11: 保留
7	ARSE	自动重载影子寄存器使能 0: 禁用 TMx_ARLD 寄存器的影子寄存器 1: 使能 TMx_ARLD 寄存器的影子寄存器
6:5	CAM	中心对齐模式选择 00: 非中心对齐模式。计数器的计数方向由 DIR 位指定 01: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置)，仅当计数器向下计数时才置位 10: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置)，仅当计数器向上计数时才置位 11: 中心对齐模式。计数器将交替向上计数和向下计数。各通道的输出比较中断标志 (在输出模式下配置)，当计数器向上或者向下计数时都会置位 当从边沿对齐模式切换到中心对齐模式时，应当禁用计数器。
4	DIR	计数方向 0: 向上计数 1: 向下计数 当定时器配置为中心对齐模式或编码器模式时，此位只可读。
3	SPM	单脉冲模式 0: 当产生计数器更新事件后，计数器仍继续计数 1: 当产生计数器更新事件后，硬件清零 CEN 位，计数器停止计数

位/位域	名称	描述
2	UPS	更新来源 本位用于设置更新事件的来源。 0: 下述任何事件都可以触发产生更新中断以及 DMA 请求: - UPG 置位 - 计数器产生溢出事件或下溢事件 - 从机模式控制器产生了更新事件 1: 仅计数器溢出/下溢事件才会触发产生更新中断以及 DMA 请求
1	UPDIS	更新禁止 本位用于使能或禁用更新事件的产生。 0: 使能更新事件。当产生下述任何事件时, 都会产生更新事件, 并且缓冲的寄存器会装载其重载值: - UPG 置位 - 计数器产生溢出事件或下溢事件 - 从机模式控制器产生了更新事件 1: 禁用计数器更新事件。假如 UG 置位或者从模式控制器产生硬件复位事件, 则缓冲的寄存器保持其当前值, 计数器与预分频器重新初始化,
0	CNTE	计数器使能 0: 禁用计数器 1: 使能计数器 当定时器 x 工作于外部时钟, 边沿触发模式以及编码器模式下, CEN 位必须由软件置位。在触发模式下, 硬件会自动将 CNTE 置位。

4.2 TMx 控制寄存器 2 (TMx_CTRL2)

地址偏移: 0x04

复位值 : 0x0000



位/位域	名称	描述
15:8	保留	请保持其复位状态
7	TI1S	通道 1 触发输入选择 0: 选择 TMx_CH1 管脚输入信号作为通道 1 的触发输入 1: 选择 TMx_CH1、CH2、CH3 的组合异或结果作为通道 1 的触发输入

位/位域	名称	描述
6:4	MMC	主模式控制 本位域用于选择在主模式下发送给从定时器的 TRGO 信号，以便实现特定的同步功能。 000: 复位。当 TMx_EVTG 寄存器的 UG 置位时，或由从模式控制器产生复位时，TRGO 脉冲将会产生。在后一种情况下，TRGO 信号相较于实际的复位信号存在一定延迟。 001: 使能。若需要同时启动多个定时器，或者要实现在一定的时间窗口内启动从定时器，那么这种模式将非常有用。在这种模式下，主模式控制器将选择计数器使能信号 CNT_EN 作为 TRGO。当 CNTE 控制位被置位，或当边沿触发模式下的触发输入为高电平时，计数器使能信号将会置位。边沿触发模式下的触发器输入信号与 TRGO 输出之间存在一定的延迟，除非选择了主-从模式。 010: 更新。在这种模式下，主模式控制器以计数器更新事件作为 TRGO。 011: 捕获/比较。在这种模式下，主模式控制器在产生捕获事件或者比较匹配事件后产生 TRGO 脉冲。 100: 比较。在这种模式下，主模式控制器选择 OC1REF 信号作为 TRGO。 101: 比较。在这种模式下，主模式控制器选择 OC2REF 信号作为 TRGO。 110: 比较。在这种模式下，主模式控制器选择 OC3REF 信号作为 TRGO。 111: 比较。在这种模式下，主模式控制器选择 OC4REF 信号作为 TRGO。
3	DMAS	DMA 请求来源选择 0: 当通道 x 事件产生时，发送通道 x 的 DMA 请求 1: 当更新事件产生时，发送通道 x 的 DMA 请求
2:0	保留	请保持其复位状态

4.3 TMx 从模式控制寄存器 (TIMx_SMC)

地址偏移: 0x08

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETPL	ECM2E	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			保留	SMC[2:0]		
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW

位/位域	名称	描述
15	ETPL	外部触发信号极性 本位指定 ETR 信号的极性。 0: ETR 信号为高电平有效或者上升沿有效 1: ETR 信号为低电平有效或者下降沿有效

位/位域	名称	描述
14	ECM2E	<p>外部时钟模式 2 使能</p> <p>在外部时钟模式 2 下，计数器由 ETRF 信号的任一有效跳变沿触发。</p> <p>0: 禁用外部时钟模式 2</p> <p>1: 使能外部时钟模式 2</p> <p>将 ECM2E 置位，与选择外部时钟模式 1 且 TRGF 连接到 ETRF (SMC=111 且 TRGS=111) 具有相同的效果。</p> <p>在复位模式、边沿驱动模式或触发模式下，可以同时使用外部时钟模式 2。不过此时 TRGS 位域绝对不能为 111。</p> <p>假如外部时钟模式 1 与外部时钟模式 2 同时被使能，则外部时钟输入将是 ETRF。</p>
13:12	ETPSC	<p>外部触发预分频器</p> <p>外部触发信号 ETRP 的频率绝对不得超过 TMx_CLK 频率的 1/4。如果外部触发信号是某种高速时钟，那么需要借助预分频器将 ETRP 信号降频。</p> <p>00: 禁用预分频器</p> <p>01: ETRP 信号将被 2 分频</p> <p>10: ETRP 信号将被 4 分频</p> <p>11: ETRP 信号将被 8 分频</p>
11:8	ETFC	<p>外部触发滤波控制</p> <p>数字滤波器中用到了事件计数器，每发生 N 个输入事件后才会改变输出。本位域指定用于采样 ETRP 信号的频率以及用在 ETRP 上的数字滤波器长度。</p> <p>0000: 禁用数字滤波。 $f_{SAMP} = f_{CK_INT}$, $N = 1$</p> <p>0001: $f_{SAMP} = f_{CK_INT}$, $N = 2$</p> <p>0010: $f_{SAMP} = f_{CK_INT}$, $N = 4$</p> <p>0011: $f_{SAMP} = f_{CK_INT}$, $N = 8$</p> <p>0100: $f_{SAMP} = f_{DTS}/2$, $N = 6$</p> <p>0101: $f_{SAMP} = f_{DTS}/2$, $N = 8$</p> <p>0110: $f_{SAMP} = f_{DTS}/4$, $N = 6$</p> <p>0111: $f_{SAMP} = f_{DTS}/4$, $N = 8$</p> <p>1000: $f_{SAMP} = f_{DTS}/8$, $N = 6$</p> <p>1001: $f_{SAMP} = f_{DTS}/8$, $N = 8$</p> <p>1010: $f_{SAMP} = f_{DTS}/16$, $N = 5$</p> <p>1011: $f_{SAMP} = f_{DTS}/16$, $N = 6$</p> <p>1100: $f_{SAMP} = f_{DTS}/16$, $N = 8$</p> <p>1101: $f_{SAMP} = f_{DTS}/32$, $N = 5$</p> <p>1110: $f_{SAMP} = f_{DTS}/32$, $N = 6$</p> <p>1111: $f_{SAMP} = f_{DTS}/32$, $N = 8$</p>
7	MSM	<p>主-从模式</p> <p>在这种模式下，触发输入事件会延迟一点儿才生效，以实现当前定时器通过 TRGO 与其从定时器之间的完美同步。假如需要将多个定时器同步到一个外部事件上，那么可以使用这种模式。</p> <p>0: 禁用主-从模式</p> <p>1: 使能主-从模式</p>

位/位域	名称	描述
6:4	TRGS	<p>从模式控制</p> <p>000: 禁用模式。当 CNTE 置位后，预分频器直接由内部时钟供给。</p> <p>001: 正交解码模式 1。计数器按照 TI2FP2 的边沿计数，计数方向取决于 TI1FP1 的电平。</p> <p>010: 正交解码模式 2。计数器按照 TI1FP1 的边沿计数，计数方向取决于 TI2FP2 的电平。</p> <p>011: 正交解码模式 3。计数器按照 TI1FP1 和 TI2FP2 的边沿计数，计数方向则取决于彼此。</p> <p>100: 重启模式。当选定的触发输入信号出现上升沿时，计数器将重新初始化，所有影子寄存器将更新。</p> <p>101: 暂停模式。触发输入为高电平时，使能计数器时钟；触发输入为低电平时，禁用计数器时钟。</p> <p>110: 触发模式。触发输入信号的上升沿将使能计数器。此时计数器是不能被从模式控制器禁用的。</p> <p>111: 外部时钟模式 1。计数器按照指定的触发输入信号的上升沿计数。由于每当 TI1F 跳变时 TI1F_ED 都会输出 1 个脉冲，并且边沿驱动模式会检查触发信号的电平状态，因此当 TI1F_ED 被选作触发输入信号时，不得使用边沿驱动模式。</p>

从定时器	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TM2	定时器 1	保留	定时器 3	定时器 4
TM3	定时器 1	定时器 2	保留	定时器 4
TM4	定时器 1	定时器 2	定时器 3	保留

3	保留	请保持其复位状态
2:0	SMC	<p>从模式控制</p> <p>000: 禁用模式。预分频器直接由内部时钟供给，当 CNTE 置位后。</p> <p>001: 正交解码模式 1。计数器按照 TI2FP2 的边沿计数，计数方向取决于 TI1FP1 的电平。</p> <p>010: 正交解码模式 2。计数器按照 TI1FP1 的边沿计数，计数方向取决于 TI2FP2 的电平。</p> <p>011: 正交解码模式 3。计数器按照 TI1FP1 和 TI2FP2 的边沿计数，计数方向则取决于彼此。</p> <p>100: 重启模式。当选定的触发输入信号上升沿时，计数器将重新初始化，所有影子寄存器将更新。</p> <p>101: 暂停模式。触发输入将使得计数器时钟，当其为高电平时，低电平时则禁用计数器。</p> <p>110: 触发模式。触发输入信号的上升沿，将使能计数器。此时计数器是不能被从模式控制器禁用的。</p> <p>111: 外部时钟模式 1。计数器按照指定的触发输入信号的上升沿计数。由于 TI1F_ED 输出 1 脉冲对每个 TI1F 的跳变沿，并且沿触发模式检查触发信号的电平状态，当 TI1F_ED 被选作触发输入信号时，不得使用沿触发模式。</p>

4.4 TMx DMA 及中断使能寄存器 (TMx_DIE)

地址偏移: 0x0C

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	TRGDE	保留	CC4DE	CC3DE	CC2DE	CC1DE	UDE	保留	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
类型		RW		RW	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW

位/位域	名称	描述
15	保留	请保持其复位状态
14	TRGDE	触发中断使能 0: 禁用触发中断 1: 使能触发中断
13	保留	请保持其复位状态
12	CH4DE	通道 4 DMA 请求使能 0: 禁用通道 4 DMA 请求 1: 使能通道 4 DMA 请求
11	CH3DE	通道 3 DMA 请求使能 0: 禁用通道 3 DMA 请求 1: 使能通道 3 DMA 请求
10	CH2DE	通道 2 DMA 请求使能 0: 禁用通道 2 DMA 请求 1: 使能通道 2 DMA 请求
9	CH1DE	通道 1 DMA 请求使能 0: 禁用通道 1 DMA 请求 1: 使能通道 1 DMA 请求
8	UPDE	计数器更新 DMA 请求使能 0: 禁用计数器更新 DMA 请求 1: 使能计数器更新 DMA 请求
7	保留	请保持其复位状态
6	TRGIE	触发中断使能 0: 禁用触发中断 1: 使能触发中断
5	保留	请保持其复位状态
4	CH4IE	通道 4 中断使能 0: 禁用通道 4 中断 1: 使能通道 4 中断
3	CH3IE	通道 3 中断使能 0: 禁用通道 3 中断 1: 使能通道 3 中断
2	CH2IE	通道 2 中断使能 0: 禁用通道 2 中断 1: 使能通道 2 中断

位/位域	名称	描述
1	CH1IE	通道 1 中断使能 0: 禁用通道 1 中断 1: 使能通道 1 中断
0	UPIE	计数器更新中断使能 0: 禁用计数器更新中断 1: 使能计数器更新中断

4.5 TMx 状态寄存器 (TMx_STA)

地址偏移: 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CH4OF	CH3OF	CH2OF	CH1OF	保留		TRGIF	保留	CH4IF	CH3IF	CH2IF	CH1IF	UPIF	
		RC_W0	RC_W0	RC_W0	RC_W0			RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

类型

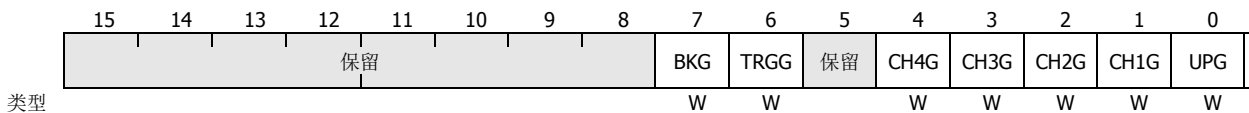
位/位域	名称	描述
31:13	保留	请保持其复位状态
12	CH4OF	通道 4 过捕获标志 参见 CH1OF 的描述。
11	CH3OF	通道 3 过捕获标志 参见 CH1OF 的描述。
10	CH2OF	通道 2 过捕获标志 参见 CH1OF 的描述。
9	CH1OF	通道 1 过捕获标志 若通道 1 配置为输入模式, 当发生捕获事件且 CH1IF 标志已经置位的情况下, 本标志将由硬件置位。本标志由软件清零。 0: 未发生过捕获中断 1: 发生过捕获中断
8:7	保留	请保持其复位状态
6	TRGIF	触发输入中断 当触发事件产生时, 本标志将由硬件置位。本标志由软件清零。在除边沿驱动之外的所有模式下, 当从模式控制器使能后, TRGI 输入的有效跳变沿将产生一个触发事件。在边沿驱动模式下, 当从模式控制器使能后, TRGI 输入的任何跳变沿都将产生触发事件。 0: 未发生触发事件 1: 已产生触发中断
5	保留	请保持其复位状态
4	CH4IE	通道 4 中断使能 参见 CH1IE 的描述。
3	CH3IE	通道 3 中断使能 参见 CH1IE 的描述。

位/位域	名称	描述
2	CH2IE	通道 2 中断使能 参见 CH1IE 的描述。
1	CH1IE	通道 1 中断使能 本标志由硬件置位，由软件清零。若通道 1 处于输入模式，则当产生捕获时间时，本标志将置位。若通道 1 处于输出模式，则当产生比较匹配事件时，本标志将置位。 0: 未产生通道 1 中断 1: 已产生通道 1 中断
0	UPIF	更新中断标志 本标志由硬件置位，当发生更新事件时。本标志由软件清零。 0: 未发生更新中断 1: 已发生更新中断

4.6 TMx 事件产生寄存器 (TMx_EVG)

地址偏移: 0x14

复位值 : 0x0000



位/位域	名称	描述
31:8	保留	请保持其复位状态
7	BKG	紧急停止事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，POE 将清零，BKIF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。 0: 不产生紧急停止事件 1: 产生紧急停止事件
6	TRGG	触发事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，TMx_STA 寄存器的 TRGIF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。 0: 不产生触发事件 1: 产生触发事件
5	保留	请保持其复位状态
4	CH4G	通道 4 捕获/比较事件产生 参见 CH1G 的描述。
3	CH3G	通道 3 捕获/比较事件产生 参见 CH1G 的描述。
2	CH2G	通道 2 捕获/比较事件产生 参见 CH1G 的描述。

位/位域	名称	描述
1	CH1G	通道 1 捕获/比较事件产生 本位由软件置位，用以在通道 1 产生一个捕获/比较事件；本位由硬件自动清零。当对本位写入 1 时，CC1IF 标志将置位，相应的中断或 DMA 请求如果被使能则将产生。此外，假如通道 1 配置为输入模式，则计数器的当前值将被捕获到 TMx_CHCC1 寄存器中，若 CH1IF 标志已经置位，还会将 CH1OF 标志置位。 0: 不产生通道 1 捕获/比较事件 1: 产生通道 1 捕获/比较事件
0	UPG	更新事件产生 本位由软件置位，由硬件自动清零。当对本位写入 1 时，若选择的工作模式是中心对齐模式或向上计数模式，则计数器将被清零；否则（向下计数模式）计数器将会装载重载值。同时，预分频计数器也会被清零。 0: 不产生更新事件 1: 产生更新事件

4.7 TMx 通道控制寄存器 1 (TMx_CHCTRL1)

地址偏移: 0x18

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2 OCE	CH2OM[2:0]			CH2 OSE	CH2 OFE	CH2M[1:0]		CH1 OCE	CH1OM[2:0]			OC1 OSE	CH1 OFE	CH1M[1:0]	
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

输出比较模式:

位/位域	名称	描述
15	CH2OCE	通道 2 输出比较清除使能 参见 CH1OCE 的描述。
14:12	CH2OM	通道 2 输出比较模式 参见 CH1OM 的描述。
11	CH2OSE	通道 2 输出比较影子寄存器使能 参见 CH1OSE 的描述。
10	CH2OFE	通道 2 输出比较快速使能 参见 CH1OFE 的描述。
9:8	CH2M	通道 2 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH2E 位为 0）时，本位域才可写。 00: 通道 2 配置为输出 01: 通道 2 配置为输入，IC2 映射到 TI2 10: 通道 2 配置为输入，IC2 映射到 TI1 11: 通道 2 配置为输入，IC2 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

位/位域	名称	描述
7	CH1OCE	通道 1 输出比较清除使能 当本位置位时，当 ETRF 输入检测到高电平时将会清除 OC1REF 信号。 0: 禁用通道 1 输出比较清除 1: 使能通道 1 输出比较清除
6:4	CH1OM	通道 1 输出比较模式 本位域指定输出用于驱动 OC1 和 OC1N 的参考信号 OC1REF 的行为方式。OC1REF 是高电平有效的，而 OC1 和 OC1N 的有效电平则取决于 CH1P 和 CH1NP 位域的设置。 000: 冻结。OC1REF 信号保持固定电平，不管输出比较寄存器 TMx_CHCC1 以及计数值的比较结果如何。 001: 匹配时拉高。当计数值与输出比较寄存器 TMx_CHCC1 匹配时，OC1REF 信号将强制拉高。 010: 匹配时拉低。当计数值与输出比较寄存器 TMx_CHCC1 匹配时，OC1REF 信号将强制拉低。 011: 匹配时翻转。当计数值与输出比较寄存器 TMx_CHCC1 匹配时，OC1REF 信号将翻转。 100: 强制拉低。OC1REF 信号始终为低电平。 101: 强制拉高。OC1REF 信号始终为高电平。 110: PWM 模式 1。当向上计数时，只要计数值小于 TMx_CHCC1，OC1REF 为高电平，否则为低电平。当向下计数时，只要计数值大于 TMx_CHCC1，OC1REF 为低电平，否则为高电平。 111: PWM 模式 2。当向上计数时，只要计数值小于 TMx_CHCC1，OC1REF 为低电平，否则为高电平。当向下计数时，只要计数值大于 TMx_CHCC1，OC1REF 为高电平，否则为低电平。 当配置为 PWM 模式时，仅当输出比较模式从“冻结”切换到“PWM”模式，或当比较结果发生变化时，OCREF 电平才会变化。 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11、且 CH1M 位域为 00，本位域不得更改。
3	CH1OSE	通道 1 输出比较影子寄存器使能 当本位置位时，TMx_CHCC1 寄存器的影子寄存器将会使能，并在每个更新事件时得到更新。 0: 禁用通道 1 输出比较影子寄存器 1: 使能通道 1 输出比较影子寄存器 仅在单脉冲模式下（TMx_CTRL1 寄存器的 OPM 置位），PWM 模式无需使能影子寄存器也可使用。 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11、且 CH1M 位域为 00，本位域不得更改。
2	CH1OFE	通道 1 快速比较输出使能 当本位置位时，假如通道配置为 PWM1 或 PWM2 模式，则触发输入事件在 CC 输出上的效果会被加速。输出通道会将触发输入信号的有效沿视为比较匹配，OC 将会变成与比较结果无关的电平。 0: 禁用通道 1 快速比较输出。触发输入的边沿到 CC1 有效输出的最短延时为 5 个时钟周期。 1: 使能通道 1 快速比较输出。触发输入的边沿到 CC1 有效输出的最短延时为 3 个时钟周期。
1:0	CH1M	通道 1 模式选择 本位域指定通道的方向，以及选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH1E 位为 0）时，本位域才可写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入，IC1 映射到 TI1 10: 通道 1 配置为输入，IC1 映射到 TI2 11: 通道 1 配置为输入，IC1 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

输入捕获模式:

位/位域	名称	描述
15:12	CH2ICF	通道 2 输入捕获滤波控制 参见 CH1ICF 的描述。
11:10	CH2ICP	通道 2 输入捕获预分频系数 参见 CH1ICP 的描述。
9:8	CH2M	通道 2 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH2E 位为 0）时，本位域才可写， 00: 通道 2 配置为输出 01: 通道 2 配置为输入，IC2 映射到 TI2 10: 通道 2 配置为输入，IC2 映射到 TI1 11: 通道 2 配置为输入，IC2 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。
7:4	CH1ICF	通道 1 输入捕获滤波控制 数字滤波器中采用了一个事件计数器，经过 N 次输入事件后，输出才会产生跳变。本位域用于指定用于采样 TI1 输入信号的频率，以及用于 TI1 的数字滤波器的长度。 0000: 禁用数字滤波。 $f_{SAMP} = f_{CK_INT}$, N = 1 0001: $f_{SAMP} = f_{CK_INT}$, N = 2 0010: $f_{SAMP} = f_{CK_INT}$, N = 4 0011: $f_{SAMP} = f_{CK_INT}$, N = 8 0100: $f_{SAMP} = f_{DTS}/2$, N = 6 0101: $f_{SAMP} = f_{DTS}/2$, N = 8 0110: $f_{SAMP} = f_{DTS}/4$, N = 6 0111: $f_{SAMP} = f_{DTS}/4$, N = 8 1000: $f_{SAMP} = f_{DTS}/8$, N = 6 1001: $f_{SAMP} = f_{DTS}/8$, N = 8 1010: $f_{SAMP} = f_{DTS}/16$, N = 5 1011: $f_{SAMP} = f_{DTS}/16$, N = 6 1100: $f_{SAMP} = f_{DTS}/16$, N = 8 1101: $f_{SAMP} = f_{DTS}/32$, N = 5 1110: $f_{SAMP} = f_{DTS}/32$, N = 6 1111: $f_{SAMP} = f_{DTS}/32$, N = 8
3:2	CH1ICP	通道 1 输入捕获预分频系数 本位域指定通道 1 输入的预分频系数。若 TMx_CHE 寄存器的 CH1E 位被清零，则预分频器将会复位。 00: 禁用预分频器，在通道输入信号每个跳变沿时捕获 01: 通道输入信号每 2 个跳变沿时捕获 10: 通道输入信号每 4 个跳变沿时捕获 11: 通道输入信号每 8 个跳变沿时捕获
1:0	CH1M	通道 1 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH1E 位为 0）时，本位域才可写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入，IC1 映射到 TI1 10: 通道 1 配置为输入，IC1 映射到 TI2 11: 通道 1 配置为输入，IC1 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

4.8 TMx 通道控制寄存器 2 (TMx_CHCTRL2)

地址偏移: 0x1C

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH4 OCE	CH4OM[2:0]			CH4 OSE	CH4 OFE	CH4M[1:0]		CH3 OCE	CH3OM[2:0]			OC3 OSE	CH3 OFE	CH3M[1:0]	
类型	CH4ICF[3:0]			CH4ICP[1:0]				CH3ICF[3:0]			CH3ICP[1:0]					
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

输出比较模式:

位/位域	名称	描述
31:16	保留	请保持其复位状态
15	CH4OCE	通道 4 输出比较清除使能 参见 CH1OCE 的描述。
14:12	CH4OM	通道 4 输出比较模式 参见 CH1OM 的描述。
11	CH4OSE	通道 4 输出比较影子寄存器使能 参见 CH1OSE 的描述。
10	CH4OFE	通道 4 输出比较快速使能 参见 CH1OFE 的描述。
9:8	CH4M	通道 4 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭 (即 TMx_CHE 寄存器的 CH4E 位为 0) 时, 本位域才可写。 00: 通道 4 配置为输出 01: 通道 4 配置为输入, IC4 映射到 TI4 10: 通道 4 配置为输入, IC4 映射到 TI3 11: 通道 4 配置为输入, IC4 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时, 这种模式才能工作。
7	CH3OCE	通道 3 输出比较清除使能 参见 CH1OCE 的描述。
6:4	CH3OM	通道 3 输出比较模式 参见 CH1OM 的描述。
3	CH3OSE	通道 3 输出比较影子寄存器使能 参见 CH1OSE 的描述。
2	CH3OFE	通道 3 输出比较快速使能 参见 CH1OFE 的描述。
1:0	CH3M	通道 3 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭 (即 TMx_CHE 寄存器的 CH3E 位为 0) 时, 本位域才可写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入, IC3 映射到 TI3 10: 通道 3 配置为输入, IC3 映射到 TI4 11: 通道 3 配置为输入, IC3 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时, 这种模式才能工作。

输入捕获模式:

位/位域	名称	描述
15:12	CH4ICF	通道 4 输入捕获滤波控制 参见 CH1ICF 的描述。
11:10	CH4ICP	通道 4 输入捕获预分频系数 参见 CH1ICP 的描述。
9:8	CH4M	通道 4 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH4E 位为 0），本位域才可写。 00: 通道 4 配置为输出 01: 通道 4 配置为输入，IC4 映射到 TI4 10: 通道 4 配置为输入，IC4 映射到 TI3 11: 通道 4 配置为输入，IC4 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。
7:4	CH3ICF	通道 3 输入捕获滤波控制 参见 CH1ICF 的描述。
3:2	CH3ICP	通道 3 输入捕获预分频系数 参见 CH1ICP 的描述。
1:0	CH3M	通道 3 模式选择 本位域指定通道的方向并选择输入信号。仅当本通道关闭（即 TMx_CHE 寄存器的 CH3E 位为 0）时，本位域才可写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入，IC3 映射到 TI3 10: 通道 3 配置为输入，IC3 映射到 TI4 11: 通道 3 配置为输入，IC3 映射到 TRC。仅当通过 TM_SMC 寄存器的 TRGS 位域选择内部触发输入时，这种模式才能工作。

4.9 TMx 通道使能寄存器 (TMx_CHE)

地址偏移: 0x20

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH4P	CH4E	保留	CH3P	CH3E	保留	CH2P	CH2E	保留	CH1P	CH1E				
	RW	RW		RW	RW		RW	RW		RW	RW				

类型

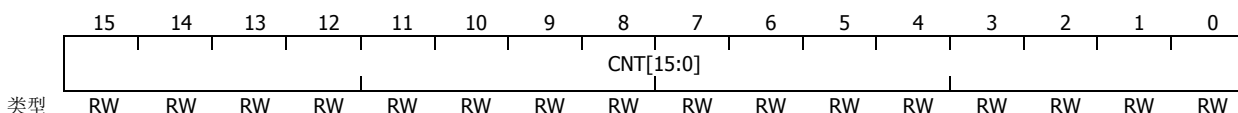
位/位域	名称	描述
31:14	保留	请保持其复位状态
13	CH4P	通道 4 极性 参见 CH1P 的描述。
12	CH4E	通道 4 使能 参见 CH1E 的描述。
11:10	保留	请保持其复位状态
9	CH3P	通道 3 极性 参见 CH1P 的描述。

位/位域	名称	描述
8	CH3E	通道 3 使能 参见 CH1E 的描述。
7:6	保留	请保持其复位状态
5	CH2P	通道 2 极性 参见 CH1P 的描述。
4	CH2E	通道 2 使能 参见 CH1E 的描述。
3:2	保留	请保持其复位状态
1	CH1P	通道 1 极性 当通道 1 配置为输入模式时，本位指定 IC1 信号的极性。当通道 1 配置为输出模式时，本位指定输出信号的极性。 0: 通道 1 高电平有效 1: 通道 1 低电平有效 若 TMx_BKDT 寄存器的 LK[1:0]位域为 11 或 10 时，本位域不得更改。
0	CH1E	通道 1 使能 当通道 1 配置为输入模式时，对本位写 1 即可将 OC1 信号置为有效状态。当通道 1 配置为输出模式时，对本位写 1 即可使能通道 1 的捕获事件。 0: 禁用通道 1 1: 使能通道 1

4.10 TMx 计数寄存器 (TMx_CNT)

地址偏移: 0x24

复位值 : 0x0000

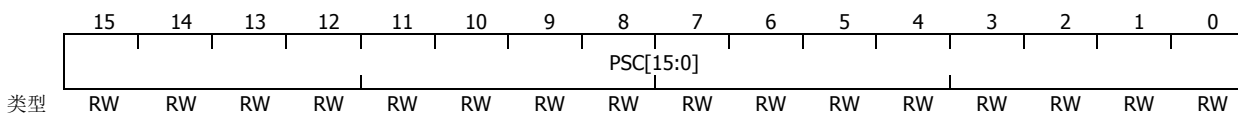


位/位域	名称	描述
15:0	CNT	本位域代表当前计数值。对本位域执行写操作，即可更改计数值。

4.11 TMx 预分频寄存器 (TMx_PSC)

地址偏移: 0x28

复位值 : 0x0000



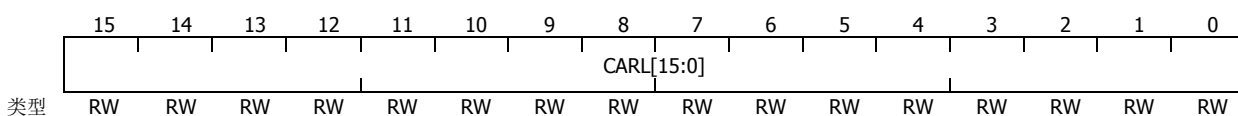
位/位域	名称	描述
------	----	----

位/位域	名称	描述
15:0	PSC	计数器时钟的预分频系数 PSC 时钟将除以(PSC+1)以获得计数器时钟。本位域的值将在每次产生更新事件时装载到相应的影子寄存器中。

4.12 TMx 计数器自动重载寄存器 (TMx_CARL)

地址偏移: 0x2C

复位值 : 0x0000

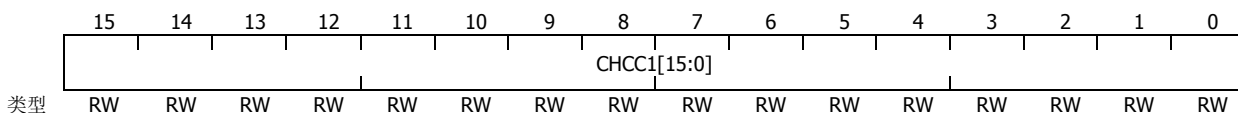


位/位域	名称	描述
15:0	CARL	计数器自动重载值 本位域指定计数器的自动重载值。

4.13 TMx 通道 1 捕获/比较寄存器 (TMx_CHCC1)

地址偏移: 0x34

复位值 : 0x0000

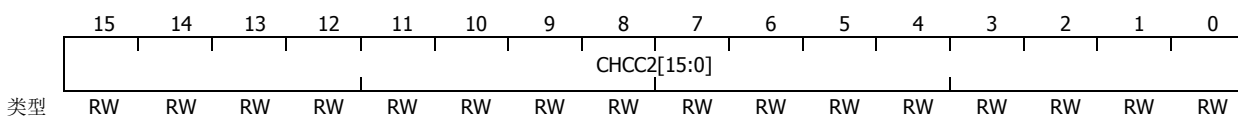


位/位域	名称	描述
15:0	CHCC1	通道 1 的捕获/比较值 当通道 1 配置为输入模式时, 本位域用于指示上一次捕获事件的计数值。 当通道 1 配置为输出模式时, 本位域的值将与计数值进行比较。当相应的影子寄存器被使能时, 影子寄存器的内容将会在每次更新事件时更新。

4.14 TMx 通道 2 捕获/比较寄存器 (TMx_CHCC2)

地址偏移: 0x38

复位值 : 0x0000

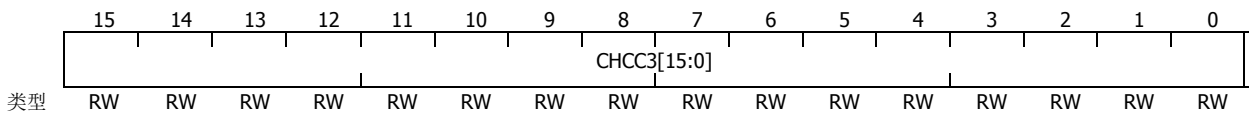


位/位域	名称	描述
15:0	CHCC2	通道 2 的捕获/比较值 当通道 2 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 2 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

4.15 TMx 通道 3 捕获/比较寄存器 (TMx_CHCC3)

地址偏移: 0x3C

复位值 : 0x0000

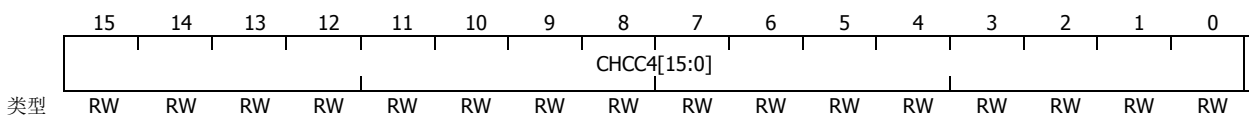


位/位域	名称	描述
15:0	CHCC3	通道 3 的捕获/比较值 当通道 3 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 3 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

4.16 TMx 通道 4 捕获/比较寄存器 (TMx_CHCC4)

地址偏移: 0x40

复位值 : 0x0000

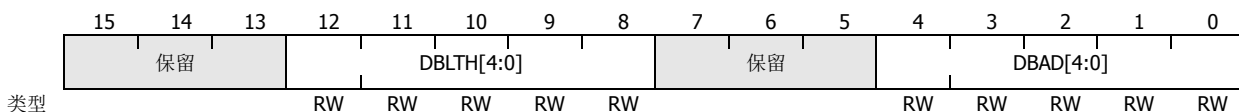


位/位域	名称	描述
15:0	CHCC4	通道 4 的捕获/比较值 当通道 4 配置为输入模式时，本位域用于指示上一次捕获事件的计数值。 当通道 4 配置为输出模式时，本位域的值将与计数值进行比较。当相应的影子寄存器被使能时，影子寄存器的内容将会在每次更新事件时更新。

4.17 TMx DMA 控制寄存器 (TMx_DCTRL)

地址偏移: 0x48

复位值 : 0x0000

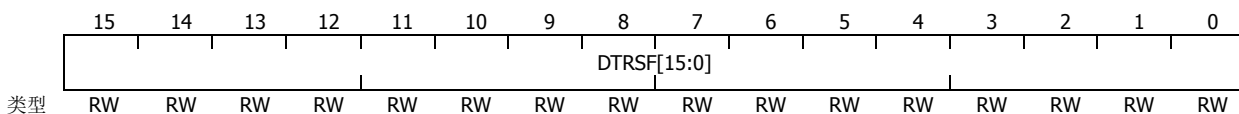


位/位域	名称	描述
31:13	保留	请保持其复位状态
12:8	DBLTH	DMA 访问猝发长度 当寄存器访问是通过 TMx_DTRSF 地址实现的, 则本位域指定猝发传输的次数。
7:5	保留	请保持其复位状态
4:0	DBAD	DMA 访问基地址 当寄存器访问是通过 TMx_DTRSF 地址实现的, 则本位域指定相较于 TMx_CTRL1 寄存器的地址偏移量。

4.18 TMx DMA 传输寄存器 (TMx_DTRSF)

地址偏移: 0x4C

复位值 : 0x0000



位/位域	名称	描述
15:0	DTRSF	DMA 传输 当对此地址进行读写操作时, 实际将会访问如下地址: (DBAD+猝发计数器)×4 猝发计数器由硬件计算, 其范围为 0~DBLTH。

第8章 实时时钟 RTC

1 简介

实时时钟（RTC）本质上是一个独立的定时器，其计数器通常用于实现日历时钟。RTC 电路分属于两个电源域：一部分在备电域中，该部分包括一个 32 位的累加计数器、一个闹钟（警报触发器）、一个预分频器、一个分频器以及 RTC 时钟配置寄存器。这就意味着系统复位或者从待机模式唤醒时，RTC 的设置以及时间都仍然保持。另一部分在 V_{DD} 电源域中，该部分包括 APB1 接口以及一个控制寄存器。我们将在本章接下来的部分详细介绍 RTC 的功能。

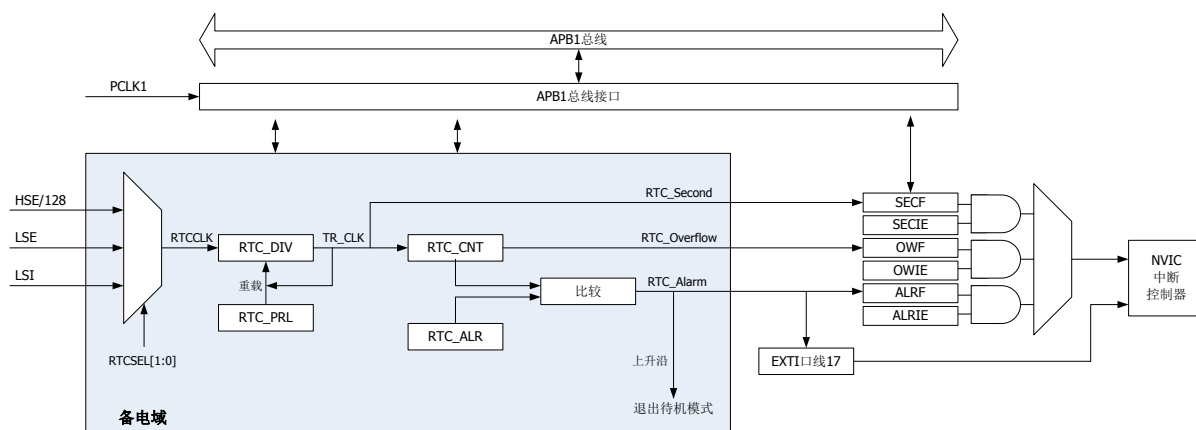
2 主要特点

- 32 位可编程计数器，用于计数已流逝的时间
- 可编程的预分频器：
 - 分频系数最高可达 2^{30}
- 两个独立的时钟
 - PCLK1 时钟
 - RTC 时钟（必须至少比 PCLK1 时钟慢 4 倍）
- RTC 时钟源可选
 - HSE 时钟除以 128
 - LSE 振荡电路时钟
 - LSI 振荡电路时钟
- 三个 RTC 中断线
 - 闹钟中断
 - 秒中断
 - 溢出中断

3 功能描述

下图描绘出 RTC 模块的内部配置。

图 92 RTC 内部框图



3.1 RTC 复位

APB1 总线接口以及 RTC 控制寄存器都随系统复位而复位。RTC 内核（预分频器、分频器、计数器以及闹钟模块）仅随备电域复位而复位。

要想在复位后访问备电域寄存器以及 RTC 寄存器，应按如下步骤操作：

1. 将 RCC_APB1CCR 寄存器的 PWREN 以及 BKPEN 置位，使能电源以及备用域接口时钟；
2. 将电源控制寄存器（PWR_CR）的 DBP 置位，使能对备电域寄存器和 RTC 的访问。

3.2 RTC 寄存器的读取

RTC 内核以及 RTC APB1 总线接口分处于两个不同的电源域。

在 RTC 内核中，只有计数器和分频器这两个寄存器是可读的寄存器。这两个寄存器的数值以及 RTC 标志在每个 RTC 时钟的上升沿都会内部更新，并按照 APB1 时钟重新同步。

当 APB1 总线接口从禁用状态使能之后，建议不要马上进行读操作，因为这些寄存器的首次内部更新过程可能尚未完成。这是因为，在系统复位、电源复位、从待机/停机模式唤醒时，APB1 总线接口是被禁用的，但 RTC 内核仍然保持运行。在这类情况下，要想进行正确的读操作，应先将 RTC 控制寄存器中的 RSF 位清零并等待其被硬件置位。WFI 和 WFE 指令对于 RTC 的 APB1 总线接口没有影响。

3.3 RTC 寄存器的配置

RTC 内核中的预分频器、计数器以及闹钟寄存器都是可写的。不过，仅当外设进入了配置模式后才允许设置这些寄存器。通过查询 RTC 控制寄存器的 CMF 标志位，可以检测配置模式的状态。当外设退出配置模式后，之前对这些寄存器写入的内容才开始生效，且需要至少 3 个 RTCCLK 周期才能完成。当写操作完成后，RTC 控制寄存器的 LWOFF 位将会变成 1。任何写操作必须等待前一次写操作完成之后再行。

配置的步骤如下：

1. 等待 LWOFF 位的值变为 1；
2. 将 CMF 置位，进入配置模式；
3. 写 RTC 寄存器；
4. 将 CMF 清零，退出配置模式；
5. 等待 LWOFF 位的值变为 1。

3.4 RTC 的标志产生

RTC 的秒标志是在 RTC 计数器更新的前一个 RTCCLK 周期产生的。

RTC 的闹钟标志是在计数值达到 RTC 闹钟值（存储于闹钟寄存器中）加 1 的前一个 RTCCLK 周期产生的。

RTC 溢出标志是在计数器到达 0x0 的一个 RTCCLK 周期产生的。

要改写 RTC 闹钟标志和秒标志，必须执行同步写操作。此时应按照以下两个序列执行：

- 启用 RTC 闹钟中断，在 RTC 中断服务子程序内更新 RTC 闹钟寄存器及/或 RTC 计数寄存器的内容；
- 等待 RTC 控制寄存器的 SCDF 置位之后，才能更新 RTC 闹钟寄存器及/或 RTC 计数寄存器的内容。

图 93 RTC 秒信号及闹钟信号的波形（PR = 3, ALARM = 2）

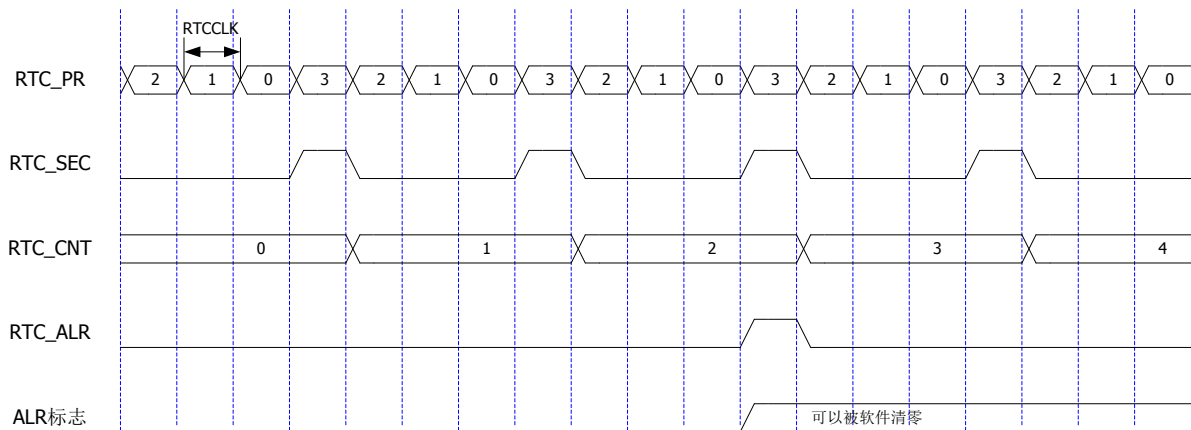
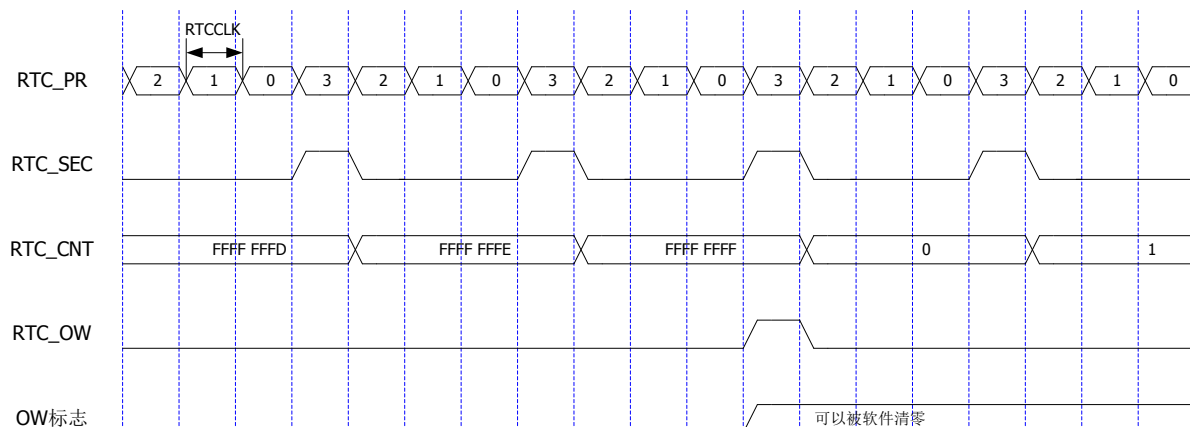


图 94 RTC 秒信号及溢出信号的波形 (PR = 3)



4 RTC 寄存器

4.1 RTC 控制寄存器 1 (RTC_CTRL1)

地址偏移: 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													OFWIE	ALMIE	SCDIE
													RW	RW	RW

类型

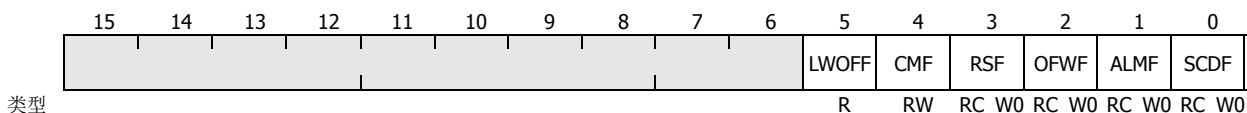
位/位域	名称	描述
15:3	保留	请保持其复位状态
2	OFWIE	溢出中断使能 0: 禁用溢出中断 1: 使能溢出中断
1	ALMIE	闹钟中断使能 0: 禁用闹钟中断 1: 使能闹钟中断
0	SCDIE	秒中断使能 0: 禁用秒中断 1: 使能秒中断

在 RTC_CTRL1 寄存器的前一次写操作完成之前, 不得再次改写该寄存器。RTC 中断请求是由这些控制位控制的。所有中断在复位后默认都是被禁用的。在修改 RTC_CTRL1 寄存器中的位之前, 请先将 RTC_CTRL2 寄存器中的对应标志清零, 以确保没有正在挂起的中断请求。

4.2 RTC 控制寄存器 2 (RTC_CTRL2)

地址偏移: 0x04

复位值 : 0x0020

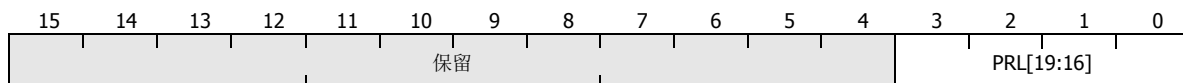


位/位域	名称	描述
15:6	保留	请保持其复位状态
5	LWOFF	前一次写操作完成标志 0: 对 RTC 寄存器的前一次写操作尚未完成 1: 对 RTC 寄存器的前一次写操作已完成 本标志位只读。
4	CMF	配置模式标志 0: 退出配置模式 1: 进入配置模式
3	RSF	寄存器同步标志 0: 寄存器尚未与 APB1 时钟同步 1: 寄存器已与 APB1 时钟同步 本标志位可读, 写入时只可写 0。
2	OFWF	溢出标志 0: 未检测到溢出事件 1: 检测到溢出事件。假如 RTC_CTRL1 寄存器的 OFWIE 置位, 则会产生中断。 本标志位在计数器溢出时由硬件自动置位。 本标志位可读, 写入时只可写 0。
1	ALMF	闹钟标志 0: 未检测到闹钟事件 1: 检测到闹钟事件。假如 RTC_CTRL1 寄存器的 ALMIE 置位, 则会产生中断。 假如 EXTI17 被使能中断模式, 则会产生 RTC 闹钟中断。 本标志位在计数值到达 RTC_ALM 时由硬件自动置位。 本标志位可读, 写入时只可写 0。
0	SCDF	秒标志 0: 未检测到秒事件 1: 检测到秒事件。假如 RTC_CTRL1 寄存器的 SCDIE 置位, 则会产生中断。 本标志位在分频器重载 RTC_LD 中的值时, 由硬件自动置位。因此会使得 RTC 计数器累加。 本标志位可读, 写入时只可写 0。

4.3 RTC 预分频装载寄存器 1 (RTC_LD1)

地址偏移: 0x08

复位值 : 0x0000

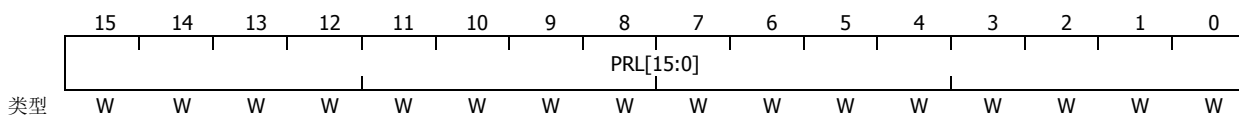


类型			W	W	W	W
位/位域	名称	描述				
15:4	保留	请保持其复位状态				
3:0	PRL[19:16]	RTC 预分频系数高 4 位				

4.4 RTC 预分频装载寄存器 2 (RTC_LD2)

地址偏移: 0x0C

复位值 : 0x0000

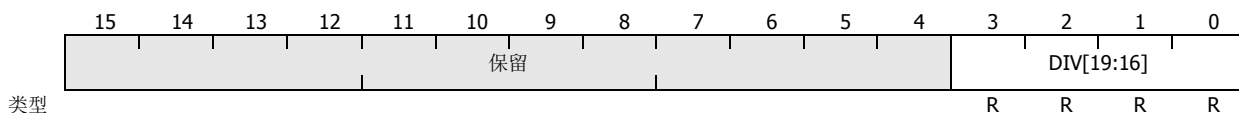


位/位域	名称	描述
15:0	PRL[15:0]	RTC 预分频系数低 16 位 TR_CLK 的时钟频率为 RTCCLK 频率除以(PRL[19:0]+1)。 32768 = 0x8000

4.5 RTC 预分频计数寄存器 1 (RTC_DIV1)

地址偏移: 0x10

复位值 : 0x0000

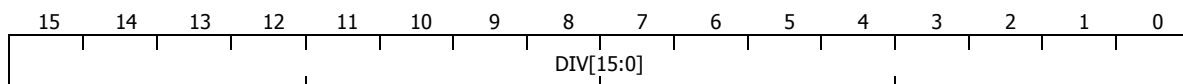


位/位域	名称	描述
15:4	保留	请保持其复位状态
3:0	DIV[19:16]	RTC 分频计数值高 4 位

4.6 RTC 域分频计数寄存器 2 (RTC_DIV2)

地址偏移: 0x14

复位值 : 0x8000



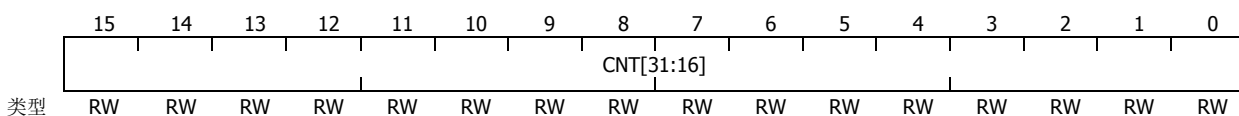
类型	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

位/位域	名称	描述
15:0	PRL[15:0]	RTC 分频计数值低 16 位 当 RTC 预分频系数值或者 RTC 计数寄存器更新时，RTC 分频计数值将由硬件自动重载。

4.7 RTC 计数寄存器 1 (RTC_CNT1)

地址偏移: 0x18

复位值 : 0x0000



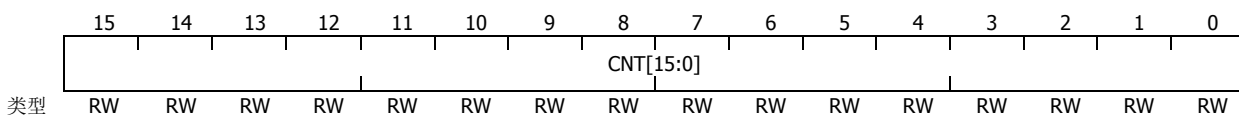
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

位/位域	名称	描述
15:0	CNT[31:16]	RTC 计数值高半字

4.8 RTC 计数寄存器 2 (RTC_CNT2)

地址偏移: 0x1C

复位值 : 0x0000



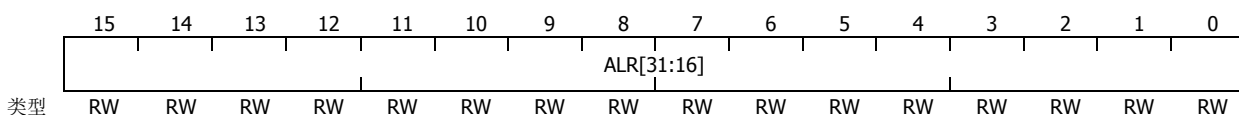
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

位/位域	名称	描述
15:0	CNT[15:0]	RTC 计数值低半字

4.9 RTC 闹钟寄存器 1 (RTC_ALM1)

地址偏移: 0x20

复位值 : 0xFFFF



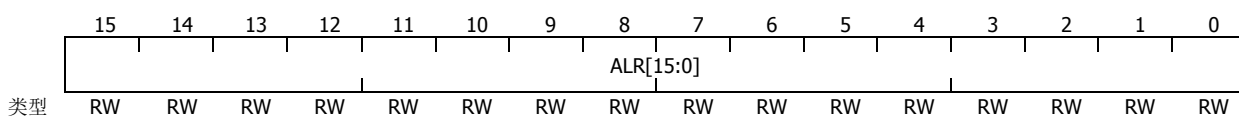
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

位/位域	名称	描述
15:0	ALR[31:16]	RTC 闹钟值高半字

4.10 RTC 闹钟寄存器 2 (RTC_ALM2)

地址偏移: 0x24

复位值 : 0xFFFF



位/位域	名称	描述
15:0	ALR[15:0]	RTC 闹钟值低半字

第9章 看门狗（WDG）

1 独立式看门狗（IWDG）

1.1 简介

看门狗（Watchdog，简称为 WDG）定时器是一种硬件定时电路，主要用于监测系统是否发生由软件工作异常而引发的故障。GD32F103xx 系列微控制器有两个看门狗外设，分别是独立式看门狗和窗口式看门狗。二者相互配合使用，实现了高度安全、使用灵活并且精准的软件运行状态监控。两种看门狗都是用来解决软件工作异常的。当看门狗计数器计数到特定值时，看门狗模块就会产生复位（或者由窗口式看门狗产生中断）。若处理器处于调试状态，则看门狗计数器可以设置暂停。独立式看门狗可以开启寄存器写保护功能，防止其配置被意外更改。

独立式看门狗（Independent Watchdog，简称为 IWDG）有独立的时钟源（LSI），因此即使主时钟出了问题，IWDG 仍能正常工作。这样比较适合需要独立环境、对计时精度要求不那么高的场合。与之相比，窗口式看门狗则适用于要求精准计时的场合。

窗口式看门狗（Window Watchdog，简称为 WWDG）的时钟是从 APB1 时钟分频得到的。假如软件在计数器到达指定值之前未能及时重载计数器，就会产生复位或者中断。此外，WWDG 还具有一个可配置的窗口值，假如软件在计数值达到窗口值之前重载计数器（过早重载），也会产生中断。因此软件必须在一个限定的时间窗口内重新加载计数器。

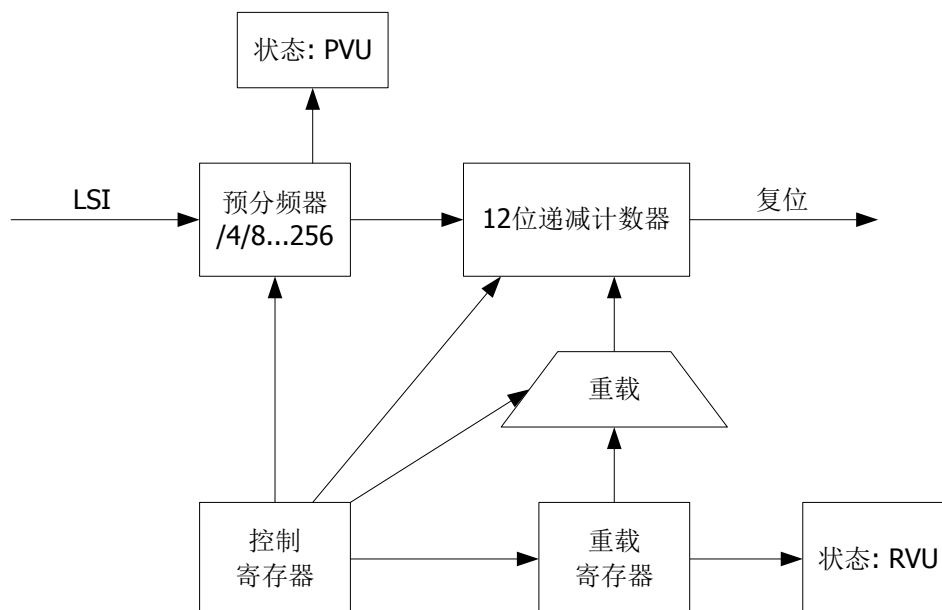
1.2 主要特点

- 自由运行的 12 位递减计数器
- 使能看门狗后，当计数值递减至 0 时产生复位
- IWDG 具有独立时钟源，即使主时钟出现问题（例如待机或停机模式）也能正常工作
- 提供硬件的独立看门狗位，决定上电时是否自动启动 IWDG
- 提供 IWDG 调试模式控制位，该位决定 IWDG 在调试模式下是继续运行还是停止

1.3 功能描述

独立式看门狗包含一个 8 级预分频器以及一个 12 位递减计数器。下图即为独立式看门狗模块的功能框图。

图 95 独立式看门狗内部框图



要使能独立式看门狗，需要向密钥寄存器（IWDG_CTR）写入 0xCCCC，之后计数器将开始递减计数。当计数值达到 0x000 时，即会产生复位。

任何时刻均可以向密钥寄存器写入 0xAAAA 来重载计数器，重载值则取自 IWDG_RLR 寄存器。软件应当在计数值达到 0x000 之前重载计数器（喂狗），防止看门狗产生复位。

若选项字中的硬件独立看门狗位被置位，则独立式看门狗在上电时自动启动。之后软件应在计数值递减到 0x000 之前及时重载计数器。

IWDG_PSR 寄存器以及 IWDG_RLR 寄存器都是写保护的。要对这些寄存器进行写操作，软件必须先向密钥寄存器写入 0x5555。写入其它值则将使得以上两个寄存器重新被写保护。预分频系数（IWDG_PSR）或重载值（IWDG_RLR）正在更新过程中，IWDG_STR 寄存器中的相关标志将会置位。

假如在调试模式下将 DBG_IWDG_STOP 位清零，那么即使 Cortex™-M3 内核停止工作（调试模式下），IWDG 也会持续工作。假如 DBG_IWDG_STOP 为 1，则 IWDG 在调试模式下将停止计数。

表 39 40kHz（LSI）频率下的 IWDG 超时范围

预分频系数	PR[2:0]	最小超时(ms) RL[11:0]=0x000	最大超时(ms) RL[11:0]=0xFFFF
1/4	000	0.1	409.6
1/8	001	0.2	819.2
1/16	010	0.4	1638.4
1/32	011	0.8	3276.8
1/64	100	1.6	6553.6

预分频系数	PR[2:0]	最小超时(ms) RL[11:0]=0x000	最大超时(ms) RL[11:0]=0xFFF
1/128	101	3.2	13107.2
1/256	110 或 111	6.4	26214.4

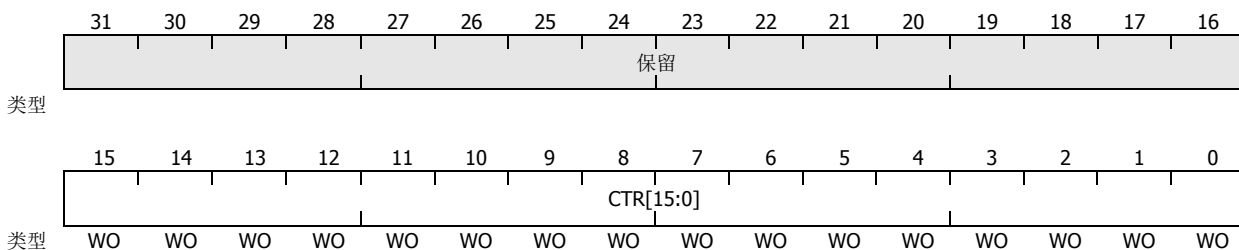
通过校准 LSI，可以得到更加精确的 IWDG 超时时间。

2 IWDG 寄存器

2.1 控制寄存器 (IWDG_CTR)

地址偏移: 0x00

复位值 : 0x0000 0000

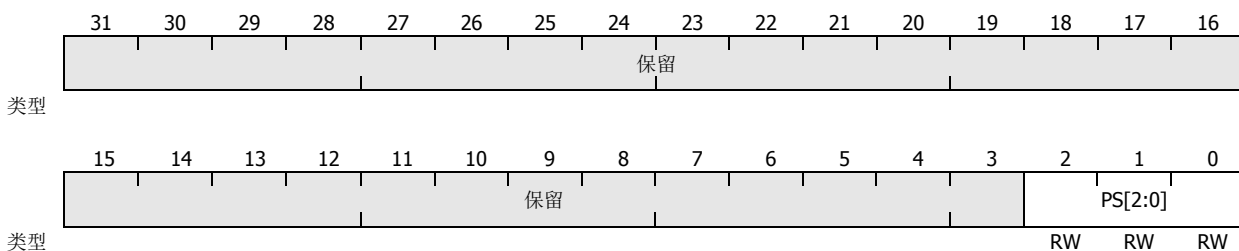


位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	CTR[15:0]	控制命令。此位域只可写，并且在写入不同数值时有不同的功能： 0x5555: 关闭 IWDG_PSR 与 IWDG_RLR 寄存器的写保护。 0xCCCC: 启动独立式看门狗计数器。当计数值递减到 0 时，独立式看门狗将产生复位。 0xAAAA: 重载计数值。

2.2 预分频系数寄存器 (IWDG_PSR)

地址偏移: 0x04

复位值 : 0x0000 0000

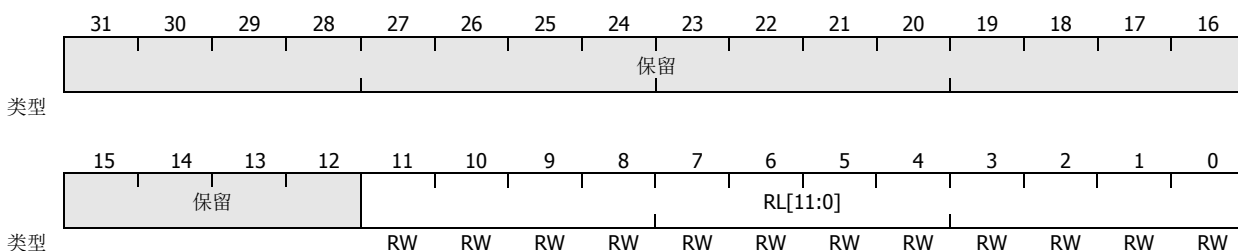


位/位域	名称	描述
31:3	保留	请保持其复位状态
2:0	PS[2:0]	独立式看门狗定时器预分频系数 对本位域操作之前, 应先对 IWDG_CTR 寄存器写入 0x5555。当对本寄存器进行写入操作时, IWDG_STR 寄存器的 PVU 将会置位, 并且在此期间对本寄存器进行读操作的结果无意义。 000: 4 分频 001: 8 分频 010: 16 分频 011: 32 分频 100: 64 分频 101: 128 分频 110/111: 256 分频

2.3 重载寄存器 (IWDG_RLR)

地址偏移: 0x08

复位值 : 0x0000 0FFF



位/位域	名称	描述
31:12	保留	请保持其复位状态
11:0	RL[11:0]	独立看门狗计数器重载值 对本位域操作之前, 应先对 IWDG_CTR 寄存器写入 0xAAAA。当对本寄存器进行写入操作时, IWDG_STR 寄存器的 RVU 将会置位, 并且在此期间对本寄存器进行读操作的结果无意义。

2.4 状态寄存器 (IWDG_STR)

地址偏移: 0x0C

复位值 : 0x0000 0000



类型			RO	RO
位/位域	名称	描述		
31:2	保留	请保持其复位状态		
1	RVU	独立式看门狗计数器重载值正在更新标志 当对 IWDG_RLR 寄存器进行写入操作时，本标志位将会置位，并且在此期间对 IWDG_RLR 寄存器进行读操作的结果无意义。		
0	PVU	独立式看门狗预分频系数值正在更新标志 当对 IWDG_PSR 寄存器进行写入操作时，本标志位将会置位，并且在此期间对 IWDG_PSR 寄存器进行读操作的结果无意义。		

3 窗口式看门狗 (WWDG)

3.1 简介

窗口式看门狗 (WWDG) 可用于监测系统是否出现因软件工作异常而导致的运行异常。启动窗口式看门狗之后，7 位的递减计数器将开始逐次递减。当计数值到达 0x3F (即第 6 位被清零) 后，就会产生复位。此外，若计数值到达窗口门限值之前被更新，WWDG 也会产生复位，因此计数器必须在一个限定的时间窗口内进行重载。窗口式看门狗还提供了一个提前唤醒中断，假如使能该中断，那么在计数值递减到 0x40 时看门狗还会产生一个中断。

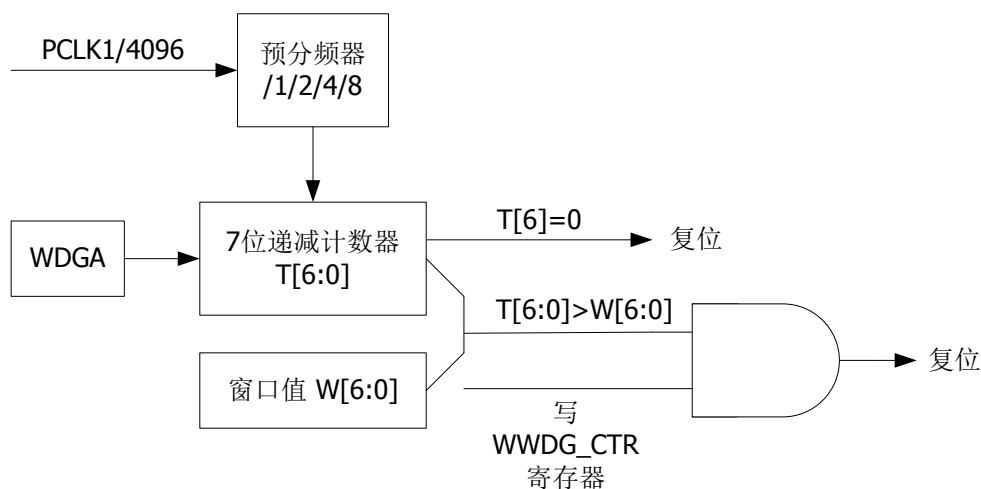
3.2 主要特点

- 自由运行的 7 位递减计数器
- 使能 WWDG 后，当计数值递减到 0x3F、或者当计数器未达到窗口门限之前就被刷新，即可有条件地复位
- 提供提前唤醒中断 (Early Wakeup Interrupt, 简称为 EWI)。假如已启动看门狗并且使能该中断，那么当计数值递减到 0x40 时会产生一个中断
- 提供 WWDG 调试模式控制位，根据此位的设置，WWDG 在调试模式下将继续运行或停止

3.3 功能描述

假如使能窗口看门狗 (将 WWDG_CTR 寄存器的 WDGA 置位)，那么当看门狗计数值递减到 0x3F (即第 6 位被清零) 后，就会产生复位。此外，如果计数器在尚未到达窗口门限之前就被刷新，同样也会产生复位。

图 96 窗口看门狗内部框图



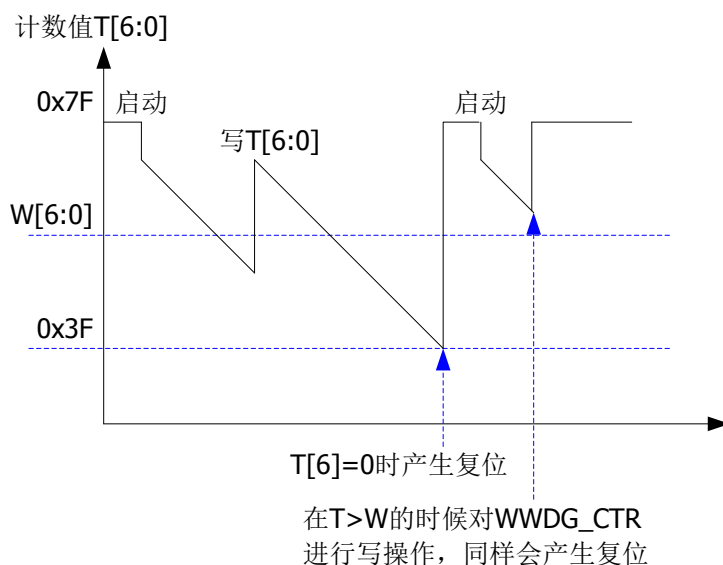
软件应当通过写 **WWDG_CTR** 寄存器来重载计数器，从而避免 **MCU** 复位。必须在计数值递减到 **0x3F** 之前、且计数值小于窗口门限值之后写 **WWDG_CTR** 寄存器。重载值必须在 **0xFF** 到 **0xC0** 范围内。

复位后窗口式看门狗是被禁用的。软件将 **WWDG_CTR** 寄存器的 **WDGA** 置位，即可启动窗口式看门狗。无论何时，只要使能了窗口式看门狗，其计数器就将不断递减，计数值应当保持大于 **0x3F**，也就是说应保持第 6 位始终为 1。**T[5:0]** 决定了两次重载之间的最大时间间隔。递减计数的速率取决于 **APB1** 时钟以及预分频系数 (**WWDG_CTR**)。配置寄存器 (**WWDG_CFR**) 中还包含窗口门限值 (**W[6:0]**)，递减计数器必须在低于窗口门限值且高于 **0x3F** 时重载，否则看门狗都将会产生复位。

将 **WWDG_CTR** 寄存器的 **EWI** 置位，即可使能提前唤醒中断 (**EWI**)。当计数值递减到 **0x40** 时，将会产生此中断。软件可以在该中断服务子程序中做些紧急处理工作，例如通信警告其它设备、或者紧急记录重要数据，以便于之后分析软件失效的原因、或者在真正复位之前紧急保存一些重要的数据。此外，软件也可以在这个中断服务子程序中重载计数器，以实现软件级的系统检查管理等等（当然，在这种情况下 **WWDG** 已经失去原本的意义，永不会产生 **WWDG** 复位）。

在 **WWDG_STR** 寄存器中对 **EWIF** 位写 0，即可清除 **EWI** 中断。

图 97 窗口式看门狗时序图



用户可以通过下面的公式计算 WWDG 超时值。

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PS}} \times (t[5:0] + 1) \text{ (ms)}$$

其中：

t_{WWDG} : WWDG 超时时间

t_{PCLK1} : APB1 时钟周期, 单位为 ms

关于 t_{WWDG} 的有效范围, 可参照下表。

 表 40 36MHz (f_{PCLK1}) 频率下的 WWDG 超时范围

预分频系数	PS[1:0]	最小超时 T[6:0]=0x40	最大超时 T[6:0]=0x7F
1/1	00	113 μs	7.28 ms
1/2	01	227 μs	14.56 ms
1/4	10	455 μs	29.12 ms
1/8	11	910 μs	58.25 ms

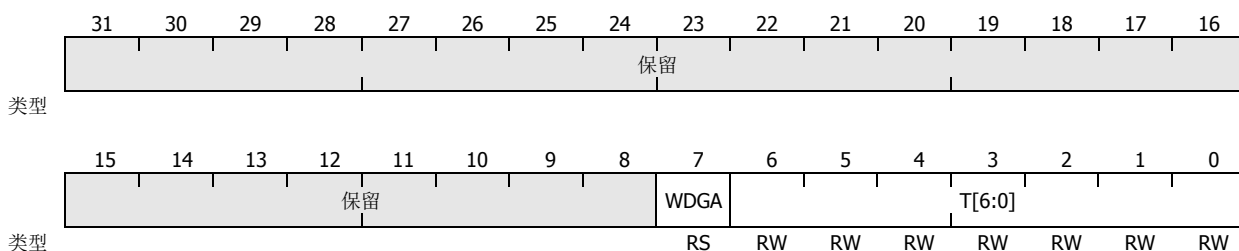
假如调试模块中的 DBG_WWDG_STOP 为 0, 那么即使 Cortex™-M3 内核停止工作 (调试模式), WWDG 也将继续工作。如果 DBG_WWDG_STOP 为 1, 那么 WWDG 在调试模式下将停止计数。

4 WWDG 寄存器

4.1 控制寄存器 (WWDG_CTRL)

地址偏移: 0x00

复位值 : 0x0000 007F

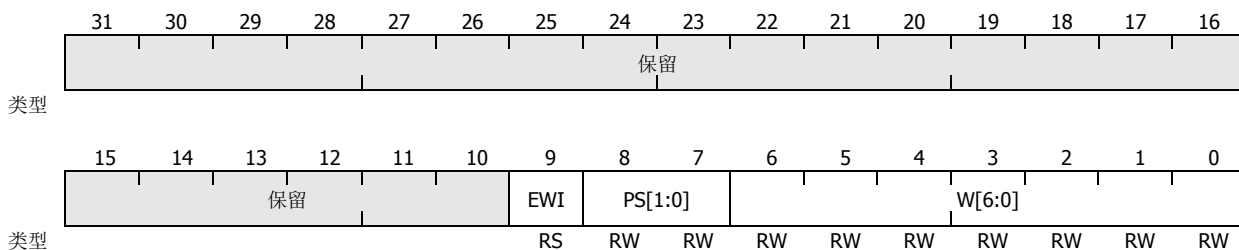


位/位域	名称	描述
31:8	保留	请保持其复位状态
7	WDGA	启动窗口式看门狗 硬件复位后清零。对本位写 0 无效果。 0: 禁用窗口式看门狗 1: 使能窗口式看门狗
6:0	T[6:0]	看门狗计数值 本位域从 0x40 递减到 0x3F 时将会产生复位。当计数值大于窗口门限值时，改写计数值同样会造成复位。

4.2 配置寄存器 (WWDG_CFR)

地址偏移: 0x04

复位值 : 0x0000 007F



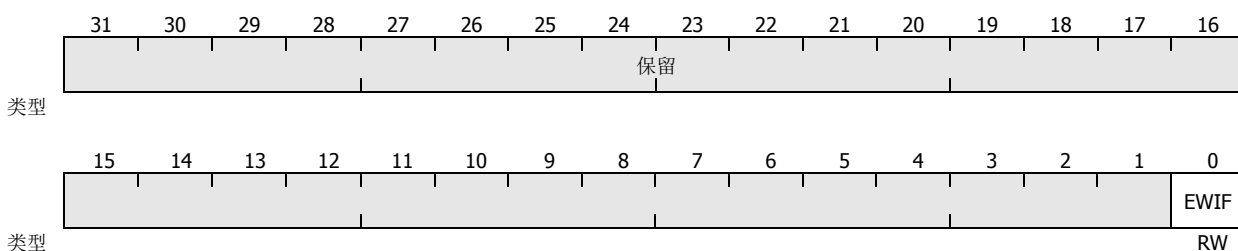
位/位域	名称	描述
31:10	保留	请保持其复位状态
9	EWI	提前唤醒中断 若本位置 1，则当计数值递减到 0x40 时会产生中断。硬件复位后本位清零。对本位写 0 无效果。
8:7	PS[1:0]	预分频系数 决定看门狗计数器的时基。 00: PCLK1/4096/1 01: PCLK1/4096/2 10: PCLK1/4096/4 11: PCLK1/4096/8

位/位域	名称	描述
6:0	W[6:0]	窗口门限值 当计数值大于窗口门限值时，改写计数值（WWDG_CTR 寄存器的 T 位域）会造成复位。

4.3 状态寄存器（WWDG_STR）

地址偏移：0x08

复位值：0x0000 0000



位/位域	名称	描述
31:1	保留	请保持其复位状态
0	EWIF	提前唤醒中断标志 当计数值递减到 0x40 时本标志将被硬件置位，即便提前唤醒中断并未使能（WWDG_CTR 寄存器的 EWIF 为 0）。写 0 即可将本位清零。写 1 无效果。

第10章 模数转换器 (ADC)

1 简介

GD32F103xx 系列微控制器集成了两路 12 位多通道的模数转换器 (Analog-Digital Converter, 简称为 ADC)。每路 ADC 总计有 18 个复用的输入通道, 其中 16 个外部通道用于连接外部模拟信号, 另有 2 个内部通道。模拟看门狗模块可以对各通道进行检测, 看哪些通道的采样值超出了有效范围。可配置的通道管理模块用于实现各种不同的转换模式: 单次转换、连续转换、扫描转换或间断转换。系统还提供了双 ADC 工作模式, 可让两个 ADC 模块协同工作实现多种高级功能。

2 主要特点

- 12 位分辨率, 最大支持 18 通道
- 多种工作模式: 单次转换、连续转换、扫描转换、间断转换
- 数据寄存器中的数据对齐方式可配置
- 常规数据传输可触发 DMA 请求
- 模拟看门狗
- 常规通道和插入通道可由外部信号触发
- ADC 输入范围: $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- 双 ADC 模式
- 提供多种中断事件: 模拟看门狗事件, 常规组转换结束事件, 插入组转换结束事件

3 功能描述

下图是 ADC 控制寄存器及数据寄存器的内部配置。此外用户还可根据后续表格查看 ADC 管脚配置。

图 98 ADC 模块内部框图

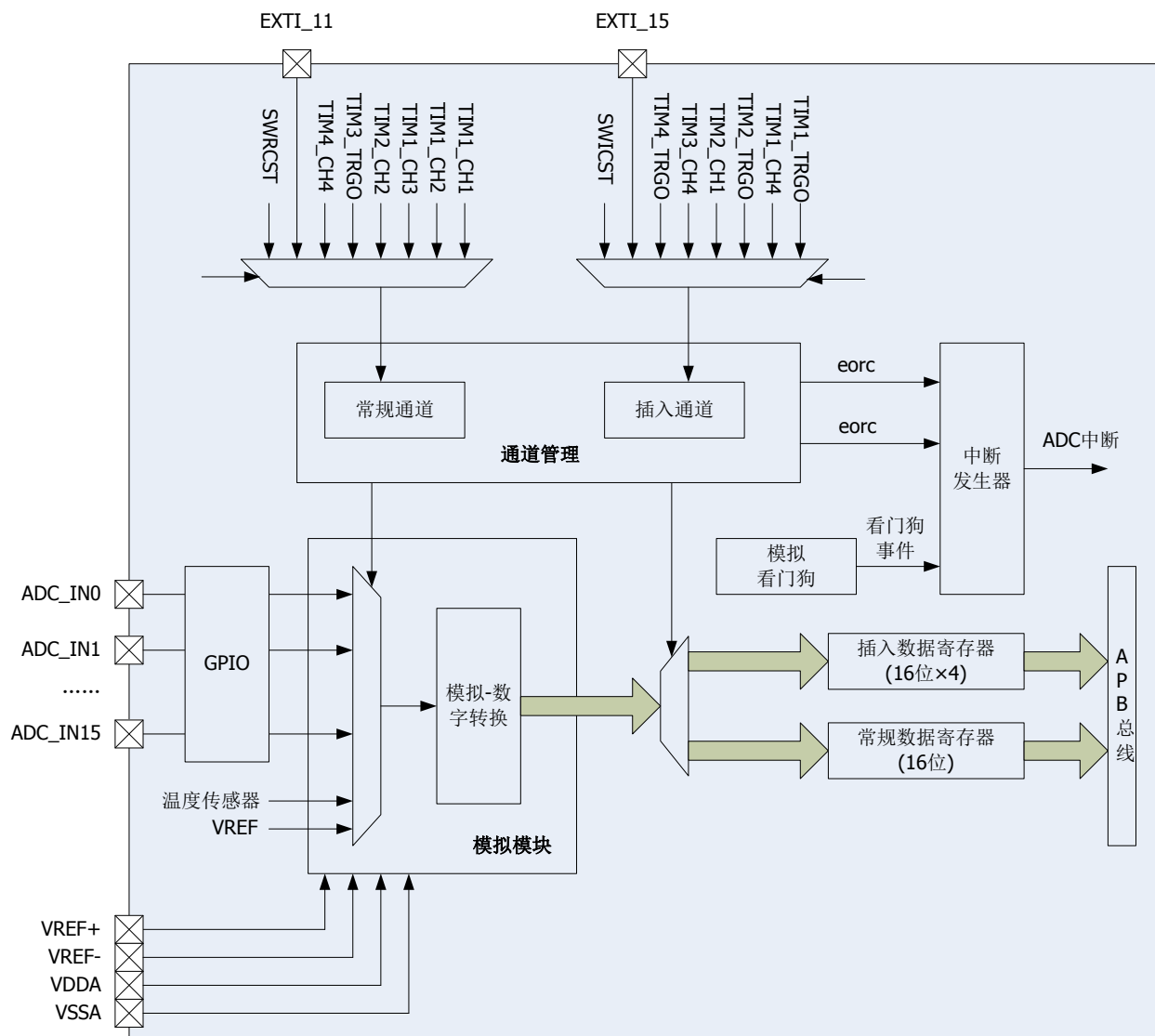


表 41 ADC 管脚定义

管脚名称	信号类型	注释
V _{REF+}	输入。模拟参考电源正极	ADC 参考电压的正极， $2.4V \leq V_{REF+} \leq V_{DDA}$
V _{DDA} ⁽¹⁾	输入。模拟电源	模拟电源输入，与 V _{DD} 等电位，且 $2.4V \leq V_{DDA} \leq 3.6V$
V _{REF-}	输入。模拟参考电源负极	ADC 参考电压的负极， $V_{REF-} = V_{SSA}$
V _{SSA} ⁽¹⁾	输入。模拟电源地	模拟电源的地，与 V _{SS} 等电位
ADCx_IN[15:0]	模拟信号输入	最大 21 个模拟输入通道

1. V_{DDA} 与 V_{SSA} 应分别接到 V_{DD} 与 V_{SS} 上。

3.1 常规以及插入通道组

ADC 支持 18 个复用通道，并且可以将转换结果分为两组：常规通道组和插入通道组。

在常规通道组中，可配置一个最多包含 16 个转换的特定序列，此时通过 ADC_RSQ1 ~ ADC_RSQ3 寄存器来指定常规通道组中每个转换所选定的通道。ADC_RSQ1 寄存器的 RL[3:0]位域可指定整个转换序列的长度。

在插入通道组中，可配置一个最多包含 4 个转换的特定序列，此时通过 ADC_ISQ 寄存器来指定插入通道组中每个转换所选定的通道。ADC_ISQ 寄存器的 IL[3:0]位域可指定整个转换序列的长度。

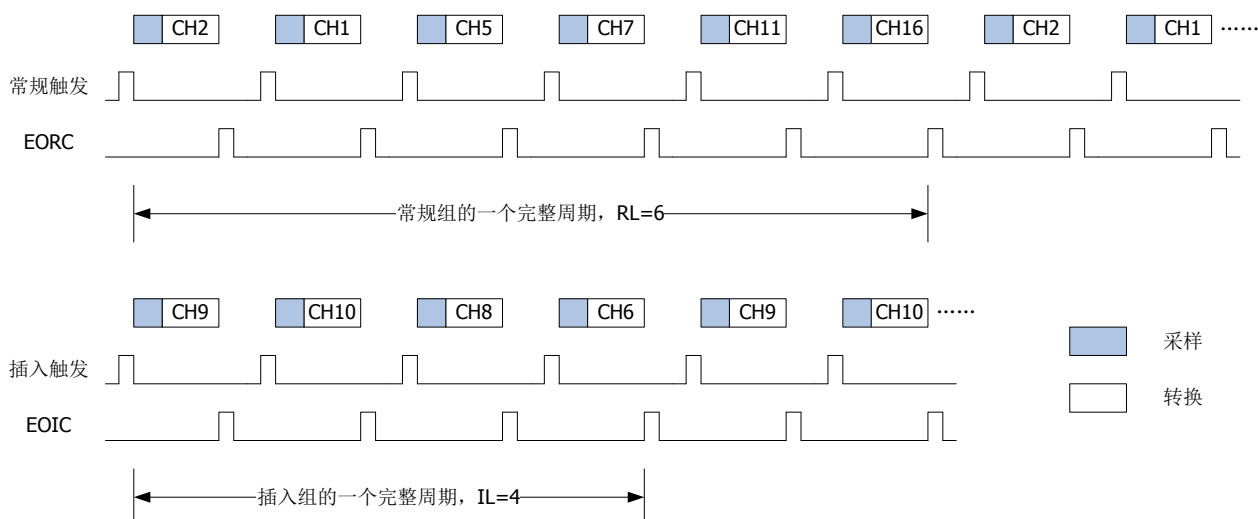
ADC 时钟是从 PCLK2 分频获得的，并与其保持同步。关于 ADC 时钟的更多信息，请参阅 RCC 模块的相关说明。

3.2 转换模式

单次转换模式

在单次转换模式下，ADC 模块将按照 ADC_RSQ1 ~ ADC_RSQ3 寄存器或 ADC_ISQ 寄存器所指定的序列执行转换。当 ADCON 被置 1 以后，每当相应的软件触发信号或外部触发信号生效时，ADC 就会对常规组或插入组中的一个通道进行采样并转换。该通道转换完毕后，转换数据将存入 ADC_RCD 或 ADC_ICD 寄存器中，并且 EORC 或 EOIC 标志将会置位。若 EORCIE 或 EOICIE 为 1，那么还会产生中断。

图 99 单次转换模式



连续转换模式

若 ADC_CTRL2 寄存器中的 CTN 位为 1，则将使能连续转换模式。在这种工作模式下，ADC 模块

将按照 ADC_RSQ1 ~ ADC_RSQ3 寄存器或 ADC_ISQ 寄存器所指定的序列执行转换。当 ADCON 被置 1 以后，当相应的软件触发信号或外部触发信号生效时，ADC 就会对常规组或插入组中的所有通道逐个进行采样并转换。各通道转换完毕后，转换数据将存入 ADC_RCD 或 ADC_ICD 寄存器中。如果禁用扫描模式，那么 EORC 或 EOIC 标志将会在每次转换完一个通道后置位；如果使能扫描模式，那么 EORC 或 EOIC 标志将会在转换完全部通道后置位。若 EORCIE 或 EOICIE 为 1，那么还会产生中断。

图 100 连续转换模式，禁用扫描

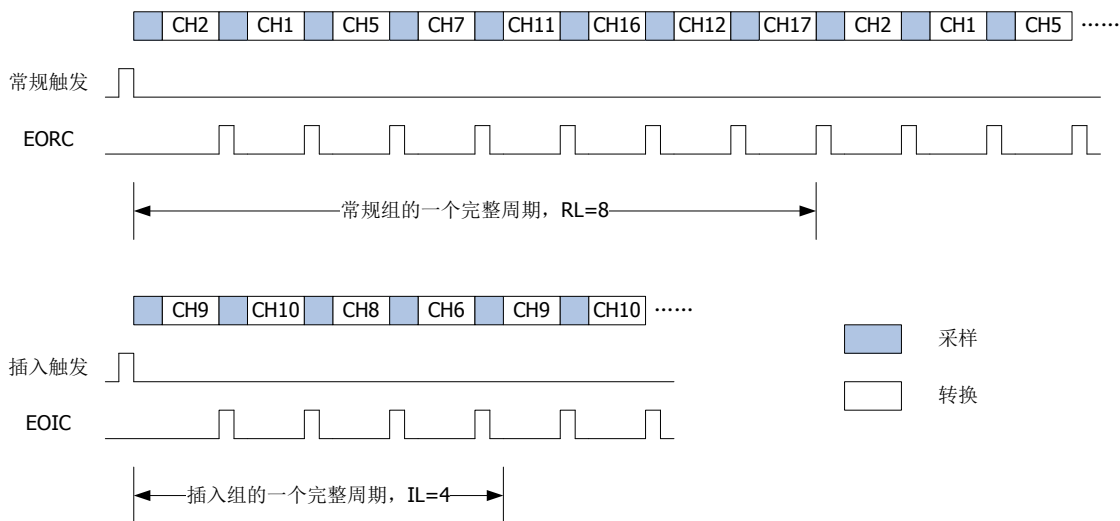
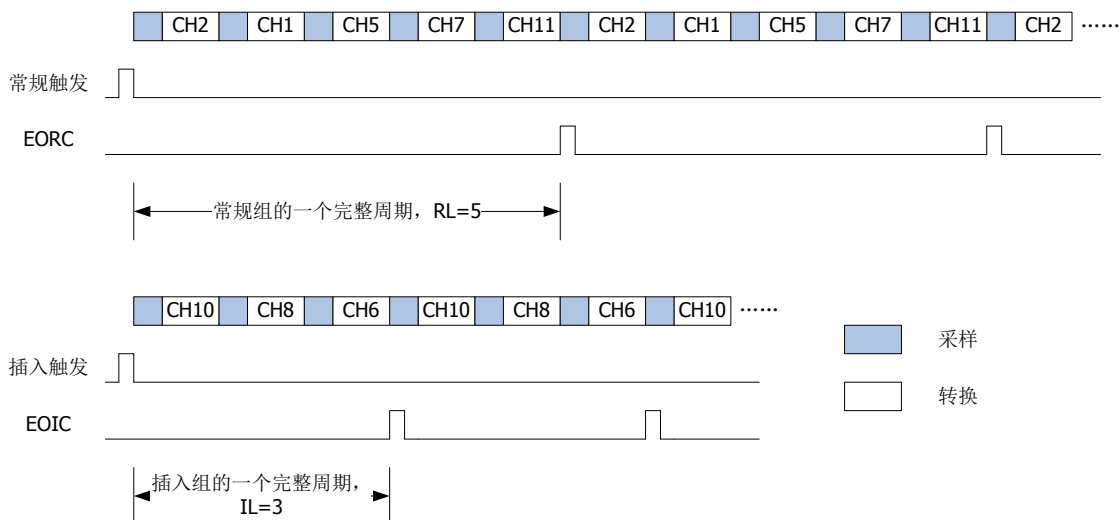


图 101 连续转换模式，使能扫描

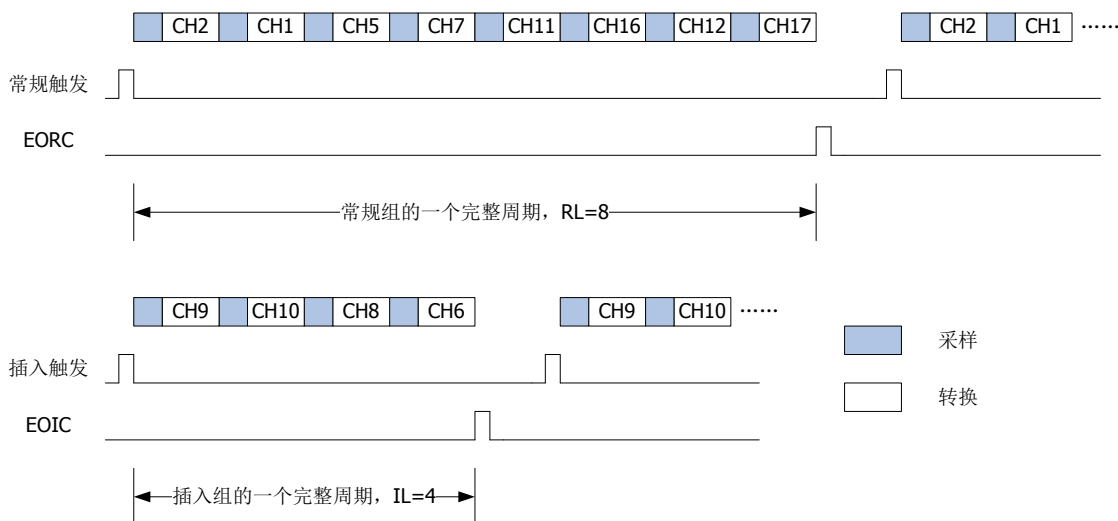


扫描转换模式

若 ADC_CTRL1 寄存器的 SM 位为 1，则将使能扫描转换模式。在这种工作模式下，ADC 模块将

按照 ADC_RSQ1 ~ ADC_RSQ3 寄存器或 ADC_ISQ 寄存器所指定的序列执行转换。当 ADCON 被置 1 以后，当相应的软件触发信号或外部触发信号生效时，ADC 就会对常规组或插入组中的所有通道逐个进行采样并转换，直至常规组或插入组的结尾。各通道转换完毕后，转换数据将存入 ADC_RCD 或 ADC_ICD 寄存器中。EORC 或 EOIC 标志将会在转换完全部通道后置位。若 EORCIE 或 EOICIE 为 1，那么还会产生中断。在扫描模式下，ADC_CTRL2 寄存器的 DMA 必须置位。

图 102 扫描转换模式



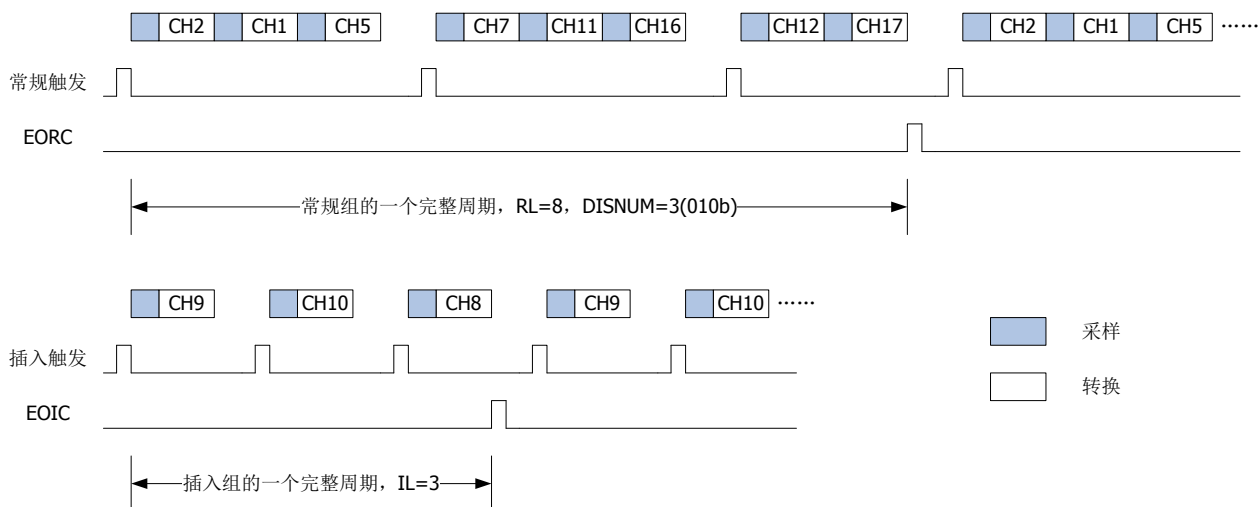
间断转换模式

对于常规通道组，若 ADC_CTRL1 寄存器的 DISRC 为 1，即可使能间断转换模式。在这种工作模式下，ADC 每次执行一个仅包含 n 次转换 ($n \leq 8$ ，由 ADC_CTRL1 寄存器的 DISNUM 位域定义) 的较短的序列，该序列是 ADC_RSQ1 ~ ADC_RSQ3 寄存器所定义的完整序列的一部分。当相应的软件触发信号或外部触发信号生效时，ADC 将按照 ADC_RSQ1 ~ ADC_RSQ3 寄存器所定义的序列转换接下来的 n 个通道，且到常规通道组的完整序列转换完毕为止。在常规通道组的整个序列执行完毕之后，EORC 标志才会置位。若 EORCIE 为 1，那么还会产生中断。

对于插入通道组，若 ADC_CTRL1 寄存器的 DISIC 为 1，即可使能间断转换模式。在这种工作模式下，ADC 每次执行一个转换，该转换是 ADC_ISQ 寄存器所定义的完整序列的其中之一。当相应的软件触发信号或外部触发信号生效时，ADC 将按照 ADC_ISQ 寄存器所定义的序列转换下一个通道，且到插入通道组的完整序列转换完毕为止。在插入通道组的整个序列执行完毕之后，EOIC 标志才会置位。若 EOICIE 为 1，那么还会产生中断。

在间断转换模式下，常规通道组和插入通道组不能同时工作，只允许将二者之一设置为间断转换模式。

图 103 间断转换模式



3.3 模拟看门狗

将 `ADC_CTRL1` 寄存器的 `AWREN` 或 `AWIEN` 置位, 即可分别为常规通道组或插入通道组开启模拟看门狗。当 ADC 转换的模拟电压低于下限或高于上限时, `ADC_STATUS` 寄存器中的 `AWE` 将会置位。若 `AWEIE` 为 1, 那么还会产生一个中断。采样有效范围的上下限分别通过 `ADC_HTR` 以及 `ADC_LTR` 寄存器来定义。由于门限比较是在数据对齐之前完成的, 因此门限值与对齐操作 (由 `ADC_CTRL2` 寄存器中的 `DAL` 位设置) 无关。通过 `ADC_CTRL1` 寄存器中的 `AWREN`、`AWIEN`、`AWSSM` 以及 `AWCS[4:0]` 位域, 可选择一个或多个通道被模拟看门狗监控。

3.4 插入通道管理

自动插入

若 `ADC_CTRL1` 寄存器中的 `ICA` 位为 1, 则插入组的通道将在常规组的通道转换之后自动进行转换。在这种模式下, 无法使能插入通道的外部触发。在这种模式下, ADC 将按照 `ADC_RSQ1 ~ ADC_RSQ3` 寄存器以及 `ADC_ISQ` 寄存器所定义的序列 (合计最大 20 个) 进行转换。若 `ICA` 位为 1 且 `CNT` 位也为 1, 那么在插入通道之前的常规通道将以连续方式转换。

当 ADC 设置为间断转换模式时, 不可使能自动插入模式。

触发插入

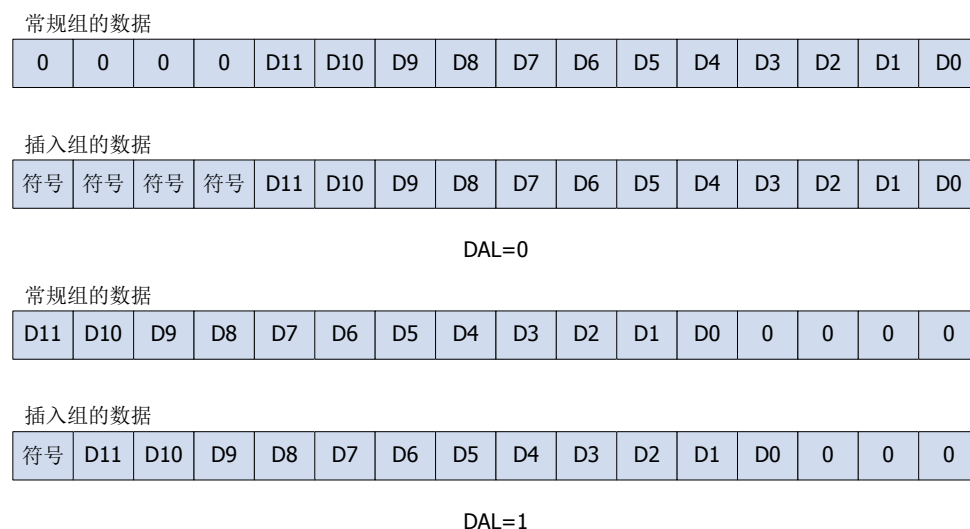
若 `ICA` 位为 0、`SM` 位为 1, 那么在常规组通道转换期间, 若检测到软件插入触发信号或外部插入触发信号, 则将触发插入转换。在这种情况下, ADC 会中止当前的转换序列 (常规组序列), 并开始按照插入通道序列以扫描模式执行转换。当插入通道组的完整序列执行完毕之后, ADC 将从上一次被打断的常规通道开始, 恢复常规通道组的序列转换过程。

3.5 数据对齐

转换完成后存储数据的对齐方式可通过 ADC_CTRL2 寄存器中的 DAL 位进行设置。

插入组数据在减去 ADC_ICOSn 寄存器中定义的偏移量之后，有可能得到负数。此时将会扩展符号位。

图 104 数据对齐



3.6 可编程的采样时间

通过对 ADC_SPT1、ADC_SPT2 寄存器的 SPTn 位域进行设置，可设定对输入电压采样所消耗的 ADC_CLK 周期数，并且可以为每个通道分别设置不同的采样时间。总的转换时间为“采样周期数 + 12.5”个 ADC_CLK 周期。

3.7 外部触发

当 ADC_CTRL2 寄存器的 ETERC 或 ETEIC 置位后，常规组或插入组的转换便可通过外部触发输入信号的上升沿触发。ETSRC 及 ETSIC 控制位域可定义 8 种可能的触发事件，据此触发常规组或插入组的转换。

表 42 常规通道的外部触发选择

ETSRC[2:0]	触发源	触发类型
000	TIM1_CH1	片内信号
001	TIM1_CH2	
010	TIM1_CH3	
011	TIM2_CH2	
100	TIM3_TRGO	
101	TIM4_CH4	

ETSRC[2:0]	触发源	触发类型
110	EXTI_11	外部信号
111	SWRCST	软件触发

表 43 插入通道的外部触发选择

ETSRC[2:0]	触发源	触发类型
000	TIM1_TRGO	片内信号
001	TIM1_CH4	
010	TIM2_TRGO	
011	TIM2_CH1	
100	TIM3_CH4	
101	TIM4_TRGO	
110	EXTI_15	外部信号
111	SWICST	软件触发

3.8 DMA 请求

当常规组要对多个通道进行转换时，可通过 DMA 请求来传输转换数据。ADC 在转换完一个常规通道之后，便会产生 DMA 请求。DMA 控制器收到该请求后，即可将转换数据从 ADC_RCD 寄存器传送到用户指定的目标地址。

请注意：仅 ADC1 模块支持 DMA 传输。只有在双 ADC 模式下，ADC2 模块所转换的数据才能有可能会通过 DMA 传输。

3.9 双 ADC 模式

在双 ADC 模式下，ADC1 与 ADC2 将以主-从关系协同工作。取决于 ADC1_CTRL1 寄存器中 DAM 位域的设置，ADC1 将作为主控驱动 ADC1、ADC2 交替进行转换或同时进行转换。

在双 ADC 模式下，若用户配置成由外部事件触发转换时，触发信号只能分配给主控设备，这是为了避免从属设备执行了预料之外的转换。不过，不管主控 ADC 还是从属 ADC 都必须使能外部触发功能。

双 ADC 模式可分为 6 种子模式：

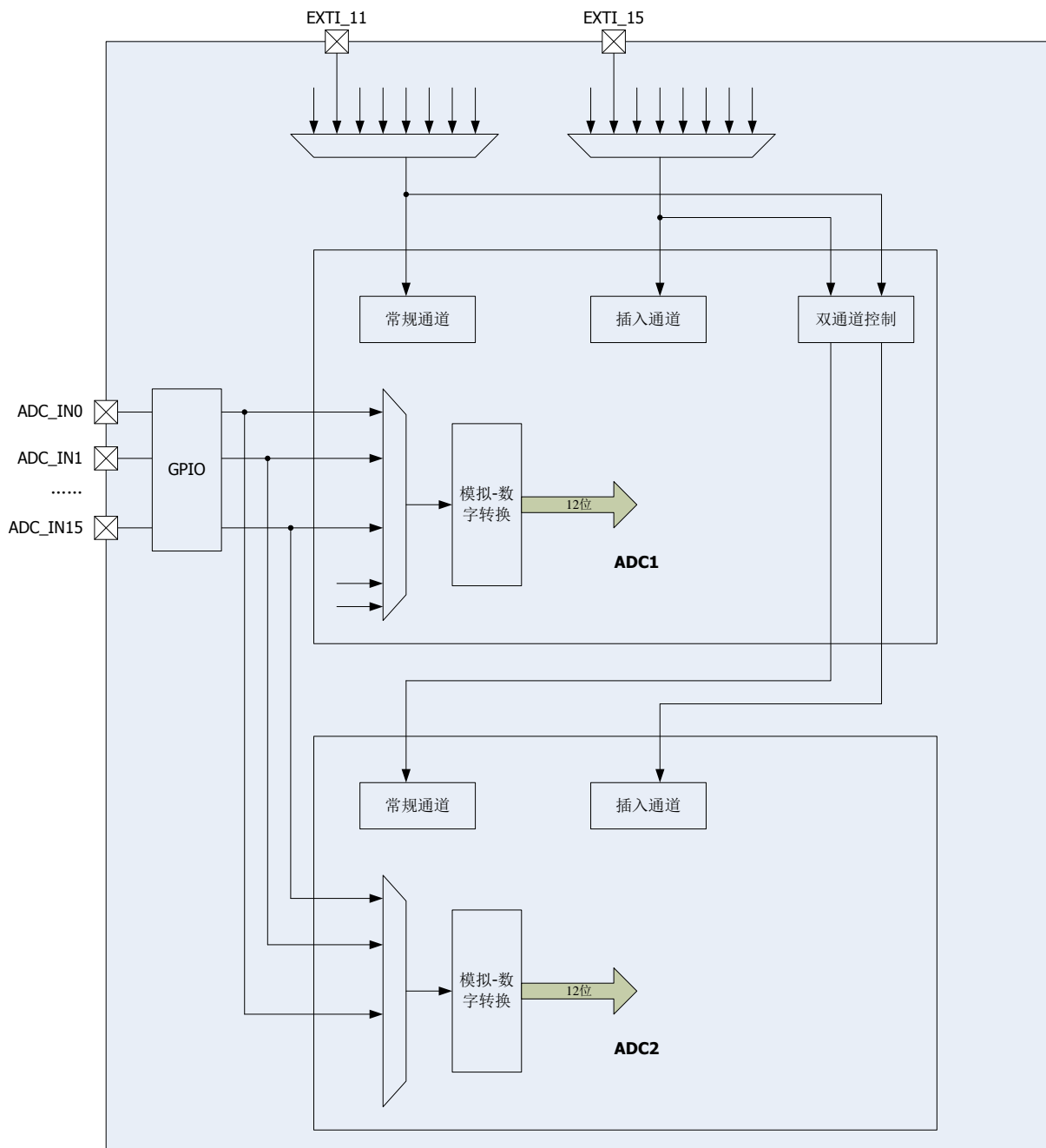
- 各行其是模式
- 常规组并发模式
- 插入组并发模式
- 交替触发模式
- 高速延迟模式
- 低速延迟模式

此外，还有一些将两个双通道子模式组合起来的工作方式：

- 常规组并发模式 + 插入组并发模式
- 常规组并发模式 + 交替触发模式
- 插入组并发模式 + 交替触发模式

在双 ADC 模式下，即使并不需要通过 DMA 来传输常规通道的 ADC 转换结果，也必须使能 DMA，这样才能在主控 ADC 的数据寄存器中读取从属 ADC 模块的转换数据。

图 105 双 ADC 模式下的功能框图



用户应当注意避免两个 ADC 模块对同一个通道的采样时间发生重叠。

各行其是模式

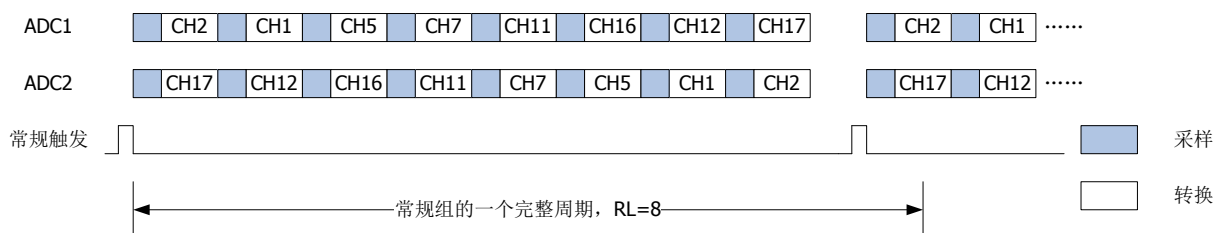
在这种模式下，ADC1 和 ADC2 各自独立工作，双 ADC 控制逻辑实际上是被禁用的。

常规组并发模式

在这种模式下，ADC1 和 ADC2 的常规通道并行工作。当 ADC1 被外部触发后，将同时触发 ADC2。当一个通道转换完成后，DMA 将转换结果数据从 ADC1 模块的 ADC_RCD 寄存器传送到 SRAM 中，该寄存器的低半字是 ADC1 转换结果、高半字是 ADC2 转换结果。如果使能了 EORC 中断，则会产生一个中断。

若两个 ADC 模块的转换序列长度不一，那么外部触发信号必须保持足够的间隔，确保两个序列都能转换完成，防止出现序列较长的 ADC 尚未转换完成时、序列较短的 ADC 又被触发重新开始执行序列的情况。

图 106 常规组并发模式

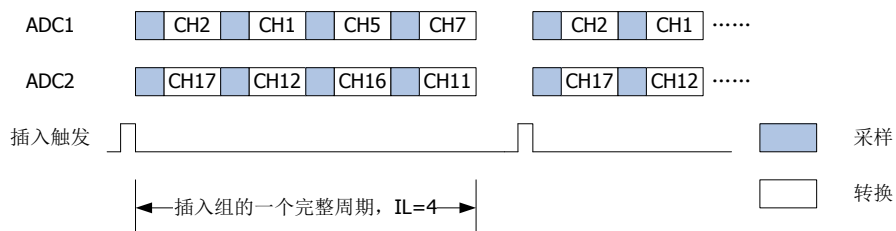


插入组并发模式

在这种模式下，ADC1 和 ADC2 的插入通道并行工作。当 ADC1 被外部触发后，将同时触发 ADC2。当一个通道转换完成后，转换结果数据将保存在 ADC_ICDn 寄存器中。如果使能了 EORC 中断，则会产生一个中断。

若两个 ADC 模块的转换序列长度不一，那么外部触发信号必须保持足够的间隔，确保两个序列都能转换完成，防止出现序列较长的 ADC 尚未转换完成时、序列较短的 ADC 又被触发重新开始执行序列的情况。

图 107 插入组并发模式



交替触发模式

这种模式只能配合插入通道组使用。ADC1 插入组受外部触发后，将交替驱动 ADC1 与 ADC2 进行转换，并且每当产生外部触发信号时，该 ADC 模块的所有插入通道将全部被转换。当插入组的所有通道全部转换完成后，若使能 EOIC 中断，则将产生一个中断。

若 ADC1 和 ADC2 都开启了插入通道的间断转换模式，那么每当产生外部触发信号时，将会转换一个插入通道。当插入组的所有通道全部转换完成后，若使能 EOIC 中断，则将产生一个中断。

图 108 交替触发模式，禁用插入间断模式

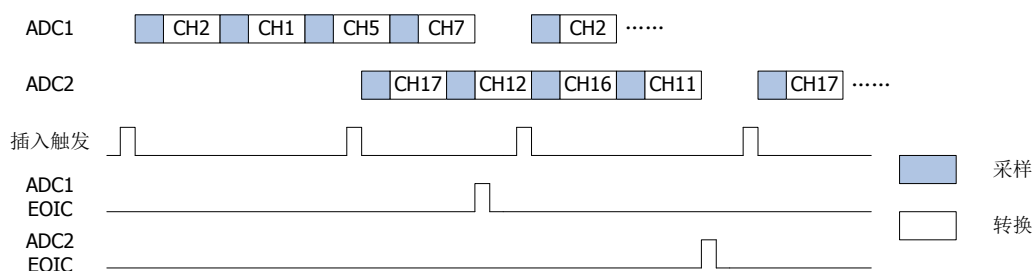
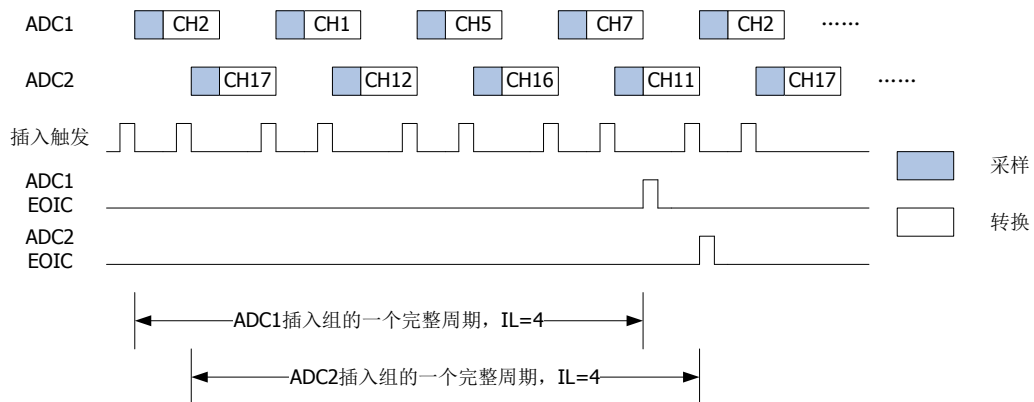


图 109 交替触发模式，使能插入间断模式



高速延迟模式

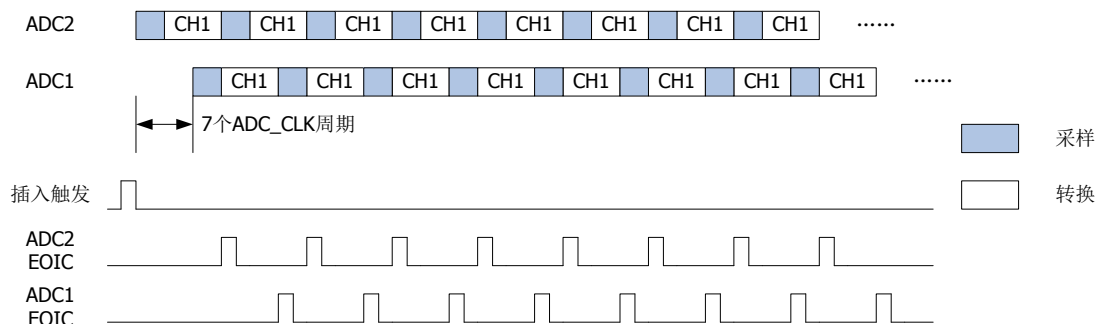
这种模式只能配合常规通道组使用，且通常只包含一个通道。ADC1 常规通道组的外部触发信号同时驱动 ADC1 和 ADC2。当受到外部触发后，ADC2 将立即开始转换，ADC1 则在延迟 7 个 ADC 时钟周期后开始转换。在这种模式下，采样时间必须配置为小于 7 个 ADCCLK 周期，以避免 ADC1 和 ADC2 都对同一个通道进行转换时、二者的采样相位出现重叠。

假如 ADC1 和 ADC2 都开启了连续转换模式，那么两个 ADC 的常规通道都将连续转换。

当一个通道转换完成后，DMA 会将转换结果数据从 ADC1 的 ADC_RCD 寄存器传送给 SRAM，其

中低半字为 ADC1 的转换结果，高半字为 ADC2 的转换结果。

图 110 高速延迟模式，使能连续转换模式



低速延迟模式

这种模式只能配合常规通道组使用，且通常只包含一个通道。ADC1 常规通道组的外部触发信号同时驱动 ADC1 和 ADC2。当受到外部触发后，DC2 将立即开始转换，ADC1 则在延迟 14 个 ADC 时钟周期后开始转换，之后再过 14 个 ADC 时钟周期后 ADC2 再次开始转换，依此类推。

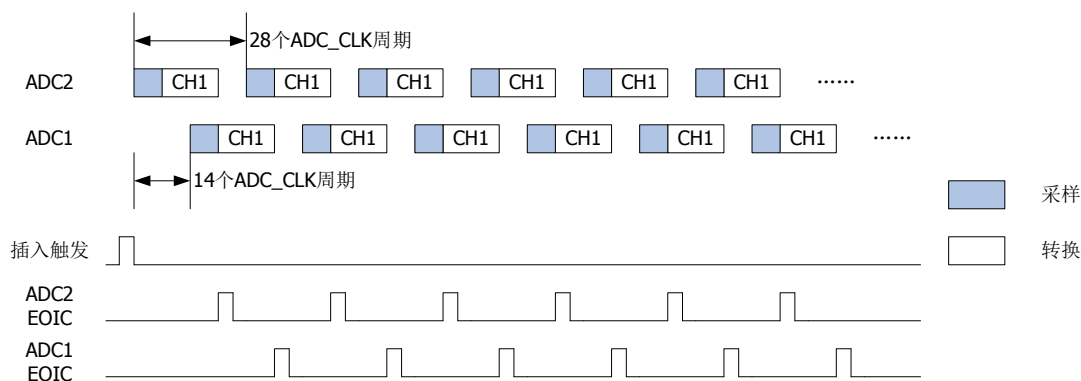
在这种模式下，采样时间必须配置为小于 14 个 ADCCLK 周期，以避免 ADC1 和 ADC2 都对同一个通道进行转换时、二者的采样相位出现重叠。

假如 ADC1 和 ADC2 都开启了连续转换模式，那么两个 ADC 的常规通道都将连续转换。

当一个通道转换完成后，DMA 会将转换结果数据从 ADC1 的 ADC_RCD 寄存器传送给 SRAM，其中低半字为 ADC1 的转换结果，高半字为 ADC2 的转换结果。

由于选定的常规通道是连续转换的，因此在这种模式下不能开启连续转换模式，并且绝对不能有插入通道的外部触发信号。

图 111 低速延迟模式



常规组并发+插入组并发模式

在这种工作模式下，ADC 常规组的并发转换可以被 ADC 插入组的并发转换所中断。

若两个 ADC 模块的转换序列长度不一，那么外部触发信号的间隔必须大于两个序列中的任何一个，防止出现序列较长的 ADC 尚未转换完成时、序列较短的 ADC 又被触发重新开始执行序列的情况。

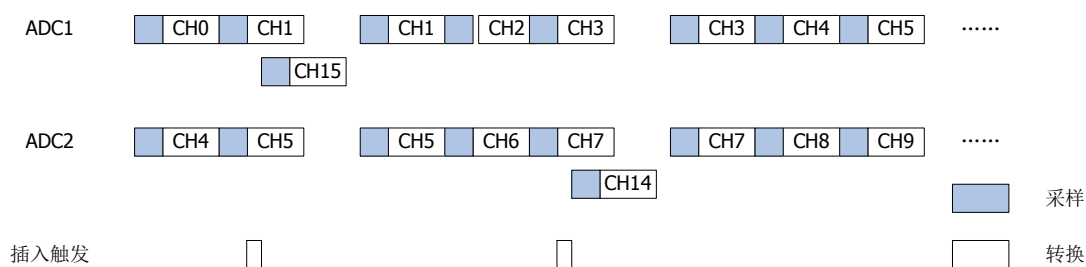
常规组并发+交替触发模式

在这种工作模式下，ADC 常规组的并发转换可以被 ADC 插入组的交替触发转换所中断。

当产生插入事件后将立即开始执行插入转换。此时两个 ADC 的常规组转换都将停止，并且在插入转换结束后同步恢复，以确保插入转换之后的同步。

若两个 ADC 模块的转换序列长度不一，那么外部触发信号的间隔必须大于两个序列中的任何一个，防止出现序列较长的 ADC 尚未转换完成时、序列较短的 ADC 又被触发重新开始执行序列的情况。

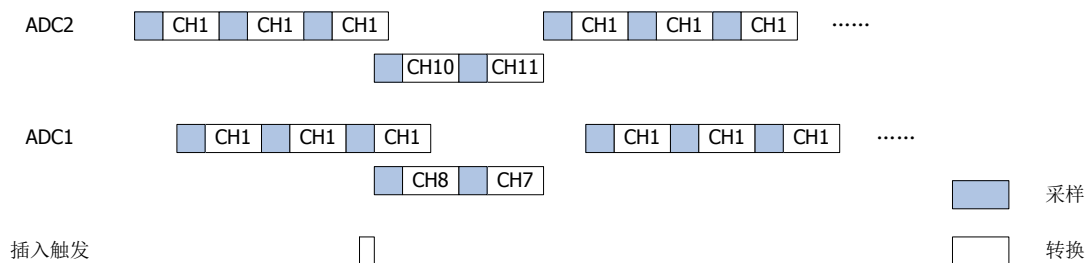
图 112 常规组并发+交替触发模式



插入组并发+延迟模式

在这种工作模式下，双 ADC 的延迟转换可以被 ADC 插入组的并发转换所中断。当插入转换序列结束后，将恢复继续进行双 ADC 延迟转换。

图 113 插入组并发+延迟模式



3.10 温度传感器及 V_{REF}

若 ADC_CTRL2 寄存器中的 TSVRE 置位，即可使能温度传感器通道（ADC1_CH16）以及 V_{REF} 通道（ADC1_CH17）。温度传感器可用于测量器件周边的环境温度，其输出电压由 ADC1 转换为数字值。温度传感器的采样时间建议设置为 17.1 μs 。当应用中不使用温度传感器时，可将 TSVRE 位清零，使得温度传感器部分不上电。

温度传感器的输出电压的变化与温度成线性关系。由于存在静态偏移（最高可达 45°C），并且器件之间由于工艺的微小变化而存在差异，从这个角度来说，片内温度传感器更适用于检测温度变化而不是测量绝对温度。如果需要实现温度值的精确测量，应当采用独立的片外温度传感器。

3.11 ADC 中断

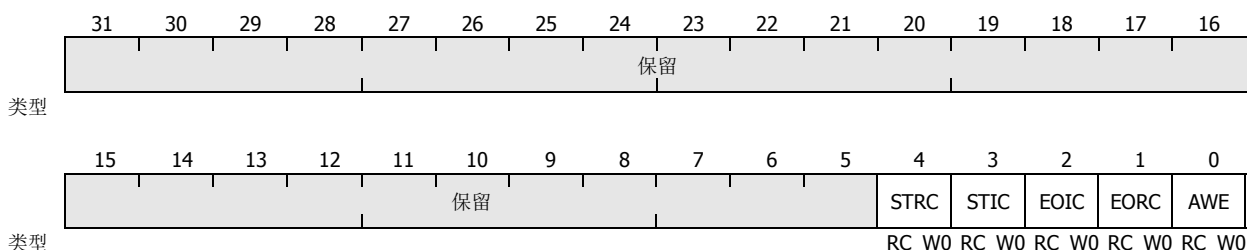
常规组和插入组的转换过程结束后可以产生中断。模拟看门狗的状态位置位时也可以产生中断。系统提供了多个相互无关的中断使能位，以确保用户使用的灵活性。请注意，ADC1 和 ADC2 的中断是映射在同一个中断向量的。

4 ADC 寄存器

4.1 ADC 状态寄存器（ADC_SR）

地址偏移：0x00

复位值：0x0000 0000



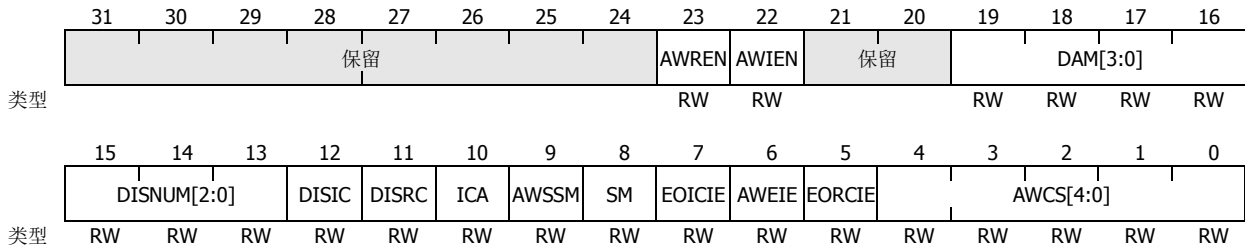
位/位域	名称	描述
31:5	保留	请保持其复位状态
4	STRC	常规通道组的启动标志 0: 常规通道组未启动 1: 常规通道组已启动
3	STIC	插入通道组的启动标志 0: 插入通道组未启动 1: 插入通道组已启动
2	EOIC	插入通道组的转换结束标志 0: 插入通道组未进入转换结束状态 1: 插入通道组的转换结束

位/位域	名称	描述
1	EORC	常规通道组的转换结束标志 0: 常规通道组未进入转换结束状态 1: 常规通道组的转换结束
0	AWE	模拟看门狗事件标志 0: 无模拟看门狗事件 1: 有模拟看门狗事件

4.2 ADC 控制寄存器 1 (ADC_CTRL1)

地址偏移: 0x04

复位值 : 0x0000 0000



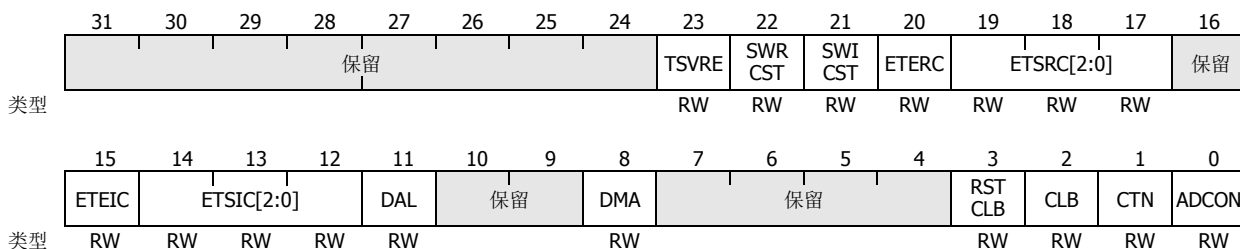
位/位域	名称	描述
31:24	保留	请保持其复位状态
23	AWREN	在常规通道是否使能模拟看门狗 0: 在常规通道禁用模拟看门狗 1: 在常规通道使能模拟看门狗
22	AWIEN	在插入通道是否使能模拟看门狗 0: 在插入通道禁用模拟看门狗 1: 在插入通道使能模拟看门狗
21:20	保留	请保持其复位状态
19:16	DAM[3:0]	双 ADC 模式配置 0000: 独立模式 0001: 常规组并发+插入组并发模式 0010: 常规组并发+交替触发模式 0011: 插入组并发+高速延迟模式 0100: 插入组并发+低速延迟模式 0101: 常规组并发模式 0110: 插入组并发模式 0111: 高速延迟模式 1000: 低速延迟模式 1001: 交替触发模式 本位域在 ADC2 模块中保留。
15:13	DISNUM[2:0]	间断模式下的转换数目 触发之后, 被转换的通道数目为 DISNUM+1。

位/位域	名称	描述
12	DISIC	插入通道的中断模式 0: 在插入通道禁用中断模式 1: 在插入通道使能中断模式
11	DISRC	常规通道的中断模式 0: 在常规通道禁用中断模式 1: 在常规通道使能中断模式
10	ICA	插入通道组自动转换 0: 禁用插入通道组的自动转换 1: 使能插入通道组的自动转换
9	AWSSM	在扫描模式下，模拟看门狗是否对单个通道有效 0: 模拟看门狗对所有通道有效 1: 模拟看门狗对单个通道有效
8	SM	扫描模式 0: 使能扫描模式 1: 禁用扫描模式
7	EOICIE	EOIC 的中断使能 0: 禁用 EOIC 中断 1: 使能 EOIC 中断
6	AWEIE	AWE 的中断使能 0: 禁用 AWE 中断 1: 使能 AWE 中断
5	EORCIE	EORC 的中断使能 0: 禁用 EORC 中断 1: 使能 EORC 中断
4:0	AWCS	模拟看门狗通道选择 00000: ADC 通道 0 00001: ADC 通道 1 00010: ADC 通道 2 01111: ADC 通道 15 10000: ADC 通道 16 10001: ADC 通道 17 其它取值保留。

4.3 ADC 控制寄存器 2 (ADC_CTRL2)

地址偏移: 0x08

复位值 : 0x0000 0000



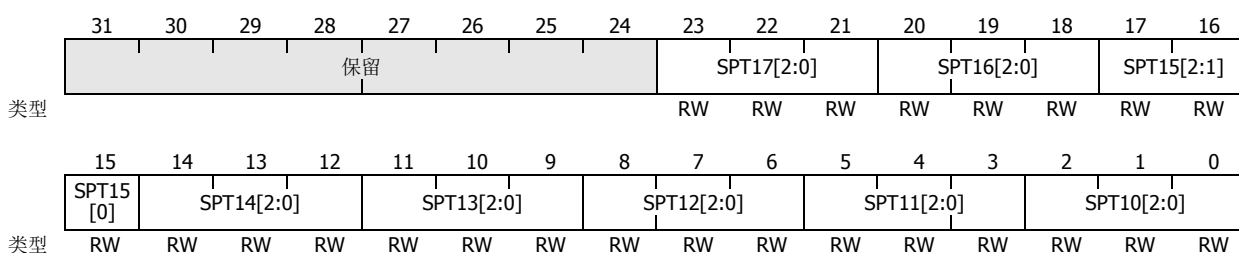
位/位域	名称	描述
31:24	保留	请保持其复位状态
23	TSVRE	ADC1 的通道 16、通道 17 使能 0: 禁用 ADC1 的通道 16、通道 17 1: 使能 ADC1 的通道 16、通道 17 本位对于 ADC2 保留。
22	SWRCST	启动常规通道 若 ETSRC 为 111, 则对本位写 1 即可启动转换一组常规通道。本位可由软件置位/清零, 或在转换开始后由硬件清零。
21	SWICST	启动插入通道 若 ETSRC 为 111, 则对本位写 1 即可启动转换一组插入通道。本位可由软件置位/清零, 或在转换开始后由硬件清零。
20	ETERC	常规通道的外部触发使能 0: 禁用常规通道的外部触发 1: 使能常规通道的外部触发
19:17	ETSRC[2:0]	常规通道的外部触发选择 000: 定时器 1 的 CC1 001: 定时器 1 的 CC2 010: 定时器 1 的 CC3 011: 定时器 2 的 CC2 100: 定时器 3 的 TRGO 101: 定时器 4 的 CC4 110: EXTI 口线 11 111: SWRCST
16	保留	请保持其复位状态
15	ETEIC	插入通道的外部触发使能 0: 禁用插入通道的外部触发 1: 使能插入通道的外部触发
14:12	ETSIC[2:0]	插入通道的外部触发选择 000: 定时器 1 的 TRGO 001: 定时器 1 的 CC4 010: 定时器 2 的 TRGO 011: 定时器 2 的 CC1 100: 定时器 3 的 CC4 101: 定时器 4 的 TRGO 110: EXTI 口线 15 111: SWICST
11	DAL	数据对齐方式 0: 按最低位对齐 1: 按最高位对齐
10:9	保留	请保持其复位状态
8	DMA	DMA 请求使能 0: 禁用 DMA 请求 1: 使能 DMA 请求 本位在 ADC2 中保留
7:4	保留	请保持其复位状态

位/位域	名称	描述
3	RSTCLB	复位校准 本位由软件置位，由硬件在校准寄存器被初始化后清零。 0: 校准寄存器初始化已完成 1: 校准寄存器开始初始化
2	CLB	ADC 校准 0: 校准已完成 1: 校准已启动
1	CTN	连续模式 0: 禁用连续模式 1: 使能连续模式
0	ADCON	ADC 开启 当本位从 0 改写为 1 时，将唤醒 ADC 模块。当本位已经是 1、又写入 1，并且寄存器的其它位没有改变时，将会启动转换。 0: ADC 模块被禁用，并且掉电 1: 使能 ADC

4.4 ADC 采样时间寄存器 1 (ADC_SPT1)

地址偏移: 0x0C

复位值 : 0x0000 0000



位/位域	名称	描述
31:24	保留	请保持其复位状态
23:0	SPTn[2:0]	通道 n 采样时间 000: 1.5 个周期 001: 7.5 个周期 010: 13.5 个周期 011: 28.5 个周期 100: 41.5 个周期 101: 55.5 个周期 110: 71.5 个周期 111: 239.5 个周期

4.5 ADC 采样时间寄存器 2 (ADC_SPT2)

地址偏移: 0x10

复位值 : 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
类型			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPT5[0]	SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]		
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:30	保留	请保持其复位状态
29:0	SPTn[2:0]	通道 n 采样时间 000: 1.5 个周期 001: 7.5 个周期 010: 13.5 个周期 011: 28.5 个周期 100: 41.5 个周期 101: 55.5 个周期 110: 71.5 个周期 111: 239.5 个周期

4.6 ADC 插入通道数据偏移寄存器 x (ADC_ICOSx) (x=1~4)

地址偏移: 0x14~0x20

复位值 : 0x0000 0000

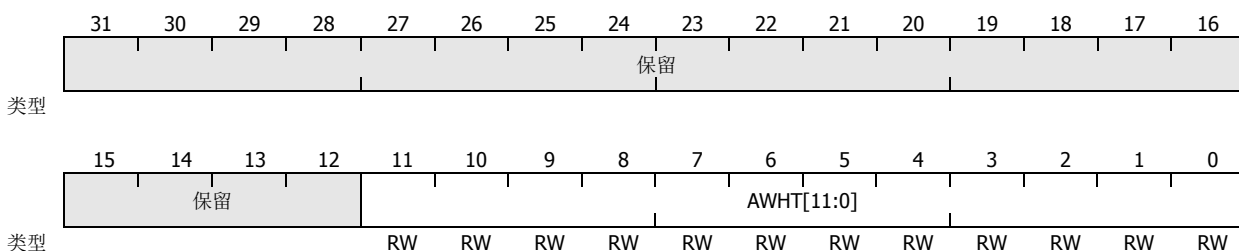
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				ICOSn[11:0]											
类型					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:12	保留	请保持其复位状态
11:0	ICOSn[11:0]	插入通道 n 的数据偏移 在对插入通道进行转换时, 将会从原始转换数据中按本位域的设置减掉偏移值。转换结果可以从 ADC_ICDx 寄存器读取。

4.7 ADC 看门狗高门限寄存器 (ADC_AWHT)

地址偏移: 0x24

复位值 : 0x0000 0FFF

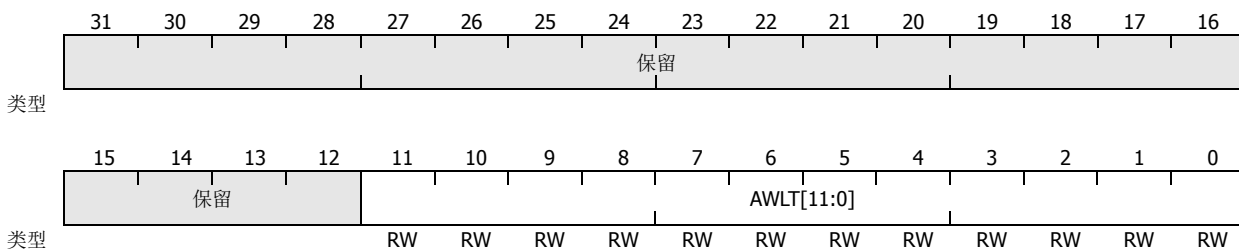


位/位域	名称	描述
31:12	保留	请保持其复位状态
11:0	AWHT[11:0]	模拟看门狗的高门限 本位域用于定义模拟看门狗的高门限。

4.8 ADC 模拟看门狗低门限寄存器 (ADC_AWLT)

地址偏移: 0x28

复位值 : 0x0000 0000

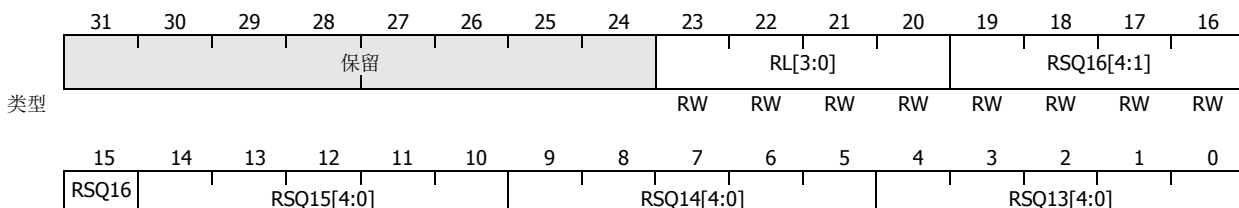


位/位域	名称	描述
31:12	保留	请保持其复位状态
11:0	AWLT[11:0]	模拟看门狗的低门限 本位域用于定义模拟看门狗的低门限。

4.9 ADC 常规序列寄存器 1 (ADC_RSQ1)

地址偏移: 0x2C

复位值 : 0x0000 0000



类型	[0]	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
----	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

位/位域	名称	描述
31:24	保留	请保持其复位状态
23:20	RL	常规通道组长度 常规组的转换总数为 RL+1。
19:0	RSQn[4:0]	本位域用于设置常规通道组中第 n 个转换所占用的通道号，通道号的有效范围是 0~17。

4.10 ADC 常规序列寄存器 2 (ADC_RSQ2)

地址偏移: 0x30

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ12[4:0]				RSQ11[4:0]				RSQ10[4:1]					
类型 RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ10[0]	RSQ9[4:0]				RSQ8[4:0]				RSQ7[4:0]						
类型 RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW															

位/位域	名称	描述
31:30	保留	请保持其复位状态
29:0	RSQn[4:0]	本位域用于设置常规通道组中第 n 个转换所占用的通道号，通道号的有效范围是 0~17。

4.11 ADC 常规序列寄存器 3 (ADC_RSQ3)

地址偏移: 0x34

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ6[4:0]				RSQ5[4:0]				RSQ4[4:1]					
类型 RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ4[0]	RSQ3[4:0]				RSQ2[4:0]				RSQ1[4:0]						
类型 RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW RW															

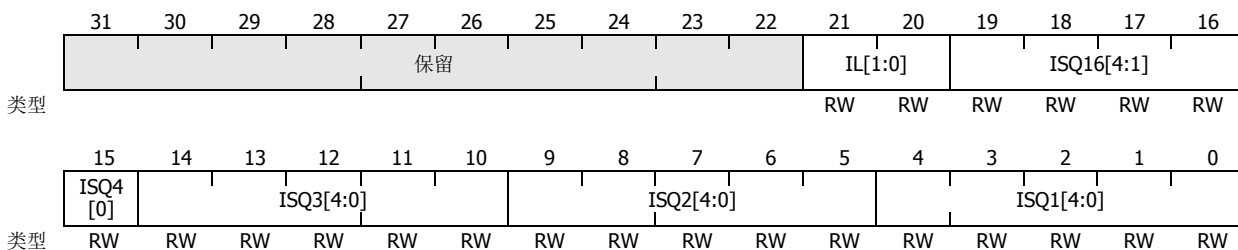
位/位域	名称	描述
------	----	----

位/位域	名称	描述
31:30	保留	请保持其复位状态
29:0	RSQn[4:0]	本位域用于设置常规通道组中第 n 个转换所占用的通道号，通道号的有效范围是 0~17。

4.12 ADC 插入序列寄存器 (ADC_ISQ)

地址偏移: 0x38

复位值 : 0x0000 0000

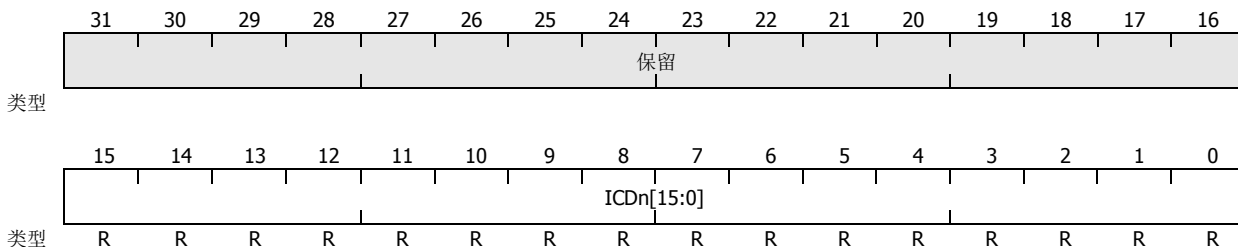


位/位域	名称	描述
31:22	保留	请保持其复位状态
21:20	IL	插入通道组长度 插入组的转换总数为 IL+1。
19:0	ISQn[4:0]	本位域用于设置插入通道组中第 n 个转换所占用的通道号，通道号的有效范围是 0~17。 与常规转换组序列不同的是，若 IL[1:0] 小于 4，则插入通道是从(4-IL)开始转换的。 IL 插入通道顺序 3 ISQ1 >> ISQ2 >> ISQ3 >> ISQ4 2 ISQ2 >> ISQ3 >> ISQ4 1 ISQ3 >> ISQ4 0 ISQ4

4.13 ADC 插入数据寄存器 x (ADC_IDRx) (x=1~4)

地址偏移: 0x3C~0x48

复位值 : 0x0000 0000



位/位域	名称	描述
31:16	保留	请保持其复位状态
15:0	ICDn[15:0]	插入组中第 n 个转换的数据 本位域仅可读。

4.14 ADC 常规数据寄存器 (ADC_RCD)

地址偏移: 0x4C

复位值 : 0x0000 0000



位/位域	名称	描述
31:16	ADC2RCD[15:0]	ADC2 常规组数据 本位域仅可读。 本位域对 ADC2 不可用。
15:0	RCD[15:0]	常规组数据 本位域仅可读。

第11章 I²C 接口

1 简介

GD32F103xx 系列微控制器的 I²C（Inter-Integrated Circuit，集成电路间总线）总线接口提供了一路符合工业标准的两线制串行接口，MCU 可通过该总线接口与外部的 I²C 设备进行通信。I²C 总线只需要两根串行线：串行数据线 SDA 以及串行时钟线 SCL。

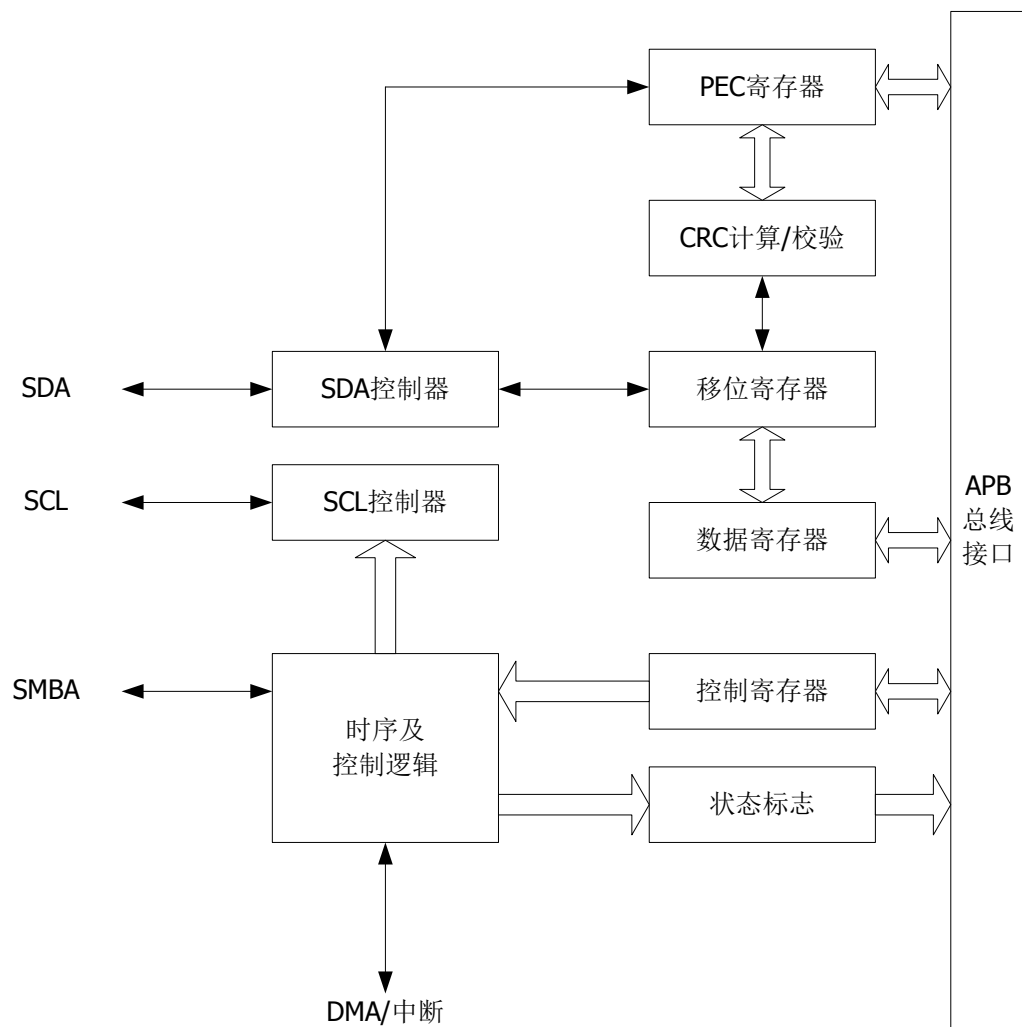
I²C 总线接口模块实现了符合标准 I²C 协议的标速或高速接口，并具备 CRC 计算及校验功能、支持 SMBus（System Management Bus，系统管理总线）以及 PMBus（Power Management Bus，电源管理总线），此外还支持多主的 I²C 总线架构。I²C 总线接口模块支持 DMA 模式，可降低 CPU 负荷率。

2 主要特点

- 并行总线至 I²C 总线协议的转换及接口
- 同一接口既可实现 I²C 主机功能、又可实现 I²C 从机功能
- 主机与从机之间可实现双向数据传输
- 支持 7 位或 10 位地址，支持广播呼叫寻址
- 支持 I²C 的多主模式
- 支持标速（最高 100kHz）和高速（最高 400kHz）
- 从机模式下可配置 SCL 主动拉低
- 支持 DMA 模式
- 兼容 SMBus 2.0 以及 PMBus
- 提供 2 个中断：字节成功发送中断以及错误事件中中断
- 可选的 PEC（报文错误检测）生成及校验功能

3 功能描述

下图详细描绘了 I²C 接口的内部配置。

图 114 I²C 模块内部框图

 表 44 I²C 总线相关术语定义

术语	描述
发送方	将数据发送到总线上的设备
接收方	从总线上接受数据的设备
主机	发起传输、产生时钟信号并且终止传输的设备
从机	被主机寻址的设备
多主	同一时刻可以有多个主机尝试获取总线控制权，并且不会破坏正在传输的报文。
同步	在两个或两个以上的设备中同步时钟信号的步骤
仲裁	当不止一个主机同时尝试控制总线时，确保只有一个主机被允许成功获取总线控制权、且竞争成功的报文免遭破坏的步骤。

3.1 SDA 及 SCL 口线

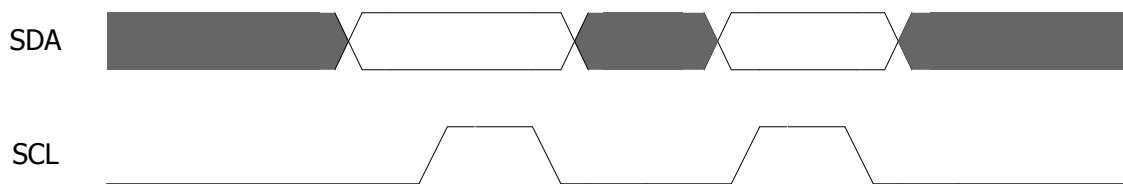
I²C 模块有两个外部口线，分别是串行数据线 SDA 以及串行时钟线 SCL。这两根口线在连接到总线的设备之间传递信息。

SDA 和 SCL 都是双向口线，通过电流源或上拉电阻连接到电源正极。当总线空闲时，两根口线的状态都是高电平。连接到总线上的设备的输出级必须为漏极开路或集电极开路结构，以实现多设备的线与特性。I²C 总线上的数据传输率在标速模式下可达 100 kbit/s，在高速模式下可达 400 kbit/s。由于各种采用不同制造工艺（CMOS、NMOS、双极晶体管）的器件都有可能接入总线，因此代表逻辑 0 的电平（低电平）和代表逻辑 1 的电平（高电平）并不是固定的，取决于相关的 V_{DD} 电压值。

3.2 数据有效性

SDA 线上的数据在时钟信号（SCL）为高电平期间必须保持稳定。数据线的电平状态只能在 SCL 为低电平期间切换，如图所示。每传输一个数据位，就需要产生一个时钟脉冲。

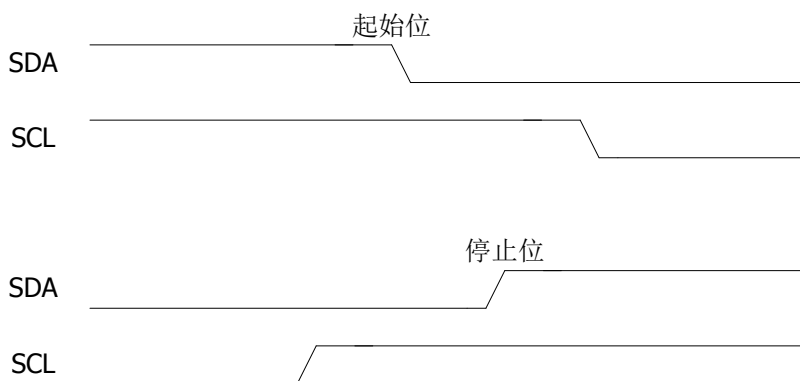
图 115 I²C 数据有效性



3.3 起始位与停止位

所有会话必须以起始位（S）开始，以停止位（P）结束，如下图所示。当 SCL 保持高电平时，SDA 从高电平到低电平的跳变沿就是起始位。当 SCL 保持高电平时，SDA 从低电平到高电平的跳变沿就是停止位。

图 116 起始位与停止位

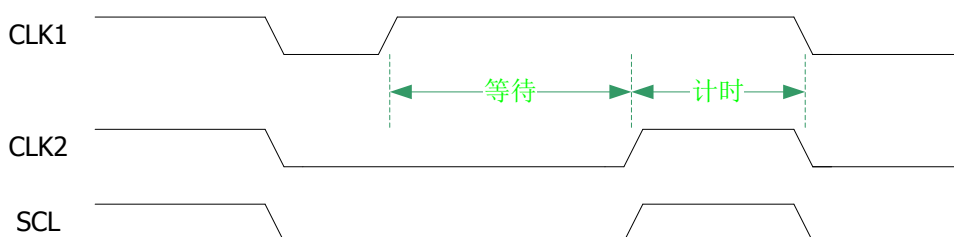


3.4 时钟同步

两个主机可能会同时在一空闲总线上发起会话，因此必须通过某种机制决定哪一主机可以获取总线控制权并完成其发送。这是通过时钟同步及仲裁实现的。在单主机系统中就无需用到时钟同步及仲裁了。

时钟同步是利用 I²C 总线接口 SCL 口线的线与功能实现的。每当 SCL 从高电平拉低时，准备发送的主机便开始对 SCL 的低电平时间计数。一旦某个主机时钟拉低，其输出即可将 SCL 线拉低并保持，直到其自身时钟输出高电平状态为止，如下图所示。不过，假如此时其它主机的时钟仍处于低电平，那么先输出低电平到高电平的主机就无法改变 SCL 线的状态。因此，SCL 将始终由低电平周期最长的主机拉低。在此期间，低电平周期较短的主机会进入高电平的等待状态。

图 117 时钟同步



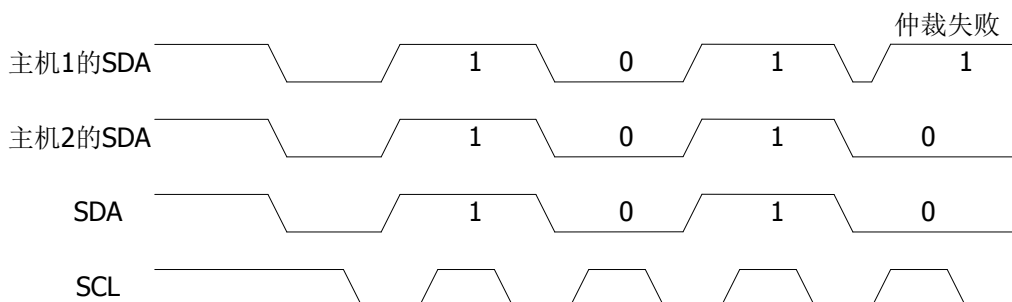
3.5 仲裁

仲裁与时钟同步类似，都只着眼于协议中多主机的相关内容。从机并不涉及仲裁流程。

只有在总线空闲时，主机才可以发起传输。两个主机可能会在起始位的最小保持时间内都产生起始位，于是总线上便会产生有效起始位。之后就要由仲裁机构判定允许哪一主机顺利完成其发送。

仲裁机构是逐位进行判定的。在每个位的传输期间，当 SCL 为高电平时，各主机分别检查 SDA 的电平是否与自身发送的电平相符。仲裁过程可能需要经过多个位的判定后才能完成。实际上，如果两个主机所发送的内容完全相同，那么它们甚至可以完成整个会话而不发生冲突。若某个主机发送高电平、却检测到低电平，该主机就认为自身仲裁失败，并立即关闭其 SDA 输出驱动部分。而赢得仲裁的主机将继续完成其会话。

图 118 SDA 线仲裁



3.6 I²C 通信流

每个 I²C 设备（不管是微控制器、LCD 驱动器、存储器或键盘接口芯片等）都通过一个唯一的地址进行识别，取决于设备的功能则作为通信的发送方或是接收方。

I²C 从机从总线上接收到起始位之后，接下来便会接收地址，并且将收到的地址与本机地址（通过软件编程）进行比较。一旦地址匹配，I²C 从机将向 I²C 总线发送一个确认应答位（ACK），并响应总线的后续命令（发送或接收特定数据）。此外，假如软件开启了广播呼叫功能，I²C 从机将始终响应广播呼叫地址（0x00）。I²C 接口模块支持 7 位地址以及 10 位地址。

I²C 主机始终通过起始位开始传输，停止位终止传输。I²C 主机还负责产生 SCL 信号。

图 119 采用 7 位地址的 I²C 通信流

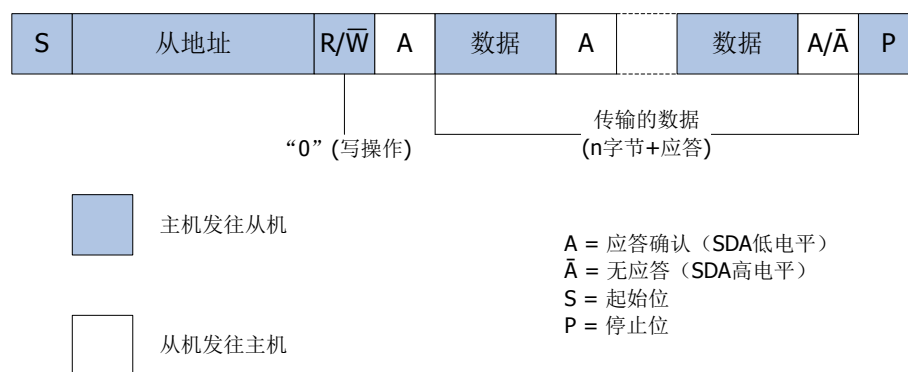
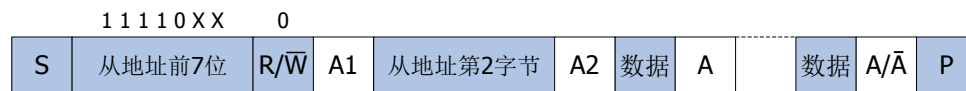


图 120 采用 10 位地址的 I²C 通信流



3.7 程序流模型

像 LCD 驱动器这样的 I²C 设备可能永远是总线的接收方，而存储器则既接收数据又发送数据。除了从发送方/接收方来区分，器件在进行数据传输时还可以分别视作主机或从机。所谓主机，就是指在总线上发起数据会话并产生时钟信号以执行传输的设备。此时，任何被寻址的器件均视为从机。

不管 I²C 设备是主机还是从机，它都有可能发送或接收数据，因此 I²C 设备会有四种工作模式：

- 主机发送方
- 主机接收方
- 从机发送方
- 从机接收方

GD32F103xx 系列微控制器的 I²C 模块支持以上全部工作模式。在系统复位后，I²C 模块默认工作

于从机模式。假如用户程序使 I²C 模块在总线上发送一个起始位，那么 I²C 模块将切换为主机模式。当用户程序操作 I²C 模块在总线上发送停止位后，I²C 模块将切换回从机模式。

从机发送方模式下的程序流模型

在从机模式下，要向总线发送数据，用户程序必须按照以下步骤操作 I²C 模块：

1. 首先，软件应使能 I²C 外设时钟，并在 I2C_CR2 寄存器中配置时钟相关的寄存器，以确保 I²C 时序正确。使能并配置 I²C 模块后，它将按照默认的从机模式工作，等待总线上的起始位以及地址。
2. 当 I²C 接收到起始位，并接收到匹配的地址（要么 7 位模式要么 10 位模式）之后，I²C 硬件会将 I²C 状态寄存器的 ADDR 置位，用户程序应以查询方式或中断方式获取该标志的状态，并在该标志置位后先读 SR1 再读 SR2 实现 ADDR 位清零。假如地址为 10 位模式，I²C 主机会产生一个重起始位，并向总线再发送一个地址头。从机检测到重起始位后会再次将 ADDR 置位。于是软件还要再次通过依次读取 SR1 和 SR2 将 ADDR 清零。
3. 现在 I²C 总线进入数据发送阶段，硬件会将 TXE 置位，因为此时从机的移位寄存器和数据寄存器 DR 都是空的。软件将第一个字节写入 DR 寄存器中，但 TXE 并不会马上清零，因为写入 DR 的字节会立即被送入内部移位寄存器中。一旦移位寄存器非空，I²C 就开始向总线发送数据。
4. 在第一个字节传输期间，软件应当向 DR 写入第二个字节。此时由于 DR 和移位寄存器都有内容，TXE 将会清零。
5. 当 TXE 置位后，只要还有待发送的数据，软件便可以继续向 DR 写入下一个字节。
6. 在发送倒数第二个字节时，软件将最后一个字节写入 DR 并清零 TXE 标志，此后不再需要检测 TXE 的状态。这样 TXE 在发送完倒数第二字节后置位，直到停止位之后才清零。
7. 按照协议，I²C 主机并不会确认最后一个字节，因此当发送完最后一个字节后，I²C 从机等待总线上出现停止位，之后将 AF（应答错误）置位以通知软件发送全部完成。软件对 AF 写 0，即可将其清零。

图 121 从机发送的程序流模型



从机接收方模式下的程序流模型

在从机模式下，要从总线接收数据，用户程序必须按照以下步骤操作 I²C 模块：

1. 首先，软件应使能 I²C 外设时钟，并在 I2C_CR2 寄存器中配置时钟相关的寄存器，以确保 I²C 时序正确。使能并配置 I²C 模块后，它将按照默认的从机模式工作，等待总线上的起始位以及地址。
2. 当 I²C 接收到起始位，并接收到匹配的地址（要么 7 位模式要么 10 位模式）之后，I²C 硬件会将 I²C 状态寄存器的 ADDR 置位，用户程序应以查询方式或中断方式获取该标志的状态，并在该标志置位后先读 SR1 再读 SR2 实现 ADDR 位清零。ADDR 清零后，I²C 模块将立即开

始接收总线上的数据。

3. 接收到首个数据字节后，硬件立即将 **RXNE** 置位。现在软件可以从 **DR** 读出首个数据字节，之后 **RXNE** 将清零。
4. 每当 **RXNE** 置位，软件就可以从 **DR** 中读出一个数据字节。
5. 最后一字节接收完成后，**RXNE** 置位。软件读出最后一字节。
6. 当 **I²C** 模块检测到总线上的停止位后，会将 **STOPF** 置位。软件可先读 **SR1** 再写 **CR1** 来清零 **STOPF** 标志位。

图 122 从机接收的程序流模型



主机发送方模式下的程序流模型

在主机模式下，要向总线发送数据，用户程序必须按照以下步骤操作 **I²C** 模块：

1. 首先，软件应使能 **I²C** 外设时钟，并在 **I2C_CR2** 寄存器中配置时钟相关的寄存器，以确保 **I²C** 时序正确。使能并配置 **I²C** 模块后，它将按照默认的从机模式工作，等待总线上的起始位以

及地址。

2. 软件将 **START** 置位，请求 **I²C** 模块在总线上产生一个起始位。
3. 发送完起始位后，**I²C** 硬件将 **I²C** 状态寄存器中的 **SB** 标志置位，并进入主机模式。现在软件应先读取 **SR1**、再将 7 位地址（或 10 位地址头）写入 **DR**，从而将 **SB** 清零。一旦 **SB** 标志清零，**I²C** 硬件将立即开始在总线上发送 7 位地址或地址头。假如所发送的地址为 10 位，那么硬件会将 **ADDR10** 置位，发送完地址头后，软件应通过先读取 **SR1**、再将 10 位低地址写入 **DR** 的方式将 **ADDR10** 清零。
4. 7 位或 10 位地址发送完成后，**I²C** 硬件将 **ADDR** 置位，软件应依次读取 **SR1**、**SR2**，从而将 **ADDR** 清零。
5. 现在 **I²C** 总线进入数据发送阶段，硬件会将 **TXE** 置位，因为此时从机的移位寄存器和数据寄存器 **DR** 都是空的。软件将第一个字节写入 **DR** 寄存器中，但 **TXE** 并不会马上清零，因为写入 **DR** 的字节会立即被送入内部移位寄存器中。一旦移位寄存器非空，**I²C** 就开始向总线发送数据。
6. 在第一个字节传输期间，软件应当向 **DR** 写入第二个字节。此时由于 **DR** 和移位寄存器都有内容，**TXE** 将会清零。
7. 当 **TXE** 置位后，只要还有待发送的数据，软件便可以继续向 **DR** 写入下一个字节。
8. 在发送倒数第二个字节时，软件将最后一个字节写入 **DR** 并清零 **TXE** 标志，此后不再需要检测 **TXE** 的状态。这样 **TXE** 在发送完倒数第二字节后置位，直到停止位之后才清零。
9. 发送完最后一字节后，由于移位寄存器和 **DR** 寄存器都已经为空，**I²C** 主机会将 **BTF** 置位。软件应当向 **I²C** 模块请求一个停止位，当发送完停止位后，**I²C** 模块将 **TXE** 和 **BTF** 都清零。

图 123 主机发送的程序流模型



主机接收方模式下的程序流模型

在主机模式下, 要从总线接收数据, 用户程序必须按照以下步骤操作 I²C 模块:

1. 首先, 软件应使能 I²C 外设时钟, 并在 I2C_CR2 寄存器中配置时钟相关的寄存器, 以确保 I²C 时序正确。使能并配置 I²C 模块后, 它将按照默认的从机模式工作, 等待总线上的起始位以

及地址。

2. 软件将 **START** 置位，请求 **I²C** 模块在总线上产生一个起始位。
3. 发送完起始位后，**I²C** 硬件将 **I²C** 状态寄存器中的 **SB** 标志置位，并进入主机模式。现在软件应先读取 **SR1**、再将 7 位地址（或 10 位地址头）写入 **DR**，从而将 **SB** 清零。一旦 **SB** 标志清零，**I²C** 硬件将立即开始在总线上发送 7 位地址或地址头。假如所发送的地址为 10 位，那么硬件会将 **ADDR10** 置位，发送完地址头后，软件应通过先读取 **SR1**、再将 10 位低地址写入 **DR** 的方式将 **ADDR10** 清零。
4. 7 位或 10 位地址发送完成后，**I²C** 硬件将 **ADDR** 置位，软件应依次读取 **SR1**、**SR2**，从而将 **ADDR** 清零。假如地址为 10 位模式，软件应当再次将 **START** 置位，在 **I²C** 总线上产生重起始位。当重起始位发送完毕后，**SB** 将会置位。软件应通过先读取 **SR1**、再将 10 位低地址写入 **DR** 的方式将 **SB** 清零。随后地址头将发送到总线上，且 **ADDR** 再次置位。软件再次依次读取 **SR1**、**SR2**，从而将 **ADDR** 清零。
5. 接收到首个数据字节后，硬件立即将 **RXNE** 置位。现在软件可以从 **DR** 读出首个数据字节，之后 **RXNE** 将清零。
6. 每当 **RXNE** 置位，软件就可以从 **DR** 中读出一个数据字节。
7. 倒数第二字节接收完成后，软件应当将 **ACK** 清零，并将 **STOP** 置位。
8. 最后一字节接收完成后，**RXNE** 置位。软件读出最后一字节。**I²C** 模块对最后一字节不发送应答，并且在最后一字节接收完成后产生停止位。

图 124 主机接收的程序流模型



DMA 模式下的程序流模型

从前面程序流模型的图中可以看出, 每当 TXE 或 RXNE 置位时, 软件都需要写入或读出一个字节, 这样可能导致 CPU 负荷过重。可利用 DMA 来处理 TXE 和 RXNE 标志: 每当 TXE 或 RXNE 置位时, DMA 自动执行读取或写入操作。

3.8 报文错误校验

I²C 模块中集成了 CRC-8 计算单元，可以对 I²C 数据进行报文错误校验。CRC 多项式为 $x^8 + x^2 + x + 1$ ，可以看出该多项式是与 SMBus 兼容的。将 ENPEC 置位，即可使能 CRC 校验功能，此时 PEC 将会对所有通过 I²C 传输的数据进行计算，包括地址。I²C 模块可以在发送完最后一个数据字节之后将 PEC 计算结果发送出去，或者通过 PEC 位将收到的 PEC 值与自身的校验结果进行比较。在 DMA 模式下，若 ENPEC 置位，则 I²C 模块将会自动发送或接收 PEC 值。

3.9 对 SMBus 的支持

系统管理总线（System Management Bus，简称为 SMBus 或 SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信。SMBus 是 I²C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见 Smart Battery Data）。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I²C 规范中数据传输格式的子集。只要 I²C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不符合这些协议的 I²C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus 采用了 I²C 硬件以及 I²C 的硬件寻址方式，但在 I²C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。协议中有个非常特别之处在于：系统的主机和所有其它设备能够有定义其名称和功能。

超时特性

SMBus 有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为 10 kHz——为了防止长时间锁死总线。I²C 在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将继续 I²C 会话。I²C 总线协议中并没有限制这个延时的上限，但在 SMBus 系统中，这个时间被限定为 35 ms。按照 SMBus 协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就并不允许从设备将时钟拉低太长时间。

报文错误校验

SMBus 2.0 以及 1.1 都采用了报文错误校验（Packet Error Checking，缩写为 PEC）。在这种模式中，每次会话最后都将传输 PEC（报文错误码）字节。该字节是按照 CRC-8 校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为 $x^8 + x^2 + x + 1$ （CRC-8-ATM HEC 算法，初始化为 0）。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus 中还定义了较少见的“主机提醒协议（Host Notify Protocol）”，基于 I²C 多主模式实现类似的提醒功能，但是可以传递更多数据。

3.10 状态、错误以及中断

I²C 中有多个状态标志和错误标志，通过设置一些寄存器位，便可以从这些标志触发中断（详见 I²C 寄存器说明）。

表 45 事件状态标志

事件标志名称	描述
SB	已发送起始位（主机）
ADDR	地址已发送或已接收
ADDR10	10 位地址头已发送
STOPF	检测到停止位
BTF	字节发送结束
TXE	发送时，DR 已空
RXNE	接收时，DR 非空

表 46 I²C 错误标志

I ² C 错误名称	描述
BERR	总线错误
ARLO	仲裁失败
OVR	当禁用 SCL 拉低后，发生了溢出或下溢
AF	未收到应答确认
PECERR	CRC 校验值不符
TIMEOUT	SMBus 模式下的总线超时
SMBALERT	SMBus 警报

4 I²C 寄存器

4.1 I²C 控制寄存器 1 (I2C_CR1)

地址偏移: 0x00

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWRST	保留	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGCR	ENPEC	ENARP	SMBTYPE	保留	SMBUS	PE
类型	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW

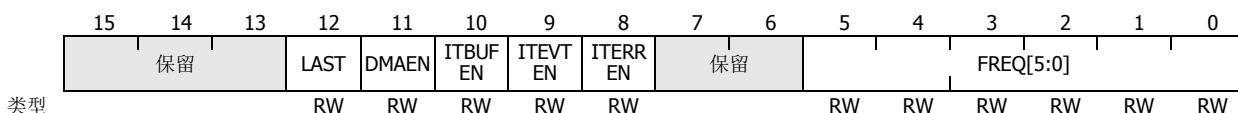
位/位域	名称	描述
15	SWRST	软件复位 I ² C, 软件应当等到 I ² C 线被释放之后再复位 I ² C。 0: I ² C 并未复位 1: I ² C 正在复位
14	保留	请保持其复位状态
13	ALERT	本位可由软件复位或置位, 硬件会将本位清零。 0: 不通过 SMBA 管脚产生警报 1: 通过 SMBA 管脚产生警报
12	PEC	报文错误校验 本位可由软件复位或置位。硬件会在 PEC 已发送、或检测到起始位/停止位、或 PE=0 时将本位清零。 0: 不发送 PEC 值 1: 发送 PEC 值
11	POS	ACK 的位置/PEC 的含义 本位可由软件复位或置位。硬件会在 PE=0 时将本位清零。 0: ACK 位决定对当前正在接收的字节是否发送 ACK; PEC 位表明 PEC 是否处于移位寄存器中。 1: ACK 位决定对下一字节是否发送 ACK; PEC 位表明下一个要接收的字节是 PEC。
10	ACK	是否发送 ACK 本位可由软件复位或置位。硬件会在 PE=0 时将本位清零。 0: 不发送 ACK 1: 发送 ACK
9	STOP	在 I ² C 总线上产生停止位 本位可由软件复位或置位。当 SMBU 超时时, 由硬件置位; 当检测到停止位时, 由硬件自动清零。 0: 不发送停止位 1: 发送停止位
8	START	在 I ² C 总线上产生起始位 本位可由软件复位或置位。硬件会在检测到起始状态、或 PE=0 时将本位置位。 0: 不发送起始位 1: 发送起始位
7	NOSTRETCH	从机模式下, 当数据未准备好时, 是否拉低 SCL 本位可由软件复位或置位。 0: 使能 SCL 主动拉低 1: 禁用 SCL 主动拉低

位/位域	名称	描述
6	ENGC	是否响应广播呼叫 (0x00) 0: 从机不响应广播呼叫 1: 从机响应广播呼叫
5	ENPEC	PEC 计算开关 0: 关闭 PEC 计算功能 1: 开启 PEC 计算功能
4	ENARP	SMBus 的 ARP 协议开关 0: 禁用 ARP 1: 使能 ARP
3	SMBTYPE	SMBus 类型 0: 从机 1: 主机
2	保留	请保持其复位状态
1	SMBUS	SMBus/I ² C 模式选择 0: I ² C 模式 1: SMBus 模式
0	PE	I ² C 外设使能 0: 禁用 I ² C 1: 使能 I ² C

4.2 I²C 控制寄存器 2 (I2C_CR2)

地址偏移: 0x04

复位值 : 0x0000



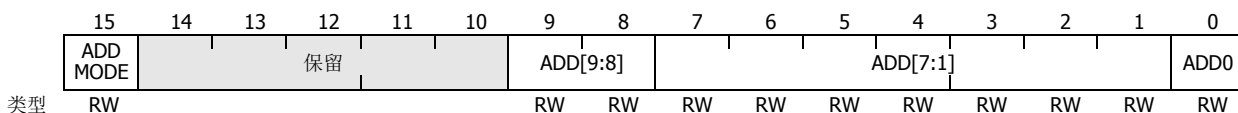
位/位域	名称	描述
15:13	保留	请保持其复位状态
12	LAST	表明 DMA 最后一次传输的标志 0: 下一次 DMA EOT 并非最后一次传输 1: 下一次 DMA EOT 是最后一次传输
11	DMAEN	DMA 模式开关 0: 禁用 DMA 模式 1: 使能 DMA 模式
10	ITBUFEN	缓冲中断使能 0: 当 TXE=1 或 RXNE=1 时, 不产生中断 1: 假如 ITEVTEN=1, 则当 TXE=1 或 RXNE=1 时, 产生中断

位/位域	名称	描述
9	ITEVTEN	事件中断使能 0: 禁用事件中断 1: 使能事件中断, 也就是说当 SB、ADDR、ADD10、STOPF 或 BTF 置位时, 将会产生中断, 或者当 ITBUFEN=1 时, 若 TXE=1 或 RXNE=1, 也会产生中断
7:6	保留	请保持其复位状态
5:0	FREQ[5:0]	I2C 外设时钟频率 FREQ[5:0]应当是输入 APB 时钟的频率, 以 MHz 为单位, 有效范围 2~32。 0~1: 保留, 禁止使用 2~36: 2MHz~36MHz 37~63: 保留, 禁止使用 (APB 时钟不支持如此高的频率)

4.3 I²C 自身地址寄存器 1 (I2C_OAR1)

地址偏移: 0x08

复位值 : 0x0000

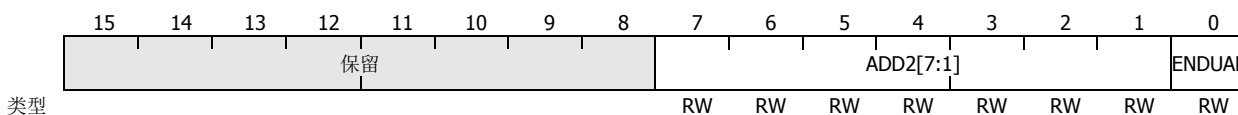


位/位域	名称	描述
15	ADDMODE	I ² C 从机地址模式 0: 7 位地址 1: 10 位地址
14:10	保留	请保持其复位状态
9:8	ADD[9:8]	10 位地址的高 2 位
7:1	ADD[7:1]	7 位地址, 或 10 位地址的低 7 位
0	ADD0	I ² C 地址的读/写位

4.4 I²C 自身地址寄存器 2 (I2C_OAR2)

地址偏移: 0x0C

复位值 : 0x0000



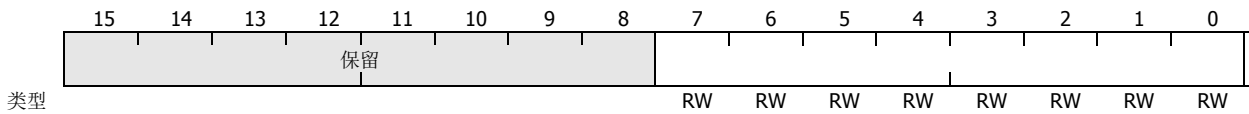
位/位域	名称	描述
15:8	保留	请保持其复位状态

位/位域	名称	描述
7:1	ADD2[7:1]	双地址模式下从机的第二个地址
0	ENDUAL	双地址模式开关 0: 禁用双地址模式 1: 使能双地址模式

4.5 I²C 数据寄存器 (I2C_DR)

地址偏移: 0x10

复位值 : 0x0000

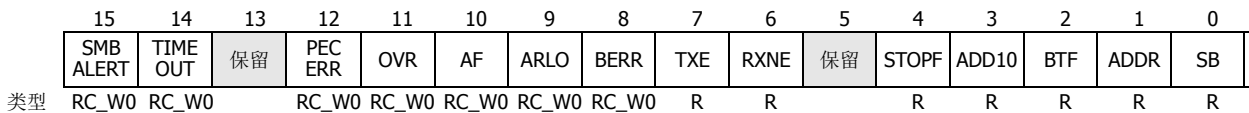


位/位域	名称	描述
15:8	保留	请保持其复位状态
7:0	DR[7:0]	收发的数据寄存器

4.6 I²C 状态寄存器 1 (I2C_SR1)

地址偏移: 0x14

复位值 : 0x0000



位/位域	名称	描述
15	SMBALERT	SMBus 警报状态 本位由硬件置位，由软件清零。 0: SMBA 管脚未拉低（从机模式），或未检测到警报（主机模式） 1: SMBA 管脚拉低（从机模式），或检测到警报（主机模式）
14	TIMEOUT	SMBus 模式下的超时信号 本位由硬件置位，由软件清零。 0: 无超时错误 1: 发生超时事件（SCL 拉低超过 25 ms）
13	保留	请保持其复位状态
12	PECERR	接收数据时的 PEC 错误 本位由硬件置位，由软件清零。 0: 收到 PEC 且校验正确 1: 收到 PEC 且校验错误，此时 I ² C 不管 ACK 位的状态如何都将发送 NACK。

位/位域	名称	描述
11	OVR	<p>当禁用 SCL 拉低功能后，在从机模式下发生了过载或欠载事件。在从机发送模式下，假如 DR 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生过载。在从机接收模式下，假如当前字节已经发送完成，而 DR 仍然为空，就会发生欠载。</p> <p>本位由硬件置位，由软件清零。 0: 无过载或欠载事件 1: 发生过载或欠载事件</p>
10	AF	<p>应答错误 本位由硬件置位，由软件清零。 0: 无应答错误 1: 出现应答错误</p>
9	ARLO	<p>主机模式下仲裁失败 本位由硬件置位，由软件清零。 0: 无仲裁失败 1: 发生仲裁失败，I²C 模块返回从机模式</p>
8	BUSERR	<p>总线错误，表明总线上出现预料之外的起始位或停止位 本位由硬件置位，由软件清零。 0: 无总线错误 1: 检测到总线错误</p>
7	TXE	<p>发送时 DR 为空 当硬件将一个字节从 DR 挪到移位寄存器之后，此标志置位。当对 DR 写入一个字节后，此标志清零。假如移位寄存器和 DR 都为空，那么写 DR 也不能将本标志清零（详见程序流模型）。 0: DR 非空 1: DR 为空，软件可向 DR 写入数据</p>
6	RXNE	<p>接收时 DR 非空 当硬件将一个字节从移位寄存器挪到 DR 之后，此标志置位。当读取 DR 后，此标志清零。假如 BTF 和 RXNE 都已经置位，那么读 DR 也不能将本标志清零，因为移位寄存器中的数据会立即挪到 DR 中。 0: DR 空 1: DR 非空，软件可从 DR 读出数据</p>
5	保留	<p>请保持其复位状态</p>
4	STOPF	<p>从机模式下检测到停止位 本位由硬件置位，清零时需先读 SR1 再读 SR2。 0: 未在从机模式下检测到停止位 1: 在从机模式下检测到停止位</p>
3	ADD10	<p>主机模式下已发送 10 位地址头 本位由硬件置位，清零时需先读 SR1 再写 DR。 0: 在主机模式下未发送 10 位地址头 1: 在主机模式下已发送 10 位地址头</p>
2	BTF	<p>字节发送结束 在接收模式下，假如移位寄存器中已经收到一个字节、但 DR 中也有数据，则当 SCL 拉低功能使能的情况下，本标志由硬件置位。 在发送模式下，假如移位寄存器中的字节已经发出、但 DR 中尚无新的数据，则在 SCL 拉低功能使能的情况下，本标志由硬件置位。 本位由硬件置位，零时需先读 SR1 再写 DR。 0: 未产生 BTF 1: 已产生 BTF</p>

位/位域	名称	描述
1	ADDR	主机模式下已发送地址，或从机模式下收到匹配地址 本位由硬件置位，清零时需先读 SR1 再读 SR2。 0: 无发送或接收的地址 1: 主机模式下已发送地址，或从机模式下已收到匹配地址
0	SB	主机模式下已发送起始位 本位由硬件置位，清零时需先读 SR1 再写 DR。 0: 未发送起始位 1: 已发送起始位

4.7 I²C 状态寄存器 2 (I2C_SR2)

地址偏移: 0x18

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEC[7:0]								DUALF	SMBHOST	SMBDEFAULT	GENCALL	保留	TRA	BUSY	MSL
类型	R	R	R	R	R	R	R	R	R	R	R	R		R	R	R

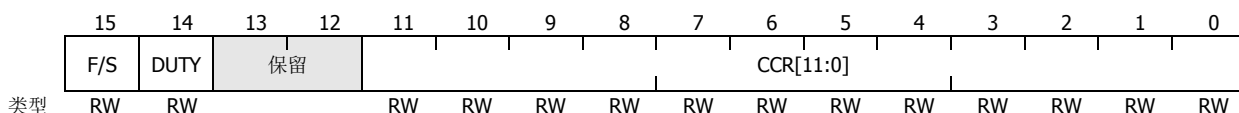
位/位域	名称	描述
15:8	PEC[7:0]	使能 PEC 后，本位域包含硬件计算的 PEC 值。
7	DUALF	双地址标志 双地址模式下，本标志用于指示收到的地址与哪一地址匹配。 检测到起始位、停止位或 PE=0 时，本位由硬件置位。 0: 与 OAR1 地址匹配 1: 与 OAR2 地址匹配
6	SMBHOST	从机模式下的 SMBus 主机头 检测到起始位、停止位或 PE=0 时，本位由硬件置位。 0: 未检测到 SMBus 主机头 1: 检测到 SMBus 主机头
5	SMBDEFAULT	从机模式下的 SMBus 设备默认地址 检测到起始位、停止位或 PE=0 时，本位由硬件置位。 0: 目前尚无 SMBus 设备默认地址 1: 收到了 SMBus 设备默认地址
4	GENCALL	是否收到了广播地址 (0x00) 检测到起始位、停止位或 PE=0 时，本位由硬件置位。 0: 未收到广播地址 (0x00) 1: 已收到广播地址 (0x00)
3	保留	请保持其复位状态
2	TRA	I ² C 是发送方还是接收方 检测到起始位、停止位、PE=0 或 ARLO 时，本位由硬件置位。 0: 接收方 1: 发送方

位/位域	名称	描述
1	BUSY	忙标志 检测到停止位后，本位由硬件置位。 0: 无 I ² C 通信 1: 正在进行 I ² C 通信
0	MSL	I ² C 是主机还是从机 检测到起始位、停止位、PE=0 或 ARLO 时，本位由硬件置位。 0: 从机模式 1: 主机模式

4.8 I²C 时钟控制寄存器 (I2C_CCR)

地址偏移: 0x1C

复位值 : 0x0000



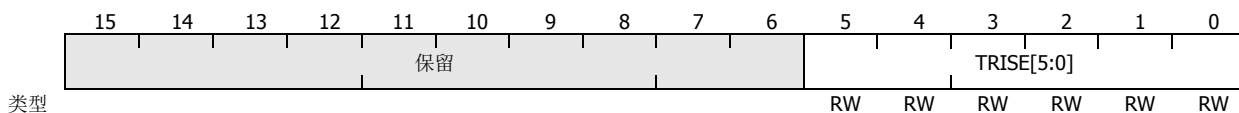
类型

位/位域	名称	描述
15	F/S	主机模式下的 I ² C 速度选择 0: 标速模式 1: 高速模式
14	DUTY	高速模式下的占空比配置 0: $T_{low}/T_{high} = 2$ 1: $T_{low}/T_{high} = 16/9$
13:12	保留	请保持其复位状态
11:0	CCR[11:0]	主机模式下的 I ² C 速度配置 在标速模式下: $T_{high} = T_{low} = CCR * T_{PCLK1}$ 在高速模式下, 若 DUTY=0, 则 $T_{high} = CCR * T_{PCLK1}$, $T_{low} = 2 * CCR * T_{PCLK1}$ 若 DUTY=1, 则 $T_{high} = 9 * CCR * T_{PCLK1}$, $T_{low} = 16 * CCR * T_{PCLK1}$

4.9 I²C TRISE 寄存器 (I2C_TRISE)

地址偏移: 0x20

复位值 : 0x0002



类型

位/位域	名称	描述
15:6	保留	请保持其复位状态

位/位域	名称	描述
5:0	TRISE[5:0]	主机模式下的最大上升沿时间 TRISE 值应设置为 SCL 上升时间最大值加 1。

第12章 SPI 接口

1 简介

串行外设接口 (Serial Peripheral Interface, 缩写为 SPI) 提供了 SPI 协议的数据发送及接收功能, 可工作在主机或从机模式下。SPI 接口用到 4 根管脚, 分别是串行数据输入脚 MISO、串行数据输出脚 MOSI、串行时钟脚 SCK 以及从机选择脚 NSS。总线上的一个 SPI 设备作为主机工作, 通过 NSS 以及 SCK 信号控制数据流, 指示数据通信的开始, 并控制数据采样速率。要接收一个数据字节, 数据位流将在指定的时钟边沿锁存, 并将结果保存到数据寄存器或 RX FIFO 中。数据发送的执行方式类似, 但顺序与之相反。模式故障检测功能实现了多主功能。SPI 接口可用于多种应用, 包括只使用两线的单工同步通信 (其中一根数据线可能双向), 或采用 CRC 校验的可靠数据通信。

2 主要特点

- 主机或从机工作
- 可编程的时钟位速率以及预分频系数
- 可编程的时钟极性和时钟相位
- 16 位宽、相互独立的发送和接收缓冲
- 8~16 位可编程的数据帧长度
- 高位在前或低位在前可编程的数据顺序
- 硬件 CRC 计算, 可自动发送 CRC 错误校验
- 通过 3 根口线实现全双工同步传输
- 通过 2 根口线实现单工同步传输
- 主机和从机都可配置 NSS 工作于软件模式或硬件模式
- SPI 总线忙标志
- 发送和接收标志, 并可映射至中断
- 主机模式故障、过载以及 CRC 校验错误标志, 并可映射至中断
- 可通过 DMA 发送或接收

3 功能描述

3.1 管脚配置

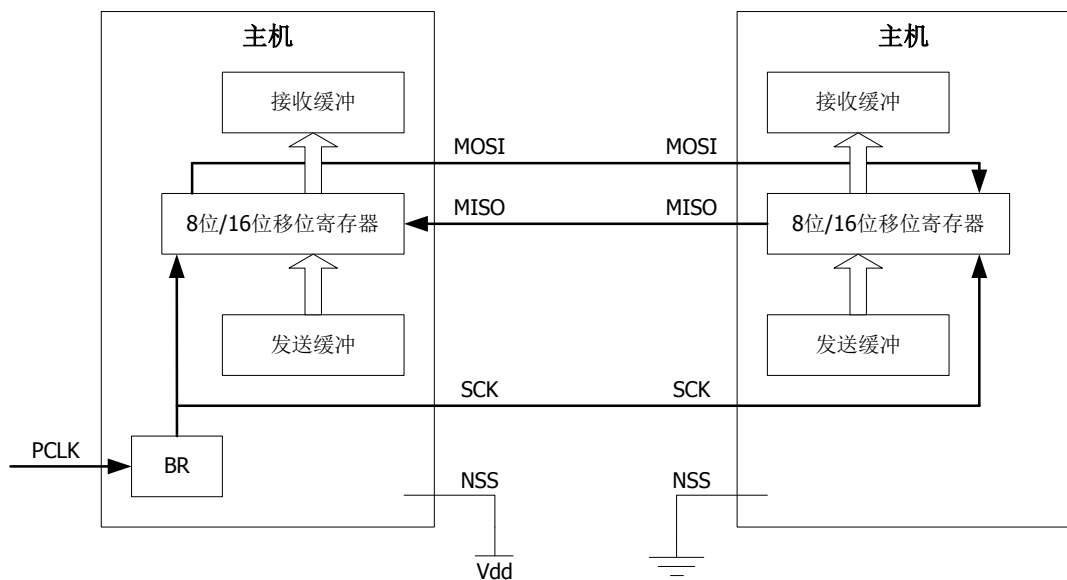
在不同模式下, SPI 可通过 2~4 根管脚连接外部器件:

- MISO: 此管脚用于在主机模式下接收数据 (Master In), 在从机模式下发送数据 (Slave Out)。

- **MOSI:** 此管脚用于在主机模式下发送数据(Master Out),在从机模式下接收数据(Slave In)。
- **SCK:** 此管脚用于在主机模式下输出时钟,在从机模式下接收时钟。
- **NSS:** 在硬件模式下(SSM 清零),NSS 管脚可作为输入脚使用。主机模式下 NSS 管脚应拉高,从机模式下 NSS 管脚应拉低。假如在主机模式下将 NSS 拉低,则将产生错误,此时 MODF 将会置位,硬件自动将 MSTR 清零。假如在从机模式下 NSS 为高电平,意味着该芯片未被选中,SPI 模块要等到 NSS 管脚被外部拉低后才能工作。在软件模式下(SSM 置位),SSI 位将替代 NSS 管脚的功能,NSS 可用作标准 I/O 口。此外,在主机模式下 NSS 可用作输出。若 SSOE 置位,当 SPI 通信开始后 NSS 管脚将拉低。

单主单从的典型连接方式如下图所示:

图 125 单主/单从应用框图



MOSI、MISO、SCK 管脚都直接互连。数据从主机通过 MOSI 口线传输给从机,从从机通过 MISO 口线传输给主机。时钟由主机产生,通过 SCK 口线传输给从机。在硬件模式下(SSM 清零),NSS 在主机模式下拉高,在从机模式下拉低;在软件模式下(SSM 置位),NSS 管脚不用。

主设备通过时钟信号控制通信。当主设备通过 MOSI 管脚发送数据给从设备时,将同时通过 SCK 管脚发送时钟给从设备。从设备按照时钟通过 MOSI 管脚接收数据且通过 MISO 管脚发送数据。

根据 SPI_CTRL1 寄存器中 SSM 位的设置,NSS 管脚可以配置为软件模式或者硬件模式。

■ 硬件 NSS 模式 (SSM = 0)

取决于 SPI_CTRL1 寄存器中 SSOE 位的设置,NSS 管脚可以用作输入或者输出(仅在主机模式下)。

- NSS 输出使能 (SSM = 0, SSOE = 1)

此配置仅适用于主机模式。当主机开始通信(开始发送时钟信号)时 NSS 信号拉低且保持低电平,直到 SPI 被禁用为止。

- NSS 输出禁用 ($SSM = 0, SSOE = 0$)
NSS 管脚用作输入脚。在主机模式下, NSS 管脚应当拉高。在从机模式下, NSS 管脚作为片选功能, 当 NSS 拉低时从机被选中, 当 NSS 拉高时从机被释放。

■ 软件 NSS 模式 ($SSM = 1$)

SPI_CTRL1 寄存器的 SSI 位替代了 NSS 管脚的功能, 在主机模式下, SSI 位应当置位。在从机模式下, 若 SSI 被清零则从机被选中, 若 SSI 置位则从机被释放。此时 NSS 管脚并不参与 SPI 通信过程。

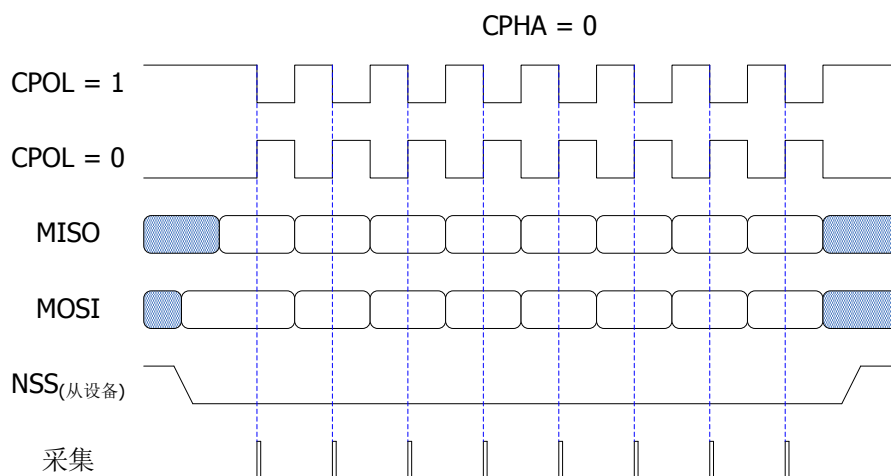
时钟相位与时钟极性

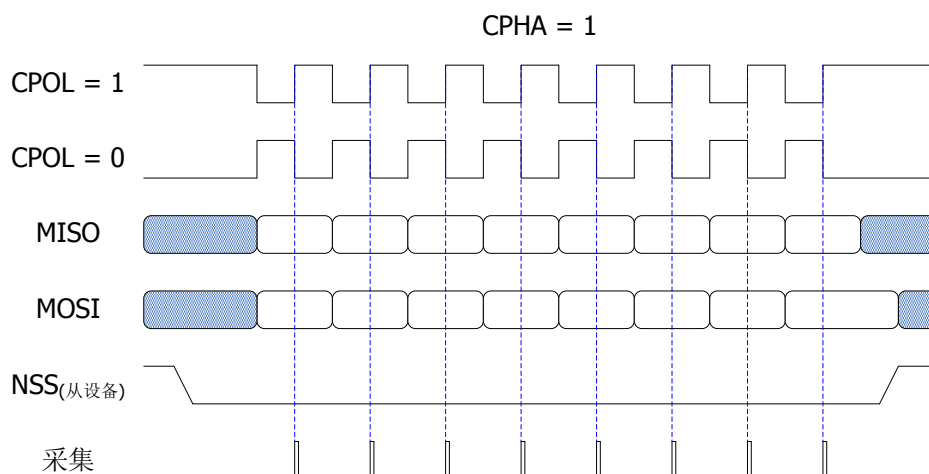
通过设置 SPI_CTRL1 寄存器中的 CPOL 和 CPHA 两个位, 可配置时钟信号的四种类型。CPOL 位用于配置时钟信号的静态状态(电平)。若 CPOL 为 0, SCK 管脚在空闲状态下为低电平; 若 CPOL 为 1, SCK 管脚在空闲状态下为高电平。

CPHA 位决定捕获数据的时序。若 CPHA 为 0, 则将在 SCK 信号的首个跳变沿采集首个数据位; 若 CPHA 为 1, 则将在 SCK 信号的第二个跳变沿采集首个数据位。

下图是四种时序关系下的 SPI 传输波形。

图 126 SPI 数据时钟时序图





数据帧格式

取决于 SPI_CTRL1 寄存器中 LSBFIRST 位的设置，数据在总线上逐位输出时可以高位在前，也可以低位在前。

当 LSBFIRST 为 0 时，数据的最高位先发送到总线上；当 LSBFIRST 为 1 时，数据的最低位先发送到总线上。取决于 SPI_CTRL1 寄存器中 DFF 位的设置，数据长度可以是 8 位或 16 位。

3.2 SPI 从机模式

在从机模式下，串行时钟接收自主设备，此时 SPI_CTRL1 寄存器的 BR[2:0] 位域无意义。通信是由主设备发起并掌控的，从设备必须在主设备发送时钟信号之前使能。

SPI 从机模式的配置步骤如下：

1. 配置数据格式（SPI_CTRL1 寄存器的 DFF）；
2. 配置时序关系（SPI_CTRL1 寄存器的 CPOL 及 CPHA），必须与主设备的配置一致；
3. 配置帧格式（SPI_CTRL1 寄存器的 LSBFIRST），必须与主设备的配置一致；
4. 配置 NSS 模式（SPI_CTRL1 寄存器的 SSM）。在硬件模式下（SSM=0），通信期间 NSS 管脚必须被拉低。在软件模式下（SSM=1），通信期间 SPI_CTRL1 寄存器的 SSI 必须被清零。
5. 设置为从机模式（将 MSTR 清零）；
6. 使能 SPI（将 SPE 置位）。

发送序列

当从设备从 SCK 管脚接收到时钟信号时，从设备开始发送：首先将最低位或最高位发送到 MOSI 管脚上，剩余各位将从发送缓冲加载到移位寄存器中。TXE 将会置位，软件根据需要可向发送缓冲写入下一个数据。硬件将根据收到的时钟信号发送移位寄存器中的剩余各位。

接收序列

在最后一个采样时钟沿后，数据传输完成，移位寄存器中的数据将复制到接收缓冲中，SPI_STR 寄存器的 RXNE 置位。此时读取 SPI_DTR 寄存器将返回接收缓冲中的数据，并且这个读操作将使得 RXNE 清零。

3.3 SPI 主机模式

在主机模式下，串行时钟由 SPI 模块产生，并发送到 SCK 管脚上。

在主机模式下，串行时钟是根据 PCLK（对 SPI1 模块而言取自 PCLK2；对 SPI2 模块而言则取自 PCLK1）产生的，BR[2:0]位域将决定波特率。

SPI 主机模式的配置步骤如下：

1. 配置 BR[2:0]位域，定义波特率；
2. 配置数据格式（SPI_CTRL1 寄存器的 DFF）；
3. 配置时序关系（SPI_CTRL1 寄存器的 CPOL 及 CPHA），必须与从设备的配置一致；
4. 配置帧格式（SPI_CTRL1 寄存器的 LSBFIRST），必须与从设备的配置一致；
5. 配置 NSS 模式（SPI_CTRL1 寄存器的 SSM）。假如 NSS 管脚用作输入，在硬件模式下（SSM=0），NSS 管脚必须拉高电平。在软件模式下（SSM=1），SPI_CTRL1 寄存器的 SSI 必须置位。假如 NSS 管脚用作输出，SSOE 必须置位，当传输开始后 NSS 管脚将会拉低；
6. 设置为主机模式（将 MSTR 置位）；
7. 使能 SPI（将 SPE 置位）。

发送序列

当数据写入发送缓冲后，主设备开始发送，首先将最低位或最高位发送到 MOSI 管脚上，剩余各位将从发送缓冲加载到移位寄存器中。当数据加载完成后，TXE 置位。

接收序列

在最后一个采样时钟沿后，数据传输完成，移位寄存器中的数据将复制到接收缓冲中，SPI_STR 寄存器的 RXNE 置位。此时读取 SPI_DTR 寄存器将返回接收缓冲中的数据，并且这个读操作将使得 RXNE 清零。假如发送缓冲中有数据，则该数据可以被连续发送。当 TXE 置位后，即可写入 DTR 寄存器。

3.4 SPI 单工通信

SPI 模块可以工作在单工模式下，并且有以下两种配置：

1. 将 SPI_CTRL1 寄存器的 BIDIMODE 置位。时钟信号将从主设备的 SCK 管脚发送到从设备的 SCK 管脚，数据信号将在主设备的 MOSI 管脚与从设备的 MISO 管脚之间传输，其方向是由 SPI_CTRL1 寄存器的 BIDIOE 位决定的，主设备以及从设备的 BIDIOE 必须相反。假如主设

备的 **BIDIOE** 为 1、从设备的 **BIDIOE** 为 0，数据将从主设备发往从设备；假如主设备的 **BIDIOE** 为 0、从设备的 **BIDIOE** 为 1，数据将从从设备发往主设备。

2. 将 **SPI_CTRL1** 寄存器的 **BIDIMODE** 清零。此时主设备以及从设备的 **RXONLY** 位必须相反。假如主设备的 **RXONLY** 为 0、从设备的 **RXONLY** 为 1，则数据从主设备发往从设备，主设备的 **MOSI** 管脚输出数据，从设备的 **MOSI** 管脚接收数据，**MISO** 管脚未用。假如主设备的 **RXONLY** 为 1、从设备的 **RXONLY** 为 0，则数据从从设备发往主设备，从设备的 **MISO** 管脚发送数据，主设备的 **MISO** 管脚接收数据，**MOSI** 管脚未用。时钟信号始终由主设备产生，通过 **SCK** 管脚发送给从设备。

在单工通信中，假如主设备配置为只收模式、从设备配置为只发模式，那么主机在 **SPI** 被使能后将立即开始接收数据。从设备必须在主设备使能 **SPI** 之前已经准备好发送，并且应当在限定时间内将数据写入发送缓冲。主设备收到倒数第二个数据之后应禁用 **SPI**（将 **SPE** 清零）；此时最后一个数据正在传输中，**SPI** 模块会在接收完最后一个数据后自动禁用。

3.5 数据接收及发送流程

每个 **SPI** 模块都有两个数据缓冲，分别是发送缓冲和接收缓冲。对 **SPI_DTR** 寄存器执行写操作，即可将数据存入发送缓冲中；对 **SPI_DTR** 寄存器执行读操作，即可将数据从接收缓冲取出。

当前一个数据发送完成后，发送缓冲中的数据将被复制到移位寄存器中，并且硬件会将 **TXE** 置位，之后如果需要的话，软件便可以写 **SPI_DTR** 寄存器将下一个数据填入发送缓冲。假如 **SPI_CTRL2** 寄存器的 **TXEIE** 置位，那么 **TXE** 置位时会产生中断。对 **SPI_DTR** 寄存器写入后可清零 **TXE** 位。如果 **TXE** 已经为 0，那么再对 **SPI_DTR** 寄存器写入数据将会使得发送缓冲中的数据被覆盖。

当数据的最后一位在采样时钟沿上被采集完成后，数据将接收并存入移位寄存器中，之后硬件会将数据从移位寄存器复制到接收缓冲中，并将 **RXNE** 置位。此时数据即准备好被软件读取。若 **SPI_CTRL2** 寄存器的 **RXNEIE** 置位，则当 **RXNE** 置位时会产生中断。读取 **SPI_DTR** 寄存器即可清零 **RXNE** 位。假如 **RXNE** 已经为 1（接收缓冲中的有效数据尚未被读出），又接收到了新的数据，那么 **OVR** 将会置位，表明发生了过载错误。

3.6 CRC 校验

CRC 校验能够提高通信的可靠性。**SPI** 模块中包含两个 **CRC** 计算单元，分别用于发送数据和接收数据。**CRC** 计算单元能够逐位计算 **CRC** 值。计算所用到的多项式是可配置的，保存在 **SPI_CRCPR** 寄存器中。

将 **SPI_CTRL1** 寄存器的 **CRCEN** 置位，即可使能 **CRC** 校验功能。在全双工模式或只发模式下，软件应当在最后一个数据写入 **SPI_DTR** 寄存器后立即将 **CRCNEXT** 置位，这样在发送完最后一个数据后还会将 **SPI_TCR** 寄存器的内容发送出去。假如数据通过 **DMA** 传输，那么 **SPI_TCR** 会自动由硬件发送，无需操作 **CRCNEXT** 位。在 **SPI_TCR** 发送期间，接收到的数据视为接收数据的 **CRC** 值，计算单元将关闭，数据将与 **SPI_RCR** 进行比对，假如二者不符，**CRCERR** 将会置位。

在只收模式中，软件应当在收到倒数第二个数据后（此时正在接收最后一个数据）写 **CRCNEXT**

位，那么在最后一个数据之后将会接收数据的 CRC 值，并与 SPI_RCR 的内容进行比较，假如二者不符，CRCERR 将会置位。

仅当发送缓冲为空的时候，CRC 值才会发送。CRC 发送期间，CRC 计算单元关闭。

采用 CRC 的 SPI 通信配置步骤如下：

1. 配置 CPOL、CPHA、LSBFIRST、BR、SSM、SSI 以及 MSTR；
2. 配置 CRC 多项式（SPI_CPR）寄存器；
3. 使能 CRC 校验功能（将 SPI_CTRL1 寄存器的 CRCEN 置位）；
4. 使能 SPI（将 SPI_CTRL1 寄存器的 SPE 置位）；
5. 开始通信；
6. 在全双工模式或只发模式下，当将最后一个数据写入 SPI_DTR 后应立即将 CRCNEXT 置位。在只收模式下，应在接收完倒数第二个数据之后将 CRCNEXT 置位。CRC 校验功能会在传输 CRC 值期间自动关闭；
7. SPI 传输完最后一个数据字节后将传输 CRC 值。收到的 CRC 值将于 SPI_RCR 的内容进行比较，若二者不相符则将 CRCERR 置位。

假如在从机模式下将 CRCEN 置位，那么即使 NSS 管脚已经拉高，CRC 计算单元也将继续工作。主机侧和从设备侧的 CRC 值必须及时清除，以确保主设备和从设备的 CRC 计算能重新同步。当 SPE 或 CRCEN 清零后，CRC 值（SPI_RCR 和 SPI_TCR）将被清除。

3.7 状态标志及错误标志

状态标志

■ 发送缓冲空标志（TXE）

当发送缓冲为空时，此标志将会置位，此时软件可以写 SPI_DTR 寄存器将下一数据送入发送缓冲。假如 TXEIE 为 1，那么当 TXE 置位的时候会产生中断。对 SPI_DTR 寄存器写入后，TXE 会自动清零。

■ 接收缓冲非空标志（RXNE）

当接收缓冲有内容时（即移位寄存器接收完成一个数据且复制到接收缓冲中），此标志将会置位，此时软件可以通过读取 SPI_DTR 寄存器将该数据取出。假如 RXNEIE 为 1，那么当 RXNE 置位的时候会产生中断。对 SPI_DTR 寄存器读取后，RXNE 会自动清零。

■ SPI 忙标志（BSY）

BSY 标志是由硬件置位及清零的。该标志反映了 SPI 通信层的工作状态。如果软件想要禁用 SPI，那么可通过检测 BSY 标志判断传输是否结束，只有这样才能避免破坏最后一个数据的传输。因此，软件必须严格按照下述步骤操作。

除了主设备的双向只收模式（MSTR = 1，BDM = 1，BDOE = 0）外，每当传输开始后 BSY 标志即会置位。当 SPI 被禁用或者产生主机模式故障（MODF = 1）后，BSY 标志被清零。当通信不连续时，在每个通信之间 BSY 为 0。当通信连续时，在从机模式下，每个通信之间

BSY 标志为 0；在主机模式下，BSY 标志在所有传输期间始终保持为 1。

错误标志

■ 主机模式故障 (MODF)

在 NSS 主机模式下，若 SSOE 禁用，则当 NSS 管脚拉低时 MODF 标志会置位。在 NSS 软件模式下，当 SSI 为 0 时 MODF 标志会置位。当 MODF 标志置位时，若 ERRIE 为 1，那么就会产生中断；SPE 和 MSTR 将被硬件清零，SPI 被禁用，设备强制返回从机模式。

MODF 位需通过下面的软件序列清零：

1. 读或写 SPI_STR 寄存器；
2. 写 SPI_CTRL1 寄存器。

在 MODF 清零之前，SPE 和 MSTR 位是写保护的。从设备的 MODF 标志不可置位。在多主配置中，工作在从机模式下的设备中 MODF 可能会置位，这意味着可能有多个主机竞争系统控制权并产生了冲突。

■ 过载 (OVR)

当 RXNE 已经置位后，假如又收到了数据，OVR 标志将会置位。这意味着前一个数据尚未被取走就来了新的数据。接收缓冲区中的旧数据将被新数据覆盖，导致前一数据丢失。当 OVR 置位时，若 ERRIE 为 1，就会产生中断。

OVR 位需通过下面的软件序列清零：先读 SPI_DTR 寄存器；然后读 SPI_STR 寄存器。

■ CRC 校验错误 (CRCERR)

当 CRCEN 为 1 时，对接收数据的 CRC 计算结果保存在 SPI_RCR 寄存器中，并且在接收完成后与接收的最后一个数据 (CRC 值) 进行比较，假如二者不相符，则 CRCERR 将会置位。对 CRCERR 写 0 即可将其清零，写 1 无效果。

3.8 禁用 SPI

传输结束后，软件将 SPE 清零以停止 SPI 模块的工作。在某些配置中，有可能当 SPE 清零后最后一个传输过程仍在进行中。为了避免破坏最后一次传输，软件应严格按照下述步骤操作：

全双工模式：

1. 等待 RXNE 从 1 变为 0，接收最后一个数据；
2. 等待直到 TXE=1；
3. 等待直到 BSY=0；
4. 禁用 SPI (SPE = 0)，进入停机模式或关闭外设时钟。

只发模式

1. 将最后一个数据写入 SPI_DTR 寄存器；
2. 等待直到 TXE=1；
3. 等待直到 BSY=0；

- 禁用 SPI ($SPE = 0$), 进入停机模式或关闭外设时钟。

主机只收模式

- 等待倒数第二个 $RXNE=1$;
- 在禁用 SPI 之前 ($SPE=0$) 等待一个 SPI 时钟周期 (通过软件延时);
- 等待最后一个 $RXNE=1$, 之后才能进入停机模式或关闭外设时钟。

从机只收模式

- SPI 随时可以禁用 ($SPE=1$), 不过 SPI 硬件会在当前传输结束之后才真正停止工作;
- 等待 $BSY=0$, 之后才能进入停机模式或关闭外设时钟。

3.9 DMA 请求

采用 DMA 来发送或接收数据, 才能真正发挥出 SPI 的最高速度。此时读写 SPI_DTR 的速度足够快, 发送的数据之间相当紧凑。

将 SPI_CTRL2 寄存器的 TXDMAEN 或 RXDMAEN 置位, 即可使能 DMA 访问。当 TXE 置位时, 将向 DMA 的发送通道产生一个 DMA 请求, 待 DMA 向 SPI_DTR 寄存器写入数据后 TXE 即清零。当 RXNE 置位时, 将向 DMA 的接收通道产生一个 DMA 请求, 待 DMA 从 SPI_DTR 寄存器读出数据后 RXNE 即清零。

当 SPI 仅发送数据时, 可以只使能 SPI Tx DMA 通道。此时 OVR 标志可能会长期置位, 因为接收缓冲中的数据被反复覆盖、未被取出。

当 SPI 仅接收数据时, 可以只使能 SPI Rx DMA 通道。

在发送模式下, 当 DMA 将所有待发送的数据都写入后 (DMAISR 寄存器的 TCIF 置位), 可通过监视 BSY 标志来判断 SPI 通信是否全部完成。这样做是为了在禁用 SPI 或进入停机模式之前避免破坏最后一次传输。软件必须先等到 $TXE=1$, 之后继续等到 $BSY=0$ 。

图 127 通过 DMA 发送

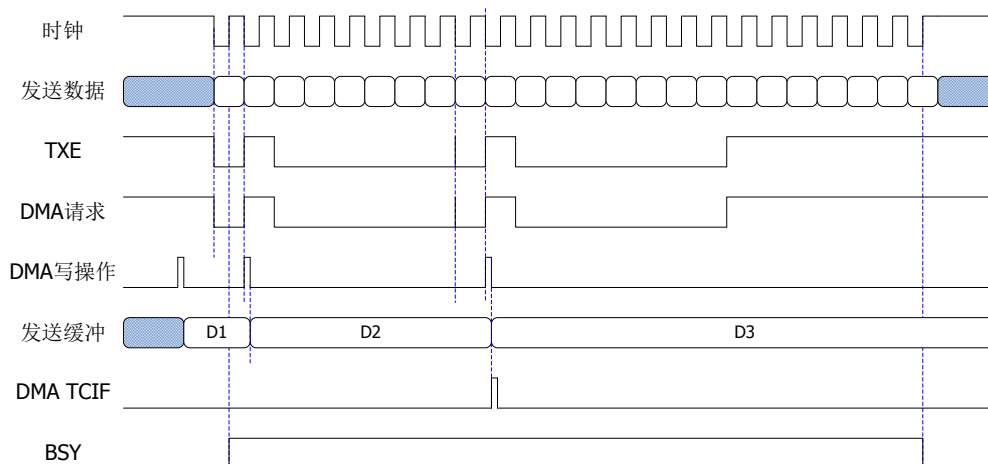
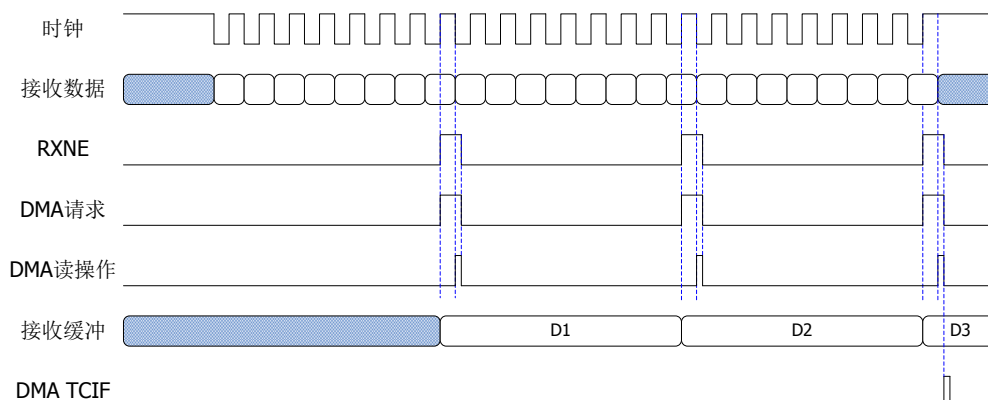


图 128 通过 DMA 接收



当启用 SPI 通信的 DMA 操作之后，若 CRCEN 为 1，则 SPI 模块会在最后一个数据之后自动发送和接收 CRC 值。此时 CRCNEXT 控制位将失去意义。接收缓冲中的 CRC 值应当及时读出，以便清零 RXNE 位。

3.10 SPI 中断

表 47 SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲空	TXE	TXEIE
接收缓冲非空	RXNE	RXNEIE
主机模式故障	MODF	ERRIE
过载错误	OVR	
CRC 校验错误	CRCERR	

4 SPI 寄存器

4.1 SPI 控制寄存器 1 (SPI_CTRL1)

地址偏移: 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRCEN	CRC NEXT	DFE	RX ONLY	SSM	SSI	LSB FIRST	SPE		BR[2:0]		MSTR	CPOL	CPHA
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域 名称 描述

位/位域	名称	描述
15	BIDIMODE	双向传输使能 0: 2 线各自单向传输模式 1: 1 线双向传输模式。信息在主机的 MOSI 脚和从设备的 MISO 脚之间传输
14	BIDIOE	双向传输的输出使能 当 BIDIMODE 置位时，本位决定数据传输的方向。 0: 工作于只收模式 1: 工作于只发模式
13	CRCEN	CRC 计算单元使能 0: 禁用 CRC 计算单元 1: 使能 CRC 计算单元
12	CRCNEXT	下次传输 CRC 计算结果 0: 下次传输数据 1: 下次传输 CRC 值 (TXCRCR) 当传输是由 DMA 控制时，CRC 值将会由硬件完成传输。此时本位应当清零。 在全双工模式或只发模式下，当最后一个数据写入 SPI_DTR 寄存器后应将本位置位。在只收模式下，应当在接收完倒数第二个数据后将本位置位。
11	DFF	数据帧格式 0: 8 位数据帧 1: 16 位数据帧
10	RXONLY	只收模式 当 BIDIMODE 为 0 时，本位决定传输的方向。 0: 全双工 1: 只收模式
9	SSM	NSS 软件模式选择 0: NSS 硬件模式。NSS 管脚输入取决于 I/O 状态 1: NSS 软件模式。NSS 管脚输入取决于 SSI 位
8	SSI	NSS 软件模式下的管脚选择 0: NSS 管脚拉低 1: NSS 管脚拉高 当 SSM 置位时，本位才体现出作用。
7	LSBFIRST	帧格式 0: 高位在前 1: 低位在前
6	SPE	SPI 使能 0: 禁用 SPI 外设 1: 使能 SPI 外设
5:3	BR	主设备时钟预分频配置 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/256 对于 SPI1 模块而言 PCLK 指 PCLK2；对于 SPI2 模块而言 PCLK 指 PCLK1
2	MSTR	主从模式选择 0: 从机模式 1: 主机模式

位/位域	名称	描述
1	CPOL	时钟极性选择 0: 当 SPI 空闲时, CLK 管脚拉低 1: 当 SPI 空闲时, CLK 管脚拉高
0	CPHA	时钟相位选择 0: 在第一个时钟跳变沿采集第一个数据 1: 在第二个时钟跳变沿采集第一个数据

4.2 SPI 控制寄存器 2 (SPI_CTRL2)

地址偏移: 0x04

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				保留				TXEIE	RXNEIE	ERRIE	保留		SSOE	TXDMAEN	RXDMAEN
								RW	RW	RW			RW	RW	RW

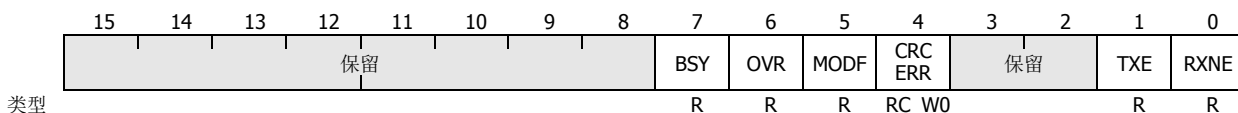
类型

位/位域	名称	描述
15:8	保留	请保持其复位状态
7	TXEIE	发送缓冲空中断 0: 禁用 TXE 中断 1: 使能 TXE 中断, 当 TXE 置位时将会产生中断
6	RXNEIE	接收缓冲非空中断 0: 禁用 RXNE 中断 1: 使能 RXNE 中断, 当 RXNE 置位时将会产生中断
5	ERRIE	SPI 错误中断 0: 禁用错误中断 1: 使能错误中断, 当 CRCERR 或 MODF 或 OVR 置位时将会产生中断
4:3	保留	请保持其复位状态
2	SSOE	SS 输出使能 0: 禁用 SS 输出 1: 假如 NSS 管脚配置为输出则使能 SS 输出。在主机模式下, 当 SPI 使能后, NSS 管脚将会拉低 假如 NSS 管脚配置为输入, 那么在主机模式下应将 NSS 管脚拉高。此时本位无实际意义。
1	TXDMAEN	发送缓冲 DMA 使能 0: 禁用发送缓冲 DMA 1: 使能发送缓冲 DMA。当 SPI_STR 寄存器的 TXE 置位时, 将会在相应的 DMA 通道产生请求。
0	RXDMAEN	接收缓冲 DMA 使能 0: 禁用接收缓冲 DMA 1: 使能接收缓冲 DMA。当 SPI_STR 寄存器的 RXNE 置位时, 将会在相应的 DMA 通道产生请求。

4.3 SPI 状态寄存器 (SPI_STR)

地址偏移: 0x08

复位值 : 0x0002



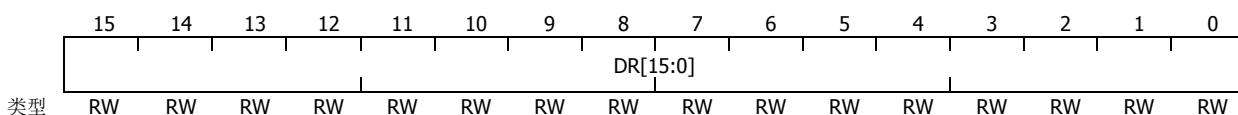
类型

位/位域	名称	描述
15:8	保留	请保持其复位状态
7	BSY	SPI 忙标志 0: SPI 空闲 1: SPI 正在收发数据, 或发送缓冲非空
6	OVR	SPI 过载标志 0: 接收缓冲为空, 或未接收到任何数据 1: 接收缓冲非空的情况下又收到了新的数据。 本位需通过特定的软件操作序列清零。
5	MODF	SPI 模式错误标志 0: 无模式错误 1: 在主机模式下, 若 NSS 由硬件控制, 则表明 NSS 管脚被外部信号拉低; 若 NSS 由软件控制, 则表明 SSI 位被清零。 本位需通过特定的软件操作序列清零。
4	CRCERR	SPI CRC 校验错误标志 0: SPI_RCR 的值与最后收到的 CRC 数据一致 1: SPI_RCR 的值与最后收到的 CRC 数据不符
3:2	保留	请保持其复位状态
1	TXE	SPI 发送缓冲空标志 0: 发送缓冲非空 1: 发送缓冲空
0	RXNE	SPI 接收缓冲非空标志 0: 接收缓冲空 1: 接收缓冲非空

4.4 SPI 数据寄存器 (SPI_DTR)

地址偏移: 0x0C

复位值 : 0x0000



类型

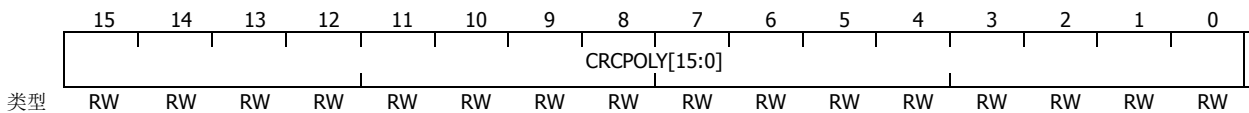
位/位域	名称	描述
------	----	----

位/位域	名称	描述
15:0	DR[15:0]	数据寄存器 每个 SPI 模块有两个缓冲，分别为发送缓冲和接收缓冲。对 DR 写操作，可将数据写入发送缓冲；对 DR 读操作，可将数据从接收缓冲取出。 当数据帧格式为 8 位时，DR[15:8]始终为 0，DR[7:0]用于发送和接收，此时发送缓冲和接收缓冲都是 8 位的。当数据帧格式为 16 位时，DR[15:0]用于发送和接收，发送缓冲和接收缓冲都是 16 位的。

4.5 SPI CRC 多项式寄存器 (SPI_CPR)

地址偏移: 0x10

复位值 : 0x0007

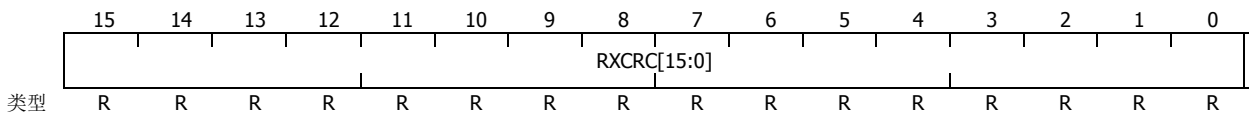


位/位域	名称	描述
15:0	CRCPOLY[15:0]	本寄存器包含用于计算 CRC 的多项式。默认值为 0x07。

4.6 SPI 接收 CRC 寄存器 (SPI_RCR)

地址偏移: 0x14

复位值 : 0x0000

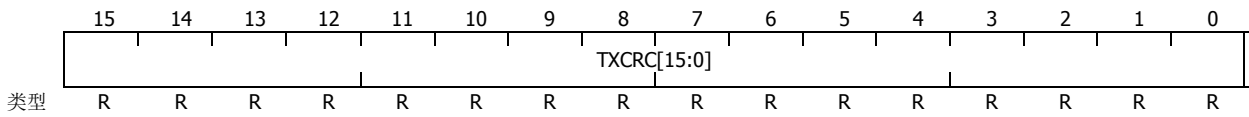


位/位域	名称	描述
15:0	RXCRC	当 SPI_CTRL1 寄存器的 CRCEN 置位时，硬件将会对接收的数据计算 CRC 值，并将其保存在 RXCRCR 寄存器中。假如数据帧格式为 8 位，CRC 计算将采用 CRC8 标准，并且将结果保存在 RXCRC[7:0]中；假如数据帧格式为 16 位，CRC 计算将采用 CRC16 标准，并且将结果保存在 RXCRC[15:0]中。 硬件在收到每个位后都会计算 CRC。当 BSY 置位时，读取本寄存器有可能返回的是计算过程中的值。 当 SPI_CTRL1 寄存器的 CRCEN 清零或 SPIEN 清零后，本寄存器将复位。

4.7 SPI 发送 CRC 寄存器 (SPI_TCR)

地址偏移: 0x18

复位值 : 0x0000



类型

位/位域	名称	描述
15:0	TXCRC	<p>当 SPI_CTRL1 寄存器的 CRCEN 置位时，硬件将会对发送的数据计算 CRC 值，并将其保存在 TXCRC 寄存器中。假如数据帧格式为 8 位，CRC 计算将采用 CRC8 标准，并且将结果保存在 TXCRC[7:0] 中；假如数据帧格式为 16 位，CRC 计算将采用 CRC16 标准，并且将结果保存在 TXCRC[15:0] 中。</p> <p>硬件在发送每个位后都会计算 CRC 值。当 BSY 置位时，读取本寄存器有可能返回的是计算过程中的值。此外请注意，帧格式不同（取决于 SPI_CTRL1 寄存器中 LSBFIRST 位的设置），CRC 计算结果也会不同。</p> <p>当 SPI_CTRL1 寄存器的 CRCEN 清零或 SPIEN 清零后，本寄存器将复位。</p>

第13章 通用同步/异步收发器（USART）

1 简介

通用同步/异步收发器（USART）提供了一个灵活方便的全双工数据交换接口，支持采用同步或异步传输方式。USART 可用于并转串的数据传输，也常用于 RS232 标准通信。USART 模块提供了可编程的波特率发生器，可对系统时钟进行高精度分频，产生 USART 收发器所需的特定频率。通过读取状态寄存器（USART_SR），可获取各种运行状态。这些状态包括传输的类型和状况，以及多种错误状况。USART 还支持 DMA 功能，可实现高速率的数据通信。

2 主要特点

- 三路完全可编程的 USART
- 支持全双工异步通信
- 支持半双工单线通信
- NRZ 标准格式（MARK/SPACE）
- 可编程的波特率发生器，最高可达 4.5 Mbit/s
- 完全可编程的串口特性
 - 数据字（8 位或 9 位）低位在前
 - 偶校验位、奇校验位、无校验位的生成/检测
 - 产生 0.5、1、1.5、2 个停止位
- LIN 暂停帧的生成和检测
- 发送方可输出时钟信号，实现同步传输
- 支持 IrDA
- 兼容 ISO 7816-3 的智能卡接口
- 支持调制解调器操作（CTS/RTS）
- 借助集中式 DMA，可实现可配置的多级缓冲通信
- 发送器和接收器可分别使能
- 传输检测标志：
 - 接收缓冲满
 - 发送缓冲空
 - 发送结束标志
- 校验位控制：
 - 发送奇偶校验位
 - 收到数据字节的奇偶校验
- 错误检测：过载、噪声、帧格式错误、奇偶校验错误

- 10 个带标志的中断源：
 - CTS 变化
 - LIN 暂停帧检测
 - 发送数据寄存器空
 - 发送完成
 - 接收数据寄存器满
 - 线路空闲
 - 过载错误
 - 帧格式错误
 - 噪声错误
 - 奇偶校验错误
- 支持多机通信
 - 假如地址不匹配，始终保持静默模式
 - 通过线路空闲或者地址掩码检测从静默模式唤醒

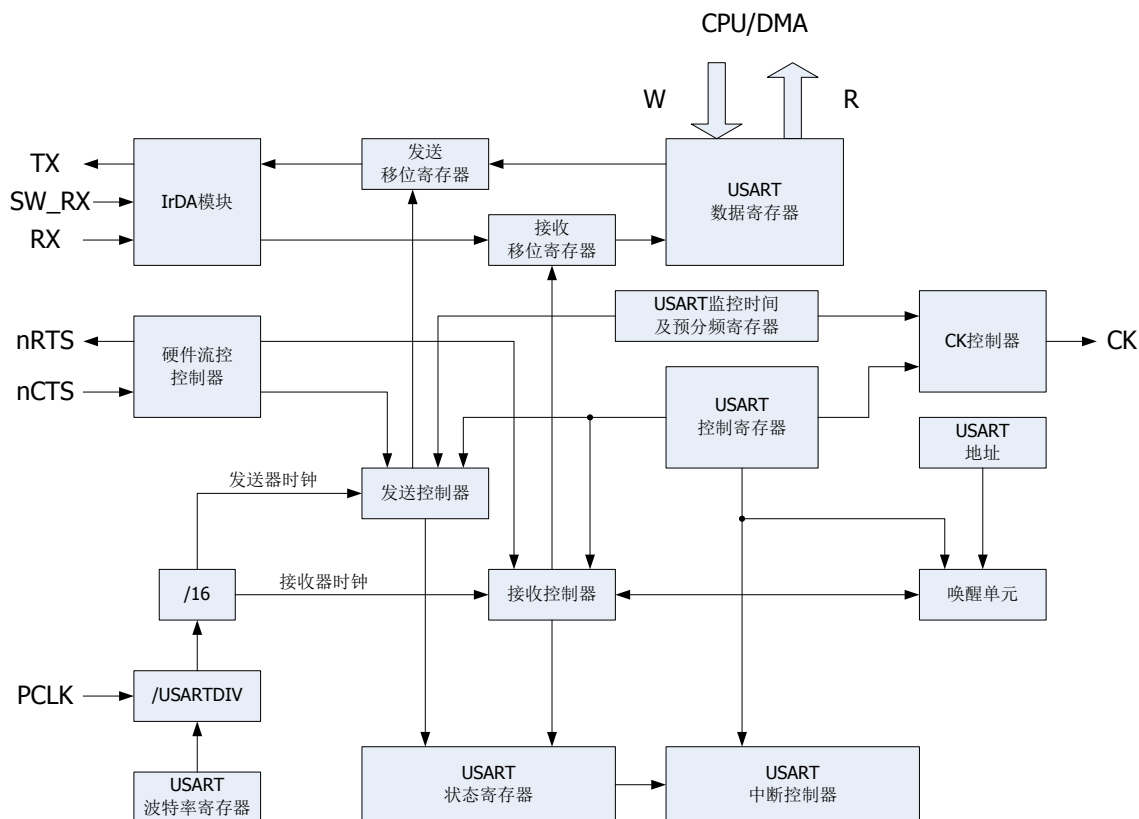
3 功能描述

USART 接口通过以下主要管脚从外部连接到其它设备：

表 48 USART 重要管脚描述

管脚名称	电气类型	描述
RX	输入	接收数据
TX	输出 I/O (单线模式/智能卡模式)	发送数据。当 USART 使能后，若无数据发送，默认为高电平。
CK	输出	串行时钟，用于同步通信
nCTS	输入	硬件流控模式的 CTS (Clear to Send, 通知发送信号)
nRTS	输出	硬件流控模式的 RTS (Request to Send, 请求发送信号)

图 129 USART 模块内部框图



3.1 USART 发送器

当 USART_CTRL1 寄存器的发送使能位（TE）置位后，波特率发生器将产生发送器时钟脉冲，并通过 CK 管脚输出。之后串行位码流将按照控制寄存器中的设置，由发送器在一个空闲帧后开始发送。

若发送被破坏，只要发送仍在进行，便不得禁用 TE 位。

假如数据可以在不覆盖前一个数据的情况下写入 USART_DR 寄存器，那么 TXE 标志将会处于置位状态。当数据写入 USART_DR 寄存器后，TXE 标志将会清零。

假如一帧数据已经发送出去，并且 TXE 标志置位，那么 TC 标志将会置 1。假如 USART_CTRL1 寄存器中相应的中断使能位（TCIE）为 1，就会产生中断。

USART 发送应按照下列步骤进行：

1. 将 USART_CTRL1 寄存器的 UE 置位，使能 USART；
2. 将 USART_CTRL1 寄存器的 WL 写 1，设置数据位长度；
3. 在 USART_CTRL2 寄存器中设置停止位长度；
4. 假如选择了多级缓冲通信方式，应在 USART_CTRL3 寄存器中使能 DMA（DET 位）；
5. 在 USART_BRR 寄存器设置波特率；

6. 将 USART_CTRL1 寄存器的 TE 置位；
7. 将数据写入 USART_DR 寄存器；
8. 等到 TC=1，发送完成。

在禁用 USART 或令其进入低功耗模式之前，必须等待 TC 置位。

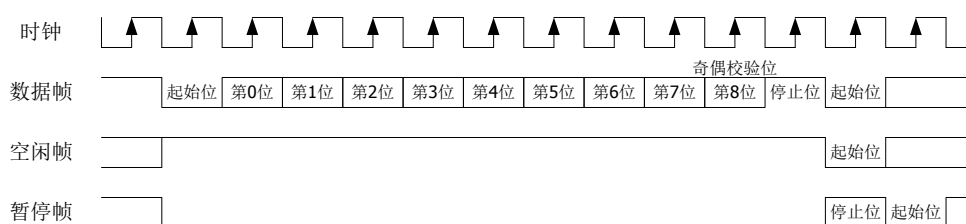
先读 USART_SR 再写 USART_DR 可将 TC 清零。在多级缓冲通信方式下，直接向 TC 写 0 也能清零 TC。

当 SDBK 标志置位时，将会发送暂停帧。发送完成后，SDBK 标志将会清零。

表 49 停止位配置

停止位长度 (位时间)	描述
0.5	智能卡模式（接收数据）
1	默认值
1.5	智能卡模式
2	标准 USART、单线以及调制解调器模式

图 130 USART 字符帧（9 数据位+1 停止位）



如果开启了奇偶校验功能，则最高位被当作奇偶校验位。但在地址掩码检测时最高位已经用作地址索引了。

假如数据位（除奇偶校验位外）中“1”的个数是偶数个，那么奇偶校验位应当为 0（偶校验）或 1（奇校验）。

3.2 USART 接收器

当检测到有效的起始位脉冲后，接收器将开始接收位码流。在此期间还会进行过载检测、奇偶校验、帧格式错误检测、暂停检测。

假如一帧接收完成，且 RXNE 置位，那么就可以读取 USART_DR 寄存器以及 USART_SR 寄存器中的相关标志位。若 USART_CTRL1 寄存器中相应的中断使能位 (RXNEIE) 为 1，就会产生中断。

RXNE 必须在下一个数据到来之前及时清零，否则就会造成过载错误。为避免破坏接收数据，在接收过程尚未完成时不得禁用 RE。需要将 RXNE 清零时，可直接向其写 0，或者在多级缓冲通信方式下读取 USART_DR。多级缓冲通信方式下的 DMA 读操作也会将 RXNE 清零。

USART 接收应按照下列步骤进行：

1. 将 USART_CTRL1 寄存器的 UE 置位，使能 USART；
2. 将 USART_CTRL1 寄存器的 WL 写 1，设置数据位长度；
3. 在 USART_CTRL2 寄存器中设置停止位长度；
4. 假如选择了多级缓冲通信方式，在 USART_CTRL3 寄存器中使能 DMA（DER 位）；
5. 在 USART_BRR 寄存器设置波特率；
6. 将 USART_CTRL1 寄存器的 RE 置位；

3.3 接收错误

若 RXNE 已经置位的情况下又收到了一个新的数据，或者前一个 DMA 请求还未及时得到服务的情况下又收到了一个新的数据，将会产生过载错误。此时 USART_SR 寄存器中的 ORE 将置位。

USART 模块会按照内部波特率发生器时钟对 RX 线状态进行过采样（出于数据恢复目的）。若检测到噪声错误，USART_SR 寄存器中的 NE 将会在 RXNE 位的上升沿置位，且将从移位寄存器收到无效数据。

当在预期时间内未检测到有效的停止位，将会产生帧格式错误。若检测到帧格式错误，USART_SR 寄存器中的 FE 将会在 RXNE 位的上升沿置位，且将从移位寄存器收到无效数据。

以上三个错误标志位都可以通过先读 USART_SR 寄存器、再读 USART_DR 寄存器的操作清零。

3.4 波特率的发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。这两部分组合所得的数值将作为决定位周期的波特率发生器。由于具有小数部分的分频系数，USART 能够更准确地实现常用波特率。

波特率分频系数（USARTDIV）与系统时钟的关系式如下：

$$\text{USARTDIV} = \frac{\text{PCLK}}{16 \times \text{波特率}}$$

对于 USART2 和 3 而言，PCLK 指 PCLK1；对于 USART1 而言，PCLK 指 PCLK2。

USART 模块还会在内部产生参考时钟，其频率为波特率的 16 倍频。该参考时钟经 16 分频后供给发送器时钟，不分频直接供给接收器时钟以实现错误检测。

3.5 多机通信

在多机通信时，地址不符的接收方将进入静默模式。在接收过程中，指定的接收设备才会激活并接收完整的报文内容，而其它不相干的接收设备则处于静默模式。通过线路空闲检测以及地址掩码检测，可以使器件进入或退出静默模式。

对于线路空闲检测，当 USART_CTRL1 寄存器的 RMMC 置位时 USART 将进入静默模式。一旦检测到空闲帧，器件会立即退出静默模式。RMMC 将被清零，但是 USART_SR 中的 IDLEF 并不置

位。

在地址掩码检测中，最高位为 1 的字节被识别为寻址字节。寻址字节中的低 4 位即为会话目标接收设备的地址。接收设备将寻址字节与自己的地址进行比较，之后分别进入或退出静默模式。当 RXNE 为 0 时，RMMC 位是可写的。

3.6 LIN 模式

将 USART_CTRL2 寄存器的 LINEN 置位，即可使能本地互连网络(Local Interconnection Network, 简称为 LIN) 模式。在 LIN 模式下，USART_CTRL2 寄存器的 CKE 应当清零，USART_CTRL3 寄存器的 STPL、SCME、HDE、IRME 也应当清零。

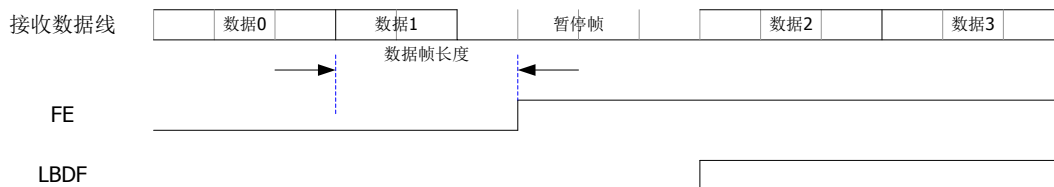
LIN 发送流程与标准的发送流程几乎相同。不过 LIN 模式下数据位长度只能为 8。暂停帧为连续的 13 个 ‘0’。

暂停检测部分是完全独立的，与标准 USART 接收器无关。因此在空闲状态下或在帧传输期间都能检测暂停。

使能接收器后，若检测到起始位，电路将开始采样下一个位。

在暂停检测期间，若遇到了帧格式错误，那么暂停检测将会取消，当 RX 线再次回到高电平后又会继续。之后重新开始检测起始位。假如在分隔符（高电平）之前检测到连续的 10/11（根据 USART_CTRL2 寄存器中 LBDL 位的设置）个 “0”，那么 USART_SR 寄存器中的 LIN 暂停标志位也会被置位。

图 131 LIN 模式下的帧格式错误检测以及暂停帧检测



3.7 半双工通信模式

将 USART_CTRL3 寄存器的 HDE 置位，即可使能半双工通信模式。在半双工通信模式下，USART_CTRL2 寄存器的 LINEN、CKE 应清零，USART_CTRL3 寄存器的 SCME、IRME 也应清零。

半双工模式下仅使用单线通信，TX 与 RX 管脚将从内部连接。TX 管脚应配置为 I/O 管脚。通信冲突应由软件负责处理。当 TE 标志置位时，数据寄存器中的数据将会发送到总线。

3.8 同步通信模式

USART 可用于全双工同步串行通信，不过仅支持主机模式。将 USART_CTRL2 寄存器的 CKE 置位，即可使能同步通信模式。在同步通信模式下，USART_CTRL2 寄存器的 LINEN、USART_CTRL3

寄存器的 SCME、HDE、IRME 都应当清零。CK 管脚是 USART 同步发送器的时钟输出，只有当 TE 只能后才会有效。在起始位和停止位期间，不会从 CK 脚输出时钟信号。USART_CTRL2 寄存器的 LBCL 用于决定在传输最高位（寻址标志）时是否输出时钟信号。在总线空闲状态、引导字以及暂停帧期间，也不会输出时钟信号。USART_CTRL2 寄存器的 CPHA 位用于决定数据是在第一个时钟沿捕获还是在第二个时钟沿捕获，CPOL 位用于决定 USART 同步模式空闲状态下的时钟线电平。

在发送器或接收器使能的情况下，不得更改这 3 个位（CPOL、CPHA、LBCL）。

时钟与已发送的数据保持同步。同步通信模式下的接收器按照发送方的时钟信号进行采样，不带任何过采样。

图 132 同步模式下的 USART 接线示例

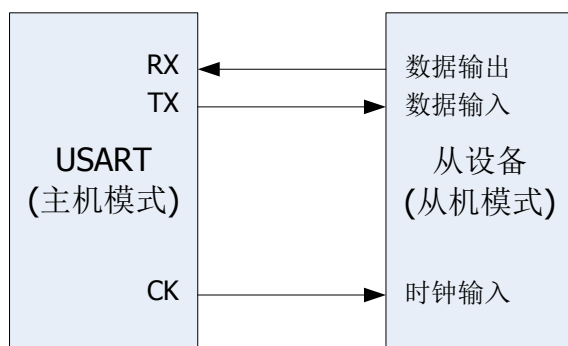
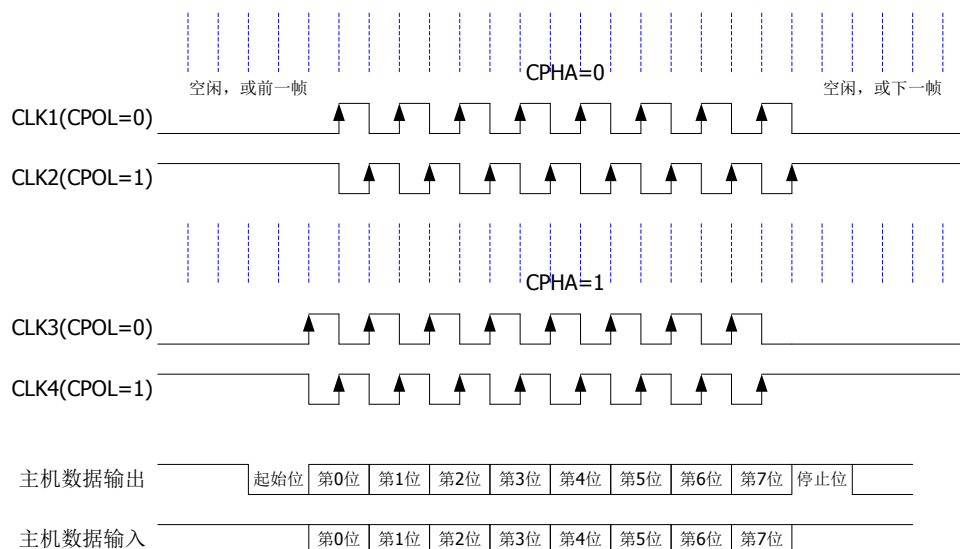


图 133 8 位格式的 USART 同步通信波形 (LBCL=1)



3.9 智能卡 (ISO7816) 模式

智能卡模式是一种异步通信模式。将 USART_CTRL3 寄存器的 SCME 置位,即可使能智能卡模式。在智能卡模式下, USART_CTRL2 寄存器的 LINEN、USART_CTRL3 寄存器的 HDE、IRME 都应当清零。

假如 CKE 置位,那么 USART 将向智能卡提供时钟。该时钟也可以分频用于其它用途。

智能卡模式下的帧格式为: 1 起始位 + 9 数据位 (包括 1 奇偶校验位) + 1.5 停止位。

智能卡模式是一种半双工的通信模式。产品设计时应将 SW_RX 管脚与 TX 管脚短接在一起。TX 管脚应配置为 I/O。

相较于标准工作方式下的通信,从发送移位寄存器到 TX 管脚将延迟半个波特率时钟,并且 TC 标志的产生将根据保护时间寄存器的设置延迟一定时间。

在 USART 接收期间,在接收完帧后如果检测到奇偶校验错误, TX 口线将拉低一个波特率时钟。这个信号相当于发送给智能卡的“NACK”(应答错误)。之后智能卡一侧就会产生帧格式错误。按照智能卡协议,智能卡将会重新发送该数据。若 USART_CTRL3 寄存器的 NACKE 置位,就会发送 NACK,并且 USART 也就不会把 NACK 信号误当做起始位。

空闲帧和暂停帧在智能卡模式下无用。

3.10 IrDA SIR ENDEC 模式

将 USART_CTRL3 寄存器的 IRME 置位,即可使能 IrDA 模式。在 IrDA 模式下, USART_CTRL2 寄存器的 LINEN、STPL、CKE、USART_CTRL3 寄存器的 HDE、SCME 都应当清零。

按照 IrDA SIR 物理层的定义,一个红外光脉冲 (RTZ 信号) 代表逻辑 0,其脉冲宽度应为 PSC 时钟周期 (由 USART_GTPR 的 PSC 位域设置) 的 2 倍,通常占一个位时间的 3/16。如果脉冲宽度小于 1 个 PSC 时钟周期,则 IrDA 无法检测到。如果脉冲宽度大于 1 个 PSC 时钟周期、小于 2 个 PSC 时钟周期,则 IrDA 不能可靠地检测到。

USART 数据帧将由 SIR 发送编码器进行调制。调制后的信号经由红外发射管进行发送。对于编码器而言,波特率应大于 115200。USART 部分经红外接收管收到调制信号后,输出解调后的数据帧。编码器所发送调制信号的极性,与解码器所接收的调制信号的极性相反。解码器输入信号通常为高电平,当检测到低电平则解码得到起始位。

在 IrDA SIR ENDEC 模块中不得同时发送和接收。

而对于低功耗模式:

- 发送器: 脉冲宽度可以为低功耗波特率的 3 倍
- 接收器: 与正常模式相同

图 134 IrDA SIR ENDEC 模块

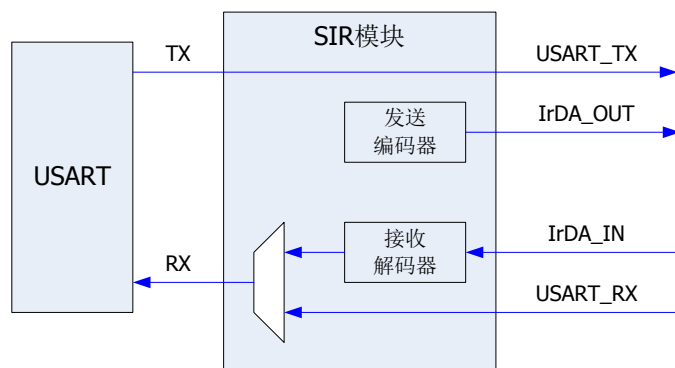
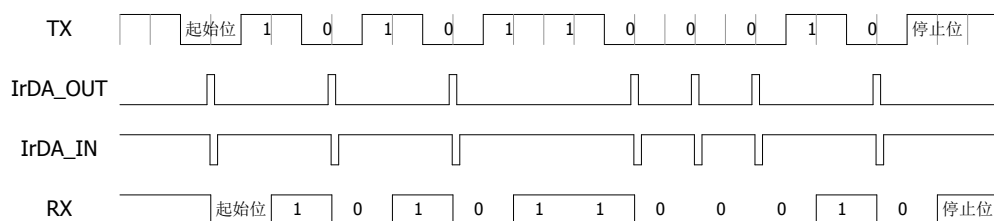


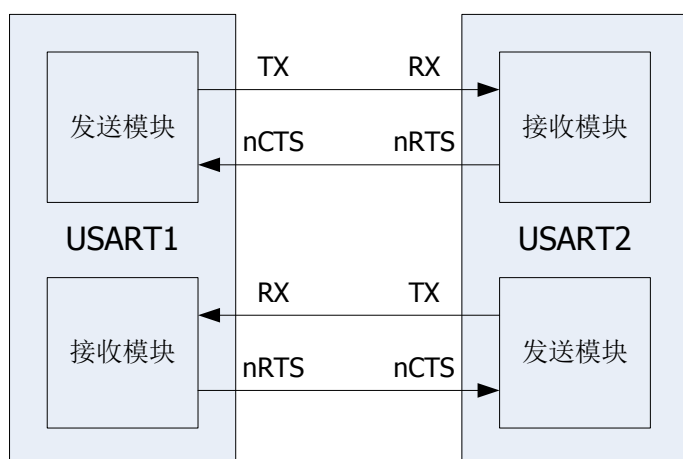
图 135 IrDA 数据调制



3.11 硬件流控

借助 nCTS 输入信号和 nRTS 输出信号，可对串行数据流进行额外的干预，这种工作方式称为硬件流控。将 USART_CTRL3 寄存器的 RTSE 位写 1，即可使能 RTS 流控；将 USART_CTRL3 寄存器的 CTSE 位写 1，即可使能 CTS 流控。

图 136 两个 USART 设备之间的硬件流控



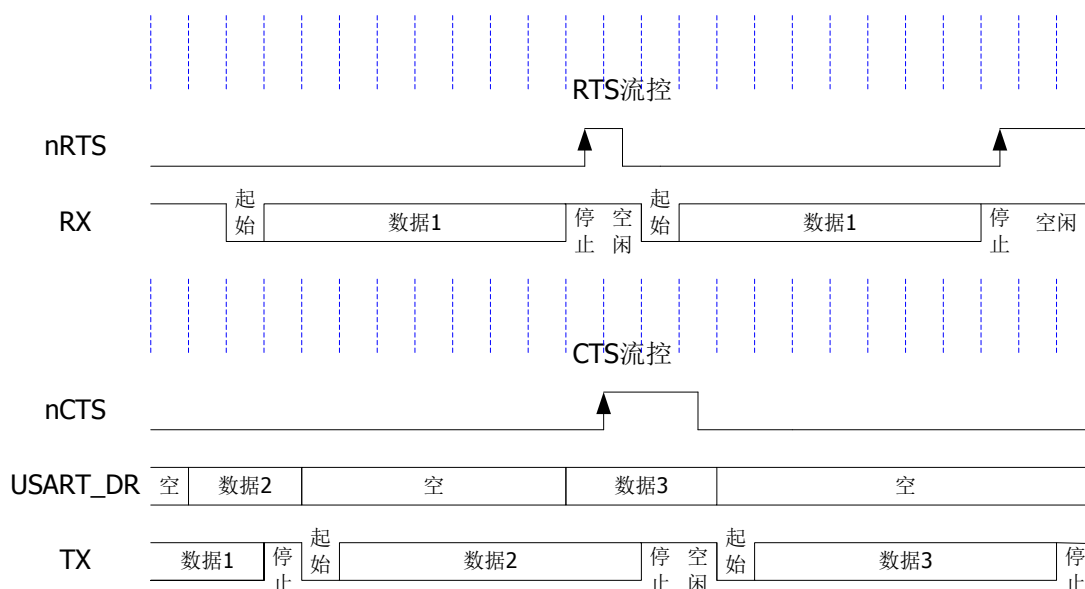
RTS 流控

仅当 nRTS 为低电平时，USART 接收模块才能够接收数据。接收期间该信号保持低电平，直到数据帧接收完成后才会拉高。当 nRTS 信号再次拉低后，表明接收模块可以进行下一次接收。当接收寄存器满的时候，该信号保持高电平。

CTS 流控

假如 USART_SR 寄存器中的 TXE 位为 0，并且 nCTS 信号为低电平，那么发送模块将发送数据帧。在发送期间，若 nCTS 信号拉高，那么发送方会在发完当前数据帧后停止发送。

图 137 硬件流控



3.12 DMA 请求

借助 DMA 可实现 USART 的连续通信。USART_CTRL3 寄存器的 DET 用于使能 DMA 发送，USART_CTRL3 寄存器的 DER 用于使能 DMA 接收。

DMA 发送配置步骤如下：

1. 配置 DMA 寄存器，包括目的地址（应填写 USART_DR 寄存器地址）、源地址（应填写存储器地址）、传输总字节数、通道优先级、DMA 中断等；
2. 对 USART_SR 寄存器的 TC 位写 0；
3. 对 DMA 控制寄存器的 EN 位写 1，使能 DMA 通道。

当 DMA_ISR 寄存器中的 TCIF 标志置位后，稍后 USART_SR 寄存器中的 TC 标志才会置位。该标志在数据传输期间始终保持为 0，当 USART 传输结束后置位。当 TC 并未置位时，进入停机模式或禁用 USART 都会导致传输被破坏。

DMA 接收的配置：配置 DMA 寄存器，包括目的地址（应填写 USART_DR 寄存器地址）、源地址（应填写存储器地址）、传输总字节数、通道优先级、DMA 中断等。

只要收到了数据，就会触发 RXNE 事件。

3.13 USART 中断

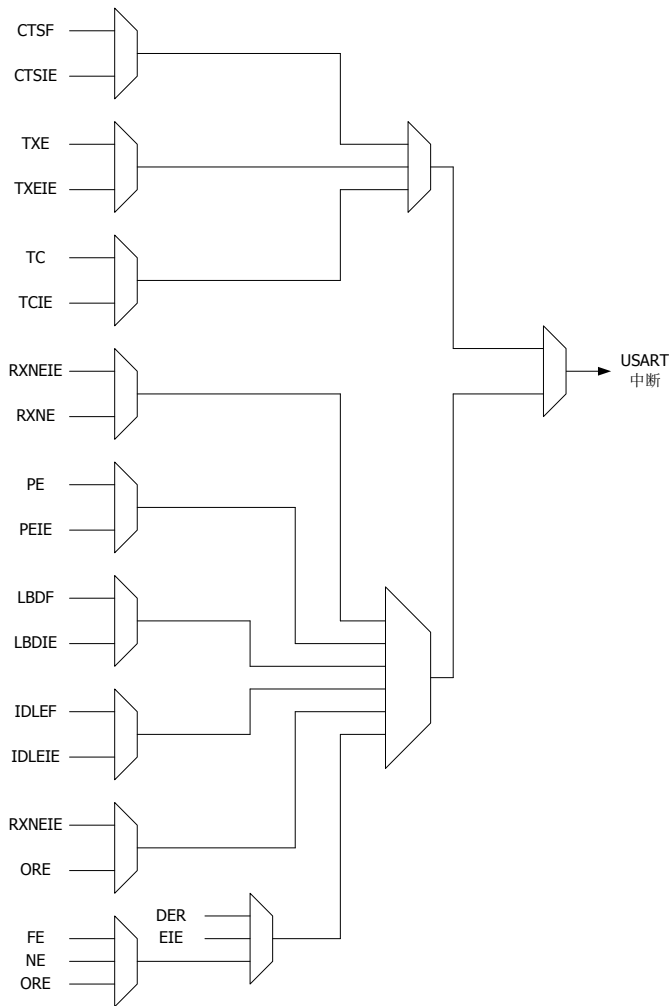
USART 中断事件及标志如下表所示：

表 50 USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TXE	TXEIE
CTS 标志	CTSIF	CTSIE
发送结束	TC	TCIE
接收数据可以读取	RXNE	RXNEIE
检测到过载错误	ORE	
检测到线路空闲	IDLEF	IDLEIE
智能卡模式下，检测到奇偶校验错误	PE	PEIE
LIN 模式下，检测到暂停帧	LBDIF	LBDIE
DMA 接收时，出现接收错误（噪声标志、过载错误、帧格式错误）	NE 或 ORE 或 FE	EIE

以上所有中断事件在发送给中断控制器之前是逻辑或的关系，因此任一时刻 UART 只能向中断控制器产生一个中断请求。不过，软件可以在同一个中断服务子程序中一次处理掉多个中断事件。

图 138 USART 中断映射框图



4 USART 寄存器

4.1 USART 状态寄存器 (USART_SR)

地址偏移: 0x00

复位值 : 0x00C0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CTSIF	LBDF	TXE	TC	RXNE	IDLEF	ORE	NE	FE	PE
						R_W0	R_W0	R	R_W0	R_W0	R	R	R	R	R

类型

位/位域 名称 描述

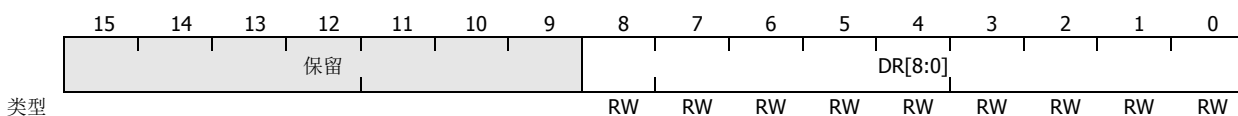
位/位域	名称	描述
15:10	保留	请保持其复位状态
9	CTS _F	<p>CTS 标志</p> <p>0: nCTS 状态线上无变化</p> <p>1: nCTS 状态线上有变化。若 USART_CTRL3 寄存器中的 CTSIE=1, 则将产生中断。</p> <p>假如 USART_CTRL3 寄存器中的 CTSE=1, 则当 nCTS 输入信号翻转时, 本标志将被硬件置位。</p> <p>本位可由软件读取, 可由软件清零。</p>
8	LBD _F	<p>LIN 暂停帧检测标志</p> <p>0: 未检测到 LIN 暂停帧</p> <p>1: 检测到 LIN 暂停帧。若 USART_CTRL2 寄存器中的 LBDIE=1, 则将产生中断。</p> <p>当检测到 LIN 暂停帧时, 本标志将被硬件置位。</p> <p>本位可由软件读取, 可由软件清零。</p>
7	TXE	<p>发送数据缓冲区空</p> <p>0: 无数据传输至移位寄存器</p> <p>1: 数据已传输至移位寄存器。若 USART_CTRL1 寄存器中的 TXEIE=1, 则将产生中断。</p> <p>当 USART_DR 寄存器的内容已经传输至移位寄存器后, 本标志将被硬件置位。</p> <p>本位只读。可由软件通过写 USART_DR 寄存器来清零。</p>
6	TC	<p>发送结束</p> <p>0: 发送过程未结束</p> <p>1: 发送过程已结束。若 USART_CTRL1 寄存器中的 TCIE=1, 则将产生中断。</p> <p>若 TXE=1, 则当一个数据帧发送完成后, 本标志将被硬件置位。</p> <p>先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。</p> <p>本位可由软件读取, 可由软件清零 (多级缓冲通信模式下)。</p>
5	RXNE	<p>接收数据缓冲区非空</p> <p>0: 未收到数据</p> <p>1: 已收到数据, 等待读取。若 USART_CTRL1 寄存器中的 RXNEIE=1, 则将产生中断。</p> <p>当串行移位寄存器的内容已经传输至 USART_DR 寄存器后, 本标志将被硬件置位。先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。</p> <p>本位可由软件读取, 可由软件清零 (多级缓冲通信模式下)。</p>
4	IDLE _F	<p>线路空闲标志</p> <p>0: 未检测到线路空闲</p> <p>1: 已检测到线路空闲。若 USART_CTRL1 寄存器中的 IDLEIE=1, 则将产生中断。</p> <p>当检测到线路空闲时, 本标志将被硬件置位。在 RXNE 置位之前, 本位是不会再次置位的。</p> <p>先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。</p> <p>本位只读。</p>
3	ORE	<p>过载错误</p> <p>0: 未检测到过载错误</p> <p>1: 已检测到过载错误。若 USART_CTRL1 寄存器中的 RXNEIE=1, 则将产生中断。而在多级缓冲通信模式下, 若 USART_CTRL1 寄存器中的 EIE=1, 则将产生中断。</p> <p>若 RXNE=1、且接收移位寄存器又接收完成一帧后, 本标志将被硬件置位。</p> <p>先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。</p> <p>本位只读。</p>

位/位域	名称	描述
2	NE	噪声错误标志 0: 未检测到噪声错误 1: 已检测到噪声错误。在多级缓冲通信模式下, 若 USART_CTRL1 寄存器中的 EIE=1, 则将产生中断。 当接收数据的过程中检测到噪声错误时, 本标志将被硬件置位。 先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。 本位只读。
1	FE	帧格式错误 0: 未检测到帧格式错误 1: 已检测到帧格式错误。在多级缓冲通信模式下, 若 USART_CTRL1 寄存器中的 EIE=1, 则将产生中断。 当接收数据的过程中检测同步丢失、过多噪声信号或者断线字符时, 本标志将被硬件置位。 先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。 本位只读。
0	PE	奇偶校验错误 0: 未检测到奇偶校验错误 1: 已检测到奇偶校验错误。若 USART_CTRL1 寄存器中的 PEIE=1, 则将产生中断。 当接收数据的过程中奇偶校验错误, 本标志将被硬件置位。 先读 USART_SR 寄存器, 再写 USART_DR 寄存器, 即可将本位清零。注意: 只有当 RXNE 置位后, 本位才可清零。 本位只读。

4.2 USART 数据寄存器 (USART_DR)

地址偏移: 0x04

复位值 : 未定义

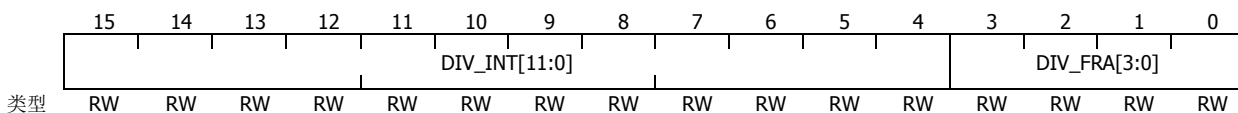


位/位域	名称	描述
15:9	保留	请保持其复位状态
8:0	DR[8:0]	数据内容

4.3 USART 波特率寄存器 (USART_BRR)

地址偏移: 0x08

复位值 : 0x0000



位/位域	名称	描述
15:4	DIV_INT[11:0]	波特率分频系数的整数部分
3:0	DIV_FRA[3:0]	波特率分频系数的小数部分

对 USART_BRR 寄存器执行写操作后，波特率计数器会更新。当 TE 或 RE 被禁用后，波特率计数器会停止运行。

4.4 USART 控制寄存器 1 (USART_CTRL1)

地址偏移: 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	UE	WL	WUPM	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RMMC	SDBK	
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域	名称	描述
15:14	保留	请保持其复位状态
13	UE	USART 使能 0: 禁用 USART 预分频器以及输出 1: 使能 USART 预分频器以及输出
12	WL	字长 0: 8 位 1: 9 位 注意: 在数据发送和接收器件绝对不得修改本位!
11	WUPM	静默模式下的唤醒方式 0: 线路空闲 1: 地址掩码
10	PCE	奇偶校验控制使能 0: 禁用奇偶校验控制 1: 使能奇偶校验控制
9	PS	奇偶校验选择 0: 偶校验 1: 奇校验
8	PEIE	奇偶校验错误中断使能 0: 禁用奇偶校验错误中断 1: 只要 USART_SR 寄存器中的 PE 置位就会产生中断。
7	TXEIE	发送寄存器空中断使能 0: 禁用发送寄存器空中断 1: 只要 USART_SR 寄存器中的 TXE 置位就会产生中断。
6	TCIE	发送结束中断使能 0: 禁用发送结束中断 1: 只要 USART_SR 寄存器中的 TC 置位就会产生中断。

位/位域	名称	描述
5	RXNEIE	接收数据寄存器非空中断/过载错误中断使能 0: 禁用接收数据寄存器非空中断、禁用过载错误中断 1: 只要 USART_SR 寄存器中的 ORE 或 RXNE 置位就会产生中断。
4	IDLEIE	线路空闲中断使能 0: 禁用线路空闲中断 1: 只要 USART_SR 寄存器中的 IDLEF 置位就会产生中断。
3	TE	发送使能 0: 禁用发送功能 1: 使能发送功能
2	RE	接收使能 0: 禁用接收功能 1: 使能接收功能, 开始检测有无起始位
1	RMMC	多级通信时的接收器工作模式 0: 接收器处于正常工作模式 1: 接收器处于静默模式 当识别出唤醒序列时, 本位将被硬件清零。 USART 必须先接收一个数据帧之后才能选择静默模式。 若配置成地址掩码唤醒, 那么在 RXNE=1 的情况下是无法修改的。
0	SDBK	发送暂停帧 0: 不发送暂停帧 1: 发送暂停帧 在暂停帧的停止位时, 本位将被硬件清零。

4.5 USART 控制寄存器 2 (USART_CTRL2)

地址偏移: 0x10

复位值 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	LINEN	STPL[1:0]		CKE	CPOL	CPHA	LBCL	保留	LBDIE	LBDL	保留	WADD[3:0]			
类型		RW	RW	RW	RW	RW	RW	RW		RW	RW		RW	RW	RW	RW

位/位域	名称	描述
15	保留	请保持其复位状态
14	LINEN	LIN 模式使能 0: 禁用 LIN 模式 1: 使能 LIN 模式
13:12	STPL[1:0]	停止位长度 00: 1 停止位 01: 0.5 停止位 10: 2 停止位 11: 1.5 停止位
11	CKE	CK 管脚使能 0: 禁用 CK 管脚 1: 使能 CK 管脚

位/位域	名称	描述
10	CPOL	时钟极性 0: 在同步通信模式下, CK 管脚对外发送的空闲状态为低电平 1: 在同步通信模式下, CK 管脚对外发送的空闲状态为高电平
9	CPHA	时钟相位 0: 在同步通信模式下, 在首个时钟跳变沿采样第一个数据位 1: 在同步通信模式下, 在第二个时钟跳变沿采样第一个数据位
8	LBCL	最后一位时钟脉冲 0: 在同步通信模式下, 最后一个数据位的时钟脉冲不输出到 CK 管脚上 1: 在同步通信模式下, 最后一个数据位的时钟脉冲输出到 CK 管脚上
7	保留	请保持其复位状态
6	LBDIE	LIN 暂停帧中断使能 0: 禁用 LIN 暂停帧中断 1: 只要 USART_SR 寄存器中的 LBDF 置位就会产生中断。
5	LBDL	LIN 暂停帧长度 0: 检测 10 位暂停帧 1: 检测 11 位暂停帧
4	保留	请保持其复位状态
3:0	WADD[3:0]	多机通信的节点地址, 用于在静默模式下匹配唤醒。

4.6 USART 控制寄存器 3 (USART_CTRL3)

地址偏移: 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CTSIE	CTSE	RTSE	DET	DER	SCME	NACKE	HDE	IRLP	IRME	EIE	
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

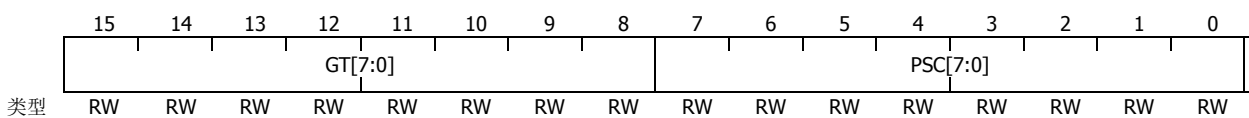
位/位域	名称	描述
15:11	保留	请保持其复位状态
10	CTSIE	CTS 中断使能 0: 禁用 CTS 中断 1: 只要 USART_SR 寄存器中的 CTS 置位就会产生中断。
9	CTSE	CTS 使能 0: 禁用 CTS 硬件流控 1: 使能 CTS 硬件流控
8	RTSE	RTS 使能 0: 禁用 RTS 硬件流控 1: 使能 RTS 硬件流控, 只有当接收缓冲有剩余空间时才请求数据
7	DET	发送 DMA 使能 0: 禁用发送的 DMA 模式 1: 使能发送的 DMA 模式

位/位域	名称	描述
6	DER	接收 DMA 使能 0: 禁用接收的 DMA 模式 1: 使能接收的 DMA 模式
5	SCME	智能卡模式使能 0: 禁用智能卡模式 1: 使能智能卡模式
4	NACK	智能卡模式下的 NACK 使能 0: 奇偶校验错误时也不发送 NACK 1: 奇偶校验错误时发送 NACK
3	HDE	半双工使能 0: 禁用半双工模式 1: 使能半双工模式
2	IRLP	IrDA 低功耗模式 0: 正常工作模式 1: 低功耗模式
1	IRME	IrDA 模式使能 0: 禁用 IrDA 模式 1: 使能 IrDA 模式
0	EIE	多级缓冲通信模式的错误中断使能 0: 禁用错误中断 1: 在多级缓冲通信模式下，只要 USART_SR 寄存器中的 FE 或 ORE 或 NE 置位就会产生中断。

4.7 USART 间隔时间及红外预分频寄存器 (USART_GTPR)

地址偏移: 0x18

复位值 : 0x0000



位/位域	名称	描述
15:8	GT[7:0]	智能卡模式下的间隔时间值。

位/位域	名称	描述
7:0	PSC[7:0]	<p>本位域所确定的预分频系数，是 IrDA 低功耗模式下对系统时钟分频、以提供低功耗红外脉冲的。</p> <p>0x00: 保留，不得编程为此数值。</p> <p>0x01: 时钟源除以 1</p> <p>0x10: 时钟源除以 2</p> <p>.....</p> <p>对于 IrDA 正常工作模式，此位域只能设置为 0x01</p> <p>在智能卡模式下，预分频系数是取值的 2 倍。</p> <p>00000: 保留，不得编程为此数值。</p> <p>00001: 时钟源除以 2</p> <p>00010: 时钟源除以 4</p> <p>00011: 时钟源除以 6</p> <p>.....</p>

第14章 通用串行总线全速接口（USB2.0 FS）

1 简介

通用串行总线（USB）2.0 全速接口是一个 4 线制的总线接口，能够在 USB 主机以及本器件所实现的功能之间进行通信。主控制器通过基于令牌的协议分配其连接的各个从设备的 USB 带宽。USB 总线支持热插拔功能以及从设备的动态配置。所有交互过程均由主控制器发起。在主控 PC 与本器件的系统存储器之间所发生的数据传输均经过专用的包缓冲，即由 USB 外设直接访问的 512 字节专用 SRAM。请注意：这 512 字节是与 CAN 共享的。

交互格式化完全由硬件保障，包括 CRC 的生成及校验。USB 主机按照 1 ms 的时间帧调度各个会话。每帧包含一个帧起始（Start-of-Frame，简称为 SOF）标识符以及会话内容，向从设备节点/从从设备节点传输数据。每个从设备最多可以有 16 个逻辑节点或 32 个物理节点。本微控制器支持最大 8 个节点。

节点的传输总共有 4 种类型：控制传输用于对从设备进行配置；中断传输用于实现周期性的数据传输；批量传输用于传输延迟并不重要的场合；同步传输有确定的传输时间，但不包含错误检测。

2 主要特点

- USB 2.0 全速从机控制器
- 支持最多 8 个可配置的节点
- 支持双缓冲的批量节点/同步节点
- 每个节点都支持控制、批量、同步或中断节点类型
- 支持挂起/恢复操作
- 与 CAN 共享 512 字节的专用 SRAM

3 功能描述

3.1 基本功能

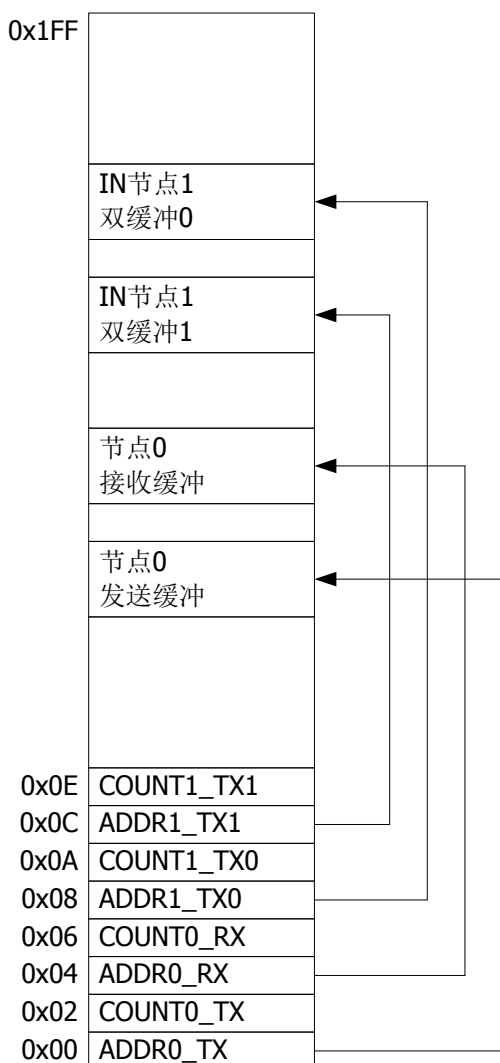
USB 控制器内置一个模拟收发器（ATX）。USB ATX 能够在 USB 总线的双向差分信号 USB_DP 及 USB_DM 上收发数据。SIE 实现了完整的 USB 协议层。出于保障速度的考虑，SIE 是完全硬件实现的，无需软件干预。它负责处理 USB RAM 的节点缓冲与 USB 总线之间的数据传输。此模块的功能包括：同步序列识别、并-串转换、位填充/去填充、CRC 校验/生成、PID 校验/生成、地址识别，以及握手包的取值/生成。

3.2 工作流程

缓冲描述符表

缓冲描述符表是一种数据结构，每个表项包含一个缓冲区的地址及长度。USB 外设通过缓冲描述符来收发数据。缓冲描述符表的首个表项是由 USB_BTABLE 定义的。一个有效的节点通常包含两个报文缓冲，分别用于接收和发送。每个表项包含 4 个 16 位字，并且按照 8 字节对齐。

图 139 缓冲描述符表的用法示例 (USB_BTABLE=0)



节点初始化

- 初始化 ADDRn_TX/ADDRn_RX 寄存器，填写发送/接收数据缓冲区地址
- 按照节点的用途，配置 USB_EPnR 寄存器的 EP_TYPE 以及 EP_KIND
- 对于发送，初始化 COUNTn_TX 并使能 STAT_TX；对于接收，初始化 COUNTn_RX、BL_SIZE

以及 NUM_BLOCK，之后使能 STAT_RX

对于同步节点和双缓冲批量节点，无论发送还是接收的相关位域都必须初始化。一旦节点使能，USB_EPnR 寄存器、缓冲地址、COUNT 位域都不得被应用软件所修改。当传输完成后，系统通过 CTR 中断提示应用软件。

数据发送（IN 包）

当某个有效节点收到了 IN 令牌包，将会从 ADDRn_TX 开始读取并发送 COUNTn_TX 个字节，并将 CRC 结果发给主机。假如节点并未使能，那么取决于 STAT_TX 的状态，将会发送 NAK 或 STALL 握手包。

USB 外设从 ADDRn_TX 发送报文数据，且始终按照低位在前的顺序。

从主机接收到 ACK 包后，USB 外设将翻转 DTOG_TX，设置 STAT_TX = 10 (NAK)，并将 CTR_TX 置位。应用软件可通过查询 USB_ISTR 寄存器的 EP_ID 以及 DIR 来确认哪个节点已经完成传输。当包存储器填满数据后，应用软件可将 STAT_TX 设置为 11 (VALID)，即可重新使能节点、开始下一个传输。

数据接收（OUT 包及 SETUP 包）

USB 外设采用 ADDRn_RX 作为缓冲来存储接收的数据，并根据实际接收到的字节数更新 COUNTn_RX 位域。BL_SIZE 以及 NUM_BLOCK 的值用于计算 BUF_COUNT，并据此判断缓冲区是否溢出。USB 外设按照低位在前的顺序从主机接收数据，同时计算 CRC 值。当检测到 DATA 包传输结束时，计算出的 CRC 值将与接收到的 CRC 值进行比较。假如没有错误产生，将向主机发送 ACK 握手包。

假如接收过程中遇到了任何错误，USB 外设将 ERR 置位，并且继续将数据复制到报文缓冲中，但并不发送 ACK 握手包。USB 外设自身能够从接收错误中恢复，并继续处理下一个传输。USB 外设的访问超出由 BL_SIZE 和 NUM_BLOCK 定义的包缓冲范围。接收的 2 字节 CRC 校验值也会复制到包缓冲中，且紧跟在数据字节后。假如数据长度大于实际分配的长度，那么超出的数据是不会写入存储器的。这种情况称为缓冲过载。此时将发送 STALL 握手包，并且将当前会话判定为失败。

根据 USB_EPnR 寄存器中 STAT_RX 的设置，假如某个被寻址的节点并不可用，从设备将发送一个 NAK 握手包或 STALL 握手包，而非 ACK 握手包，并且不会向接收缓冲写入数据。接收缓冲按照 ADDRn_RX 中包含的地址开始写入，写入的字节数取决于接收到的数据包长度（包括 CRC 校验值，即数据载荷长度+2），或已分配的最高存储器地址（由 BL_SIZE 以及 NUM_BLOCK 决定），是这二者中的较小值。通过这种方式，USB 外设绝对不会占用超出被分配的接收缓冲区域。假如数据包载荷的长度（应用所占用的实际字节数）大于已分配的缓冲区，USB 外设即会检测到缓冲过载状况。此时，从机将向主机发送 STALL 握手包而非 ACK 握手包，提醒主机当前会话出现问题，不会产生中断，并且将当前会话判定为失败。

假如没有遇到任何错误，那么 USB 外设将向主机发送 ACK 握手包，翻转 DTOG_RX 位，将 STAT_RX

设置为 10(NAK), 并将 CTR_RX 置位。应用软件可查询 USB_ISTR 寄存器的 EP_ID 以及 DIR 位, 以确定哪一节点触发的 CTR_RX。处理完接收的数据后, 应用软件可将 STAT_RX 设置为 11(VVALID), 启动另一个会话。

控制传输

控制传输要求主机向从设备发送一个 SETUP 会话, 描述从设备应当执行的控制访问的类型。SETUP 会话后面将是 0 个或若干个控制 DATA 会话, 携带被请求的访问的特定信息。最后将通过一个 STATUS 会话完成控制传输, 并且允许节点将控制传输的状态返回给客户端软件。当控制传输的 STATUS 会话完成后, 主机可以操作该节点的下一个控制传输。

要初始化控制传输, 应将 DTOG_TX 和 DTOG_RX 分别置为 1 和 0, 并将 STAT_TX 和 STAT_RX 都设置为 01 (NAK)。根据 SETUP 包的内容, 应用软件可以确定下一个会话是 IN 还是 OUT。当控制传输期间发生了 CTR_RX 事件时, 必须先检查 SETUP 位。假如 SETUP=1, 那就表明是 SETUP 会话, 否则就是标准的 OUT 会话。USB 外设通过解析 SETUP 会话的内容, 获取其关心的数据数量和传输方向, 并且要求 USB 外设将未使用的方向置为 STALL, 除了最后一个数据级。

在最后一个数据级, 应用软件对 NAK 设置相反方向。这将使得主机等待控制操作的完成。假如操作成功完成, 软件将把 NAK 改为 VALID, 否则改为 STALL。假如状态级是一个 OUT, 那么应当将 STATUS_OUT 置位, 这样携带数据非零的 STATUS 会话将会被应答以 STALL, 表明已经发生了错误。

当 STAT_RX = 01 (STALL) 或 10 (NAK), 并且已收到 SETUP 令牌时, USB 将接收数据, 执行被要求的数据传输并且回发一个 ACK 握手包。假如该节点之前有 CTR_RX 请求尚未被应用软件确认 (即 CTR_RX 仍然被前一次完成的接收置位), 那么 USB 将丢弃 SETUP 会话, 不管状态如何都不应答任何握手包, 这样可以模拟出接收错误状态, 强制主机再次发送 SETUP 令牌。这样做是为了避免出现对同一个节点的背靠背的 SETUP 会话时, 在已经触发 CTR_RX 中断的情况下无法及时提醒主机后一次会话的结果。

双缓冲节点

双缓冲功能是为了提高批量传输的性能。DTOG 位和 SW_BUF 位负责流控。当传输完成后, USB 外设翻转 DTOG 位; 当数据被复制后, 应用软件需要翻转 SW_BUF 位。除了首次传输以外, 只要 DTOG 位与 SW_BUF 位相同, 传输将会暂停, 并且将向主机发送 NAK 握手包。当两个位不同时, 传输将会继续。

要使能双缓冲功能, EP_TYPE 以及 EP_KIND 需要被配置为:

- EP_TYPE = 00
- EP_KIND = 1

表 51 双缓冲的用法

节点类型	DTOG	SW_BUF	USB 外设所使用的包缓冲	应用软件所使用的包缓冲
OUT	0	1	ADDRn_RX_0/COUNTn_RX_0 缓冲描述符列表地址	ADDRn_RX_1/COUNTn_RX_1 缓冲描述符列表地址
	1	0	ADDRn_RX_1/COUNTn_RX_1 缓冲描述符列表地址	ADDRn_RX_0/COUNTn_RX_0 缓冲描述符列表地址
IN	0	1	ADDRn_TX_0/COUNTn_TX_0 缓冲描述符列表地址	ADDRn_TX_1/COUNTn_TX_1 缓冲描述符列表地址
	1	0	ADDRn_TX_1/COUNTn_TX_1 缓冲描述符列表地址	ADDRn_TX_0/COUNTn_TX_0 缓冲描述符列表地址

3.3 USB 中断

USB 控制器有 3 个中断线：USB 低优先级中断（通道 20）、USB 高优先级中断（通道 19）以及 USB 唤醒中断（通道 42）。软件可对 USB 中断路由寄存器的相应位进行编程，以将特定的中断条件链接到位于 NVIC 中的这 3 个中断信号上。假如中断状态位和相应的中断使能位都置位，那么硬件就会产生中断。只要中断条件产生，不管中断使能位有没有置位，中断状态位都将被硬件置位。

- USB 低优先级中断（通道 20）：可被所有 USB 事件触发
- USB 高优先级中断（通道 19）：只能被正确传输事件触发，用于同步传输或双缓冲批量传输
- USB 唤醒中断（通道 42）：由唤醒事件触发

3.4 同步传输

同步传输可以实现固定的数据传输速率以及固定的延迟，但是并不支持数据重发以及总线错误响应。此时只有接收方能够确定发送是否产生错误。USB 底层协议并不允许同步传输通道中将握手包回发给发送方。一般来说，不会向发送方发送握手包来说明接收方是否成功接收。因此，同步会话并没有握手阶段，传输数据包之后也没有 ACK 包。同步会话不支持数据翻转，只有 DATA0 PID 用于开始一个数据包。

将 EP_TYPE 设置为 10，即可将该节点定义为同步节点。此时 STAT 位域只有两个可用的值：00（禁用）和 11（VALID），其他值均为非法。应用软件可以自行实现双缓冲以提高性能。通过在每个会话时交换发送和接收包缓冲，应用软件可以将数据复制进缓冲或复制出缓冲，同时由 USB 外设继续处理另一个缓冲的数据发送或接收。通过查询 DTOG 位即可知道 USB 外设当前正在使用哪个缓冲区。

应用软件按照要用的首个缓冲区初始化 DTOG；在每次会话结束时，取决于使能的方向，CTR_RX 或 CTR_TX 标志将会置位（不管是否有 CRC 错误或缓冲过载状况。假如有错误，则 ERR 将会置位）。同时，USB 外设将翻转 DTOG 位，但并不影响 STAT 位域。

3.5 挂起/恢复事件

USB 协议一向强调由 USB 从设备进行电源管理的重要性，因为当从设备从总线获取电源（总线供电设备）的话，这一点将尤为重要。总线供电设备必须满足下述限制：

- 处于非配置状态的从设备，从 USB 总线最多获取 100 mA 电流。
- 已配置的设备，只能按照配置描述符的 Max Power 域的设置获取电流，且最大不得超过 500 mA。
- 已挂起设备最多获取 500 μ A 电流。

假如 USB 总线上没有活动超过 3 ms，那么设备将进入挂起态。假如有来自主机的发送，则挂起设备将会唤醒。USB 外设还支持软件初始化的远程唤醒。要启动远程唤醒功能，应用软件必须使能所有时钟，并清零挂起位。这将导致硬件产生一个远程唤醒信号的上行数据流。

将 FSUSP 位置为 1，即可使能挂起模式，这样将会禁用检查 SOF 接收。将 LP_MPDE 置为 1，将会关闭 USB 模拟收发器的静态功耗，但是此时仍能检测到恢复信号。

4 USB 寄存器

4.1 USB 控制寄存器（USB_CNTR）

地址偏移：0x40

复位值：0x0003

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	保留			RESUME	FSUSP	LPMPDE	PDWN	FRES
类型	RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

位/位域	名称	描述
15	CTRM	正确传输中断掩码
14	PMAOVRM	包缓冲区溢出/下溢中断掩码
13	ERRM	错误中断掩码
12	WKUPM	唤醒中断掩码
11	SUSPM	挂起模式中断掩码
10	RESETM	USB 复位中断掩码
9	SOFM	帧起始中断掩码
8	ESOFM	预期的帧起始中断掩码
7:5	保留	请保持其复位状态
4	RESUME	恢复请求
3	FSUSP	强制挂起

位/位域	名称	描述
2	LP_MODE	低功耗模式
1	PDWN	关闭供电
0	FRES	强制复位

4.2 USB 中断状态寄存器 (USB_ISTR)

地址偏移: 0x44

复位值 : 0x0000

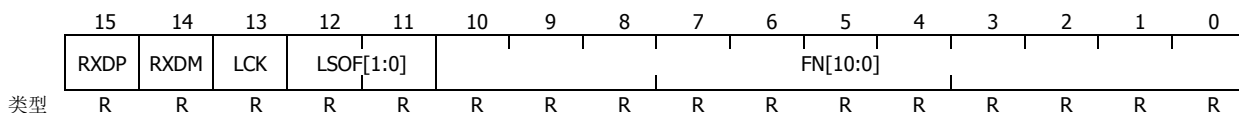
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMA OVM	ERR	WKUP	SUSP	RESET	SOF	ESOF	保留			DIR	EP_ID[3:0]			
类型	R	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0			R	R	R	R	R

位/位域	名称	描述
15	CTR	正确传输中断 当一个会话成功完成后, 本位将被硬件置位。
14	PMAOVR	包缓冲区过载/欠载中断 本位由硬件置位, 表明包缓冲大小并不适合保持所传输的数据。
13	ERR	错误中断 会话期间发生错误。
12	WKUP	唤醒中断 挂起模式下由硬件置位, 表明检测到了总线活动。
11	SUSP	挂起模式中断 当无流量超过 3 ms, 硬件将本位置位, 表明产生了挂起请求。
10	RESET	USB 复位中断 当检测到 USB 复位信号时, 本位将被硬件置位。
9	SOF	帧起始中断 当接收到了新的 SOF 时, 本位将被硬件置位。
8	ESOF	预期的帧起始中断 本位由硬件置位, 表明原本预期收到一个 SOF 包, 但却没有收到。
7:5	保留	请保持其复位状态
4	DIR	会话传输方向 本位由硬件置位, 表明会话的传输方向。 0: OUT 类型 1: IN 类型
3:0	EP_ID[3:0]	节点标识符 由硬件改写, 用以表明当前会话所关联的节点。

4.3 USB 帧编号寄存器 (USB_FNR)

地址偏移: 0x48

复位值 : 0x0XXX, 其中 X 代表未定义

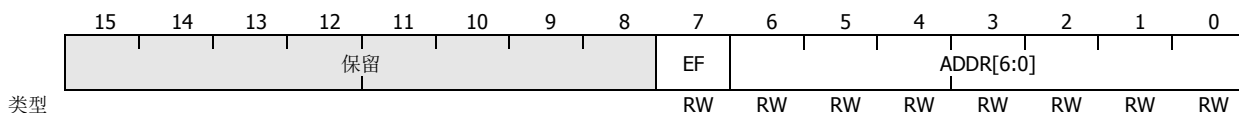


位/位域	名称	描述
15	RXDP	接收数据+线状态 代表 DP 线状态
14	RXDM	接收数据-线状态 代表 DM 线状态
13	LCK	已锁定 本位由硬件置位, 表明接收到了至少两个连续的 SOF 包。
12:11	LSOF[1:0]	丢失 SOF 每当发生 ESOF 事件时, 此计数器累加。 一旦收到了 SOF, 本位域将自动清零。
10:0	FN[10:0]	帧编号 11 位的帧编号。

4.4 USB 从设备地址 (USB_DADDR)

地址偏移: 0x4C

复位值 : 0x0000

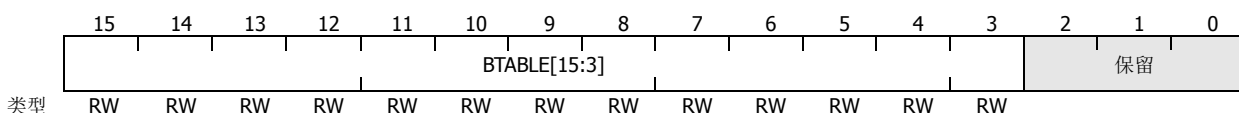


位/位域	名称	描述
15:8	保留	请保持其复位状态
7	EF	使能功能 本位可由软件置位, 使能 USB 从设备。
6:0	ADDR[6:0]	从设备地址 系统复位后, 从设备地址默认恢复至 0x00。若使能位置位, 则从设备会响应功能地址 DEV_ADDR 的报文。

4.5 缓冲表地址 (USB_BTABLE)

地址偏移: 0x50

复位值 : 0x0000

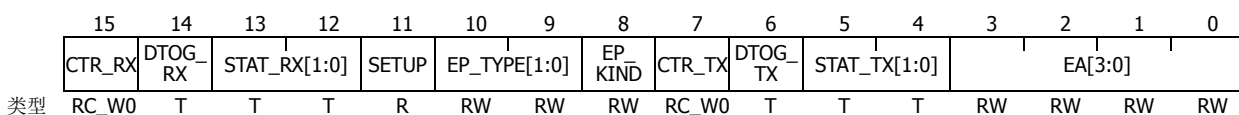


位/位域	名称	描述
15:3	BTABLE[15:3]	缓冲表 所分配缓冲（512 字节片上 SRAM）的起始地址，用于保存缓冲描述符表、包缓冲。
2:0	保留	请保持其复位状态

4.6 USB 节点 n 寄存器 (USB_EPnR) (n=1~7)

地址偏移：0x00~0x1C

复位值：0x0000



位/位域	名称	描述
15	CTR_RX	接收的正确传输 当 OUT/SETUP 会话成功完成后，本位将被硬件置位。 软件向本位写 0，即可将其清零。
14	DTOG_RX	接收的数据翻转 本标志位代表非同步节点的翻转数据位（0=DATA0,1=DATA1）。 本位用于实现双缓冲节点的流控功能。 本位用于同步节点的缓冲区交换。
13:12	STAT_RX[1:0]	接收传输的状态位 软件向本位域写 1，即可翻转。写 0 则不变。 参见表 52。
11	SETUP	SETUP 会话结束 当 SETUP 会话结束后，本位将由硬件置位。
10:9	EP_TYPE[1:0]	节点类型 参见表 53。
8	EP_KIND	节点类别 其具体含义取决于节点类型的设置。 参见表 54。
7	CTR_TX	发送的正确传输 当 IN 会话成功完成后，本位将被硬件置位。 软件向本位写 0，即可将其清零。
6	DTOG_TX	发送的数据翻转 本标志位代表非同步节点的翻转数据位（0=DATA0,1=DATA1）。 本位用于实现双缓冲节点的流控功能。 本位用于同步节点的缓冲区交换。

位/位域	名称	描述
5:4	STAT_TX[1:0]	发送传输的状态位 参见表 55。
3:0	EA	节点地址 用于指示会话到目标节点。

表 52 接收状态编码

STAT_RX[1:0]	含义
00	DISABLED : 忽略此节点的所有接收请求
01	STALL : 握手状态为 STALL
10	NAK : 握手状态为 NAK
11	VALID : 使能节点的接收

表 53 节点类型编码

EP_TYPE[1:0]	含义
00	BULK 猝发节点
01	CONTROL 控制节点
10	ISO 同步节点
11	INTERRUPT 中断节点

表 54 节点类别含义

EP_TYPE[1:0]	EP_KIND 含义	
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT

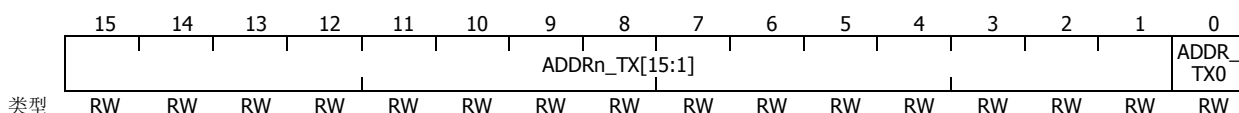
表 55 发送状态编码

STAT_TX[1:0]	含义
00	DISABLED : 忽略此节点的所有发送请求
01	STALL : 握手状态为 STALL
10	NAK : 握手状态为 NAK
11	VALID : 使能节点的发送

4.7 发送缓冲地址 n (USB_ADDRn_TX)

地址偏移: [USB_BTABLE] + n*16

USB 本地地址: [USB_BTABLE] + n*8

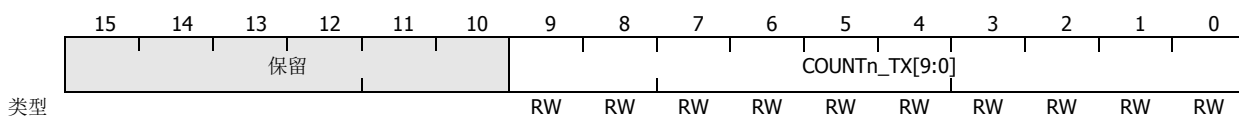


位/位域	名称	描述
15:1	ADDRn_TX[15:1]	发送缓冲地址 IN 令牌的报文缓冲起始地址。
0	ADDR_TX0	本位必须为 0

4.8 发送字节数目 n (USB_COUNTn_TX)

地址偏移: $[USB_BTABLE] + n * 16 + 4$

USB 本地地址: $[USB_BTABLE] + n * 8 + 2$

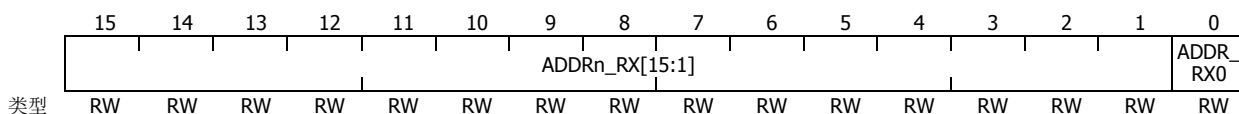


位/位域	名称	描述
15:10	保留	请保持其复位状态
9:0	COUNTn_TX[9:0]	发送字节数 收到下一个 IN 令牌后, 要发送的字节数。

4.9 接收缓冲地址 n (USB_ADDRn_RX)

地址偏移: $[USB_BTABLE] + n * 16 + 8$

USB 本地地址: $[USB_BTABLE] + n * 8 + 4$

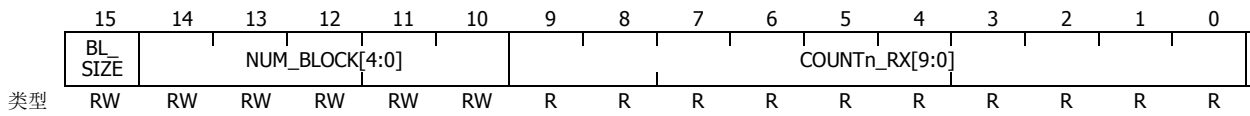


位/位域	名称	描述
15:1	ADDRn_RX[15:1]	发送缓冲地址 OUT 令牌的报文缓冲起始地址。
0	ADDR_RX0	本位必须为 0

4.10 接收字节数目 n (USB_COUNTn_RX)

地址偏移: $[USB_BTABLE] + n * 16 + 12$

USB 本地地址: $[USB_BTABLE] + n*8 + 6$



位/位域	名称	描述
15	BL_SIZE	块大小 0: 块大小为 2 字节 1: 块大小为 32 字节
14:10	NUM_BLOCK[4:0]	块数目 包缓冲所分配的块数目。
9:0	COUNTn_RX[9:0]	接收字节数 收到下一个 OUT/SETUP 令牌后, 要接收的字节数。

第15章 控制器局域网（bxCAN）

1 简介

bxCAN (Basic Extended CAN, 于基础上扩展 CAN) 模块用于连接 CAN 网络。该模块支持 CAN 2.0A 以及 2.0B 协议。bxCAN 模块能够全自动地处理 CAN 报文的收发。bxCAN 提供 14 个可裁剪、可配置的标识符过滤槽, 可从输入报文中甄选出软件所需的报文, 并自动丢弃其余不需要的报文。bxCAN 提供 3 个发送邮箱, 以供软件建立报文。发送调度器决定哪一个邮箱中的报文应当优先发送。每个 FIFO 中可存储 3 条完整的报文。FIFO 完全由硬件管理。硬件使用两个接收 FIFO 保存输入的报文。CAN 控制器还支持时间触发的 CAN 通讯的所有硬件功能, 可应用于安全关键的应用场合。

bxCAN 提供了多个控制寄存器、状态寄存器、配置寄存器, 以便:

- 配置 CAN 参数, 例如波特率
- 请求发送
- 处理接收
- 管理中断
- 获取诊断信息

2 主要特点

- 支持 CAN 2.0A 及 2.0B 协议
- 波特率最高可达 1Mbit/s
- 支持时间触发通讯 (Time Triggered Communication, 简称为 TTC)
- 可屏蔽的中断

发送

- 三个发送邮箱
- 报文具有优先级
- SOF 发送具有时间戳

接收

- 两个接收 FIFO, 每个深度 3 条报文
- 14 个可裁剪、可配置的标识符过滤槽
- FIFO 锁定
- SOF 接收具有时间戳

时间触发通信

- 自动禁止重发
- 16 位的自由定时器
- 时间戳在最后两个数据字节中发送

请注意：CAN 与 USB 不能同时使用，因为二者要共用一块 512 字节的专用 SRAM（软件不可占用）。二者可以在同一个应用中，但无法同时使用。

3 功能描述

3.1 工作模式

bxCAN 接口有三种工作模式：

- 睡眠工作模式
- 初始化工作模式
- 正常工作模式

睡眠工作模式

睡眠工作模式是复位后的默认工作模式。此外 bxCAN 模块的时钟停止，模块处于低功耗状态。

当 MCR 寄存器中的 SLEEP 标志位置位时，bxCAN 模块将进入睡眠模式。之后 MSR 寄存器中的 SLEEPF 标志位将置位。

当 MCR 寄存器的 AWUM 位为 1 时，若 Rx 信号中检测到 SOF 位，即可自动退出睡眠工作模式。软件将 MCR 寄存器的 SLEEP 位清零，亦可退出睡眠工作模式。

初始化工作模式

当需要修改 CAN 总线通信相关参数时，应将 bxCAN 模块进入初始化模式。

将 MCR 寄存器的 INRQ 置位，即可进入初始化工作模式；反之，将其清零即可退出初始化工作模式。

正常工作模式

初始化完成后，bxCAN 模块即可进入正常工作模式，此时即可准备与其它 CAN 节点进行通讯。

将 MCR 寄存器的 INRQ 清零，即可进入正常工作模式。

3.2 通讯模式

bxCAN 接口有以下四种通讯模式：

- 静默通讯模式
- 环回通讯模式
- 环回+静默通讯模式
- 正常通讯模式

静默通讯模式

所谓静默通讯模式，就是允许接收、禁止发送的工作模式。

此时 **bxCAN** 接口的 **Rx** 脚可以正常从 **CAN** 网络获取信号，而 **Tx** 脚始终保持逻辑 1。

将 **BTR** 寄存器的 **SILM** 置位，**bxCAN** 模块即进入静默通讯模式。将 **SLIM** 清零，则 **bxCAN** 模块退出静默通讯模式。

当节点需要用于监控网络报文时，静默通讯模式很有用。

环回通讯模式

所谓环回通讯模式，就是发送的报文将在芯片内部转发到接收邮箱 **FIFO** 中。

将 **BTR** 寄存器的 **LBKM** 置位，即可进入环回通讯模式；反之，将 **LBKM** 清零即可退出环回通讯模式。

环回通讯模式适用于节点自检。

环回+静默通讯模式

所谓环回+静默通讯模式，就是 **RX** 脚和 **TX** 脚实际上并不与 **CAN** 网络发生信号交互，节点发送的报文将在芯片内部转发到接收邮箱 **FIFO** 中。

将 **BTR** 寄存器的 **LBKM** 和 **SILM** 都置位，即可进入环回+静默通讯模式；反之，将其至少一个清零，即可退出环回+静默通讯模式。

环回+静默通讯模式适用于节点自检。此时 **TX** 管脚将保持逻辑 1，**RX** 管脚保持高阻态。

正常通讯模式

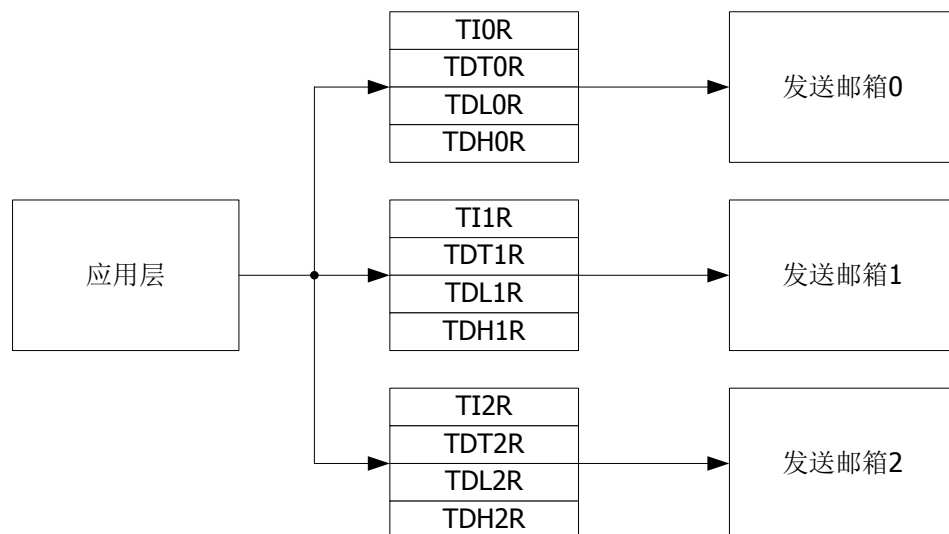
只要 **BTR** 寄存器中的 **LBKM** 及 **SILM** 未置位，则默认就是正常通讯模式。

3.3 数据发送

发送寄存器

bxCAN 提供三个发送邮箱对应用可见。用户可以通过以下四个寄存器访问发送邮箱：**TIR**、**TDTR**、**TDLR**、**TDHR**，如下图所示。

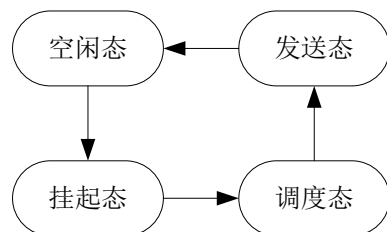
图 140 发送寄存器



发送邮箱状态

发送邮箱为空闲时才可使用，此谓空闲态。假如数据已经写入邮箱，那么将 **TIR.TXRQ** 置位即准备发送该报文，此谓挂起态。假如多个邮箱同时处于挂起态，则需要调度其发送顺序，此谓调度态。优先级最高的邮箱才能进入发送态并开始发送报文。报文发送完成后，邮箱返回至空闲态。如下图所示。

图 141 发送邮箱的基本工作状态



状态及错误

TSR 寄存器包含发送状态及错误的标志位：**RQCP**、**TXOK**、**ALST**、**TERR**。

- **RQCP**: 即 Request Completed 之缩写。一般来说，当发送邮箱中的报文发送完成后，此标志位置位。
- **TXOK**: 即 Transmitted OK 之缩写。当发送邮箱中的报文发送完成后，此标志位置位。
- **ALST**: 即 Arbitration Lost 之缩写。当发送邮箱中的报文因仲裁失败而未能发出时，此标志位置位。
- **TERR**: 即 Transmission Error 之缩写。当检测到 CAN 总线上的错误、导致报文发送失败，

此标志位置位。

发送报文的步骤

按以下步骤从 **bxCAN** 发送报文：

第 1 步：选择一个空闲的发送邮箱。

第 2 步：将应用数据填入四个寄存器内。

第 3 步：将 **TIR.TXRQ** 置位。

第 4 步：检查发送状态。例如，当发送完成后，**RQCP** 以及 **TXOK** 应当置位。

发送选项

中止

将 **TSR.ABRQ** 置位，可中止某个邮箱报文的发送。

假如发送邮箱的状态为挂起态或调度态，那么中止发送可立即完成。

假如发送邮箱处于发送态，那么中止发送可能导致两种后果：若发送成功，则 **TSR** 寄存器的 **TXOK** 与 **RQCP** 将会置位，并且邮箱状态将变为空闲态。假如发送失败，则邮箱状态变为调度态，并立即完成中止发送。

优先级

当挂起的发送邮箱不止一个时，发送顺序取决于 **MCR** 寄存器的 **TXFP** 位。

若 **TXFP = 1**，那么三个发送邮箱将按照 **FIFO** 形式工作。

若 **TXFP = 0**，那么标识符最小的发送邮箱将具有最高的发送优先级。假如标识符都相同，那么编号最小的邮箱将最先被调度。

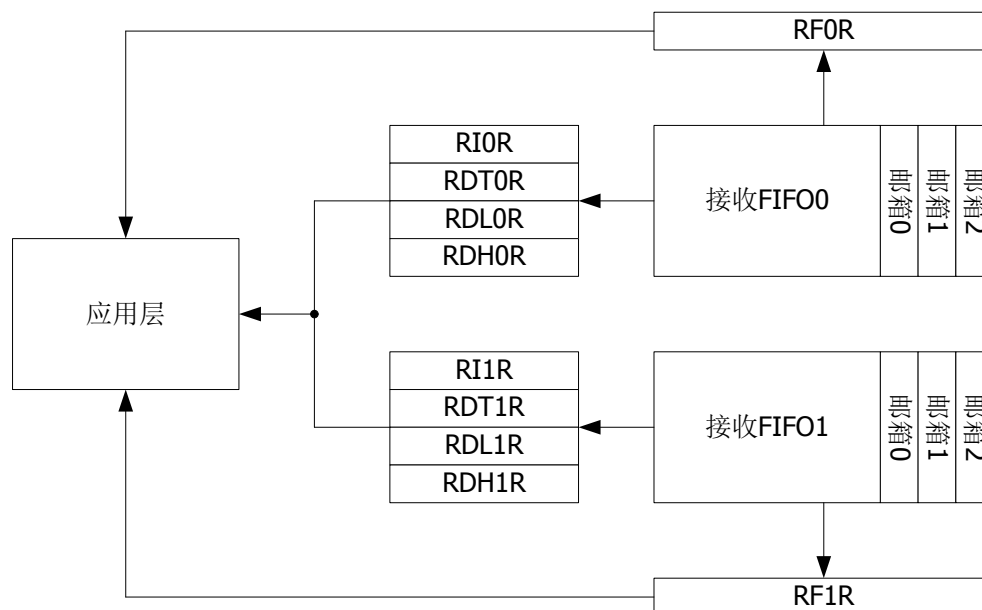
3.4 数据接收

接收寄存器

bxCAN 提供两个接收 **FIFO** 对应用可见。用户可以通过以下五个寄存器访问接收 **FIFO**：**RFR**、**RIR**、**RDTR**、**RDLR** 及 **RDHR**。**FIFO** 的状态及操作是由 **RFR** 寄存器掌控的。接收的报文数据可通过以下寄存器获得：**RIR**、**RDTR**、**RDLR** 及 **RDHR**。

每个 **FIFO** 包含 3 个接收邮箱，如下图所示。

图 142 接收寄存器



接收 FIFO

每个接收 FIFO 有三个接收邮箱。接收的报文按接收顺序保存在邮箱中。最先收到的报文可以最先被应用访问。

接收 FIFO 中报文的数量及状态可通过 RF0R 及 RF1R 寄存器访问。

假如接收 FIFO 0 中储存了不止一条报文，那么将 RF0R.RFOM 置位即可从该接收 FIFO 中读出一条报文。报文的数据将保持在 RI0R、RDT0R、RDL0R、RDH0R 寄存器中，直到 RF0R.RFOM 清零。

接收 FIFO 状态

RFR.FMP: 报文的数量。0 表示该 FIFO 中没有报文，3 表示该 FIFO 已满。

RFR.FULL: 当该 FIFO 已存储 3 条报文时，此标志位置位。

RFR.OVR: 当该 FIFO 已存储 3 条报文时，若又收到新的报文，则此标志位置位。

接收报文的步骤

第 1 步: 检查接收 FIFO 中的报文数量。

第 2 步: 将 RFR.RFOM 置位。

第 3 步: 等到 RFOM 清零后，即可读取 RIR、RDTR、RDLR 以及 RDHR。

3.5 过滤功能

bxCAN 模块可以接收来自 CAN 总线的报文。若本节点需要某条报文，即可将其复制到接收 FIFO 中；否则可直接丢弃，无需软件干涉。

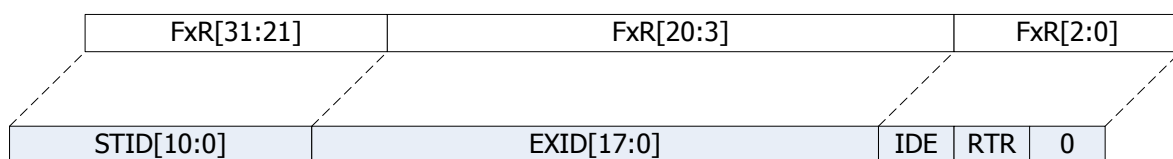
来自 CAN 总线的报文标识符参与过滤的匹配校验。

过滤器包含 14 个过滤槽：过滤槽 0~过滤槽 13。每个过滤槽包含两个 32 位寄存器，分别是 FxR0 以及 FxR1（其中 x = 0~13）。

每个过滤槽可以配置成 32 位或 16 位。

32 位：STDID[10:0]、EXTID[17:0]、IDE 及 RTR 位，如下图所示。

图 143 32 位过滤



16 位：STDID[10:0]、RTR、IDE 及 EXTID[17:15]位，如下图所示。

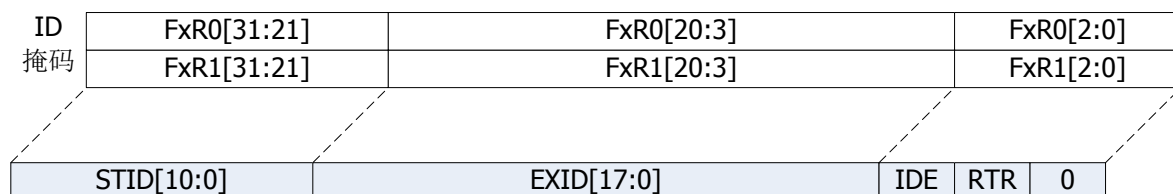
图 144 16 位过滤



掩码模式

在掩码模式中，标识符寄存器与掩码寄存器组合工作，由掩码寄存器决定标识符中的哪些位参与判定（“必须匹配”或“无关”）。下图是 32 位掩码模式的示意图：

图 145 32 位掩码过滤

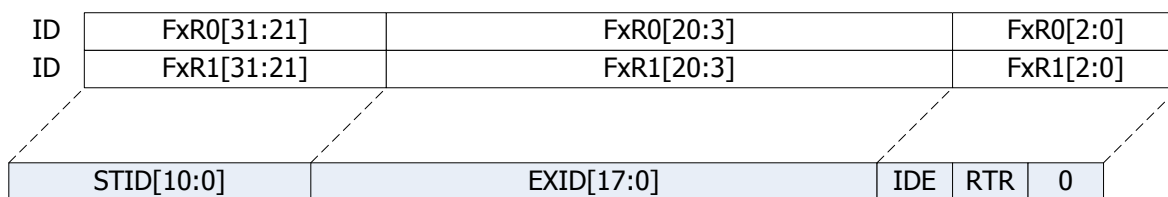


列表模式

过滤器包含的内容都是报文标识符。过滤条目本身即可决定报文是否被丢弃。当收到一条报文后，过滤器将判定哪一条目可匹配该报文的标识符。

下图是 32 位列表模式的示意图。

图 146 32 位掩码过滤



过滤条目编号

每个过滤槽中的过滤条目将按照从 0 到最大值进行累加编号，具体编号则取决于模式和各个过滤槽的大小。

例如，有两个过滤槽，其中过滤槽 0 配置为 32 位掩码模式，过滤槽 1 配置为 16 位列表模式。那么过滤器编号将如下表所示：

表 56 32 位列表模式过滤条目编号

过滤槽	过滤寄存器	过滤条目编号
0	F0R0: 32 位 ID	0
	F0R1: 32 位掩码	
1	F1R0: 32 位 ID	1
	F1R1: 32 位 ID	2

关联 FIFO

14 个过滤槽可分别关联到 FIFO 0 或 FIFO 1。假如某个过滤槽关联到了 FIFO 0，那么透过该过滤槽、符合存储条件的报文将写入到 FIFO 0 中。

激活

假如应用程序希望某个过滤槽工作，则应当激活该过滤槽。应用程序未用到的过滤槽应保持未激活状态。

匹配序号

每个编号的过滤条目都匹配一个过滤规则。当来自 CAN 总线的报文透过某个过滤条目时，其过滤条目编号必须与该报文关联。过滤条目编号称为匹配索引。当被应用程序读取时，该索引保存在 RDTR.FMI 位域中。

每个 FIFO 对与其关联的过滤槽内的过滤条目编号，不管该过滤槽是否已被激活。

下表即为匹配索引的示例。

表 57 匹配索引

过滤槽	FIFO 0	是否激活	过滤条目序号
0	F0R0: 32 位 ID	是	0
	F0R1: 32 位掩码		
1	F1R0: 32 位 ID	是	1
	F1R1: 32 位 ID		2
3	F3R0[15:0]: 16 位 ID	否	3
	F3R0[31:16]: 16 位掩码		
	F3R1[15:0]: 16 位 ID		4
	F3R1[31:16]: 16 位掩码		
7	F7R0[15:0]: 16 位 ID	否	5
	F7R0[31:16]: 16 位 ID		6
	F7R1[15:0]: 16 位 ID		7
	F7R1[31:16]: 16 位 ID		8
8	F8R0[15:0]: 16 位 ID	是	9
	F8R0[31:16]: 16 位 ID		10
	F8R1[15:0]: 16 位 ID		11
	F8R1[31:16]: 16 位 ID		12
9	F9R0[15:0]: 16 位 ID	是	13
	F9R0[31:16]: 16 位掩码		
	F9R1[15:0]: 16 位 ID		14
	F9R1[31:16]: 16 位掩码		
12	F12R0: 32 位 ID	是	15
	F12R1: 32 位掩码		

过滤槽	FIFO 1	是否激活	过滤条目序号
2	F2R0[15:0]: 16 位 ID	是	0
	F2R0[31:16]: 16 位掩码		
	F2R1[15:0]: 16 位 ID		1
F2R1[31:16]: 16 位掩码			
4	F4R0: 32 位 ID	否	2
	F4R1: 32 位掩码		
5	F5R0: 32 位 ID	否	3
	F5R1: 32 位 ID		4
6	F6R0[15:0]: 16 位 ID	是	5
	F6R0[31:16]: 16 位 ID		6
	F6R1[15:0]: 16 位 ID		7
	F6R1[31:16]: 16 位 ID		8
10	F10R0[15:0]: 16 位 ID	否	9
	F10R0[31:16]: 16 位掩码		
	F10R1[15:0]: 16 位 ID		10
	F10R1[31:16]: 16 位掩码		
11	F11R0[15:0]: 16 位 ID	否	11
	F11R0[31:16]: 16 位 ID		12
	F11R1[15:0]: 16 位 ID		13
	F11R1[31:16]: 16 位 ID		14
13	F13R0: 32 位 ID	是	15
	F13R1: 32 位 ID		16

优先级

过滤条目是具有优先级的：

1. 32 位的过滤器比 16 位的优先级高；
2. 列表模式优于掩码模式；
3. 过滤编号越小，优先级越高。

3.6 时间触发通讯

时间触发的 CAN 协议是基于 CAN 数据链路层的一种上层协议。时间触发通讯意味着总线活动是按照时间片的流逝而触发的。在时间触发通讯的系统中，所有报文发送的时间点都是在系统开发阶段就已经定义好的。时间触发通讯的系统非常适合那些数据传输具有周期性特点的应用场合。

在这种工作模式下，CAN 硬件提供一个 16 位内部计数器，该计数器将激活用于产生时间戳，时间戳值对接收和发送而言分别保存在 RDTR 以及 TDTR 寄存器中。内部计数器在每个 CAN 位时间累加。在发送/接收过程中，内部计数值会在 SOF（Start of Frame，帧起始）采样点捕获。

请注意，在时间触发 CAN 通讯中，CAN 报文是不具自动重发机制的。

3.7 通讯参数

非自动重发模式

该模式主要针对 CAN 标准中的时间触发 CAN 通讯而设计。将 MCR 寄存器的 NART 置位，即可配置为这种模式。

在这种模式下，每次报文发送将只尝试一次。只要首次发送尝试失败，不管是因为仲裁丢失还是因为总线错误，硬件都不再重复发送此报文。

首次发送尝试后，硬件即认为请求已完成，因此将 TSR 寄存器的 RQCP 置位。发送的实际结果可通过 TSR 寄存器的 TXOK、ALST 及 TERR 位判断。

位定时

在位这个级别，CAN 协议是采用同步的位传输方式的。这样可以提高发送能力，但也意味着需要更加复杂的位同步机制。在基于字符的传输（异步通信）方式中，一般通过接收各字符的起始位来实现同步；而对于同步传输协议，每帧中仅在帧开始位置有一个起始位。为了确保接收方能够正确接收到完整的报文，需要持续不断地重新同步。为此 CAN 在位时间内、采样点的前后分别设置了相位缓冲段（Phase Buffer Segments）。

CAN 协议规定总线访问是按照位级别进行仲裁的。信号从发送方到接收方再返回到发送方的信号传输延迟，绝对不得超过单个位时间。出于改善同步性能的考虑，除相位缓冲段外，CAN 还在单个位时间内添加了一个传输延迟段（Propagation Delay Segment）。传输延迟段主要用于补偿总线上的信号传播延时，以及收发节点所引入的信号延迟（例如光耦、保护器件分布电容等）。

遵从 CAN 协议并经过 bxCAN 简化后的额定位时间可分为以下三段：

同步段（SYNC_SEG）：该时间段内是有可能发生位状态变化的。同步段的长度固定为 1 个时间分量。

位时段 1（BS1）：定义采样点在位时间中的位置。BS1 包含了 CAN 标准中的传输延迟段以及相位缓冲段 1 两个部分，其长度在 1~16 个时间分量。不过此时间段有可能自动延长，以补偿网络中不同节点间频率差异所引入的正向的相位漂移。

位时段 2（BS2）：定义发送点的位置。BS2 包含了 CAN 标准中的相位缓冲段 2，其长度在 1~8 个时间分量。不过此时间段有可能自动缩短，以补偿负向的相位漂移。

位时间的组成如下图所示。

图 147 位时间的组成



再同步跳转宽度 (SJW): 定义位时间段伸长或缩短的上限, 其范围为 1~4 个时间分量。

所谓有效沿, 是指对自身并不发送隐形位的控制器而言, 在位时间内首个从显性电平跳变到隐性电平的跳变沿。

假如不是在同步段而是在位时段 1 中检测到了有效沿, 那么位时段 1 将适当延长, 最多达到再同步跳转宽度, 使得采样点后移。

反之, 假如不是在同步段而是在位时段 2 中检测到了有效沿, 那么位时段 2 将适当缩短, 最多达到再同步跳转宽度, 使得发送点前移。

波特率

bxCAN 的时钟是从 APB1 总线获得的。bxCAN 模块的波特率计算公式如下:

$$\text{波特率} = \frac{1}{\text{额定位时间}}$$

$$\text{额定位时间} = t_{\text{SYNC_SEG}} + t_{\text{BS1}} + t_{\text{BS2}}$$

其中:

$$t_{\text{SYNC_SEG}} = 1 \times t_q$$

$$t_{\text{BS1}} = (1 + \text{BTR.TS1}) \times t_q$$

$$t_{\text{BS2}} = (1 + \text{BTR.TS2}) \times t_q$$

$$t_q = (1 + \text{BTR.BRP}) \times t_{\text{pCLK1}}$$

3.8 错误标志

按照 CAN 协议的要求, 错误管理是完全由硬件施行的, 其外在表现为一个发送错误计数器 (ESR 寄存器中的 TEC) 以及一个接收错误计数器 (ESR 寄存器中的 REC), 二者都会根据错误状态自动增减。关于 TEC、REC 管理的详细信息, 请参阅 CAN 标准。

TEC 域和 REC 域都可以通过软件读取, 以确定网络的稳定性。

此外, CAN 硬件在 ESR 寄存器中提供了关于当前错误状态的详细信息。通过 IER 寄存器 (ERRIE 等位), 软件能够以非常灵活的方式配置错误检测时是否产生中断。

离线恢复

当 TEC 的数值大于 255 后，节点将进入离线状态，该状态可通过 ESR 寄存器的 BOFF 标志位体现。在离线状态下，bxCAN 模块将不再发送或接收报文。

取决于 MCR 寄存器的 ABOM 位的设置，bxCAN 模块经硬件自动或由软件请求，可从离线状态中恢复（再次成为主动错误状态）。不管哪种情况，bxCAN 模块都必须至少等待 CAN 标准（在 CANRX 管脚上连续检测到 128 个序列，每序列连续 11 个隐形位）的恢复序列。

若 ABOM 置位，那么 bxCAN 进入离线模式后将自动开始恢复序列。

若 ABOM 清零，那么软件必须自行初始化恢复序列，方法是先将 bxCAN 模块进入初始化模式、接下来退出初始化模式。

3.9 bxCAN 中断

bxCAN 模块分配有 4 个中断向量。每个中断源都可以通过 CAN 中断使能寄存器（CAN_IER）分别使能或禁用。

中断来源可以分为以下四类：

- 发送中断
- FIFO 0 中断
- FIFO 1 中断
- 错误及状态变化中断

发送中断

发送中断可由以下事件产生：

- 发送邮箱 0 空，TSR 寄存器中的 RQCP0 置位
- 发送邮箱 1 空，TSR 寄存器中的 RQCP1 置位
- 发送邮箱 2 空，TSR 寄存器中的 RQCP2 置位

FIFO 0 中断

FIFO 0 中断可由以下事件产生：

- 新收到一条报文，RF0R 寄存器中的 FMP0 位域不再是 ‘00’
- FIFO 0 满，RF0R 寄存器中的 FULL0 标志置位
- FIFO 0 过载，RF0R 寄存器中的 FOVR0 标志置位

FIFO 1 中断

FIFO 1 中断可由以下事件产生：

- 新收到一条报文，RF1R 寄存器中的 FMP 位域不再是 ‘00’
- FIFO 1 满，RF1R 寄存器中的 FULL1 标志置位
- FIFO 1 过载，RF1R 寄存器中的 FOVR1 标志置位

错误及状态变化中断

错误及状态变化中断可由以下事件产生：

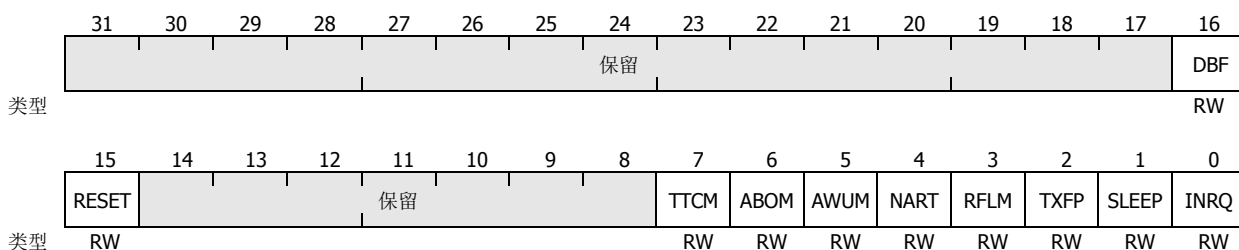
- 出错，详见 CAN 错误状态寄存器（ESR）的介绍
- 唤醒事件，CAN RX 管脚上检测到了 SOF
- 进入休眠模式

4 bxCAN 寄存器

4.1 CAN 主机控制寄存器（CAN_MCR）

地址偏移：0x00

复位值：0x0001 0002



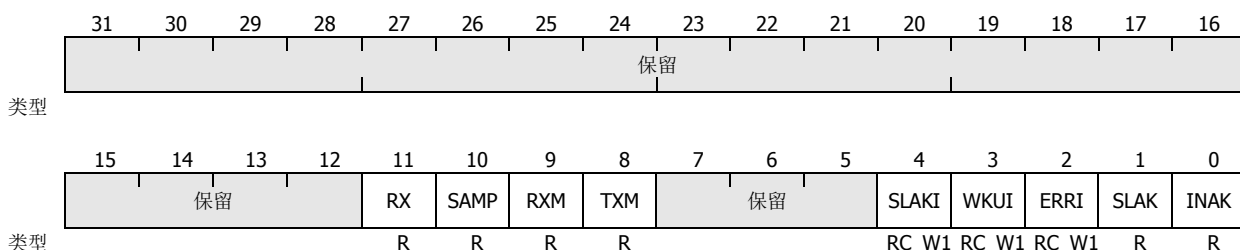
位/位域	名称	描述
31:17	保留	请保持其复位状态
16	DBF	调试冻结 0: 调试期间，CAN 正常工作 1: 调试器件，CAN 的收发均冻结。接收 FIFO 仍然可以正常访问/控制。
15	RESET	bxCAN 模块软件复位 0: 正常工作 1: 强制 bxCAN 模块复位->复位后从休眠模式激活（FMP 位域以及 CAN_MCR 寄存器将恢复到复位值）。本位会自动复位为 0。
14:8	保留	请保持其复位状态
7	TTCM	时间触发通信模式 0: 禁用时间触发通信模式 1: 使能时间触发通信模式
6	ABOM	自动离线管理 本位用于控制 CAN 硬件对于退出离线状态的行为。 0: 基于软件方式退出离线状态。当检测到 128 次 11 个隐形位的序列后，若软件将 MCR 寄存器的 INRQ 先写 1 再写 0，则将退出离线状态。 1: 基于硬件方式退出离线状态。只要检测到 128 次 11 个隐形位的序列后，就退出离线状态。

位/位域	名称	描述
5	AWUM	自动唤醒模式 本位用于控制 CAN 硬件在休眠模式下接收报文的行为。 0: 基于软件方式退出休眠模式。软件将 MCR 寄存器的 SLEEP 位清零, 即可退出休眠模式。 1: 基于硬件方式退出休眠模式。只要检测到 CAN 报文就退出休眠模式。此时 CAN_MCR 寄存器中的 SLEEP 位以及 CAN_MSR 寄存器中的 SLAK 位由硬件清零。
4	NART	关闭自动重发 0: CAN 硬件自动重发报文, 直到报文 (按 CAN 标准来说) 已成功发送完成。 1: 报文只发送一次, 与发送结果 (成功、错误或仲裁失败) 无关。
3	RFLM	接收 FIFO 锁定模式 0: 接收 FIFO 在过载时不锁定, 新收到的报文将会覆盖前一条报文。 1: 接收 FIFO 在过载时锁定, 新收到的报文将会被丢弃。
2	TXFP	发送 FIFO 优先级 本位用于控制当多个邮箱同时处于挂起态时的发送顺序。 0: 按报文标识符决定优先级。 1: 按请求顺序 (FIFO) 决定优先级。
1	SLEEP	休眠请求 本位由软件置位, 请求 CAN 模块进入休眠模式。当前 CAN 活动 (发送或接收一条 CAN 报文) 结束之后, 将会立即进入休眠模式。 软件将本位清零, 即可退出休眠模式。 当 AWUM=1 时, 如果 CAN RX 管脚检测到了 SOF, 则本位将被硬件自动清零。系统复位后, 本位默认为 1 (CAN 初始化为休眠模式)。
0	INRQ	初始化请求 本位由软件清零, 之后 CAN 才会进入正常工作模式。当硬件检测到 CAN RX 管脚上出现了连续 11 个隐形位后, 硬件即根据其进行同步, 并准备进行收发。此时硬件会将 MSR 寄存器的 INAK 标志置位。 若软件向本位写 1, 即请求 CAN 硬件进入初始化模式。此时 CAN 硬件将等到下一个 CAN 活动 (发送或接收) 完成后才会进入初始化模式。此时硬件会将 MSR 寄存器的 INAK 标志置位。

4.2 CAN 主机状态寄存器 (CAN_MSR)

地址偏移: 0x04

复位值 : 0x0000 0C02



位/位域	名称	描述
31:12	保留	请保持其复位状态

位/位域	名称	描述
11	RX	CAN RX 信号 本位代表 RX 管脚上的当前状态。
10	SAMP	上一采样点 上一个采样点时的 RX 状态（即当前已接收的位的值）。
9	RXM	接收模式 CAN 硬件当前工作在接收状态。
8	TXM	发送模式 CAN 硬件当前工作在发送状态。
7:5	保留	请保持其复位状态
4	SLAKI	休眠确认中断 当 SLAKE=1 时，本位由硬件置位，表明 bxCAN 进入休眠模式。若 IER 寄存器中的 SLKIE=1，则还会产生一个状态变化中断。 本位可由软件清零，或当 SLAK 清零时由硬件自动清零。 请注意： 当 SLKIE=0 时，SLAKI 不会进行查询。此时应用软件应查询 SLAK 的状态。
3	WKUI	唤醒中断 若 CAN 硬件处于休眠模式下，则当检测到 SOF 时，本位将被硬件置位。若 IER 寄存器中的 WKUIE=1，则还会产生一个状态变化中断。 本位由软件清零。
2	ERRI	错误中断 若 ESR 寄存器中的某个标志置位、并且 IER 寄存器中对应的中断使能，那么本位将被硬件置位。若 IER 寄存器中的 ERRIE=1，则还会产生一个状态变化中断。 本位由软件清零。
1	SLAK	休眠确认 若硬件将本位置位，表明 CAN 硬件正处于休眠模式。本位用于确认来自软件的休眠请求（对 MCR 寄存器的 SLEEP 写 1）。 若硬件将本位清零，表明 CAN 硬件已退出休眠模式（将与 CAN 总线同步）。要实现同步，硬件需要在 CAN RX 管脚上检测到连续 11 个隐形位的序列。 请注意： 退出休眠模式的序列是在 MCR 寄存器中的 SLEEP 清零时触发的。关于清零 SLEEP 的相关说明，请参见 MCR 寄存器的 AWUM 位的描述。
0	INAK	初始化确认 若硬件将本位置位，表明 CAN 硬件正处于初始化模式。本位用于确认来自软件的初始化请求（对 MCR 寄存器的 INRQ 写 1）。 若硬件将本位清零，表明 CAN 硬件已退出初始化模式（将与 CAN 总线同步）。要实现同步，硬件需要在 CAN RX 管脚上检测到连续 11 个隐形位的序列。

4.3 CAN 发送状态寄存器 (CAN_TSR)

地址偏移：0x08

复位值：0x1C00 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]	ABRQ2		保留		TERR2	ALST2	TXOK2	RQCP2	
类型	R	R	R	R	R	R	R	R	RS			RC_W1	RC_W1	RC_W1	RC_W1	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ABRQ1	保留	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	保留	TERR0	ALST0	TXOK0	RQCP0
类型 RS		RC_W1	RC_W1	RC_W1	RC_W1	RS		RC_W1	RC_W1	RC_W1	RC_W1

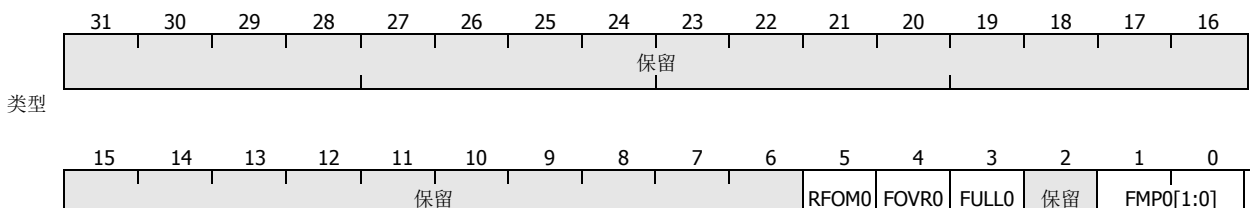
位/位域	名称	描述
31	LOW2	邮箱 2 最低优先级标志 如果不止一个邮箱挂起发送请求，并且邮箱 2 的优先级最低，则本位将被硬件置位。
30	LOW1	邮箱 1 最低优先级标志 如果不止一个邮箱挂起发送请求，并且邮箱 1 的优先级最低，则本位将被硬件置位。
29	LOW0	邮箱 0 最低优先级标志 如果不止一个邮箱挂起发送请求，并且邮箱 0 的优先级最低，则本位将被硬件置位。 请注意： 若只有一个邮箱挂起发送请求，则 LOW[2:0]位域为 0。
28	TME2	邮箱 2 无发送标志 如果邮箱 2 中没有挂起的发送请求，则本位将被硬件置位。
27	TME1	邮箱 1 无发送标志 如果邮箱 1 中没有挂起的发送请求，则本位将被硬件置位。
26	TME0	邮箱 0 无发送标志 如果邮箱 0 中没有挂起的发送请求，则本位将被硬件置位。
25:24	CODE[1:0]	邮箱编号 如果至少有一个邮箱空闲，那么此编号就是下一个可发送的邮箱编号。 如果所有邮箱都挂起发送请求，那么此代码就是优先级最低的邮箱编号。
23	ABRQ2	中止邮箱 2 请求 若软件对本位写 1，则将中止邮箱 2 的发送请求。 当邮箱空闲（发送完成）后，本位将被硬件清零。 如果邮箱并未挂起发送请求，那么对本位写 1 无任何作用。
22:20	保留	请保持其复位状态
19	TERR2	邮箱 2 发送错误 当前一次发送因出错而失败时，本标志将会置位。
18	ALST2	邮箱 2 仲裁失败 当前一次发送因仲裁失败而失败时，本标志将会置位。
17	TXOK2	邮箱 2 发送成功 每次发送后硬件都会更新本标志。 0: 前一次发送失败 1: 前一次发送成功 当邮箱 2 的发送请求被成功执行后，本位将由硬件置位。
16	RQCP2	邮箱 2 请求完成 当上一次请求（发送或中止）被成功执行后，本位将由硬件置位。 软件对本位写 1，即可将本位清零；有新的发送请求（TI2R 寄存器中的 TXRQ2 置位），本位将被硬件清零。 清零本位后，邮箱 2 的所有状态位（TXOK2、ALST2 及 TERR2）都清零。
15	ABRQ1	中止邮箱 1 请求 若软件对本位写 1，则将中止邮箱 1 的发送请求。 当邮箱空闲（发送完成）后，本位将被硬件清零。 如果邮箱并未挂起发送请求，那么对本位写 1 无任何作用。

位/位域	名称	描述
14:12	保留	请保持其复位状态
11	TERR1	邮箱 1 发送错误 当前一次发送因出错而失败时，本标志将会置位。
10	ALST1	邮箱 1 仲裁失败 当前一次发送因仲裁失败而失败时，本标志将会置位。
9	TXOK1	邮箱 1 发送成功 每次发送后硬件都会更新本标志。 0: 前一次发送失败 1: 前一次发送成功 当邮箱 1 的发送请求被成功执行后，本位将由硬件置位。
8	RQCP1	邮箱 1 请求完成 当上一次请求（发送或中止）被成功执行后，本位将由硬件置位。 软件对本位写 1，即可将本位清零；有新的发送请求（TI1R 寄存器中的 TXRQ1 置位），本位将被硬件清零。 清零本位后，邮箱 1 的所有状态位（TXOK1、ALST1 及 TERR1）都清零。
7	ABRQ0	中止邮箱 0 请求 若软件对本位写 1，则将中止邮箱 0 的发送请求。 当邮箱空闲（发送完成）后，本位将被硬件清零。 如果邮箱并未挂起发送请求，那么对本位写 1 无任何作用。
6:4	保留	请保持其复位状态
3	TERR0	邮箱 0 发送错误 当前一次发送因出错而失败时，本标志将会置位。
2	ALST0	邮箱 0 仲裁失败 当前一次发送因仲裁失败而失败时，本标志将会置位。
1	TXOK0	邮箱 0 发送成功 每次发送后硬件都会更新本标志。 0: 前一次发送失败 1: 前一次发送成功 当邮箱 0 的发送请求被成功执行后，本位将由硬件置位。
0	RQCP0	邮箱 0 请求完成 当上一次请求（发送或中止）被成功执行后，本位将由硬件置位。 软件对本位写 1，即可将本位清零；有新的发送请求（TI0R 寄存器中的 TXRQ0 置位），本位将被硬件清零。 清零本位后，邮箱 1 的所有状态位（TXOK0、ALST0 及 TERR0）都清零。

4.4 CAN 接收 FIFO 0 寄存器（CAN_RF0R）

地址偏移：0x0C

复位值：0x0000 0000

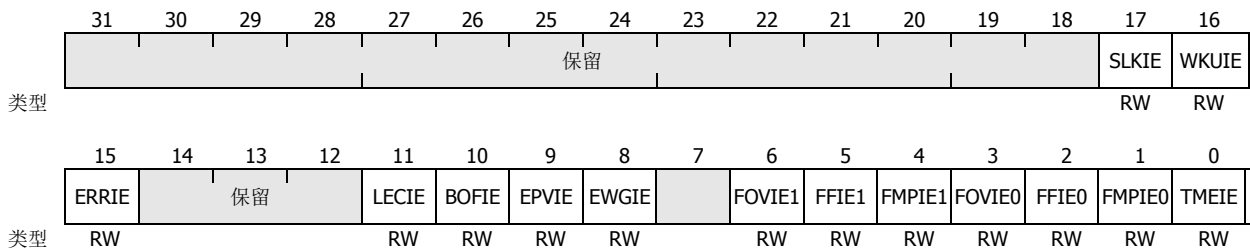


位/位域	名称	描述
4	FOVR1	FIFO 1 过载 若 FIFO 已满，并且又有一条报文接收完成且符合过滤条件，本位将被硬件置位。 本位由软件清零。
3	FULL1	FIFO 1 满 当 FIFO 中已存储 3 条报文时，本位将被硬件置位。 本位由软件清零。
2	保留	请保持其复位状态
1:0	FMP1[1:0]	FIFO 1 报文挂起 本位域表明接收 FIFO 中挂起几条报文。 FIFO 中每存入新的一条报文，FMP 就会加 1。每当软件释放输出邮箱（清零 RFOM1），FMP 就会减 1。

4.6 CAN 中断使能寄存器 (CAN_IER)

地址偏移：0x14

复位值：0x0000 0000



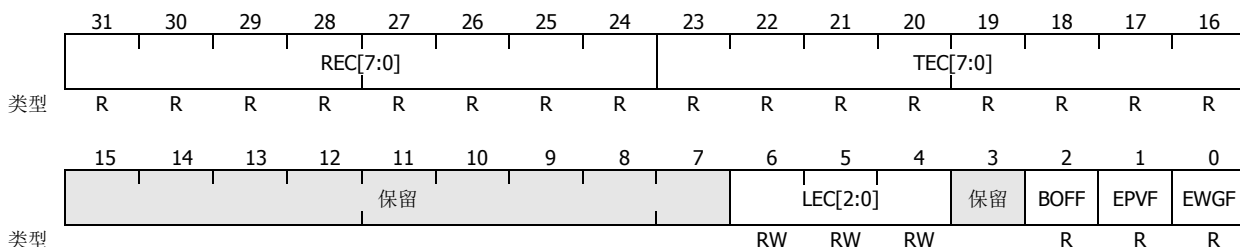
位/位域	名称	描述
31:18	保留	请保持其复位状态
17	SLKIE	休眠中断使能 0: 当 SLAKI 置位时，不产生中断 1: 当 SLAKI 置位时，产生中断
16	WKUIE	唤醒中断使能 0: 当 WKUI 置位时，不产生中断 1: 当 WKUI 置位时，产生中断
15	ERRIE	错误中断使能 0: 当 ESR 寄存器中有错误标志挂起时，不产生中断 1: 当 ESR 寄存器中有错误标志挂起时，产生中断
14:12	保留	请保持其复位状态
11	LECIE	上一错误码中断使能 0: 当硬件设置 LEC[2:0]中的内容时，不置位 ERRI 标志 1: 当硬件设置 LEC[2:0]中的内容时，置位 ERRI 标志
10	BOFIE	离线中断使能 0: 当 BOFF 置位时，不置位 ERRI 标志 1: 当 BOFF 置位时，置位 ERRI 标志

位/位域	名称	描述
9	EPVIE	被动错误中断使能 0: 当 EPVF 置位时, 不置位 ERRI 标志 1: 当 EPVF 置位时, 置位 ERRI 标志
8	EWGIE	错误警告中断使能 0: 当 EWGF 置位时, 不置位 ERRI 标志 1: 当 EWGF 置位时, 置位 ERRI 标志
7	保留	请保持其复位状态
6	FOVIE1	FIFO 过载中断使能 0: 当 FOVR 置位时, 不产生中断 1: 当 FOVR 置位时, 产生中断
5	FFIE1	FIFO 满中断使能 0: 当 FULL 置位时, 不产生中断 1: 当 FULL 置位时, 产生中断
4	FMPIE1	FIFO 报文挂起中断使能 0: 当 FMP[1:0]位域并非 00b 时, 不产生中断 1: 当 FMP[1:0]位域并非 00b 时, 产生中断
3	FOVIE0	FIFO 过载中断使能 0: 当 FOVR 置位时, 不产生中断 1: 当 FOVR 置位时, 产生中断
2	FFIE0	FIFO 满中断使能 0: 当 FULL 置位时, 不产生中断 1: 当 FULL 置位时, 产生中断
1	FMPIE0	FIFO 报文挂起中断使能 0: 当 FMP[1:0]位域并非 00b 时, 不产生中断 1: 当 FMP[1:0]位域并非 00b 时, 产生中断
0	TIMIE	发送报文空中断使能 0: 当 RQCPx 置位时, 不产生中断 1: 当 RQCPx 置位时, 产生中断

4.7 CAN 错误状态寄存器 (CAN_ESR)

地址偏移: 0x18

复位值 : 0x0000 0000



位/位域	名称	描述
------	----	----

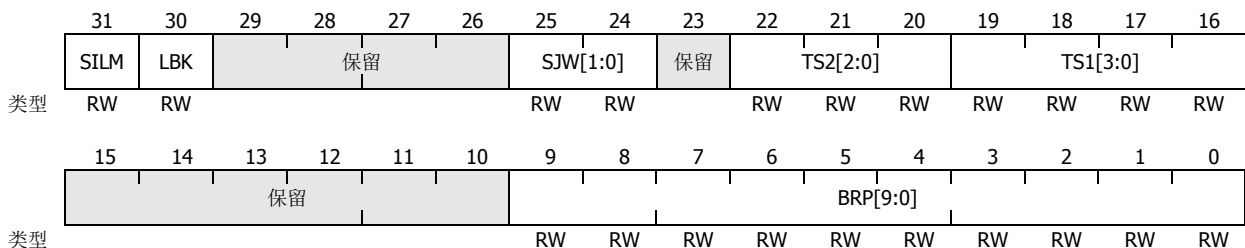
位/位域	名称	描述
31:24	REC[7:0]	接收错误计数器 CAN 协议故障确认机制的实现。一旦接收期间出现错误，取决于 CAN 标准所定义的出错状况，本计数器将加 1 或加 8。在每次成功接收之后，本计数器将减 1，或数值大于 128 时复位到 120。当计数值大于 127 后，CAN 控制器将进入被动错误状态。
23:16	TEC[7:0]	9 位发送错误计数器的低字节 CAN 协议故障确认机制的实现。
15:7	保留	请保持其复位状态
6:4	LEC[2:0]	上次错误代码 本位域由硬件写入，其保存的代码表明在 CAN 总线上检测的上次错误状况。假如一条报文已经传输（接收或发送）完成而没有错误，那么此位域将会清零。 LEC[2:0]位域可由软件置为 0b111。本位域由硬件更新，表明当前的通信状态。 000: 无错误 001: 位填充错误 010: 帧格式错误 011: 确认位错误 100: 位隐性错误 101: 位显性错误 110: CRC 校验错误 111: 软件设置
3	保留	请保持其复位状态
2	BOFF	离线标志 当 CAN 硬件进入离线状态后，本位将由硬件置位。当 TEC 溢出（大于 255）时，将会进入离线状态。
1	EPVF	被动错误标志 当到达被动错误门限（接收错误计数器或发送错误计数器>127）后，本位将由硬件置位。
0	EWGF	错误警告标志 当到达警告门限（接收错误计数器或发送错误计数器≥96）后，本位将由硬件置位。

4.8 CAN 位定时寄存器 (CAN_BTR)

地址偏移: 0x1C

复位值 : 0x0123 0000

仅当 CAN 硬件处于初始化模式时，软件才能访问本寄存器。



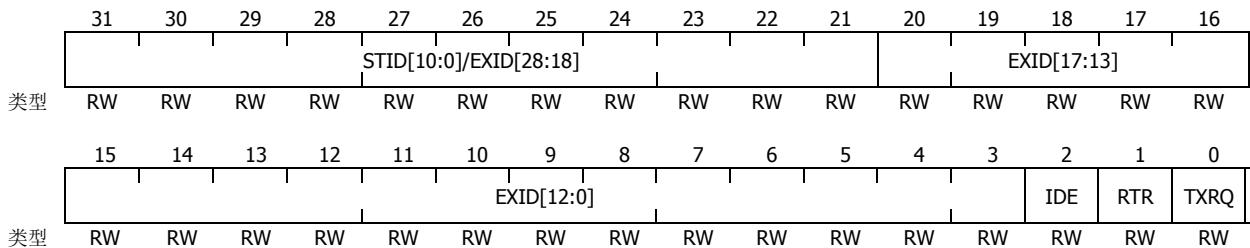
位/位域	名称	描述
31	SILM	静默通信模式 0: 正常工作模式 1: 静默模式
30	LBKM	环回通信模式 0: 禁用环回模式 1: 使能环回模式
29:26	保留	请保持其复位状态
25:24	SJW[1:0]	再同步跳转宽度 CAN 硬件在进行再同步时, 可对位时间进行延长或缩短。本位域定义位时间变化的最大时间份额数目。
23	保留	请保持其复位状态
22:20	TS2[2:0]	位时段 2 本位域定义位时段 2 的时间份额数目。
19:16	TS1[3:0]	位时段 1 本位域定义位时段 1 的时间份额数目。
15:10	保留	请保持其复位状态
9:0	BRP[9:0]	波特率预分频器 本位域定义一个时间份额的长度。

4.9 CAN TX 邮箱标识符寄存器 (CAN_TIxR) (x=0~2)

地址偏移: 0x180、0x190、0x1A0

复位值 : 0xFFFF XXXX (除了第 0 位 TXRQ=0)

当邮箱挂起传输请求时 (TMEx 为 0 时), 所有 TX 寄存器都是写保护的。本寄存器还实现了 TX 请求控制 (第 0 位, 其复位值为 0)。



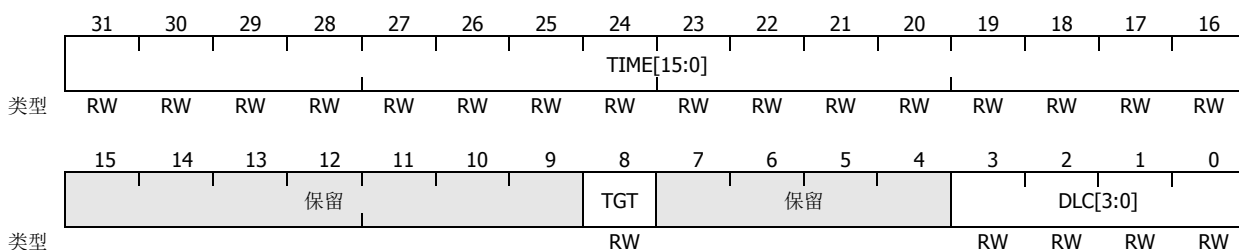
位/位域	名称	描述
31:21	STID[10:0] / EXID[28:18]	标准标识符或扩展标识符 标准标识符, 或扩展标识符的高 11 位 (取决于 IDE 位的值)。
20:3	EXID[17:0]	扩展标识符 扩展标识符的低 18 位。

位/位域	名称	描述
2	IDE	标识符类型 本位定义邮箱中报文的标识符类型。 0: 标准标识符 1: 扩展标识符
1	RTR	远程发送请求 0: 数据帧 1: 远程帧
0	TXRQ	发送邮箱请求 软件对本位写 1, 即可从相应邮箱请求发送。当邮箱清空 (发送完成) 后, 本位由硬件清零。

4.10 CAN 发送邮箱数据长度控制及时间戳寄存器 (CAN_TDTxR) (x=0~2)

地址偏移: 0x184、0x194、0x1A4

复位值 : 0xXXXX XXXX

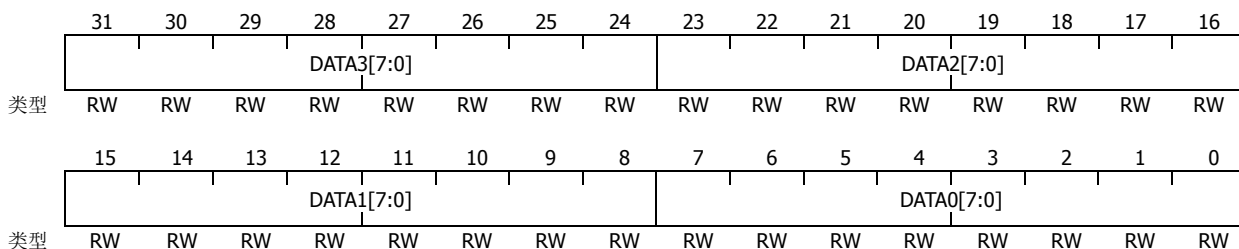


位/位域	名称	描述
31:16	TIME[15:0]	报文时间戳 本位域包含在 SOF 发送时所捕获的 16 位定时器值。
15:9	保留	请保持其复位状态
8	TGT	发送全局时间 只有当硬件处于事件触发通信模式下 (MCR 寄存器中的 TTCM 置位), 本位才有效。 0: 不发送时间戳 TIME[15:0] 1: 时间戳 TIME[15:0] 添加到 8 字节报文的最后 2 个数据字节中: TIME[7:0] 添加到第 7 字节, TIME[15:8] 添加到第 6 字节, 分别替换掉 TDHxR[31:16] 寄存器中的数据 (DATA6[7:0] 和 DATA7[7:0])。DLC 必须编程为 8, 以便时间戳能够正确地在 CAN 总线上传输。
7:4	保留	请保持其复位状态
3:0	DLC[3:0]	数据长度码 本位域定义数据帧或远程帧请求中包含的数据字节数。 根据 DLC 位域的数值, 一条报文可以包含 0 到 8 个数据字节。

4.11 CAN 发送邮箱数据低寄存器 (CAN_TDLxR) (x=0~2)

地址偏移: 0x188、0x198、0x1A8

复位值 : 0XXXXX XXXX

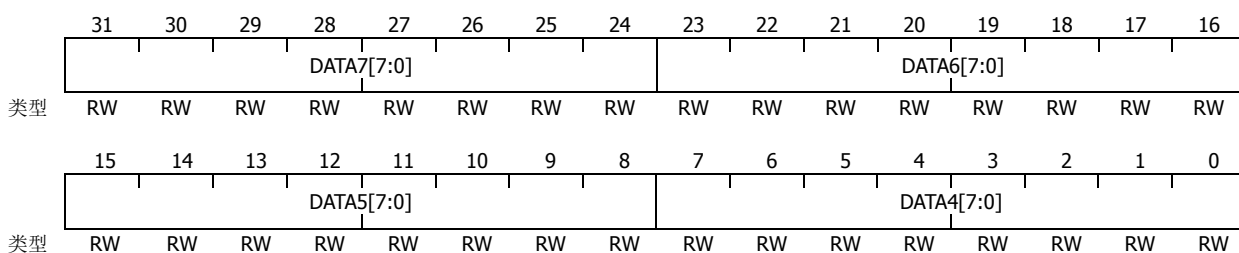


位/位域	名称	描述
31:24	DATA3[7:0]	数据字节 3 报文的数据字节 3
23:16	DATA2[7:0]	数据字节 2 报文的数据字节 2
15:8	DATA1[7:0]	数据字节 1 报文的数据字节 1
7:0	DATA0[7:0]	数据字节 0 报文的数据字节 0 一条报文可以包含 0~8 个数据字节，数据域是从数据字节 0 开始的。

4.12 CAN 发送邮箱数据高寄存器 (CAN_TDHxR) (x=0~2)

地址偏移: 0x18C、0x19C、0x1AC

复位值 : 0XXXXX XXXX



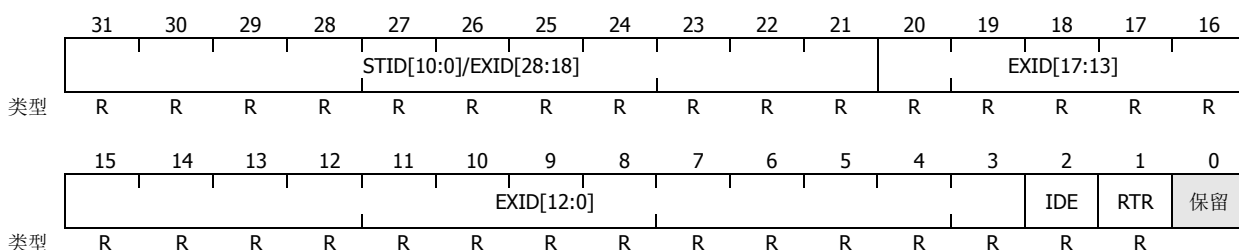
位/位域	名称	描述
31:24	DATA7[7:0]	数据字节 7 报文的数据字节 7 <i>请注意: 假如报文的 TGT、TTCM 有效, 那么 DATA7 和 DATA6 将被自动替换为时间戳。</i>
23:16	DATA6[7:0]	数据字节 6 报文的数据字节 6

位/位域	名称	描述
15:8	DATA5[7:0]	数据字节 5 报文的数据字节 5
7:0	DATA4[7:0]	数据字节 4 报文的数据字节 4

4.13 CAN 接收 FIFO 邮箱标识符寄存器 (CAN_RIxR) (x=0~1)

地址偏移: 0x1B0、0x1C0

复位值 : 0xFFFF FFFF

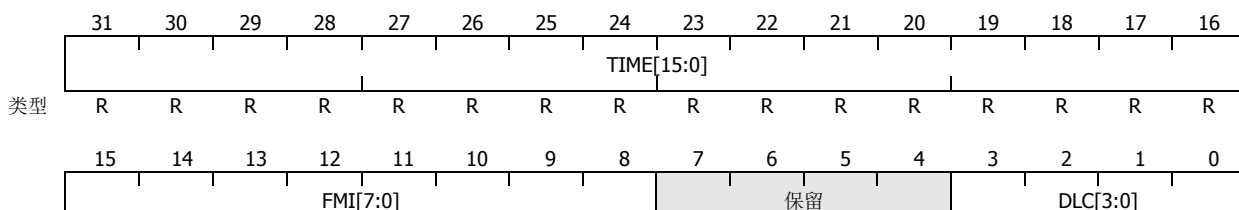


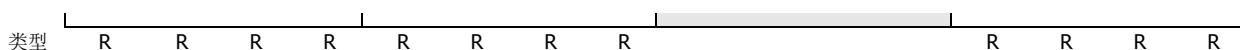
位/位域	名称	描述
31:21	STID[10:0] / EXID[28:18]	标准标识符或扩展标识符 标准标识符, 或扩展标识符的高 11 位 (取决于 IDE 位的值)。
20:3	EXID[17:0]	扩展标识符 扩展标识符的低 18 位。
2	IDE	标识符类型 本位定义邮箱中报文的标识符类型。 0: 标准标识符 1: 扩展标识符
1	RTR	远程发送请求 0: 数据帧 1: 远程帧
0	保留	请保持其复位状态

4.14 CAN 接收 FIFO 邮箱数据长度控制及时间戳寄存器 (CAN_RDTxR) (x=0~1)

地址偏移: 0x1B4、0x1C4

复位值 : 0xFFFF FFFF



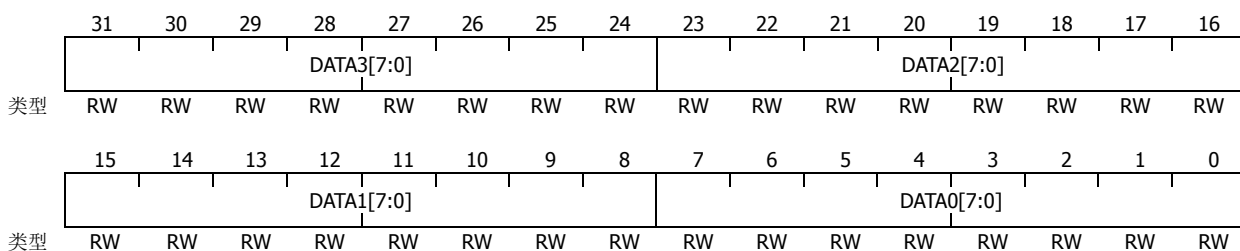


位/位域	名称	描述
31:16	TIME[15:0]	报文时间戳 本位域包含在 SOF 发送时所捕获的 16 位定时器值。
15:8	FMI[7:0]	过滤条件匹配索引 保存在邮箱中的报文符合特定的过滤条件，本寄存器包含该过滤条件索引号。
7:4	保留	请保持其复位状态
3:0	DLC[3:0]	数据长度码 本位域定义数据帧或远程帧请求中包含的数据字节数。 根据 DLC 位域的数值，一条报文可以包含 0 到 8 个数据字节。

4.15 CAN 接收邮箱数据低寄存器 (CAN_RDLxR) (x=0~1)

地址偏移: 0x1B8、0x1C8

复位值 : 0xFFFF XXXX

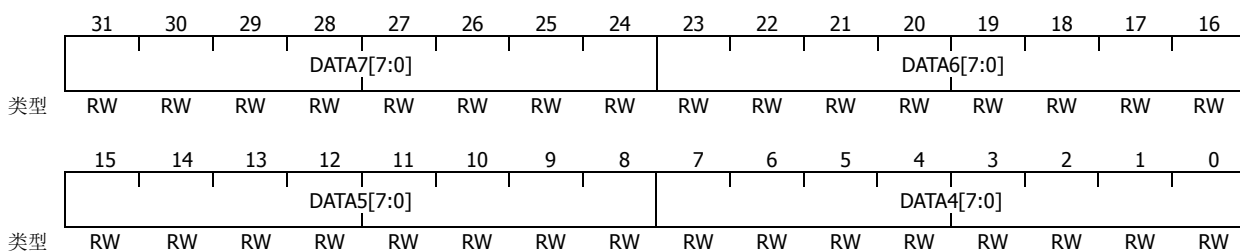


位/位域	名称	描述
31:24	DATA3[7:0]	数据字节 3 报文的数据字节 3
23:16	DATA2[7:0]	数据字节 2 报文的数据字节 2
15:8	DATA1[7:0]	数据字节 1 报文的数据字节 1
7:0	DATA0[7:0]	数据字节 0 报文的数据字节 0 一条报文可以包含 0~8 个数据字节，数据域是从数据字节 0 开始的。

4.16 CAN 接收邮箱数据高寄存器 (CAN_RDHxR) (x=0~1)

地址偏移: 0x1BC、0x1CC

复位值 : 0xFFFF XXXX

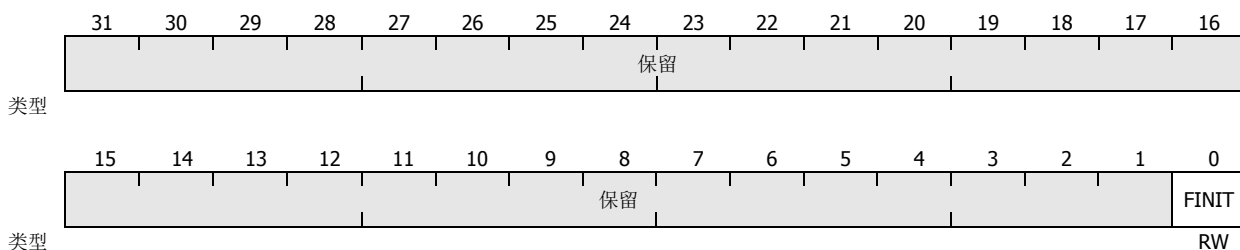


位/位域	名称	描述
31:24	DATA7[7:0]	数据字节 7 报文的数据字节 7
23:16	DATA6[7:0]	数据字节 6 报文的数据字节 6
15:8	DATA5[7:0]	数据字节 5 报文的数据字节 5
7:0	DATA4[7:0]	数据字节 4 报文的数据字节 4

4.17 CAN 过滤主控寄存器 (CAN_FMR)

地址偏移: 0x200

复位值 : 0x2A1C 0E01

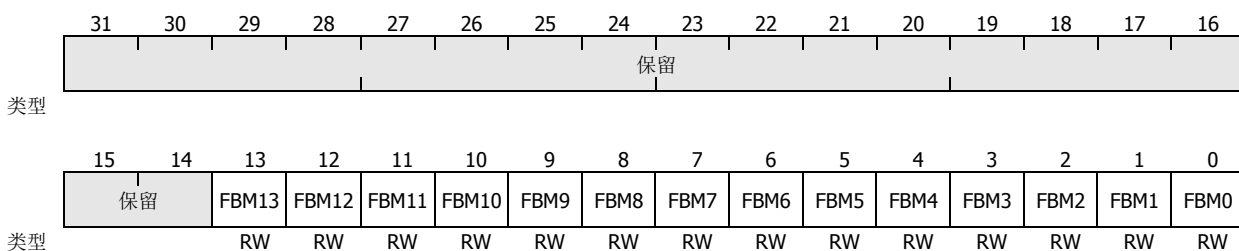


位/位域	名称	描述
31:1	保留	请保持其复位状态
0	FINIT	过滤初始化模式 过滤 bank 的初始化模式。 0: 激活过滤模式 1: 过滤的初始化模式

4.18 CAN 过滤模式寄存器 (CAN_FM1R)

地址偏移: 0x204

复位值 : 0x0000 0000

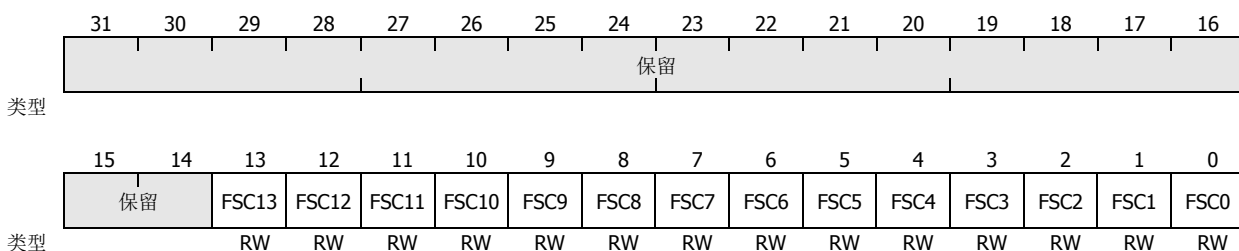


位/位域	名称	描述
31:14	保留	请保持其复位状态
13:0	FBMx	过滤模式 过滤条件 x 的工作模式。 0: 该过滤 bank 的两个 32 位寄存器工作于标识符掩码模式 1: 该过滤 bank 的两个 32 位寄存器工作于标识符列表模式

4.19 CAN 过滤尺寸寄存器 (CAN_FS1R)

地址偏移: 0x20C

复位值 : 0x0000 0000

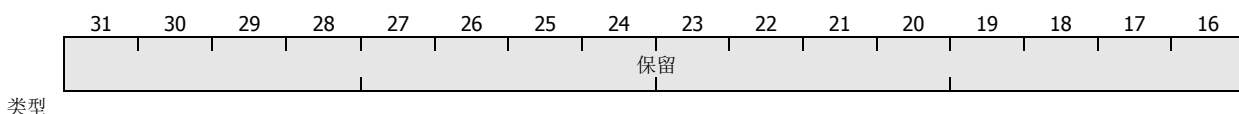


位/位域	名称	描述
31:14	保留	请保持其复位状态
13:0	FSCx	过滤尺寸配置 这些位定义过滤条件 x 的配置。 0: 配置为两个 16 位过滤条件 1: 配置为一个 32 位过滤条件

4.20 CAN 过滤 FIFO 分配寄存器 (CAN_FFA1R)

地址偏移: 0x214

复位值 : 0x0000 0000



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域	名称	描述
31:14	保留	请保持其复位状态
13:0	FFAx	过滤条件 x 的 FIFO 分配 通过相应过滤条件的报文将存储在指定的报文中。 0: 过滤条件分配给 FIFO 0 1: 过滤条件分配给 FIFO 1

4.21 CAN 过滤激活寄存器 (CAN_FA1R)

地址偏移: 0x21C

复位值 : 0x0000 0000

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

类型

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

位/位域	名称	描述
31:14	保留	请保持其复位状态
13:0	FACTx	过滤条件激活 软件对某个位写 1, 即可激活对应的过滤条件。要修改过滤条件 x 寄存器 (CAN_FxR[0:7]), 必须先清零 FACTx, 或将 CAN_FMR 寄存器中的 FINIT 置位。 0: 过滤条件 x 未激活 1: 过滤条件 x 已激活

4.22 过滤槽 i 寄存器 x (CAN_FiRx) (i=0~27, x=1~2)

地址偏移: 0x240~0x31C

复位值 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

类型

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB[15:0]															

310

类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
位/位域	名称		描述														
31:0	FB[31:0]		过滤条件 列表项 此时寄存器的每个位对应于相应标识符的每个位。 0: 预期显性位 1: 预期隐形位 掩码项 此时寄存器的每个位代表的含义是：标识符中的对应位是否参与匹配校验。 0: 无关，该位不参与匹配校验。 1: 该位的内容必须与前一列表项中的对应位一致。														

第16章 外部存储器接口（EXMC）

1 简介

EXMC 就是外部存储器控制器（External Memory Controller）的缩写。本模块用于访问各种片外的同步存储器，例如 SRAM、ROM、NOR Flash 以及 PSRAM 等。本模块划分为若干个子 bank，每个子 bank 都有其独立的片选信号，然而，同一时刻只允许访问一个 bank。EXMC 模块可配置为两到三种不同的访问时序，以满足不同外部存储器的协议要求。

EXMC 也可用于提供两类信号：数据信号或控制信号。所提供的信号能够兼容大多数常用的 LCD 控制器所需的摩托罗拉 6800 时序或英特尔 8080 时序，因此 EXMC 可灵活地支持典型接口的 LCD 模块，从而降低系统成本及复杂性。

2 主要特点

- 支持多种外部存储器：
 - SRAM
 - PSRAM
 - ROM
 - 异步 NOR Flash
- 16 位或 8 位数据总线
- 每个 bank 可分别配置两到三组时序
- 控制信号的建立时间/等待时间/保持时间均可设置
- 对某些器件，可提供写入使能信号以及字节选择信号，例如 RAM
- 当 AHB 总线数据宽度大于外部存储器数据宽度时，可自动分割 EXMC 操作。例如，32 位的 AHB 操作将被分为两次连续的 16 位 EXMC 操作
- 可直接连接支持摩托罗拉 6800 或英特尔 8080 协议的 LCD

3 功能描述

3.1 概述

下图详细描绘出 EXMC 接口的内部配置。

图 148 EXMC 内部框图

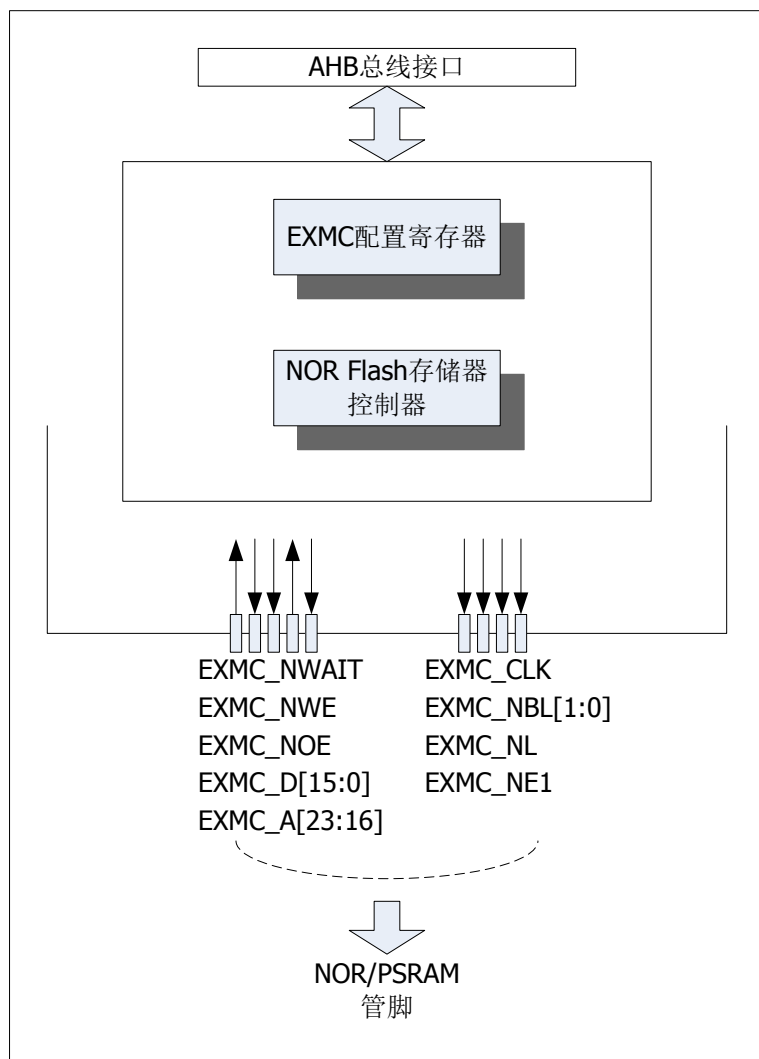
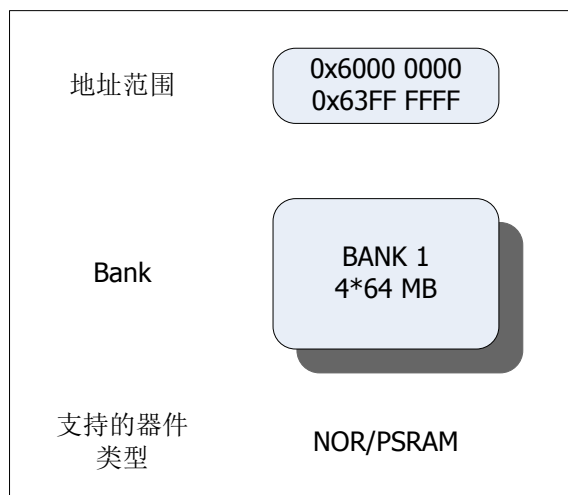


图 149 外部存储器设备地址映射



3.2 EXMC 访问的基本读/写规范

- AHB 总线数据宽度等于外设时，支持读/写访问
- AHB 总线数据宽度小于外设，只支持读访问
- AHB 总线数据宽度小于外设，仅当外设具有字节选择功能（例如 PSRAM、ROM），才支持写访问
- 若 AHB 总线数据宽度大于外设，则读写访问都支持

3.3 出错响应管理

对于以下情况，EXMC 模块将生成一个 AHB 错误：

- EXMC 的某个 bank 并未使能，却对其执行访问
- 并未开启 NOR Flash 的 FACC 功能，却对其执行访问

3.4 NOR Flash/PSRAM 控制器功能描述

EXMC 模块的 bank 1 用于访问 NOR Flash、PSRAM、SRAM、ROM、蜂窝 RAM 等外部存储器。

异步访问外部存储器支持：SRAM、ROM、PSRAM 以及 NOR Flash。

注意： 在异步模式下，所有控制器输出信号在内部 AHB 总线时钟（HCLK）的上升沿改变。
在同步模式下，所有控制器输出数据在外部存储器时钟（EXMC_CLK）的下降沿改变。

3.5 NOR Flash、SRAM、PSRAM 外部存储设备接口功能

表 58 NOR Flash 接口信号功能

EXMC 管脚	方向	模式	功能描述
CLK	输出	同步	时钟信号
A[23:16]	输出	异步	地址总线信号
D[15:0]	输入/输出	异步/复用	地址/数据总线信号
	输入/输出	异步/非复用	数据总线信号
NE1	输出	异步/同步	片选信号
NOE	输出	异步/同步	读操作使能信号
NWE	输出	异步/同步	写操作使能信号
NL	输出	异步/同步	锁存使能信号（存储器信号名 NADV）
NWAIT	输入	异步/同步	等待输入信号

表 59 PSRAM 接口信号功能

EXMC 管脚	方向	模式	功能描述
CLK	输出	同步	时钟信号
A[23:16]	输出	异步/同步	地址总线信号

EXMC 管脚	方向	模式	功能描述
D[15:0]	输入/输出	异步/同步	数据总线信号
NWE	输出	异步/同步	写操作使能信号
NOE	输出	异步/同步	读操作使能信号
NL	输出	异步/同步	锁存使能信号（存储器信号名 NADV）
NBL[1:0]	输出	异步/同步	高/低字节使能
NWAIT	输入	异步/同步	等待输入信号

3.6 对 NOR Flash 以及 PSRAM 不支持的存储器访问模式

表 60 PSRAM 接口信号功能

存储器	访问模式	操作	AHB 会话大小	存储器会话大小
NOR Flash	异步	写	8	16
	异步页访问	读		16
	同步	读	8	16
PSRAM	异步页访问	读		16
	同步	读	8	16

3.7 外部存储器异步访问的时序图及配置示例

下面的内容将介绍异步的 NOR Flash/PSRAM 控制器的会话过程。当设置为扩展模式后，可以将 A、B、C、D 四种模式混合起来，边读边写。

■ 示例 1

模式 1: SRAM

表 61 示例 1 模式 1 相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
5:4	MWID	取决于存储器的参数
3:2	MTYP	取决于存储器的参数
1	MUXEN	0x1
0	MBKEN	0x1
EXMC_BTR 寄存器		
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 150 示例 1 模式 1 写操作时序

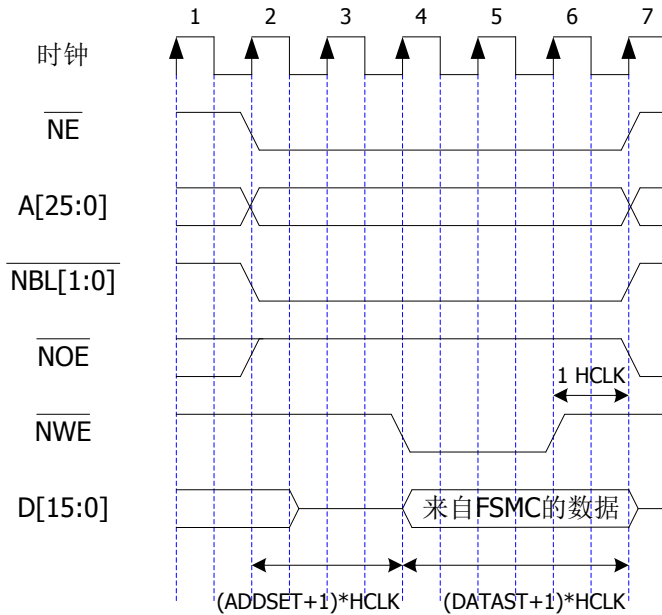
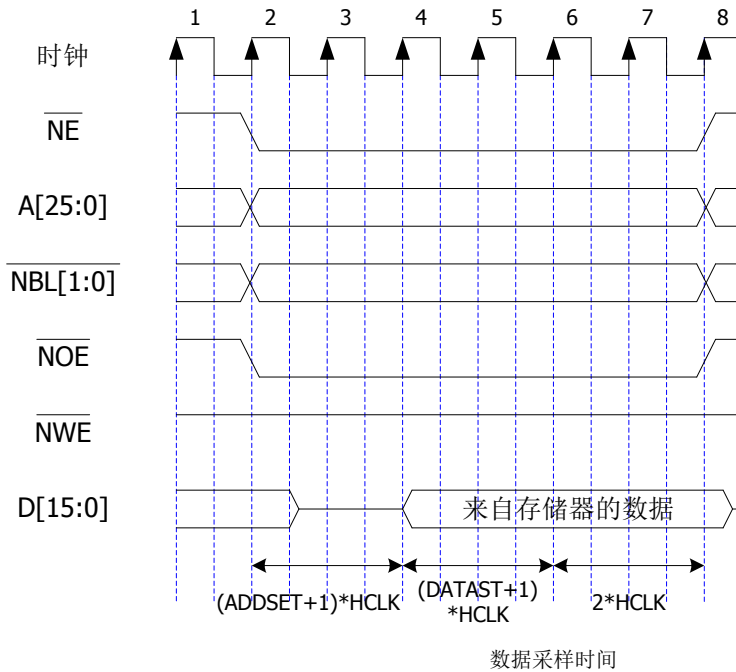


图 151 示例 1 模式 1 读操作时序



■ 示例 2

模式 A: SRAM/PSRAM OE 翻转

表 62 示例 1 模式 A 相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
14	EXTMOD	0x1
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
5:4	MWID	取决于存储器的参数
3:2	MTYP	取决于存储器的参数
1	MUXEN	0x0
0	MBKEN	0x1
EXMC_BTR 寄存器		
29:28	ACCMOD	0x0
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求
EXMC_BWTR 寄存器		
29:28	ACCMOD	0x0
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 152 示例 2 模式 A 写操作时序

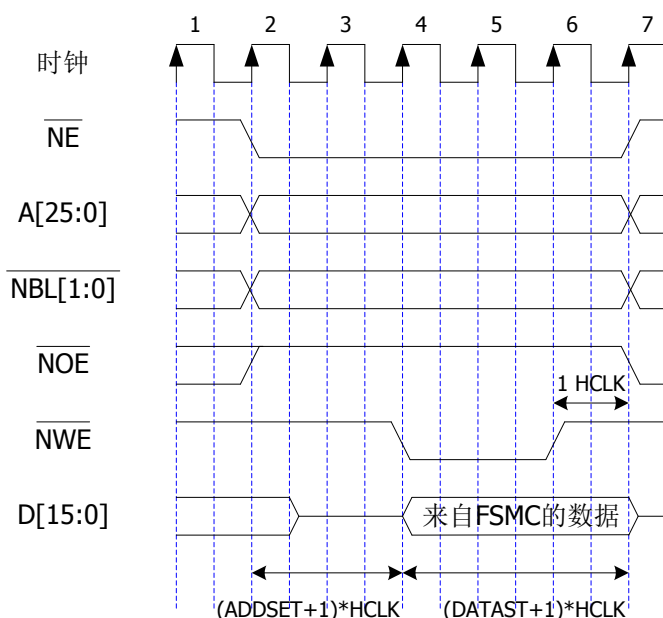
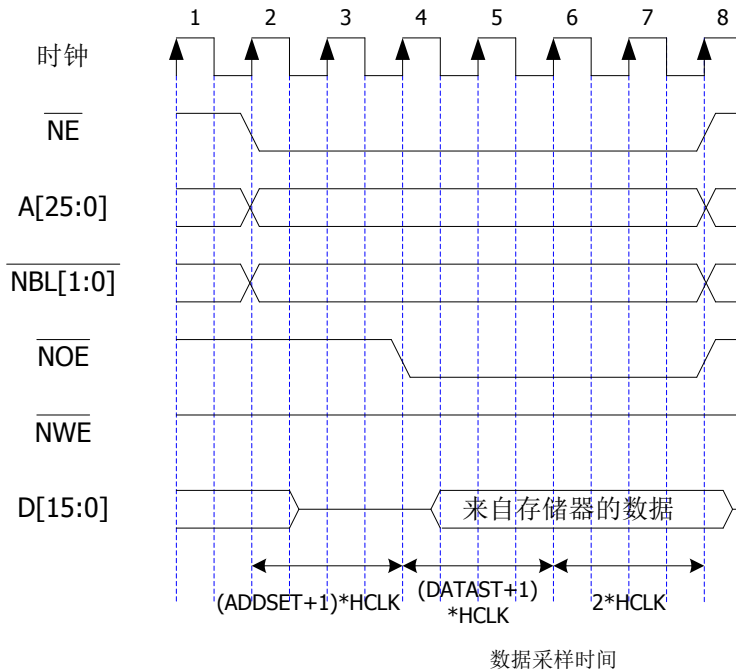


图 153 示例 2 模式 A 读操作时序



■ 示例 3

模式 2/B: NOR Flash

表 63 示例 3 模式 2/B 相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
14	EXTMOD	模式 B: 0x1; 模式 2: 0x0
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
5:4	MWID	取决于存储器的参数
3:2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1
EXMC_BTR 寄存器		
29:28	ACCMOD	若设置为扩展模式, 则为 0x1
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

EXMC_BWTR 寄存器		
29:28	ACCMOD	若设置为扩展模式，则为 0x1
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 154 示例 3 模式 B 写操作时序

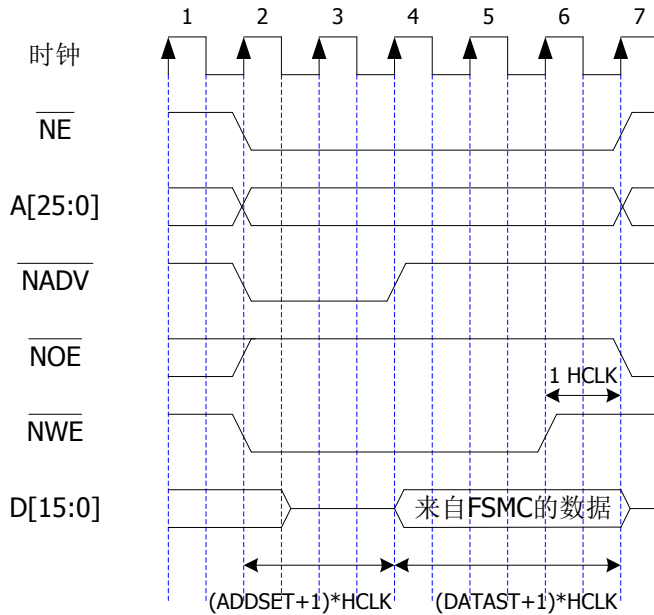


图 155 示例 3 模式 2 写操作时序

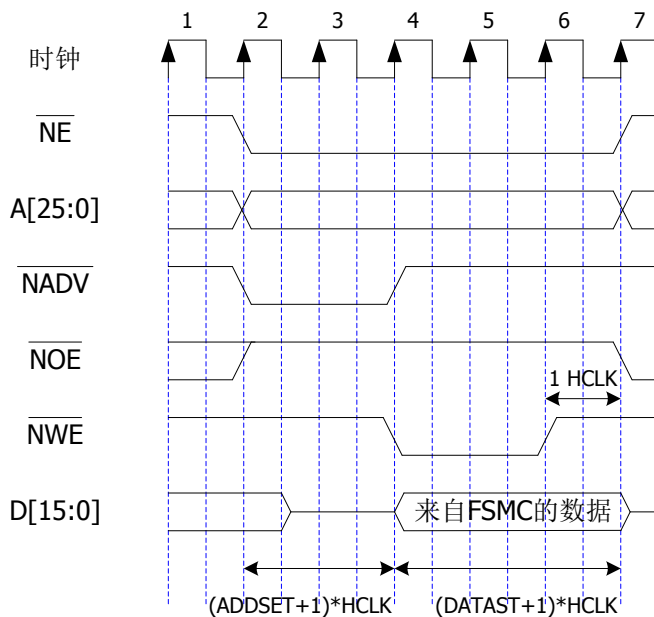
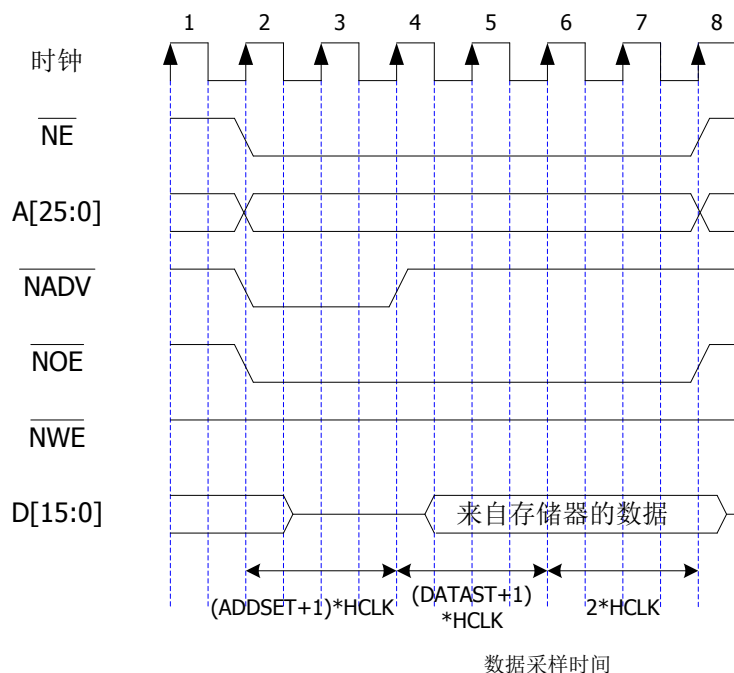


图 156 示例 3 模式 2/B 读操作时序



■ 示例 4

模式 C: NOR Flash OE 翻转

表 64 示例 4 模式 C 相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
14	EXTMOD	0x1
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
5:4	MWID	取决于存储器的参数
3:2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1
EXMC_BTR 寄存器		
29:28	ACCMOD	0x2
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求
EXMC_BWTR 寄存器		
29:28	ACCMOD	0x2

15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 157 示例 4 模式 C 写操作时序

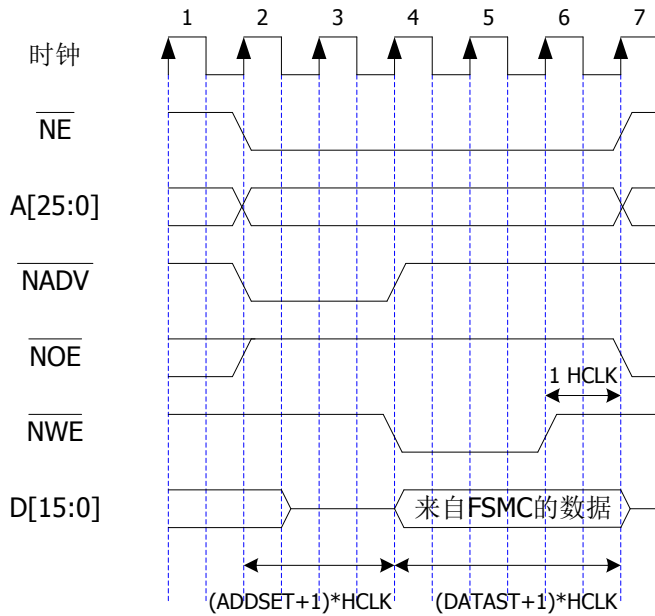
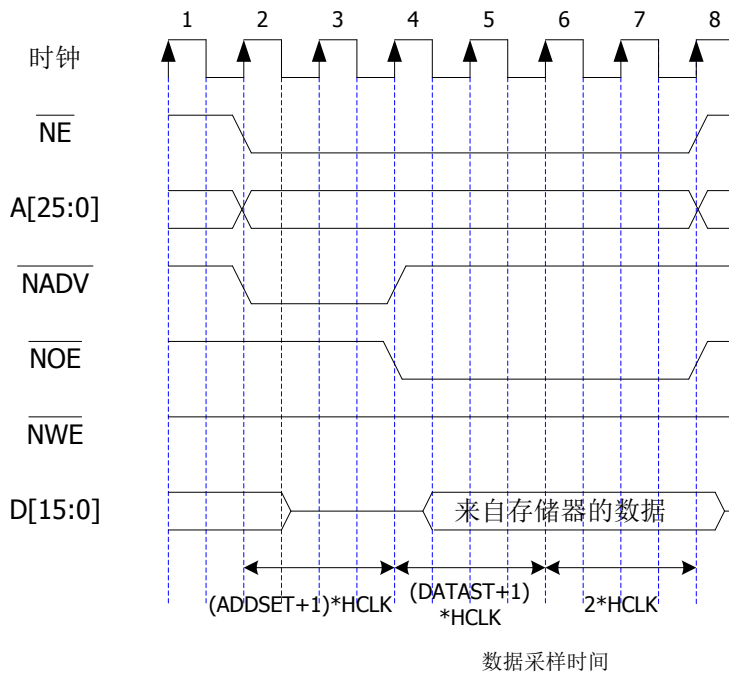


图 158 示例 4 模式 C 读操作时序



■ 示例 5

模式 D: 异步访问扩展地址

表 65 示例 5 模式 D 相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
14	EXTMOD	0x1
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
5:4	MWID	取决于存储器的参数
3:2	MTYP	取决于存储器的参数
1	MUXEN	0x0
0	MBKEN	0x1
EXMC_BTR 寄存器		
29:28	ACCMOD	0x2
15:8	DATAST	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求
EXMC_BWTR 寄存器		
29:28	ACCMOD	0x2
15:8	DATAST	取决于存储器的参数以及用户的要求
7:4	ADDHLD	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 159 示例 5 模式 D 写操作时序

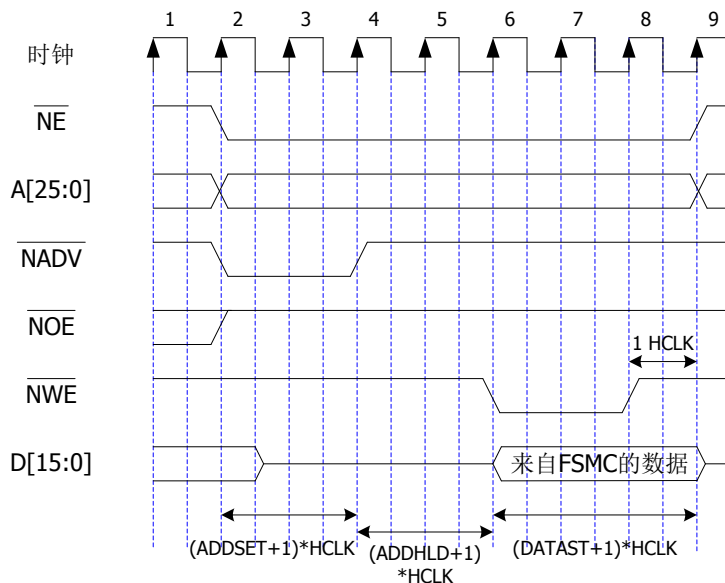
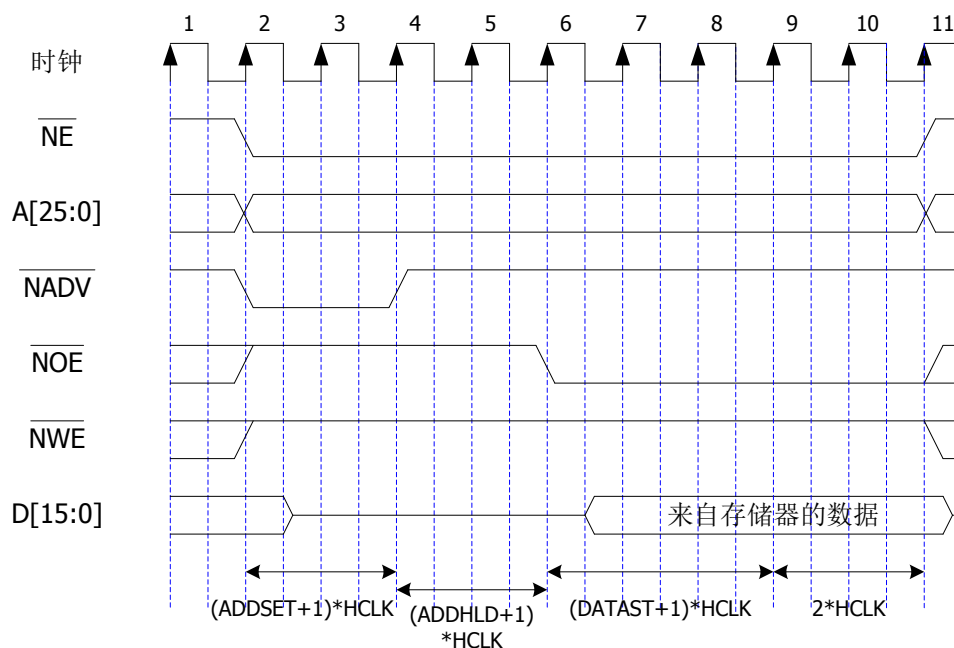


图 160 示例 5 模式 D 读操作时序



■ 示例 6

混合模式：异步访问复用 NOR Flash

表 66 示例 6 复用模式相关寄存器配置

EXMC_BCR 寄存器		
位域	位域名称	设置值
15	ASYNCWAIT	取决于存储器的参数
14	EXTMOD	0x0
9	WAITPOL	仅当第 15 位为 1 时才生效
8	BURSTEN	0x0
6	FACCEN	0x1
5:4	MWID	取决于存储器的参数
3:2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x1
0	MBKEN	0x1
EXMC_BTR 寄存器		
19:16	BUSTURN	取决于存储器的参数以及用户的要求
15:8	DATAST	取决于存储器的参数以及用户的要求
7:4	ADDHLD	取决于存储器的参数以及用户的要求
3:0	ADDSET	取决于存储器的参数以及用户的要求

图 161 示例 6 复用模式写操作时序

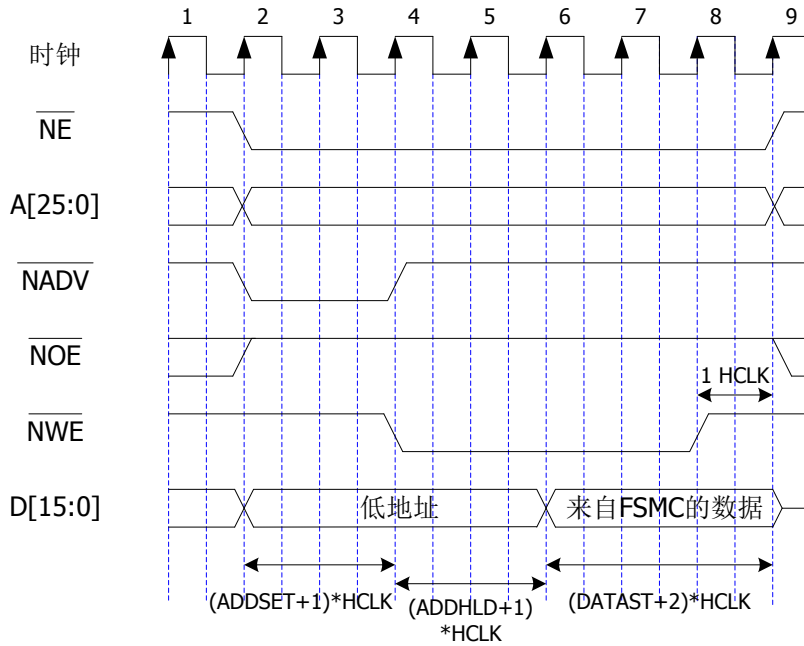
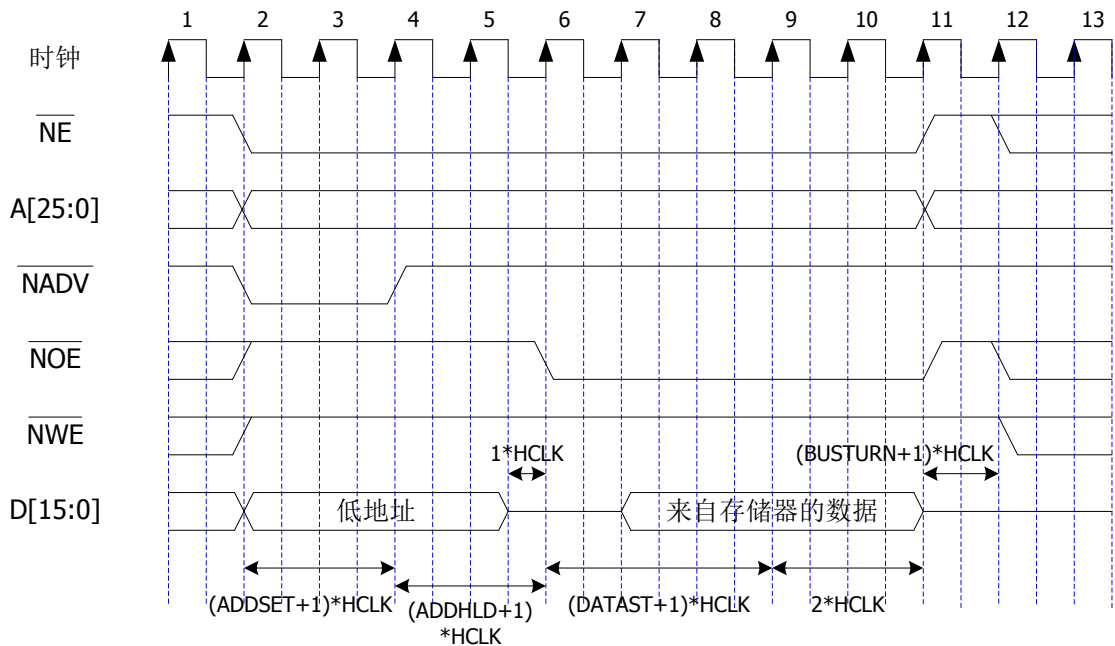


图 162 示例 6 复用模式读操作时序



■ 示例 7

等待功能

等待功能是通过 EXMC_BCR 寄存器的 ASYNCWAIT 控制的。当 EXMC 模块访问外部存储器期间，

若 $ASYNCWAIT=1$ ，则数据建立阶段将会自动延长。延长时间的计算如下：

(1) 若存储器等待信号按照 NOE/NWE 信号对齐：

$$\text{数据建立时间} \geq \text{等待信号有效时间} + 4 * HCLK$$

(2) 若存储器等待信号按照 NE 对齐：

if (等待信号有效时间 > (寻址时间 + 保持时间))

$$\text{数据建立时间} \geq (\text{等待信号有效时间} - \text{寻址时间} - \text{保持时间}) + 4 * HCLK$$

否则

$$\text{数据建立时间} \geq 4 * HCLK$$

图 163 示例 7 异步等待信号作用时的写操作时序

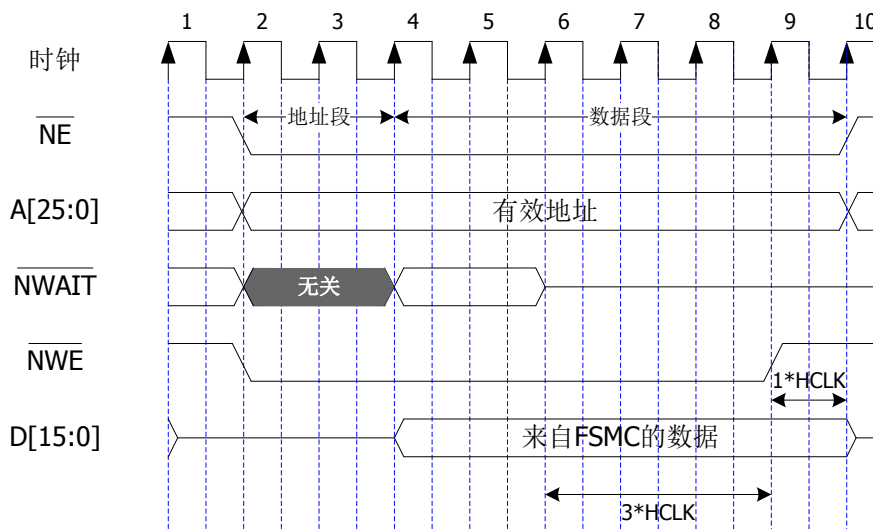
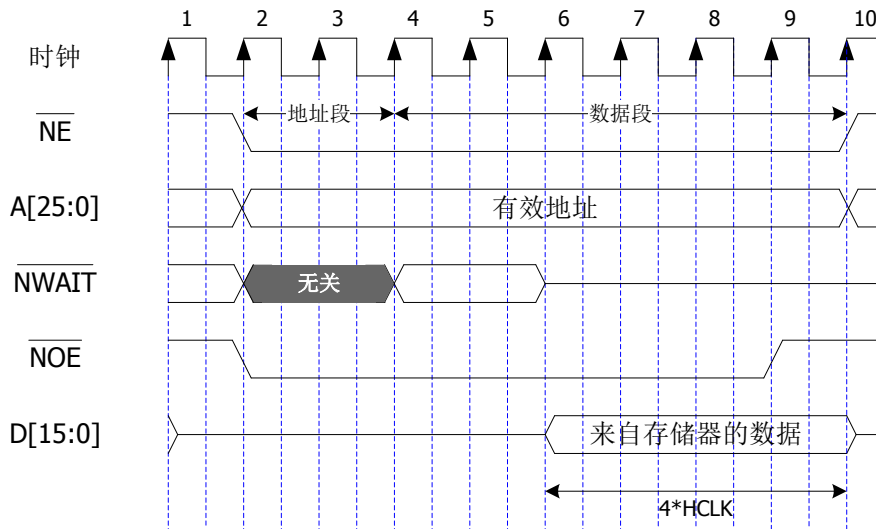


图 164 示例 7 异步等待信号作用时的读操作时序



4 EXMC 寄存器

4.1 SRAM/NOR Flash 片选控制寄存器 n (EXMC_BCRn) (n=1~4)

地址偏移: $0xA000\ 0000 + 8*(n-1)$

复位值 : $0x0000\ 30DX$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												CBURSTRW	保留		
												RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC WAIT	EXT MOD	WAIT EN	WREN	WAIT CFG	WRAP MOD	WAIT POL	BURST EN	保留	FACC EN	MWID	MTYP	MUXED	MBKEN		
RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:20	保留	请保持其复位状态
19	CBURSTRW	猝发写入使能 0: 异步写操作模式 1: 同步写操作模式
18:16	保留	请保持其复位状态
15	ASYNCWAIT	异步等待功能使能位 0: 禁用异步等待功能 1: 使能异步等待功能
14	EXTMOD	扩展模式使能位 0: 禁用扩展模式 1: 使能扩展模式

位/位域	名称	描述
13	WAITEN	同步等待功能使能位 0: 禁用同步等待时间插入 1: 使能同步等待时间插入
12	WREN	写操作使能位 0: 禁用写操作 1: 使能写操作
11	WAITCFG	等待时序配置，仅对同步模式有效 0: NWAIT 信号比等待提前一个数据周期生效 1: 等待期间，NWAIT 信号处于生效状态
10	WRAPMOD	滚动猝发模式支持 0: 禁用滚动猝发模式支持 1: 使能滚动猝发模式支持
9	WAITPOL	等待信号极性设置 0: NWAIT 信号低电平有效 1: NWAIT 信号高电平有效
8	BUSRTEN	同步猝发模式使能位 0: 禁用猝发访问模式 1: 使能猝发访问模式
7	保留	请保持其复位状态
6	FACCEN	Flash 访问使能位 0: 禁用 Flash 访问 1: 使能 Flash 访问
5:4	MWID	外部存储器数据宽度 0x0: 8 位 0x1: 16 位
3:2	MTYP	外部存储器类型 0x0: SRAM、ROM 0x1: PSRAM 0x2: NOR Flash 0x3: 保留
1	MUXED	地址/数据管脚复用使能位，仅对 NOR 和 PSRAM 有效 0: 禁用复用功能 1: 使能复用功能
0	MBKEN	bank 使能位 0: 禁用相关的存储器 bank 1: 使能相关的存储器 bank

4.2 SRAM/NOR Flash 片选时序寄存器 n (EXMC_BTRn) (n=1~4)

地址偏移: $0xA000\ 0000 + 0x04 + 8*(n-1)$

复位值 : 0x0FFF FFFF

若 EXMC_BCR 寄存器中的 EXTMOD=1，本寄存器配置读操作时序，EXMC_BWTR 寄存器用于配置写操作时序。

若 EXMC_BCR 寄存器中的 EXTMOD=0，本寄存器配置读写操作时序。

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		ACCMOD		DATLAT				CLKDIV				BUSTURN			
类型			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATAST								ADDHLD				ADDSET			
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:30	保留	请保持其复位状态
29:28	ACCMOD	访问模式 0x0: 模式 A 0x1: 模式 B 0x2: 模式 C 0x3: 模式 D
27:24	DATLAT	NOR Flash 的数据延迟。仅对同步访问有效 0x0: 首次猝发访问的数据延迟为 2 个 CLK 0x1: 首次猝发访问的数据延迟为 3 个 CLK 0xF: 首次猝发访问的数据延迟为 17 个 CLK 请注意: 这里的 CLK 并非 HCLK, 而是 EXMC 的输出时钟。本位域仅在同步模式下有效。
23:20	CLKDIV	获取 EXMC_CLK 的 AHB 总线时钟分频系数 0x0: 保留 0x1: EXMC_CLK 周期 = 2*HCLK 周期 0x2: EXMC_CLK 周期 = 3*HCLK 周期 0xF: EXMC_CLK 周期 = 16*HCLK 周期 请注意: 本位域仅在同步模式下有效。
19:16	BUSTURN	总线翻转延迟 当出现连续背靠背的总线访问时, 为防止总线竞争紊乱, 可设置本位域。 0x0: 总线翻转延迟 = 1*HCLK 周期 0xF: 总线翻转延迟 = 16*HCLK 周期
15:8	DATAST	数据段延迟的 HCLK 数 0x00: 保留 0x01: 数据段延迟 = 2*HCLK 周期 0x02: 数据段延迟 = 3*HCLK 周期 0xFF: 数据段延迟 = 256*HCLK 周期 请注意: 本位域仅在异步模式下有效。
7:4	ADDHLD	地址保持段时间 本位域用于设置地址保持段的时长, 仅在模式 D 以及复用模式下有效。 0x0: 保留 0x1: 地址保持段时间 = 2*HCLK 周期 0x2: 地址保持段时间 = 3*HCLK 周期 0xF: 地址保持段时间 = 16*HCLK 周期

位/位域	名称	描述
3:0	ADDSET	地址建立段时间 本位域用于定义地址建立段的时长。 请注意： 本位域仅在异步模式下且针对 SRAM、ROM、NOR Flash 时有效。

4.3 SRAM/NOR Flash 写时序寄存器 n (EXMC_BWTRn) (n=1~4)

地址偏移: $0xA000\ 0000 + 0x104 + 8*(n-1)$

复位值 : 0x0FFF FFFF

仅当 EXMC_BCR 寄存器中的 EXTMOD=1 时, 本寄存器的配置才会生效。

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

位/位域	名称	描述
31:30	保留	请保持其复位状态
29:28	ACCMOD	访问模式 0x0: 模式 A 0x1: 模式 B 0x2: 模式 C 0x3: 模式 D
27:24	DATLAT	NOR Flash 的数据延迟。仅对同步访问有效 0x0: 首次猝发访问的数据延迟为 2 个 CLK 0x1: 首次猝发访问的数据延迟为 3 个 CLK 0xF: 首次猝发访问的数据延迟为 17 个 CLK 请注意： 这里的 CLK 并非 HCLK, 而是 EXMC 的输出时钟。本位域仅在同步模式下有效。
23:20	CLKDIV	获取 EXMC_CLK 的 AHB 总线时钟分频系数 0x0: 保留 0x1: EXMC_CLK 周期 = 2*HCLK 周期 0x2: EXMC_CLK 周期 = 3*HCLK 周期 0xF: EXMC_CLK 周期 = 16*HCLK 周期 请注意： 本位域仅在同步模式下有效。
19:16	BUSTURN	总线翻转延迟 当出现连续背靠背的总线访问时, 为防止总线竞争紊乱, 可设置本位域。 0x0: 总线翻转延迟 = 1*HCLK 周期 0xF: 总线翻转延迟 = 16*HCLK 周期

位/位域	名称	描述
15:8	DATAS	数据段延迟的 HCLK 数 0x00: 保留 0x01: 数据段延迟 = 2*HCLK 周期 0x02: 数据段延迟 = 3*HCLK 周期 0xFF: 数据段延迟 = 256*HCLK 周期 请注意: 本位域仅在异步模式下有效。
7:4	ADDHLD	地址保持段时间 本位域用于设置地址保持段的时长，仅在模式 D 以及复用模式下有效。 0x0: 保留 0x1: 地址保持段时间 = 2*HCLK 周期 0x2: 地址保持段时间 = 3*HCLK 周期 0xF: 地址保持段时间 = 16*HCLK 周期
3:0	ADDSET	地址建立段时间 本位域用于定义地址建立段的时长。 请注意: 本位域仅在异步模式下且针对 SRAM、ROM、NOR Flash 时有效。

第17章 文档修订记录

表 67 修订历史

修订版本	描述	修订日期
1.0	初版	2013年3月8日