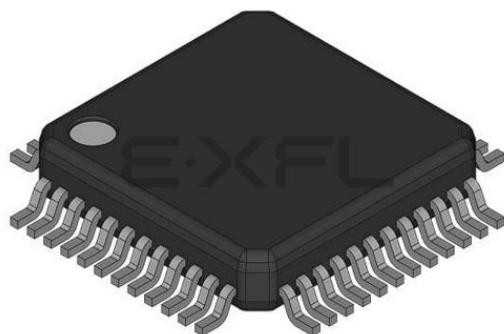




Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	39
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dz32clf

4.5.4 突发编程执行

突发编程命令能在比标准编程命令更短的时间内对数据的连续字节进行编程。这是因为 Flash 阵列的高电压在编程操作之间不需要断开。通常情况下，在发出编程或擦除命令后，必须启用与 Flash 相关的一个内部电荷泵用于为阵列提供高电压。命令执行完成后，该电荷泵会被关闭。发出突发编程命令后，电荷泵在以下两种条件下会被开启而且在突发编程操作完成后将保持开启状态：

- 下一个突发编程命令序列在设置 **FCCF** 位之前已开始。
- 下一个顺序地址从所编程的当前字节所在的相同突发块中选择一个字节。该 Flash 中的突发块包括 32 个字节。新的突发块在每个 32 字节地址的边界开始。

在突发模式下对一系列连续字节的第一个字节进行编程所需要的时间与标准模式下编程一个字节所需的时间相同。如果达到上述两个条件，后面的字节将在突发编程时间内编程。如果下一个顺序地址是新的一行的开始，那么该字节的编程时间将是标准时间而不是突发时间。这是因为到阵列的高电压必须断开后重新开启。如果在当前命令完成前，队列中没有任何新的突发命令，那么电荷泵将关闭，高电压将从阵列上断开。

图 4-3 为执行突发编程操作的流程。

表 4-15. FSTAT 寄存器字段描述 (continued)

字段	描述
4 FACCERR	访问错误标记 — FACCERR 在以下情况下自动设置：正确的命令顺序没有严格遵守（错误的命令将被忽略），FCDIV 寄存器初始化之前尝试进行编程或擦除操作，或在命令正在执行时 MCU 进入停止模式。若欲了解会被认为是访问错误的具体操作的更详尽信息，请参见 4.5.6，“访问错误”。FACCERR 通过向 FACCERR 中写入一个 1 来清除。向 FACCERR 中写入 0 没有任何意义或效果。 0 没有访问错误。 1 发生了访问错误。
2 FBLANK	验证为全空（被擦除）标记 — FBLANK 在空白检查命令完成后自动设置为 1（如果整个 Flash 或 EEPROM 阵列被确认已擦除）。FBLANK 通过清除 FCBEF 以写入新的有效命令来清除。向 FBLANK 中写入没有任何意义或效果。 0 在空白检查命令执行完成而且 FCCF = 1 的情况下，FBLANK = 0 表示 Flash 或 EEPROM 阵列未被完全擦除。 1 在空白检查命令执行完成而且 FCCF = 1 的情况下，FBLANK = 1 表示 Flash 或 EEPROM 阵列已完全擦除（全部为 0xFFFF）。

4.5.11.6 Flash 和 EEPROM 命令寄存器 (FCMD)

如表 4-16 所示，正常用户模式下只能识别 6 种命令代码。所有其他命令代码都是非法的，会产生访问错误。请参见 4.5.3，“编程和擦除命令的执行”来了解对 Flash 和 EEPROM 编程及擦除操作的详细描述。

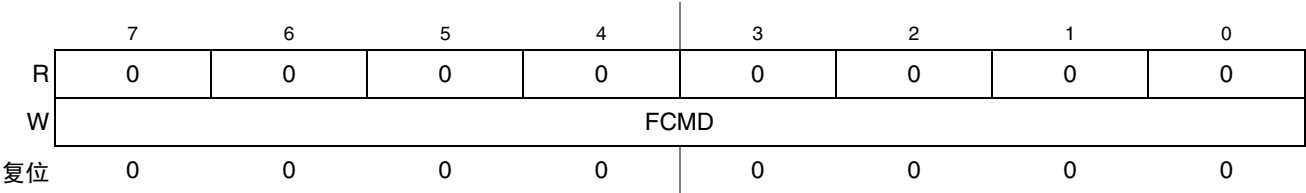


图 4-10. Flash 和 EEPROM 命令寄存器 (FCMD)

表 4-16. Flash 和 EEPROM 命令

命令	FCMD	等同文件标签
空白检查	0x05	mBlank
字节编程	0x20	mByteProg
突发编程	0x25	mBurstProg
分区擦除	0x40	mSectorErase
整体擦除	0x41	mMassErase
分区擦除终止	0x47	mEraseAbort

进行整体擦除操作后没有必要再执行空白检查命令。空白检查只是安全解锁机制的一部分。

5.5.2.2 边沿和电平敏感度

IRQMOD 控制位重新配置检测逻辑，这样它就能检测边沿事件和管脚电平。在边沿和电平检测模式中，当检测到边沿时（IRQ 管脚从断言解除改为断言），IRQF 状态标记就被设置，但只要 IRQ 管脚处于断言级，就会连续设置该标记（并且不能清除）。

5.5.3 中断向量、源和本地掩码

表 5-1 总结了所有中断源。较高优先级的源位于表格下方。中断服务程序地址的高阶字节位于向量地址栏的第一个地址，中断服务程序地址的低阶字节位于下一个较高阶地址中。

当出现中断时，相关标记位被设置。如果相关的本地中断激活位是 1，中断请求会发送到 CPU。在 CPU 中，如果全球中断屏蔽（CCR 中的 I 位）是 0，CPU 将完成当前指令；堆栈 PCL、PCH、X、A 和 CCR CPU 寄存器；设置 I 位；然后为挂起的最高优先级中断获取中断向量。然后继续处理中断服务程序。

表 7-3. 操作码映射（第 2 页，共 2 页）

Bit-Manipulation		Branch	Read-Modify-Write				Control				Register/Memory							
							9E60	6				9ED0	5	9EE0	4			
							NEG	3	SP1			SUB	4	SUB	3	SP1		
							9E61	6				9ED1	5	9EE1	4			
							CBEQ	4	SP1			CMP	4	CMP	3	SP1		
												9ED2	5	9EE2	4			
												SBC	4	SBC	3	SP1		
							9E63	6				9ED3	5	9EE3	4		9EF3	6
							COM	3	SP1			CPX	4	CPX	3	SP1	CPHX	3
							9E64	6				9ED4	5	9EE4	4			
							LSR	3	SP1			AND	4	AND	3	SP1		
												9ED5	5	9EE5	4			
												BIT	4	BIT	3	SP1		
							9E66	6				9ED6	5	9EE6	4			
							ROR	3	SP1			LDA	4	LDA	3	SP1		
							9E67	6				9ED7	5	9EE7	4			
							ASR	3	SP1			STA	4	STA	3	SP1		
							9E68	6				9ED8	5	9EE8	4			
							LSL	3	SP1			EOR	4	EOR	3	SP1		
							9E69	6				9ED9	5	9EE9	4			
							ROL	3	SP1			ADC	4	ADC	3	SP1		
							9E6A	6				9EDA	5	9EEA	4			
							DEC	3	SP1			ORA	4	ORA	3	SP1		
							9E6B	8				9EDB	5	9EEB	4			
							DBNZ	4	SP1			ADD	4	ADD	3	SP1		
							9E6C	6										
							INC	3	SP1									
							9E6D	5										
							TST	3	SP1									
											9EAE	5	9EBE	6	9ECE	5	9EDE	5
											LDHX	2	LDHX	4	LDHX	3	LDX	4
												IX	IX2	IX1	SP2	SP1	SP1	SP1
							9E6F	6				9EDF	5	9EEF	4		9EFF	5
							CLR	3	SP1			STX	4	STX	3	SP1	STHX	3

INH 固有
 IMM 即时
 DIR 直接
 EXT 扩展
 DD DIR 至 DIR
 IX+D IX+ 至 DIR
 REL 相关
 IX 索引，无偏移
 IX1 索引，8 位偏移
 IX2 索引，16 位偏移
 IMD IMM 至 DIR
 DIX+ DIR 至 IX+
 SP1 堆栈指针，8 位偏移
 SP2 堆栈指针，16 位偏移
 IX+ 有索引，无偏移、带后增量
 IX1+ 有索引，1 字节偏移、带后增量

注意：第 2 页的所有操作码前面都带有 Page 2 Prebyte（9E）

十六进制操作码 | 9E60 6
 字节数 | NEG 3 SP1
 HCS08 周期
 指令助记符
 寻址模式

8.5.1.9 Stop

每当 MCU 进入 STOP 状态就进入 STOP 模式。在该模式中，FLL 和 PLL 被禁止，所有 MCG 时钟信号静止，但下列情况除外：

当满足下列条件时，MCGIRCLK 将在停止模式中使能：

- IRCLKEN = 1
- IREFSTEN = 1

当满足下列条件时，MCGERCLK 将在停止模式中使能：

- ERCLKEN = 1
- EREFSTEN = 1

8.5.2 模式切换

当在 Engaged Internal 和 Engaged External 间进行模式切换时，IREFS 位可以随时修改，但必须同时修改 RDIV 位，这样参考频率处于 PLLS 位状态所要求的范围内（如果选择 FLL，则介于 31.25 kHz -39.0625 kHz 之间；如果选择 PLL，则介于 1 MHz -2 MHz 之间）。在 IREFS 值更改后，FLL 或 PLL 在切换完成后会被再次锁定。切换完成由 IREFST 位标志。

对于切换到 FBE 模式后立即进入停止模式的特殊情况，如果外部时钟和内部时钟在停止模式中被禁止（EREFSTEN = 0，IREFSTEN = 0），在清除 IREFST 位后必须允许 100us 来切断内部参考。在大多数情况下，由于指令执行次数造成的延迟都足够用。

CLKS 位也可以随时修改，但为了正确配置 MCGLCLK，必须同时修改 RDIV 位，这样参考频率就处于 PLLS 位状态所要求的范围内（如果选择 FLL，则介于 31.25 kHz -39.0625 kHz 之间；如果选择 PLL，则介于 1 MHz -2 MHz 之间）。实际切换到刚选择的时钟由 CLKST 位表示。如果刚选择的时钟不可用，则仍选择原来的时钟。

如需了解更多信息，请参见图 8-8。

8.5.3 总线分频器

BDIV 位可以随时修改，实际切换到新频率将立即进行。

8.5.4 低功率位使用

提供低功率位（LP）的目的是为了在不需要使用 FLL 或 PLL 时禁止它们，从而达到节电目的。然而，在有些应用中可能需要使能 FLL 或 PLL，使它在切换到下一个 Engaged 模式前锁定，以实现最大精确度，只需向 LP 位写入 0，即可完成该操作。

第 10 章

数模转换器 (S08ADC12V1)

10.1 介绍

12 位数模转换器 (ADC) 是逐次逼近式 ADC，为集成微控制器片上系统设计。

注意

MC9S08DZ60 系列器件的工作电压 (2.7 V -5.5 V) 较高，不支持 STOP1 模式。请忽略 STOP1 参考。

10.1.1 模拟功率和接地信号名称

本章的 V_{DDAD} 和 V_{SSAD} 参考分别对应 V_{DDA} 和 V_{SSA} 信号。

10.1.2 信道分配

注意

MC9S08DZ60 系列器件的 ADC 通道分配如表 10-1. 所示。保留通道转换结果未知。

本章介绍了所有 S08ADC12V1 通道位。MC9S08DZ60 系列 MCU 并不全部使用所有这些通道。器件上不可用通道对应的所有位都被保留。

表 12-1. CANCTL0 寄存器字段描述 (continued)

字段	描述
1 SLPRQ ⁴	睡眠模式请求—该位请求 MSCAN 进入睡眠模式，这是一个内部节电模式（参见 12.5.5.4，“MSCAN 睡眠模式”）。当 CAN 总线空闲时，也就是说该模块不接收任何报文且所有发送缓冲器空，睡眠模式请求被受理。通过设置 SLPK = 1（参见 12.3.2，“控制寄存器 1 (CANCTL1)”），表示该模块进入睡眠模式。当设置了 WUPIF 标记时（参见 12.3.4.1，“MSCAN 接收器标志寄存器 (CANRFLG)”），不能设置 SLPRQ。睡眠模式维持有效，直到 SLPRQ 被 CPU 清除或者根据 WUPE 的设置，MSCAN 检测到 CAN 总线上有有效并自行清除。 0 运行中 — MSCAN 正常工作 1 睡眠模式请求— 当 CAN 总线空闲时 MSCAN 进入睡眠模式
0 INITRQ ^{5,6}	INITRQ6、7 初始化模式请求— 当 CPU 设置该位时，MSCAN 切换至初始化模式（参见 12.5.5.5，“MSCAN I 初始化模式”）。任何正在进行的发送或接收都将被中止，与 CAN 总线的同步也丢失。通过设置 INITAK = 1（见 12.3.2 小节“MSCAN 控制寄存器 1 (CANCTL1)”），表示该模块进入初始化模式。 以下寄存器进入其硬复位状态并恢复它们的默认值：CANCTL08、CANRFLG9、CANRIER10、CANTFLG、CANTIER、CANTARQ、CANTAARQ 和 CANTBSEL。 μ 盞 SCAN 处于初始化模式（INITRQ = 1，INITAK = 1）时，寄存器 CANCTL1、CANBTR0、CANBTR1、CANIDAC、CANIDAR0-7 和 CANIDMR0-7 只能通过 CPU 写入。错误计数器的值不受初始化模式的影响。 当该位通过 CPU 清除时，MSCAN 重启，然后试图与 CAN 总线同步。如果 MSCAN 未处于总线脱离状态，它在 CAN 总线上出现 11 个连续隐性位后同步。如果 MSCAN 处于总线脱离状态，它将继续等待 11 个连续隐性位重复出现 128 次。 只有当退出初始化模式后，才可以在 CANCTL0、CANRFLG、CANRIER、CANTFLG 或 CANTIER 中写入其他位，这时 INITRQ = 0，INITAK = 0。 0 正常运行 1 MSCAN 处于初始化模式

¹ 要设置该位，MSCAN 必须处于正常模式。² 如需了解发送器和接收器状态的详细定义，（参见 12.5.5.2，“等待模式中的操作”和 12.5.5.3，“停止模式中的操作”）。³ 为了防止意外违反 CAN 协议，当 CPU 进入等待（CSWAI = 1）或停止模式（参见 12.3.5，“MSCAN 接收器中断使能寄存器 (CANRIER)”），立即强制 TXCAN 管脚进入隐性状态。⁴ 如果需从停止或等待模式中进行恢复的机制，CPU 必须确保 WUPE 位和 WUPIE 唤醒中断使能位（参见 12.3.5，“MSCAN 接收器中断使能寄存器 (CANRIER)”）（CANRIER）被使能。⁵ 在 MSCAN 进入睡眠模式（SLPRQ = 1，SLPAK = 1）前，CPU 不能清除 SLPRQ。⁶ 在 MSCAN 进入初始化模式（INITRQ = 1，INITAK = 1）前，CPU 不能清除 INITRQ。

表 12-6. CANBTR1 寄存器字段描述

字段	描述
7 SAMP	采样 — 该位确定每位时间所采集的 CAN 总线样本数量 0 每位 1 个样本 1 每位 3 个样本 1。 如果 SAMP = 0，得到的位值等于采样点上定位的单个位的值。如果 SAMP = 1，得到的位值通过在总共三个采样点上使用多数规则来决定。要实现更高比特速率，建议每个位时间只采集一个样本（SAMP = 0）。).
6:4 TSEG2[2:0]	TSEG2[2:0] 时间段 2—位时间内的时间段固定每个位时间的时钟周期数和采样点的位置 (参见图 12-43). 时间段 2（TSEG2）值可以如表 12-7 所示进行编程。
3:0 TSEG1[3:0]	时间段 1—位时间内的时间段固定每个位时间的时钟周期数和采样点的位置 (参见图 12-43). 时间段 1（TSEG1）值可以如表 12-8 所示进行编程。.

表 12-7. 时间段 2 值

TSEG22	TSEG21	TSEG20	时间段 2
0	0	0	1 Tq 时钟周期 ¹
0	0	1	2 Tq 时钟周期
:	:	:	:
1	1	0	7 Tq 时钟周期
1	1	1	8 Tq 时钟周期

¹ This setting is not valid. Please refer to 表 12-35 for valid settings.

表 12-8. 时间段 1 值

TSEG13	TSEG12	TSEG11	TSEG10	时间 段 1
0	0	0	0	1 Tq c 时钟周期 ¹
0	0	0	1	2 Tq 时钟周期 ¹
0	0	1	0	3 Tq 时钟周期 ¹
0	0	1	1	4 Tq 时钟周期
:	:	:	:	:
1	1	1	0	15 Tq 时钟周期
1	1	1	1	16 Tq 时钟周期

¹ 该设置无效，请参见表 12-35 查看有效设置。

位时间由振荡器频率、波特率预分频器和每位的时间冲量（Tq）数量确定（如表 12-7 和表 12-8).

等式 12-1

Bit Time= $\frac{(\text{Prescaler value})}{f_{\text{CANCLK}}} \cdot (1 + \text{TimeSegment1} + \text{TimeSegment2})$

12.4 报文存储模式

下面这一节详细介绍了接收和发送报文缓冲器的结构，以及相关控制寄存器。

了简化程序员界面，接收和发送报文缓冲器的轮廓相同。每个报文缓冲器都在包含 13 字节数据结构的存储器映射中都分配 16 个字节。

我们还为发送缓冲器定义了一个发送缓冲器优先级寄存器（TBPR）。在该存储器映射的最后两个字节中，MSCAN 保存一个特殊的 16 位时间标签，采样于报文成功传输或接收后的内部计时器。如果设置了 TIME 位，这种功能只出现于发送和接收器缓冲器（参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”）。

时间标签寄存器由 MSCAN 写入。CPU 只能读这些寄存器。

表 12-24. 报文缓冲器结构

Offset Address	Register	Access
0x00X0	Identifier Register 0	
0x00X1	Identifier Register 1	
0x00X2	Identifier Register 2	
0x00X3	Identifier Register 3	
0x00X4	Data Segment Register 0	
0x00X5	Data Segment Register 1	
0x00X6	Data Segment Register 2	
0x00X7	Data Segment Register 3	
0x00X8	Data Segment Register 4	
0x00X9	Data Segment Register 5	
0x00XA	Data Segment Register 6	
0x00XB	Data Segment Register 7	
0x00XC	Data Length Register	
0x00XD	Transmit Buffer Priority Register ¹	
0x00XE	Time Stamp Register (High Byte) ²	
0x00XF	Time Stamp Register (Low Byte) ³	

¹ Not applicable for receive buffers

² Read-only for CPU

³ Read-only for CPU

图 12-23 为扩展标识符显示了接收和发送缓冲器常用的 13 字节数据结构。标准标识符在 IDR 寄存器中的映射图 12-24。

由于基于 RAM1，接收和发送缓冲器的所有位在复位时均为 ‘x’。接收和发送缓冲器的所有保留或不使用位始终读为 ‘x’。

12.5.7.7 从恢复停止或 等待

MSCAN 可以通过唤醒中断从停止或等待中恢复。只有当 MSCAN 在进入断电模式前处于 睡眠模式 (SLPRQ = 1, SLPK = 1) 时, 唤醒选项被使能 (WUPE = 1), 唤醒中断使能 (WUPIE = 1), 这种中断才能发生。

12.6 初始化 / 应用信息

12.6.1 MSCAN 初始化

系统复位后, 初始化 MSCAN 模块的流程如下:

1. 置位 CANE
2. 写入处于初始化模式的配置寄存器
3. 清除 INITRQ, 离开初始化模式, 进入正常模式

当 MSCAN 模块处于正常模式下, 需要更改只能在初始化模式中写入的寄存器:

1. CAN 总线空闲后, 通过设置 SLPRQ 并等待 SLPK 进行确认, 将模块置入睡眠模式。
2. 进入初始化模式: 确定 INITRQ 并等待 INITAK
3. 写入处于初始化模式的配置寄存器
4. 清除 INITRQ, 离开初始化模式, 继续保持正常模式

12.6.2 总线脱离恢复

用户可配置总线脱离恢复功能。总线脱离状态既可以自动退出, 也可以在用户的请求下退出。

出于向前兼容原因, 复位后, MSCAN 默认为自动恢复。在这种情况下, 在计数 128 次 CAN 总线上 11 个连续隐性位的出现后, MSCAN 将重新变成 ERROR ACTIVE (详情请参见 Bosch CAN 规范)。

如果 MSCAN 配置为用于用户请求模式 12.3.2, “控制寄存器 1 (CANCTL1)” 中设置的 BORM), 从总线脱离中恢复依赖于以下两个独立事件都成立后:

- 发现 128 次 CAN 总线上的 11 个连续隐性位
- 12.3.12, “MSCAN 其他寄存器 (CANMISC)” 中的 BOHOLD 已经被用户清除

这两个事件的发生顺序任意。

表 13-1. SPIC1 字段描述 (continued)

字段	描述
1 SSOE	辅选择输出使能 — 该位的使用结合 SPCR2 中的模式故障使能 (MODFEN) 位和主从 (MSTR) 控制位，以确定 SS 管脚的功能，如表 13-2 所示。
0 LSBFE	LSB 先发 (移位器方向) 0 SPI 串行数据传输始于最高位 1 SPI 串行数据传输始于最低位

表 13-2. $\overline{\text{SS}}$ 管脚功能

MODFEN	SSOE	主模式	辅模式
0	0	通用 I/O (非 SPI)	从选择输入
0	1	通用 I/O (非 SPI)	从选择输入
1	0	模式故障的 SS 输入	从选择输入
1	1	自动 SS 输出	从选择输入

注意

确保在位更改为 CPHA 位的同时 SPI 不得禁止 (SPE=0)。这些更改应作为独立操作执行，否则可能发生意外。

13.4.2 SPI 控制寄存器 2 (SPIC2)

该读 / 写寄存器用来控制 SPI 系统的可选功能。位 7、6、5 和 2 不执行，始终读为 0。

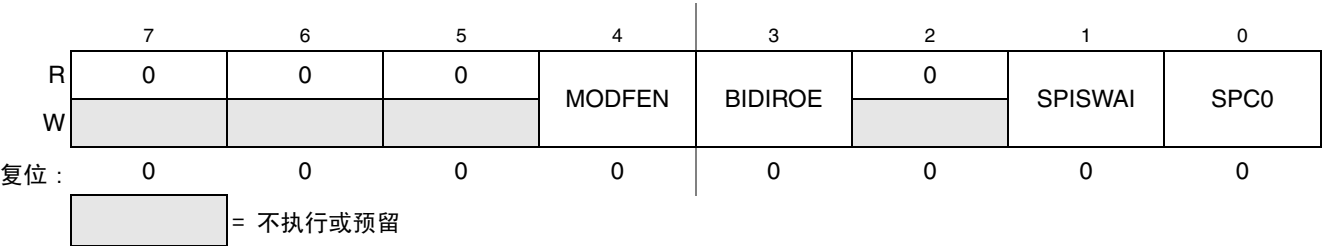


图 13-6. SPI 控制寄存器 2 (SPIC2)

表 13-3. SPIC2 寄存器字段描述

字段	描述
4 MODFEN	主模式故障功能使能 — 当为辅模式配置 SPI 时，该位没有意义或影响 (SS 管脚是从选择输入)。在主模式中，该位决定 SS 管脚的使用方式 (如需了解更多信息，参见表 13-2)。 0 模式故障功能禁止，主 SS 管脚恢复为不受 SPI 控制的通用 I/O。 1 模式故障功能使能，主 SS 管脚用作模式故障输入或辅选择输出
3 BIDIROE	双向模式输出使能 — 双向模式由 SPI 管脚控制 0 (SPC0 = 1) 使能时，BIDIROE 决定 SPI 数据输出驱动器是否被使能为单个双向 SPI I/O 管脚。根据 SPI 是配置为主 SPI 还是从 SPI，它将 MOSI (MOMI) 或 MISO (SISO) 管脚分别用作单个 SPI 数据 I/O 管脚，当 SPC0 = 0，BIDIROE 没有意义或影响。 0 输出驱动器禁止，因此 SPI 数据 I/O 管脚作为输入 1 SPI I/O 管脚作为输出使能

13.5.1 SPI 时钟格式

为了适应不同制造商的各种同步串行外围器件，SPI 系统有一个时钟极性（CPOL）位和一个时钟相位（CPHA）控制位，从四种时钟格式中选择一种进行数据传输。CPOL 有选择性地插入了与时钟串行的取反逻辑。CPHA 选择时钟和数据间的两种不同时钟相位关系。

图 13-10 显示了 CPHA = 1 时的时钟格式。在图的顶部，显示了 8 个位时间，作为参考。第一个位始于第一个 SPSCCK 边沿，第八个位结束于第 16 个 SPSCCK 边沿后的半个 SPSCCK 周期。MSB First 和 LSB First 线根据 LSBFE 中的设置显示了 SPI 数据位的顺序。SPSCCK 极性的两个变化都显示了出来，但这两个波形中只有一个适用于特定传输，具体哪一个取决于 CPOL 中的值。SAMPLE IN 波形适用于辅器件的 MOSI 输入或主器件的 MISO 输入。MOSI 波形适用于主器件的 MOSI 输出管脚，MISO 波形适用于辅器件的 MISO 输出。SS OUT 波形适用于主器件的辅选择输出（如果 MODFEN，SSOE = 1）。在传输开始前的半个 SPSCCK 周期，主 SS 输出信号进入有效低电平状态，在传输的第 8 个位时间结束时返回有效高态。SS IN 波形适用于辅器件的辅选择输入。

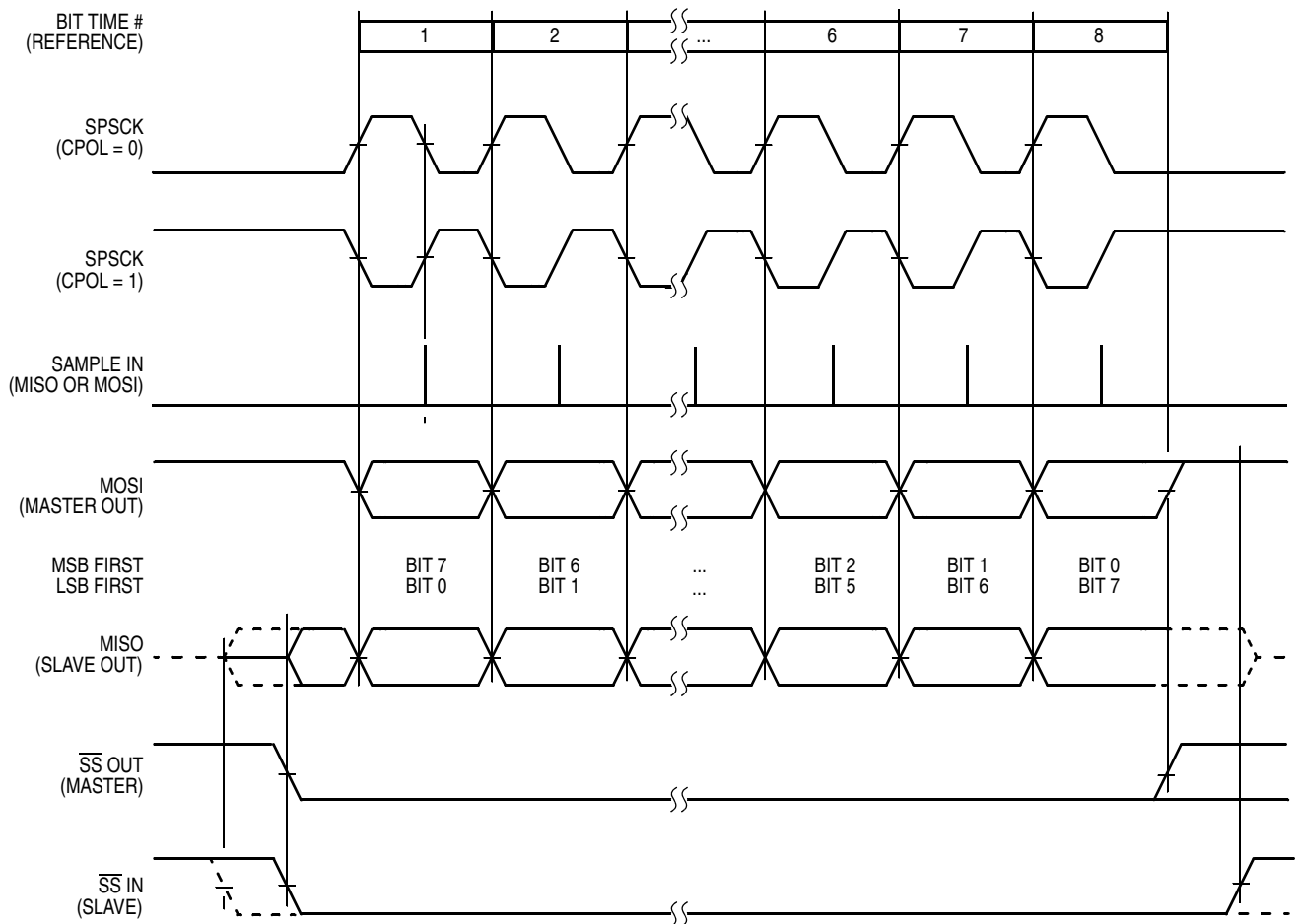


图 13-10. SPI 钟格式（CPHA = 1）

14.2 寄存器定义

SCI 有 8 个 8 位寄存器，来控制波特速率、选择 SCI 选项、报告 SCI 状态和发送 / 接收数据。
如需了解所有 SCI 寄存器的绝对地址分配，参见本文 Memory 章的直接页面寄存器概述。本小节只提及了寄存器和控制位的名称，飞思卡尔提供的对照表或头文件用来把这些名称转换成适当的绝对地址。

14.2.1 SCI 波特率寄存器 (SCIxBDH, SCIxBDL)

这两个寄存器控制着生成 SCI 波特率的预分频系数。要更新 13 位波特率设置 [SBR12:SBR0]，首先写入 SCIxBDH，缓存生成的高位字节，然后再写入 SCIxBDL。SCIxBDH 中的值在写入 SCIxBDL 前不会变化。
SCIxBDL 复位为非零值，因此复位后，波特率发生器一直保持禁止，直到第一次使能接收器或发射器（SCIxC2 中的 RE 或 TE 位写为 1）。

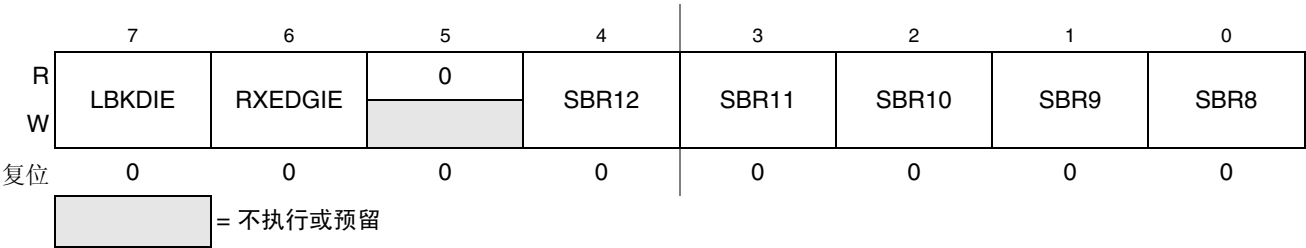


图 14-4. SCI 波特率寄存器 (SCIxBDH)

表 14-2. SCIxBDH 字段描述

字段	描述
7 LBKDIE	LIN 断点检测中断使能（用于 LBKDIF） 0 来自 LBKDIF 的硬件中断禁止（使用轮询） 1 当 LBKDIF 标记为 1 时，硬件中断允许
6 RXEDGIE	RxD 输入活动边中断使能（用于 RXEDGIF） 0 禁止来自 RXEDGIF 的中断（使用轮询）。 1 当 RXEDGIF 标记为 1 时，允许硬件中断。
4:0 SBR[12:8]	波特率模数除数 — SBR[12:0] 中的 13 个位统称为 BR，它们为 SCI 波特率发生器设置模数除数系数。当 BR = 0，SCI 波特率发生器禁止，以降低电源电流。当 BR = 1~8191 时，SCI 波特率 = BUSCLK/（16xBR）。也可参见中的 BR 位。表 14-3 中的 BR 位。

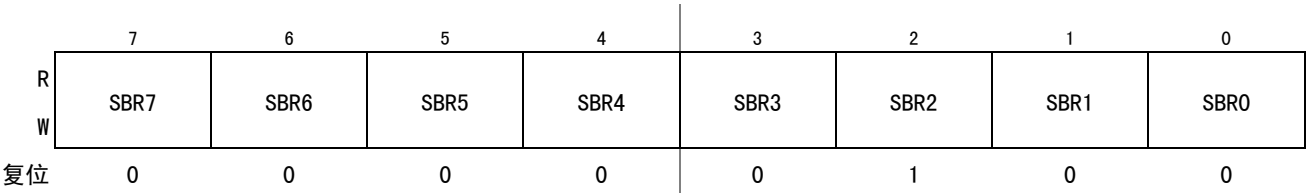


图 14-5. 波特率寄存器 (SCIxBDL)

15.3.1 RTC 状态和控制寄存器 (RTCSC)

RTCSC 包含实时中断状态标记 (RTIF)、时钟选择位 (RTCLKS)、实时中断启动位 (RTIE) 和预分频器选择位 (RTCPs)。

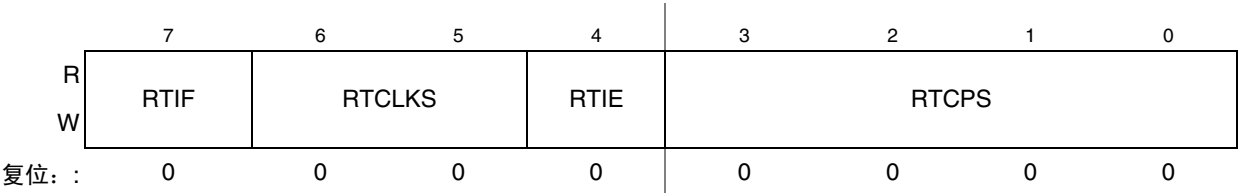


图 15-3. RTC 状态和控制寄存器 (RTCSC)

表 15-2. RTCSC 字段描述

字段	描述
7 RTIF	实时中断标记 — 本状态位指示 RTC 计数器寄存器计数值达到 RTC 模数寄存器中设定的值。写入逻辑数 0 是无效的。写入逻辑数 1 可清除此位和实时中断请求。复位将 RTIF 清除为 0。 0 RTC 计数器未达到 RTC 模数寄存器中规定的值。 1 RTC 计数器已达到 RTC 模数寄存器中规定的值。
6–5 RTCLKS	实时时钟源选择 — 这两个读 / 写位为 RTC 预分频器选择时钟源输入。更换时钟源可清除预分频器和 RTCCNT 计数器。选择时钟源时，确保时钟源被正常使能（如果适用），以确保 RTC 的正确运行。复位将把 RTCLKS 清除为 00。 00 实时时钟源为 1-kHz 低功率振荡器（LPO） 01 实时时钟源为外部时钟（ERCLK） 1x 实时时钟源为内部时钟（IRCLK）
4 RTIE	实时中断使能 — 这个读 / 写位使能实时中断。如果 RTIE 被设置，那么 RTIF 被设置时会生成中断。复位将把 RTIE 清除为 0。 0 实时中断请求被关闭。采用软件轮询。 1 实时中断请求被允许。
3–0 RTCPs	实时时钟预分频器选择 — 这四个读 / 写位为时钟源选择二进制或十进制除数值。请参见 表 15-3. 修改预分频器可清除预分频器和 RTCCNT 计数器。复位将把 RTCPs 清除为 0000。

表 15-3. RTC 预分频器除数值

RTCLKS[0]	RTCPs															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	Off	2 ³	2 ⁵	2 ⁶	2 ⁷	2 ⁸	2 ⁹	2 ¹⁰	1	2	2 ²	10	2 ⁴	10 ²	5x10 ²	10 ³
1	Off	2 ¹⁰	2 ¹¹	2 ¹²	2 ¹³	2 ¹⁴	2 ¹⁵	2 ¹⁶	10 ³	2x10 ³	5x10 ³	10 ⁴	2x10 ⁴	5x10 ⁴	10 ⁵	2x10 ⁵

表 16-3. TPM- 时钟源选择

CLKSB:CLKSA	预分频器输入的 TPM 时钟源
00	未选择时钟（TPM 计数器关闭）
01	总线速率时钟
10	固定系统时钟
11	外部源

表 16-4. 预分频器因子选择

PS2:PS1:PS0	TPM 时钟源除数
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

16.3.2 计数器的寄存器（TPMxCNTH:TPMxCNTL）

两个只读 TPM 计数器寄存器包含 TPM 计数器中值的高低字节。读取任何一个字节（无论是 TPMxCNTH 还是 TPMxCNTL）都会使两个字节的 内容锁入到缓冲器中。这些内容一直被锁定在那里，直到另一半被读取。这允许按从小到大或从大到小的顺序进行连贯的 16 位读取，使它更加友好地适应各种编译器编译。一致性机制可通过 MCU 复位或定时器状态 / 控制寄存器（TPMxSC）写入操作自动重启。

复位可清除 TPM 计数器寄存器。此外，向 TPMxCNTH 或 TPMxCNTL 中写入任何值也可清除 TPM 计数器（TPMxCNTH:TPMxCNTL）并复位一致性机制，而不管写入操作所涉及的数据。



图 16-8. TPM 计数器寄存器高字节（TPMxCNTH）

图 17-3 显示主机从目标 HCS08 MCU 收到逻辑 1。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机保持低 BKGD 管脚足够长的时间，使目标识别它（至少两个目标 BDC 周期）。主机必须在目标 MCU 在其认为的位计时开始后驱动瞬时高态加速脉冲七个周期前，释放低电平驱动。主机应该在其启动位时间约 10 个周期后采样位电平。

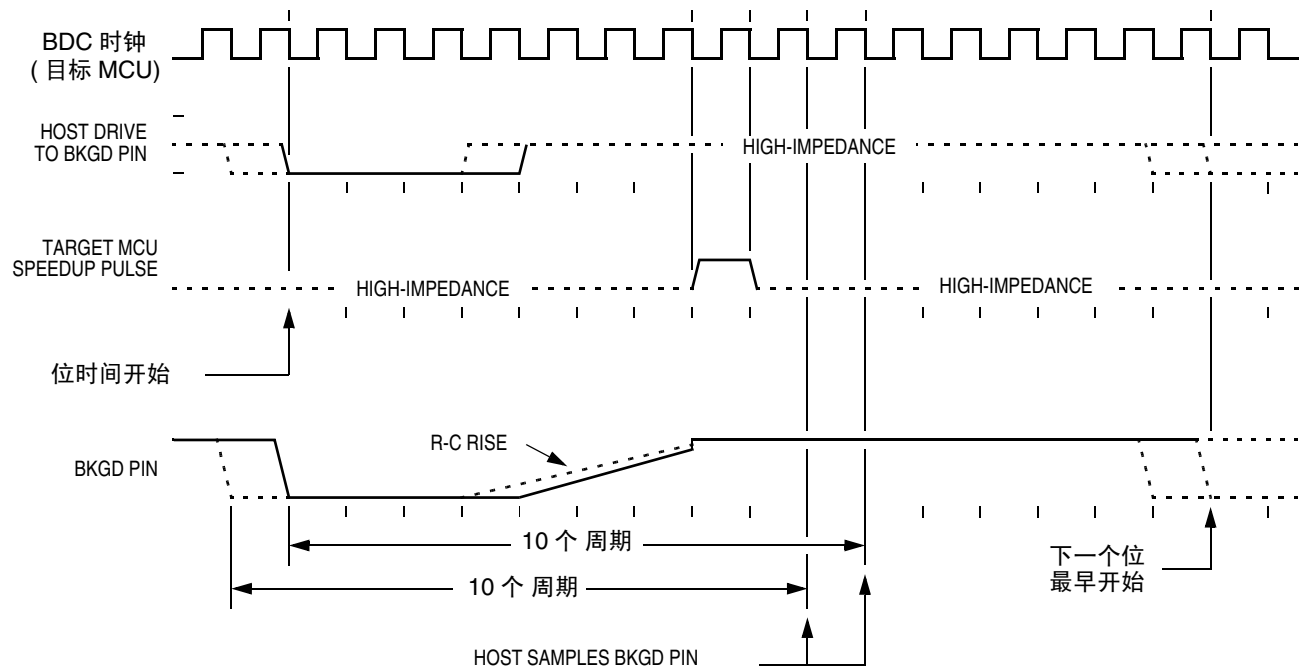


图 17-3. BDC 目标 - 到 - 主机串行位时序 (逻辑 1)

图 17-4 显示了主机从目标 HCS08 MCU 收到逻辑 0。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机启动位时间，但是目标 HCS08 MCU 完成它。由于目标希望主机接收逻辑 0，它保持低 BKGD 管脚 13 个 BKGD 管脚周期，然后驱动引脚置高，加速上升沿。主机在启动位时间约 10 个周期后采样位电平。

上半部分没有被使用，仅仅通过读 DBGFL 来从 FIFO 中读出数据。每次读 DBGFL 时，FIFO 都会移动，这样通过 DBGFL 的 FIFO 数据端口可以获得下一个数据值。

在触发模式中，FIFO 保存流变化地址，CPU 地址与 FIFO 的输入端有一个延迟。由于这个延迟，如果触发事件本身是一个流变化地址或在触发事件启动 FIFO 后下两个周期中出现了流变化地址，它将不保存在 FIFO 中。如果是结束 - 跟踪的情况，当触发事件是一个流变化，则它将保存为运行的调试器的最后一个流变化入口。

当调试器没有打开时，FIFO 还可以用来生成所执行指令地址的分析。当 ARM = 0, 读 DBGFL 会使最近获取的操作码的地址保存在 FIFO 中。采用分析功能，主机调试器将从 FIFO 中读取地址，即以常规的间隔先读 DBGFH 然后读 DBGFL，。前 8 个值将被丢弃，因为它们对应于初始需要填充 FIFO 的 8 个 DBGFL 读取。DBGFH 和 DBGFL 的其它周期读取则返回关于所执行指令的延迟信息，这样主机调试器可以对执行指令地址进行分析。

17.3.3 流变化信息

为了减少存储在 FIFO 中的信息数量，只保存与使正常的指令执行顺序发生变化的指令相关的信息。知道存储在目标系统中的源和对象代码程序后，外部调试器可以通过来自 FIFO 中存储的大量流变化信息的许多指令来重现执行路径。

对于采用了分支的条件分支指令（分支条件为真），则保存源地址（条件分支操作码的地址）。由于 BRA 和 BRN 指令不是条件的，这些事件不会使流变化信息存储在 FIFO 中。

间接 JMP 和 JSR 指令采用 H:X 间址寄存器对的当前内容，确定目的地址，这样调试系统为任何间接 JMP 或 JSR 保存运行时的目的地址。对于中断，RTI 或 RTS, 目的地址作为流变化信息存储在 FIFO 中。

17.3.4 标记 vs. 强制断点和触发器

标记一词指当指令操作码被取到指令队列时识别它，但是不采取任何其它操作，直到且除非指令被 CPU 真正执行。这种区分非常重要，因为任何因跳转、分支、子例程调用、或中断而发生的流变化都会导致一些指令被取到指令队列，未执行就被丢弃。

强制类型的断点等待当前指令完成，然后执行断点请求操作。通常操作是进入激活背景调试模式，而不是继续用户应用程序中的下一个指令。

标记 vs. 强制这一术语在调试模块的两种情况下使用。第一种情况指从调试模块向 CPU 发送断点请求。第二种情况指从比较器向调试控制逻辑发送匹配信号。当标记类断点发送给 CPU 时，信号与操作码一起进入指令队列，这样当这个操作码被执行时，CPU 将有效地用 BGND 操作码代替被标记的操作码，这样 CPU 进入激活背景调试模式，而不是执行被标记的指令。当 DBGTR 寄存器中的 TRGSEL 控制位被设置为选择标记类操作，比较器 A 或 B 的输出被调试模块中的逻辑块鉴定，这个逻辑块跟踪操作码，如果比较地址的操作码被实际执行，则只向该调试器生成一个触发。每个比较器都有单独的操作码跟踪逻辑，这样整个指令队列一次不只一个比较事件被跟踪。

中断标记和启动与 16 位主计数器相关。定时器溢出标记 (TOF) 是一种显示定时器计数器溢出的软件可接入指示。TOF 标记等于 1 时自动生成静态硬件中断的情况下, 启动信号都在软件轮询 (TOIE=0) (无硬件中断生成) 或中断驱动操作 (TOIE=1) 之间选择。

导致 TOF 被设置的条件取决于 (向上或向上 / 向下) 计数模式。在向上计数模式中, 16 位主计数器从 0x0000 计数到 0xFFFF, 并且在下一次计数时钟中溢出到 0x0000。在从 0xFFFF 向 0x0000 过渡时, TOF 被设置。设置了模数限制时, TOF 会在从模数寄存器中设置的值向 0x0000 过渡时设置。当 16 位主计数器以向上 / 向下模式运行的情况下, 计数器在从模数寄存器中设置的值和下一个更低的计数值过渡而改变方向时 TOF 标记被设置。这对应 PWM 周期的结束。(0x0000 计算值对应周期的中央。)

因为 HCS08 MCU 是一种 8 位的架构, 所以一致性机制被设计到定时器计数器中以进行读取操作。当计数器的任何一个字节 (TPMxCNTH or TPMxCNTL) 被读取时, 两个字节都被捕获到缓冲器中, 这样当另一个字节被读取时, 值会显示读取第一个字节时计数的另一个字节。计数器继续正常计数, 但是没有新的值从任意字节中读取, 直到旧的计数的两个字节都被读取。

主定时器的计数器可随时通过将任何值写入 TPMxCNTH 或 TPMxCNTL 计数的任一比特来手动复位。如果在复位计数前只有计数器的一个字节被读取, 那么这种计数器复位方式还会复位一致性机制。

B.6.2 通道模式选择

如果 CPWMS=0 (未规定中央对齐的 PWM 操作), 那么通道 n 状态和控制寄存器中 MSnB 和 MSnA 控制位为相应通道确定基本运行模式。选择包括输入捕获、输出对比或缓冲的边缘对齐 PWM。

B.6.2.1 输入捕获模式

利用输入捕获功能, TPM 可捕获外部事件发生的时间。当输入捕获通道的管脚上发生激活边时, TPM 会将 TPM 计数器的内容锁入到通道值寄存器中 (TPMxCnVH:TPMxCnVL)。上升边、下降边或任何边可被选为触发输入捕获的活动边。

当 16 位捕获寄存器的任何一个字节被读取时, 两个字节都被锁入到缓冲器中, 以支持连贯的 16 位接入而不受顺序的影响。一致性序列可通过向通道状态 / 控制寄存器 (TPMxCnSC) 中写入值来手动复位。

输入捕获事件会设置标记位 (CHnF), 该标记可选择生成一个 CPU 中断请求。

B.6.2.2 输出比较模式

通过输出比较功能, TPM 可生成具有可编程位置、极性、持续时间和频率的定时脉冲。当定时器达到输出对比通道的通道值寄存器中的值时, TPM 可以置 1, 置 0, 翻转引脚状态通道。

在输出比较模式中, 值只有在 16 位寄存器的两个 8 位字节都被写入后被传输到相应定时器的通道寄存器中。这种一致性序列可通过向通道状态 / 控制寄存器 (TPMxCnSC) 中写入值来手动复位。

输出比较事件会设置标记位 (CHnF), 该标记可选择产生 CPU 中断请求。

