



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 <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz60f2vlh">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz60f2vlh</a>

## 第 3 章 操作模式

### 3.1 简介

本章介绍 MC9S08DZ60 系列产品的操作模式，同时描述了如何进入各种模式、如何从各种模式中退出及各种模式下可提供的功能。

### 3.2 特性

- 主动后台模式：用于代码开发
- 等待模式：CPU 关闭以省电；系统时钟正常运行，内部稳压器正常工作
- 停止模式：系统时钟被关闭，内部稳压器处于待机状态
  - Stop3 — 所有内部电路都接通电源以实现快速恢复
  - Stop2 — 内部电路的部分电源被关闭；RAM 内容被保留

### 3.3 运行模式

这是 MC9S08DZ60 系列产品的正常操作模式。当 BKGD/MS 管脚位于复位的上升边最高位置时选择该模式。在这种模式下，CPU 执行内部存储器中的代码。代码在复位完成后运行，并且从内存 0xFFFFE - 0xFFFF 上获取其起始地址。

### 3.4 主动后台模式

主动后台模式功能通过 HCS08 内核中的后台调试控制器（BDC）进行管理。在软件开发过程中，BDC 与片上调试模块（DBG）一起用于分析 MCU 的运行情况。

进入主动后台模式的方式有以下五种：

- 当处于复位的上升沿时，BKGD/MS 脚置于低电平；
- 通过 BKGD/MS 脚收到 BACKGROUND 命令时；
- 执行 BGND 指令时；
- 遇到 BDC 断点时；
- 遇到 DBG 断点时。

进入主动后台模式后，CPU 保持挂起状态，等待串行后台命令而不执行来自用户应用程序的指令。

表 3-1 列出了影响停止模式选择的所有控制位及各种条件下选择的模式。被选择的模式在执行一个 STOP 指令后进入。

表 3-1. 停止模式选择

STOPE	ENBDM <sup>1</sup>	LVDE	LVDSE	PPDC	停止模式
0	x	x	x	x	停止模式被禁用；如果执行了 STOP 指令，则进行非法 Opcode 代码复位
1	1	x	x	x	Stop3 模式，BDM 被启用 <sup>2</sup>
1	0	两个位都必须为 1	x	x	Stop3 模式，电压调节器处于活动状态
1	0	其中一个位为 0	0	0	Stop3 模式
1	0	其中一个位为 0	1	1	Stop2 模式

<sup>1</sup> ENBDM 位于 BDCSCR（只能通过 BDC 命令访问）中。详细信息请参见第 17.4.1.1 部分“BDC 状态和控制寄存器（BDCSCR）”。

<sup>2</sup> 处于 Stop3 模式而且 BDM 被启用时，因为启用了内部时钟，SIDD 将接近 RIDD 水平。

### 3.6.1 Stop3 模式

Stop3 模式通过表 3-1 所述条件下执行 STOP 指令进入。所有此时的内部寄存器和逻辑、RAM 内容和 I/O 管脚状态都被维持。

从 Stop3 模式中退出的操作通过 RESET 管脚或异步中断脚实现。这些异步中断脚包括 IRQ、PIA0 ~ PIA7、PIB0 ~ PIB7 和 PID0 ~ PID7。从 Stop3 模式中退出的操作也可以通过低压检测（LVD）复位、低压警告（LVW）中断、ADC 转换完成中断、实时时钟（RTC）中断、MSCAN 唤醒中断或 SCI 接收器中断完成。

如果通过 RESET 脚的方式退出 Stop3 模式，MCU 将复位，操作将在获取复位向量后恢复。如果通过中断的方式退出，MCU 将获取相应的中断向量。

#### 3.6.1.1 在 Stop3 模式中启用 LVD

在电源电压下降到 LVD 电压以下时，LVD 系统可以生成一个中断或复位。在 CPU 执行 STOP 指令时，如果 LVD 在停止模式下被启用（SPMSC1 中的 LVDE 和 LVDSE 位均被设置），那么内部稳压器在停止模式下将继续保持活动状态。

要使 ADC 正常运行，LVD 必须在进入 Stop3 时处于启用状态。

#### 3.6.1.2 在 Stop3 模式中启用活动 BDM

如果 BDCSCR 中设置了 ENBDM，将启用从运行模式进入主动后台模式的操作。该寄存器在第 17 章，“开发支持”。中有详细描述。如果 CPU 执行 STOP 指令时设置了 ENBDM，连接到后台调试逻辑的系统时钟将在 MCU 进入停止模式时继续保持活动状态。因此，这种情况下后台调试通信仍可进行。此外，内部稳压器不会进入低功耗待机状态，而保持正常工作。

大多数后台命令在停止模式不能使用。带状态内存访问命令不允许内存访问，但它们会上报错误，指出 MCU 处于停止或等待模式。可以使用后台命令将 MCU 从停止模式中唤醒并进入主动后台模式（如果 ENBDM 位已设置）。进入后台调试模式后，所有后台命令都可以使用。

表 3-2. 停止模式

外围设备	模式	
	Stop2	Stop3
CPU	关闭	待机
RAM	待机	待机
Flash/EEPROM	关闭	待机
并行端口寄存器	关闭	待机
ACMP	关闭	关闭
ADC	关闭	可选择开启 <sup>1</sup>
IIC	关闭	待机
MCG	关闭	可选择开启 <sup>2</sup>
MSCAN	关闭	待机
RTC	可选择开启 <sup>3</sup>	可选择开启 <sup>3</sup>
SCI	关闭	待机
SPI	关闭	待机
TPM	关闭	待机
电压调节器	关闭	可选择开启 <sup>4</sup>
XOSC	关闭	可选择开启 <sup>5</sup>
I/O 管脚	状态被保持	状态被保持
BDM	关闭 <sup>6</sup>	可选择开启
LVD/LVW	关闭 <sup>7</sup>	可选择开启

<sup>1</sup> 要求启用 DC 时钟和 LVD，否则为待机。

<sup>2</sup> MCGC1 中设置 IRCLKEN 和 IREFSTEN，否则为待机。

<sup>3</sup> 要求启用 RTC，否则为待机。

<sup>4</sup> 要求启用 LVD 或 BDC。

<sup>5</sup> MCGC2 中设置 ERCLKEN 和 EREFSTEN，否则为待机。在高频率范围（MCGC2 中设置 RANGE）还要求在 Stop3 中启用 LVD。

<sup>6</sup> 如果进入 Stop2 模式时设置了 ENBDM，MCU 实际上会进入 Stop3 模式。

<sup>7</sup> 如果进入 Stop2 模式时设置了 LVDSE，MCU 实际上会进入 Stop3 模式。

### 4.3 寄存器地址和位分配

MC9S08DZ60 系列产品中的寄存器可分为以下几组：

- 直接页面寄存器，位于存储器映象的前 128 个位置上。这些寄存器可以通过高效的直接寻址模式指令访问。
- 高端页面（High-page）寄存器，不经常使用，因此位于存储器映象中 0x1800 以上。在直接页面寄存器中为经常使用的寄存器和 RAM 留出了更多空间。
- 非易失性寄存器，由 Flash 中 0xFFB0 ~ 0xFFBF 之间 16 个位置组成的位置段组成。非易失性寄存器位置包括：
  - NVPROT 和 NVOPT，在复位时上载到工作寄存器中。
  - 一个 8 字节后门对比密钥，可选择为用户分配有控制的安全内存访问权限。

由于非易失性寄存器的位置是在 Flash 中，所以必须像其他位置 Flash 一样擦除和编程。

直接页面寄存器可以通过高效的直接寻址模式指令访问。位操作指令可用于访问任何直接页面寄存器中的任何位。表 4-2 总结了所有用户可访问的直接页面寄存器和控制位。

表 4-2 所列的直接页面寄存器可以使用更高效的直接寻址模式（这种模式只需要地址的较低字节）。因此，第 1 栏中地址的较低字节用粗体显示。在表 4-3 和表 4-5 中，第 1 栏中的整个地址都用粗体显示。在表 4-2, 表 4-3, 和表 4-5 中，第 2 栏中的寄存器名称用粗体显示以便与右侧的位名称区分。与所列出的位不相关的单元在阴影中显示。带有 0 的阴影单元表示这个未使用的位始终应为 0。带有破折号的阴影单元表示未使用的或预留的位，可以是 1 或 0。

表 4-3 中列出的高端页面寄存器的访问频率比其它 I/O 和控制寄存器低很多，因此存放在可直接寻址的内存空间外，从 0x1800 开始。

表 4-3. 高端页面寄存器总结（第 1 页，共 3 页）

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
0x1800	SRS	POR	PIN	COP	ILOP	ILAD	LOCS	LVD	0
0x1801	SBDFR	0	0	0	0	0	0	0	BDFR
0x1802	SOPT1	COPT		STOPE	SCI2PS	IICPS	0	0	0
0x1803	SOPT2	COPCLKS	COPW	0	ADHTS	0	MCSEL		
0x1804– 0x1805	预留	—	—	—	—	—	—	—	—
0x1806	SDIDH	—	—	—	—	ID11	ID10	ID9	ID8
0x1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x1808	预留	—	—	—	—	—	—	—	—
0x1809	SPMSC1	LVWF	LVWACK	LVWIE	LVDRE	LVDSE	LVDE	0	BGBE
0x180A	SPMSC2	0	0	LVDV	LVWV	PPDF	PPDACK	0	PPDC
0x180B– 0x180F	预留	—	—	—	—	—	—	—	—
0x1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
0x1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
0x1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
0x1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
0x1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
0x1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
0x1816	DBGC	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
0x1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
0x1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
0x1819– 0x181F	预留	—	—	—	—	—	—	—	—
0x1820	FCDIV	DIVLD	PRDIV8	DIV					
0x1821	FOPT	KEYEN	FNORED	EPGMOD	0	0	0	SEC	
0x1822	预留	—	—	—	—	—	—	—	—
0x1823	FCNFG	0	EPGSEL	KEYACC	Reserved <sup>1</sup>	0	0	0	1
0x1824	FPROT	EPS			FPS				
0x1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
0x1826	FCMD	FCMD							
0x1827– 0x183F	预留	—	—	—	—	—	—	—	—
0x1840	PTAPE	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
0x1841	PTASE	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
0x1842	PTADS	PTADS7	PTADS6	PTADS5	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
0x1843	预留	—	—	—	—	—	—	—	—
0x1844	PTASC	0	0	0	0	PTAIF	PTAACK	PTAIE	PTAMOD

### 7.4.3 等待模式操作

WAIT 指令通过清除 CCR 中的 I 位来使能中断。然后它暂停 CPU 时钟，以减少整体功耗，CPU 此时正在等待将把 CPU 从等待模式唤醒的中断或复位事件。当发生中断或复位事件时，CPU 时钟会重新开始工作，中断或复位事件被正常处理。

如果串行 BACKGROUND 命令通过背景调试接口发送到 MCU，与此同时 CPU 处于等待模式，那么 CPU 时钟会重新开始工作，CPU 进入活动后台模式，可以处理其他串行后台命令。这样就确保了即使主机开发系统处于等待模式，它仍能访问目标 MCU。

### 7.4.4 停止模式操作

在通常情况下，包括晶体振荡器（使用时）在内的所有系统时钟在停止模式中都会暂停，以最大限度地减少功耗。在这类系统中，需要外部电路来控制停止模式所花费的时间，并且在需要重新开始处理时发出一个信号来唤醒目标 MCU。与早期 M68HC05 和 M68HC08 MCU 不同的是，HCS08 可以在经过配置后使停止模式以最少的时钟运行。这同样允许内部周期信号将 MCU 从停止模式唤醒。

如果主机调试系统与背景调试管脚（BKGD）连接，且串行命令通过后台接口设置了 ENBDM 控制位（或者因为 MCU 被重置为活动后台模式），那么当 MCU 进入停止模式时，振荡器就被迫保持活动状态。这时，当通过背景调试接口将串行 BACKGROUND 命令发送到 MCU，而与此同时 CPU 处于停止模式时，CPU 时钟会重新开始工作，CPU 进入可以处理其他串行后台命令的活动后台模式。这样就确保了即使 MCU 处于等待模式主机开发系统仍能访问目标 MCU。

停止模式恢复取决于特殊 HCS08 及振荡器是否在停止模式中停止。更多信息请参见“运行模式”。

### 7.4.5 BGND 指令

相对于 M68HC08，BGND 指令是 HCS08 的新增指令。普通用户程序中不会使用 BGND，因为它强制 CPU 停止处理用户指令而进入活动后台模式。重新执行用户程序的唯一方法是通过复位，或由主机调试系统通过背景调试接口发出 GO、TRACE1 或 AGGO 串行命令。

基于软件的断点可以通过用 BGND 操作码在所需断点地址上取代操作码的方式进行设置。当程序到达该断点地址时，CPU 会被迫进入活动后台模式，而不是继续用户程序。

### 8.6.2.3 示例 3: 从 BLPI 转换到 FEE 模式: 外部晶体 = 4 MHz、总线频率 = 16 MHz

本例中, MCG 将选择适当的运行模式, 从以基于内部参考时钟, 运行于 16 kHz 总线频率的 BLPI 模式 (参见前例) 转换到 4MHz 晶体频率、16 MHz 总线频率的 FEE 模式。示例中首先介绍了代码序列, 然后提供了一个演示该顺序的流程图。

1. 首先, BLPI 必须转换到 FBI 模式。
  - a) MCGC2 = 0x00 (%00000000)
    - MCGSC 中的 LP (位 3) 是 0
  - b) 循环检测, 直到 MCGSC 中的 LOCK (位 6) 置位, 表明 FLL 已经获得锁定。尽管在 FBI 模式中 FLL 被旁通, 但它仍使能并运行。
2. 接下来, FBI 将转换到 FEE 模式。
  - a) MCGC2 = 0x36 (%00110110)
    - RANGE (位 5) 设置为 1, 因为 4 MHz 频率在高频范围内。
    - HGO (位 4) 设置为 1, 为高增益运行配置外部振荡器。
    - EREFS (位 2) 设置为 1, 因为正在使用晶体。
    - ERCLKEN (位 1) 设置为 1, 确保外部参考时钟处于活动状态。
  - b) 循环检测, 直到 MCGSC 中的 OSCINIT (位 1) 是 1, 表明 EREFS 位选择的晶体已经完成初始化。
  - c) MCGC1 = 0x38 (%00111000)
    - CLKS (位 7 和 6) 设置为 %00, 以便将 FLL 输出选为系统时钟源。
    - RDIV (位 5-3) 设置为 %111 或除以 128, 因为  $4 \text{ MHz} / 128 = 31.25 \text{ kHz}$ , 这在 FLL 要求的 31.25 kHz -- 39.0625 kHz 频率范围内。
    - IREFS (位 1) 清除至 0, 选择外部参考时钟
  - d) 循环检测, 直到 MCGSC 中的 IREFST (位 4) 是 0, 表明外部参考时钟是参考时钟的当前源。
  - e) 循环检, 直到 MCGSC 中的 LOCK (位 6) 置位, 表明 FLL 重新获得了锁定。
  - f) 循环检测, 直到 MCGSC 中的 CLKST (位 3 和 2) 是 %00, 表明已经选择 FLL 输出为 MCGOUT 馈电。

## 10.5 功能描述

复位期间或当 ADCH 位都高时，ADC 模块禁止。当已经完成当前转换，而下一次转换还未发起时，模块进入空闲状态。空闲时，模块处于最低功耗状态。

ADC 可以对软件选择的任意通道实施模数转换。在 12 位和 10 位模式中，所选的通道电压通过逐次逼近算法被转换成 12 位数字结果。在 8 位模式中，所选的通道电压通过逐次逼近算法被转换成 9 位数字结果。

转换完成后，结果保存在数据寄存器（ADCRH 和 ADCRL）中。在 10 位模式中，结果被圆整到 10 位并保存在数据寄存器（ADCRH 和 ADCRL）中。在 8 位模式中，结果被圆整到 8 位并保存在 ADCRL 中。然后设置转换完成标记（COCO），如果已经使能了转换完成中断（AIEN = 1），则触发中断。

ADC 模块能够自动地把转换结果与比较寄存器的内容进行比较。通过设置 ACFE 位并结合任意一种转换模式和配置一起运行，就使能了比较功能。

### 10.5.1 时钟选择和分频控制

ADC 模块可选择 4 个时钟源之一，然后由可配置值进行分频，生成转换器的输入时钟（ADCK）。时钟源通过 ADICLK 位设置从以下源中选择。

- 总线时钟，等于软件运行的频率。这是复位后的默认选择。
- 总线时钟除以 2，如果总线时钟很高，允许总线时钟最大除以 16。
- ALTCLK，由此 MCU 定义（参见本章节的概述部分）。
- 异步时钟（ADACK）- 该时钟从 ADC 模块内部时钟源产生。当 MCU 处于等待或 STOP3 模式时，此时钟源仍然有效，从而实现此模式下的低噪音转换。

无论选择哪种时钟，其频率必须在 ADCK 的指定频率范围内。如果可用时钟太慢，ADC 将无法保证正常运行。如果可用时钟太快，那么时钟必须分频为适当的频率。除数由 ADIV 位指定，可以是 1、2、4 或 8。

### 10.5.2 输入选择和管脚控制

管脚控制寄存器（APCTL3，APCTL2 和 APCTL1）用来禁止对作为模拟输入的管脚的 I/O 控制功能。当置位管脚控制寄存器相应位时，对应的 MCU 管脚进入以下状态：

- 输出缓冲器进入高阻抗状态。
- 输入缓冲器禁止。对于其输入缓冲器被禁止的任何管脚，I/O 端口读数均返回 0。
- 上拉禁止。

### 10.5.3 硬件触发

ADC 模块有一个可选的异步硬件转换触发 ADHWT，当设置了 ADTRG 位时，ADHWT 使能。并不是所有 MCU 上都有这个源。如需了解该 MCU 的特定 ADHWT 源的更多报文，请参见本章概述部分。

可能会出现外部系统工作造成辐射或传导噪音，或者过多  $V_{DD}$  噪音被耦合到 ADC 的情况。在这些情况下，或者当 MCU 不能置于等待或 STOP3 模式，或 I/O 跳变不能暂停时，下面这些推荐操作会降低噪音对精度的影响：

- 将所选输入通道与  $V_{REFL}$  或  $V_{SSAD}$  之间放置  $0.01 \mu F$  电容器 ( $C_{AS}$ )，（这将改善噪音问题，但仍会影响采样速率，基于模拟源的阻抗）。
- 通过对连续多次转换结果求算术平均。要消除 1LSB，一次采样的错误，需要 4 次转换。
- 通过关闭异步时钟 (ADACK) 和算术平均可以降低同步噪音的影响。与 ADCK 同步的噪音无法通过平均消除。

#### 10.7.2.4 编码宽度和量化错误

ADC 将理想的直线传递函数量化为 4096 个步骤（在 12 位模式中）。每个步骤刚好都有相同的高度（1 个代码）和宽度。宽度的定义是两个代码转换点间的  $\delta$ 。N 位转换器（本例中，N 可以为 8、10 或 12）理想的代码宽度是（就像 1LSB 定义的那样）：

$$1 \text{ lsb} = (V_{REFH} - V_{REFL}) / 2^N \quad \text{等式 10-2}$$

由于结果的数字化原因，可能会出现固有的量化错误。对于 8 位或 10 位转换，当电压位于两点间的中点时，代码就会进行切换，此时直线传递函数与实际的传递函数相一致。因此，在 8 位或 10 位模式中，量化错误是  $\pm 1/2 \text{ lsb}$ 。但是，由此导致的第一次转换 (0x000) 的代码宽度为  $1/2 \text{ lsb}$ ，而最后一次转换 (0xFF or 0x3FF) 的代码宽度为  $1.5 \text{ lsb}$ 。

对于 12 位转换，代码仅在整个代码宽度出现后才进行转换，因此量化错误在  $-1 \text{ lsb}$  至  $0 \text{ lsb}$  之间，每个步骤的代码宽度均为  $1 \text{ lsb}$ 。

#### 10.7.2.5 线性错误

ADC 还可能呈现几种形式的非线性。大量的努力被用来减少这些错误，但系统应知道这些错误，因为它们会影响总体精度。这些错误包括：

- 零刻度错误 ( $E_{ZS}$ )（有时称作偏移）-- 这个错误的定义是第一次转换的实际代码宽度和理想代码宽度（在 8 位/10 位中为  $1/2 \text{ lsb}$ ，在 12 位模式中为  $1 \text{ LSB}$ ）之间的差异。注意，如果第一次转换是 0x001，那就使用实际的 0x001 代码宽度及其理想宽度（ $1 \text{ LSB}$ ）之间的差异。
- 满刻度错误 ( $E_{FS}$ )—这个错误的定义是最后一次转换的实际代码宽度和理想代码宽度（在 8 位/10 位中为  $1.5 \text{ lsb}$ ，在 12 位模式中为  $1 \text{ LSB}$ ）之间的差异。注意，如果最后一次转换是 0x3FE，那么就使用实际的 0x3FE 代码宽度及其理想宽度（ $1 \text{ LSB}$ ）之间的差异。
- 差分非线性错误 (DNL)—这个错误的定义是所有转换的实际代码宽度和理想代码宽度之间的最大差异。
- 积分非线性错误 (INL)—这个错误的定义是 DNL 求和达到的最高值（绝对值）。更简单地说，对于所有代码来说，这是给定代码的实际转换电压和相应的理想转换电压之间的最大差异。
- 总不可调整错误 (TUE)—这个错误的定义是实际转换功能和理想直线转换功能之间的差异，因此包括所有形式的错误。

**注意**

当初始化模式处于有效状态时，CANTFLG 寄存器保持复位状态 (INITRQ=1, INITAK=1)。当未处于初始化模式时，该寄存器可以写入 (INITRQ = 0 and INITAK = 0)。

读取：任何时间

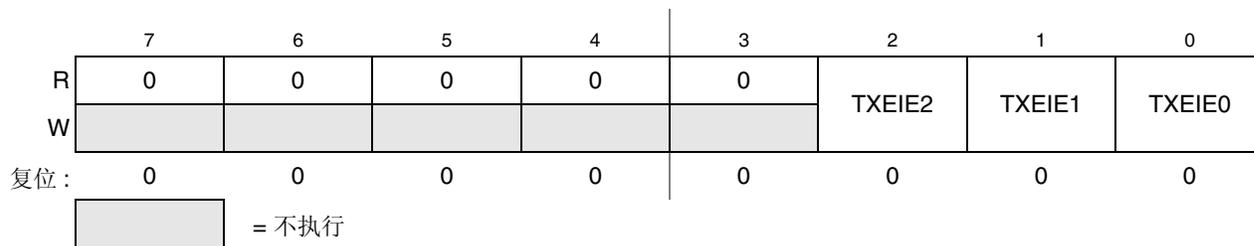
写入：TXEx 标志不处于初始化模式的任何时间；写入 1 清除标志，写入 0 忽略标志。

**表 12-11. CANTFLG 寄存器字段描述**

字段	描述
2:0 TXE[2:0]	<p>发送器缓冲器空—该标志表示相关发送报文缓冲器空，因此没有安排用于发送。在发送缓冲器中放好报文并准备好发送后，CPU 必须清除该标志。报文发送成功后，MSCAN 设置该标志。当发送请求由于中止请求而被成功中止时，MSCAN 也设置该标志（参见 12.3.8，“MSCAN Transmitter 发送器报文中止请求寄存器 (CANTARQ)”）。如果未被屏蔽，当设置了该标志时产生发送中断。</p> <p>清除 TXEx 标志也会清除相应的 ABTAKx（参见 12.3.9，“MSCAN 发送器报文中止确认寄存器 (CANTAACK)”）。当设置了 TXEx 标志时，相应的 ABTRQx 位被清除（参见 12.3.8，“MSCAN Transmitter 发送器报文中止请求寄存器 (CANTARQ)”）。</p> <p>当监听模式处于有效状态时（参见 12.3.2，“控制寄存器 1 (CANCTL1)”）TXEx 标志不能清除且不进行任何发送。</p> <p>当相应的 TXEx 位被清除 (TXEx = 0) 且缓冲器被安排用于发送时，对发送缓冲器的读写操作会被拦截。</p> <p>0 相关报文缓冲器已满（加载了准备发送的报文） 1 相关报文缓冲器空（未安排）</p>

### 12.3.7 MSCAN 发送器中断使能寄存器 (CANTIER)

该寄存器包含发送缓冲器空中断标志的中断使能位。



**图 12-11. MSCAN 发送器中断使能寄存器 (CANTIER)**

**注意**

当初始化模式处于有效状态时，CANTIER 寄存器保持复位状态 (INITRQ=1, INITAK=1)。当未处于初始化模式时，该寄存器可以写入 (INITRQ=0, INITAK=0)。

读取：任何时间

写入：未处于初始化模式的任何时间

### 12.5.3.2 协议违反保护

MSCAN 能够防止用户由于编程错误而意外违反 CAN 协议。保护逻辑实施以下功能：

- 接收和发送错误计数器不能写入或以别的方式操作。
- 当 MSCAN 在线时，控制 MSCAN 的配置的所有寄存器均不能被修改。MSCAN 必须处于初始化模式。CANCTL0/CANCTL1 寄存器中的相应 INITRQ/INITAK 握手位（参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”）作为一个锁来保护以下寄存器：
  - MSCAN 控制 1 寄存器 (CANCTL1)
  - MSCAN 总线定时寄存器 0 和 1 (CANBTR0, CANBTR1)
  - MSCAN 标识符接收控制寄存器 (CANIDAC)
  - MSCAN 标识符接收寄存器 (CANIDAR0 - CANIDAR7)
  - MSCAN 标识符掩码寄存器 (CANIDMR0 - CANIDMR7)
- 当 MSCAN 进入节电模式或初始化模式时，TXCAN 管脚立即被强制进入隐性状态（参见 12.5.5.6，“MSCAN 断电模式”和 12.5.5.5，“MSCAN I 初始化模式”）。
- MSCAN 使能位 (CANE) 在正常系统操作模式下只能写入一次，从而为意外禁止 MSCAN 提供了进一步保护。

### 12.5.3.3 时钟系统

图 12-42 显示 MSCAN 时钟发生电路的结构。

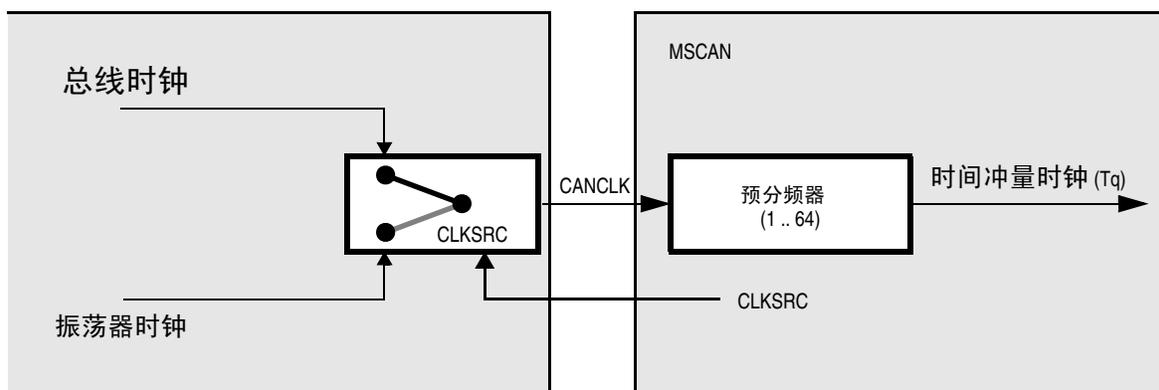


图 12-42. MSCAN 时钟机制

CANCTL1 寄存器 (12.3.2/-214) 中的时钟源位 (CLKSRC) 决定内部 CANCLK 是连接到晶体振荡器（振荡器时钟）输出还是连接到总线时钟。

必须选择能满足 CAN 协议的振荡器精度要求（高达 0.4%）的时钟源。此外，对于高 CAN 总线速率（1 Mbps）来说，要求 45%-55% 的时钟占空比。

如果总线时钟从 PLL 中生成，由于抖动，建议选择振荡器时钟而不是总线时钟，特别是以较快的 CAN 总线速率时。PLL 锁可能太宽，不能确保所需的时钟精度。

对于那些没有时钟和复位发生器 (CRG) 的微控制器，CANCLK 的驱动则来自晶体振荡器（振荡时钟）。

表 13-6. SPI 波特率系数

SPR2:SPR1:SPR0	速率系数
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

### 13.4.4 SPI 状态寄存器 (SPIS)

该寄存器有 3 个只读状态位。位 6, 3, 2, 1 和 0 不执行，始终读 0。写没有意义或影响。

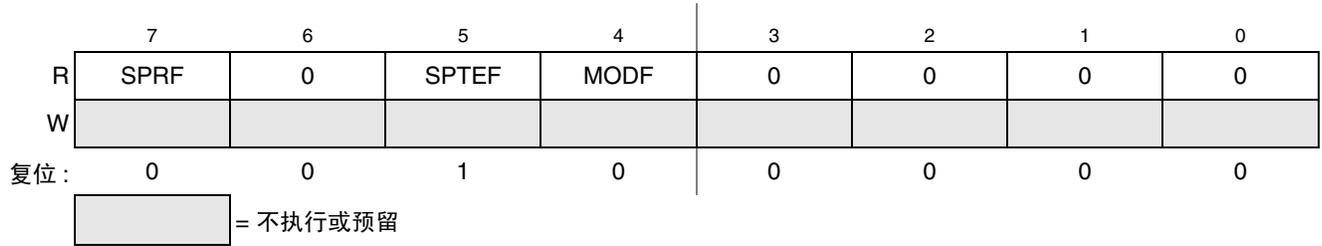


图 13-8. SPI 状态寄存器 (SPIS)

表 13-7. SPIS 寄存器字段描述

字段	描述
7 SPRF	<b>SPI 读缓冲器已满标记</b> — 在完成 SPI 传输时设置 SPRF，显示所接收数据可以从 SPI 数据寄存器 (SPID) 中读取。在设置 SPRF 的同时读取它可以清除 SPRF，然后读 SPI 数据寄存器。 0 接收数据缓冲器中无数据可用 1 接收数据缓冲器中有数据
5 SPTEF	<b>SPI 发送缓冲器空标记</b> — 该位在发送数据缓冲器中有空间时设置。通过读取已设置了 SPTEF 的 SPIS 可以清除该位，然后将数据值写入 SPID 上的发送缓冲器。在向 SPID 写入数据前，SPIS 必须被读到 SPTEF=1 标记，否则写 SPID 将被忽略。如果还设置了 SPIC1 中的 SPTIE 位，SPTEF 生成 SPTEF CPU 中断请求。当数据字节从发送缓冲器传输到发送移位寄存器时，SPTEF 被自动设置。对于闲置 SPI (发送缓冲器或移位寄存器中没有数据，且没有正在进行的传输) 来说，写入 SPID 的数据会被立即传输到移位器，这样 SPTEF 就在两个总线周期内被设置，从而允许第二个 8 位数据值排队进入发送缓冲器。转换寄存器中的值传输完成后，来自发送缓冲器的排队值将自动移到移位器，同时设置 SPTEF，显示发送缓冲器为空。如果发送缓冲器中没有新数据在等待，SPTEF 只需保持设置，数据不会从缓冲器移位到移位器。 0 SPI 发送缓冲器不空 1 SPI 发送缓冲器空
4 MODF	<b>主模式故障标记</b> — 如果 SPI 配置为主 SPI 且从选择输入进入低态，就设置 MODF，显示其他一些 SPI 器件也配置为主 SPI。只有当 MSTR = 1、MODFEN = 1、SSOE = 0 时，SS 管脚才作为模式故障错误输入。否则，将永远不会设置 MODF。当 MODF 为 1 时读 MODF 可以清除 MODF，然后写入 SPI 控制寄存器 1 (SPIC1)。 0 无模式故障错误 1 检测到模式故障错误

表 14-5. SC1xC2 字段说明 (continued)

字段	描述
2 RE	接收器使能 — 当 SCI 接收器关闭时, RxD 管脚恢复为通用端口 I/O 管脚。如果 LOOPS = 1, RxD 管脚恢复为通用 I/O 管脚, 即使 RE = 1。 0 接收器关闭。 1 接收器打开。
1 RWU	接收器唤醒控制 — 该位可以写入 1, 将 SCI 接收器置于待机状态, 等待所选唤醒条件的自动硬件检测。唤醒条件既可以是信息间的闲置线路 (WAKE = 0, 闲置线路唤醒), 也可以是某个字符中最高数据位中的逻辑 1 (WAKE = 1, 地址标记唤醒)。应用软件设置 RWU, (正常情况下) 且所选的硬件条件自动清除 RWU。如需了解更多信息, 14.3.3.2, “接收器唤醒操作”。 0 正常的 SCI 接收器运行。 1 处于待机状态的 SCI 接收器等待唤醒条件。
0 SBK	发送中止字符 — 先后将 1 和 0 写入 SBK, 即在发送数据流中排入了一个中止字符。只要 SBK=1, 多达 10 或 11 (如果 BRK13 = 1, 则为 13 或 14 位) 位时间的逻辑 0 中止字符被加入队列。根据当前正在发送信息有关的 SBK 的设置和清除时间, 第二个中止字符可以在软件清除 SBK 前排入队列。如需了解更多信息, 14.3.2.1, “发送中断和排队闲置”。 0 正常的发射器运行。 1 将发送的队列中止字符。

## 14.2.4 SCI 状态寄存器 1 (SC1xS1)

该寄存器有 8 种只读状态标记。写没有影响, 特殊软件顺序 (不包括写入该寄存器) 用来清除这些状态标记。

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
复位	1	1	0	0	0	0	0	0

 = 不执行或预留

图 14-8. SCI 状态寄存器 1(SC1xS1)

表 14-6. SC1xS1 字段描述

字段	描述
7 TDRE	发送数据寄存器空标记 — TDRE 设置于复位, 当发送数据值从发送数据缓冲器传输到发送移位器时, 就在缓冲器中为新字符留出空间。要清除 TDRE, 当 TDRE = 1 时读 SC1xS1, 然后写入 SCI 数据寄存器 (SC1xD)。 0 发送数据寄存器 (缓冲器) 已满。 1 发送数据寄存器 (缓冲器) 为空。
6 TC	发送完成标记 — TC 设置于复位, 当 TDRE = 1 时, 无数据、前导信号或中止字符在发送。 0 发射器活动 (发送数据、前导信号或中止字符)。 1 发射器闲置 (发送活动完成) 当 TC = 1 时读取 SC1xS1 时可以自动清除 TC, 然后进行以下三种操作中一种: <ul style="list-style-type: none"> <li>• 写入 SCI 数据寄存器 (SC1xD), 以发送新数据</li> <li>• 通过把 TE 从 0 变为 1, 排队前导信号</li> <li>• 将 1 写入 SC1xC2 中的 SBK, 排队中止字符。</li> </ul>

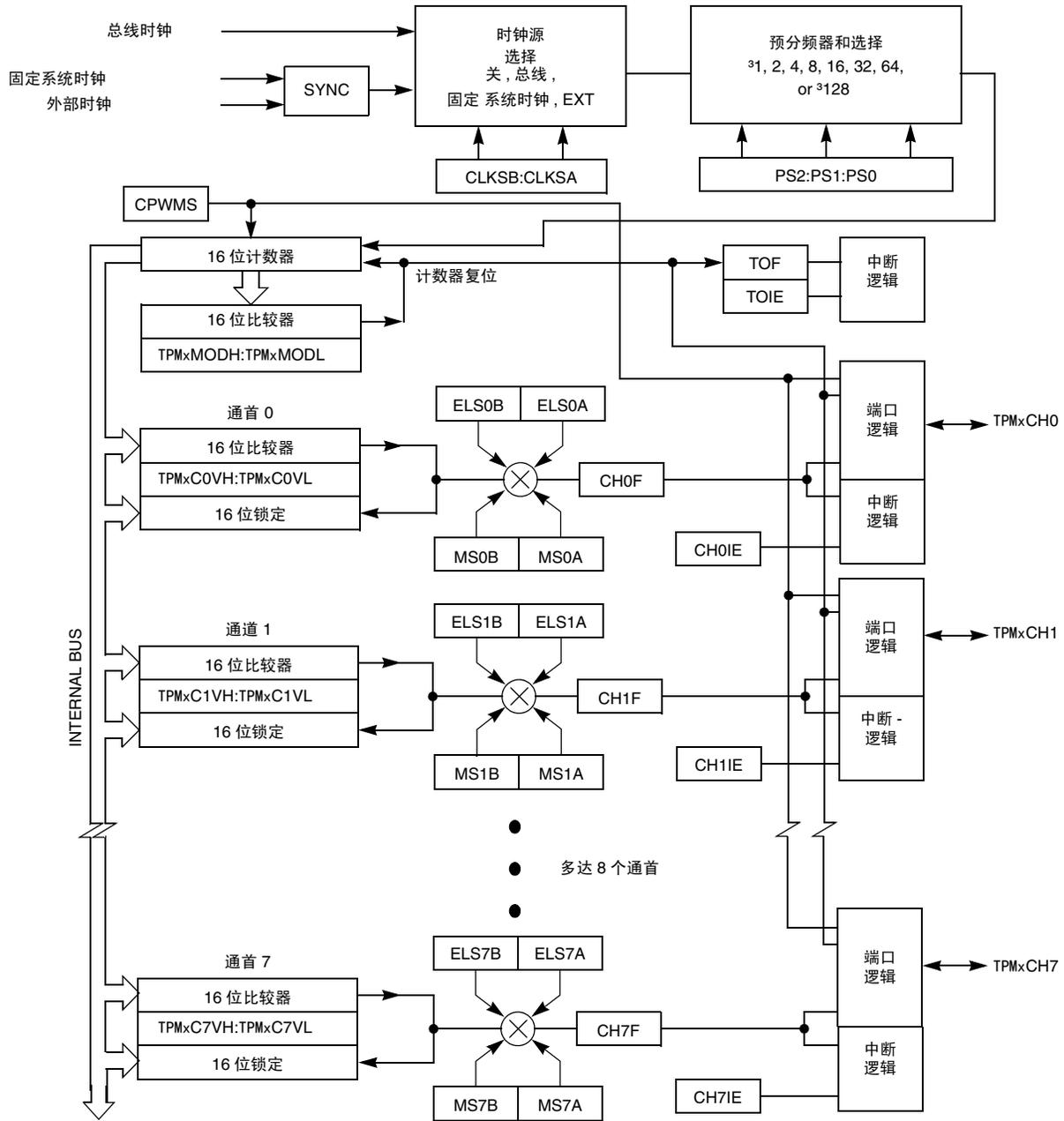


图 16-2. TPM 结构图

## 17.2 背景调试控制器 (BDC)

HCS08 系列中的所有 MCU 都包含一个单线背景调试接口，它支持片上非易失性存储器的在线编程和先进的非侵入式调试功能。与早期的 8- 位 MCU 的调试接口不同，这个系统不干扰正常的应用资源。它不使用任何用户存储器或存储器映射中的地址，也不分享任何片上外设。

BDC 命令分为两大组：

- 激活背景调试模式命令要求目标 MCU 处于激活背景调试模式(用户程序未运行)。激活背景调试模式命令允许读写 CPU 寄存器，允许用户一次跟踪一个用户指令，或从激活背景调试模式进入用户程序。
- 非侵入式命令可以随时执行，即使用户的程序正在运行。  
非侵入式命令允许用户在背景调试控制器中读写 MCU 存储器位置或存取状态和控制寄存器。

一般地，可以用相当简单的接口盒将来自主机的命令转换为与单线背景调试系统连接所需的串行命令。根据开发工具供应商的不同，这个接口盒可以采用标准 232 串行端口，或是并行打印端口，或是其它类型的通信端口，如用来与 PC 通信的 USB 接口。这个接口盒一般通过接地、BKGD 管脚、RESET，有时还有 VDD 与目标系统连接。RESET 引脚的开漏连接允许主机强制目标系统复位，这有助于重新获得对已失目标系统的控制，或在片上非易失性存储器重新编程之前，控制目标系统的启动。有时可以用 VDD 来允许接口盒使用目标系统的电源，避免再使用一个电源。但是，如果单独对接口盒供电，它可以连接到一个正在运行的目标系统，而不必强制目标系统复位，否则会干扰正在运行的应用程序。

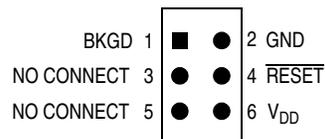


图 17-1. 工具接口

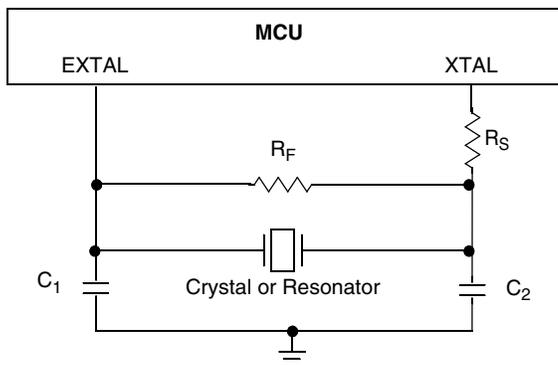
### 17.2.1 BKGD 管脚描述

BKGD 是单线背景调试接口管脚。这个管脚的主要功能是实现激活背景调试模式命令和数据的单向串行通信。在复位过程中，这个管脚用来选择激活背景调试模式启动或启用用户的应用程序。这个管脚还用来请求定时同步响应脉冲，允许主机开发工具确定背景调试串行通信的正确时钟频率。

BDC 串行通信采用首先引入在微处理器 M68HC12 系列上的定制串行协议。这个协议假定主机知道通信时钟速率，这个速率按目标 BDC 所有通信通过主机启动和控制所有通信，主机驱动高到低边沿发出每个位时间开始信号。命令和数据以最重要的位先发 (MSB 先发) 的方式发送。有关通信协议的详细信息，请参见 17.2.2，“通信详细介绍”。

如果主机尝试与 BDC 时钟速率未知的目标 MCU 沟通，可以发送 SYNC 命令给目标 MCU，请求定时同步响应信号，通过这个信号，主机可以判断正确的通信速率。

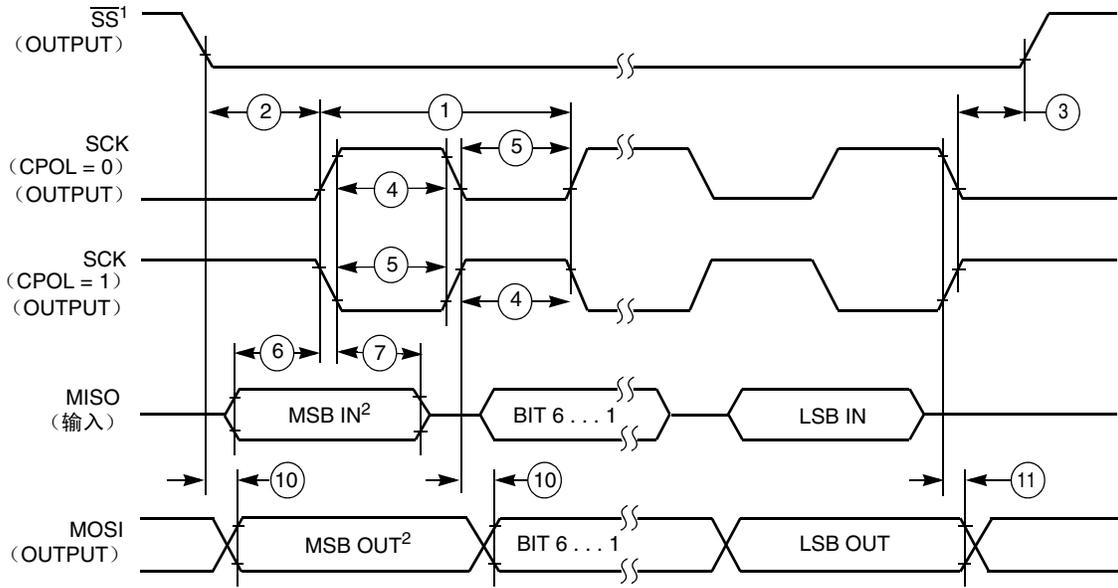
BKGD 是伪开漏管脚，有一个片上上拉，因此不需要外部上拉电阻。与典型的开漏管脚不同，管脚上的外部 RC 时间常数（受外部容性的影响），在信号上升时间上几乎不起作用。定制协议提供瞬态加速脉冲，强制提高这个管脚的上升时间，而没有驱动电平冲突风险。参见 17.2.2，“通信详细介绍”，了解更多详情。



## A.11 MCG 规范

表 A-12. CG 频率规范（温度范围 = -40 至 125 °C）

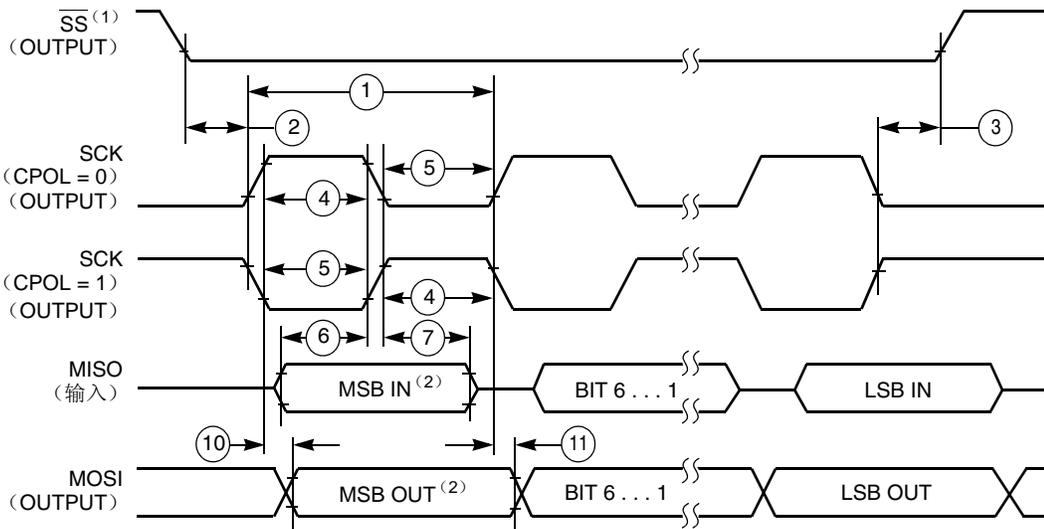
编号	C	参数	符号	最小值	典型值	最大值	单位
1	P	当 $V_{DD} = 5\text{ V}$ 、温度 = 25 °C 时的工厂调整值	$f_{\text{int\_ft}}$	—	31.25	—	kHz
2	P	平均内部参考频率 - 未调整 <sup>1</sup>	$f_{\text{int\_ut}}$	25	32.7	41.66	kHz
3	P	平均内部参考频率 - 用户调整	$f_{\text{int\_t}}$	31.25	—	39.0625	kHz
4	D	内部参考启动时间	$t_{\text{irefst}}$	—	60	100	us
5	—	DCO 输出频率范围 - 为参考提供的未调整值 <sup>1</sup> : $f_{\text{dco\_ut}} = 1024 \times f_{\text{int\_ut}}$	$f_{\text{dco\_ut}}$	25.6	33.48	42.66	MHz
6	P	DCO 输出频率范围 - 已调整	$f_{\text{dco\_t}}$	32	—	40	MHz
7	C	电压和温度固定时经调整的 DCO 输出频率的分辨率 (使用 FTRIM)	$\Delta f_{\text{dco\_res\_t}}$	—	$\pm 0.1$	$\pm 0.2$	% $f_{\text{dco}}$
8	C	电压和温度固定时经调整的 DCO 输出频率的分辨率 (不使用 FTRIM)	$\Delta f_{\text{dco\_res\_t}}$	—	$\pm 0.2$	$\pm 0.4$	% $f_{\text{dco}}$
9	P	已调整 DCO 输出频率的电压和温度总误差	$\Delta f_{\text{dco\_t}}$	—	+ 0.5 - 1.0	$\pm 2$	% $f_{\text{dco}}$
10	C	0 - 70 °C 的温度范围内时已调整 DCO 输出频率的总误差	$\Delta f_{\text{dco\_t}}$	—	$\pm 0.5$	$\pm 1$	% $f_{\text{dco}}$
11	C	FLL 获取时间 <sup>2</sup>	$t_{\text{fill\_acquire}}$	—	—	1	ms
12	D	PLL 获取时间 <sup>3</sup>	$t_{\text{pll\_acquire}}$	—	—	1	ms
13	C	输出时钟的长时间抖动（平均间隔为 2ms） <sup>4</sup>	$C_{\text{jitter}}$	—	0.02	0.2	% $f_{\text{dco}}$
14	D	VCO 操作频率	$f_{\text{vco}}$	7.0	—	55.0	MHz
15	D	PLL 参考频率范围	$f_{\text{pll\_ref}}$	1.0	—	2.0	MHz
16	T	输出时钟的长期准确性（平均为 2 ms）	$f_{\text{pll\_jitter\_2ms}}$	—	0.590 <sup>5</sup>	—	% $f_{\text{pll}}$
17	T	基于 625 ns <sup>6</sup> 测量的 PLL 输出时钟抖动	$f_{\text{pll\_jitter\_625ns}}$	—	0.566 <sup>5</sup>	—	% $f_{\text{pll}}$
18	D	锁定进入频率容限 <sup>7</sup>	$D_{\text{lock}}$	$\pm 1.49$	—	$\pm 2.98$	%
19	D	锁定退出频率容限 <sup>8</sup>	$D_{\text{unl}}$	$\pm 4.47$	—	$\pm 5.97$	%



注释:

1. SS 输出模式 (MODFEN = 1, SSOE = 1).
2. LSBF = 0。当 LSBF = 1 时, 位顺序是 LSB、位 1、...、位 6、MSB。

图 A-7. SPI 主时序 (CPHA = 0)



注释

1. SS 输出模式 (MODFEN = 1, SSOE = 1)
2. LSBF = 0。当 LSBF = 1 时, 位顺序是 LSB、位 1、...、位 6、MSB。B.

图 A-8. SPI 主时序 (CPHA = 1)

### B.5.1 定时器状态和控制寄存器 (TPMxSC)

TPMxSC 包含溢出状态标志和控制位，用来配置中断使能、TPM 配置、时钟源和预分频器除数。这些控制与本定时器模块中的所有通道相关。

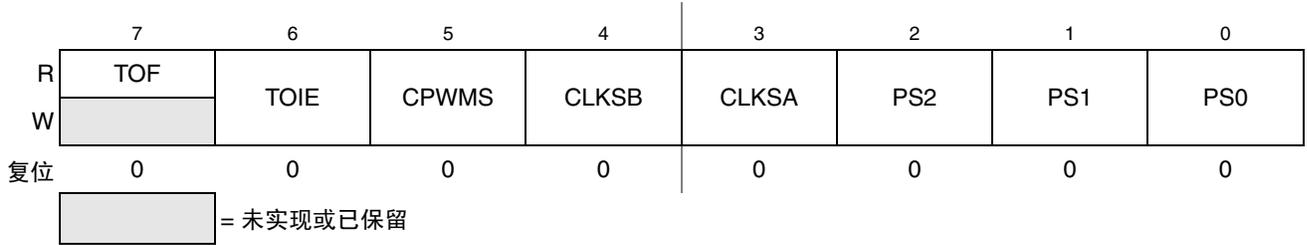


图 B-2. 定时器状态和控制寄存器 (TPMxSC)

表 B-1. TPMxSC 寄存器字段描述

字段	描述
7 TOF	<b>定时器溢出标记</b> — 该标记用于 TPM 计数器变更为 x0000 时，并且已经到达 TPM 计数器模量寄存器中设定的模量值。当 TPM 配置用于 CPWM 时，计数器达到模量寄存器的值后设置 TOF，这时转化为下一个较低的计数值。TOF 设置完毕后，读取 TPM 状态和控制寄存器，在 TOF 中写入 0，可以清除 TOF。如果在清除顺序完成之前，发生另一个 TPM 溢出，该顺序要复位，这样前一个 TOF 的清除顺序完成后，后一个 TOF 仍然能保持为设置状态。复位可清除 TOF。在 TOF 中写入 1 不会产生影响。 0 TPM 计数器没有达到模量值或没有溢出 1 TPM 计数器已经溢出
6 TOIE	<b>定时器溢出中断使能</b> — 这个读 / 写位使能 TPM 溢出中断。如果 TOIE 被设置，那么在 TOF 等于 1 时会生成中断。复位可清除 TOIE。 0 TOF 中断关闭（用于软件轮询） 1 TOF 中断允许
5 CPWMS	<b>中心对齐 PWM 选择</b> — 这个读写位选择 CPWM 的操作模式。复位清除该位，这样 TPM 就在向上计数模式中运行，完成输入捕捉、输出比较和边沿对齐 PWM 功能。设置 CPWMS 可以重新配置 TPM，使它以向上 / 下计数模式操作完成 CPWM 功能。复位清除 CPWMS。 0 所有 TPMx 通道以输入捕捉、输出比较或边沿对齐 PWM 模式操作，由各通道的状态和控制寄存器中 MSnB:MSnA 控制位的选择 1 所有 TPMx 通道以中心对齐 PWM 模式操作
4:3 CLKS[B:A]	<b>时钟源选择</b> — 如表 A-2 所示，2 位字段用来禁止 TPM 系统或在 3 个时钟源中任选一个驱动计数器预分频器。外部源和 XCLK 通过片上同步电路，完成与总线的时钟同步。
2:0 PS[2:0]	<b>预分频器除数选择</b> — 这个 3 位字段从 8 个除数中选择一个，用作 TPM 时钟输入，如表 B-3 所示。完成时钟源同步或选定时钟源后，这个预分频器的位置也就确定了，因此无论选择什么时钟源驱动 TPM 系统都会产生影响。

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。

