



Welcome to [E-XFL.COM](http://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 ~ 125°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/pro/item?MUrl=&amp;PartUrl=mc9s08dz60mlh">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc9s08dz60mlh</a>



表 1-2 为芯片模块的功能版本。

表 1-2. 模块版本

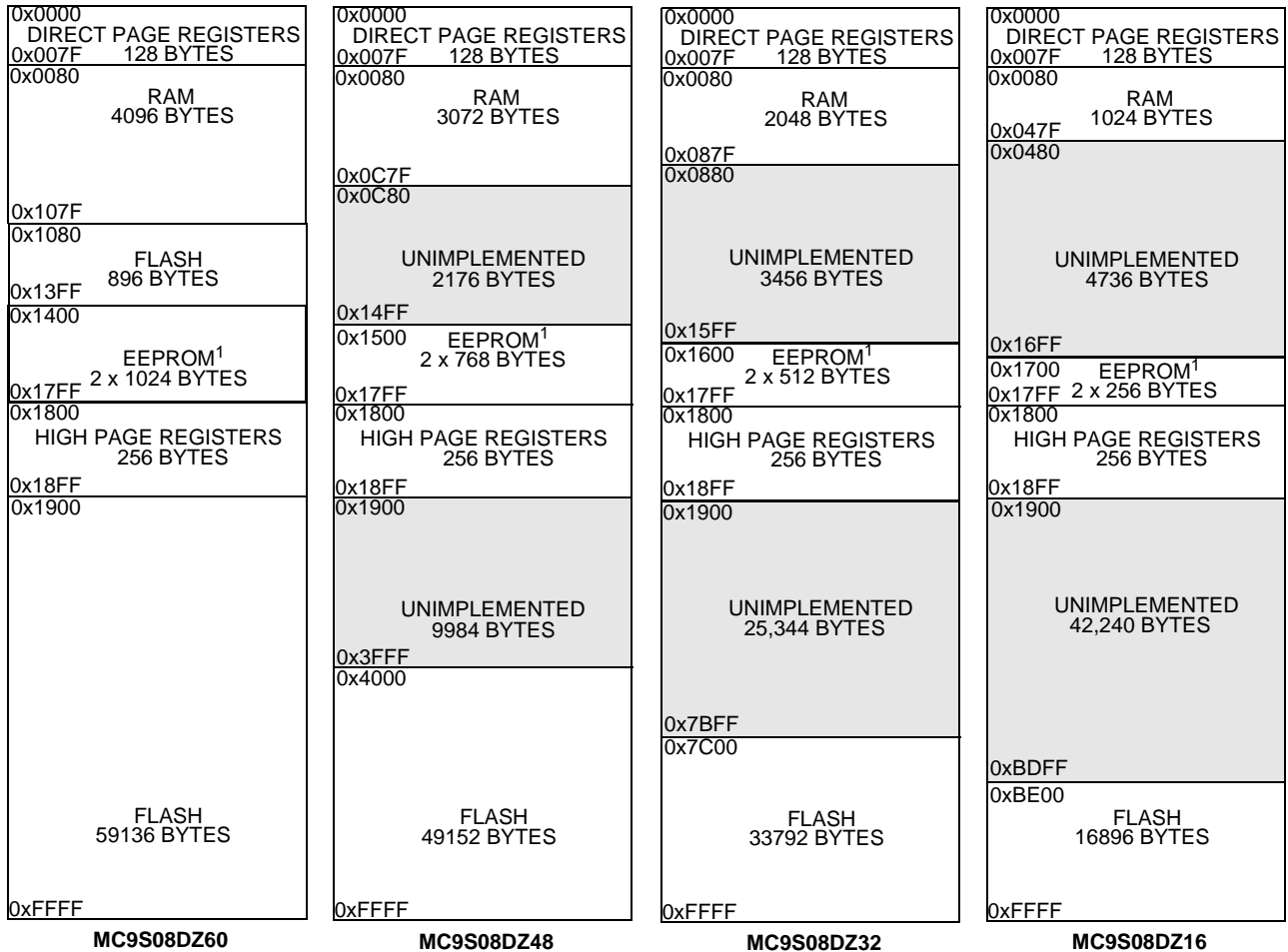
模块	版本
中央处理器 (CPU)	3
多功能时钟生成器 (MCG)	1
模拟比较器 (ACMP)	3
模数转换器 (ADC)	1
IIC 总线 (IIC)	2
飞思卡尔的 CANN (MSCAN)	1
串行外围接口 (SPI)	3
串行通信接口 (SCI)	4
实时计数器 (RTC)	1
定时器脉宽调制器 (TPM)	3 <sup>1</sup>
调试模块 (DBG)	2

# 第 4 章 存储器

## 4.1 MC9S08DZ60 系列产品存储器映射

MC9S08DZ60 系列产品中的片上存储器包括 RAM、EEPROM、用于非易失性数据存储的 Flash 程序存储器、I/O 和控制 / 状态寄存器。这些寄存器可分为以下 3 类：

- 直接页面寄存器 (0x0000 ~ 0x007F)
- 高端页面 (High-page) 寄存器 (0x1800 ~ 0x18FF)
- 非易失性寄存器 (0xFFB0 ~ 0xFFBF)



<sup>1</sup> EEPROM 地址范围显示总 EEPROM 的一半。详尽信息请参见 4.5.10, “EEPROM 映射”。

图 4-1. MC9S08DZ60 存储器图

如果密钥启用（KEYEN）位为 1，那么 8 字节对比密钥可用于暂时脱离内存安全的限制。这种密钥机制只能通过安全内存中运行的用户代码访问。（安全密钥不能通过后台调试命令直接输入。）这个安全密钥可通过将 KEYEN 位设为 0 来完全禁用。如果这个安全密钥被禁用，那么脱离安全限制的唯一方式是整体擦除 Flash（通常通过后台调试接口）并确认 Flash 是空的。为了避免在下次复位后返回到安全模式，应该将安全位（SEC）设置为非安全状态（1: 0）。

## 4.4 RAM

MC9S08DZ60 系列包括静态 RAM。RAM 中 0x0100 以下的位置可以使用更高效的直接寻址模式访问，而这一区域中的任何单一比特可以通过位操作指令（BCLR、BSET、BRCLR 和 BRSET）访问。首选的方式是在这一区域中查找 RAM 最频繁访问的程序变量。

在 MCU 处于低功耗等待、Stop2 或 Stop3 模式时，RAM 会保留数据。加电启动时，RAM 中的内容不会被初始化。如果电源电压没有降低到 RAM 保留 ( $V_{RAM}$ ) 的最低值以下，RAM 数据就不会受到复位的任何影响。

为了实现与 M68HC05 MCU 的兼容性，HCS08 会将栈指针复位为 0x00FF。在 MC9S08DZ60 系列中，最好的方法通常是将栈指针重新初始化到 RAM 顶部，以便使经常被访问的 RAM 变量和位可寻址程序变量处于直接寄存器。复位初始化程序（其中的 RamLast 等于飞思卡尔半导体等同文件中 RAM 的最高地址）中包含以下两个指令序列。

```
LDHX    #RamLast+1    ;point one past RAM
TXS                    ;SP<-(H:X-1)
```

在启用了安全性的情况下，RAM 被认为是一种安全的内存资源，不能通过 BDM 或从非安全内存中执行代码来访问。若欲了解有关安全特性的更详尽描述，请参见 4.5.9，“安全性”。

## 4.5 Flash 和 EEPROM

MC9S08DZ60 系列器件包括 Flash 和 EEPROM 存储器。这些存储器主要用于保存程序和数据。在线编程使正在运行的程序和数据可以在应用产品的最终组装完成后分别上载到 Flash 和 EEPROM 中。我们可以通过单线后台调试接口对阵列进行编程。由于擦除和编程操作不需要特殊的电压，所以也可以通过其他软件控制的通信路径来实现应用编程。有关在线和应用内编程的更详尽描述，请参见“HCS08 系列参考手册，第 1 卷”（飞思卡尔半导体文件编号 HCS08RMv1）。

### 4.5.1 特性

Flash 和 EEPROM 存储器具有以下特性：

- 阵列大小请参见表 1-1
- Flash 分区大小：768 字节
- EEPROM 分区大小：可选 4 字节或 8 字节分区映射操作
- 单一电源程序和擦除
- 用于快速编程和擦除操作的命令接口
- 一般电压和温度下最多 100,000 个编程 / 擦除循环
- 灵活的块保护和向量重定向

## 5.5 中断

在执行中断服务程序 (ISR) 前，当前 CPU 状态和寄存器被保存，而在执行中断服务程序 (ISR) 后，保存的 CPU 状态将被恢复。这样可以从中断前的位置重新开始处理。与软件中断 (SWI) 不同 (SWI 由程序指令触发)，中断是由诸如 IRQ 管脚边沿或定时器溢出样的硬件事件触发。调试模块也可以在特定环境下导致 SWI。

如果中断源内的事件发生，将会设置相关的只读状态标记。但不会响应 CPU，除非是由本地中断使能位置为 1 导致并且 CCR 中的 I 位为 0 来允许的中断。CCR 中的全球中断屏蔽 (I 位) 在复位后首次设置会阻止所有可屏蔽的中断源。在清除 I 位之前，用户程序初始化堆栈指针，执行其他系统设置，以便允许 CPU 响应中断。

当 CPU 接收到符合条件的中断请求时，它会在响应中断前先完成当前指令。中断顺序与 SWI 指令的逐周期顺序相同，这个顺序是：

- 在堆栈上保存 CPU 寄存器；
- 在 CCR 中设置 I 位，禁止中断；
- 为当前悬而未决的最高优先级中断获取中断向量；
- 用程序报文的前 3 个字节填写指令队列，程序报文从在中断向量位置上获取的地址开始；

当 CPU 响应中断时，会自动设置 I 位以避免出现中断 ISR 自身的另外一个中断 (这也叫做中断嵌套)。在正常情况下，当 CCR 从 ISR 入口处堆栈的值进行恢复时，I 位就恢复为 0。在极个别情况中，I 位可以在 ISR 内部清除 (在清除生成中断的状态标志后)，所以无需等待第一个业务程序完成，就可以执行另一个中断。该操作仅供有丰富经验的程序员使用，因为它可能导致难以调试发现的程序错误。

中断服务程序以中断恢复 (RTI) 指令作为结束。RTI 指令从堆栈中读取先前保存的报文，将 CCR、A、X 和 PC 寄存器恢复为中断前的值。

### 注意

为了实现与 M68HC08 器件的兼容，H 寄存器不能自动保存和恢复。建议在中断服务程序 (ISR) 开始时就将 H 推到堆栈上，并在 RTI (用来从 ISR 中恢复) 前立即恢复它。

当 I 位被清除时有多个挂起的中断，处理优先级最高的最先被处理 (参见表 5-1)。

### 5.5.1 中断堆栈帧

图 5-1 为堆栈帧的内容和结构。在中断前，堆栈指针 (SP) 指向堆栈的下一个可用字节。CPU 寄存器的当前值保存在堆栈中，以程序计数器 (PCL) 的低阶字节开始，以 CCR 结束。在一次堆栈操作后，SP 指向堆栈的下一个可用位置，该堆栈是比保存 CCR 的地址小一的地址。被堆栈的 PC 值是主程序的指令地址，如果中断没有发生，那么将在下一次中断中实施主程序。

### 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。

### 7.2.3 堆栈指针 (SP)

这个 16 位地址指针寄存器指向自动后进先出 (LIFO) 堆栈上的下一个可用位置。该堆栈可以位于 64Kb 地址空间的任意位置，64Kb 地址空间具有 RAM，大小可以是可用 RAM 量的任意值。该堆栈用于自动保存子程序调用的返回地址，以及在中断期间供本地变量使用的返回地址和 CPU 寄存器。AIS (立即值加到堆栈指针) 指令向 SP 添加一个 8 位带符号的立即值。这通常供用于堆栈上的本地变量的空间分配或取消分配。

SP 在复位时被强制放在 0x00FF 上，以实现与早期 M68HC05 系列的兼容性。在复位初始化期间，HCS08 程序通常将 SP 中的值更改为片上 RAM 最后位置 (最高地址) 的地址，以释放直接页面 RAM (从片上寄存器末端到 0x00FF)。

为了实现与 M68HC05 系列的兼容，指令中还包括 RSP (复位堆栈指针) 指令，但很少在 HCS08 程序中使用，因为它只影响堆栈指针的低阶部分。

### 7.2.4 程序计数器 (PC)

程序计数器是一个 16 位寄存器，它包含将要获取的下一个指令或操作数的地址。

在正常的程序执行过程中，每次获取指令或操作数时，程序计数器就会自动累加到下一个顺序存储器位置。跳转、分支、中断和返回操作加载地址到程序计数器，而非下一个顺序位置的存储器地址，这被称为 change-of-flow (流程转换)。

复位时，程序计数器被加载位于 0xFFFFE 和 0xFFFF 的复位向量。这里保存的向量是退出复位状态后将要执行的第一个指令的地址。

### 7.2.5 条件码寄存器 (CCR)

8 位条件码寄存器包括中断屏蔽 (I) 和 5 个显示刚执行指令的结果的标记。位 6 和 5 永远设为 1。下面的几个段落描述了一般条件下的条件码位功能。如需了解各个指令如何设置 CCR 位的更多信息，请参见“HCS08 系列参考手册第 1 卷”，飞思卡尔半导体文档订购编号 HCS08RMv1。

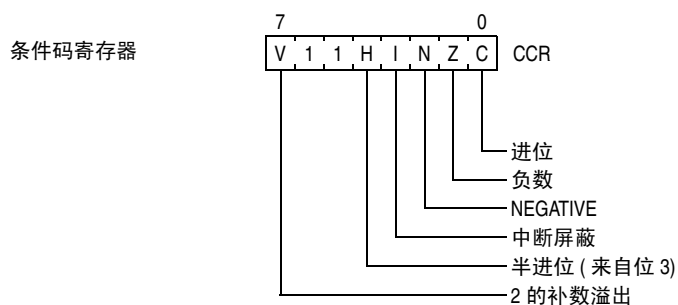


图 7-2. 条件码寄存器



## 7.3 寻址模式

寻址模式定义 CPU 访问操作数和数据的方式。在 HCS08 中，所有存储器、状态和控制寄存器、输入 / 输出 (I/O) 端口共用一个 64Kb 线性地址空间，因此 16 位二进制地址能够识别任意存储器位置。这种安排意味着，访问 RAM 中变量的相同指令还可以用来访问 I/O 和控制寄存器或非易失性程序空间。

有些指令使用一个以上的寻址模式。例如，移动指令用一种寻址模式来指定源操作数，用另外一种寻址模式来指定目的地地址。而诸如 BRCLR、BRSET、CBEQ 和 DBNZ 这样的指令则使用一种寻址模式来指定测试用操作数的位置，当测试条件成立时，又使用相关寻址模式来指定分支目的地地址。对于 BRCLR、BRSET、CBEQ 和 DBNZ 而言，指令集表里列出的寻址模式是访问将要测试的操作数所需的寻址模式，而相关寻址模式则意味着分支地址寻址模式。

### 7.3.1 固有寻址模式 (INH)

在这种寻址模式中，完成指令（如果有的话）所需的操作数位于 CPU 寄存器上，因此 CPU 不需要访问存储器以获得任何操作数。

### 7.3.2 关联寻址模式 (REL)

关联寻址模式用来指定分支指令的目的地址。一个带符号的 8 位偏移值位于紧接操作码的存储器位置。在执行期间，如果分支条件成立，带符号的偏移被符号扩展为 16 位值，并被添加到程序计数器的当前内容中，造成在分支目的地地址上继续执行程序。

### 7.3.3 立即寻址模式 (IMM)

在立即寻址模式中，完成指令所需的操作数包含在紧跟存储器的指令操作码的目标码中。如果是 16 位直接操作数，高阶字节位于操作码后的下一个存储器位置，低阶字节位于高阶存储器后的下一个存储器位置。

### 7.3.4 直接寻址模式 (DIR)

在直接寻址模式中，指令包括直接页面 (0x0000 - 0x00FF) 地址的低阶 8 位。在执行过程中，16 位地址的高 8 位部分隐含为 0x00，低 8 位地址从指令中直接获得，以此构成 16 位地址指向指令所需的操作数。这比为操作数指定整个 16 位地址更快、存储器也更有效。

### 7.3.5 扩展寻址模式 (EXT)

在扩展寻址模式中，操作数的整个 16 位地址都位于紧接操作码后的程序存储器的两个字节上（高阶字节优先）。

### 7.3.6 索引寻址模式

索引寻址模式带有 7 个变种，5 个使用 16 位 H:X 索引寄存器对和两个使用堆栈指针作为基本参考。

表 7-2. 指令集小结 (第 8 页, 共 9 页)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V I 1 H	I N Z C
SUB #opr8i SUB opr8a SUB opr16a SUB oprx16,X SUB oprx8,X SUB ,X SUB oprx16,SP SUB oprx8,SP	减 A'' (A) - (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 ii B0 dd C0 hh ll D0 ee ff E0 ff F0 9E D0 ee ff 9E E0 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↑ 1 1 -	- ↓ ↓ ↓
SWI	软件中断 PC'' (PC) + \$0001 推 (PCL); SP'' (SP) - \$0001 推 (PCH); SP'' (SP) - \$0001 推 (X); SP'' (SP) - \$0001 推 (A); SP'' (SP) - \$0001 推 (CCR); SP'' (SP) - \$0001 I'' 1; PCH'' 中断向量高字节 PCL'' 中断向量低字节	INH	83	11	sssssvvfppp	- 1 1 -	1 - - -
TAP	将累加器转移到 CCR CCR'' (A)	INH	84	1	p	↑ 1 1 ↑	↓ ↓ ↓ ↓
TAX	将累加器转移到 X (索引寄存器低) X'' (A)	INH	97	1	p	- 1 1 -	- - - -
TPA	将 CCR 转移到累加器 A'' (CCR)	INH	85	1	p	- 1 1 -	- - - -
TST opr8a TSTA TSTX TST oprx8,X TST ,X TST oprx8,SP	负数或 0(M) 测试 (M) - \$00 (A) - \$00 (X) - \$00 (M) - \$00 (M) - \$00 (M) - \$00	DIR INH INH IX1 IX SP1	3D dd 4D 5D 6D ff 7D 9E 6D ff	4 1 1 4 3 5	rfpp p p rfpp rfp prfpp	0 1 1 -	- ↓ ↓ -
TSX	将 SP 转移到索引寄存器。 H:X'' (SP) + \$0001	INH	95	2	fp	- 1 1 -	- - - -
TXA	将 X(索引寄存器低) 转移到累加器上 A'' (X)	INH	9F	1	p	- 1 1 -	- - - -

## 第 8 章

# 多功能时钟发生器 (S08MCGV1)

### 8.1 介绍

多功能时钟发生器 (MCG) 模块为 MCU 提供了几个时钟源选项。MCG 模块中包含 1 个锁频环 (FLL) 和 1 个锁相环 (PLL)，可以由内部或外部参考时钟控制。模块可以选择 FLL 或 PLL 时钟作为 MCU 系统时钟，也可以选择内部或外部参考时钟作为 MCU 系统时钟。无论选择哪个时钟源，它都要通过降阶总线分频器，该分频器允许生成更低的输出时钟频率。MCG 还控制一个外部振荡器 (XOSC)，以便把晶体或共鸣器用作外部参考时钟。

MC9S08DZ60 系列的所有器件都含有 MCG 模块。

#### 注意

如需了解整个芯片的分配时钟源的更多信息，请参见 1.3，“系统时钟分配”。

## 10.2.6 特性

ADC 模块的特性包括：

- 具有 12 位分辨率的线性逐次逼近算法；
- 高达 28 个模拟输入；
- 12、10 或或 8 位右对齐输出格式；
- 单次转换或连续转换（单转换后自动返回空闲状态）；
- 采样时间和转换速度 / 功率可配置；
- 转换完成标志和中断；
- 最多可选择 4 个输入时钟源；
- 在等待或 STOP3 模式下实现了低噪音运行；
- 异步时钟源实现了低噪音运行；
- 可选的异步硬件转换触发；
- 与小于、大于或等于可编程值自动比较的中断；

## 10.2.7 结构图

图 10-2 是 ADC 模块的结构图

### 10.4.3 数据结果高地址寄存器 (ADCRH)

	7	6	5	4	3	2	1	0
R	0	0	0	0	ADR11	ADR10	ADR9	ADR8
W								
复位:	0	0	0	0	0	0	0	0

图 10-6. 数据结果高地址寄存器 (ADCRH)

在 10 位模式中，ADCRH 包含 10 位转换结果中的高 2 位。在配置 10 位模式时，ADR11 - ADR10 等于 0。当为 8 位模式进行配置时，ADR11 - ADR8 等于 0。

在 12 位和 10 位模式中，每次完成转换时 ADCRH 都被更新，除非使能了自动比较但不满足比较条件。在 12 位和 10 位模式中，读取 ADCRH 能够防止在读取 ADCRL 之前 ADC 将后续转换结果传输到结果寄存器。如果 ADCRL 是在下一次转换完成后才被读取，那么中间转换结果会被丢失。在 8 位模式中，没有与 ADCRL 的互锁。

当 MODE 位被更改时，ADCRH 中的任何数据都将无效。数据结果低地址寄存器 (ADCRL)

ADCRL 包含 12 位或 10 位转换结果中的低 8 位。每次转换完成时，8 位转换寄存器的 8 个位都被更新，除非使能了自动比较但不满足比较条件。在 12 位和 10 位模式中，读取 ADCRH 能够防止在读取 ADCRL 之前 ADC 将后续转换结果传输到结果寄存器。如果 ADCRL 是在下一次转换完成后才被读取，那么中间转换结果会被丢失。在 8 位模式中，没有与 ADCRH 的互锁。当 MODE 位被更改时，ADCRL 中的任何数据都将无效。

	7	6	5	4	3	2	1	0
R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
W								
复位:	0	0	0	0	0	0	0	0

图 10-7. 数据结果低态寄存器 (ADCRL)

### 10.4.4 比较值高地址寄存器 (ADCCVH)

在 12 位模式中，ADCCVH 寄存器包含 12 位比较值的高 4 位。当比较功能使能时，这些位与 12 位模式转换结果的高 4 位进行比较。

	7	6	5	4	3	2	1	0
R	0	0	0	0	ADCV11	ADCV10	ADCV9	ADCV8
W								
复位:	0	0	0	0	0	0	0	0

图 10-8. 比较值高态寄存器 (ADCCVH)

在 10 位模式中，ADCCVH 寄存器包含 10 位比较值 (ADCV9 - ADCV8) 的高 2 位。当比较功能使能时，这些位就与 10 位模式转换结果的高 2 位进行比较。

当 CAN 总线赢得仲裁时，如果有一个以上的缓冲器等待发送，MSCAN 则使用三个缓冲器的本地优先级设置来决定优先顺序。因此，每个发送缓冲器都有 8 位本地优先级字段（PRIO）。当报文建立时，应用软件就编辑该字段。本地优先级反应了在从该节点发送的有关报文之间的优先顺序。具有最低二进制代码的 PRIO 字段占最高优先级。每当 MSCAN 为 CAN 总线进行仲裁时，就会引发内部调度程序。当出现发送错误时也会如此。

当应用软件安排了高优先级报文时，可能有必要中止三个发送缓冲器的某一个低优先级报文。由于正发送的报文不能中止，因此用户必须通过设置相应中止请求位（ABTRQ）请求中止（参见 12.3.8，“MSCAN Transmitter 发送器报文中止请求寄存器（CANTARQ）”）。可能的话，MSCAN 通过以下方式同意该请求：

1. 在 CANTAACK 寄存器中设置相应的中止确认标志（ABTAK）。
2. 设置相关的 TXE 标志来释放缓冲器。
3. 生成发送中断。发送中断处理程序软件能够根据 ABTAK 标志的设置确定是报文中止（ABTAK = 1）还是已发送（ABTAK = 0）。

### 12.5.2.3 接收结构

收到的报文保存在 5 级输入 FIFO 中。5 个报文缓冲器被交替映射到单个存储器区域（参见图 12-38）。后台接收缓冲器（RxBG）只与 MSCAN 相关，但前景接收缓冲器可以通过 CPU 寻址（参见图 12-38）。这种机制简化了处理程序软件，因为接收流程只需访问一个地址。

如使能的话，所有接收缓冲器都有 15 字节大小空间来保存 CAN 控制位、标识符（标准或扩展）、数据内容（参见 12.4，“报文存储模式”）。

接收器已满标志（RXF）（参见 12.3.4.1，“MSCAN 接收器标志寄存器（CANRFLG）”）显示前景接收缓冲器的状态。当缓冲器包含带有匹配标识符的正确接收报文时，设置该标志。

接收时，检查每条报文，看看它是否通过滤波器（参见 12.5.3，“标识符接收滤波器”），同时被写入有效 RxBG。成功接收到有效报文后，MSCAN 将 RxBG 的内容转移到接收器 FIFO2，设置 RXF 标志并向 CPU3 生成一个接收中断（参见 12.5.7.3，“接收中断”）。用户的接收处理程序必须从 RxFG 读取收到的报文，然后复位 RXF 标志，确认中断、释放前景缓冲器。在某些情况下，紧跟 CAN 帧的 IFS 字段后的新报文，将被接收到下一个可用 RxBG 中。如果 MSCAN 在其 RxBG 中接收到无效报文（错误标识符、发送错误等），缓冲器的实际内容将被下一条报文覆盖。缓冲器随后不会转移到 FIFO。

当 MSCAN 模块正在发送报文时，MSCAN 把其自己发送的报文接收到后台接收缓冲器 RxBG，但不会将它转移到接收器 FIFO，生成接收中断或在 CAN 总线上响应其自己的报文。这一规则的例外是在环回模式（参见 12.3.2，“控制寄存器 1（CANCTL1）”）中，这时 MSCAN 会完全按照同所有其他报文一样的方式处理其自己的报文。当仲裁丢失时，MSCAN 接收其自己发送的报文。如果仲裁丢失，MSCAN 必须做好成为接收器的准备。

当 FIFO 中的所有接收报文缓冲器充满了带有已接收标识符的正确接收报文，且从 CAN 总线中正确接收到另外一条带有已接收标识符的报文时，就会出现溢出。后面这一条报文被丢弃，并生成带有溢出标志的错误中断（参见 12.5.7.5，“错误中断”）。当接收器 FIFO 已满时，MSCAN 仍能发送报文，但所有进入报文会被丢弃。一旦 FIFO 中的接收缓冲器重新可用，就能接收新的有效报文。

只有当出现以下情形时，MSCAN 才能够退出睡眠模式（唤醒）：

- 出现 CAN 总线有效和  $WUPE = 1$   
或
- CPU 清除  $SLPRQ$  位

### 注意

在使能睡眠模式（ $SLPRQ = 1$ ， $SLPAK = 1$ ）前，CPU 不能清除  $SLPRQ$  位。

唤醒之后，MSCAN 等待 11 个连续隐性位与 CAN 总线同步。因此，如果 MSCAN 被 CAN 帧唤醒，就不会收到该帧。

如果在进入睡眠模式前已经收到报文，接收报文缓冲器（ $RxFG$  和  $RxBG$ ）就包含该报文。所有挂起操作在唤醒后执行，复制  $RxBG$  至  $RxFG$ ，报文中止和报文发送。如果在退出睡眠模式后 MSCAN 仍处于总线脱离状态，它将继续计数 128 次 11 个连续隐性位的出现。

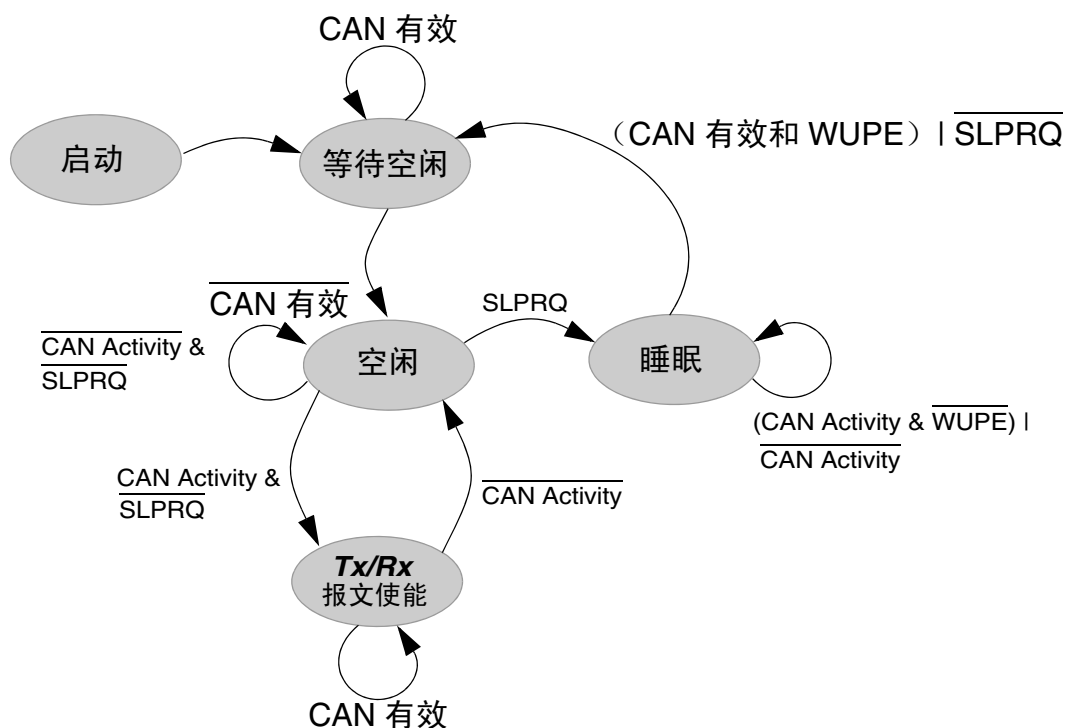


图 12-45. 进入 / 退出睡眠模式的简单状态 转换

```
/******  
Function Name : RTC_ISR  
Notes : Interrupt service routine for RTC module.  
*****/  
#pragma TRAP_PROC  
void RTC_ISR(void)  
{  
    /* Clear the interrupt flag */  
    RTCSC.byte = RTCSC.byte | 0x80;  
    /* RTC interrupts every 1 Second */  
    Seconds++;  
    /* 60 seconds in a minute */  
    if (Seconds > 59){  
        Minutes++;  
        Seconds = 0;  
    }  
    /* 60 minutes in an hour */  
    if (Minutes > 59){  
        Hours++;  
        Minutes = 0;  
    }  
    /* 24 hours in a day */  
    if (Hours > 23){  
        Days ++;  
        Hours = 0;  
    }  
}
```



TPM 中断标志通过两步清除：一是标志位被设置（为 1）时读取，二是向其中写入 0。如果这两步之间检测到新事件，序列被复位，并且在完成第二步后中断标志仍被设置以避免丢失新事件。

### 16.6.2.1 定时器溢出中断 (TOF) 介绍

根据 TPM 系统的运行模式（通用定时功能和中央对齐 PWM 操作），TOF 中断操作的意义和细节会有细微变化。标志通过上述两步序列清除。

#### 16.6.2.1.1 常见情况

当定时器计数器从 0xFFFF 变为 0x0000 时，TOF 通常被设置。当 TPM 没有被配置为中央对齐 PWM 时 (CPWMS=0)，TOF 在定时器计数器从终端计数（模数寄存器中的值）变为 0x0000 时被设置。这种情况与计数器溢出的正常意义相对应。

#### 16.6.2.1.2 中央对齐 PWM 情况

当 CPWMS=1 时，TOF 在定时器计数器方向在终端计数（模数寄存器中的值）结束时从向上计数变为向下计数时被设置。在这种情况下，TOF 与 PWM 周期结束相对应。

### 16.6.2.2 通道事件中断描述

通道中断的意义取决于通道的当前模式（输入捕捉、输出比较、边缘对齐 PWM 或中央对齐 PWM）。

#### 16.6.2.2.1 输入捕捉事件

当通道被配置为输入捕捉通道时，ELSnB:ELSnA 控制位选择无边沿（关）、上升边沿、下降边沿或任何边沿作为触发输入捕捉事件的边沿。检测到选定边沿时，中断标志被设置。标志通过 16.6.2，“中断操作描述”中所述的两步序列清除。

#### 16.6.2.2.2 输出比较事件

如果通道被配置为输出比较通道，中断标志在每次主定时器计数器与通道值寄存器中的 16 位值匹配时被设置。标志通过 16.6.2，“中断操作描述”中所述的两步序列清除。

#### 16.6.2.2.3 PWM 占空比结束事件

对于配置用于 PWM 操作的通道，有两种可能性。当通道配置用于边缘对齐 PWM 时，通道标志在定时器计数器与标志占空比周期结束的通道值寄存器匹配时被设置。当通道配置为中央对齐 PWM 时，定时器计数在每个 PWM 周期内两次与通道值寄存器相匹配。在这种 CPWM 情况中，通道标志在占空比周期时间开始和结束时（定时器计数器与通道值寄存器匹配时）被设置。标志通过 16.6.2，“中断操作描述”中所述的两步序列清除。

SYNC 命令与其它 BDC 命令不同，因为主机不必要知道 BDC 通信的正确通信速率，直到它分析完 SYNC 命令的响应后。

要发出 SYNC 命令，主机：

- 保持 BKGD 管脚为低电平至少 128 周期，而且是以最慢的 BDC 时钟来计 (最慢的时钟一般是参考振荡器 /64 或自时钟速率 /64。)
- 驱动 BKGD 达到高电平，实现瞬态加速，快速上升时间 (这个加速脉冲一般是系统中最快的时钟的一个周期)
- 去除 BKGD 管脚的所有驱动，这样它可回复到高阻抗。
- 监视 BKGD 管脚得到同步响应脉冲

当检测到主机的 SYNC 请求 (比在正常 BDC 通信过程中发生的慢时钟要长)，则目标：

- 等待 BKGD 返回到逻辑高电平
- 延迟 16 个周期，允许主机停止驱动高电平加速脉冲
- 驱动 BKGD 低态 128 BDC 时钟周期
- 驱动一个周期的高电平加速脉冲，在 BKGD 上实现快速上升时间
- 去除 BKGD 管脚的所有驱动，这样它可回复到高阻抗。

主机测量这个 128 周期的响应脉冲的低电平时间，判断速率，进行后续的 BDC 通信。主机一般可以确定正确的通信速率，与实际目标速率的误差只有百分之几，通信协议能够接受百分之几的速率误差。

## 17.2.4 BDC 硬件断点

BDC 包括一个相对简单的硬件断点，将 CPU 地址总线与 BDCBKPT 寄存器中的 16- 位匹配值进行比较。这个断点可以生成强制断点或标记断点。强制断点使 CPU 在存取断点地址后的第一个指令边界进入激活背景调试模式。标记的断点使指令操作码在断点地址被标记，这样当 CPU 到达指令队列的终点时，将进入激活后台模式，而不是执行该指令。这意味着标记的断点可能放置在指令操作代码的地址上，而强制断点可以设置在任何地址。

BDC 状态和控制寄存器 (BDCSCR) 中的断点使能 (BKPTEN) 控制位用来激活断点逻辑 (BKPTEN = 1)。当 BKPTEN = 0 (复位后它的默认值)，断点逻辑禁止，无论其它 BDC 断点中的值是多少，也不管控制位如何，均不请求断点。BDCSCR 中的强制 / 标记选择 (FTS) 控制位用来选择强制 (FTS = 1) 或标记 (FTS = 0) 类型断点。

片上调试模块 (DBG) 包括两个额外的硬件断点的电路，这两个硬件断点比 BDC 模块中的简单断点更灵活。

## A.10 外部振荡器 (XOSC) 特性

表 A-11. 振荡器电气规范 (温度范围 = -40 - 125 °C)

编号	C	参数	符号	最小值	典型值 <sup>1</sup>	最大值	单位	
16	C	振荡器晶体或共鸣器 (EREFS = 1, ERCLKEN = 1)						
		低量程 (RANGE = 0)	$f_{lo}$	32	—	38.4	kHz	
		高量程 (RANGE = 1) FEE 或 FBE 模式 <sup>2</sup>	$f_{hi-fll}$	1	—	5	MHz	
		高量程 (RANGE = 1) PEE 或 PBE 模式 <sup>3</sup>	$f_{hi-pll}$	1	—	16	MHz	
		高量程 (RANGE = 1, HGO = 1) BLPE 模式	$f_{hi-hgo}$	1	—	16	MHz	
		高量程 (RANGE = 1, HGO = 0) BLPE 模式	$f_{hi-lp}$	1	—	8	MHz	
17	—	载荷电容器	$C_1$ $C_2$	参见晶体或共鸣器制造商的推荐。				
18	—	反馈电阻器						
		低量程 (32 kHz to 100 kHz)	$R_F$	—	10	—	$M\Omega$	
高量程 (1 MHz to 16 MHz)	—	1		—	$M\Omega$			
19	—	串行电阻器						
		低量程, 低增益 (RANGE = 0, HGO = 0)	$R_S$	—	0	—	$k\Omega$	
		低量程, 高增益 (RANGE = 0, HGO = 1)		—	100	—		
		高量程, 低增益 (RANGE = 1, HGO = 0)		—	0	—		
		高量程, 高增益 (RANGE = 1, HGO = 1) $\geq 8$ MHz		—	0	0		
4 MHz	—	0		10				
1 MHz	—	0	20					
20	T	晶体启动时间 <sup>4</sup>						
		低量程, 低增益 (RANGE = 0, HGO = 0)	$t_{CSTL-LP}$	—	200	—	ms	
		低量程, 高增益 (RANGE = 0, HGO = 1)	$t_{CSTL-HGO}$	—	400	—		
		高量程, 低增益 (RANGE = 1, HGO = 0) <sup>5</sup>	$t_{CSTH-LP}$	—	5	—		
高量程, 高增益 (RANGE = 1, HGO = 1) <sup>4</sup>	$t_{CSTH-HGO}$	—	15	—				
21	T	方波输入时钟频率 (EREFS = 0, ERCLKEN = 1)						
		FEE 或 FBE 模式 <sup>2</sup>	$f_{extal}$	0.03125	—	5	MHz	
		PEE 或 PBE 模式 <sup>3</sup>		1	—	16		
BLPE 模式	0	—		40				

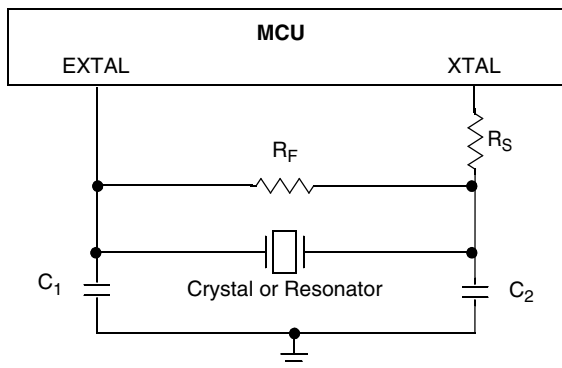
<sup>1</sup> 典型数据是电压为 3.0 V、温度为 25°C 时的数据或推荐值。

<sup>2</sup> 当为 FEE 或 FBE 模式配置 MCG 时, 必须能够用 RDIV 将输入时钟源分割在 31.25 kHz 至 39.0625 kHz 的范围内。

<sup>3</sup> 当为 PEE 或 PBE 模式配置 MCG 时, 能够能够用 RDIV 将输入时钟源分割在 1 kHz 至 2 kHz 的范围内。

<sup>4</sup> 该参数是描述性数据, 未在每个器件上进行测试。要达到规范, 必须遵守正确的 PC 主板布局流程。

<sup>5</sup> 4 MHz 晶体。



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

