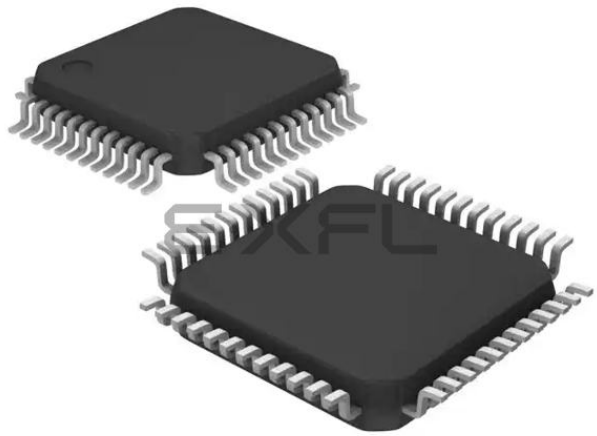

Welcome to E-XFL.COM

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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dz32mlf

6.5.1.3 A 端口上拉使能寄存器 (PTAPE)

	7	6	5	4	3	2	1	0
R W	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
复位 :	0	0	0	0	0	0	0	0

图 6-5. A 端口寄存器的内部上拉使能 (PTAPE)

表 6-3. PTAPE 寄存器字段描述

字段	描述
7:0 PTAPE[7:0]	**A 端口内部上拉使能位** — 对于 PTA 管脚，这些控制位决定着为相关 PTA 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 A 端口管脚，这些位不会产生影响，同时内部上拉器件被禁止。 0 A 端口位 - 内部上拉 / 下拉器件被禁止。 1 A 端口位 - 内部上拉 / 下拉器件使能。

注意

只有当使用管脚中断功能且配置了相应的边沿选择和管脚选择功能时，才能使用下拉器件。

6.5.1.4 A 端口斜率使能寄存器 (PTASE)

	7	6	5	4	3	2	1	0
R W	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
复位 :	0	0	0	0	0	0	0	0

图 6-6. A 端口寄存器的斜率使能 (PTASE)

表 6-4. PTASE 寄存器字段描述

字段	描述
7:0 PTASE[7:0]	**A 端口位输出斜率使能** — 这些控制位决定着是否为相关 PTA 管脚使能输出斜率控制。对于配置为输入的 A 端口管脚，这些位不会产生任何影响。 0 A 端口位 - 输出斜率控制被禁止。 1 A 端口位 - 输出斜率控制使能。

注意：工程样品和最终成品的斜率复位默认值可能不同。一定要将斜率控制初始化为所需的值，以确保正确的操作。

6.5.3.3 C 端口上拉使能寄存器 (PTCPE)

	7	6	5	4	3	2	1	0
R W	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
复位 :	0	0	0	0	0	0	0	0

图 6-21. C 端口寄存器内部上拉使能 (PTCPE)

表 6-19. PTCPE 寄存器字段描述

字段	描述
7:0 PTCPE[7:0]	**C 端口内部上拉使能位** — 这些控制位决定着是否为相关 PTC 管脚使能内部上拉器件。对于配置为输出的 C 端口管脚，这些位不会产生影响，同时内部拉器件被禁止。 0 C 端口位 - 内部上拉器件被禁止。 1 C 端口位 - 内部上拉器件使能。

注意

只有当使用管脚中断功能且配置了相应的边沿选择和管脚选择功能时，才能使用下拉器件。

6.5.3.4 C 端口斜率使能寄存器 (PTCSE)

	7	6	5	4	3	2	1	0
R W	PTCSE7	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
复位 :	0	0	0	0	0	0	0	0

图 6-22. C 端口寄存器斜率使能 (PTCSE)

表 6-20. PTCSE 寄存器字段描述

字段	描述
7:0 PTCSE[7:0]	**C 端口位输出斜率使能** — 这些控制位决定着是否为相关 PTC 管脚使能输出斜率控制。对于配置为输入的 C 端口管脚，这些位不会产生任何影响。 0 C 端口位 - 输出斜率控制禁止。 1 C 端口位 - 输出斜率控制使能。

注意：工程样品和最终成品的斜率复位默认值可能不同。一定要将斜率控制初始化为所需的值，以确保正确的操作。

6.5.6.1 F 端口数据寄存器 (PTFD)

	7	6	5	4	3	2	1	0
R W	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
复位 :	0	0	0	0	0	0	0	0

图 6-37. F 端口数据寄存器 (PTFD)

表 6-35. PTFD 寄存器字段描述

字段	描述
7:0 PTFD[7:0]	**F 端口数据寄存器位** — 对于配置为输入的 F 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 F 端口管脚，读数返回写入寄存器的最后一个值。 写入值被锁定在本寄存器的所有位中。对于配置为输出的 F 端口管脚，逻辑电平被输出到驱出相应的 MCU 管脚。 复位强制 PTFD 都为 0，但是这些 0 未被输出到驱出相应的管脚，因为复位还会将所有端口管脚配置为上拉 / 下拉被禁止的高抗阻输入。

6.5.6.2 F 端口数据方向寄存器 (PTFDD)

	7	6	5	4	3	2	1	0
R W	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
复位 :	0	0	0	0	0	0	0	0

图 6-38. F 端口数据方向寄存器 (PTFDD)

表 6-36. PTFDD 寄存器字段描述

字段	描述
7:0 PTFDD[7:0]	**F 端口位的数据方向** — 这些读 / 写位控制着 F 端口管脚的方向以及为 PTFD 读数读取的内容。 0 输入 （输出驱动被禁止），读数返回管脚值。 1 B 端口位 - 输出驱动使能， PTFD 读数返回 PTFDn 内容。

7.4 特殊运算

CPU 执行一些特殊运算，这些运算和指令类似，但不像其他 CPU 指令那样有操作码。此外，有些指令，如 STOP 和 WAIT，还会直接影响其他 MCU 电路。本节提供了有关这些运算报文。

7.4.1 复位顺序

上电复位 （POR）事件、内部条件 （如 COP 看门狗）或外部低效复位管脚有效状态等都可以导致复位。发生复位事件时， CPU 立即停止正在处理的任何操作 （ MCU 在响应复位事件前，无需等待指令边界）。如需了解 MCU 如何识别复位和决定源的详细信息，请参见 “复位，中断和系统配置”章。

当确定复位是否来自内部源的顺序已经确定且复位管脚的电平不再处于复位有效状态时，复位事件就视为已经结束。复位事件结束后， CPU 执行一个 6 周期顺序，从 0xFFFE 和 0xFFFF 中获取复位向量，并填写指令队列，为执行第一个程序指令做准备。

7.4.2 中断时序

发生中断请求时， CPU 在响应中断前会完成当前指令。此时，程序计数器指向下一个指令的开始位置，这个位置也是 CPU 完成中断操作后的返回位置。 CPU 通过执行与软件中断 （SWI）指令一样的运算顺序响应中断，但不同的是，用于向量获取的地址由中断时序开始时处于等待状态优先级最高的中断决定。

CPU 中断的时序为：

1. 把 PCL、 PCH、 X、 A 和 CCR 的内容顺序保存到堆栈上；
2. 在 CCR 中设置 I 位；
3. 获取中断向量高阶部分；
4. 获取中断向量低阶部分；
5. 延迟一个空闲总线周期；
6. 获取从中断向量显示的地址开始的 3 个字节程序报文来填写指令队列，为执行中断服务程序的第一个指令做好准备。

当 CCR 内容被推送到堆栈后，在 CCR 中 I 位被置位，防止中断服务程序中出现其他中断。尽管可以用中断服务程序的指令来清除 I 位，但这样会发生中断嵌套 （不推荐使用该方法，因为它会让程序难以调试和维护）。

为了实现与早期 M68HC05 的兼容， H:X 索引寄存器对 （H）的高阶部分未作为中断顺序的一部分保存在堆栈上。用户必须在服务程序开始时使用 PSHH 指令来保存 H，然后在结束中断服务程序的 RTI 前使用 PULH 指令。如果您确定中断服务程序不使用任何指令或可能修改 H 值的自动增量寻址模式，就没有必要保存 H。

软件中断 （SWI）指令类似硬件中断，只是它不是由 CCR 中的全局 I 位进行屏蔽，它与程序中的指令操作码有关，因此它不同步于程序执行。

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。

8.6.2.2 示例 2: 从 PEE 切换到 BLPI 模式：外部晶体 = 4 MHz、总线频率 =16 kHz

本例中， MCG 将通过适当的运行模式，从晶体频率为 4MHz、总线频率为 8MHz 的 PEE 模式（参见前一示例）切换到总线频率为 16kHz 的 BLPI 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， PEE 必须转换到 PBE 模式：
   a) MCGC1 = 0x90 (%10010000)
   — CLKS （位 7 和 6）设置为 %10，以便把系统时钟源切换到外部参考时钟。
   b) 循环检测，直到 MCGSC 中 CLKST （位 3 和 2）是 %10，表明已经选择外部参考时钟为 MCGOUT 馈电。
2. 然后， PBE 必须要么直接转换到 FBE 模式，要么先转换到 BLPE 模式，然后再转换到 FBE 模式：
   a) BLPE：如果需要从 BLPE 模式转换，首先把 MCGC2 中的 LP （位 3）设置为 1。
   b) BLPE/FBE: MCGC1 = 0xB8 (%10111000)
   — RDIV （位 5-3）设置为 %111 或除以 128，因为 4 MHz / 128 = 31.25 kHz，这在 FLL 要求的 31.25 kHz -- 39.0625 kHz 频率范围内。在 BLPE 模式中， RDIV 的配置不重要，因为 FLL 和 PLL 都被禁止。更改它们只会建立供 FLL 在 FBE 模式中使用的分频器。
   c) BLPE/FBE: MCGC3 = 0x04 (%00000100)
   — PLLS （位 6）清除至 0，选择 FLL。在 BLPE 模式中，更改该位只会让 MCG 准备在 FBE 模式中的 FLL 使用。如果 PLLS = 0， VDIV 值不重要。
   d) BLPE：如果通过 BLPE 模式转换，将 MCGC2LP （位 3）中的 LP 清除至 0，切换到 FBE 模式。
   e) FBE：循环检测，直到 MCGSC 中的 PLLST （位 5）已经清除，表明 PLLS 时钟的当前源是 FLL。
   f) FBE：循环检测，直到在 MCGSC 中的 LOCK （位 6）已经置位，表明 FLL 已经获得锁定。尽管在 FBE 模式中 FLL 被旁通，但它仍使能且在运行。
3. 接下来， FBE 模式转换到 FBI 模式：
   a) MCGC1 = 0x44 (%01000100)
   — MCGSC1 中的 CLKS （位 7 和 6）设置为 %01，以便将系统时钟切换到内部参考时钟。
   — IREFS （位 2）设置为 1，选择内部参考时钟为参考时钟源。
   — RDIV （位 5-3）设置为 %000 或除以 1，因为调整后的内部参考应在 FLL 要求的 31.25 kHz--39.0625 kHz 频率范围内。
   b) 循环检测，直到 MCGSC 中的 IREFST （位 4）是 1，表明已经选择内部参考时钟为参考时钟源。
   c) 循环检测，直到 MCGSC 中的 CLKST （位 3 和 2）是 %01，表明已经选择内部参考时钟为 MCGOUT 馈电。

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 MHz / 128 = 31.25 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。

表 9-3. ACMPxSC 字段描述

字段	描述
2 ACOPE	模拟比较器输出管脚使能。将使能在外部管脚 ACMPxO 上的比较器输出。 0 模拟比较器输出在 ACMPxO 上不可用 1 模拟比较器输出被驱动到 ACMPxO 上
1:0 ACMOD	模拟比较器模式。ACMOD 择置位 ACF 的比较器事件类型。 00 Encoding 0 — 比较器输出下降沿 01 Encoding 1 — 比较器输出上升沿 10 Encoding 2 — 比较器输出下降沿 11 Encoding 3 — 比较器输出上升或下降沿 9.4 功能描述

9.4 功能描述

模拟比较器可以比较 ACMPx+ 和 ACMPx- 管脚上的模拟输入电压，或者它也可以将 ACMPx- 管脚上的模拟输入电压和内部带死区参考电压比较。ACBGS 选择带死区参考电压或 ACMPx+ 管脚为模拟比较器同相输入的输入。当同相输入大于反相输入时，比较器输出为高电平；当同相输入小于反相输入时，比较器输出为低电平。ACMOD 选择置位 ACF 的条件。 ACF 可以在比较器输出的上升沿、比较器输出的下降沿、或上升及下降沿 （切换）时置位。比较器输出可以通过 ACO 直接读取。使用 ACOPE，比较器输出可以直接驱动到 ACMPxO 管脚上。。

ADCRH/L = 0xxx

保留转换结果。读取低字节 （ADCRL）前的高字节 （ADCRH），这样转换数据就不会被下一次转换的数据覆盖。

ADCCVH/L = 0xxx

当比较功能使能时保留比较值

APCTL1=0x02

AD1 管脚 I/O 控制禁止。其他 AD 管脚仍为通用 I/O 管脚

APCTL2=0x00

所有其他 AD 管脚仍为通用 I/O 管脚

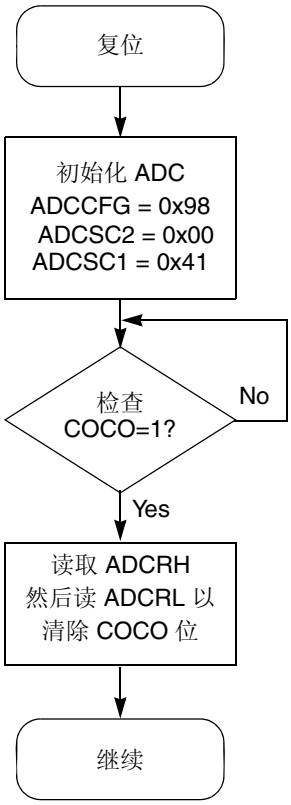

图 10-14. 初始化流程图示例

11.5 功能描述

本小节详细描述了 IIC 模块的全部功能。

11.5.1 IIC 协议

IIC 总线系统为数据传输使用串行数据线 （SDA）和串行时钟线 （SCL）。与其连接的所有器件必须具有开漏或开极输出。逻辑与功能通过外部上拉电阻在两条线上执行。这些电阻的值与系统相关。

一般地，标准通信由以下四部分组成：

- 启动信号
- 从机地址发送
- 数据传输
- 停止信号

停止信号不应与 CPU 停止指令相混淆。 IIC 总线系统通信将在后面进行简要地描述，并在图 11-9 中进行了阐释。

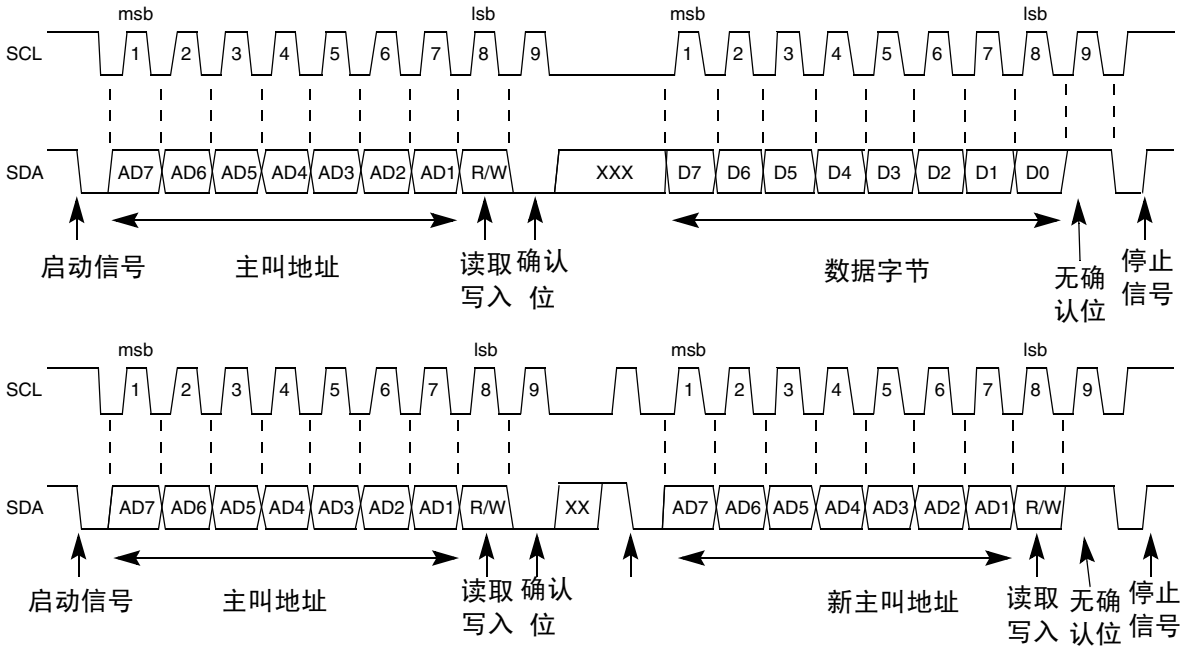

图 11-9. IIC 总线传输信号

S	从机前 7 位 11110 + AD10 + AD9	R/W 0	A1	从机第二个字节 AD[8:1]	A2	Sr	从机前 7 位 11110 + AD10 + AD9	R/W 1	A3	数据	A	...	数据	A	P

表 11-10. 主接收器寻址 10 位地址的从发射器

在主机发送器已经发送了 10 位地址的第一个字节后，从机接收器产生 IIC 中断。软件必须确保 IICD 的内容被忽略，且不作为该中断的有效数据对待。

11.5.3 通用呼叫地址

通用呼叫可以是 7 位地址或 10 位地址。如果设置了 GCAEM 位， IIC 就匹配通用呼叫地址及其自己的从机地址。当 IIC 响应通用呼叫时，它用作从接收器，且在地址周期后设置 IAAS 位。传输完首字节后，软件必须读取 IICD 寄存器，以确定是地址匹配其自己的从机还是通用呼叫。如果值为 00，匹配是通用呼叫。如果 GCAEN 位为 0， IIC 则通过不发送应答的方式忽略通用呼叫地址提供的任何数据。

11.6 复位

IIC 在复位后被禁止， IIC 不能引起 MCU 复位。

11.7 中断

IIC 只产生一个中断。

假设设置了 IICIE 位，当发生表 11-11 中的任意一个事件时， IIC 就生成中断。中断由位 IICIF（IIC 状态寄存器的位）驱动，用位 IICIE（IIC 控制寄存器的位）屏蔽。 IICIF 位必须通过软件在中断程序中向其写入 1 来清除。您可以通过读取状态寄存器确定中断类型。

表 11-11. 中断摘要

中断源	状态	标记	本地使能
完成 1 字节传输	TCF	IICIF	IICIE
匹配到收到的主叫地址	IAAS	IICIF	IICIE
仲裁丢失	ARBL	IICIF	IICIE

11.7.1 字节传输中断

TCF （传输完成标记）位在第 9 时钟的下降边沿设置，表示字节传输完成。

11.7.2 地址检测中断

当主叫地址匹配已编程的从机地址 （IIC 地址寄存器）或者当设置了 GCAEN 位且收到通用呼叫时，就设置状态寄存器中的 IAAS 位。假设设置了 IICIE， CPU 就被中断。 CPU 必须检查 SRW 位并相应设置其 Tx 模式。

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

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

表 12-36. CPU 与 MSCAN 运行模式之比较

CPU 模式	MSCAN 模式			
	正常	功耗降低		
		睡眠	断电	禁止 (CANE=0)
运行	CSWAI = X[1] SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = 1 SLPAK = 1		CSWAI = X SLPRQ = X SLPAK = X
等待	CSWAI = 0 SLPRQ = 0 SLPAK = 0	CSWAI = 0 SLPRQ = 1 SLPAK = 1	CSWAI = 1 SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X
Stop3		CSWAI = X[2] SLPRQ = 1 SLPAK = 1	CSWAI = X SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = X SLPAK = X
Stop1 or 2			CSWAI = X SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X

1 ‘X’ 表示 “不关心”

2 为了从睡眠模式安全唤醒，进入 STOP3 模式前，必须把 SLPRQ 和 SLPAK 设置为 1。

12.5.5.1 运行模式中的操作

如表 12-37 所示，当 CPU 处于运行模式时，只有 MSCAN 睡眠模式作为低功率选项。

12.5.5.2 等待模式中的操作

WAIT 指令将 MCU 置入低功耗待机模式中。如果设置了 CSWAI 位，还可以在断电模式下节省更多电力，因为 CPU 时钟停止。退出断电模式后， MSCAN 重新启动其内部控制器，并再次进入正常模式。

当 CPU 处于等待模式时， MSCAN 可以在正常模式下运行并生成中断 （寄存器可以通过背景调试模式访问）。根据 SLPRQ/SLPAK 和 CSWAI 位的值， MSCAN 也可以在任意一种低功率模式下运行，如在表 12-37 中看到的那样。

12.5.5.3 停止模式中的操作

STOP 指令将 CPU 置入低功耗待机模式中。在 STOP1 或 STOP2 模式中， MSCAN 在被置为断电模式，无论 SLPRQ/SLPAK 的值如何。在 STOP3 模式中，断电或睡眠模式由进入 STOP3 前设置的 SLPRQ/SLPAK 值决定。 CSWAI 位在任意一种停止模式中都不发挥作用。参见表 12-37。

12.5.5.6 MSCAN 断电模式

当出现以下情况时， MSCAN 处于断电模式 （表 12-36）

- CPU 处于停止模式，
  或
- CPU 处于等待模式且设置了 CSWAI 位

当进入断电模式时， MSCAN 立即停止正在进行的所有发送和接收，可能造成违反 CAN 协议。为了防止 CAN 总线系统出现违反上述规则的严重后果， MSCAN 立即驱动 TXCAN 管脚进入隐性状态。

注意

进入初始化模式时，用户负责保证 MSCAN 不在工作态。推荐步骤是在 CANCTL0 寄存器中设置 INITRQ 位前，把 MSCAN 置入睡眠模式 （SLPRQ = 1， SLPAK = 1）。否则，中止正在发送的报文可能导致错误情况，并影响到其他 CAN 总线节点。

在断电模式中，所有时钟停止，且不能访问寄存器。如果在断电模式 有效前 MSCAN 未处于睡眠模式，通电后该模块执行一个内部恢复周期。这会给模块再次进入正常模式带来某些固定延迟。

12.5.5.7 可编程唤醒功能

只要检测到 CAN 总线有效 （参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”中的控制位 WUPE）。就可以对 MSCAN 进行编程以唤醒 MSCAN。当处于睡眠模式时，通过将低通滤波器功能应用于 RXCAN 输入，可以更改 CAN 总线检测的灵敏度 （参见 12.3.2，“控制寄存器 1 (CANCTL1)”中的控制位 WUPM）。

该功能可以用来防止由于 CAN 总线线路上的短脉冲而唤醒 MSCAN。例如，嘈杂环境中的电磁干扰可以引起尖峰脉冲。

12.5.6 复位初始化

各个单个位的复位状态在 12.3，“寄存器定义”，其中详细阐述了所有寄存器及其位字段。

12.5.7 中断

本小节描述了由 MSCAN 引发的所有中断，列出了使能位和触发标志。文中单独列出并描述了每个中断。

12.5.7.1 中断运行描述

MSCAN 支持四个中断矢量 （参见表 12-37），任意一个矢量都可以单独屏蔽。 12.3.5，“MSCAN 接收器中断使能寄存器 (CANRIER)”至 12.3.7，“MSCAN 发送器中断使能寄存器 (CANTIER)”）。

注意

专用的中断矢量地址在 Resets and Interrupts 章中有详细说明。

表 14-6. SCIxS1 字段描述

字段	描述
5 RDRF	接收数据寄存器已满标记 — 当字符从接收移位器传输到接收数据寄存器 （SCIxD）时，设置 RDRF。要清除 RDRF，当 RDRF = 1 时读 SCIxS1，然后读 SCI 数据寄存器 （SCIxD）。 0 接收数据寄存器空。 1 接收数据寄存器已满。
4 IDLE	闲置线路标记 — 在一段时间的活动后，当 SCI 接收线路已经闲置了一个全字符时间时，就设置 IDLE。当 ILT = 0，接收器在起始位后开始计数闲置位时间。因此，如果接收字符都为 1，这些位时间和停止位时间计数入接收器用于探测一个闲置线路所需逻辑高态 （10 或 11 个位时间，取决于 M 控制位）的全字符时间。当 ILT = 1，接收器直到停止位后才开始计数闲置位时间。因此，停止位和前一字符末端的任何逻辑高态位时间不会计数入接收器用于探测一个闲置线路所需逻辑高态的全字符时间。 要清除 IDLE，当 IDLE = 1 时读取 SCIxS1，然后读取 SCI 数据寄存器 （SCIxD）。清除 IDLE 后，不能再次进行设置，直到接收到新字符且已设置了 RDRF。 IDLE 只设置一次，即便接收线路闲置了很长一段时间。 0 没有检测到闲置线路 1 检测到闲置线路
3 OR	接收器溢出标记 — 当新的串行字符做好了传输到接收数据寄存器 （缓冲器）的准备时，但原来接收的字符还没有从 SCIxD 读取，设置 OR。在这种情况下，新字符 （和所有相关错误信息）丢失，因为没有空间将它们移到 SCIxD。要清除 OR，当 OR = 1 时读 SCIxS1，然后读 SCI 数据寄存器 （SCIxD）。 0 没有溢出 1 接收溢出 （新 SCI 数据丢失）
2 NF	噪音标记 — 接收器中采用的先进的采样技术在起始位中提取 7 个样本，在每个数据位和停止位中提取 3 个样本。如果这些样本中任何一个样本与帧中任何位时间内的其余样本不一致，就要在 RDRF 为这个字符而置 1 的同时设置标记 NF。要清除 NF，读 SCIxS1，然后读 SCI 数据寄存器 （SCIxD）。 0 没有检测到噪音 1 SCIxD 中的已接收字符中检测到噪音
1 FE	成帧错误标记 — 当接收器在应该是停止位的时候检测到逻辑 0 时，同时设置 FE 和 RDRF。这表示接收器与字符帧没有完全统一。要清除 FE，当 FE = 1 时读 SCIxS1，然后读 SCI 数据寄存器 （SCIxD）。 0 未检测到成帧错误，这不能保证成帧正确。 1 成帧错误。
0 PF	奇偶效验错误标记 — 当奇偶效验使能 （PE = 1）且已接收字符中的奇偶校验位与预期奇偶效验值不一致时，同时设置 PF 和 RDRF。要清除 PF，读 SCIxS1，然后读 SCI 数据寄存器 （SCIxD）。 0 没有奇偶效验错误 1 奇偶效验错误

14.2.5 SCI 状态寄存器 2 (SCIxS2)

该寄存器有一个只读状态标记。

	7	6	5	4	3	2	1	0
R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
W								
复位	0	0	0	0	0	0	0	0

= 不执行或预留

图 14-9. SCI 状态寄存器 2 (SCIxS2)

当没有调试盒连接 6- 管脚的 BDM 接口连接器时，BKGD c 的内部上拉会选择正常的操作模式。当调试盒连接到 BKGD 时，可以在 MCU 复位后强制它进入激活背景调试模式。强制激活背景调试的具体条件取决于 HCS08 衍生产品 (参见 “开发支持”小节的介绍)。不必复位目标 MCU 来通过背景调试接口来与之通信。

17.2.2 通信详细介绍

BDC 串行接口需要外部控制器来生成 BKGD 管脚上的下降沿，指示每个位时间的开始。无论数据是发送或接收，外部控制器都会提供这个下降边沿。

BKGD 是伪开漏管脚，可以被外部控制器或 MCU 来驱动。数据以 MSB 先发的形式且以每位 16 个 BDC 时钟周期的速率 (标定速率) 发送。如果来自主机的下降边沿之间产生 512 BDC 时钟周期，则该接口超时。出果出现超时，任何正在进行的 BDC 命令被中止，对目标 MCU 系统的存储器或操作模式没有影响。

定制串行协议要求调试盒知道目标 BDC 通信时钟速率。

BDC 状态和控制寄存器中的时钟开关 (CLKSW) 控制位允许用户选择 BDC 时钟源。BDC 时钟源可以是总线，或备用的 BDC 时钟源。

BKGD 管脚可以接收高或低电平，或发送高或低电平。下图显示了每种情况的时序。接口时序与目标 BDC 中的时钟同步，但是与外部主机异步。显示的内部 BDC 时钟信号是计数周期的参考。

图 17-2 显示了外部主机将逻辑 1 或 0 发送到目标 HCS08 MCU 的 BKGD 管脚。主机与目标异步，因此主机生成的 BKGD 下降边沿与目标所认为的位时间起始点有 0- 到 -1 周期的延迟。 10 个目标 BDC 时钟周期后，目标获得 BKGD 管脚的电平。一般地，主机在主机到目标方向的传输过程中驱动 BKGD 管脚 ，以加快上升边沿。由于目标在主机至目标方向的传输周期中不驱动 BKGD 管脚，因此没有必要在此期间将线路作为开漏信号。

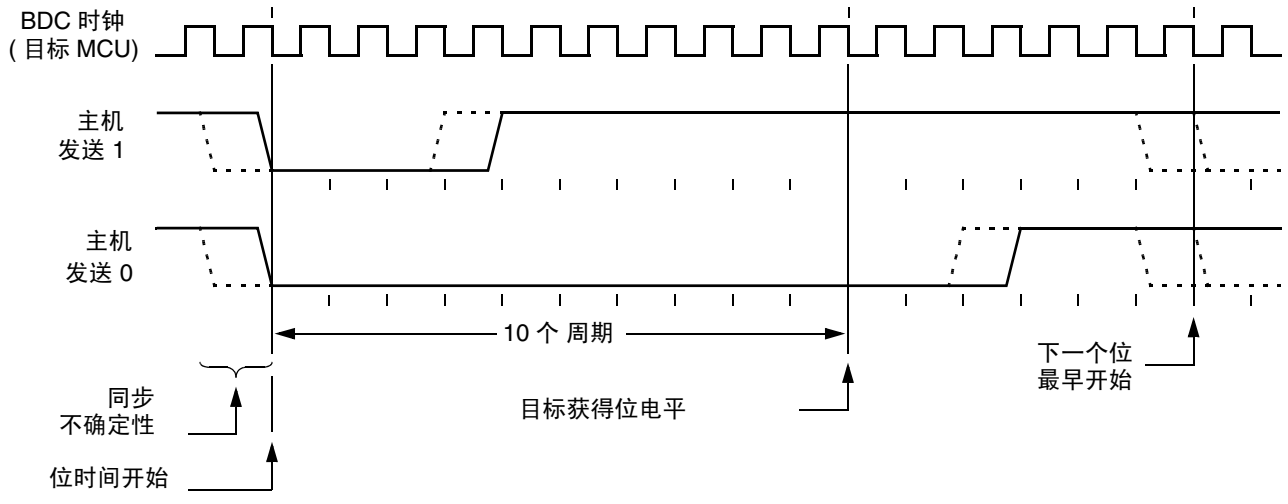

图 17-2. BDC 主机到目标方向串行位时序

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 DBGT 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。

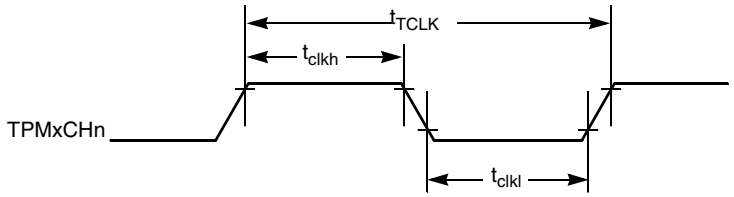

图 A-5. 定时器外部时钟

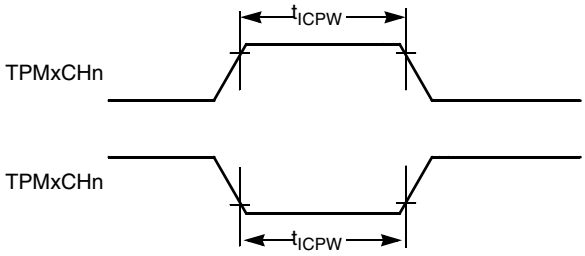

图 A-6. 定时器输入捕捉脉冲

A.12.3 MSCAN

表 A-15. MSCAN 唤醒脉冲特性

编号	C	参数	符号	最小值	典型值值	最大值	单位
14	D	MSCAN 唤醒显性脉冲过滤	t_{WUP}	—	—	2	ms
15	D	MSCAN 唤醒显性脉冲通过	t_{WUP}	5	—	—	ms

如果通道值寄存器 TPMxCnVH:TPMxCnVL 为零或负数 （位 15 被设置），工作周期将是 0%。如果 TPMxCnVH:TPMxCnVL 是正值 （位 15 被清除）并大于 （非零）模数设置，工作周期将为 100%，因为工作周期比较将不会发生。这意味着模数寄存器设置的可用周期范围为 0x0001 至 0x7FFE （如果不需要 100% 的工作周期，则为 0x7FFF）。这不是一个重要的限制条件，因为结果周期远远长于正常应用所需的周期。

TPMxMODH:TPMxMODL = 0x0000 是不应与中央对齐 PWM 模式一同使用的特例。当 CPWMS=0 时，这一情况与在 0x0000 和 0xFFFF 之间自由运行的计数器对应，然而当 CPWMS=1 时，计数器需要与 0x0000 以外的某个模数寄存器值有效匹配，以便将方向从向上计数改变为向下计数。

图 B-11 显示了 TPM 通道寄存器 （乘以 2）中的输出比较值。该值决定 CPWM 信号的脉冲宽度（工作周期）。如果 ELSnA=0，向上计数时的比较匹配会强制 CPWM 输出信号降低；而向下计数时的比较匹配会强制输出升高。计数器向上计数，直到达到中的模数设置，然后向下计数，直到 0。这样就可将周期设置为 TPMxMODH:TPMxMODL 的两倍。

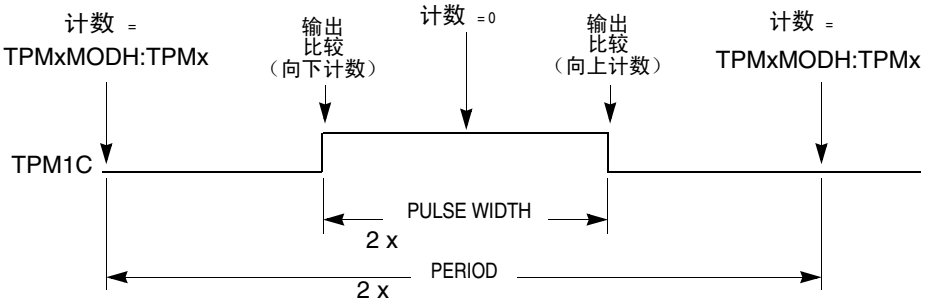

图 B-11. CPWM 周期和脉冲宽度 (ELSnA = 0)

中央对齐 PWM 输出生成的噪音一般比边缘对齐 PWM 小，因为相同系统时钟边上排列的输入 / 输出过渡更少。某些类型的电机也需要这类 PWM。

因为 HCS08 是一个 8 位 MCU 系列。定时器通道寄存器中的设置被缓存，以确保连贯的 16 位更新并避免意外的 PWM 脉冲宽度。写入到任何寄存器，不管是 TPMxMODH、TPMxMODL, TPMxCnVH、还是 TPMxCnVL, 实际就是写入到缓冲器寄存器。只有在 16 位寄存器的两个 8 位字节都被写入而且定时器计数器溢出后 （在模数寄存器上的终端计数结束后从向上计数改变为向下计数），值才被发送到相应的定时器通道寄存器中。这一 TPMxCNT 溢出要求只适用于 PWM 通道，而不是输出比较。

当 TPMxCNTH:TPMxCNTL = TPMxMODH:TPMxMODL 时， TPM 可在计数结束时生成一个 TOF 中断。用户可以选择重新上载任何数量的 PWM 缓冲器，并且它们可以在新的周期开始时同时更新。

TPMxSC 写入操作会取消写入到 TPMxMODH and/or TPMxMODL 中的任何值，并且为模数寄存器复位一致性机制。 TPMxCnSC 写入操作会取消写入到通道值寄存器中的任何值，并且为 TPMxCnVH:TPMxCnVL 复位一致机制。