



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, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	39
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K 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/mc9s08dz16mlf

1.3 系统时钟分配

图 1-2 为简化的时钟连接结构图。MCU 的某些模块的时钟输入可选。到各模块的时钟输入用于驱动该模块的功能。

下面列出了本 MCU 中使用的时钟：

- BUSCLK — 总线频率始终为 MCGOUT 的一半
- LPO — 独立的 1 kHz 时钟，可以作为 COP 和 RTC 模块的时钟源。
- MCGOUT — MCG 的主输出，为总线频率的两倍。
- MCGLCLK — 在 BUSCLK 被配置为以很低的频率运行的系统中，开发工具可以选择这一时钟源来加快 BDC 通信。
- MCGERCLK — 外部参考时钟，可用作 RTC 时钟源。它还可以用作 ADC 和 MSCAN 的备用时钟。
- MCGIRCLK — 内部参考时钟，可用作 RTC 时钟源。
- MCGFFCLK — 固定频率时钟，可用作 TPM1 和 TPM2 的时钟源。
- TPM1CLK — TPM1 的外部输入时钟源。
- TPM2CLK — TPM2 的外部输入时钟源。

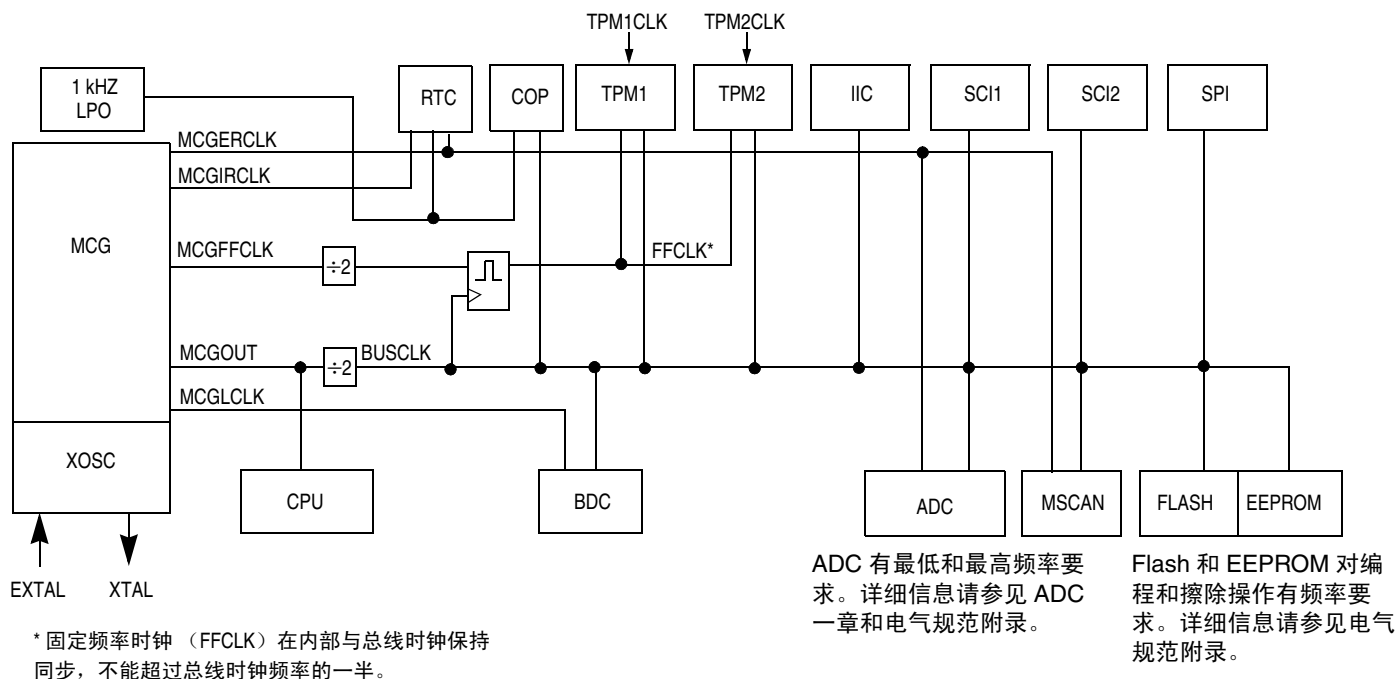


图 1-2. MC9S08DZ60 系统时钟分配图

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。

5.8.1 中断管脚请求状态和控制寄存器 (IRQSC)

该直接页面寄存器包括用来配置 IRQ 功能、报告状态和确认 IRQ 事件的状态和控制位。

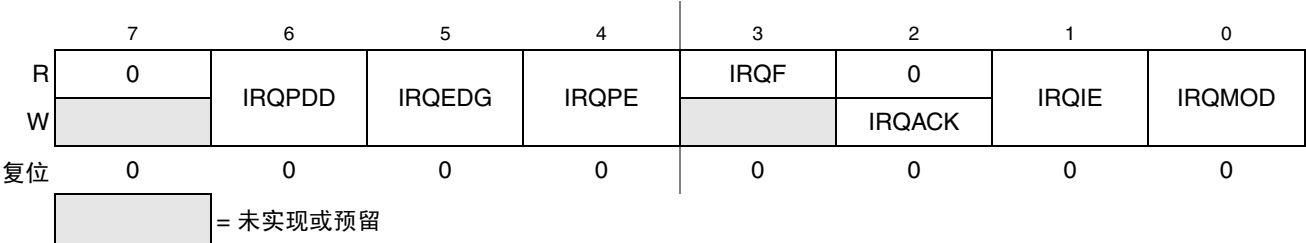


图 5-2. 中断请求状态和控制寄存器 (IRQSC)

表 5-2. IRQSC 寄存器字段描述

字段	描述
6 IRQPDD	中断请求 (IRQ) 上拉器件禁止 — IRQ 管脚激活时 (IRQPE = 1)，这个读 / 写控制位用来禁止内部上拉 / 下拉器件，从而允许使用外部器件。 0 如果 IRQPE = 1，IRQ 上拉器件激活； 1 如果 IRQPE = 1，IRQ 上拉器件禁止。
5 IRQEDG	
4 IRQPE	IRQ 管脚使能 — 这个读写控制位使能 IRQ 管脚。当设置了该位时，IRQ 管脚可以用作中断请求。 0 IRQ 管脚禁止； 1 IRQ 管脚使能。
3 IRQF	IRQ 标志 — 该只读状态位显示中断请求事件发生的时间。 0 无 IRQ 请求； 1 检测到 IRQ 事件。
2 IRQACK	IRQ 确认 — 这个只写位用来确认中断请求事件（写 1 来清除 IRQF）。写入 0 则没有任何意义或影响。读总是返回 0。如果选择了边沿和电平检测（IRQMOD = 1），则不能清除 IRQF，并且 IRQ 管脚位于断言级。
1 IRQIE	IRQ 中断使能 — 这个读 / 写控制位决定 IRQ 事件是否生成中断请求。 0 当 IRQF = 1 时不产生中断请求（使用轮询）； 1 当 IRQF = 1 时产生中断请求。
0 IRQMOD	IRQ 检测模式 — 这个读 / 写控制位选择只边沿检测还是边沿和电平检测。IRQEDG 控制位决定检测为中断请求事件的边沿和电平极性。如需了解更多信息，请参见 5.5.2.2，“边沿和电平敏感度”。 0 只下降边沿或上升边沿的 IRQ 事件； 1 下降边沿和低电平 IRQ 事件或上升边沿和高电平 IRQ 事件。

5.8.2 系统复位状态寄存器 (SRS)

该高页寄存器包括只读状态标记，以显示最近复位的源。将 1 写入 SBDFR 寄存器的 BDFR 时，调试主机强制完成复位，SRS 中不设置任何状态位。COP 使能时，在寄存器地址中写入任意值

(0x55 和 0xAA 除外) 都会导致 COP 复位。在该地址中写入 0x55-0xAA 顺序会清除 COP 看门狗定时器, 但不会对寄存器内容造成影响。这些位的复位状态取决于导致 MCU 复位的原因。

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	ILAD	LOC	LVD	0
W	在 SRS 地址中写入 0x55 和 0xAA, 清除 COP 看门狗定时器							
POR:	1	0	0	0	0	0	1	0
LVD:	u	0	0	0	0	0	1	0
其他 复位:	0	注意 (1)	注意 (1)	注意 (1)	注意 (1)	0	0	0

¹ 复位时活动的任何复位源会造成设置相应的位; 复位时不活动的源对应的位将被清除。

图 5-3. 系统复位状态 (SRS)

表 5-3. SRS 寄存器字段描述

字段	描述
7 POR	加电复位 — 复位由电源开检测逻辑造成。由于此时内部电源电压在上升, 所以还要设置低压复位 (LVD) 状态位以显示已进行的复位, 这时内部电源低于 LVD 阈值。 0 非 POR 造成的复位。 1 POR 造成的复位。
6 PIN	外部复位管脚 — 复位由外部复位管脚上的活动低电平造成。 0 非外部复位管脚造成的复位。 1 外部复位管脚造成的复位。
5 COP	计算机正常操作 (COP) 看门狗 — 复位由 COP 看门狗定时器超时导致。该复位源可以由 COPE = 0 拦截。 0 非 COP 超时造成的复位。 1 超时造成的复位。
4 ILOP	非法操作码 — 复位由试图实施未实现或非法操作码导致。如果停止由 SOPT 寄存器里 STOPE = 0 禁止, STOP 指令被视为非法。如果主动后台模式由 BDCSC 寄存器里 ENBDM = 0 禁止, BGND 指令指令被视为非法。 0 不是由非法操作码造成的复位。 1 非法操作码造成的复位。
3 ILAD	非法地址 — 复位由试图在未实现的存储器地址上存取数据或指令导致。 0 不是由非法地址造成的复位。 1 由非法地址造成的复位。
2 LOC	时钟丢失 — 复位由外部时钟丢失导致。 0 不是由外部时钟丢失造成的复位。 1 由外部时钟丢失造成的复位。
1 LVD	低压检测 — 如果设置了 LVDRE 位且电源降到 LVD 脱扣电压以下, 就会发生 LVD 复位。该位也可以由 POR 设置。 0 不是由 LVD 脱扣或 POR 造成的复位。 1 由 LVD 脱扣或 POR 造成的复位。

第 6 章

并行输入 / 输出控制

本小节解释了与并行输入 / 输出和管脚控制相关的软件控制。MC9S08DZ60 系列有 7 个并行输入 / 输出端口，这 7 个端口总共包含 53 个输入 / 输出管脚和 1 个仅输入管脚。如需了解这些管脚的管脚分配和外部硬件注意事项的更多信息，请参见第 2 章，“管脚和连接”。

这些管脚中的很多都在片上外围设备中共用，如定时器系统、通信系统和管脚中断，如表 2-1 所示。外围设备模块的优先级把通用输入 / 输出功能的优先级高，因此当使能某个外围设备时，与该共用管脚相关的输入 / 输出功能被禁止。

复位后，共用外围设备功能被禁止，管脚被配置为输入（PTxDDn = 0）。每个管脚的管脚控制功能都配置如下：斜率控制使能（PTxSEn = 1）、低驱动强度选定（PTxDSn = 0）、内部上拉被禁止（PTxPEN = 0）。

注意

不是所有封装都提供通用输入 / 输出管脚。为了避免从输入引脚浮接抽取过多电流，应用程序中的用户复位初始化程序必须要么使能片上拉器件，要么将未连接管脚的方向更改为输出，使管脚不会浮接。

6.1 端口数据和数据方向

通过端口数据寄存器执行并行输入 / 输出读取 / 写入。不管是输入还是输出方向，都由端口数据方向寄存器控制。图 6-1 中的块状示意图介绍了单个管脚的并行输入 / 输出端口功能。

数据方向控制位（PTxDDn）决定是否启动相关管脚使用的输出缓冲器，同时控制端口数据寄存器读取的源。相关管脚的输入缓冲器总是处于使能状态，除非管脚用作模拟功能或输出管脚。

当为管脚使能共用数字功能时，输出缓冲器由共用功能控制。但是，数据方向寄存器位将继续控制端口数据寄存器读取的源。

当为管脚使能共用模拟功能时，输出和输出缓冲器都被禁止。当该位为输入位（PTxDDn = 0），输入缓冲器禁止时，任意端口数据位的读数均为 0。总体来说，每当数字功能和模拟功能共用一个管脚时，模拟功能都优先。因此数字和模拟功能同时使能时，管脚由模拟功能控制。

一个不错的编程习惯是在把端口管脚方向修改为输出前就写入端口数据寄存器，这确保不会用在端口数据寄存器内的旧数据值来临时驱动管脚。

6.5.3.1 C 端口数据寄存器 (PTCD)

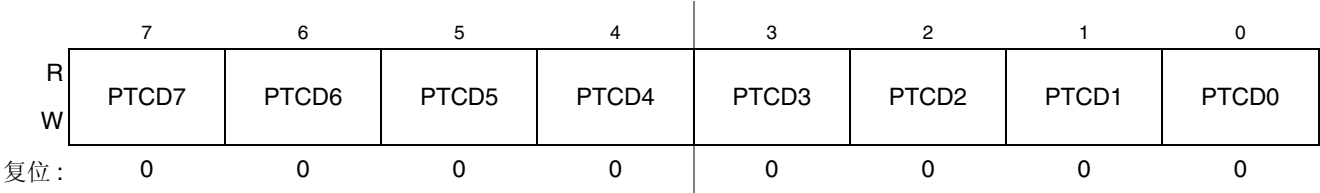


图 6-19. C 端口数据寄存器 (PTCD)

表 6-17. PTCD 寄存器字段描述

字段	描述
7:0 PTCD[7:0]	C 端口数据寄存器位 — 对于配置为输入的 C 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 C 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 C 端口管脚，逻辑电平被输出到相应的 MCU 管脚。 复位强制 PTCD 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 / 下拉禁止的高抗阻输入。

6.5.3.2 C 端口数据方向寄存器 (PTCDD)

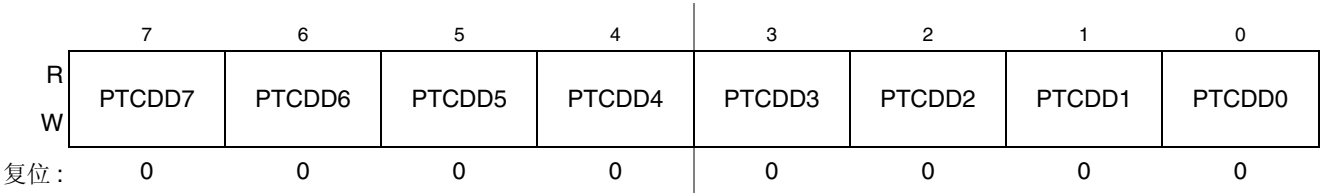


图 6-20. C 端口数据方向寄存器 (PTCDD)

表 6-18. PTCDD 寄存器字段描述

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

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。

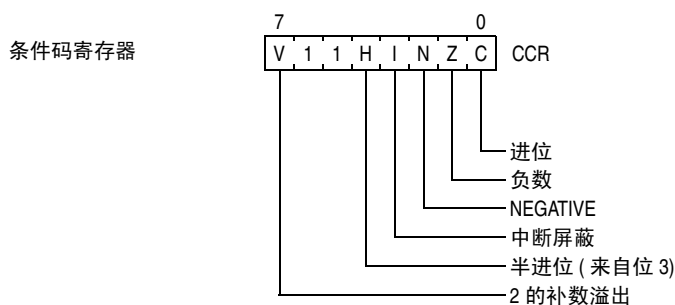
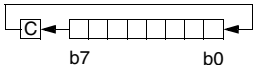
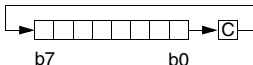


图 7-2. 条件码寄存器

表 7-2. 指令集小结 (第 6 页, 共 9 页)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	移动 (M) _{destination} ← (M) _{source} 在 IX+/DIR 和 DIR/IX+ 模式, H:X ← (H:X) + \$0001	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rpwpp rhwpp pwpp rhwpp	0 1 1 -	-↑↑↓-
MUL	不带符号的乘法 X:A ← (X) × (A)	INH	42	5	ffffp	- 1 1 0	- - - 0
NEG <i>opr8a</i> NEGA NEGX NEG <i>opr8,X</i> NEG <i>,X</i> NEG <i>opr8,SP</i>	否定 (2 的补数) M ← (M) = \$00 - (M) A ← (A) = \$00 - (A) X ← (X) = \$00 - (X) M ← (M) = \$00 - (M) M ← (M) = \$00 - (M) M ← (M) = \$00 - (M)	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rhwpp p p rhwpp rhw prhwpp	↑ 1 1 -	-↑↑↑
NOP	无操作 — 使用 1 总线周期	INH	9D	1	p	- 1 1 -	- - - -
NSA	半字节交换累加器 A ← (A[3:0]:A[7:4])	INH	62	1	p	- 1 1 -	- - - -
ORA <i>#opr8i</i> ORA <i>opr8a</i> ORA <i>opr16a</i> ORA <i>opr8,X</i> ORA <i>opr8,X</i> ORA <i>,X</i> ORA <i>opr8,SP</i> ORA <i>opr8,SP</i>	累加器或存储器 " 兼或 " A ← (A) (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	-↑↑↓-
PSHA	将累加器推送到堆栈 推 (A); SP ← (SP) - \$0001	INH	87	2	sp	- 1 1 -	- - - -
PSHH	将 H (索引寄存器高) 推送到堆栈上 推 (H); SP ← (SP) - \$0001	INH	8B	2	sp	- 1 1 -	- - - -
PSHX	将 X (索引寄存器低) 推送到堆栈上 推 (X); SP ← (SP) - \$0001	INH	89	2	sp	- 1 1 -	- - - -
PULA	从堆栈拉累加器 SP ← (SP + \$0001); 拉 (A)	INH	86	3	ufp	- 1 1 -	- - - -
PULH	从堆栈拉 H (索引寄存器高) SP ← (SP + \$0001); Pull (H)	INH	8A	3	ufp	- 1 1 -	- - - -
PULX	从堆栈拉 X (索引寄存器低) SP ← (SP + \$0001); 拉 (X)	INH	88	3	ufp	- 1 1 -	- - - -
ROL <i>opr8a</i> ROLA ROLX ROL <i>opr8,X</i> ROL <i>,X</i> ROL <i>opr8,SP</i>	通过进位左旋转 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rhwpp p p rhwpp rhw prhwpp	↑ 1 1 -	-↑↑↑
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	通过进位右旋转 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rhwpp p p rhwpp rhw prhwpp	↑ 1 1 -	-↑↑↑

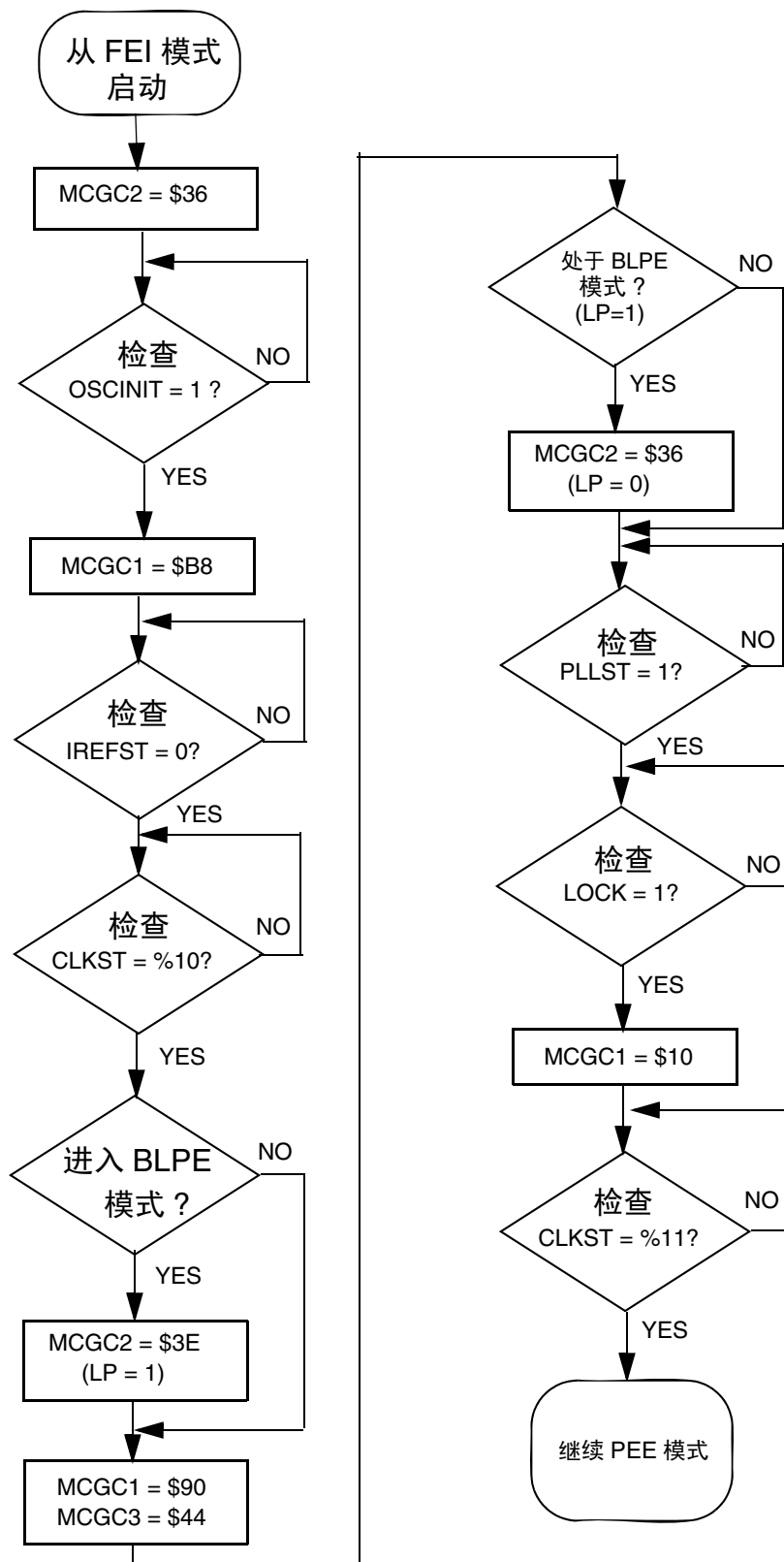


图 8-9. 使用 4 MHz 晶体从 FEI 转换到 PEE 模式的流程图

10.3 外部信号描述

ADC 模块最多可支持 28 个独立模拟输入。它还需要 4 个电源 / 参考 / 接地连接。

表 10-2. 信号属性

名称	功能
AD27-AD0	模拟通道输入
V_{REFH}	高参考电压
V_{REFL}	低参考电压
V_{DDAD}	模拟电源
V_{SSAD}	模拟接地

10.3.1 模拟电源 (V_{DDAD})

ADC 模拟部分使用 V_{DDAD} 作为其电源连接。在有些封装中, V_{DDAD} 与 V_{DD} 是内部连接。如果是外部连接, 将 V_{DDAD} 管脚连到与 V_{DD} 相同的电平。为了确保干净的 V_{DDAD} 信号, 可能还需要外部滤波。

10.3.2 模拟接地 (V_{SSAD})

ADC 模拟部分使用 V_{SSAD} 作为其接地连接。在有些封装中, V_{SSAD} 与 V_{SS} 是内部连接。如果是外部连接, 将 V_{SSAD} 管脚连到与 V_{SS} 相同的电平。

10.3.3 参考电压高 (V_{REFH})

V_{REFH} 是转换器的高参考电压。在有些封装中, V_{REFH} 与 V_{DDAD} 是内部连接。如果是外部连接, V_{REFH} 可以连接到与 V_{DDAD} 相同的电平, 或者由介于 V_{DDAD} 最低限值和 V_{DDAD} 电平之间的外部源驱动 (V_{REFH} 必须不能超过 V_{DDAD})。

10.3.4 参考电压低 (V_{REFL})

V_{REFL} 是转换器的低参考电压。在有些封装中, V_{REFL} 与 V_{SSAD} 是内部连接。如果是外部连接, 将 V_{REFL} 管脚连到和 V_{SSAD} 相同的电平。

10.3.5 模拟通道输入 (ADx)

ADC 模块最多可支持 28 个独立的模拟输入。通过 $ADCH$ 通道选择位选择转换。

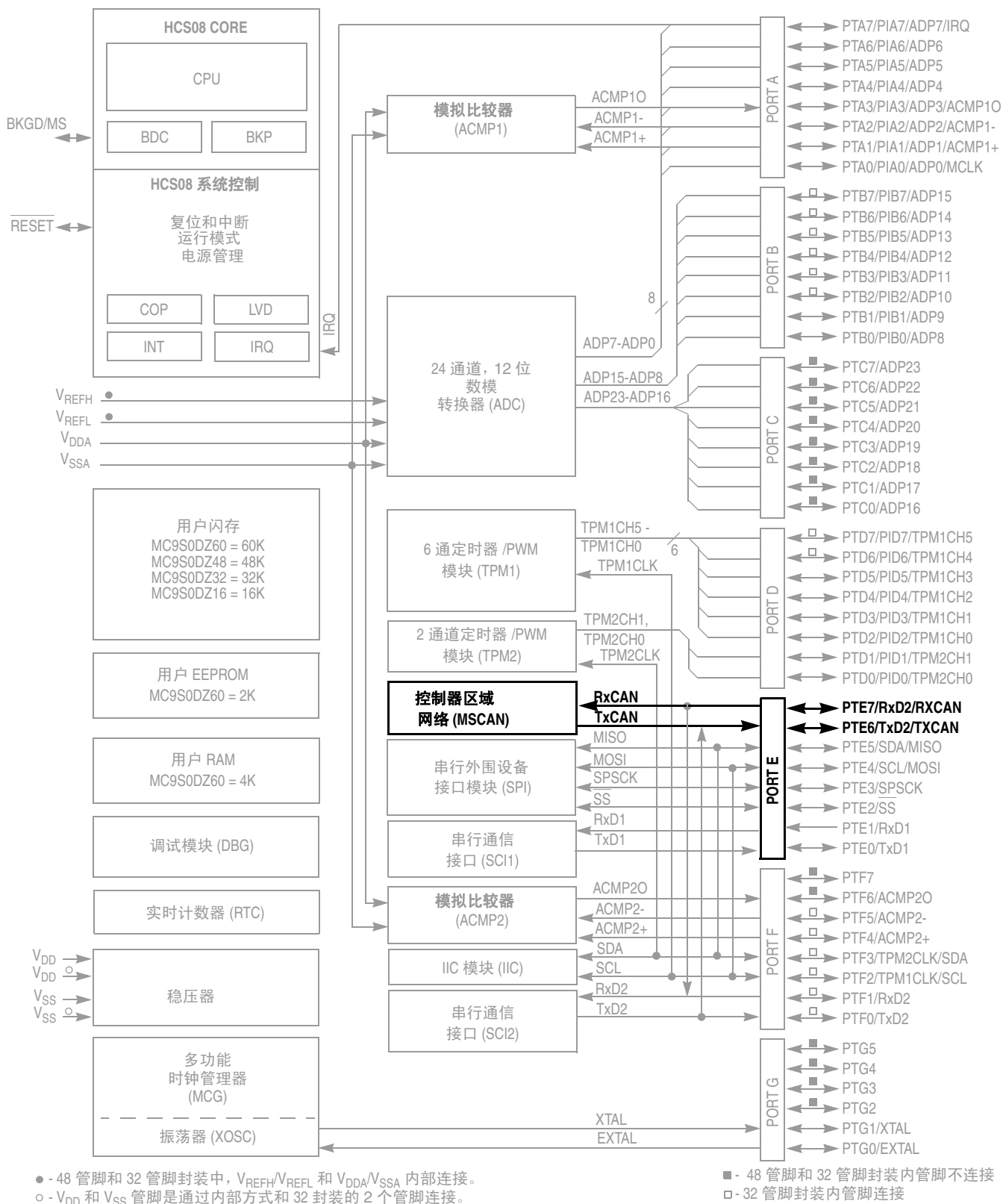


图 12-1. MC9S08DZ60 结构图

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。

13.2 外部信号描述

SPI 共享 4 个端口管脚。这些管脚的功能取决于 SPI 控制位的设置。当 SPI 禁止 ($SPE = 0$) 时, 这 4 个管脚恢复为不受 SPI 控制的通用端口 I/O 管脚。

13.2.1 SPSCK — SPI 串行时钟

当 SPI 作为从 SPI 被使能时, 该管脚是串行时钟输入。当 SPI 作为主 SPI 被使能时, 该管脚是串行时钟输出。

13.2.2 MOSI — Master Data Out, Slave Data In

当 SPI 作为主 SPI 被使能且 SPI 管脚控制零 ($SPC0$) 为 0 (不是双向模式) 时, 该管脚是串行数据输出。当 SPI 作为从 SPI 被使能且 $SPC0 = 0$ 时, 该管脚为串行数据输入。如果 $SPC0 = 1$, 选择单线双向模式并选择了主模式, 该管脚就成为双向数据 I/O 管脚 ($MOMI$)。同样, 双向模式输出使能位决定该管脚作为输入 ($BIDIROE = 0$) 管脚还是输出管脚 ($BIDIROE = 1$)。如果 $SPC0 = 1$ 且选择了辅模式, 该管脚不会被 SPI 使用, 并恢复为通用端口 I/O 管脚。

13.2.3 MISO — Master Data In, Slave Data Out

当 SPI 作为主 SPI 被使能且 SPI 管脚控制零 ($SPC0$) 为 0 (不是双向模式) 时, 该管脚是串行数据输入。当 SPI 作为从 SPI 被使能且 $SPC0 = 0$ 时, 该管脚是串行数据输出。如果 $SPC0 = 1$, 选择单线双向模式并选择了辅模式, 该管脚变成双向数据 I/O 管脚 ($SISO$)。双向模式输出使能位决定该管脚作为输入 ($BIDIROE = 0$) 管脚还是输出管脚 ($BIDIROE = 1$)。如果 $SPC0 = 1$ 且选择了辅模式, 该管脚不会被 SPI 使用, 并恢复为通用端口 I/O 管脚。

13.2.4 $\overline{SS}$ — 从选择

当 SPI 作为从 SPI 被使能时, 该管脚是低电平有效的从选择输入。当 SPI 作为主 SPI 被使能且模式默认使能关闭 ($MODFEN = 0$) 时, 该管脚不被 SPI 使用, 并恢复为通用端口 I/O 管脚。当 SPI 作为主 SPI 被使能时, $MODFEN = 1$, 辅选择输出使能位决定该管脚是作为模式默认输入 ($SSOE = 0$) 还是辅选择输出 ($SSOE = 1$)。

13.3 运行模式

13.3.1 处于停止模式的 SPI

SPI 在所有停止模式禁止, 无论在执行 STOP 指令前的设置如何。在 STOP1 或 STOP2 模式期间, SPI 模块将完全关闭。一旦从 STOP1 或 STOP2 模式唤醒, SPI 模块将进入复位状态。在 STOP3 模式期间, SPI 模块的时钟暂停。寄存器没受到影响。如果通过复位退出 STOP3, SPI 被置复位状态。如果通过中断退出 STOP3, SPI 继续保持其进入 STOP3 时所处的状态。

图 14-3 SCI 接收器结构图

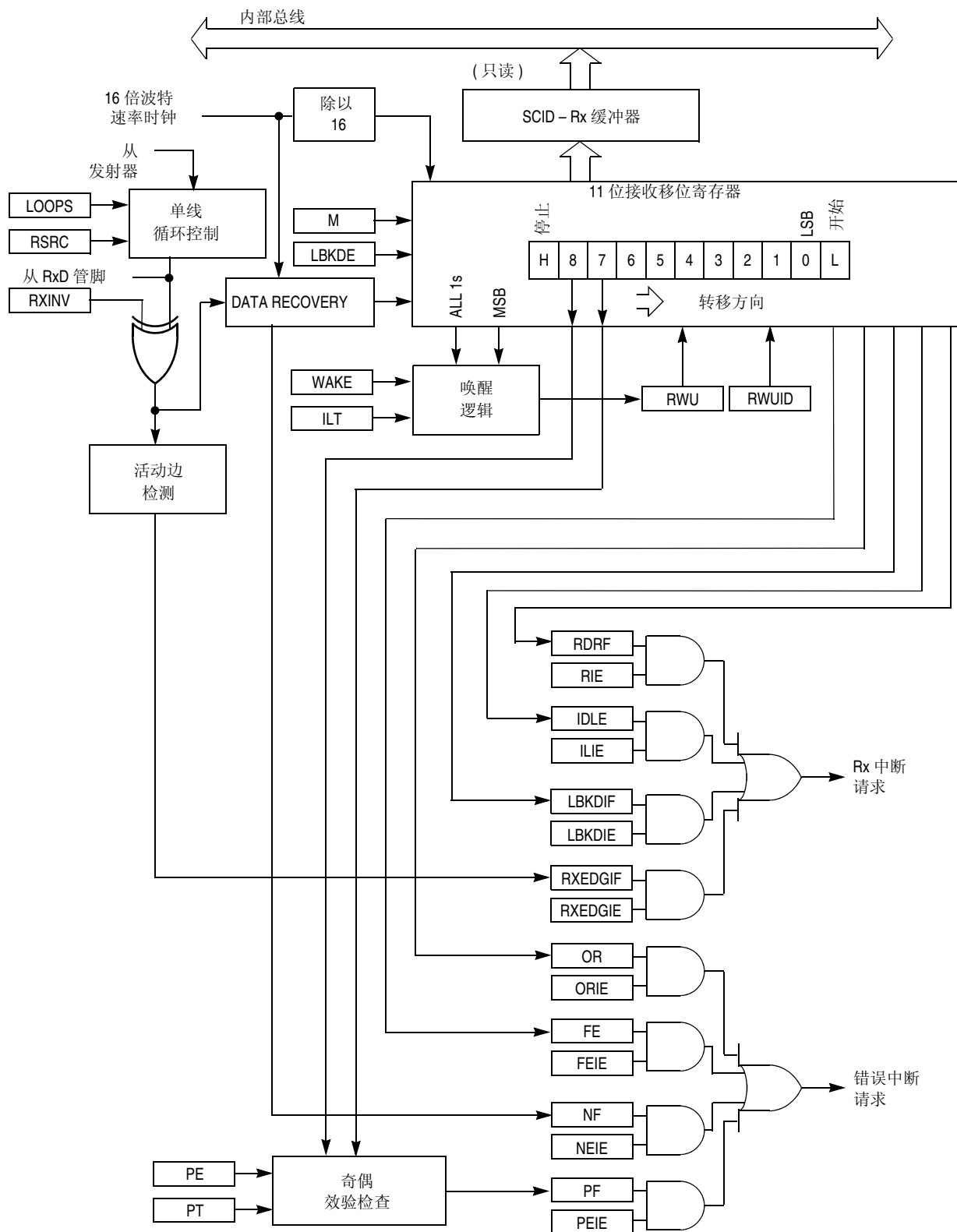


图 14-3. SCI 接收器结构图

如果相应 TIE 或 TCIE 本地中断允许位为 0，那么就使用软件轮询来监控 TDRE 和 TC 状态标记，而不是发生软件中断。

当程序检测到接收数据寄存器已满（RDRF = 1）时，它通过读 SCiXD 从接收数据寄存器获取数据。RDRF = 1 时读 SCiXS1，这样 RDRF 标记就被清除，然后再读 SCiXD。

当使用轮询时，该顺序自然在用户程序的正常过程中得到满足。如果使用硬件中断，就必须在中断服务程序（ISR）中读 SCiXS1。在正常情况下，这无论如何都要在 ISR 中完成，以检查接收错误，这样该顺序就自动满足了。

当 RxD 线路在很长一段时间内保持闲置时，IDLE 状态标记包括可以防止其进行重复设置的逻辑。当 IDLE = 1 时，读 SCiXS1 可以清除 IDLE，然后再读 SCiXD。在已经清除 IDLE 后，它不能再次进行设置，直到接收器已经收到至少一个新字符并已设置 RDRF。

如果在造成设置 RDRF 的已接收字符中检测到有关错误，就在设置 RDRF 的同时设置错误标记，如噪音标记（NF）、成帧错误（FE）和奇偶效验错误标记（PF）。这些标记不会在溢出情况下设置。

如果当一个新字符准备好从接收移位器传输到接收数据缓冲器时已经设置了 RDRF，就设置溢出（OR）标记，而数据及任何有关的 NF、FE 或 PF 条件丢失。

任何时候，RxD 串行数据输入管脚上的活动边沿都会引起 RXEDGIF 标记设置。把 1 写入 RXEDGIF 会清除该标记。该功能取决于正被使能（RE = 1）的接收器。

14.3.5 其他 SCI 功能

以下几节描述其他 SCI 功能。

14.3.5.1 8 位和 9 位数据模式

通过在 SCiXC1 中设置 M 控制位，SCI 系统（发射器和接收器）在经过配置后可以运行在 9 位数据模式中。在 9 位模式中，在 SCI 数据寄存器的 MSB 的左侧有一个第 9 数据位。对发送数据缓冲器来说，该位保存在 SCiXC3 中的 T8。对接收器来说，第 9 位保存在 SCiXC3 中的 R8。

为了连贯写入发送数据寄存器，写入 SCiXD 前请先写入 T8 位。

如果作为新字符第 9 位发送的位值和上一字符的位值相同，则不需要重新写入 T8。当数据从发送数据缓冲器传输到发送移位器时，在数据从 SCiXD 传输到移位器的同时 T8 中的值被复制。

9 位数据模式通常和奇偶效验一起使用，以允许数据的 8 个位加第 9 位中的奇偶效验位。或者与地址标记唤醒一起使用，这样第 9 数据位可以作为唤醒位。在自定义协议中，第 9 位还可以作为软件控制标记。

外部时钟信号与通道输入 / 输出管脚共享相同的管脚，因此选择为外部时钟源时通道管脚不能用于通道输入 / 输出功能。用户应负责避免这种设置。如果这个管脚被用作外部时钟源 (CLKSB:CLKSA = 1:1)，通道仍可作为软件定时器 (ELSnB:ELSnA = 0:0) 用于输出比较模式。

16.2.1.2 TPMxCHn — 通道 n 输入 / 输出管脚

每个 TPM 通道都与 MCU 上的一个输入 / 输出管脚相关联。这个管脚的功能取决于通道配置。TPM 管脚与通用输入 / 输出管脚共享，其中每个管脚都有一个端口数据寄存器位和一个数据方向控制位，而且端口有可选的被动上拉器件。该上拉器件可在端口管脚作为输入设备时使能。

当 (ELSnB:ELSnA = 0:0) 或 (CLKSB:CLKSA = 0:0) 时，TPM 通道不会控制输入 / 输出管脚，因此通常恢复到由通用输入 / 输出控制的状态。当 CPWMS = 1 (and ELSnB:ELSnA not = 0:0) 时，TPM 中的所有通道被配置用于中央对齐 PWM，而 TPMxCHn 管脚全部由 TPM 系统控制。当 CPWMS=0 时，MSnB:MSnA 控制位决定通道配置用于输入捕捉、输出比较还是边缘对齐 PWM。

当通道被配置用于输入捕捉 (CPWMS=0, MSnB:MSnA = 0:0 and ELSnB:ELSnA not = 0:0) 时，TPMxCHn 管脚被强制用作 TPM 的对边缘敏感的输入。ELSnB:ELSnA 控制位决定哪个或哪些极性边将触发输入捕捉事件。一个基于总线时钟的同步器用于同步输入边和总线时钟。这意味着输入捕捉管脚上可以可靠检测的最小脉冲宽度是 4 个总线时钟周期 (可检测尽可能靠近 2 个总线时钟的最佳时钟脉冲)。TPM 使用该管脚作为输入捕捉输入，为相同管脚改写端口数据和数据方向控制。

当通道被配置用于输出比较 (CPWMS=0, MSnB:MSnA = 0:1 and ELSnB:ELSnA not = 0:0) 时，相关数据方向控制被改写；TPMxCHn 管脚被看作是由 TPM 控制的输出，ELSnB:ELSnA 控制位决定如何控制管脚。ELSnB:ELSnA 的其余三个组合决定在每次 16 位通道值寄存器与定时器计数器匹配时是否切换、清除或者设置 TPMxCHn 管脚。

刚完成选择输出比较切换模式时，管脚上的以前的值一直被驱动，直到发生下一个输出比较事件，然后管脚被切换。

16.4 功能描述

所有 TPM 功能都与一个允许灵活选择时钟源和预分频器因数的中央 16 位计数器相关。此外，还有一与主计数器关联的 16 位模数寄存器。

CPWMS 控制位可在 PWM 中所有通道的中央对齐 TPM 操作 (CPWMS=1) 或通用定时功能 (CPWMS=0) 间选择。在后一种情况下，每个通道可独立配置，以输入捕捉、输出比较或边缘对齐 PWM 模式运行。CPWMS 控制位位于主 TPM 状态和控制寄存器中，因为它会影响 TPM 中的所有通道，而且会影响主计数器的运行方式。(在 CPWM 模式下，计数器变为向上 / 向下模式，而不是用于通用定时器功能的向上计数模式。)

后面各小节介绍了主计数器及计数器的每一种运行模式 (输入捕捉、输出比较、边缘对齐 PWM 和中央对齐 PWM)。因为管脚运行和中断活动的细节取决于运行模式，这些主题将在相关模式的说明部分中介绍。

16.4.1 计数器

所有定时器功能都基于 16 位主计数器 (TPMxCNTH:TPMxCNTL)。这一部分讨论时钟源的选择、计数终止溢出、向上计数和向下计数以及手动计数器复位。

16.4.1.1 计数器时钟源

计数器状态和控制寄存器 (TPMxSC) 中的 2 位字段 CLKSB:CLKSAs 从三个可能的时钟源中进行选择或选择 OFF (可有效地关闭 TPM)。请参见表 16-3。任何 MCU 复位后，CLKSB:CLKSA=0:0，因此不会选择任何时钟源，TPM 处于非常低功耗的状态。这些控制位可随时读取或写入，关闭定时器 (将 00 写入到 CLKSB:CLKSA f 字段) 不会影响计数器或其他定时器寄存器中的值。

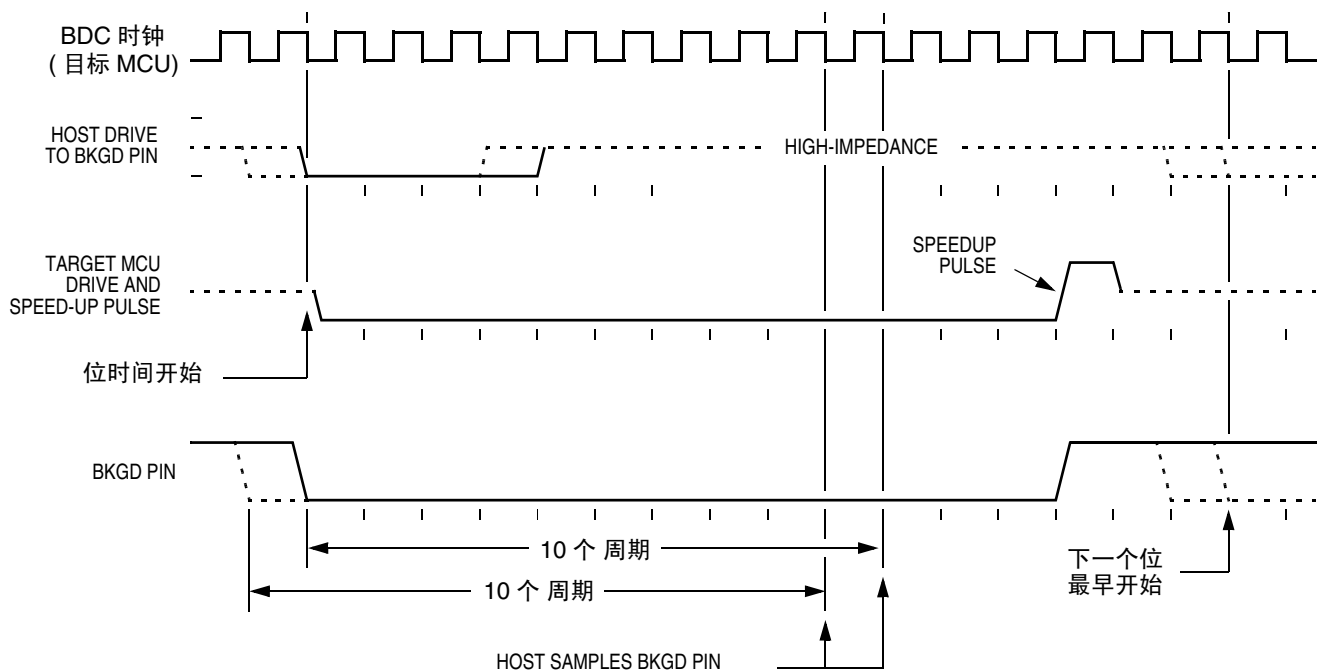


图 17-4. BDM 目标 - 到 - 主机串行位时序 (逻辑 0)

17.2.3 BDC 命令

BDC 命令以串行形式从主机发送到目标 HCS08 MCU 的 BKGD 管脚。所有命令和数据都采用定制 BDC 通信协议以 MSB- 先发的形式发送。激活背景调试模式命令要求目标 MCU 当前处于激活背景调试模式，而非侵入式命令可以随时发出，无论目标 MCU 是处于激活背景调试模式还是运行用户应用程序。

表 17-1 显示了所有 HCS08 BDC 命令，并简要描述了它们的编码结构，以及每个命令的含义。

编码结构术语

表 17-1 中所用的术语描述了 BDC 命令的编码结构。

命令在主机 - 到 - 目标方向上始于一个 8- 位十六进制命令代码 (MSB 先发)

- / = 将命令的各部分分开
- d = 延迟 16 个目标 BDC 时钟周期
- AAAA = 主机 - 到 - 目标方向上的一个 16- 位地址
- RD = 目标 - 到 - 主机方向上的 8- 位读数据
- WD = 主机 - 到 - 目标方向上的 8- 位写数据
- RD16 = 目标 - 到 - 主机方向上的 16 位读数据
- WD16 = 主机 - 到 - 目标方向上的 16 位写数据
- SS = 目标 - 到 - 主机方向 (STATUS) 上的 BDCSCR 内容
- CC = 主机 - 到 - 目标方向 (CONTROL) 方向上的 8 位写数据
- RBKP = 目标 - 到 - 主机方向 (从 BDCBKPT 断点寄存器) 上的 16 位读数据
- WBKP = 主机 - 到 - 目标方向 (至 BDCBKPT 断点寄存器) 16 位写数据

表 B-5. 模式、边和电平选择

CPWMS	MSnB:MSnA	ELSnB:ELSnA	模式	配置
X	XX	00	不用于 TPM 通道的管脚；作为 TPM 的外部时钟使用或恢复为通用输入 / 输出	
0	00	01	输入捕获	仅在上升边捕获
		10		仅在下降边捕获
		11		在上升或下降边捕获
	01	00	输出对比	仅对比软件
		01		切换对比输出
		10		清除对比输出
		11		设置对比输出
	1X	10	边缘对齐 PWM	高保真脉冲（清除对比输出）
		X1		低保真脉冲（设置对比输出）
1	XX	10	中央对齐 PWM	高保真脉冲（清除向上对比输出）
		X1		低保真脉冲（设置向上对比输出）

如果相应的端口管脚在改变成输入捕获模式前至少在两个总线时钟周期内不稳定，系统可能会提供边沿触发器的意外指示。一般，在改变通道配置位之后和启动通道中断之前，程序会清除状态标记，或使用状态标记避免任何意外行为。。

B.5.5 TPM 通道值寄存器 (TPMxCnVH:TPMxCnVL)

这些读 / 写寄存器包含输入捕获功能捕获的 TPM 计数器值，或输出对比或 PWM 功能的输出对比值。通过复位可清除通道值寄存器。

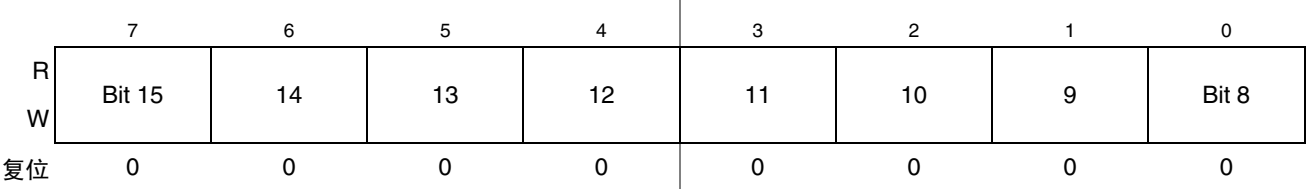


图 B-8. 定时器通道值寄存器高 (TPMxCnVH)

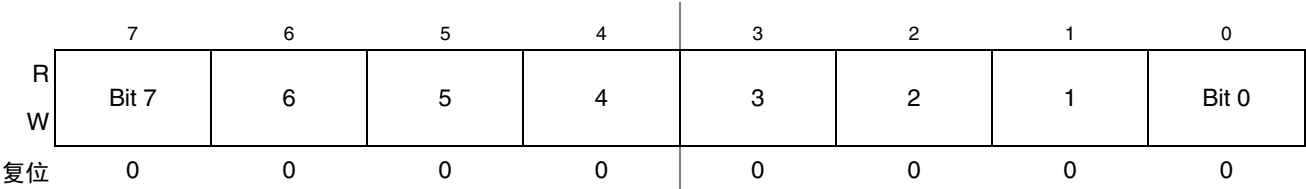


图 B-9. 定时器通道值寄存器低 (TPMxCnVL)