



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	Active
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz48f2clh

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 ¹
IIC	关闭	待机
MCG	关闭	可选择开启 ²
MSCAN	关闭	待机
RTC	可选择开启 ³	可选择开启 ³
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 ⁴
XOSC	关闭	可选择开启 ⁵
I/O 管脚	状态被保持	状态被保持
BDM	关闭 ⁶	可选择开启
LVD/LVW	关闭 ⁷	可选择开启

¹ 要求启用 DC 时钟和 LVD，否则为待机。

² MCGC1 中设置 IRCLKEN 和 IREFSTEN，否则为待机。

³ 要求启用 RTC，否则为待机。

⁴ 要求启用 LVD 或 BDC。

⁵ 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。

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

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

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
0x1881	CANCTL1	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
0x1882	CANBTR0	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
0x1883	CANBTR1	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
0x1884	CANRFLG	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
0x1885	CANRIER	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
0x1886	CANTFLG	0	0	0	0	0	TXE2	TXE1	TXE0
0x1887	CANTIER	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
0x1888	CANTARQ	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
0x1889	CANTAACK	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
0x188A	CANTBSEL	0	0	0	0	0	TX2	TX1	TX0
0x188B	CANIDAC	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
0x188C	预留	0	0	0	0	0	0	0	0
0x188D	CANMISC	0	0	0	0	0	0	0	BOHOLD
0x188E	CANRXERR	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
0x188F	CANTXERR	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
0x1890 – 0x1893	CANIDAR0 – CANIDAR3	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x1894 – 0x1897	CANIDMR0 – CANIDMR3	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x1898 – 0x189B	CANIDAR4 – CANIDAR7	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x189C – 0x189F	CANIDMR4 – CANIDMR7	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x18BE	CANTTSRH	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
0x18BF	CANTTSRL	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
0x18C0 – 0x18FF	预留	—	—	—	—	—	—	—	—

¹ 该位被预留。用户必须在该位上写一个 1。否则可能会导致出现异常。

6.5.2.1 B 端口数据寄存器 (PTBD)

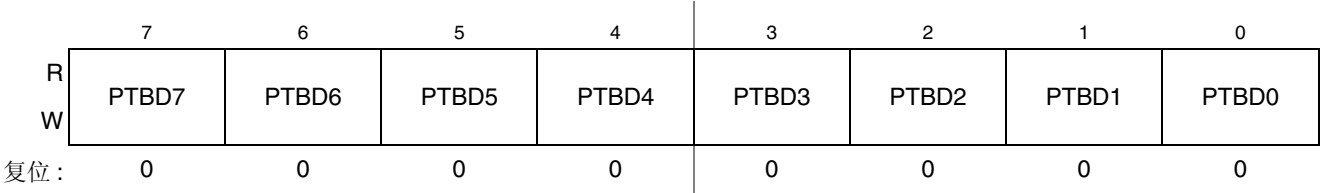


图 6-11. B 端口数据寄存器 (PTBD)

表 6-9. PTBD 寄存器字段描述

字段	描述
7:0 PTBD[7:0]	B 端口数据寄存器位 — 对于配置为输入的 B 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 B 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 B 端口管脚，逻辑电平被输出到相应的 MCU 管脚。 复位强制 PTBD 都为 0，但是这些 0 未被输出到相应的管脚，因为复位还会将所有端口管脚配置为上拉 / 下拉被禁止的高阻抗输入。

6.5.2.2 B 端口数据方向寄存器 (PTBDD)

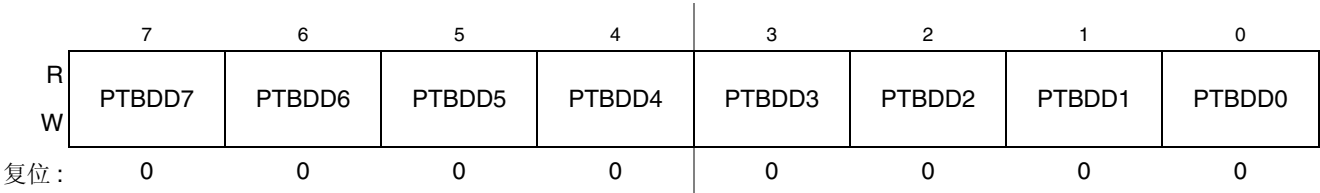


图 6-12. B 端口数据方向寄存器 (PTBDD)

表 6-10. PTBDD 寄存器字段描述

字段	描述
7:0 PTBDD[7:0]	B 端口位的数据方向 — 这些读 / 写位控制着 B 端口管脚的方向以及为 PTBD 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，PTBD 读数返回 PTBDn 内容。

6.5.3.1 C 端口数据寄存器 (PTCD)

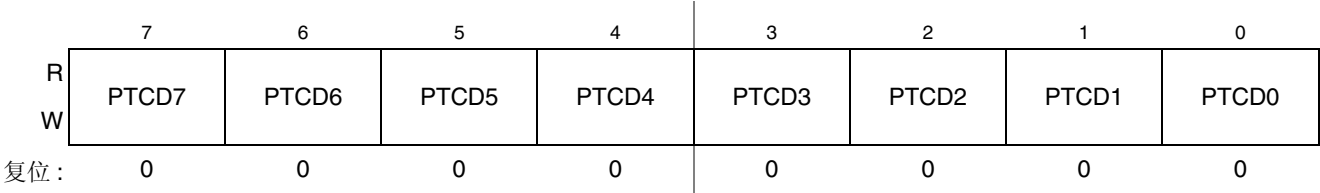


图 6-19. C 端口数据寄存器 (PTCD)

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)

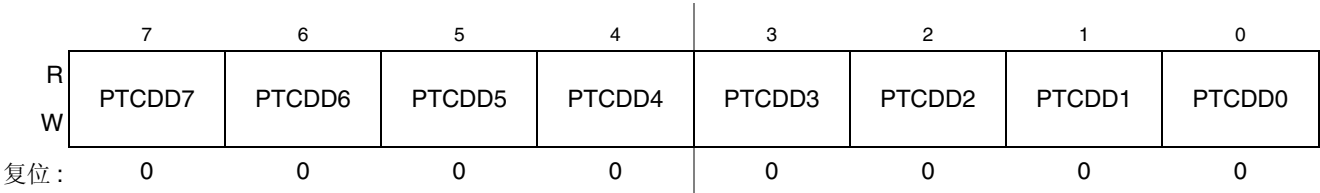


图 6-20. C 端口数据方向寄存器 (PTCDD)

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 PTCDDn 内容。

6.5.6.5 F 端口驱动强度选择寄存器 (PTFDS)

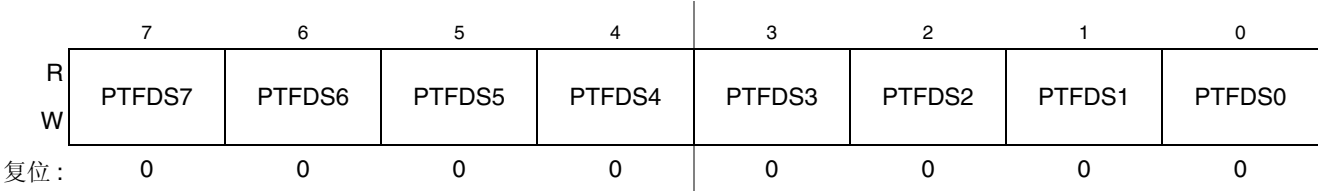


图 6-41. F 端口寄存器驱动强度选择 (PTFDS)

表 6-39. PTFDS 寄存器字段描述

字段	描述
7:0 PTFDS[7:0]	F 端口位的输出驱动强度选择 — 这些控制位为相关的 PTF 管脚选择低输出驱动和高输出驱动。对于配置为输入的 F 端口管脚，这些位不会产生任何影响。 0 F 端口位 - 选择的低输出驱动强度。 1 F 端口位 - 选择的高输出驱动强度。

6.5.7 G 端口寄存器

G 端口由下列寄存器控制。

6.5.7.1 G 端口数据寄存器 (PTGD)

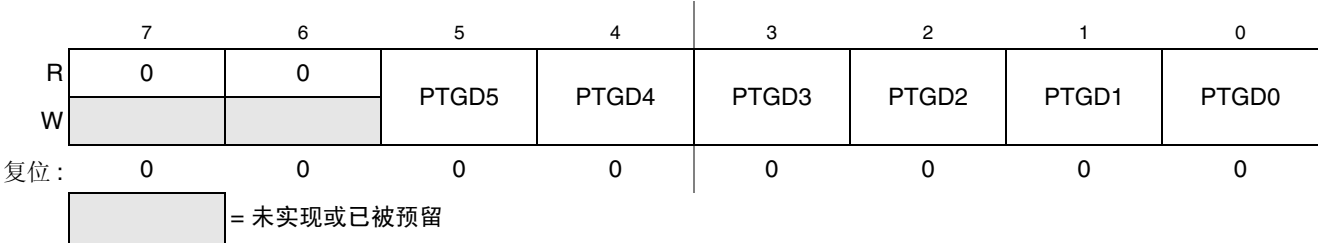


图 6-42. G 端口数据寄存器 (PTGD)

表 6-40. PTGD 寄存器字段描述

字段	描述
5:0 PTGD[5:0]	G 端口数据寄存器位 — 对于配置为输入的 G 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 G 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 G 端口管脚，逻辑电平被输出到相应的 MCU 管脚。 复位强制 PTGD 都为 0，但是这些 0 未被输出到相应的管脚，因为复位还会将所有端口管脚配置为上拉 / 下拉被禁止的高阻抗输入。

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。

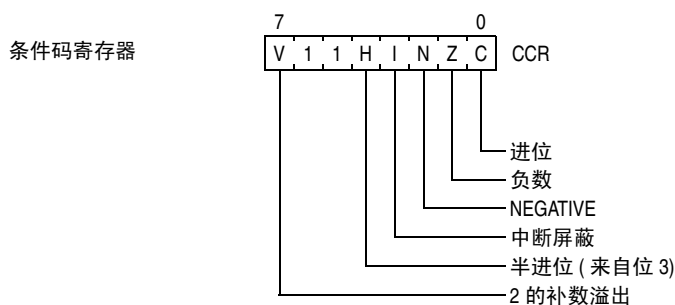


图 7-2. 条件码寄存器

8.5.1.1 FLL Engaged Internal (FEI)

FLL Engaged Internal (FEI) 是默认运行模式并且当满足下列条件时就进入该模式：

- CLKS 位写入 00
- IREFS 位写入 1
- PLLS 位写入 0
- RDIV 位写入 000。因为内部参考时钟频率在修正后应介于 31.25 kHz- 39.0625 kHz 之间，所以不需要进一步分频。

在 FLL Engaged Internal 模式中，MCGOUT 时钟源自 FLL 时钟，由内部参考时钟控制。FLL 时钟频率是由 RDIV 位选择的参考频率的 1024 倍。MCGLCLK 来自 FLL，PLL 被禁止并处于低功率状态。

8.5.1.2 FLL Engaged External (FEE)

当满足下列条件时就进入 FLL engaged External (FEE) 模式：

- CLKS 位写入 00
- IREFS 位写入 0
- PLLS 位写入 0
- RDIV 位写入介于 31.25 kHz- 39.0625 kHz 频率范围内的分频参考时钟。

在 FLL Engaged External 模式中，MCGOUT 时钟来自 FLL 时钟，由外部参考时钟控制。使能的外部参考时钟可以是外部晶体 / 谐振器，也可以是另外一个外部时钟源。FLL 时钟频率是由 RDIV 位选择的参考频率的 1024 倍。MCGLCLK 来自 FLL，PLL 被禁止并处于低功率状态。

8.5.1.3 FLL Bypassed Internal (FBI)

在 FLL Bypassed Internal (FBI) 模式中，MCGOUT 时钟来自内部参考时钟，FLL 处于运行状态但其输出时钟未使用。该模式对允许 FLL 获得目标频率非常有用，同时 MCGOUT 时钟由内部参考时钟驱动。

当满足以下条件时就进入 FLL Bypassed Internal 模式：

- CLKS 位写入 01
- IREFS 位写入 1
- PLLS 位写入 0
- RDIV 位写入 000。由于内部参考时钟频率在修正后应已经介于 31.25 kHz- 39.0625 kHz 之间，所以不需要进一步的分频。
- LP 位写入 0

在 FLL Bypassed Internal 模式中，MCGOUT 时钟源自内部参考时钟。FLL 时钟由内部参考时钟控制。FLL 时钟频率是由 RDIV 位选择的参考频率的 1024 倍。MCGLCLK 来自 FLL，PLL 被禁止并处于低功率状态。

11.2.1 特性

IIC 包括以下明显的特性：

- 兼容 IIC 总线标准
- 多主操作
- 可以软件编程为 64 种不同串行时钟频率的任意一种
- 软件可选应答位
- 中断驱动的逐字节数据传输
- 仲裁丢失中断，可以自动地从主模式切换到从模式
- 主叫地址识别中断
- 开始和停止信号生成 / 检测
- 重复生成启动信号
- 应答位生成 / 检测
- 总线忙检测
- 通用呼叫识别
- 10 位地址扩展

11.2.2 运行模式

不同 MCU 模式中的 IIC 的简要描述如下：

- 运行模式 — 这是基本运行模式。要降低这种模式下的功耗，只需禁止模块。
- 等待模式 — 当 MCU 处于等待模式时，模块继续运行并能够提供唤醒中断。
- 停止模式 — IIC 在 STOP3 模式中是不停止的，以降低功耗。停止指令不会影响 IIC 寄存器状态。STOP2 复位寄存器内容。

表 11-2. IICF 字段描述

字段	描述
7–6 MULT	IIC 倍频因子。MULT 位定义倍频因子 mul。该倍频因子与 SCL 分频器一起，共同生成 IIC 波特率。MULT 位定义的倍频因子 mul 如下： 00 mul = 01 01 mul = 02 10 mul = 04 11 保留
5–0 ICR	IIC 时钟率。IIC 位用来预分频总线时钟，以便进行比特率选择。这些位和 MULT 位确定 IIC 波特率、SDA 保持时间、SCL 开始保持时间和 SCL 停止保持时间。表 11-4 为 ICR 的相应值提供了 SCL 分频器和保持值。 由倍频因子 mul 和 SCL 分频器生成 IIC 波特率。 $\text{IIC baud rate} = \frac{\text{bus speed (Hz)}}{\text{mul} \times \text{SCLdivider}} \quad \text{等式 11-1}$ SDA 保持时间是从 SCL（IIC 时钟）的下降边沿到 SDA（IIC 数据）变化的延迟。 $\text{SDA 保持时间} = \text{总线周期 (s)} \times \text{mul} \times \text{SDA 保持值} \quad \text{等式 11-2}$ SCL 开始保持时间是从 SDA（IIC 数据）的下降边沿（这时 SCL 处于高一开始状态）到 SCL（IIC 时钟）下降边沿的延迟。 $\text{SCL 开始保持时间} = \text{总线周期 (s)} \times \text{mul} \times \text{SCL 开始保持值} \quad \text{等式 11-3}$ SCL 停止保持时间是从 SCL（IIC 时钟）的上升边沿到 SDA（IIC 数据）上升边沿的延迟（这时 SCL 处于高一停止状态） $\text{SCL 停止保持时间} = \text{总线周期 (s)} \times \text{mul} \times \text{SCL 停止保持值} \quad \text{等式 11-4}$

例如，如果总线速率为 8 MHz，下表显示了不同 ICR 和 MULT 选择为达到 100kbps 的 IIC 波特率的可能保持时间值。

表 11-3. 8 MHz 总线速度保持时间值

MULT	ICR	保持时间 (μ s)		
		SDA	SDA SCL 开始	SCL 停止
0x2	0x00	3.500	3.000	5.500
0x1	0x07	2.500	4.000	5.250
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.125	4.250	5.125
0x0	0x18	1.125	4.750	5.125

12.1.1 特性

MSCAN 的基本特性如下：

- 实施 CAN 协议—2.0A/B 版
 - 标准和扩展数据帧
 - 0-8 字节数据长度
 - 高达 1 Mbps¹ 的可编程比特率¹
 - 支持远程帧
- 5 个具有 FIFO 存储机制的接收缓冲器
- 3 个具有使用“本地优先”概念的内部优先顺序的发送缓冲器 t
- 灵活可掩码标识符滤波器支持 2 个全尺寸（32 位）扩展标识符滤波器或 4 个 16 位滤波器或 8 个 8 位滤波器
- 集成低通滤波器的可编程唤醒功能 r
- 可编程环回模式支持自测操作
- 可编程监听模式用于 CAN 总线监控
- 可编程总线脱离恢复功能
- 独立的信号和中断功能适用于所有 CAN 接收器和发射器错误状态（警报、错误严重状态、总线脱离）
- 可编程 MSCAN 时钟源，采用总线时钟或振荡器时钟
- 内部计时器提供给接收和发送的报文的时间标签
- 三种低功耗模式：睡眠、关机和 MSCAN 使能
- 配置寄存器的全局初始化

12.1.2 运行模式

以下运行模式是 MSCAN 的特定运行模式。详情 12.5，“功能描述”。

- 监听模式
- MSCAN 睡眠模式
- MSCAN 初始化模式
- MSCAN 关机模式
- 环回自测模式

¹ Depending on the actual bit timing and the clock jitter of the PLL.

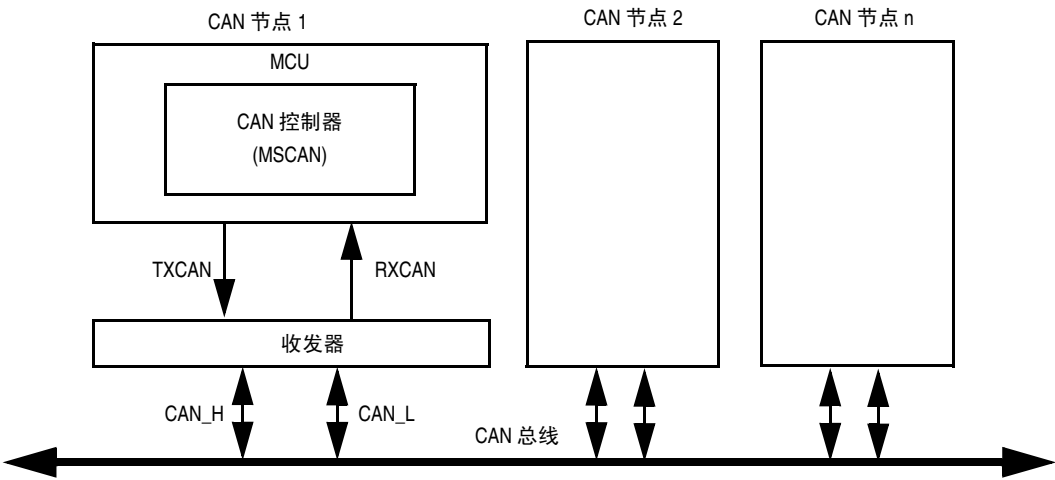


图 12-3. CAN 系统

12.3 寄存器定义

本节详细描述 MSCAN 模块中的所有寄存器和寄存器位。每个描述都包括带有相关图形编号的标准寄存器示意图。寄存器位和字段功能的详细说明在寄存器图后面，按位顺序。该模块中所有寄存器的所有位在寄存器读取过程中都与内部时钟完全同步。

12.3.1 MSCAN 控制寄存器 0 (CANCTL0)

The CANCTL0 寄存器提供了如下所述的 MSCAN 模块的各种位控制。

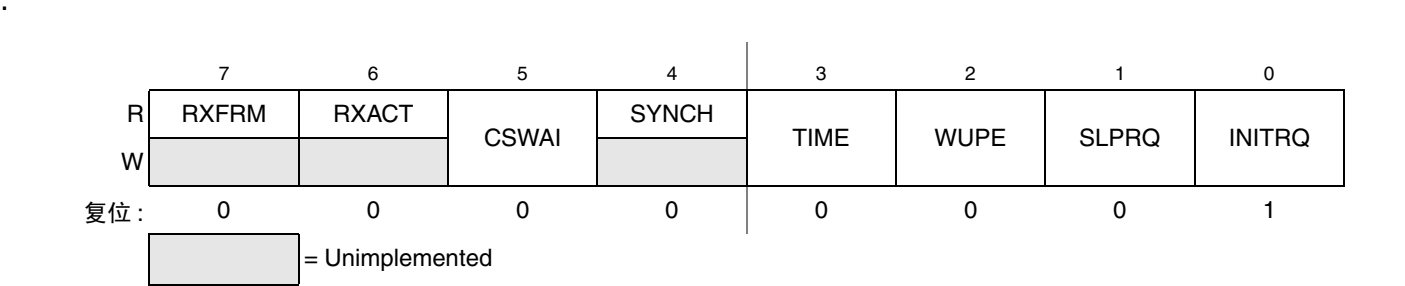


图 12-4. 控制寄存器 0 (CANCTL0)

注意

当初始化模式处于有效 (INITRQ = 1 and INITAK = 1). 时，除 WUPE、INITRQ 和 SLPRQ 外的所有 CANCTL0 寄存器位都处于复位状态。只要退出初始化模式 (INITRQ = 0, INITAK = 0)，该寄存器可以再次写入。

读取：任何时间

写入：退出初始化模式的任何时间；例外是只读 RXACT、SYNCH、RXFRM （只由该模块设置）和 INITRQ （也可以在初始化模式中写入）。。

表 12-2. 寄存器字段描述

字段	描述
1 SLPAK	睡眠模式确认— 该标记显示 MSCAN 模块是否已经进入睡眠模式 (参见 12.5.5.4, “MSCAN 睡眠模式”)。它用作 SLPRQ 睡眠模式请求的握手标志。t. 当 SLPRQ = 1、SLPAK = 1 时, 睡眠模式是有效的。根据 WUPE 设置, 如果在处于睡眠模式检测到 CAN 总线有信号, MSCAN 将清除该标志。CPU 清除 SLPRQ 位也将复位 SLPK 位。 0 正在运行 —MSCAN 正常运行 1 睡眠模式使能 — MSCAN 已经进入睡眠模式
0 INITAK	初始化模式确认— 该标志显示 MSCAN 模块是否处于初始化模式 (参见 12.5.5.5, “MSCAN I 初始化模式”)。它用作 INITRQ 初始化模式请求的握手标志。当 INITRQ = 1, INITAK = 1 时, 初始化模式被使能。当 MSCAN 处于初始化模式时, 寄存器 CANCTL1、CANBTR0、CANBTR1、CANIDAC、CANIDAR0 - CANIDAR7 和 CANIDMR0 - CANIDMR7 只能通过 CPU 写入 0 正在运行 —MSCAN 正常运行 1 初始化模式使能— MSCAN 处于初始化模式

12.3.3 MSCAN 总线计时寄存 0 (CANBTR0)

CANBTR0 寄存器配置 MSCAN 模块的各种 CAN 总线计时参数。

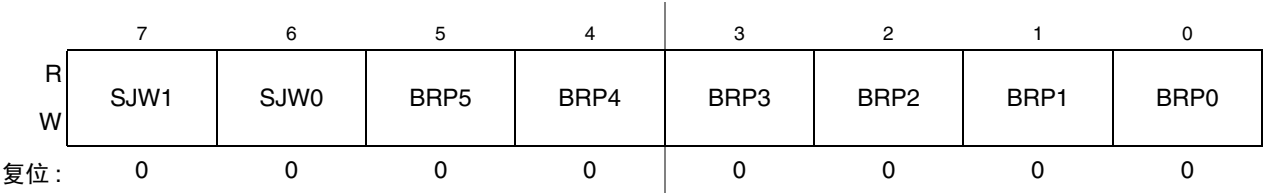


图 12-6. MSCAN 总线计时寄存器 0(CANBTR0)

读取: 任何时间

写入: 处于初始化模式 (INITRQ = 1, INITAK = 1) 的任何时间

表 12-3. CANBTR0 寄存器字段描述

字段	描述
7:6 SJW[1:0]	同步跳转宽度—同步跳转宽度决定要实现 CAN 总线上的数据传输重新同步, 一个位可以缩短或延长的时间冲量 (Tq) 的最大值 (参见表 12-4).
5:0 BRP[5:0]	波特率预分频器—该位确定用来构建位计时的时间冲量 (Tq) 时钟 (参见表 12-5).

表 12-25. IDR0 寄存器字段描述 — 扩展

字段	描述
7:0 ID[28:21]	扩展格式标识符— 该标识符由 29 个扩展格式位（ID[28:0]）组成。ID28 是最高的位，仲裁流程期间最先在 CAN 总线上发送。标识符的优先级定义为处于最高位的最小二进制数。

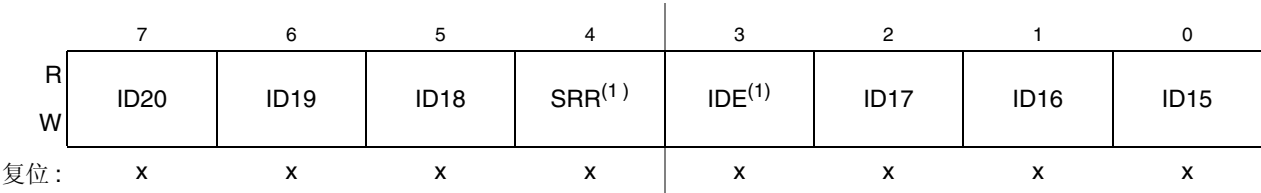


图 12-26. I 标识符寄存器 3（IDR3）— 扩展标识符映射

¹ 1 SRR 和 IDE 都为 1。

表 12-26. 标识符寄存器 1（IDR1）— 扩展标识符映射

字段	描述
7:5 ID[20:18]	扩展格式标识符—该标识符由 29 个扩展格式位（ID[28:0]）组成。ID28 是最高的位，仲裁流程期间最先在 CAN 总线上发送。标识符的优先级定义为处于最高位的最小二进制数。
4 SRR	替代远程请求— 该固定隐性位仅用于扩展格式。它必须由用户为传输缓冲器置位为 1，，并为接收缓冲器在 CAN 总线上置位为接收。
3 IDE	扩展—该标志显示扩展或标准标识符格式是否应用于该缓冲器。在接收缓冲器中，该标志设置为已接收，并向 CPU 显示如何处理缓冲器标识符寄存器。在发送缓冲器中，该标志向 MSCAN 显示将发送的标识符类型。 0 标准格式 （11 位） 1 扩展格式 （29 位） 2:0
2:0 ID[17:15]	扩展格式标识符—该标识符由 29 个扩展格式位（ID[28:0]）组成。ID28 是最高的位，仲裁流程期间最先在 CAN 总线上发送。标识符的优先级定义为处于最高位的最小二进制数。

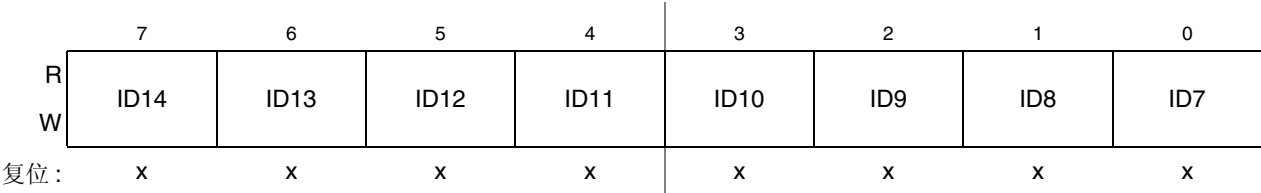


图 12-27. 标识符寄存器 2（IDR2）— 扩展标识符映射

表 12-27. IDR2 寄存器字段说明—扩展

字段	描述
7:0 ID[14:7]	扩展格式标识符—该标识符由 29 个扩展格式位（ID[28:0]）组成。ID28 是最高的位，仲裁流程期间最先在 CAN 总线上发送。标识符的优先级定义为处于最高位的最小二进制数。

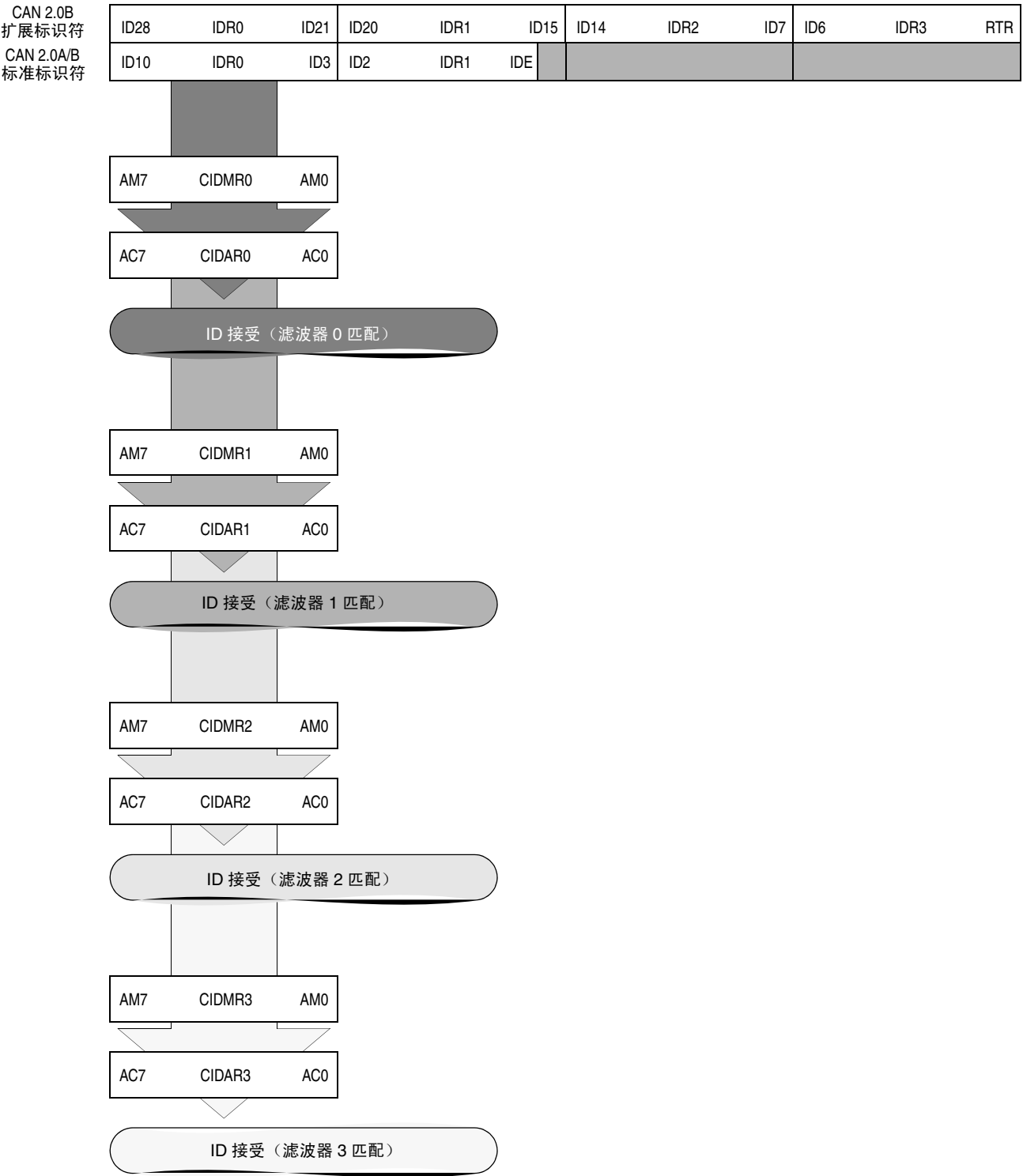


图 12-41. 8 位可屏蔽标识符接收滤波器

MSCAN 滤波器使用三组寄存器来提供滤波器配置。首先，CANIDAC 寄存器决定页配置中的滤波器大小和滤波器数量；其次，寄存器 CANIDMR0/1/2/3 通过把 ‘0’ 放在滤波器寄存器中的

可编程预分频器从 CANCLK 生成时间冲量 (Tq) 时钟。时间冲量是 MSCAN 所处理时间的原子单位。

等式 12-2

$$f_{Tq} = \frac{f_{CANCLK}}{(\text{Prescaler value})}$$

位时间再分成三段，如 Bosch CAN 规范所述（见图 12-43）:

- SYNC_SEG: 该段有一个长度固定的时间冲量，信号边沿预计出现在本段。
- 时段 1: 本段包括 CAN 标准的 PROP_SEG 和 PHASE_SEG1。通过设置参数 TSEG1，使之包含 4-16 个时间冲量，可以对其进行编程。
- 时段 2: 本段表示 CAN 标准的 PHASE_SEG2。通过设置 TSEG2 参数，使之具有 2-8 个时间冲量长，可以对其进行编程。

等式 12-3

$$\text{Bit Rate} = \frac{f_{Tq}}{(\text{number of Time Quanta})}$$

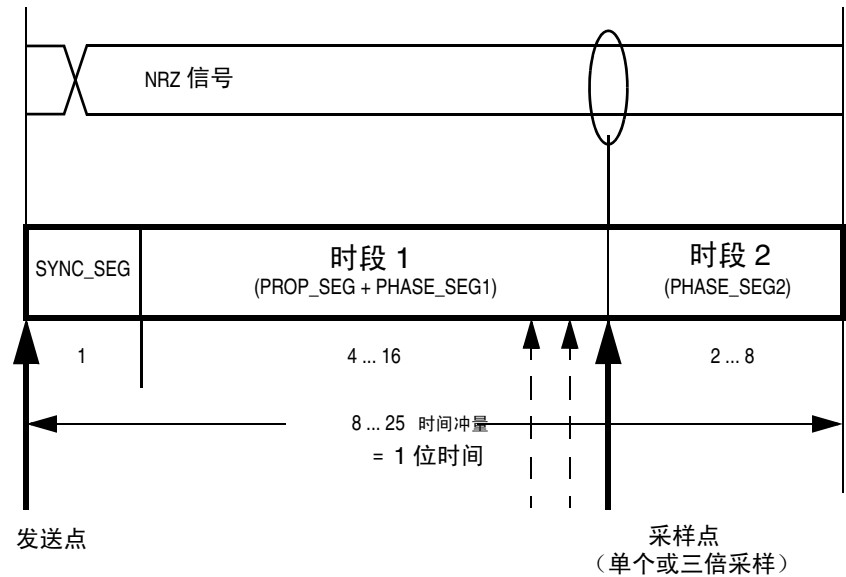


图 12-43. 位时间内的段

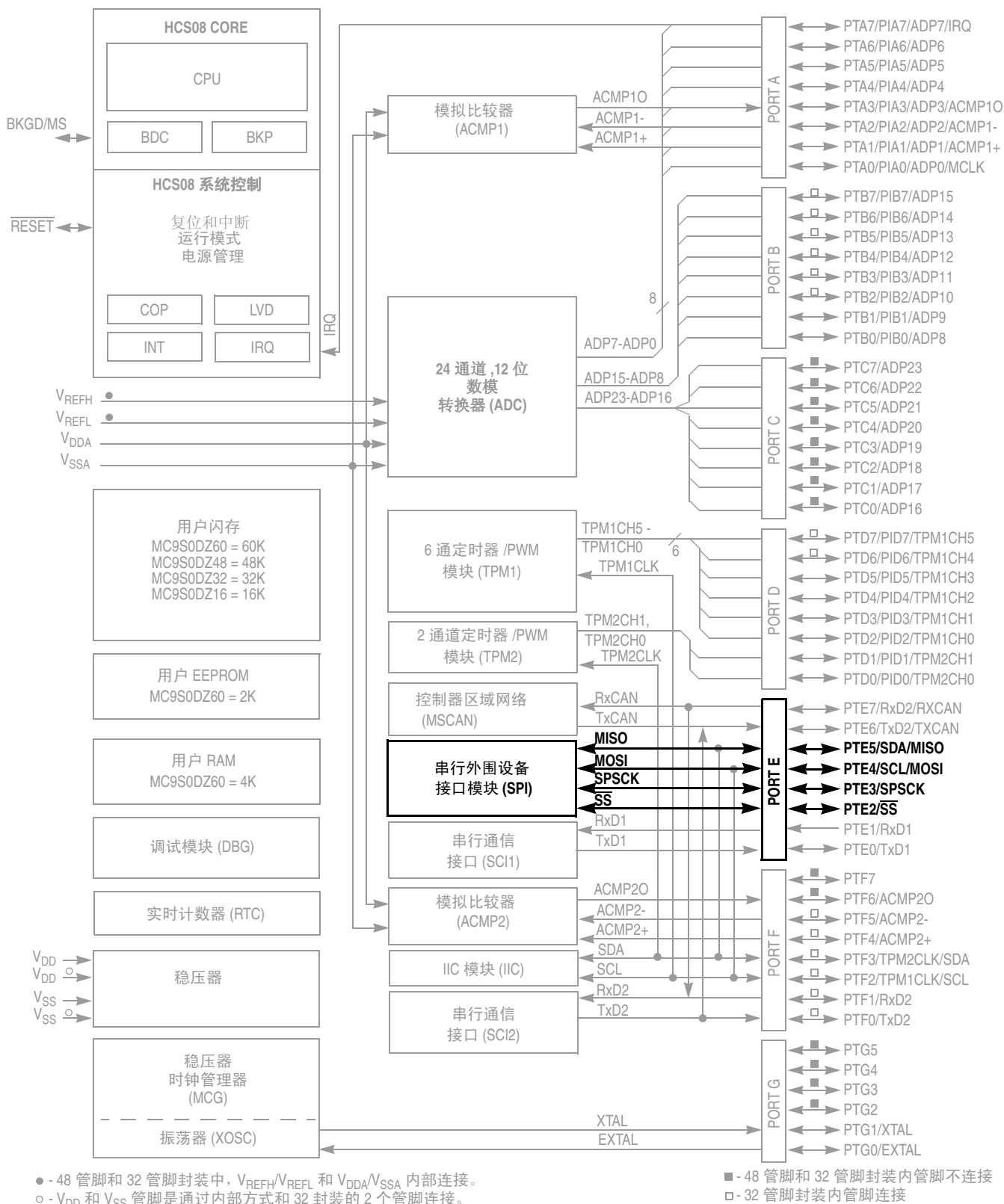


图 13-1. 强调 ADC 模块和管脚的 MC9S08DZ60 结构图

第 17 章 开发支持

17.1 介绍

HCS08 中的开发支持系统包括背景调试控制器 (BDC) 和片上调试模块 (DBG)。BDC 提供单线调试接口，与目标连接，通过这个接口可以方便地进行片上闪存和其它非易失性存储器的编程。BDC 也是开发用的主要调试接口，允许以非侵入式方式存取存储器数据和传统调试功能，如 CPU 寄存器修改、断点和单指令跟踪命令等。

在 HCS08 产品系列中，外部管脚不包括地址和数据总线信号（即使在测试模式也不包括）。调试的执行是通过单线背景调试接口向目标 MCU 传输命令来实现的。调试模块提供了一种有选择性地触发和捕获总线信息的方式，这样外部开发系统可以对 MCU 内发生的事件按周期进行重现，而不需要从外部存取 MCU 的地址和数据信号。

17.1.1 强制激活背景调试

强制激活背景调试模式的方式取决于具体的 HCS08 衍生产品。对，你可以在上电复位后强制激活背景调试，即当器件退出复位时保持 BKGD 引脚为低。你还可以通过将 SBDPR 寄存器的 BDFR 位置 1 的串行背景调试命令之后拉低 BKGD 管脚从而强制激活背景调试中的 BKGD 管脚后立即使低。如果没有调试盒连接到 BKGD 管脚，MCU 将总是复位到正常操作模式。

在输入捕获模式中，读取任何一个字节（无论是 TPMxCnVH 还是 TPMxCnVL）都会使两个字节的內容被锁入到缓冲器中。这些內容一直锁定在这个缓冲器中，直到另一个字节被读取。当 TPMxCnSC 寄存器被写入时，锁存机制可复位（变为未锁存状态）。

在输出对比或 PWM 模式中，写入任何一个字节（无论是 TPMxCnVH 还是 TPMxCnVL）都会使该值被锁入到缓冲器中。两个字节都被写入后，它们会作为连贯的 16 位值传输到定时器通道值寄存器中。这一锁存机制可以通过写入 TPMxCnSC 寄存器来人工进行复位。

这种锁定机制允许以任何顺序进行连贯的 16 位写入，这对各种编译器实施方案都很友好。

B.6 功能介绍

所有 TPM 功能都与允许灵活选择时钟源和预分频器的 16 位主计数器相关。此外，16 位模数寄存器还与 TPM 中的 16 位主计数器相关。每个 TPM 通道可与 MCU 管脚及可屏蔽的中断功能相关。

TPM 具有中央对齐的功能（由 TPMxSC 中 CPWMS 控制位控制）。当 CPWMS 被设置为 1 时，定时器计数器 TPMxCNT 改变为向上 / 向下计数器并且相关 TPM 中的所有通道都作为中央对齐的 PWM 通道。当 CPWMS=0 时，每个通道可独立配置，以便以输入捕获、输出对比或缓冲的边缘对齐 PWM 模式运行。

后面各小节介绍 16 位主计数器和计数器的每种运行模式（输入捕获、输出对比、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操作模式，这些主题将在相关模式的章节中介绍。

B.6.1 计数器

所有定时器功能都基于 16 位主计数器（TPMxCNTH:TPMxCNTL）。本小节讨论时钟源选择、向上计数和向下计数、计数结束溢出和手动计数器复位。

在任何 MCU 复位后，CLKSB:CLKSA = 0:0，所以没有选择时钟源，并且 TPM 是不活动的。正常情况下，CLKSB:CLKSA 将设置为 0:1，使总线时钟驱动定时器计数器。TPM 的时钟源可以选为关闭、总线时钟（BUSCLK）、固定系统时钟（XCLK）或外部输入。外部时钟方法的最大允许频率为总线速率的 1/4。参见 B.5.1，“定时器状态和控制寄存器 (TPMxSC)”及表 B-2 来了解有关时钟源的更多信息。

当微控制器处于活动后台模式时，TPM 会临时挂起所有计数，直到微控制器返回到正常用户操作模式。在停止模式下，所有 TPM 时钟被停止；因此在时钟恢复前，TPM 一直被有效地关闭。在等待模式期间，TPM 继续正常运行。

16 位主计数器有两种计数模式。选择中央对齐 PWM 时（CPWMS = 1），计数器以向上 / 向下计数模式运行。否则，计数器作为简单的向上计数器运行。用作向上计数器时，16 位主计数器从 0x0000 开始计数，直到终端计数，然后重新从 0x0000 开始。最大计数为 0xFFFF 或 TPMxMODH:TPMxMODL 中的模数值。

当规定了中央对齐 PWM 操作时，计数器从 0x0000 向上计数，直到达到终端计数，然后向下计数到 0x0000，再从这里向上计数。0x0000 和终端计数值（TPMxMODH:TPMxMODL 中的值）为正常长度计数（一个定时器时钟周期长度）。