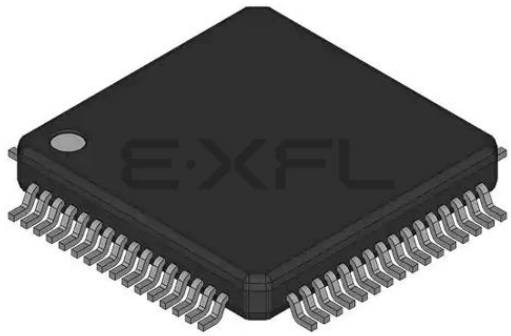

Welcome to E-XFL.COM

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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
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 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dz32aclh

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。

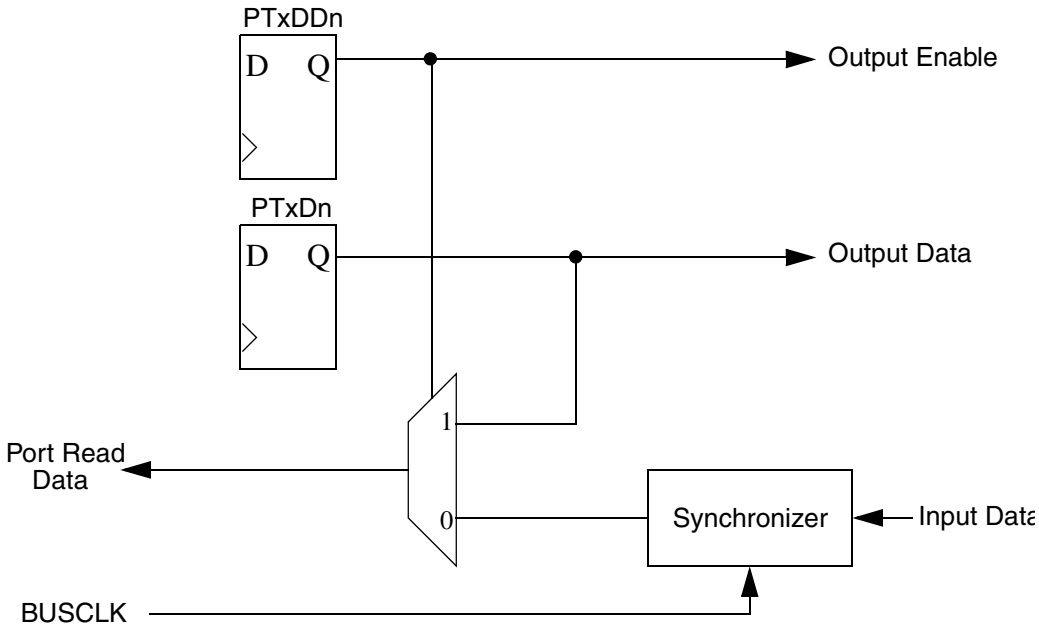

图 6-1. 并行输入 / 输出示意图

6.2 上拉、斜率和驱动强度

与并行输入 / 输出端口相关的是位于高页寄存器空间的一套寄存器，它们的操作独立于输入 / 输出寄存器。这些寄存器用来控制管脚的上拉、斜率和驱动强度。

在上拉寄存器 （PTxPEn）中设置对应的位可以为所对应的端口管脚使能内部上拉器件。如果并行输入 / 输出控制逻辑或任何外围设备功能将管脚配置为输出，上拉器件就会被禁止，这与对应的上拉寄存器位的状态无关。如果管脚由模拟功能控制，上拉器件同样会被禁止。

在斜率控制寄存器 （PTxSEn）中设置对应的位可以为所对应的端口管脚使能斜率控制。该功能启动时，转换控制限制输出的转换速度，减少 EMC 发射。斜率控制对配置为输入的管脚没有任何影响。

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规定的值，以保证正确的操作。

在驱动强度选择寄存器 （PTxDSn）中设置对应的位可以选择一个具有高输出驱动强度的输出管脚。选定好高驱动后，管脚就能够送出和吸收更大的电流。即使可以将每个 I/O 管脚选择为高驱动，用户也必须保证不超出 MCU 的总送出和吸收电流限制。驱动强度选择旨在影响 I/O 管脚的 DC 行为，然而， AC 行为也会受到影响。高驱动允许管脚以与低驱动使能管脚在更小负载中一样的交换速度，驱动更大的负载。正因为如此， EMC 放射可能会由于使能管脚作为高驱动而受到影响。

6.5.1.5 A 端口驱动强度选择寄存器 (PTADS)

	7	6	5	4	3	2	1	0
R W	PTADS7	PTADS6	PTADS5	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
复位 :	0	0	0	0	0	0	0	0

图 6-7. A 端口寄存器的驱动强度选择 (PTADS)

表 6-5. PTADS 寄存器字段描述

字段	描述
7:0 PTADS[7:0]	**A 端口位的输出驱动强度选择** — 这些控制位为相关 PTA 管脚选择低输出驱动和高输出驱动。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 - 选择的低输出驱动强度。 1 A 端口位 - 选择的高输出驱动强度。

6.5.1.6 A 端口中断状态和控制寄存器 (PTASC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAIF	0	PTAIE	PTAMOD
W						PTAACK		
复位 :	0	0	0	0	0	0	0	0

= 未实现或已被预留

图 6-8. A 端口中断状态和控制寄存器 (PTASC)

表 6-6. PTASC 寄存器字段描述

字段	描述
3 PTAIF	**A 端口中断标志** — PTAIF 显示是否检测到 A 端口中断。写入对 PTAIF 没有任何影响。 0 未检测到 A 端口中断。 1 检测到 A 端口中断。
2 PTAACK	**A 端口中断确认** — 向 PTAACK 写入 1 是标记清除机制的一部分。 PTAACK 的读数总为 0。
1 PTAIE	**A 端口中断使能** — PTAIE 决定是否请求 A 端口中断。 0 A 端口中断请求禁止。 1 A 端口中断请求使能。
0 PTAMOD	**A 端口检测模式** — PTAMOD （同 PTAES 位一起）控制着 A 端口中断管脚的检测模式。 0 A 端口管脚只检测边沿。 1 A 端口管脚同时检测边沿和电平。

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。

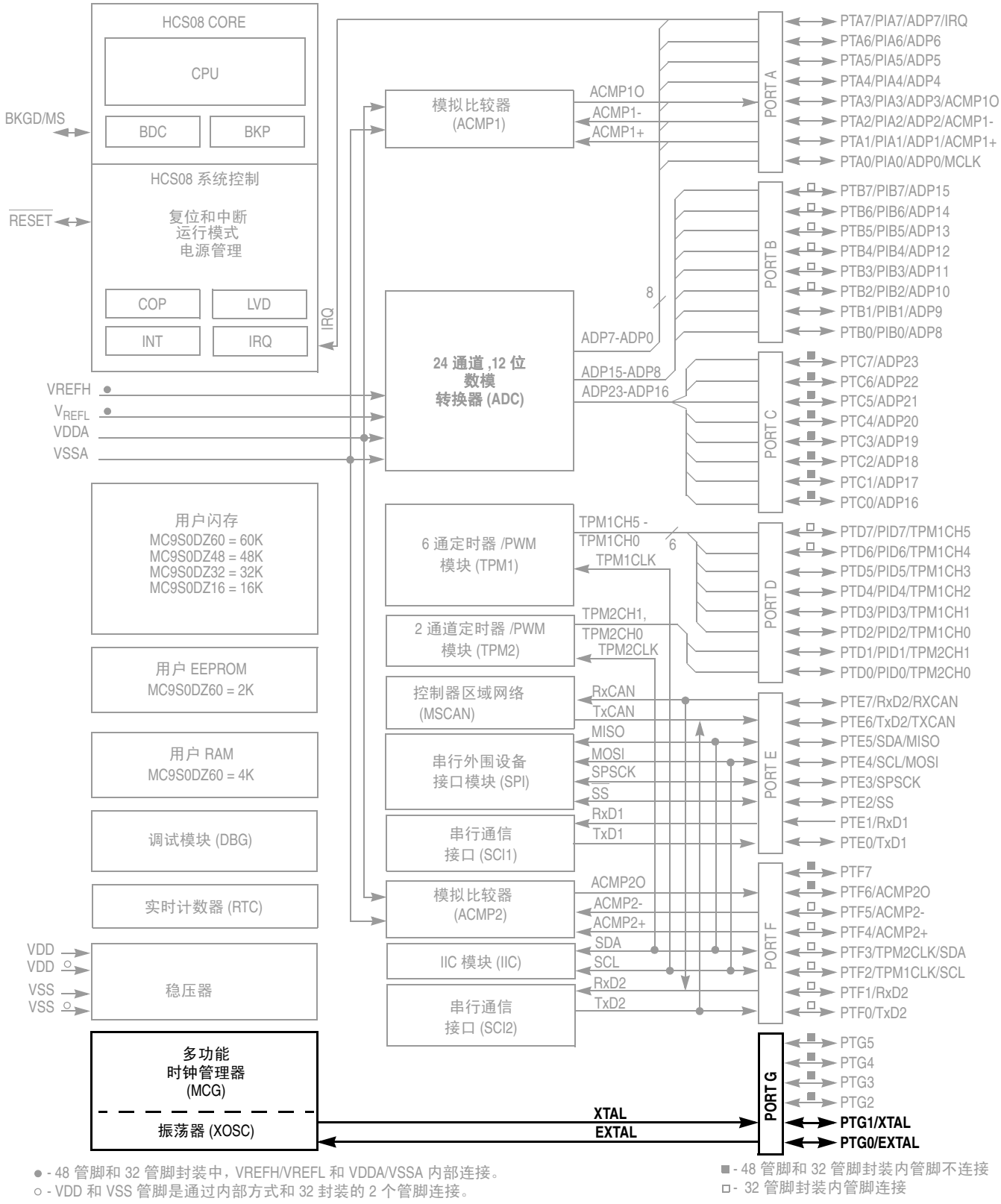

图 8-1. MC9S08DZ60 结构图

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。

第 11 章
IIC 模块 (S08IICV2)

11.1 介绍

IIC 模块 (IIC) 提供了不同器件间的通信方法。接口可以在最大的总线负载和时序下，支持最高 100kbps 的传输速率。器件可以在较低总线负载下、以更高的波特率 （最高时钟 /20）运行。最大通信长度和可以连接的器件数量受 400 pF 的最高总线电容限制。

MC9S08DZ60 系列的所有 MCU 都具有 IIC 功能，如以下结构图所示。

注意

在使用 IIC 模块时，禁止为 IIC 管脚使用驱动强度 （DSE=0），以实现正确操作。

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。

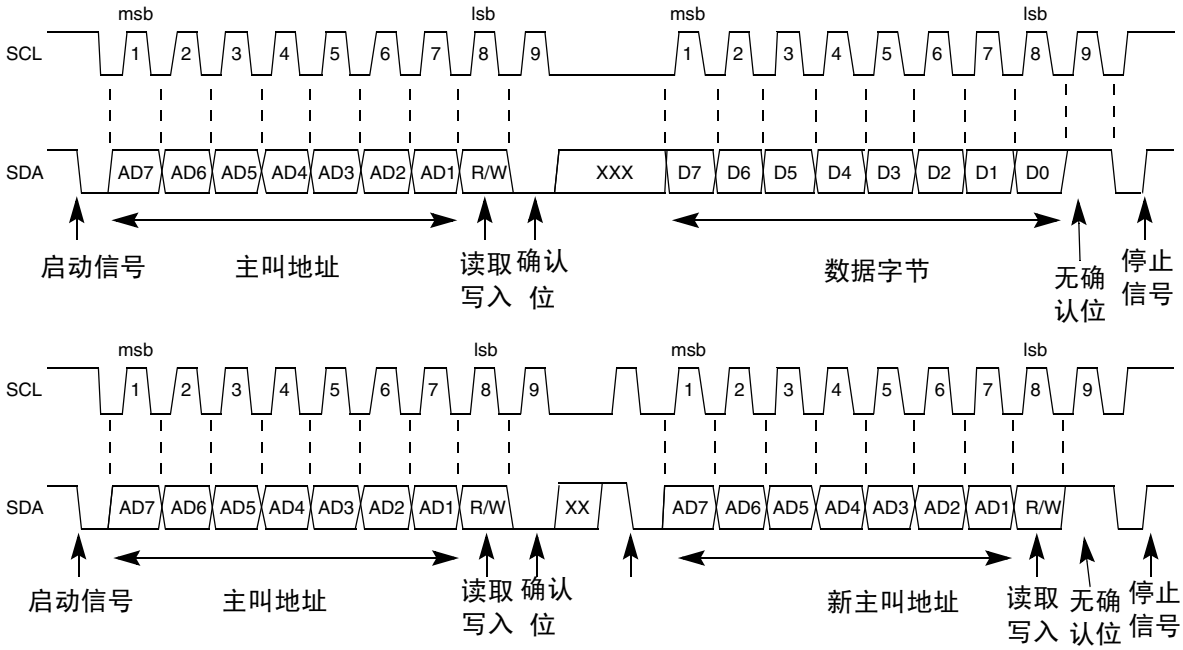

图 11-9. IIC 总线传输信号

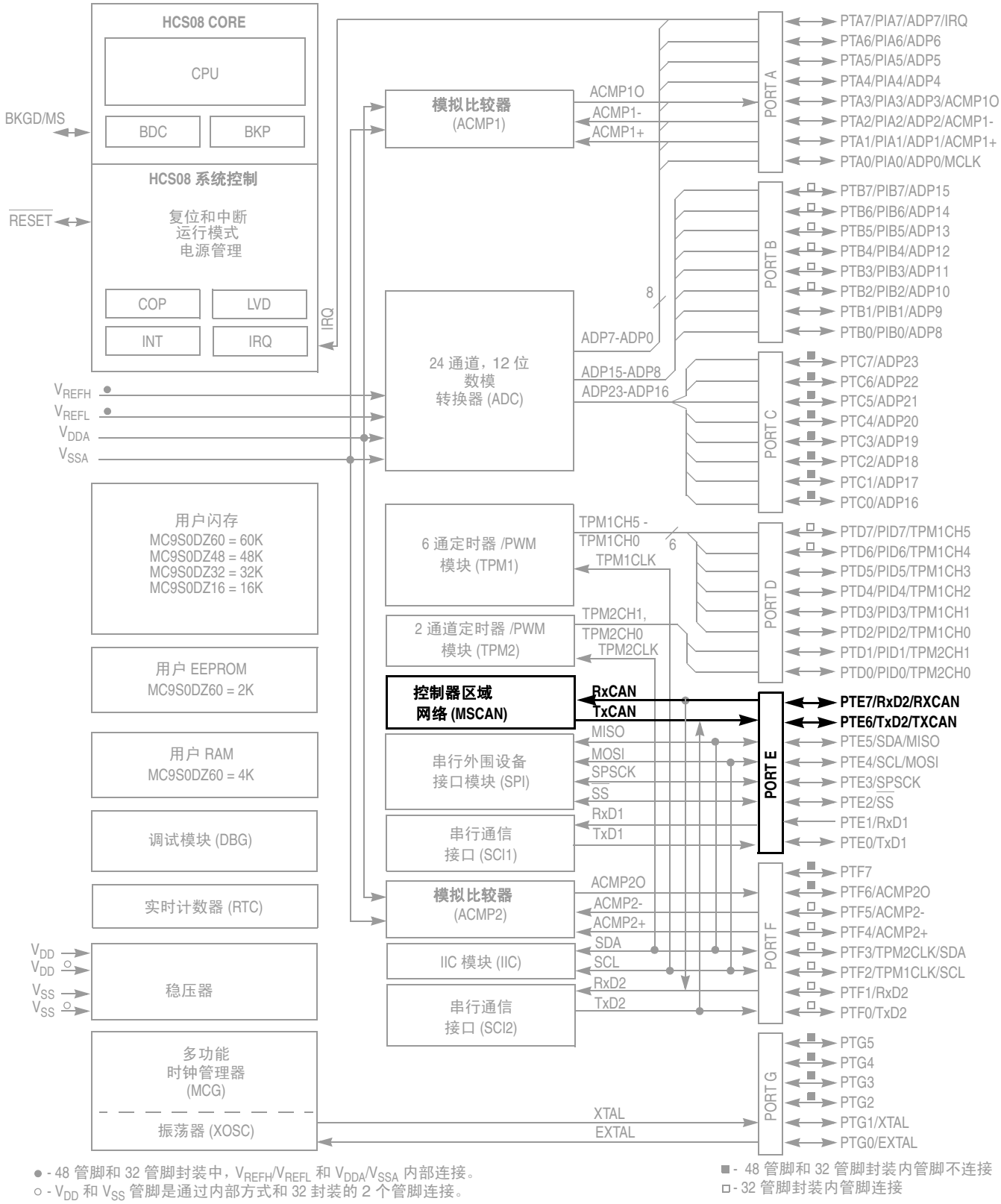

图 12-1. MC9S08DZ60 结构图

12.3.2 控制寄存器 1 (CANCTL1)

CANCTL1 寄存器如下所述提供了 MSCAN 模块的各种控制位和握手状态报文。

.

	7	6	5	4	3	2	1	0
R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
W								
复位 :	0	0	0	1	0	0	0	1

= Unimplemented

图 12-5. 控制寄存器 1(CANCTL1)

读取：任何时间
写入：当 INITRQ = 1 和 INITAK = 1 时的任何时间， CANE 例外，可以在正常情况下写入一次，以及当 MSCAN 处于初始化模式（INITRQ = 1， NITAK = 1）的特殊系统运行模式时任何时间写入。

表 12-2. 寄存器字段描述

字段	描述
7 CANE	**MSCAN** 使能 0 MSCAN 模块禁止 1 MSCAN 模块使能
6 CLKSRC	**MSCAN** 时钟源 — 该位定义 MSCAN 模块的时钟源 (仅适用于具有时钟发生模块的系统； 12.5.3.3，“时钟系统”和图 12-42.，“MSCAN 时钟机制”). 0 MSCAN 时钟源是振荡器时 1 MSCAN 时钟源是总线时钟
5 LOOPB	环回自测模式 — 当设置了该位时，MSCAN 执行可用于自测操作的内部环回。T 发送器的位流输出从内部流回接收器。 12.5.4.6，“环回自测模式” . 0 环回自测禁止 1 环回自测使能
4 LISTEN	º 监听模式 — 该位把 SCAN 配置为 CAN 总线监控器。当设置了 LISTEN 时，会收到带有匹配 ID 的所有有效 CAN，但不发出确认或错误帧 (参见 12.5.4.4，“监听模式”)。此外，错误计数器停止计数。监听模式可以支持需要 “热插拔”或 “吞吐量分析”的应用。当监听模式处于有效状态时， MSCAN 不能发送任何报文。 0 正常运行 1 监听模式使能 d
3 BORM	总线脱离恢复模式 — 该位配置 MSCAN 的总线关断恢复模式。更多报文总线脱离恢复模式— 该位配置 MSCAN 的总线关断恢复模式。更多报文 12.6.2，“总线脱离恢复”。 0 自动总线脱离恢复 （参见 Bosch CAN 2.0A/B 协议规范） 1 用户请求的总线脱离恢复
2 WUPM	唤醒模式 — 如果 CANCTL0 中的 WUPE 被使能，该位决定是否应用集成低通滤波器来防止 MSCAN 出现假唤醒 (参见 12.5.5.4，“MSCAN 睡眠模式”) 0 MSCAN 被 CAN 总线上的任意显性信号唤醒 1 MSCAN 只有在 CAN 总线上的显性脉冲长度为 Twup 时才唤醒。

表 12-10. CANRIER 寄存器字段描述

字段	描述
7 WUPIE[1]	**唤醒中断使能** 0 无中断请求从该事件中产生。 1 唤醒事件引起唤醒中断请求。
6 CSCIE	**CAN 状态变化中断使能** 0 无中断请求从该事件中产生。 1 CAN 状态变化事件引起错误中断请求。
5:4 RSTATE[1:0]	接收器状态变化使能—这些 RSTAT 使能位控制接收器状态变化而引起 CSCIF 中断的电平状态。独立于所选电平状态， RSTAT 标志继续显示实际接收器状态，且只有在没有 CSCIF 中断产生时才会更新。 00 未生成由于接收器状态变化而引起的任何 CSCIF 中断。 01 只有当接收器进入或离开 “总线脱离”状态时才会生成 CSCIF 中断。为生成 CSCIF 中断丢弃其他接收器状态变化。 10 只有当接收器进入或离开 “RxErr”或 “总线脱离” [2] 状态时才会生成 CSCIF 中断。为生成 CSCIF 中断丢弃其他接收器状态变化。 11 所有状态变化都生成 CSCIF 中断。
3:2 TSTATE[1:0]	**T** 发送器状态变化使能—这些 TSTAT 使能位控制发送器状态变化而引起 CSCIF 中断的电平状态。独立于所选电平状态， TSTAT 标志继续显示实际发送器状态，且只有在没有 CSCIF 中断产生时才会更新。 00 未生成由于接收器状态变化而引起的任何 CSCIF 中断。 01 只有当发送器进入或离开 “总线脱离”状态时才会生成 CSCIF 中断。为生成 CSCIF 中断丢弃其他接收器状态变化。 10 只有当发送器进入或离开 “总线脱离”状态时才会生成 CSCIF 中断。为生成 CSCIF 中断丢弃其他接收器状态变化。 11 所有状态变化都生成 CSCIF 中断。
1 OVRIE	**溢出中断使能** 0 无中断请求从该事件中生成。 1 溢出事件引起错误中断请求。
0 RXFIE	接收器已满中断使能 0 无中断请求从该事件中生成。 1 接收缓冲器已满 （成功报文接收）事件引起接收器中断请求。

[1] 如果需要从停止或等待模式中进行恢复的机制，必须同时使能 WUPIE 和 WUPE(参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”).

12.3.6 MSCAN 发送器标志寄存器 (CANTFLG)

每个发送缓冲器空标志在 CANTIER 寄存器中都有相关的中断使能位。 r.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXE2	TXE1	TXE0
W								
复位 :	0	0	0	0	0	1	1	1

= 不执行

图 12-10. MSCAN 发送器标志寄存器 (CANTFLG)

12.4 报文存储模式

下面这一节详细介绍了接收和发送报文缓冲器的结构，以及相关控制寄存器。

了简化程序员界面，接收和发送报文缓冲器的轮廓相同。每个报文缓冲器都在包含 13 字节数据结构的存储器映射中都分配 16 个字节。

我们还为发送缓冲器定义了一个发送缓冲器优先级寄存器 （TBPR）。在该存储器映射的最后两个字节中， MSCAN 保存一个特殊的 16 位时间标签，采样于报文成功传输或接收后的内部计时器。如果设置了 TIME 位，这种功能只出现于发送和接收器缓冲器 (参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”)。

时间标签寄存器由 MSCAN 写入。 CPU 只能读这些寄存器。

表 12-24. 报文缓冲器结构

Offset Address	Register	Access
0x00X0	Identifier Register 0	
0x00X1	Identifier Register 1	
0x00X2	Identifier Register 2	
0x00X3	Identifier Register 3	
0x00X4	Data Segment Register 0	
0x00X5	Data Segment Register 1	
0x00X6	Data Segment Register 2	
0x00X7	Data Segment Register 3	
0x00X8	Data Segment Register 4	
0x00X9	Data Segment Register 5	
0x00XA	Data Segment Register 6	
0x00XB	Data Segment Register 7	
0x00XC	Data Length Register	
0x00XD	Transmit Buffer Priority Register[1]	
0x00XE	Time Stamp Register (High Byte)[2]	
0x00XF	Time Stamp Register (Low Byte)[3]	

1 Not applicable for receive buffers

2 Read-only for CPU

3 Read-only for CPU

图 12-23 为扩展标识符显示了接收和发送缓冲器常用的 13 字节数据结构。标准标识符在 IDR 寄存器中的映射图 12-24。

由于基于 RAM1，接收和发送缓冲器的所有位在复位时均为 ‘x’。接收和发送缓冲器的所有保留或不使用位始终读为 ‘x’。

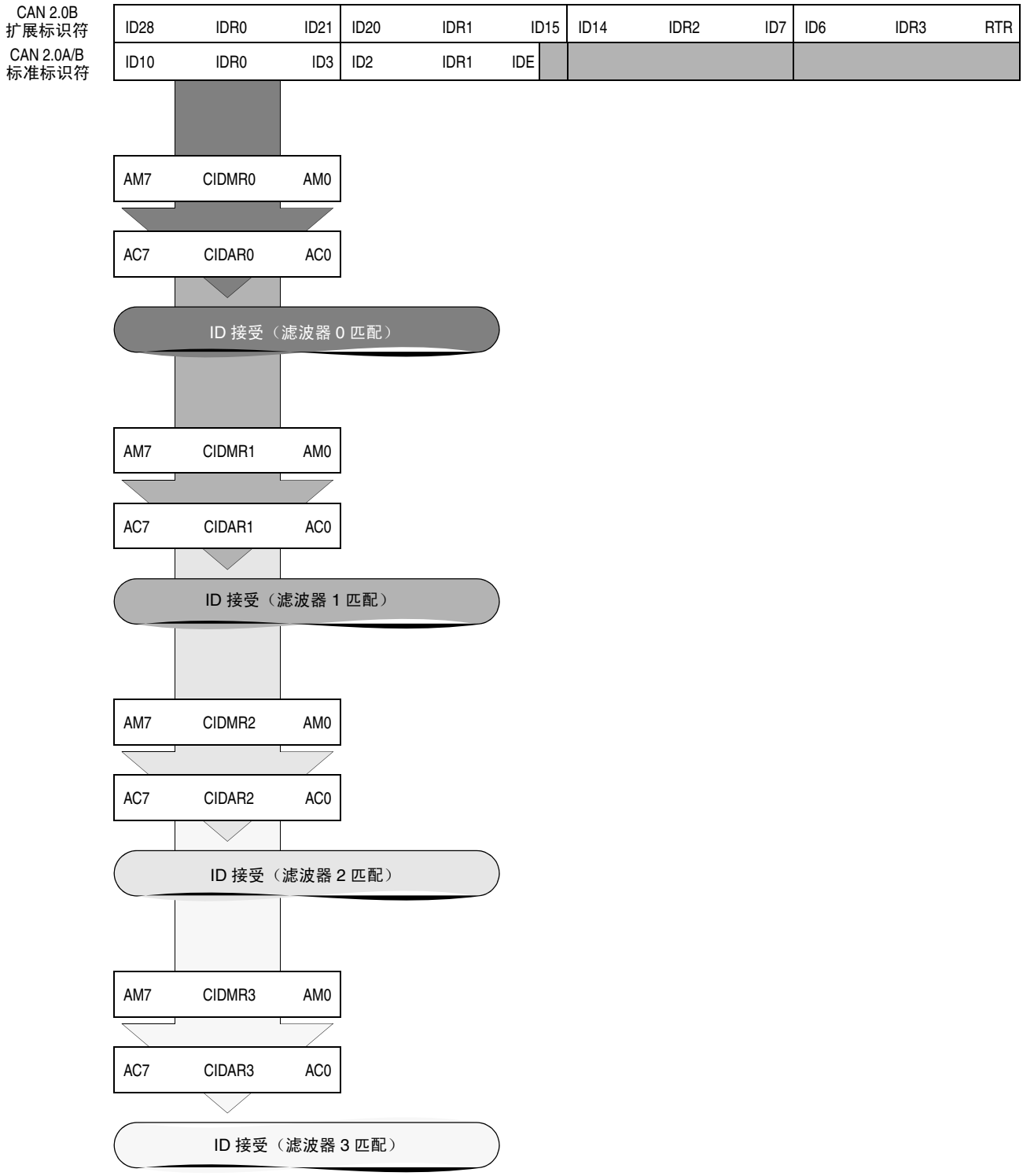

图 12-41. 8 位可屏蔽标识符接收滤波器

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

适当位位置来决定将比较的滤波器位；最后，寄存器 CANIDAR0/1/2/3 决定 CANIDMR0/1/2/3 所决定的位的值。

例如，当滤波器值为：

0001x1001x0

The CANIDMR0/1/2/3 寄存器将被配置为：

00001000010

因此，所有报文标识符位 （除位 1 和位 6）会与 CANIDAR0/1/2/3 寄存器进行比较。这些寄存器将配置为：

00010100100

在这种情况下，位 1 和 6 设置为 ‘0’，但由于它们被忽略，因此效力等同于把它们设置为 ‘1’。

12.5.3.1 标识符接收滤波器示例

正如上面所描述的那样，滤波器是通过与 CAN 报文标识符字段中的个别位的比较进行工作的。滤波器将检查标准 CAN 报文标识符的 11 位的每个位。假设滤波器值为 0001x1001x0。在这个简单示例中，只可能有三种 CAN 报文。

滤波器值：0001x1001x0

报文 1: 00011100110

报文 2: 00110100110

报文 3: 00010100100

由于第三高位不是 ‘0’ – 001，报文 2 被拒绝。滤波器只是提供了 CPU 需要接收的报文组的便捷方式。对于扩展 CAN 报文标识符的全部 29 位，滤波器能辨认两组报文：一组是它接收的报文，一组是它拒绝的报文。此外，滤波器可以一分为二。这允许 MSCAN 只检查报文标识符的前 16 位，但允许让两个独立滤波器执行检查。见下面的示例：

滤波器值 A: 0001x1001x0

滤波器值 B: 00x101x01x0

报文 1: 00011100110

报文 2: 00110100110

报文 3: 00010100100

MSCAN 会接受所有三条报文。滤波器 A 像以前一样将接受报文 1 和 3 ，滤波器 B 接受报文 2。实践中，哪个滤波器接受报文并不重要，任意一个滤波器接受的报文都将放入输入缓冲器。一条报文可由多个滤波器接受。

表 12-37. 中断矢量

中断源	CCR 掩码	本地使能
唤醒中断 （WUPIF）	I 位	CANRIER (WUPIE)
错误中断 (CSCIF, OVRIF)	I 位	CANRIER(CSCIE, OVRIE)
接入中断 (RXF)	I 位	CANRIER (RXFIE)
发送中断 (TXE[2:0])	I bit	CANRIER (TXEIE[2:0])

12.5.7.2 发送中断

三个发送缓冲器中至少有一个空 （未安排发送），并可以写入报文发送。空报文缓冲器的 TXEx 标志已置位。

12.5.7.3 接收中断

报文成功接收，并转移到接收器 FIFO 的前景缓冲器 （RxFG）。收到 EOF 符号后，立即生成该中断。RXF 标志已置位。如果接收器 FIFO 中有多条报文，一旦下一条报文转移到前景缓冲器，就立即设置 RXF 标志。

12.5.7.4 唤醒中断

如果在 MSCAN 内部睡眠模式期间 CAN 总线上有信号，就生成唤醒中断。 WUPE （参见 12.3.1，“MSCAN 控制寄存器 0 (CANCTL0)”）必须使能。

12.5.7.5 错误中断

如果出现了接收器 FIFO 溢出、错误、警报或总线脱离情况，就出现错误中断 . 12.3.4.1，“MSCAN 接收器标志寄存器 (CANRFLG)”显示以下情况中的一种：

- **溢出** — 出现了如 12.5.2.3，“接收结构”所述的接收器 FIFO 的溢出情况。
- **CAN** 状态变化 — 实际值控制着 MSCAN 的 CAN 总线状态。只要错误计数器进入关键范围（Tx/Rx 警报、 Tx/Rx 错误、总线脱离），MSCAN 就标志错误情况。造成错误情况的状态变化用 TSTAT 和 RSTAT 标志表示 (参见 12.3.4.1，“MSCAN 接收器标志寄存器 (CANRFLG)”和 12.3.5，“MSCAN 接收器中断使能寄存器 (CANRIER)”）。

12.5.7.6 中断响应

中断与 12.3.4.1，“MSCAN 接收器标志寄存器 (CANRFLG)”或 12.3.6，“MSCAN 发送器标志寄存器 (CANTFLG)”中的一个或多个状态标志相关。CANRFLG 和 CANTFLG 中的标志必须在中断处理程序内复位。将 1 写入相应位来清零标志。如果中断条件仍然存在，标志不能清除。只要设置了相应标志中的一个，中断就产生。

注意

必须确保 CPU 只清除引起当前中断的位。正是因为这个原因，不能用位操作指令 （BSET）清除中断标志。这种指令可能造成意外清除进入 当前中断服务程序后设置的中断标志。

14.3.3 接收器功能描述

在本小节，首先把接收器结构图 (图 14-3) 作为接收器总体功能描述的指南使用。然后详细描述了用来重建接收器数据的数据采样方法。最后解释了接收器唤醒功能的两个变种。

通过设置 RXINV = 1，接收器输入被反转。通过设置 SCIxC2 中的 RE 位，接收器被使能。字符帧由逻辑 0 的起始位、 8 个 （或 9 个）数据位 （LSB 先发）和逻辑 1 的停止位组成。如需了解 9 位数据模式的有关信息，参见 14.3.5.1，“8 位和 9 位数据模式” .” 对于本小节的其余部分，我们假设 SCI 配置用于正常的 8 位数据模式。

在把停止位接收到接收移位器后，如果接收数据寄存器还未满，数据字符就被传输到接收数据寄存器，设置接收数据寄存器已满 （RDRF）状态标记。如果已经设置了显示接收数据寄存器（缓冲器）已满的 RDRF，就设置溢出 （OR）状态标记，新数据丢失。因为 SCI 接收器是双缓冲的，程序在设置 RDRF 后、读取接收数据缓冲器的数据前，有一个全字符时间，以避免接收器溢出。

当程序检测到接收数据寄存器已满 （RDRF = 1）时，它通过读 SCIxD 从接收数据寄存器中获取数据。 RDRF 标记由一个 2 步式顺序自动清除，这个 2 步式顺序通常在处理接收数据的用户程序中满足。如需了解标记清除的更多详细信息，参见 14.3.4，“中断和状态标记”。

14.3.3.1 数据采样方法

SCI 接收器使用 16 倍波特率时钟进行采样。接收器通过以 16 倍波特率提取逻辑电平样本，以搜索 RxD 串行数据输入管脚上的下降边沿。下降边沿的定义是 3 个连续逻辑 1 采样后的逻辑 0 样本。 16 倍波特率时钟用来把位时间划分为 16 个段，分别标记为 RT1 到 RT16。当定位了下降边沿时，还要从 RT3、 RT5 和 RT7 中提取三个样本，以确保这是真正的起始位，而不仅仅噪音。如果这三个样本至少有两个样本为 0，接收器假设它与接收器字符同步。

接收器然后在 RT8, RT9 和 RT10 的每个位时间上进行采样，包括起始启位和停止位，以决定该位的逻辑电平。逻辑电平是位时间期间提取的绝大多数样本的逻辑电平。在起始位中，如果 RT3、 RT5 和 RT7 上的样本中至少有 2 个样本为 0，那么就假设该位为 0，即便在 RT8、 RT9 和 RT10 上提取的一个或所有样本均为 1。如果字符帧的任意位时间 （包括起始和停止位）中的任意样本不能与该位的逻辑电平保持一致，当收到的字符传输到接收数据缓冲器时，都设置噪音标记 （NF）。

下降边沿检测逻辑不断寻找下降边沿，如果检测到边沿，样本时钟重新同步位时间。这样当出现噪音或不匹配波特率时，就可以提高接收器的可靠性。它不能改进最坏情况分析，因为有些字符在字符帧的任何地方都没有额外的下降边沿。

在成帧错误情况下，假设收到的字符不是中止字符，搜索下降边沿的采样逻辑就充满 3 个逻辑 1 样本，这样一个新起始位几乎可以立即检测到。

在成帧错误情况下，接收器禁止接收任何新字符，直到成帧错误标记被清除。如果仍设置 FE，接收移位寄存器继续发挥作用，但整个字符不能传输到接收数据缓冲器。

15.1.4 结构图

RTC 模块的结构图如图 15-2 所示。

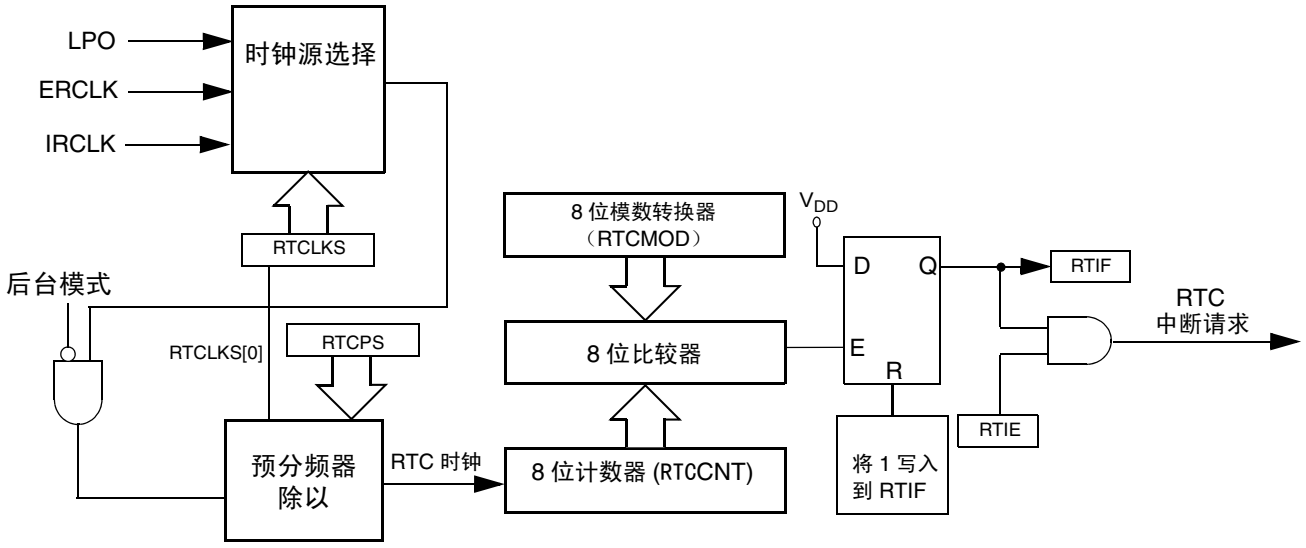

图 15-2. 实时计数器 （RTC）结构图

15.2 外部信号描述

RTC 不包括任何片外信号。

15.3 寄存器定义

RTC 包括一个状态和控制寄存器、一个 8 位计数器寄存器和一个 8 位模数寄存器。

请参考本产品介绍内存部分的 “直接页面寄存器总结”小节，了解所有 RTC 寄存器的绝对地址分配

表 15-1 为 RTC 寄存器的总结。

表 15-1. RTC 寄存器总结

Name		7	6	5	4	3	2	1	0
RTCSC	R	RTIF	RTCLKS		RTIE	RTCPS			
	W								
RTCCNT	R	RTCCNT							
	W								
RTCMOD	R	RTCMOD							
	W								

100%，因为占空比比较将不会发生。这意味着模数寄存器设置的可用范围周期为 0x0001 至 0x7FFE （如果不需要 100% 的占空比，则为 0x7FFF）。这不是一个重大的限制。所能产生的周期将远远长于正常应用所需的周期。

TPMxMODH:TPMxMODL=0x0000 是不用于中央对齐 PWM 模式的特例。当 CPWMS=0 时，这一情况与计数器从 0x0000 自由运行到 0xFFFF 的情况相对应，但当 CPWMS=1 时，计数器需要与 0x0000 以外的模数寄存器有效匹配，以便将方向从向上计数变为向下计数。

TPM 通道寄存器 （2 倍）中的输出比较值决定 CPWM 信号的脉冲宽度 （占空比）（图 16-16）。如果 ELSnA=0，当向上计数时发生数值比较会强制 CPWM 输出信号进入低态；当向下计数时发生数值比较会强制输出进入高态。计数器达到 TPMxMODH:TPMxMODL 中的模数设置后才开始向上计数；然后向下计数直到 0。这将周期设置为 TPMxMODH:TPMxMODL 的两倍。

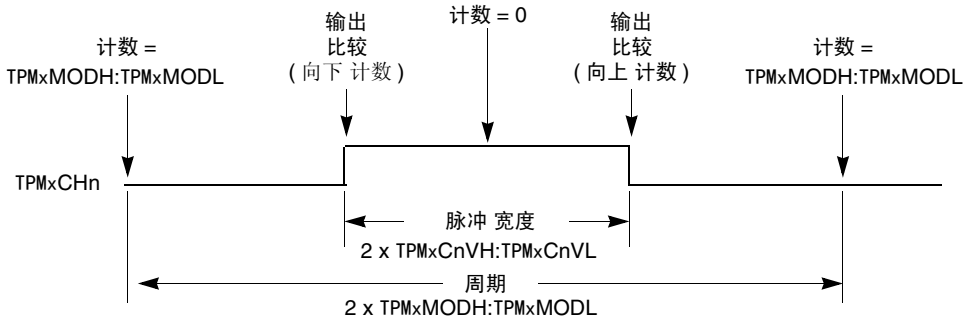

图 16-16. CPWM 周期和脉冲宽度 （ELSnA=0）

中央对齐 PWM 输出的噪音一般比边缘对齐 PWM 小，因为相同系统时钟边上的输入 / 输出管脚过渡更少。有些类型的电机需要这类 PWM 应用。

当计数器以向上 / 向下计数模式运行时，输入捕捉、输出比较和边缘对齐 PWM 功能没有意义。因此这意味着当 CPWMS=1 时， TPM 中的所有使能通道必须用于 CPWM 模式中。

TPM 可用在 8 位 MCU 中。定时器通道寄存器中的设置被缓冲，以确保连贯的 16 位更新并避免意外的 PWM 脉冲宽度。向任何寄存器 TPMxMODH、 TPMxMODL、 TPMxCnVH 和 TPMxCnVL 中写入实际上就是写入到缓冲器寄存器中。

在中央对齐 PWM 模式下， TPMxCnVH:L 寄存器根据 CLKSB:CLKSA 位的值通过写入缓冲器的值得到更新，因此：

- 如果 （clksb:clksa = 0:0），寄存器在第二个字节被写入时更新。
- 如果 （clksb:clksa not = 0:0），寄存器在两个字节都被写入， tpm 计数器从 （tpmxmodh:tpmxmodl - 1）变为 （tpmxmodh:tpmxmodl）后更新。如果 tpm 计数器为自由运行的计数器，那么更新在 tpm 计数器从 0xfffe 变为 0xffff 时进行。

当 TPMxCNTH:TPMxCNTL=TPMxMODH:TPMxMODL 时， TPM 可选择生成 TOF 中断 （在该计数结束时）。

写入 TPMxSC 的操作会取消写入到 TPMxMODH 和 / 或 TPMxMODL 中的任何值，并且复位模数寄存器的一致性机制。写入 TPMxCnSC 的操作会取消写入到通道值寄存器中的任何值，并且为 TPMxCnVH:TPMxCnVL 复位一致性机制。

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。

17.3.5 触发模式

触发模式控制调试器运行的整体行为。DBGT 寄存器中的 4- 位 TRG 字段选择九个触发模块中的一个。当 DBGT 寄存器中的 TRGSEL = 1, 比较器的输出必须在触发 FIFO 操作前通过操作码跟踪电路传播。 DBGT 中的 BEGIN 位选择当检测到合格的触发时 FIFO 是否开始存储数据 （开始跟踪），或 FIFO 从其打开之时开始循环存储数据，直到检测到合格的触发 （结束触发）。

将 1 写入到寄存器中的 ARM 位便可启动调试运行，它设置 DBGS 中的 ARMF 标记，并清除 AF 和 BF 标记及 CNT 位。开始跟踪调试运行当 FIFO 满时结束。结束跟踪运行则在所选触发事件发生时结束。任何调试运行均可通过将 0 写入到 DBGC 中的 ARM 或 DBGEN 位停止。

除纯事件模式外的所有触发模式中，FIFO 都存储流变化地址。在纯事件触发模式中，FIFO 将数据存储在 FIFO 的八低八位。

控制位在纯事件触发模式中被忽略，而且所有这样的调试运行都是开始类型跟踪。当 TRGSEL = 1 选择操作码获取触发器，没有必要在比较中使用 R/W ，因为操作码标签只应用于操作码获取，而这一直都是读周期。在采用全模式触发器时，规定 TRGSEL = 1 也是不正常的，因为操作码的值通常在特定的地址可以知道。

下面的触发模式描述只说明了导致触发的主要比较器条件。比较器 A 或 B 通常都可以被 R/W 进一步鉴定，通过将 RWAEN (RWBEN) 和相应的 RWA (RWB) 值设置为与 R/W 相匹配。如果 BRKEN = 1，来自比较器的带可选 R/W 鉴定的信号，用来请求 CPU 断点， TAG 决定 CPU 请求是标记请求还是强制请求。

只 A— 当地址匹配比较器 A 的值时触发

A 或 B — 当地址匹配比较器 A 或 B 的值时触发

A 然后 B — 当地址匹配比较器 B 但只能在另一个周期的地址匹配比较器 A 的值以后，触发。可能在 A 匹配后 B 匹配前有许多周期。

A 和 B 数据 （全模式）— 这称为全模式，因为地址，数据和 R/W (可选) 必须在同一个总线周期内匹配，才能产生触发事件。比较器 A 检查地址，比较器的低阶字节检查数据，如果 RWAEN = 1， R/W 对照 RWA 进行检查。比较器 B 的高半部分没有使用。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

A 但非 B 数据 （全模式）— 地址必须匹配比较器 A, 数据必须不能匹配比较器 B 的低阶部分，如果 RWAEN = 1， R/W 必须匹配 RWA。所有三个条件必须在同一个总线周期中达到才能引起触发。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

纯事件 B （存储数据）— 当地址每次匹配比较器 B 的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。

A 然后纯事件 B （存储数据）— 当地址匹配比较器 A 中的值后，每次地址匹配比较器 B 中的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。