



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, I ² 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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dz60aclh

表 3-1 列出了影响停止模式选择的所有控制位及各种条件下选择的模式。被选择的模式在执行一个 STOP 指令后进入。

表 3-1. 停止模式选择

STOPE	ENBDM ¹	LVDE	LVDSE	PPDC	停止模式
0	x	x	x	x	停止模式被禁用；如果执行了 STOP 指令，则进行非法 Opcode 代码复位
1	1	x	x	x	Stop3 模式，BDM 被启用 ²
1	0	两个位都必须为 1	x	x	Stop3 模式，电压调节器处于活动状态
1	0	其中一个位为 0	0	0	Stop3 模式
1	0	其中一个位为 0	1	1	Stop2 模式

¹ ENBDM 位于 BDCSCR（只能通过 BDC 命令访问）中。详细信息请参见第 17.4.1.1 部分“BDC 状态和控制寄存器（BDCSCR）”。

² 处于 Stop3 模式而且 BDM 被启用时，因为启用了内部时钟，SIDD 将接近 RIDD 水平。

3.6.1 Stop3 模式

Stop3 模式通过表 3-1 所述条件下执行 STOP 指令进入。所有此时的内部寄存器和逻辑、RAM 内容和 I/O 管脚状态都被维持。

从 Stop3 模式中退出的操作通过 RESET 管脚或异步中断脚实现。这些异步中断脚包括 IRQ、PIA0 ~ PIA7、PIB0 ~ PIB7 和 PID0 ~ PID7。从 Stop3 模式中退出的操作也可以通过低压检测（LVD）复位、低压警告（LVW）中断、ADC 转换完成中断、实时时钟（RTC）中断、MSCAN 唤醒中断或 SCI 接收器中断完成。

如果通过 RESET 脚的方式退出 Stop3 模式，MCU 将复位，操作将在获取复位向量后恢复。如果通过中断的方式退出，MCU 将获取相应的中断向量。

3.6.1.1 在 Stop3 模式中启用 LVD

在电源电压下降到 LVD 电压以下时，LVD 系统可以生成一个中断或复位。在 CPU 执行 STOP 指令时，如果 LVD 在停止模式下被启用（SPMSC1 中的 LVDE 和 LVDSE 位均被设置），那么内部稳压器在停止模式下将继续保持活动状态。

要使 ADC 正常运行，LVD 必须在进入 Stop3 时处于启用状态。

3.6.1.2 在 Stop3 模式中启用活动 BDM

如果 BDCSCR 中设置了 ENBDM，将启用从运行模式进入主动后台模式的操作。该寄存器在第 17 章，“开发支持”。中有详细描述。如果 CPU 执行 STOP 指令时设置了 ENBDM，连接到后台调试逻辑的系统时钟将在 MCU 进入停止模式时继续保持活动状态。因此，这种情况下后台调试通信仍可进行。此外，内部稳压器不会进入低功耗待机状态，而保持正常工作。

大多数后台命令在停止模式不能使用。带状态内存访问命令不允许内存访问，但它们会上报错误，指出 MCU 处于停止或等待模式。可以使用后台命令将 MCU 从停止模式中唤醒并进入主动后台模式（如果 ENBDM 位已设置）。进入后台调试模式后，所有后台命令都可以使用。

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)

0x0000 DIRECT PAGE REGISTERS 0x007F 128 BYTES	0x0000 DIRECT PAGE REGISTERS 0x007F 128 BYTES	0x0000 DIRECT PAGE REGISTERS 0x007F 128 BYTES	0x0000 DIRECT PAGE REGISTERS 0x007F 128 BYTES
0x0080 RAM 4096 BYTES	0x0080 RAM 3072 BYTES	0x0080 RAM 2048 BYTES	0x0080 RAM 1024 BYTES
0x107F 0x1080 FLASH 896 BYTES	0x0C7F 0x0C80 UNIMPLEMENTED 2176 BYTES	0x087F 0x0880 UNIMPLEMENTED 3456 BYTES	0x047F 0x0480 UNIMPLEMENTED 4736 BYTES
0x13FF 0x1400 EEPROM ¹ 2 x 1024 BYTES	0x14FF 0x1500 EEPROM ¹ 2 x 768 BYTES	0x15FF 0x1600 EEPROM ¹ 2 x 512 BYTES	0x16FF 0x1700 EEPROM ¹ 2 x 256 BYTES
0x17FF 0x1800 HIGH PAGE REGISTERS 256 BYTES	0x17FF 0x1800 HIGH PAGE REGISTERS 256 BYTES	0x17FF 0x1800 HIGH PAGE REGISTERS 256 BYTES	0x17FF 0x1800 HIGH PAGE REGISTERS 256 BYTES
0x18FF 0x1900 UNIMPLEMENTED 59136 BYTES	0x18FF 0x1900 UNIMPLEMENTED 9984 BYTES	0x18FF 0x1900 UNIMPLEMENTED 25,344 BYTES	0x18FF 0x1900 UNIMPLEMENTED 42,240 BYTES
0x3FFF 0x4000 FLASH 59136 BYTES	0x3FFF 0x4000 FLASH 49152 BYTES	0x7BFF 0x7C00 FLASH 33792 BYTES	0xBDFF 0xBE00 FLASH 16896 BYTES
0xFFFF	0xFFFF	0xFFFF	0xFFFF
MC9S08DZ60	MC9S08DZ48	MC9S08DZ32	MC9S08DZ16

¹ 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

安全密钥只能从安全存储器（RAM、EEPROM 或 Flash）中写入，因此在没有安全的用户程序协作的情况下不能通过后台命令输入。

后门对比密钥 (NVBACKKEY through NVBACKKEY+7) 保存在非易失性寄存器空间内的 Flash 位置上，因此用户可以准确地编程这些位置，就象编程任何其他 Flash 位置一样。非易失性寄存器与复位和中断向量在 Flash 的同一个 768 字节块中，因此对这一空间进行块保护同时也可以保护后门对比密钥。块保护不能从用户应用程序上修改，因此，如果向量空间受到块保护，后门安全密钥机制就不能永久性地修改块保护、安全设置或后门密钥。

通过以下步骤，您可以通过后台调试接口始终关闭安全性：

1. 通过写入 FPROT 来禁用任何块保护。FPROT 只能通过后台调试命令写入而不能通过应用软件写入。
2. 在必要时整体擦除 Flash。
3. 对 Flash 进行空白检查。如果 Flash 内完全擦除，那么在下一次复位前安全性一直处于关闭状态。为了避免在下次复位后返回到安全模式，对 NVOPT 进行编程使 SEC = 1:0。

4.5.10 EEPROM 映射

只有一半 EEPROM 处于存储器映象中。FCNFG 寄存器中的 EPGSEL 位用于确定阵列的哪一半可从前台访问，而另一半不能从后台访问。对于配置 8 字节 EEPROM 分区，有两种映射模式可供选择：4 字节模式和 8 字节模式。每种模式都通过 FOPT 寄存器中的 EPGMOD 位确定。

在 4 字节分区模式 (EPGMOD = 0) 中，每个 8 个字节分区被分成两部分，4 个字节在前台，4 个在后台，但都在相同的地址上。EPGSEL 位确定哪 4 个字节可以访问。在分区擦除过程中，整个 8 字节分区（前台的 4 个字节和后台的 4 个字节）都被擦除。

在 8 字节分区模式 (EPGMOD = 1) 中，每个 8 字节分区在一个页面上。EPGSEL 位确定哪些分区在后台。在分区擦除过程中，前台的整个 8 字节分区会被擦除。

4.5.11 Flash 和 EEPROM 寄存器及控制位

Flash 和 EEPROM 模块在高端页面寄存器空间内有 7 个 8 位寄存器，在 Flash 的非易失性寄存器空间内有 3 个位置。这些位置中的两个在复位时被拷贝到两个相应的高端页面控制寄存器中。Flash 中还有一个 8 字节对比密钥。对于 Flash 和 EEPROM 寄存器的绝对地址分配情况，请参见表 4-3 和表 4-5。这一部分用名称指代寄存器和控制位。通常飞思卡尔半导体提供变量定义或头文件用于将这些名称转换为相应的绝对地址。

4.5.11.1 Flash 和 EEPROM 时钟分频寄存器 (FCDIV)

该寄存器的第 7 位是一个只读标记。6:0 位可以在任何时候读取但只能写入一次。在开始任何擦除或编程操作之前，写入该寄存器以将非易失性内存系统的时钟频率设置在可接受的限度内。

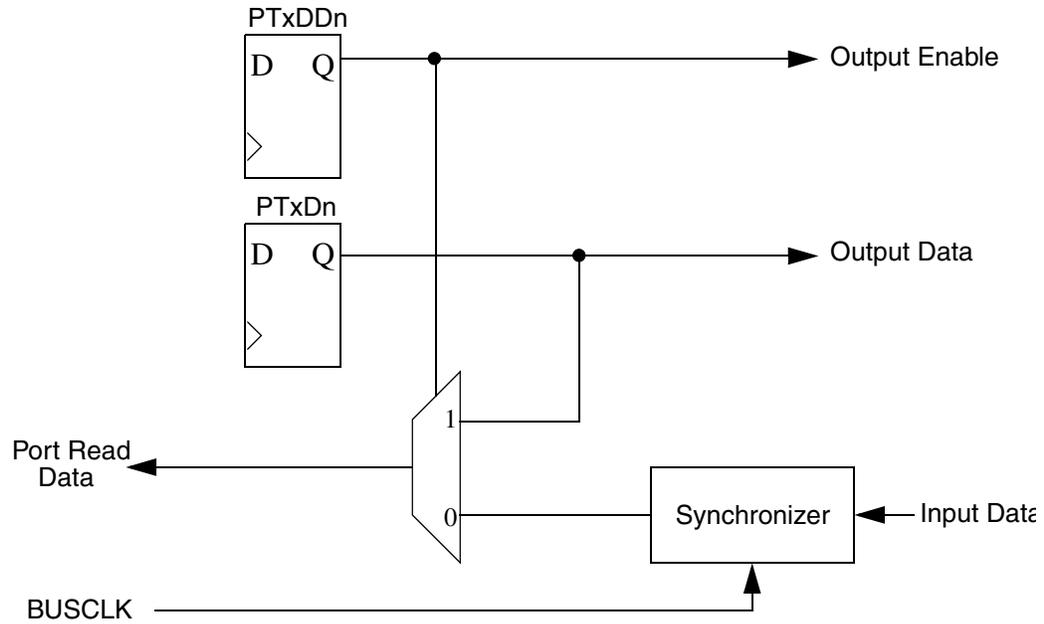


图 6-1. 并行输入 / 输出示意图

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（PTxPE_n）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（PTxSE_n）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（PTxDS_n）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。

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。

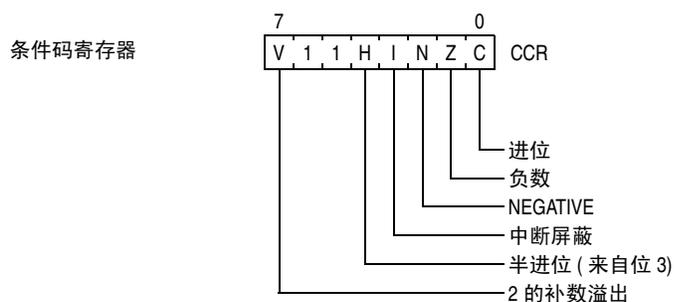


图 7-2. 条件码寄存器

表 7-1. CCR 寄存器字段描述

字段	描述
7 V	2 的补数溢出标记 — 当出现 2 的补数溢出时, CPU 设置溢出标记。带符号的分支指令 BGT、BGE、BLE 和 BLT 使用溢出标记。 0 无溢出 1 溢出
4 H	半进位标记 — 在 add-without-carry (ADD) 或 add-with-carry (ADC) 运算中, 当累加器位 3 和 4 间发生进位时 CPU 设置半进位标记。十进制计数码 (BCD) 算术运算中要求半进位标记。DAA 指令使用 H 和 C 条件码位状态, 将更正数值从原来 BCD 操作数上的 ADD 或 ADC 自动添加到结果中, 以便将结果更正为有效的 BCD 值。 0 位 3 和 4 间无进位 1 位 3 和 4 间有进位
3 I	中断屏蔽位 — 当设置了中断屏蔽时, 禁止所有可屏蔽 CPU 中断。清除中断屏蔽后, CPU 中断使能。发生 CPU 中断时, 中断屏蔽会在 CPU 寄存器保存在堆栈上后但执行中断服务程序第一个指令执行前被自动设置。在执行了任何清除 I (CLI 或 TAP) 的指令后, 中断都不能在指令边界识别。这样可以保证当设置了 I 时, CLI 或 TAP 后的下一个指令总被执行, 而不会出现介于其间的中断。 0 中断使能 1 中断禁止
2 N	负数标志 — 算术运算、逻辑运算或数据处理产生一个负数结果且设置该结果的位 7 时, CPU 设置负数标志。如果已加载或已保存值的最高位是 1, 加载或保存 8 位或 16 位值都会导致设置 N。 0 非负数结果 1 负数结果
1 Z	零标志 — 当算术运算、逻辑运算和数据处理产生 0x00 或 0x0000 结果时, CPU 设置零标志。如果已加载或已保存值的全为 0, 加载或保存 8 位或 16 位值都会导致设置 Z。 0 非 0 结果 1 0 结果
0 C	进位 / 借位标记 — 当加法运算在累加器的位 7 外产生一个进位, 或者减法运算需要一个借位时, CPU 设置进位 / 借位标记。有些指令, 如位测试和分支、移位和旋转, 也清除或设置进位 / 借位标记。 0 位 7 无进位 1 位 7 有进位

表 7-2. 指令集小结 (第 5 页, 共 9 页)

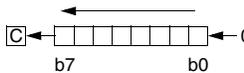
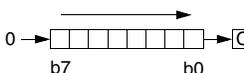
Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V I 1 H I N Z C	
INC <i>opr8a</i> INCA INCX INC <i>opr8,X</i> INC <i>,X</i> INC <i>opr8,SP</i>	增量 M ⁻ (M) + \$01 A ⁻ (A) + \$01 X ⁻ (X) + \$01 M ⁻ (M) + \$01 M ⁻ (M) + \$01 M ⁻ (M) + \$01	DIR INH INH IX1 IX SP1	3C dd 4C 5C 6C ff 7C 9E 6C ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↑ 1 1 -	- ↓ ↓ -
JMP <i>opr8a</i> JMP <i>opr16a</i> JMP <i>opr16,X</i> JMP <i>opr8,X</i> JMP <i>,X</i>	跳转 PC ⁻ 跳转地址	DIR EXT IX2 IX1 IX	BC dd CC hh ll DC ee ff EC ff FC	3 4 4 3 3	ppp pppp pppp ppp ppp	- 1 1 -	- - - - -
JSR <i>opr8a</i> JSR <i>opr16a</i> JSR <i>opr16,X</i> JSR <i>opr8,X</i> JSR <i>,X</i>	跳转到子程序 PC ⁻ (PC) + n (n = 1, 2, 或 3) 推 (PCL); SP ⁻ (SP) - \$0001 推 (PCH); SP ⁻ (SP) - \$0001 PC ⁻ 无条件地址	DIR EXT IX2 IX1 IX	BD dd CD hh ll DD ee ff ED ff FD	5 6 6 5 5	ssppp pssppp pssppp ssppp ssppp	- 1 1 -	- - - - -
LDA # <i>opr8i</i> LDA <i>opr8a</i> LDA <i>opr16a</i> LDA <i>opr16,X</i> LDA <i>opr8,X</i> LDA <i>,X</i> LDA <i>opr16,SP</i> LDA <i>opr8,SP</i>	从存储器那里加载累加器 A ⁻ (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 ii B6 dd C6 hh ll D6 ee ff E6 ff F6 9E D6 ee ff 9E E6 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- ↓ ↓ -
LDHX # <i>opr16i</i> LDHX <i>opr8a</i> LDHX <i>opr16a</i> LDHX <i>,X</i> LDHX <i>opr16,X</i> LDHX <i>opr8,X</i> LDHX <i>opr8,SP</i>	加载索引寄存器 (H:X) H:X ⁻ (M:M + \$0001)	IMM DIR EXT IX IX2 IX1 SP1	45 jj kk 55 dd 32 hh ll 9E AE 9E BE ee ff 9E CE ff 9E FE ff	3 4 5 5 6 5 5	ppp rrpp prpp prrrfp pprrpp prrrpp prrrpp	0 1 1 -	- ↓ ↓ -
LDX # <i>opr8i</i> LDX <i>opr8a</i> LDX <i>opr16a</i> LDX <i>opr16,X</i> LDX <i>opr8,X</i> LDX <i>,X</i> LDX <i>opr16,SP</i> LDX <i>opr8,SP</i>	从存储器那里加载 X (索引寄存器低) X ⁻ (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE ii BE dd CE hh ll DE ee ff EE ff FE 9E DE ee ff 9E EE ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- ↓ ↓ -
LSL <i>opr8a</i> LSLA LSLX LSL <i>opr8,X</i> LSL <i>,X</i> LSL <i>opr8,SP</i>	逻辑左移位 t  (同 ASL)	DIR INH INH IX1 IX SP1	38 dd 48 58 68 ff 78 9E 68 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↑ 1 1 -	- ↓ ↓ ↓
LSR <i>opr8a</i> LSRA LSRX LSR <i>opr8,X</i> LSR <i>,X</i> LSR <i>opr8,SP</i>	逻辑右移位 t 	DIR INH INH IX1 IX SP1	34 dd 44 54 64 ff 74 9E 64 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↑ 1 1 -	- 0 ↓ ↓

表 7-3. 操作码映射 (第 1 页, 共 2 页)

Bit-Manipulation		Branch		Read-Modify-Write				Control				Register/Memory							
00 5 BRSET0 3 DIR	10 5 BSET0 2 DIR	20 3 BRA REL	30 5 NEG DIR	40 1 NEGA 1 INH	50 1 NEGX 1 INH	60 5 NEG IX1	70 4 NEG IX	80 9 RTI 1 INH	90 3 BGE REL	A0 2 SUB IMM	B0 3 SUB DIR	C0 4 SUB EXT	D0 4 SUB IX2	E0 3 SUB IX1	F0 3 SUB IX				
01 5 BRCLR0 3 DIR	11 5 BCLR0 2 DIR	21 3 BRN REL	31 5 CBEQ DIR	41 4 CBEQA 3 IMM	51 4 CBEQX 3 IMM	61 5 CBEQ IX1+	71 5 CBEQ IX+	81 6 RTS 1 INH	91 3 BLT REL	A1 2 CMP IMM	B1 3 CMP DIR	C1 4 CMP EXT	D1 4 CMP IX2	E1 3 CMP IX1	F1 3 CMP IX				
02 5 BRSET1 3 DIR	12 5 BSET1 2 DIR	22 3 BHI REL	32 5 LDHX EXT	42 5 MUL 1 INH	52 6 DIV 1 INH	62 1 NSA 1 INH	72 4 DAA 1 INH	82 5+ BGND 1 INH	92 3 BGT REL	A2 2 SBC IMM	B2 3 SBC DIR	C2 4 SBC EXT	D2 4 SBC IX2	E2 3 SBC IX1	F2 3 SBC IX				
03 5 BRCLR1 3 DIR	13 5 BCLR1 2 DIR	23 3 BLS REL	33 5 COM DIR	43 1 COMA 1 INH	53 1 COMX 1 INH	63 5 COM IX1	73 4 COM IX	83 11 SWI 1 INH	93 3 BLE REL	A3 2 CPX IMM	B3 3 CPX DIR	C3 4 CPX EXT	D3 4 CPX IX2	E3 3 CPX IX1	F3 3 CPX IX				
04 5 BRSET2 3 DIR	14 5 BSET2 2 DIR	24 3 BCC REL	34 5 LSR DIR	44 1 LSRA 1 INH	54 1 LSRX 1 INH	64 5 LSR IX1	74 4 LSR IX	84 1 TAP 1 INH	94 2 TXS 1 INH	A4 2 AND IMM	B4 3 AND DIR	C4 4 AND EXT	D4 4 AND IX2	E4 3 AND IX1	F4 3 AND IX				
05 5 BRCLR2 3 DIR	15 5 BCLR2 2 DIR	25 3 BCS REL	35 4 STHX DIR	45 3 LDHX 3 IMM	55 4 LDHX DIR	65 3 CPHX 3 IMM	75 5 CPHX DIR	85 1 TPA 1 INH	95 2 TSX 1 INH	A5 2 BIT IMM	B5 3 BIT DIR	C5 4 BIT EXT	D5 4 BIT IX2	E5 3 BIT IX1	F5 3 BIT IX				
06 5 BRSET3 3 DIR	16 5 BSET3 2 DIR	26 3 BNE REL	36 5 ROR DIR	46 1 RORA 1 INH	56 1 RORX 1 INH	66 5 ROR IX1	76 4 ROR IX	86 3 PULA 1 INH	96 5 STHX EXT	A6 2 LDA IMM	B6 3 LDA DIR	C6 4 LDA EXT	D6 4 LDA IX2	E6 3 LDA IX1	F6 3 LDA IX				
07 5 BRCLR3 3 DIR	17 5 BCLR3 2 DIR	27 3 BEQ REL	37 5 ASR DIR	47 1 ASRA 1 INH	57 1 ASRX 1 INH	67 5 ASR IX1	77 4 ASR IX	87 2 PSHA 1 INH	97 1 TAX 1 INH	A7 2 AIS IMM	B7 3 STA DIR	C7 4 STA EXT	D7 4 STA IX2	E7 3 STA IX1	F7 2 STA IX				
08 5 BRSET4 3 DIR	18 5 BSET4 2 DIR	28 3 BHCC REL	38 5 LSL DIR	48 1 LSLA 1 INH	58 1 LSLX 1 INH	68 5 LSL IX1	78 4 LSL IX	88 3 PULX 1 INH	98 1 CLC 1 INH	A8 2 EOR IMM	B8 3 EOR DIR	C8 4 EOR EXT	D8 4 EOR IX2	E8 3 EOR IX1	F8 3 EOR IX				
09 5 BRCLR4 3 DIR	19 5 BCLR4 2 DIR	29 3 BHCS REL	39 5 ROL DIR	49 1 ROLA 1 INH	59 1 ROLX 1 INH	69 5 ROL IX1	79 4 ROL IX	89 2 PSHX 1 INH	99 1 SEC 1 INH	A9 2 ADC IMM	B9 3 ADC DIR	C9 4 ADC EXT	D9 4 ADC IX2	E9 3 ADC IX1	F9 3 ADC IX				
0A 5 BRSET5 3 DIR	1A 5 BSET5 2 DIR	2A 3 BPL REL	3A 5 DEC DIR	4A 1 DECA 1 INH	5A 1 DECX 1 INH	6A 5 DEC IX1	7A 4 DEC IX	8A 3 PULH 1 INH	9A 1 CLI 1 INH	AA 2 ORA IMM	BA 3 ORA DIR	CA 4 ORA EXT	DA 4 ORA IX2	EA 3 ORA IX1	FA 3 ORA IX				
0B 5 BRCLR5 3 DIR	1B 5 BCLR5 2 DIR	2B 3 BMI REL	3B 7 DBNZ DIR	4B 4 DBNZA 2 INH	5B 4 DBNZX 2 INH	6B 7 DBNZ IX1	7B 6 DBNZ IX	8B 2 PSHH 1 INH	9B 1 SEI 1 INH	AB 2 ADD IMM	BB 3 ADD DIR	CB 4 ADD EXT	DB 4 ADD IX2	EB 3 ADD IX1	FB 3 ADD IX				
0C 5 BRSET6 3 DIR	1C 5 BSET6 2 DIR	2C 3 BMC REL	3C 5 INC DIR	4C 1 INCA 1 INH	5C 1 INCX 1 INH	6C 5 INC IX1	7C 4 INC IX	8C 1 CLRH 1 INH	9C 1 RSP 1 INH	BC 3 JMP DIR	CC 4 JMP EXT	DC 4 JMP IX2	EC 3 JMP IX1	FC 3 JMP IX					
0D 5 BRCLR6 3 DIR	1D 5 BCLR6 2 DIR	2D 3 BMS REL	3D 4 TST DIR	4D 1 TSTA 1 INH	5D 1 TSTX 1 INH	6D 4 TST IX1	7D 3 TST IX	8D 1 NOP 1 INH	9D 1 NOP 1 INH	AD 5 BSR REL	BD 5 JSR DIR	CD 6 JSR EXT	DD 6 JSR IX2	ED 5 JSR IX1	FD 5 JSR IX				
0E 5 BRSET7 3 DIR	1E 5 BSET7 2 DIR	2E 3 BIL REL	3E 6 CPHX EXT	4E 5 MOV DD	5E 5 MOV DIX+	6E 4 MOV IMD	7E 5 MOV IX+D	8E 2+ STOP 1 INH	9E Page 2	AE 2 LDX IMM	BE 3 LDX DIR	CE 4 LDX EXT	DE 4 LDX IX2	EE 3 LDX IX1	FE 3 LDX IX				
0F 5 BRCLR7 3 DIR	1F 5 BCLR7 2 DIR	2F 3 BIH REL	3F 5 CLR DIR	4F 1 CLRA 1 INH	5F 1 CLR 1 INH	6F 5 CLR IX1	7F 4 CLR IX	8F 2+ WAIT 1 INH	9F 1 TXA 1 INH	AF 2 AIX IMM	BF 3 STX DIR	CF 4 STX EXT	DF 4 STX IX2	EF 3 STX IX1	FF 2 STX IX				

INH 固有
IMM 立即
DIR 直接
EXT 扩展
DD DIR 至 DIR
IX+D IX+ 至 DIR
REL 相关
IX 索引、无偏移
IX1 索引、8 位偏移
IX2 索引、16 位偏移
IMD IMM 至 DIR
DIX+ DIR 至 IX+

SP1 堆栈指针, 8 位偏移
SP2 堆栈指针, 16 位偏移
IX+ 有索引、无偏移、带后增量
IX1+ 有索引、1 字节偏移、带后增量

十六进制操作码 F0 3
字节数 SUB IX HCS08 周期
指令助记符
寻址模式

表 10-10. APCTL2 寄存器字段描述

字段	描述
7 ADPC15	ADC 管脚控制 15 — ADPC15 用来控制与通道 AD15 连接的管脚。 0 AD15 管脚 I/O 控制使能 1 AD15 管脚 I/O 控制禁止
6 ADPC14	ADC 管脚控制 14 — ADPC14 用来控制与通道 AD14 连接的管脚。 0 AD14 管脚 I/O 控制使能 1 AD14 管脚 I/O 控制禁止
5 ADPC13	ADC 管脚控制 13 — ADPC13 用来控制与通道 AD13 连接的管脚。 0 AD13 管脚 I/O 控制使能 1 AD13 管脚 I/O 控制禁止
4 ADPC12	ADC 管脚控制 12 — ADPC12 用来控制与通道 AD12 连接的管脚。 0 AD12 管脚 I/O 控制使能 1 AD12 管脚 I/O 控制禁止
3 ADPC11	ADC 管脚控制 11 — ADPC11 用来控制与通道 AD11 连接的管脚。 0 AD11 管脚 I/O 控制使能 1 AD11 管脚 I/O 控制禁止
2 ADPC10	ADC 管脚控制 10 — ADPC10 用来控制与通道 AD10 连接的管脚。 0 AD10 管脚 I/O 控制使能 1 AD10 管脚 I/O 控制禁止
1 ADPC9	ADC 管脚控制 9 — ADPC9 用来控制与通道 AD9 连接的管脚。 0 AD9 管脚 I/O 控制使能 1 AD9 管脚 I/O 控制禁止
0 ADPC8	ADC 管脚控制 8 — ADPC8 用来控制与通道 AD8 连接的管脚。 0 AD8 管脚 I/O 控制使能 1 AD8 管脚 I/O 控制禁止

10.4.10 管脚控制寄存器 3 (APCTL3)

APCTL3 用来控制 ADC 模块的通道 16-23。

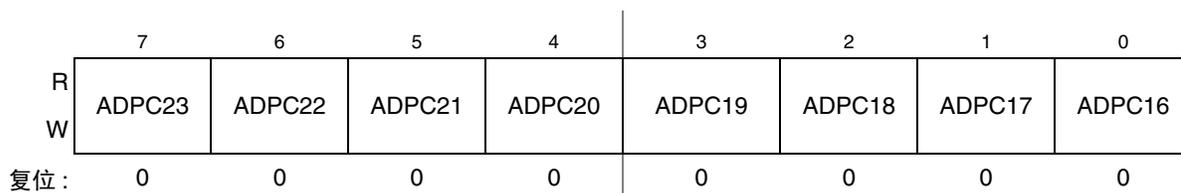


图 10-13. 管脚控制寄存器 3 (APCTL3)

11.4 寄存器定义

本节按地址顺序对 IIC 寄存器进行了描述。

如需了解所有 IIC 寄存器的绝对地址分配报文，请参见本文档存储器章的直接页面寄存器概述。本节仅按参考寄存器和控制位名称对其进行参考。飞思卡尔提供的等式或标头文件用于把这些名称转换成适当的绝对地址。

11.4.1 IIC 地址寄存器 (IICA)

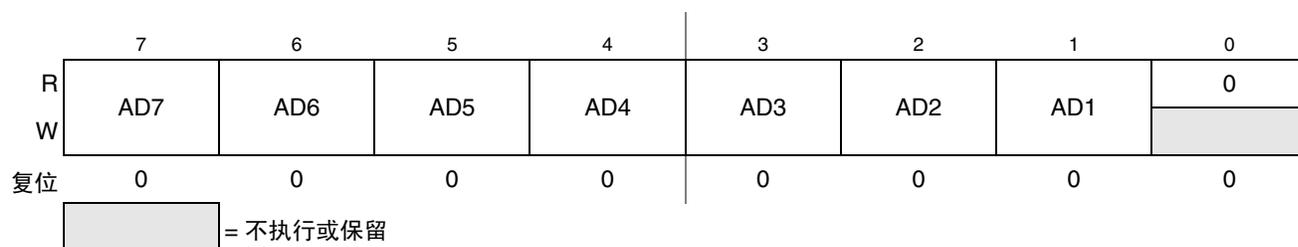


图 11-3. IIC 地址寄存器 (IICA)

表 11-1. IICA 字段描述

字段	描述
7-1 AD[7:1]	从机地址。AD[7:1] 字段包含 IIC 模块将要使用的从机地址。该字段在 7 位地址机制和 10 位地址机制的低 7 位中使用。

11.4.2 11.3.2 IIC 分频器寄存器 (IICF)

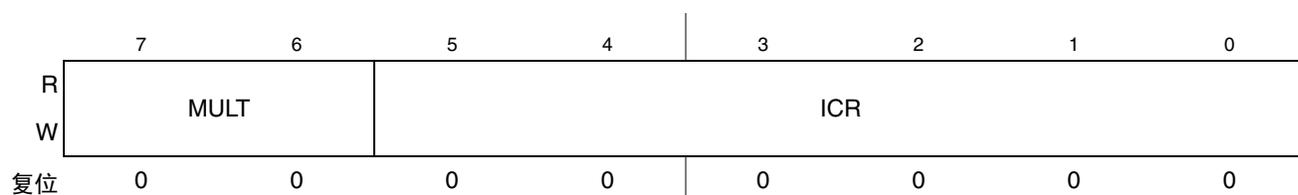


图 11-4. IIC 分频器寄存器 (IICF)

表 12-34. 时段句法

名称	描述
SYNC_SEG	系统希望该时段内在 CAN 总线上出现电平转换。
发送点	正处于发送模式的节点在该点上向 CAN 总线传输一个新值。
采样点	正处于接收模式的节点在该点采样 CAN 总线。如果选择了每位采样三次模式，那么该点标志第三采样点的位置。

同步跳转宽度（如需了解详细报文，参见 Bosch CAN 规范）可以通过设置 SJW 参数，在 1-4 个时间冲量范围内进行编程。

SYNC_SEG、TSEG1、TSEG2 和 SJW 参数通过编程 MSCAN 总线计时寄存器（CANBTR0、CANBTR1）进行设置（参见 12.3.3，“MSCAN 总线计时寄存 0 (CANBTR0)” 和 12.3.4，“MSCAN 总线计时寄存器 (CANBTR1)”）。

表 12-35 概括地描述了 CAN 段设置和相关参数值。

注意

用户有责任确保位时间设置遵从 CAN 标准。

表 12-35. 遵从 CAN 标准的位时段设置

时段 1	TSEG1	时段 2	TSEG2	同步跳转宽度	SJW
5 .. 10	4 .. 9	2	1	1 .. 2	0 .. 1
4 .. 11	3 .. 10	3	2	1 .. 3	0 .. 2
5 .. 12	4 .. 11	4	3	1 .. 4	0 .. 3
6 .. 13	5 .. 12	5	4	1 .. 4	0 .. 3
7 .. 14	6 .. 13	6	5	1 .. 4	0 .. 3
8 .. 15	7 .. 14	7	6	1 .. 4	0 .. 3
9 .. 16	8 .. 15	8	7	1 .. 4	0 .. 3

12.5.4 运行模式

12.5.4.1 正常模式

在普通系统模式中，MSCAN 模块如本规范所述运行。

12.5.4.2 特殊模式

在特殊系统模式中，MSCAN 模块如本规范所述运行。

12.5.4.3 仿真模式

在所有仿真模式中，MSCAN 模块如在普通系统模式下一样，如本规范所述运行。

表 13-1. SPIC1 字段描述 (continued)

字段	描述
1 SSOE	辅选择输出使能 — 该位的使用结合 SPCR2 中的模式故障使能 (MODFEN) 位和主从 (MSTR) 控制位, 以确定 SS 管脚的功能, 如表 13-2 所示。
0 LSBFE	LSB 先发 (移位器方向) 0 SPI 串行数据传输始于最高位 1 SPI 串行数据传输始于最低位

表 13-2. SS 管脚功能

MODFEN	SSOE	主模式	辅模式
0	0	通用 I/O (非 SPI)	从选择输入
0	1	通用 I/O (非 SPI)	从选择输入
1	0	模式故障的 SS 输入	从选择输入
1	1	自动 SS 输出	从选择输入

注意

确保在位更改为 CPHA 位的同时 SPI 不得禁止 (SPE=0)。这些更改应作为独立操作执行, 否则可能发生意外。

13.4.2 SPI 控制寄存器 2 (SPIC2)

该读 / 写寄存器用来控制 SPI 系统的可选功能。位 7、6、5 和 2 不执行, 始终读为 0。

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
复位:	0	0	0	0	0	0	0	0

= 不执行或预留

图 13-6. SPI 控制寄存器 2 (SPIC2)

表 13-3. SPIC2 寄存器字段描述

字段	描述
4 MODFEN	主模式故障功能使能 — 当为辅模式配置 SPI 时, 该位没有意义或影响 (SS 管脚是从选择输入)。在主模式中, 该位决定 SS 管脚的使用方式 (如需了解更多信息, 参见表 13-2)。 0 模式故障功能禁止, 主 SS 管脚恢复为不受 SPI 控制的通用 I/O。 1 模式故障功能使能, 主 SS 管脚用作模式故障输入或辅选择输出
3 BIDIROE	双向模式输出使能 — 双向模式由 SPI 管脚控制 0 (SPC0 = 1) 使能时, BIDIROE 决定 SPI 数据输出驱动器是否被使能为单个双向 SPI I/O 管脚。根据 SPI 是配置为主 SPI 还是从 SPI, 它将 MOSI (MOMI) 或 MISO (SISO) 管脚分别用作单个 SPI 数据 I/O 管脚, 当 SPC0 = 0, BIDIROE 没有意义或影响。 0 输出驱动器禁止, 因此 SPI 数据 I/O 管脚作为输入 1 SPI I/O 管脚作为输出使能

13.5.1 SPI 时钟格式

为了适应不同制造商的各种同步串行外围器件，SPI 系统有一个时钟极性（CPOL）位和一个时钟相位（CPHA）控制位，从四种时钟格式中选择一种进行数据传输。CPOL 有选择性地插入了与时钟串行的取反逻辑。CPHA 选择时钟和数据间的两种不同时钟相位关系。

图 13-10 显示了 CPHA = 1 时的时钟格式。在图的顶部，显示了 8 个位时间，作为参考。第一个位始于第一个 SPSCCK 边沿，第八个位结束于第 16 个 SPSCCK 边沿后的半个 SPSCCK 周期。MSB First 和 LSB First 线根据 LSBFE 中的设置显示了 SPI 数据位的顺序。SPSCCK 极性的两个变化都显示了出来，但这两个波形中只有一个适用于特定传输，具体哪一个取决于 CPOL 中的值。SAMPLE IN 波形适用于辅器件的 MOSI 输入或主器件的 MISO 输入。MOSI 波形适用于主器件的 MOSI 输出管脚，MISO 波形适用于辅器件的 MISO 输出。SS OUT 波形适用于主器件的辅选择输出（如果 MODFEN，SSOE = 1）。在传输开始前的半个 SPSCCK 周期，主 SS 输出信号进入有效低电平状态，在传输的第 8 个位时间结束时返回有效高态。SS IN 波形适用于辅器件的辅选择输入。

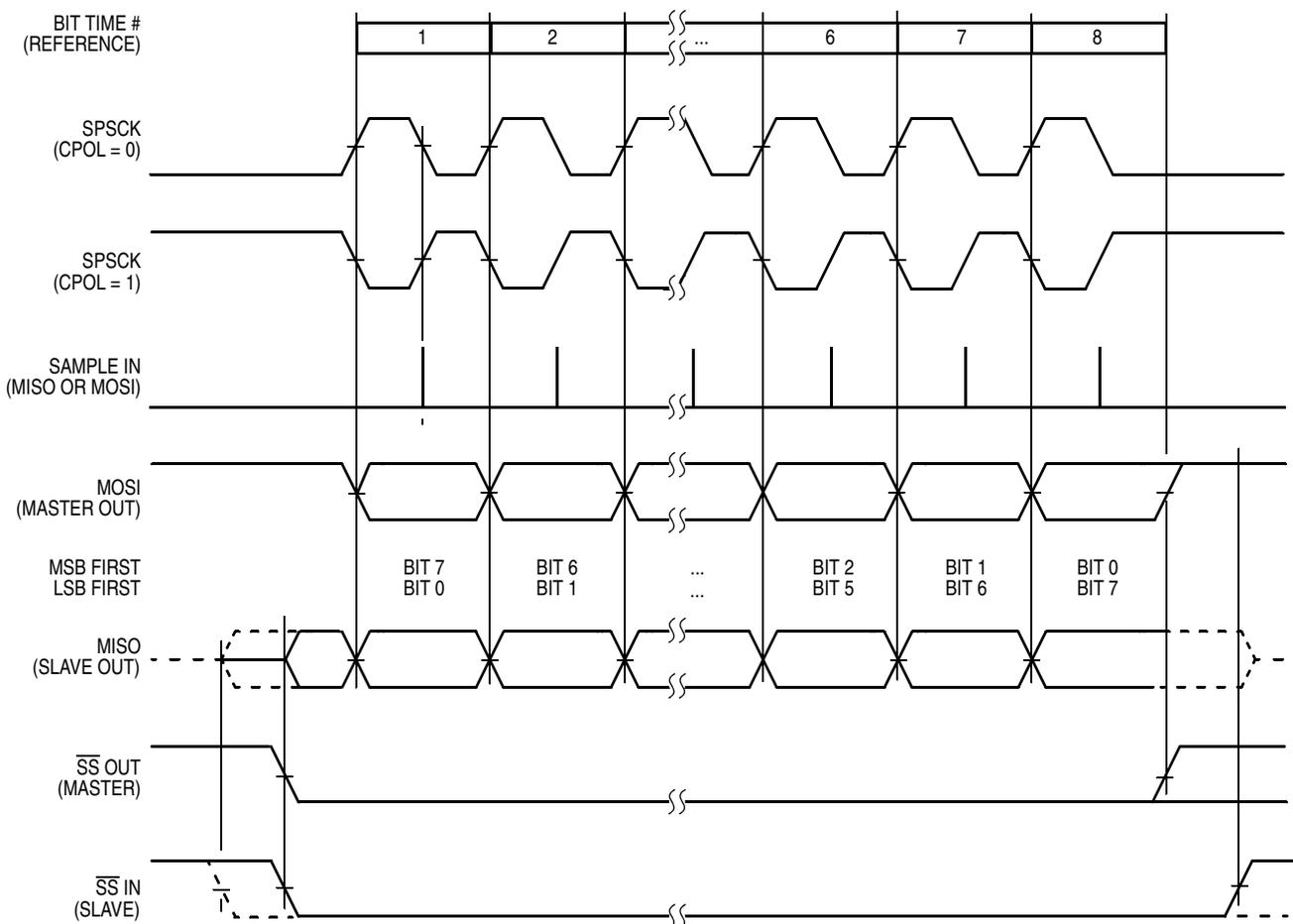


图 13-10. SPI 钟格式 (CPHA = 1)

15.3.2 RTC 计数器寄存器 (RTCCNT)

RTCCNT 是 8 位计数器的当前 RTC 计数的只读值。

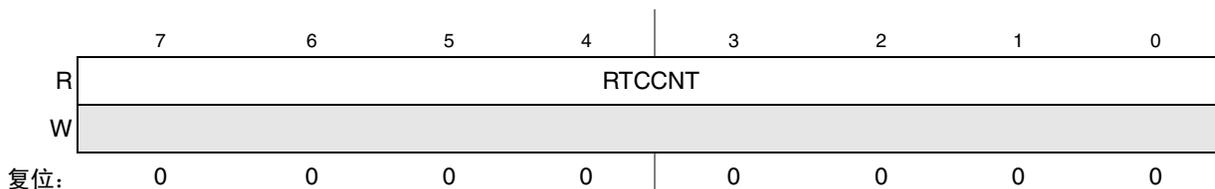


图 15-4. RTC 计数器寄存器 (RTCCNT)

表 15-4. RTCCNT 字段描述

字段	描述
7:0 RTCCNT	RTC 计数—这 8 个只读位包含 8 位计数器的当前值。写入操作对该寄存器无效。复位、写入 RTCMOD 或向 RTCLKS 和 RTCPS 写入不同值将把计数清除为 0x00。

15.3.3 RTC 模数寄存器 (RTCMOD)

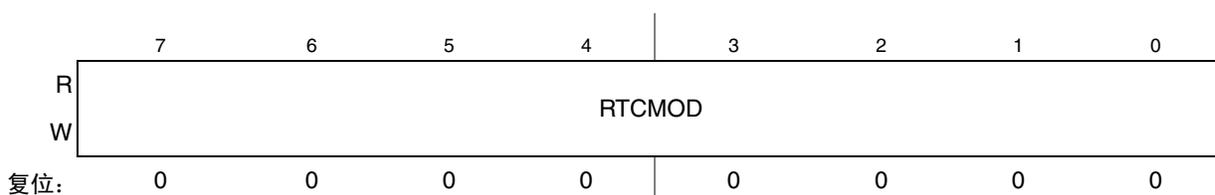


图 15-5. RTC 模数寄存器 (RTCMOD)

表 15-5. RTCMOD 字段描述

字段	描述
7:0 RTCMOD	RTC 模数 —这 8 个读 / 写位包含用于在比较匹配后将计数复位为 0x00 并设置 RTIF 状态位的模数值。0x00 值会在预分频器输出的每个上升边沿设置 RTIF 位。写入 RTCMOD 将把预分频器和 RTCCNT 计数器复位为 0x00。复位将模数寄存器设置为 0x00。

15.4 功能描述

RTC 由一个带有 8 位模数寄存器的主 8 位向上计数器、一个时钟源选择器和一个带有二进制和十进制可选值的预分频器组件组成。该模块还包含软件可选的中断逻辑。

任何 MCU 复位后，计数器被停止并复位为 0x00；模数寄存器被设置为 0x00，而且预分频器将关闭。1-kHz 内部振荡器时钟被选择为默认时钟源。要启动预分频器，向预分频器选择位 (RTCPS) 写入零以外的任何值。

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。

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
W								
正常 复位	0	0	0	0	0	0	0	0
在激活 BDM 中复位	1	1	0	0	1	0	0	0

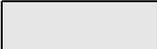
 = 未实施或预留

图 17-5. BDC 状态和控制寄存器 (BDCSCR)

表 17-2. BDCSCR 寄存器字段描述

字段	描述
7 ENBDM	激活 BDM (允许激活背景调试模式) — 一般而言, 这个位在调试开始后不久, 或只要调试主机复位目标, 由调试主机写为 1, 并保留 1, 直到通过正常的复位清除它。 0 BDM 不能激活 (非侵入式命令仍然被允许) 1 BDM 可以激活, 允许激活后台模式命令
6 BDMACT	背景调试模式激活状态 — 这是只读状态位。 0 BDM 未激活 (用户应用程序运行) 1 BDM 激活并等待串行命令
5 BKPTEN	BDC 断点激活 — 如果这个位清零, BDC 断点处于处活状态, FTS (强制标签选择) 控制位和 BDCBKPT 匹配寄存器被忽略。 0 BDC 断点禁止 1 BDC 断点激活
4 FTS	强制 / 标签选择 — 当 FTS = 1, 只要 CPU 地址总线匹配 BDCBKPT 匹配寄存器, 则请求断点。当 FTS = 0, CPU 地址总线与 BDCBKPT 寄存器之间的匹配会造成获取的操作码被标记。如果标记的操作码到达指令队列的末端, CPU 则进入激活后台模式, 而不是执行标记的操作码。 0 在断点地址标记操作码, 如果 CPU 试图执行该指令, 则进入激活后台模式 1 断点匹配强制在下一个指令边界进入激活后台模式 (地址不必是操作码)
3 CLKSW	选择 BDC 通信时钟的源 — CLKSW 默认 0, 选择其它 BDC 时钟源。 0 其它 BDC 时钟源 1 MCU 总线时钟表
2 WS	等待或停止状态 — 当目标 CPU 处于等待或停止状态时, 大多数 BDC 命令不起作用。但是可以用后台命令来强制目标 CPU 从等待或停止状态进入激活后台模式, 这样所有 BDC 命令都可以起作用。只要主机强制目标 MCU 进入激活背景调试模式, 主机应该发出 READ_STATUS 命令, 在尝试其它 BDC 命令前, 检查 BDMACT = 1。 0 目标 CPU 运行用户应用代码, 或处于激活背景调试模式 (当后台激活时, 它不处于等待或停止模式) 1 目标 CPU 处于等待或停止模式, 或者后台命令用来将其从等待或停止状态改变为激活背景调试模式
1 WSF	等待或停止失败状态 — 如果这存储器存取命令因目标 CPU 在大约相同时间执行等待或停止指令而失败, 则设置这个状态位。通常的恢复策略是发出后台命令, 从等待或停止模式进入激活后台模式, 重复失败的命令, 然后返回到用户程序。(一般地, 主机应该恢复 CPU 寄存器, 准备值, 重新执行等待或停止指令。) 0 存储器存取与等待或停止指令不冲突 1 存储器存取命令失败, 因为 CPU 已进入等待或停止模式
0 DVF	数据有效失败状态 — 这个状态位没有在 MC9S08DZ60 系列中使用, 因为它没有慢存取存储器。 0 存储器存取与慢存储器接入不冲突 1 存储器存取命令失败, 因为 CPU 没有完成慢存储器接入

17.4.3.3 调试比较器 B 高寄存器 (DBGCBH)

这个寄存器包含比较器 B 的高 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

17.4.3.4 调试比较器 B 低寄存器 (DBGCBL)

这个寄存器包含比较器 B 的低 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

17.4.3.5 调试 FIFO 高寄存器 (DBGFH)

这个寄存器提供对 FIFO 的高 8 位的只读接入。写到这个寄存器没有意义或无效果。在纯事件触发模式中，FIFO 只将数据存储在每个 FIFO 字的低字节，因此这个寄存器不能使用，将读 0x00。

读 DBGFH 不会导致 FIFO 移动到下一个字。当从 FIFO 中读出 16- 位字时，在读 DBGFL 前先读 DBGFH，因为读 DBGFL 会导致 FIFO 先于下个字的信息。

17.4.3.6 调试 FIFO 低寄存器 (DBGFL)

这个寄存器提供对 FIFO 的低 8 位的只读存取。写到这个寄存器没有意义或无效果。

读 DBGFL 会导致 FIFO 移动到下一个字的信息。当调试模块以纯事件模式运行时，只有 8- 位数据存储在 FIFO（每个 FIFO 字的高字节部分没有使用）。当从 FIFO 中读出 8- 位字时，只需重复地读 DBGFL，从 FIFO 中获得数据的连续的字节。在这种情况下，没有必要读 DBGFH。

当 FIFO 仍然打开时（打开后，但 FIFO 充满或 ARMF 被清除前）不要试图从其中读数据，因为在 DBGFL 读取过程中，FIFO 不能进一步操作。这可以干扰正常的 FIFO 的读取顺序。

在调试器没有打开的情况下读会使最近获取的操作码的地址存储到 FIFO 中的最后的位置。读取 DBGFL，然后定期 DBGFL，外部主机软件可以开发程序执行的概况。在对 FIFO 进行八次读取后，第九次读取将返回第一次读取结果的信息。要使用分析功能，则需要读取 FIFO 八次，且不使用启动顺序的数据，然后开始使用数据来获取已执行地址的延迟概貌。存储在 FIFO 中的关于 DBGFL（且 FIFO 没有打开）读取的信息就是最近所获操作码的地址。

17.4.3.7 调试控制寄存器 (DBGC)

这个寄存器可以在任何时间读或写。



图 17-7. 调试控制寄存器 (DBGC)

表 17-4. DBGC 寄存器字段描述

字段	描述
7 DBGEN	调试模块启用 — 用来启用调试模块。DBGEN 不能设置为 1，如果 MCU 是安全的。 0 DBG 禁用 1 DBG 启用
6 ARM	打开控制 — 控制调试器是否在 FIFO 中比较和存储信息。采用写操作来设置该位 (和 ARMF)，完成调试运行就是自动清除它。将 ARM 或 DBGEN 写为 0，可以停止任何调试运行。 0 调试器没有打开 1 调试器被打开
5 TAG	标记 / 强制选择 — 控制送到 CPU 的中断请求是否为标签或强制型请求。如果 BRKEN = 0，这个位就没有意义或无效。 0 CPU 中断请求作为强制型请求 1 CPU 中断请求作为标签型请求
4 BRKEN	中断启用 — 控制触发事件是否向 CPU 生成中断请求。触发事件可以使信息存储在 FIFO 中而不必向 CP 生成中断请求。对于结束跟踪，如果比较器 (s) 和 R/W 满足触发条件，则发出 CPU 中断请求。对于起始跟踪，则当 FIFO 满时发出 CPU 中断请求。TRGSEL 不影响 CPU 中断请求的定时。 0 CPU 中断请求未启用 1 触发器触发向 CPU 发出中断请求
3 RWA	比较器 A 的 R/W 比较值 — 当 RWAEN = 1，这个位确定是否用读或写接入来鉴定比较器 A，当 RWAEN = 0，RWA 和 R/W 信号不影响比较器 A。 0 比较器 A 只在写周期上匹配 1 比较器 A 只在读周期上匹配
2 RWAEN	启用比较器 A 的 R/W — 控制比较器 A 的匹配是否考虑这个水平的 R/W。 0 R/W 未用在比较 A 中 1 R/W 用在比较 A 中
1 RWB	比较器 B 的 R/W 比较值 — 当 RWBEN = 1，这个位确定是否用读或写接入来鉴定比较器 B。当 RWBEN = 0，RWA 和 R/W 信号不影响比较器 B。 0 比较器 B 只在写周期上匹配 1 比较器 B 只在读周期上匹配
0 RWBEN	启用比较器 B 的 R/W B — 控制比较器 B 的匹配是否考虑这个水平的 R/W。 0 R/W 未用在比较 B 中 1 R/W 用在比较 B 中

附录 B

定时器脉宽调制器 (TPMV2)

注意

本章节参考 S08TPM 第二版本，它适用于该器件的 3M05C 及更旧的掩膜版本。0M74K 和更新的掩膜版本采用 S08TPM 第三版本。如果你的器件采用 0M74K 或更新的掩膜，请参见 299 页上的第 16 章，“定时器脉冲宽度调节器（S08TPMV3）”，了解该模块的信息。

B.1 介绍

TPM 采用每通道一个 I/O 管脚，TPMxCHn，其中 x 是 TPM 数量（如 1 或 2），n 是通道的数量（如 0 - 4）。TPM 与通用 I/O 端口管脚共享其 I/O 管脚（参见 Pins and Connections 章节，了解更多信息）。

B.2 特性

TPM 提供以下特性：

- 每个 TPM 可以配置为所有通道上缓冲的且中央对齐的脉宽调制 (CPWM)
- 可以为每个 TPM 独立选择时钟源（多个 TPM 器件）
- 时钟源可选择（根据器件选择）：总线时钟、固定系统时钟、外部管脚
- 时钟预分频点按 1, 2, 4, 8, 16, 32, 64, 或 128 分
- 16- 位自由运行或上 / 下 (CPWM) 计数操作
- 16- 位模量寄存器控制计数器范围
- 定时器系统使能
- 每个通道一个中断加上每个 TPM 模块的终端计数中断（多个 TPM 器件）
- 通道特性：
 - 每个通道可以是输入捕捉，输出比较，或缓冲边沿对齐的 PWM
 - 上升边，下降边或任意边输入捕捉触发器
 - 设置、清除或固定输出比较行动
 - 在 PWM 输出上可选择极性