



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	25
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 10x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dz32avlc

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
0x1845	PTAPS	PTAPS7	PTAPS6	PTAPS5	PTAPS4	PTAPS3	PTAPS2	PTAPS1	PTAPS0
0x1846	PTAES	PTAES7	PTAES6	PTAES5	PTAES4	PTAES3	PTAES2	PTAES1	PTAES0
0x1847	预留	—	—	—	—	—	—	—	—
0x1848	PTBPE	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0x1849	PTBSE	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
0x184A	PTBDS	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
0x184B	预留	—	—	—	—	—	—	—	—
0x184C	PTBSC	0	0	0	0	PTBIF	PTBACK	PTBIE	PTBMOD
0x184D	PTBPS	PTBPS7	PTBPS6	PTBPS5	PTBPS4	PTBPS3	PTBPS2	PTBPS1	PTBPS0
0x184E	PTBES	PTBES7	PTBES6	PTBES5	PTBES4	PTBES3	PTBES2	PTBES1	PTBES0
0x184F	预留	—	—	—	—	—	—	—	—
0x1850	PTCPE	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0x1851	PTCSE	PTCSE7	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
0x1852	PTCDS	PTCDS7	PTCDS6	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
0x1853– 0x1857	预留	—	—	—	—	—	—	—	—
0x1858	PTDPE	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
0x1859	PTDSE	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
0x185A	PTDDS	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
0x185B	预留	—	—	—	—	—	—	—	—
0x185C	PTDSC	0	0	0	0	PTDIF	PTDACK	PTDIE	PTDMOD
0x185D	PTDPS	PTDPS7	PTDPS6	PTDPS5	PTDPS4	PTDPS3	PTDPS2	PTDPS1	PTDPS0
0x185E	PTDES	PTDES7	PTDES6	PTDES5	PTDES4	PTDES3	PTDES2	PTDES1	PTDES0
0x185F	预留	—	—	—	—	—	—	—	—
0x1860	PTEPE	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
0x1861	PTESE	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
0x1862	PTEDS	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
0x1863– 0x1867	预留	—	—	—	—	—	—	—	—
0x1868	PTFPE	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
0x1869	PTFSE	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
0x186A	PTFDS	PTFDS7	PTFDS6	PTFDS5	PTFDS4	PTFDS3	PTFDS2	PTFDS1	PTFDS0
0x186B– 0x186F	预留	—	—	—	—	—	—	—	—
0x1870	PTGPE	0	0	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x1871	PTGSE	0	0	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x1872	PTGDS	0	0	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
0x1873– 0x187F	预留	—	—	—	—	—	—	—	—
0x1880	CANCTL0	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ

4.5.11.2 Flash 和 EEPROM 选项寄存器 (FOPT 和 NVOPT)

在复位过程中，非易失性位置 NVOPT 上的内容从 Flash 拷贝到 FOPT 中。若想修改这个寄存器中的值，可对 Flash 中的 NVOPT 位置进行擦除和重新编程，然后发出新的 MCU 复位命令。

	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	EPGMOD	0	0	0	SEC	
W								
复位	F	F	F	0	0	0	F	F

= 未实现或被预留
 F = 在复位期间从非易失性位置 NVOPT 上上载

图 4-6. Flash 和 EEPROM 选项寄存器 (FOPT)

表 4-9. FOPT 寄存器字段描述

字段	描述
7 KEYEN	后门密钥机制启动 — 该位设置为 0 时，后门密钥机制不能用于关闭安全性。后门密钥机制只能从用户（受保护）固件上访问。BDM 命令不能用于写入可能会解锁后门密钥的密钥对比值。若欲了解有关后门密钥机制的更详尽信息，请参见 4.5.9，“安全性”。 0 不允许后门密钥访问。 1 如果用户固件写入一个与非易失性后门密钥（按顺序为 NVBACKKEY 到 NVBACKKEY+7）相匹配的 8 字节值，安全性在下一次 MCU 复位前会暂时关闭。
6 FNORED	向量重定向禁用 — 该位为 1 时向量重定向被禁用。 0 向量重定向启用。 1 向量重定向禁用。
5 EPGMOD	EEPROM 分区模式 — 该位为 0 时，每个分区分为两个页面（4 字节模式）。该位为 1 时，每个分区在一个页面中（8 字节模式）。 0 每个 EEPROM 分区的一半在页面 0 中而另一半在页面 1 中。 1 每个分区在一个页面中。
1:0 SEC	安全状态代码 — 这个 2 位字段决定 MCU 的安全状态，如表 4-10 所示。MCU 处于安全状态时，RAM、EEPROM 和 Flash 中的内容不能通过指令从不安全的源（包括后台调试接口）上访问。后门密钥被成功输入或对 Flash 进行了成功的空白检查后，SEC 将变为 1:0。若欲了解有关安全性的更详尽信息，请参见 4.5.9，“安全性”。

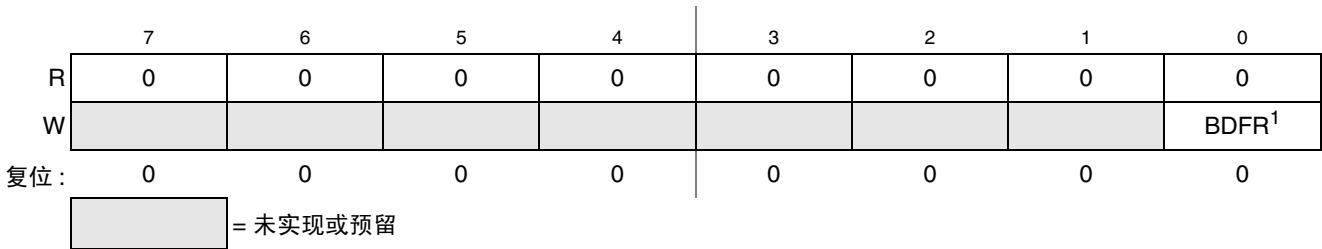
表 4-10. Security States¹

SEC[1:0]	描述
0:0	安全
0:1	安全
1:0	不安全
1:1	安全

¹ 后门密钥被成功输入或成功地对 Flash 进行了空白检查后，SEC 将变为 1:0。

5.8.3 系统后台调试强制复位寄存器 (SBDFR)

这个高页寄存器只包括一个只写控制位。串行后台命令，如 `WRITE_BYTE` 必须用来写入 SBDFR。从用户程序写入寄存器的尝试被忽略。读总是返回 0x00。



¹ 只能通过串行后台调试命令，而非用户程序写入 BDFR。

图 5-4. 后台调试强制复位寄存器 (SBDFR)

表 5-4. SBDFR 寄存器字段描述

字段	描述
0 BDFR	后台调试强制复位 — 可以使用串行后台命令，如 <code>WRITE_BYTE</code> ，使外部调试主机强制进行目标系统复位。在该位中写入 1 就能强制进行 MCU 复位。该位不能从用户程序中写入。

5.8.4 系统选项寄存器 1 (SOPT1)

该高页寄存器是 `write-once` 寄存器，因此只重视复位后的第一次写入。它可以在任何时候读取。任何后续 `SOPT1` 写入尝试（有意或无意）都将被忽略，以避免对这些敏感器件的意外修改。该寄存器应在用户复位初始化程序期间写入，以设置期望的控制，即便期望的设置与复位设置相同。

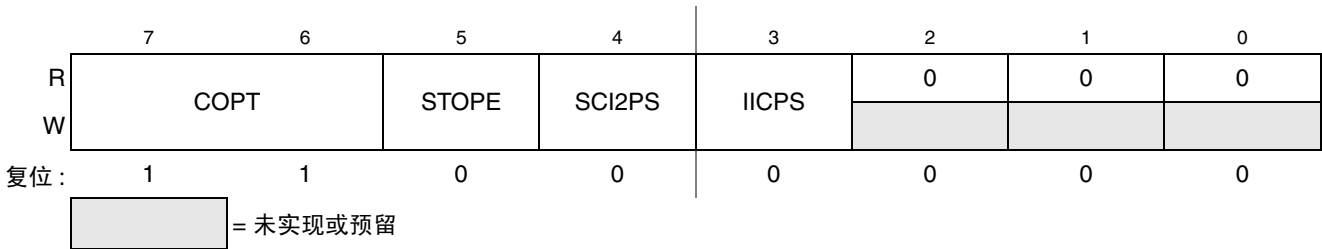


图 5-5. 系统选项寄存器 1 (SOPT1)

表 5-5. SOPT1 寄存器字段描述

字段	描述
7:6 COPT[1:0]	COP 看门狗超时 — 这些单次写入有效的位选择 COP 的超时周期。STOP2 中的 COPT 和 COPCLKS 定义 COP 超时周期。参见表 5-6。
5 STOPE	停止模式使能 — 这个单次写入有效的位用来使能停止模式。如果停止模式禁止且用户程序试图实施 STOP 指令，则会强制进行非法操作码复位。 0 停止模式禁止。 1 停止模式使能。

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。

6.5.1.1 A 端口数据寄存器 (PTAD)

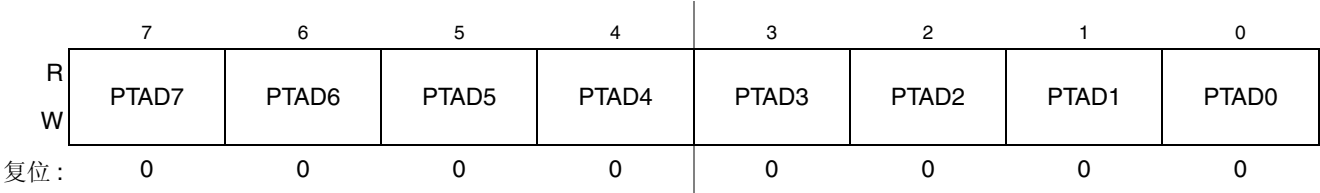


图 6-3. A 端口数据寄存器 (PTAD)

表 6-1. PTAD 寄存器字段描述

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

6.5.1.2 A 端口数据方向寄存器 (PTADD)

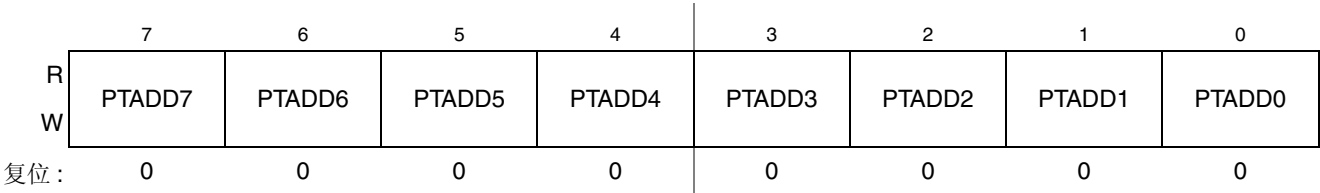


图 6-4. 端口数据方向寄存器 (PTADD)

表 6-2. PTADD 寄存器字段描述

字段	描述
7:0 PTADD[7:0]	A 端口位的数据方向 — 这些读 / 写位控制着 A 端口管脚的方向以及为 PTAD 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 A 端口位 - 输出驱动使能，PTAD 读数返回 PTADn 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。

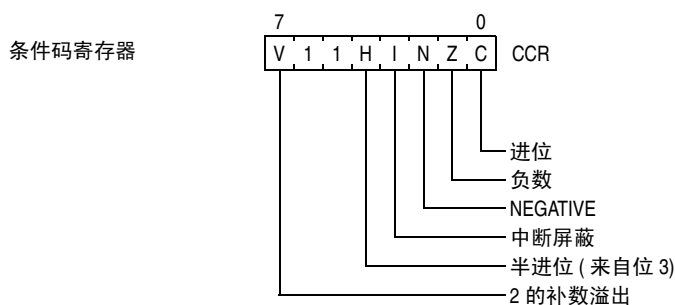


图 7-2. 条件码寄存器

表 7-2. 指令集小结 (第 6 页, 共 9 页)

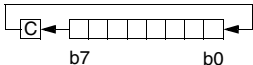
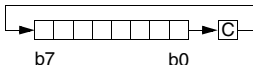
Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	移动 (M) _{destination} ← (M) _{source} 在 IX+/DIR 和 DIR/IX+ 模式, H:X ← (H:X) + \$0001	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rpwpp rfwpp pwpp rfwpp	0 1 1 -	-↑↑↓-
MUL	不带符号的乘法 X:A ← (X) × (A)	INH	42	5	fffffp	- 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	rfwpp p p rfwpp rfwp prfwpp	↑ 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	rfwpp p p rfwpp rfwp prfwpp	↑ 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	rfwpp p p rfwpp rfwp prfwpp	↑ 1 1 -	-↑↑↑↑

表 7-2. 指令集小结 (第 8 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
SUB #opr8i SUB opr8a SUB opr16a SUB oprx16,X SUB oprx8,X SUB ,X SUB oprx16,SP SUB oprx8,SP	减 $A \leftarrow (A) - (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 ii B0 dd C0 hh ll D0 ee ff E0 ff F0 9E D0 ee ff 9E E0 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↑ 1 1 -	- ↓ ↓ ↓
SWI	软件中断 $PC \leftarrow (PC) + \$0001$ 推 (PCL); $SP \leftarrow (SP) - \$0001$ 推 (PCH); $SP \leftarrow (SP) - \$0001$ 推 (X); $SP \leftarrow (SP) - \$0001$ 推 (A); $SP \leftarrow (SP) - \$0001$ 推 (CCR); $SP \leftarrow (SP) - \$0001$ $I \leftarrow 1$; PCH 中断向量高字节 PCL 中断向量低字节	INH	83	11	sssssvvfppp	- 1 1 -	1 - - -
TAP	将累加器转移到 CCR $CCR \leftarrow (A)$	INH	84	1	p	↑ 1 1 ↑	↑ ↓ ↓ ↓
TAX	将累加器转移到 X (索引寄存器低) $X \leftarrow (A)$	INH	97	1	p	- 1 1 -	- - - -
TPA	将 CCR 转移到累加器 $A \leftarrow (CCR)$	INH	85	1	p	- 1 1 -	- - - -
TST opr8a TSTA TSTX TST oprx8,X TST ,X TST oprx8,SP	负数或 0(M) 测试 $(M) - \$00$ $(A) - \$00$ $(X) - \$00$ $(M) - \$00$ $(M) - \$00$ $(M) - \$00$	DIR INH INH IX1 IX SP1	3D dd 4D 5D 6D ff 7D 9E 6D ff	4 1 1 4 3 5	rfpp p p rfpp rfp prfpp	0 1 1 -	- ↓ ↓ -
TSX	将 SP 转移到索引寄存器。 $H:X \leftarrow (SP) + \$0001$	INH	95	2	fp	- 1 1 -	- - - -
TXA	将 X (索引寄存器低) 转移到累加器上 $A \leftarrow (X)$	INH	9F	1	p	- 1 1 -	- - - -

8.5.1.4 FLL Bypassed External (FBE)

在 FLL Bypassed External (FBE) 模式中, MCGOUT 时钟来自外部参考时钟, 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 External 模式:

- CLKS 位写入 10
- IREFS 位写入 0
- PLLS 位写入 0
- 位写入介于 31.25 kHz- 39.0625 kHz 频率范围内的分频参考时钟。
- LP 位写入 0

在 FLL Bypassed External 模式中, MCGOUT 时钟源自 FLL 时钟。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 时钟由外部参考时钟控制, FLL 时钟频率是由 RDIV 位选择的参考频率的 1024 倍。MCGCLK 来自 FLL, PLL 被禁止处于低功率状态。

注意

可以用大于指定最大频率的 FLL 参考时钟频率在 FBE 模式中短时间运行。这在使用频率大于 5 MHz 的外部晶体运行于 PEE 模式中是必需的。如需了解详细的示例信息, 请参见 8.6.2.4, “示例 4: 从 FEI 转换到 PEE 模式: 外部晶体 = 8 MHz、总线频率 = 8 MHz”。

8.5.1.5 PLL Engaged External (PEE)

当满足以下条件时就进入 PLL Engaged External (PEE) 模式:

- CLKS 位写入 00
- IREFS 位写入 0
- PLLS 位写入 1
- RDIV 位写入介于 1 MHz - 2 MHz 频率范围内的分频参考时钟。

在 PLL Engaged External 模式中, MCGOUT 时钟源自 PLL 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。PLL 时钟频率是参考频率 (RDIV 位所选) 和倍频因子 (VDIV 位所选) 乘积。如果使能 BDM, MCGCLK 值就是 DCO 除以 2 (开放环路模式) 的得数。如果禁止 BDM, 那么 FLL 被禁止且处于低功率状态。

8.5.1.6 PLL Bypassed External (PBE)

在 PLL Bypassed External (PBE) 模式中, MCGOUT 时钟源自外部参考时钟, PLL 处于运行状态但其输出时钟未使用。该模式对允许 PLL 获得目标频率非常有用, 同时 MCGOUT 时钟由内部参考时钟驱动。

表 10-11. APCTL3 寄存器字段描述

字段	描述
7 ADPC23	ADC 管脚控制 23 — ADPC23 用来控制与通道 AD23 连接的管脚。 0 AD23 管脚 I/O 控制使能 1 AD23 管脚 I/O 控制禁止
6 ADPC22	ADC 管脚控制 22 — ADPC22 用来控制与通道 AD22 连接的管脚。 0 AD22 管脚 I/O 控制使能 1 AD22 管脚 I/O 控制禁止
5 ADPC21	ADC 管脚控制 21 — ADPC21 用来控制与通道 AD21 连接的管脚。 0 AD21 管脚 I/O 控制使能 1 AD21 管脚 I/O 控制禁止
4 ADPC20	ADC 管脚控制 20 — ADPC20 用来控制与通道 AD20 连接的管脚。 0 AD20 管脚 I/O 控制使能 1 AD20 管脚 I/O 控制禁止
3 ADPC19	ADC 管脚控制 19 — ADPC19 用来控制与通道 AD19 连接的管脚。 0 AD19 管脚 I/O 控制使能 1 AD19 管脚 I/O 控制禁止
2 ADPC18	ADC 管脚控制 18 — ADPC18 用来控制与通道 AD18 连接的管脚。 0 AD18 管脚 I/O 控制使能 1 AD18 管脚 I/O 控制禁止
1 ADPC17	ADC 管脚控制 17 — ADPC17 用来控制与通道 AD17 连接的管脚。 0 AD17 管脚 I/O 控制使能 1 AD17 管脚 I/O 控制禁止
0 ADPC16	ADC 管脚控制 16 — ADPC16 用来控制与通道 AD16 连接的管脚。 0 AD16 管脚 I/O 控制使能 1 AD16 管脚 I/O 控制禁止

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。

12.5.3 标识符接收滤波器

MSCAN 标识符接收寄存器（参见 12.3.11，“MSCAN 标识符验收控制寄存器 (CANIDAC)”）定义标准或扩展标识符（ID[10:0] 或 ID[28:0]）的可接受模式。这些位中的任意一个都可以在 MSCAN 标识符掩码寄存器中标志为“不比较”（参见 12.3.16，“MSCAN 标识符掩码寄存器 (CANIDMR0-CANIDMR7)”）。

一次滤波器匹配可由接收缓冲器已满标志（RXF = 1）和 CANIDAC 寄存器中的 3 个位（参见 12.3.11，“MSCAN 标识符验收控制寄存器 (CANIDAC)”）。通知给应用软件。这些标识符匹配标志（IDHIT[2:0]）能够清晰识别引起接收的滤波寄存器。它们简化了应用软件处理接收器中断来源的任务。如果出现一次以上的匹配（两个或多个滤波器匹配），低地址的寄存器具有优先权。

非常灵活的可编程通用标识符接收滤波器可以有效降低 CPU 的中断负载，该滤波器在经过编程后可在四种不同模式中运行（Bosch CAN 2.0A/B 协议规范）：

- 两个标识符接收滤波器，每个将应用于：
 - 扩展标识符的全部 29 位和 CAN2.0B 帧的以下位：
 - 远程发送请求 (RTR)
 - 标识符扩展 (IDE)
 - 替代远程请求 (SRR)
 - 标准标识符的 11 位，加上 CAN 2.0A/B 报文的 RTR 和 IDE 位¹。这种模式为符合 CAN 2.0B 标准的长扩展标识符提供两个滤波器。图 12-39 显示第一个 32 位滤波器页（CANIDAR0 - CANIDAR3、CANIDMR0 - CANIDMR3）如何产生滤波器 0 匹配。同样，第二个滤波器页（CANIDAR4 - CANIDAR7、CANIDMR4 - CANIDMR7）产生滤波器 1 匹配。
- 4 个标识符接收滤波器，每个应用于：
 - a) 扩展标识符的 14 个最重要位，加上 CAN 2.0B 报文的 SRR 和 IDE 位，或
 - b) 标准标识符的 11 位、CAN 2.0A/B 报文的 RTR 和 IDE 位。图 12-40 显示第一个 32 位滤波器页（CANIDAR0 - CANIDAR3、CANIDMR0 - 3CANIDMR）如何产生滤波器 0 和 1 匹配。同样，第二个滤波器页（CANIDAR4 - CANIDAR7、CANIDMR4 - CANIDMR7）产生滤波器 2 和 3 匹配。
- 8 个标识符接收滤波器，每个应用于标识符的前 8 位。这种模式为符合 CAN 2.0A/B 的标准标识符或符合 CAN 2.0B 的扩展标识符的前 8 个位实施 8 个独立的滤波器。图 12-41 显示第一个 32 位滤波器页（CANIDAR0 - CANIDAR3、CANIDMR0 - CANIDMR3）如何产生滤波器 0-3 匹配。同样，第二个滤波器页（CANIDAR4 - CANIDAR7、CANIDMR4 - CANIDMR7）产生滤波器 4-7 匹配。
- 关闭滤波器。没有 CAN 报文被复制到前景缓冲器 RxFG，且从不设置 RXF 标志。

1. 尽管这种模式可以用于标准标识符，但我们还是建议为标准标识符使用 4 个或 8 个标识符接收滤波器。

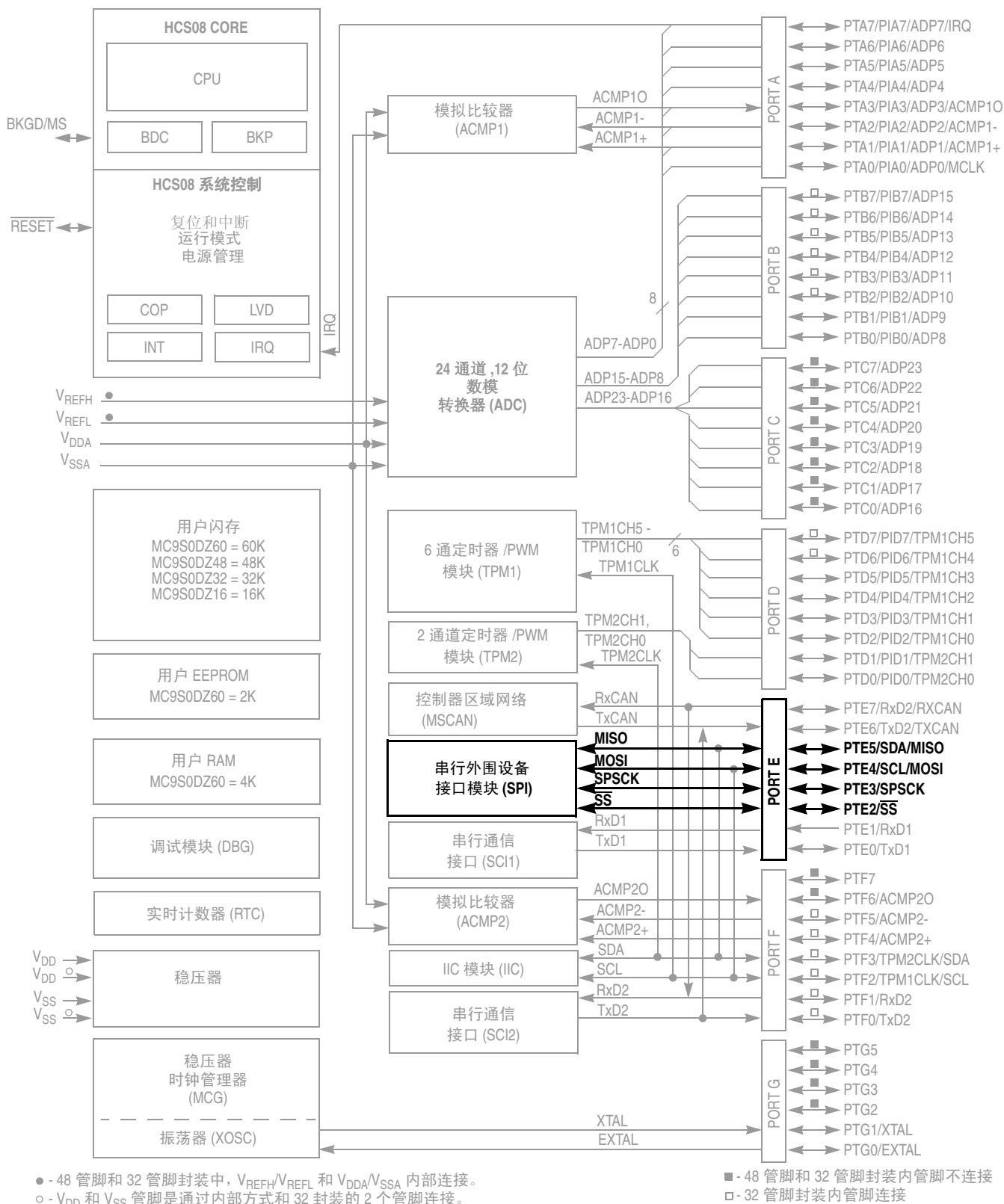


图 13-1. 强调 ADC 模块和管脚的 MC9S08DZ60 结构图

表 14-7. SCIXS2 字段描述

字段	描述
7 LBKDIF	中止符检测中断标记 — 当使能了 LIN 中止符检测电路且检测到 LIN 中止字符时，就设置 LBKDIF。将“1”写入其中可以清除 LBKDIF。 0 未检测到 LIN 中止字符。 1 检测到 LIN 中止字符。
6 RXEDGIF	RxD 管脚活动边沿中断标记 — 当 RxD 管脚上出现活动边沿（如果 RXINV = 0，下降；如果 RXINV = 1，上升）时，就设置 RXEDGIF。将“1”写入其中清除 RXEDGIF。 0 接收管脚上未出现活动边沿。 1 接收管脚上出现活动边沿。
4 RXINV ¹	接收数据颠倒 — 设置该位反转已接收数据输入的极性。 0 接收数据未被反转 1 接收数据被反转
3 RWUID	接收唤醒闲置检测 — RWUID 控制着唤醒接收器的闲置字符是否设置 IDLE 位。 0 在接收待机状态（RWU = 1）期间，检测到闲置字符时不设置 IDLE 位。 1 在接收待机状态（RWU = 1）期间，检测到闲置字符时设置 IDLE 位。
2 BRK13	中止字符生成长度 — BRK13 用于选择较长的发送中止字符长度。成帧错误的检测不受该位状态的影响。 0 中止字符用 10 位时间（如果 M = 1，则是 11 位时间）长度发送 1 中止字符用 13 位时间（如果 M = 1，则是 14 位时间）长度发送
1 LBKDE	LIN 中止字符检测使能 — LBKDE 用来选择较长中止字符检测长度。当设置了 LBKDE 时，防止设置成帧错误（FE）和接收数据寄存器已满（RDRF）标记。 0 中止字符在 10 位时间（如果 M = 1，则是 11 位时间）长度上检测。 1 中止字符在 11 位时间（如果 M = 1，则是 12 位时间）长度上检测。
0 RAF	接收器活动标记 — 当 SCI 接收器检测到有效起始位开始时，设置 RAF，并且当接收器检测到闲置线路时，RAF 被自动清除。这种状态标记可以用来检查在引导 MCU 进入停止模式前，是否正在接收 SCI 字符。 0 SCI 接收器闲置，正在等待起始位。 1 SCI 接收器活动（RxD 输入不闲置）。

¹ 设置 RXINV 会反转所有情况下的 RxD 输入：数据位、起始位和停止位、中止字符和闲置。

当在 LIN 系统中使用内部振荡器时，需要把中止符检测阈值提高 1 个位时间。在 LIN 中所允许的最坏计时情况下，0x00 数据字符在运行速度比主器件快 14% 的辅器件上的长度可能达到 10.26 位时间。这将触发旨在检测 10 位中断符号的常规中止符检测电路。当设置了 LBKDE 位时，成帧错误被禁止，中止符检测阈值从 10 位变为 11 位，从而防止把 0x00 数据字符错误检测为一个 LIN 的中止符。



图 16-9. TPM 计数器寄存器低字节 (TPMxCNTL)

当 BDM 处于有效状态，定时器计数器被冻结（这是用户将读取的值）；一致性机制是冻结的，这样当 BDM 处于有效状态时，缓冲器中锁定的内容仍将保持锁定状态，即使计数器的一半或两个一半都在 BDM 处于有效状态时被读取。这确保了如果 BDM 处于有效状态时用户正在读取 16 位寄存器，那么返回到正常操作后将从 16 位值的另一半中读取适当值。

在 BDM 模式下，向 TPMxSC, TPMxCNTH 或 TPMxCNTL 寄存器写入任何值可复位 TPMxCNTH:L 寄存器的读取一致性机制而不受写入涉及的数据影响。

16.3.3 TPM 计数器模数寄存器 (TPMxMODH:TPMxMODL)

这个读 / 写 TPM 模数寄存器包含 TPM 计数器的模数值。TPM 计数器达到模数值后，TPM 计数器在下一个时钟周期内又从 0x0000 开始计数，同时会设置一个溢出标记（TOF）。向 TPMxMODH 或 TPMxMODL 中写入一个值可禁止 TOF 位和溢出中断，直到另一个字节也被写入。复位操作将 TPM 计数器的模数寄存器设置为 0x0000，进而导致自由运行的定时器计数器（模数被关闭）。

写入任何一个字节（无论是 TPMxMODH 还是 TPMxMODL）都会使值锁入到缓冲器中，同时寄存器会根据 CLKSb:CLKSa bits, so 位的值以它们的写入缓冲器的值得到更新，因此：

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

锁定机制可通过向 TPMxSC 地址（无论 BDM 是否活动）中写入一个值来手动复位。

当 BDM 处于有效状态时，一致性机制是冻结的（除非通过写入 TPMxSC register 寄存器复位）。这样，当 BDM 处于有效状态时，缓冲器中锁定的内容仍然保持锁定状态，即使模数寄存器的一半或两个一半都在 BDM 处于有效状态时被写入。当 BDM 处于有效状态时，任何向模数寄存器的写入行为都会绕过缓冲器锁定并直接写入到模数寄存器中。

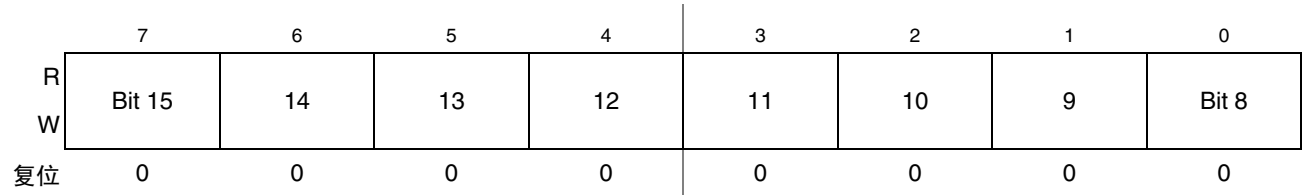


图 16-10. TPM 计数器模数寄存器高字节 (TPMxMODH)

16.4.2.3 边缘对齐 PWM 模式

这类 PWM 输出使用定时器计数器的正常向上计数模式 (CPWMS=0)；当相同 TPM 中的其他通道被配置用于输入捕捉或输出比较功能时，也可使用它。这个 PWM 信号的周期由模数寄存器 (TPMxMODH:TPMxMODL) 的值加 1 确定。占空比由定时器通道寄存器 (TPMxCnVH:TPMxCnVL) 中的设置确定。这个 PWM 信号的极性由 ELSnA 控制位中的设置确定。0% 和 100% 的占空比都是可能的。

TPM 通道寄存器中的输出比较值决定 PWM 信号的脉冲宽度 (占空比) (图 16-15)。T 模数溢出和输出比较间的时间间隔为脉冲宽度。如果 ELSnA=0，计数器溢出强迫 PWM 信号进入高态；而输出比较强制 PWM 信号进入低态。如果 ELSnA=1，计数器溢出强制 PWM 信号进入低态；而输出比较强制 PWM 信号进入高态。

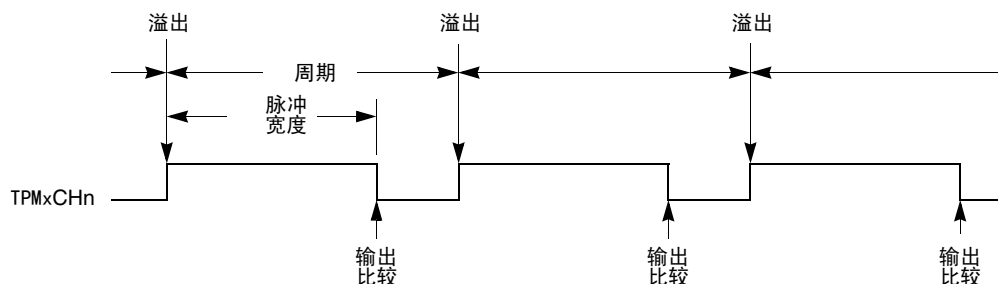


图 16-15. PWM 周期和脉冲宽度 (ELSnA=0)

当通道值寄存器被设为 0x0000 时，占空比为 0%。100%。通过将定时器通道计数器 (TPMxCnVH:TPMxCnVL) 设置为大于模数设置的值可实现 100% 的占空比。这意味着要实现 100% 的占空比，模数设置必须小于 0x FFFF。

因为 TPM 可用在 8 位 MCU 中，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 TPMxCnVH 和 TPMxCnVL 中意味着实际写入到缓冲器寄存器中。在边缘对齐 PWM 模式下，值根据 CLKSB:CLKSA 位的值被传输到相应的定时器通道寄存器中，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 0xffff 变为 0x0000 时进行更新。

16.4.2.4 中央对齐 PWM 模式

这类 PWM 输出使用定时器计数器 (CPWMS=1) 的向上 / 向下计数模式。TPMxCnVH:TPMxCnVL 中的输出比较值决定 PWM 信号的脉冲宽度 (占空比) 而 TPMxMODH:TPMxMODL 中的值决定周期。TPMxMODH:TPMxMODL 应保持在 0x0001 至 0x7FFF 之间，因为这一范围外的值可生成模糊结果。ELSnA 将决定 CPWM 输出的极性。

脉冲宽度 = 2 x (TPMxCnVH:TPMxCnVL)

周期 = 2 x (TPMxMODH:TPMxMODL)；TPMxMODH:TPMxMODL=0x0001-0x7FFF

如果通道值寄存器 TPMxCnVH:TPMxCnVL 为零或负数 (位 15 被设置)，占空比将为 0%。如果 TPMxCnVH:TPMxCnVL 是正值 (位 15 被清除) 并大于模数设置 (非零)，占空比将为

范围内 ($A \leq \text{地址} \leq B$) — 当地址大于或等于比较器 A 的值，且小于等于比较器 B 的值时，触发发生。

范围外 ($\text{地址} < A$ 或 $\text{地址} > B$) — 当地址小于比较器 A 的值，或大于比较器 B 的值时，触发发生。

17.3.6 硬件断点

DBG 寄存器中的 **BRKEN** 控制位可以设置为 1，来允许使用 17.3.5，“触发模式”所描述的任何触发条件，向 CPU 生成硬件断点请求。DBG 中的 **TAG** 控制断点请求是否处理为标记类断点或强制类断点。标记断点使当前的操作码进入指令队列时被标记。如果标记的操作码达到队列的末端，CPU 执行 **BGND** 指令，进入激活背景调试模式，而不是执行被标记的操作码。强制类断点使 CPU 完成当前指令，然后进入激活背景调试模式。

如果背景调试模式没有被通过 **BKGD** 管脚的串行 **WRITE_CONTROL** 命令激活 (**ENBDM** = 1)，CPU 将执行 **SWI** 指令，而不是进入激活背景调试模式。

17.4 寄存器定义

本小节描述了 **BDC** 和 **DBG** 寄存器及控制位。

参见本文的器件概述章节中的 **high-page** 寄存器一览，了解所有 **DBG** 寄存器的绝对地址分配。本小节只按名字参考了寄存器和控制位。使用飞思卡尔提供的等式或头文件，将这些名称翻译为相应的绝对地址。

17.4.1 BDC 寄存器和控制位

BDC 有两个寄存器：

- 状态和控制寄存器 (**BDCSCR**) 是一个包含背景调试控制器控制和状态位的 8- 位寄存器。
- **BDC** 断点匹配寄存器 (**BDCBKPT**) 拥有一个 16- 位断点匹配地址。

这些寄存器通过专门的串行 **BDC** 命令接入，没有位于目标 MCU 的存储器空间中 (因此，它们没有地址，用户程序不能接入)。

BDCSCR 中的一些位有写限制，否则这些寄存器可以随时被读或写。例如，当 MCU 处于激活背景调试模式中时，**ENBDM** 控制位不能被写。(这防止了在 MCU 已经处于激活后台模式时，禁止激活后台模式的控制位的模糊条件) 而且，有四个状态位 (**BDMACT**, **WS**, **WSF**, 和 **DVF**) 是只读状态指示符，永远也不能被 **WRITE_CONTROL** 串行 **BDC** 命令写。时钟开关 (**CLKSW**) 控制位随时都可读或写。

17.4.1.1 BDC 状态和控制寄存器 (BDCSCR)

这个寄存器可以被串行 **BDC** 命令 (**READ_STATUS** 和 **WRITE_CONTROL**) 读或写，但是用户程序不能存取它，因为它不位于 MCU 的正常的存储器映射空间中。

表 17-6. DBGS 寄存器字段描述

字段	描述
7 AF	触发匹配 A 标记 — 在调试运行开始时清除 AF，指示武装后是否满足触发 匹配 A 条件。 0 比较器 A 未匹配 1 比较器 A 匹配
6 BF	触发匹配 B 标记 — 在调试运行开始时清除 BF，指示武装后是否满足触发 匹配 B 条件。 0 比较器 B 未匹配 1 比较器 B 匹配
5 ARMF	打开标记 — 当 DBGEN=1 时，这个位为 DBGCR 中 ARM 的只读镜像。将 DBGCR 中的 ARM 控制位写为 1（当 DBGEN = 1）可设置该位，在调试运行结束时自动清除它。当 FIFO 为满时（始起跟踪），或当探测到触发事件时（结束跟踪），调度运行完成。将 DBGCR 中的 ARM 或 DBGEN 写为 0，可以人工停止调试运行。 0 调试器没有打开 1 调试器被打开
3:0 CNT[3:0]	FIFO 有效计数 — 这些位在调试运行开始时清除，指示调试运行结束时 FIFO 中的有效数据的字数。当数据从 FIFO 中读出时，CNT 中的值不减少。当信息从 FIFO 中读出时，外部调试主机负责计数的跟踪。 0000 FIFO 中的有效字数 = 无有效数据 0001 FIFO 中的有效字数 = 1 0010 FIFO 中的有效字数 = 2 0011 FIFO 中的有效字数 = 3 0100 FIFO 中的有效字数 = 4 0101 FIFO 中的有效字数 = 5 0110 FIFO 中的有效字数 = 6 0111 FIFO 中的有效字数 = 7 1000 FIFO 中的有效字数 = 8

平均芯片接面温度 (T_J) (单位 $^{\circ}\text{C}$) 可以用如下等式计算:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad \text{等式 A-1}$$

其中:

T_A = 周围温度, $^{\circ}\text{C}$

θ_{JA} = 封装热阻, 结到环境, $^{\circ}\text{C/W}$

$P_D = P_{\text{int}} + P_{\text{I/O}}$

$P_{\text{int}} = I_{\text{DD}} \times V_{\text{DD}}$, 瓦特 — 芯片内部功率

$P_{\text{I/O}}$ = 输入和输出管脚上的功耗 — 由用户决定

对大多数应用来说, $P_{\text{I/O}} \ll P_{\text{int}}$, 可以忽略不计。 P_D 和 T_J 间的近似关系 (如果 $P_{\text{I/O}}$ 忽略不计) 是:

$$P_D = K \div (T_J + 273^{\circ}\text{C}) \quad \text{等式 A-2}$$

将等式 1 代入和等式 2, 求出 K 为:

$$K = P_D \times (T_A + 273^{\circ}\text{C}) + \theta_{JA} \times (P_D)^2 \quad \text{等式 A-3}$$

其中, K 是一个与特殊部件有关的常量。K 可以通过测量已知 T_A 的 P_D (平衡时), 从等式 3 计算出来。如果 K 值已知, 用等式 1 和等式 2 相式代入, 就可以得出任意 T_A 值的 P_D 和 T_J 。

A.5 ESD 保护和抗闭锁方法

尽管这些器件上的静电放电 (ESD) 损害要比早期的 CMOS 电路的 ESD 要小得多, 但仍应采取一些正常的处理防范措施, 防止静电。要执行一些鉴定测试, 以确保这些器件可以忍受合理水平的静电, 而不会造成任何永久的损害。

整个 ESD 测试符合 AEC-Q100 汽车集成电路的应力测试鉴定。在进行器件鉴定过程中, 要执行人体放电模式 (HBM) 和充电器件模式 (CDM) 的 ESD 应力测试。

如果暴露给 ESD 脉冲后, 器件不再符合器件规范, 那么就认定器件测试失败。在进行完高温测试后, 还要在室温下根据每个可适用的器件规范进行完整的 DC 参数及功能测试, 除非设备规范中另有指定。

表 A-4. ESD 和闭锁测试条件

模式	描述	符号	值	单位
人体	串联电阻	R1	1500	W
	存储电容	C	100	pF
	每管脚脉冲数	—	3	
闭锁	最小输入电压限制		-2.5	V
	最大输入电压限制		7.5	V

表 B-2. TPM 时钟源选择

CLKSB:CLKSA	连接预分频器输入的 TPM 时钟源
0:0	未选中任何时钟 (TPM _x 禁止)
0:1	线速率时钟 (BUSCLK)
1:0	固定系统时钟 (XCLK)
1:1	外部源 (TPM _x CLK) ^{1,2}

- ¹ 外部时钟的最大允许频率是总线频率的 1/4。
- ² 如果外部时钟输入共享通道 n，并且选用未 TPM 时钟源，对应的 ELSnB:ELSnA 控制位应当设为 0:0，这样通道 n 就不会使用相同管脚，从而避免了冲突。

表 B-3. 预分频器除数选择

PS2:PS1:PS0	TPM 时钟源除以
0:0:0	1
0:0:1	2
0:1:0	4
0:1:1	8
1:0:0	16
1:0:1	32
1:1:0	64
1:1:1	128

B.5.2 定时器计数器寄存器 (TPMxCNTH:TPMxCNTL)

这两个只读 TPM 计数器寄存器包括 TPM 计数器数值的高字节和低字节。读取任何一个字节 (TPMxCNTH 或 TPMxCNTL) 都能将两个字节的內容锁定到缓冲器中。它们将在该缓冲器中保持锁定状态，直到另一个字节被读取为止。在 MCU 复位，在 TPMxCNTH 或 TPMxCNTL 中写入任意值或者定时器状态 / 控制寄存器 (TPMxSC) 进行任何写入操作时，一致性机制都会自动重启。

复位清除 TPM 计数器寄存器。

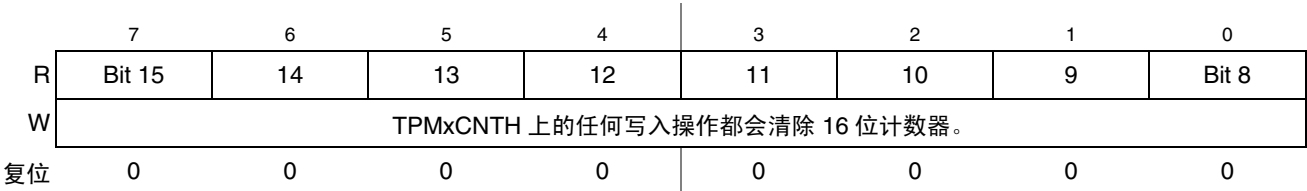


图 B-3. 定时器计数器寄存器高 (TPMxCNTH)