



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	39
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 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/pro/item?MUrl=&PartUrl=mc9s08dz32amlf

表 2-1. 管脚可用性（按封装管脚数）

管脚编号			<-- 最低 优先级 --> 最高			
64	48	32	端口管脚 / 中断		Alt 1	Alt 2
1	1	—	PTB6	PIB6	ADP14	
2	—	—	PTC5		ADP21	
3	2	1	PTA7	PIA7	ADP7	IRQ
4	—	—	PTC6		ADP22	
5	3	—	PTB7	PIB7	ADP15	
6	—	—	PTC7		ADP23	
7	4	2				V _{DD}
8	5	3				V _{SS}
9	6	4	PTG0		EXTAL	
10	7	5	PTG1		XTAL	
11	8	6				RESET
12	9	—	PTF4			ACMP2+
13	10	—	PTF5			ACMP2-
14	—	—	PTF6			ACMP2O
15	11	7	PTE0		TxD1	
16	12	8	PTE1		RxD1	
17	13	9	PTE2			SS
18	14	10	PTE3			SPSCK
19	15	11	PTE4		SCL ³	MOSI
20	16	12	PTE5		SDA ³	MISO
21	—	—	PTG2			
22	—	—	PTG3			
23	17	—	PTF0			TxD2 ⁴
24	18	—	PTF1			RxD2 ⁴
25	19	—	PTF2		TPM1CLK	SCL ³
26	20	—	PTF3		TPM2CLK	SDA ³
27	—	—	PTG4			
28	—	—	PTG5			
29	21	13	PTE6		TxD2 ⁴	TXCAN
30	22	14	PTE7		RxD2 ⁴	RxCAN
31	23	15	PTD0	PID0		TPM2CH0
32	24	16	PTD1	PID1		TPM2CH1

管脚编号			<-- 最低 优先级 --> 最高			
64	48	32	端口管脚 / 中断		Alt 1	Alt 2
33	25	17	PTD2	PID2		TPM1CH0
34	26	18	PTD3	PID3		TPM1CH1
35	27	19	PTD4	PID4		TPM1CH2
36	28	20	PTD5	PID5		TPM1CH3
37	—	—	PTF7			
38	29	—				V _{SS}
39	30	—				V _{DD}
40	31	—	PTD6	PID6		TPM1CH4
41	32	—	PTD7	PID7		TPM1CH5
42	33	21			BKGD	MS
43	—	—	PTC0		ADP16	
44	34	22	PTB0	PIB0	ADP8	
45	—	—	PTC1		ADP17	
46	35	23	PTA0	PIA0	ADP0	MCLK
47	—	—	PTC2		ADP18	
48	36	24	PTB1	PIB1	ADP9	
49	37	25	PTA1	PIA1	ADP1 ¹	ACMP1+ ¹
50	38	—	PTB2	PIB2	ADP10	
51	39	26	PTA2	PIA2	ADP2	ACMP1- ¹
52	—	—	PTC3		ADP19	
53	40	—	PTB3	PIB3	ADP11	
54	41	27	PTA3	PIA3	ADP3	ACMP1O
55	42	28				V _{SSA}
56						V _{REFL}
57	43	29				V _{REFH}
58						V _{DDA}
59	44	30	PTA4	PIA4	ADP4	
60	45	—	PTB4	PIB4	ADP12	
61	—	—	PTC4		ADP20	
62	46	31	PTA5	PIA5	ADP5	
63	47	—	PTB5	PIB5	ADP13	
64	48	32	PTA6	PIA6	ADP6	

1. 如果这两个模拟模块都被启用，他们都将可以访问该管脚。

2. 内部上拉器件被启用时在该管脚上测得的电压可能会低达 V_{DD} - 0.7 V。连接到该管脚上的内部门被拉到 V_{DD} 上。

3. IIC 模块管脚可以通过 SOPT1 寄存器中的 IICPS 位进行重定位。缺省复位位置在 PTF2 和 PTF3 上。

4. SCI2 模块管脚可以通过 SOPT1 寄存器中的 SCI2PS 位进行重定位。缺省复位位置在 PTF0 和 PTF1 上。

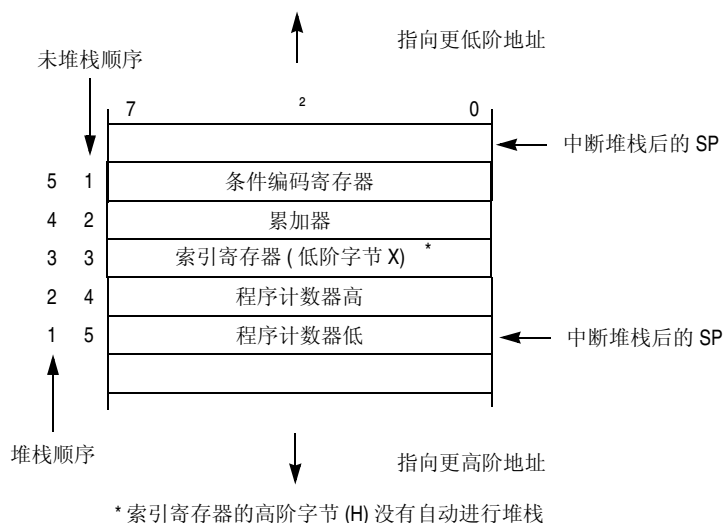


图 5-1. 中断堆栈帧

实施 RTI 指令时，这些值以相反顺序从堆栈中恢复。作为 RTI 顺序的一部分，CPU 通过读取程序报文的 3 个字节（从堆栈中恢复的 PC 地址开始）来填写指令。

在从 ISR 返回前，必须确认 / 清除与中断源对应的状态标记。通常，该标记在 ISR 开始时被清除，这样如果同一源生成另外一个中断，它就可以被登记，从而在当前 ISR 完成后为其提供服务。

5.5.2 外部中断请求（IRQ）管脚

外部中断由 **IRQ** 状态和控制寄存器（**IRQSC**）管理。当 **IRQ** 功能打开后，同步逻辑监控管脚是否发生边沿或边沿加电平的触发事件。当 **MCU** 处于停止模式且系统时钟关闭时，将使用单独的异步路径，这样 **IRQ**（如果被打开）就可以唤醒 **MCU**。

5.5.2.1 管脚配置选项

IRQSC 中的 IRQ 管脚激活 (IRQPE) 控制位必须为 1，这样 IRQ 管脚才能作为中断请求 (IRQ) 输入。作为 IRQ 输入，用户可以选择检测边沿或电平 (IRQEDG) 的极性，无论管脚只检测到边沿还是同时检测到边沿和电平 (IRQMOD)，也无论事件触发的中断还是只设置了软件可以轮询的 IROF 标记。

IRQ 管脚激活后，默认使用内部上拉器件（IRQPDD = 0），器件究竟是上拉还是下拉取决于所选的极性。如果用户希望使用外部上拉或下拉，可以在 IROPDD 中写入 1，以关闭内部器件。

当把 IRQ 管脚配置为 IRQ 输入时，可以使用 BIH 和 BIL 指令来检测 IRQ 管脚上的电平。

注意

这个管脚不包括 V_{DD} 的钳位二极管，并且在高于 V_{DD} 时不应驱动。内部上拉 IRQ 管脚上测量到的电压可能只有 $V_{DD} - 0.7\text{ V}$ 。与该管脚相连的内门一直拉到 V_{DD} 。

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。

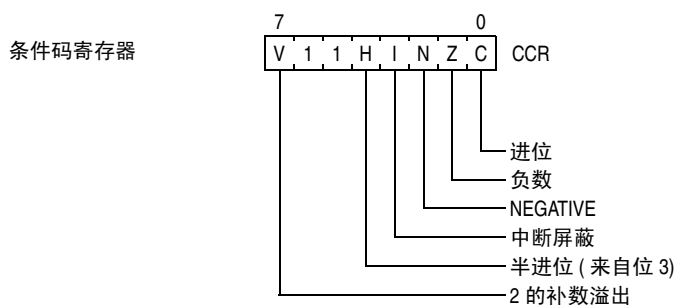


图 7-2. 条件码寄存器

表 7-2. 指令集小结 (第 6 页, 共 9 页)

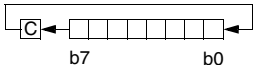
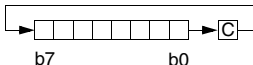
Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	移动 (M) _{destination} ← (M) _{source} 在 IX+/DIR 和 DIR/IX+ 模式, H:X ← (H:X) + \$0001	DIR/DIR DIR/IX+ IMM/DIR 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 -	-↑↑↑

表 10-1. ADC 通道分配

ADCH	通道	输入	ADCH	通道	输入
00000	AD0	PTA0/ADP0/MCLK	01111	AD15	PTB7/ADP15
00001	AD1	PTA1/ADP1/ACMP1+	10000	AD16	PTC0/ADP16
00010	AD2	PTA2/ADP2/ACMP1P-	10001	AD17	PTC1/ADP17
00011	AD3	PTA3/ADP3/ACMP1O	10010	AD18	PTC2/ADP18
00100	AD4	PTA4/ADP4	10011	AD19	PTC3/ADP19
00101	AD5	PTA5/ADP5	10100	AD20	PTC4/ADP20
00110	AD6	PTA6/ADP6	10101	AD21	PTC5/ADP21
00111	AD7	PTA7/ADP7	10110	AD22	PTC6/ADP22
01000	AD8	PTB0/ADP8	10111	AD23	PTC7/ADP23
01001	AD9	PTB1/ADP9	11000– 11001	AD24 through AD25	预留
01010	AD10	PTB2/ADP10	11010	AD26	温度传感器 ¹
01011	AD11	PTB3/ADP11	11011	AD27	内部隙带 ²
01100	AD12	PTB4/ADP12	11100	预留	预留
01101	AD13	PTB5/ADP13	11101	V _{REFH}	V _{REFH}
01110	AD14	PTB6/ADP14	11110	V _{REFL}	V _{REFL}

10.1.3 替代时钟

ADC 模块的时钟源可以是 MCU 总线时钟，总线时钟二分频，模块内的本地异步时钟（ADACK）或替代时钟 ALTCLK。MC9S08DZ60 系列 MCU 器件的替代时钟是外部参考时钟（MCGERCLK）。

所选的时钟源必须运行在一定的频率范围内，这样 ADC 转换时钟 (ADCK) 就可以通过 ADIV 位的设置，在从 ALTCLK 分频后，运行在指定的频率范围 (f_{ADCK}) 内。

当 MCU 处于等待模式时，ALTCLK 是使能的 (满足以上条件)。这使得当 MCU 处于等待模式时，ALTCLK 可以用作 ADC 的工作时钟源。

当 MCU 处于 STOP2 或 STOP3 时，ALTCLK 不能用作 ADC 工作时钟源。

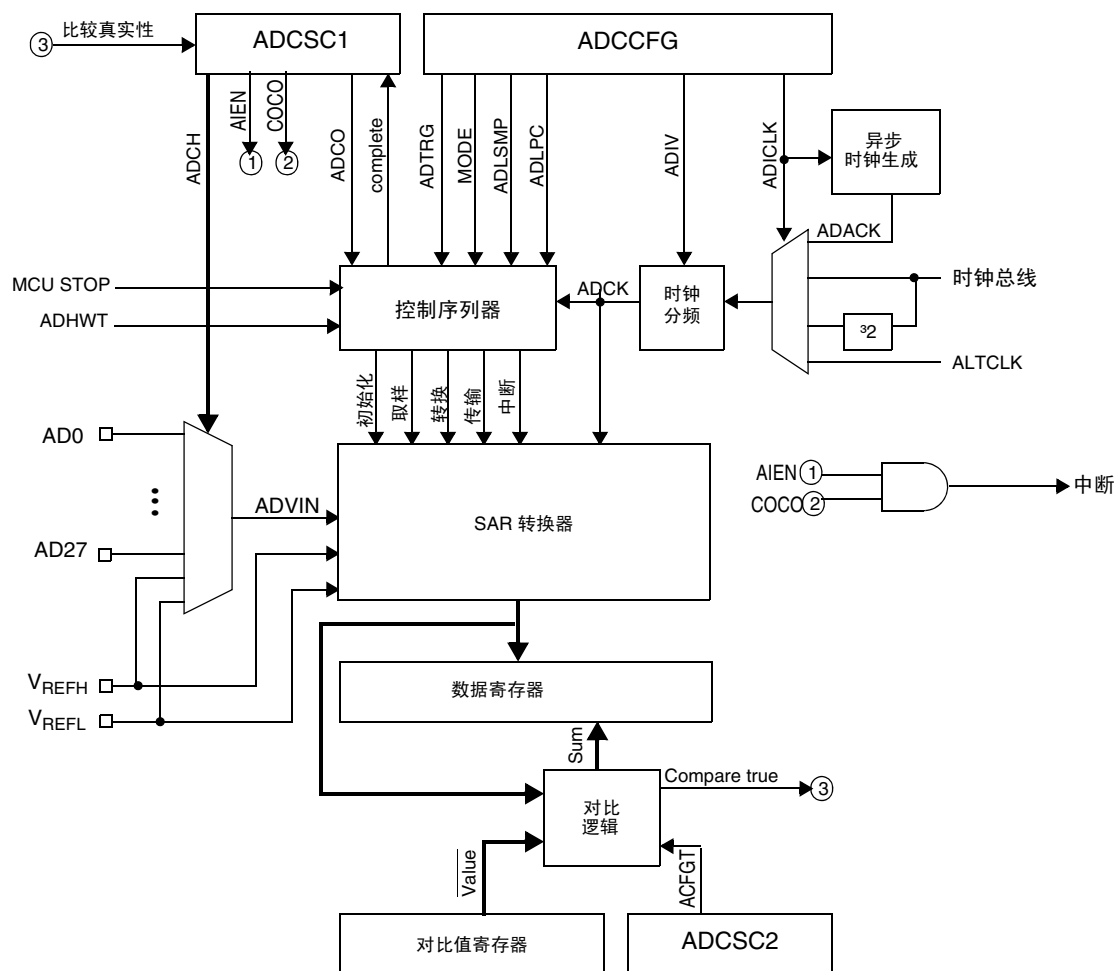


图 10-2. ADC 结构图

10.4 寄存器定义

以下这些存储器映射寄存器控制和显示 ADC 的运行状态：

- 状态和控制寄存器， ADCSC1
- 状态和控制寄存器， ADCSC2
- 数据结果寄存器， ADCRH 和 ADCRL
- 比较值寄存器， ADCCVH 和 ADCCVL
- 配置寄存器， ADCCFG
- 管脚使能寄存器， APCTL1、APCTL2 和 APCTL3

10.4.1 状态和控制寄存器 1（ADCSC1）

本节介绍 ADC 状态和控制寄存器（ADCSC1）的功能。写入 ADCSC1 会中断当前转换，并发起一个新转换（如果 ADCH 位等于全 1 以外的值）。



图 10-3. 状态和控制寄存器（ADCSC1）

表 10-3. ADCSC1 寄存器字段描述

字段	描述
7 COCO	转换完成标志 — COCO 标记是只读位，在每次转换完成时置位，这时比较功能被禁止（ACFE = 0）。当比较功能使能（ACFE =1）时，只有当比较结果为真时才在转换完成时设置 COCO 标记。每次当写入 ADCSC1 或读取 ADCRL 时，该位就被清除。 0 转换未完成 1 转换完成
6 AIEN	中断使能 — AIEN 用来使能转换完成中断。当 COCO 已置位且 AIEN 为 1 时，中断将被触发。 0 转换完成中断禁止 1 转换完成中断使能
5 ADCO	连续转换使能 — ADCO 用来使能连续转换。 0 当选择软件触发时写入 ADCSC1，进行一次转换，或者当选择硬件触发时 ADHWT 触发后，进行一次转换。 1 当选择软件触发时写入 ADCSC1，开始连续转换，或者当选择硬件触发时 ADHWT 触发后，开始连续转换。
4:0 ADCH	输入通道选择 — ADCH 位形成一个 5 位字段，该字段用来选择其中的一个输入通道。图 10-5 详细地描述了输入通道。 通道选择位全部设置为 1 时，逐次逼近转换器子系统关闭。该功能能够彻底禁止 ADC，隔离所有输入通道。用此种方式终止连续转换，可以防止多余的的一次单次转换。当禁止连续转换时，就不需要通过把通道选择位都设置为 1 的方式使 ADC 处于低功耗状态，因为当转换完成时模块会自动进入低功耗状态。

表 10-9. APCTL1 寄存器字段描述

字段	描述
7 ADPC7	ADC 管脚控制 7 — ADPC7 用来控制与通道 AD7 连接的管脚。 0 AD7 管脚 I/O 控制使能 1 AD7 管脚 I/O 控制禁止
6 ADPC6	ADC 管脚控制 6 — ADPC6 用来控制与通道 AD6 连接的管脚。 0 AD6 管脚 I/O 控制使能 1 AD6 管脚 I/O 控制禁止
5 ADPC5	ADC 管脚控制 5 — ADPC5 用来控制与通道 AD5 连接的管脚。 0 AD5 管脚 I/O 控制使能 1 AD5 管脚 I/O 控制禁止
4 ADPC4	ADC 管脚控制 4 — ADPC4 用来控制与通道 AD4 连接的管脚。 0 AD4 管脚 I/O 控制使能 1 AD4 管脚 I/O 控制禁止
3 ADPC3	ADC 管脚控制 3 — ADPC3 用来控制与通道 AD3 连接的管脚。 0 AD3 管脚 I/O 控制使能 1 AD3 管脚 I/O 控制禁止
2 ADPC2	ADC 管脚控制 2 — ADPC2 用来控制与通道 AD2 连接的管脚。 0 AD2 管脚 I/O 控制使能 1 AD2 管脚 I/O 控制禁止
1 ADPC1	ADC 管脚控制 1 — ADPC1 用来控制与通道 AD1 连接的管脚。 0 AD1 管脚 I/O 控制使能 1 AD1 管脚 I/O 控制禁止
0 ADPC0	ADC 管脚控制 0 — ADPC0 用来控制与通道 AD0 连接的管脚。 0 AD0 管脚 I/O 控制使能 1 AD0 管脚 I/O 控制禁止

10.4.9 管脚控制寄存器 2 (APCTL2)

APCTL2 用来控制 ADC 模块的通道 8-15。

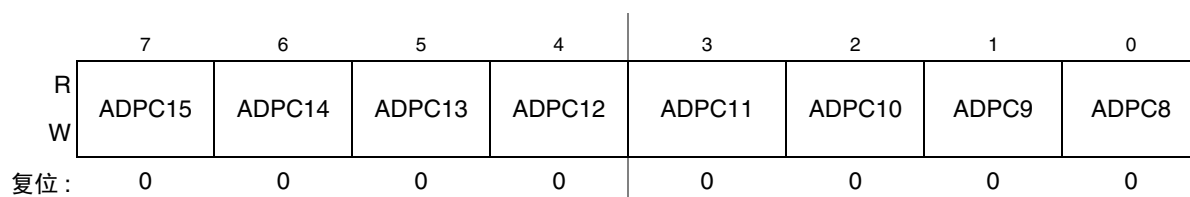


图 10-12. 管脚控制寄存器 2 (APCTL2)

11.2.3 结构图

图 11-3 是 ICC 的结构图。

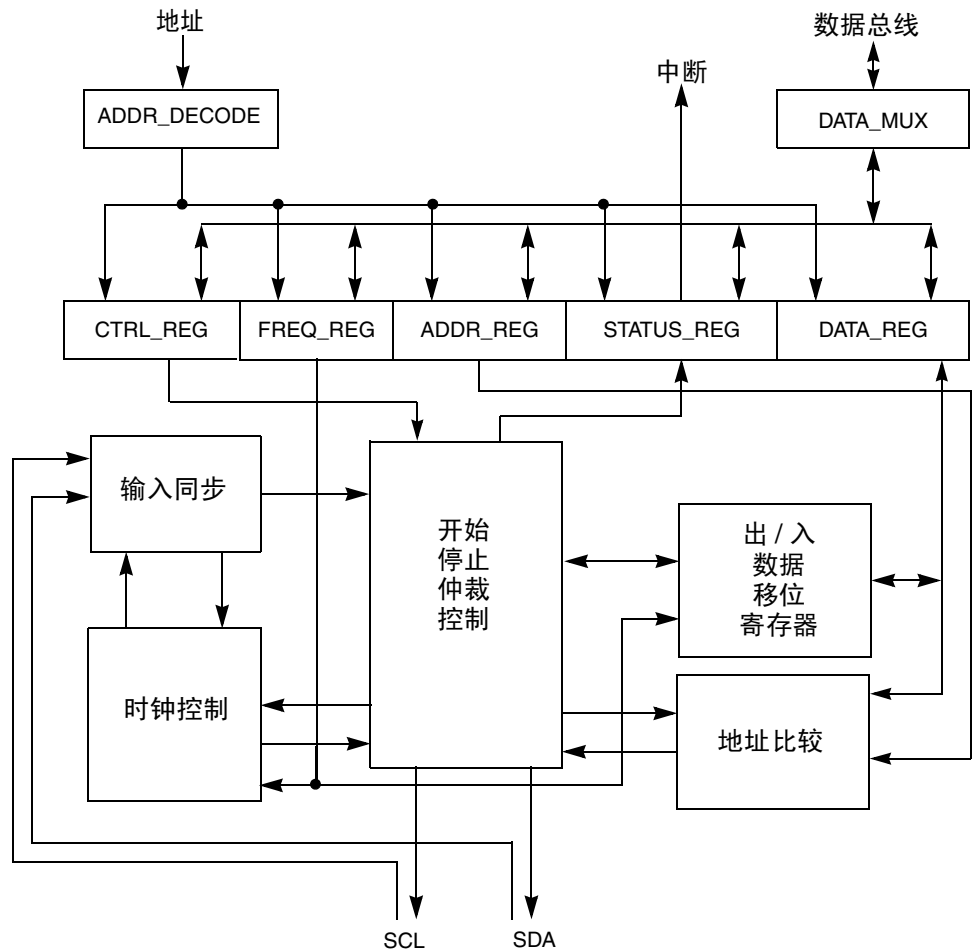


图 11-2. IIC 功能结构图

11.3 外部信号描述

本节描述了用户可连接各个管脚信号。

11.3.1 SCL — 串行时钟线

双向 SCL 是 IIC 系统的串行时钟线。

11.3.2 SDA — 串行数据线

双向 SDA 是 IIC 系统的串行数据线。

注意

当初始化模式处于有效状态时， CANTFLG 寄存器保持复位状态（INITRQ=1， INITAK=1）。当未处于初始化模式时，该寄存器可以写入（INITRQ = 0 and INITAK = 0）。

读取：任何时间
写入：TXEx 标志不处于初始化模式的任何时间；写入 1 清除标志，写入 0 忽略标志。

表 12-11. CANTFLG 寄存器字段描述

字段	描述
2:0 TXE[2:0]	发送器缓冲器空—该标志表示相关发送报文缓冲器空，因此没有安排用于发送。在发送缓冲器中放好报文并准备好发送后，CPU 必须清除该标志。报文发送成功后，MSCAN 设置该标志。当发送请求由于中止请求而被成功中止时，MSCAN 也设置该标志（参见 12.3.8，“MSCAN Transmitter 发送器报文中止请求寄存器 (CANTARQ)”）。如果未被屏蔽，当设置了该标志时产生发送中断。 清除 TXEx 标志也会清除相应的 ABTAKx（参见 12.3.9，“MSCAN 发送器报文中止确认寄存器 (CANTAACK)”）。当设置了 TXEx 标志时，相应的 ABTRQx 位被清除（参见 12.3.8，“MSCAN Transmitter 发送器报文中止请求寄存器 (CANTARQ)”）。 当监听模式处于有效状态时（参见 12.3.2，“控制寄存器 1 (CANCTL1)”）TXEx 标志不能清除且不进行任何发送。 当相应的 TXEx 位被清除（TXEx = 0）且缓冲器被安排用于发送时，对发送缓冲器的读写操作会被拦截。 0 相关报文缓冲器已满（加载了准备发送的报文） 1 相关报文缓冲器空（未安排）

12.3.7 MSCAN 发送器中断使能寄存器 (CANTIER)

该寄存器包含发送缓冲器空中断标志的中断使能位。

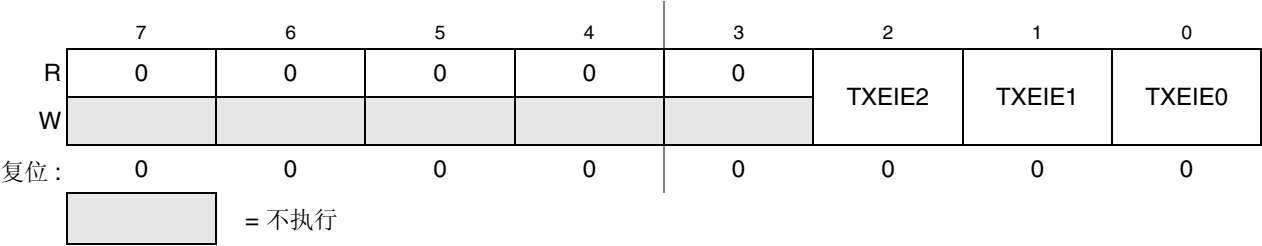


图 12-11. MSCAN 发送器中断使能寄存器 (CANTIER)

注意

当初始化模式处于有效状态时， CANTIER 寄存器保持复位状态（INITRQ=1， INITAK=1）。当未处于初始化模式时，该寄存器可以写入（INITRQ=0， INITAK=0）。

读取：任何时间
写入：未处于初始化模式的任何时间

14.1.2 特性

SCI 模块的特性包括：

- 全双工、标准的不归零（NRZ）格式
- 具有独立使能的双缓冲发射器和接收器
- 可编程波特率（13 位模数分频器）
- 中断驱动型或轮询操作：
 - 发送数据寄存器空，发送完成
 - 接收数据寄存器已满
 - 接收溢出、奇偶效验错误、成帧错误和噪音错误
 - 闲置接收器检测
 - 接收管脚上的活动边沿
 - 支持 LIN 的断点检测
- 硬件奇偶效验生成和检查
- 可编程 8 位或 9 位字符长度
- 按闲置线路或地址标记的接收器唤醒
- 可选的 13 位中止字符生成 / 11 位中止字符检测
- 可选的发射器输出极性

14.1.3 运行模式

如需了解有关以下模式中的 SCI 操作的详细信息，参见 14.3，“功能描述”

- 8 位和 9 位数据模式
- 停止模式操作
- 循环模式
- 单线模式

14.1.4 结构图

图 14-2 显示了 SCI 的发射器部分。

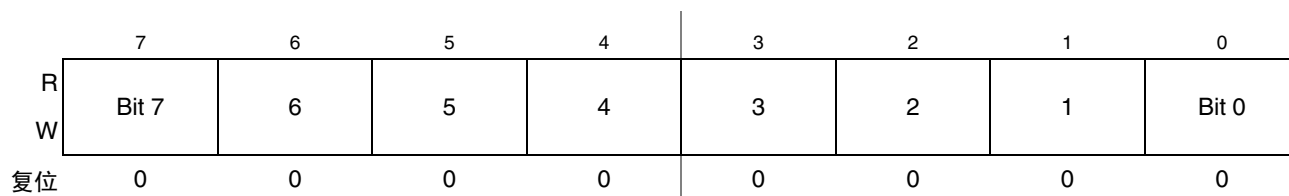


图 16-14. TPM 通道值寄存器低字节 (TPMxCnVL)

在输入捕捉模式下，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锁入到缓冲器中。这些内容一直锁定在那里，直到另一半被读取。当 TPMxCnSC 寄存器被写入时（不管 BDM 模式是否使能），锁存机制可复位（为未锁定状态）。向通道寄存器的任何写入操作在输入捕捉模式下将被忽视。

BDM 处于有效状态时，一致性机制是冻结的（除非通过写入 TPMxCnSC 寄存器复位），这样缓冲器中锁定的内容将一直保持 BDM 处于有效状态时的锁定状态，即使通道寄存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 w 处于有效状态时用户正在读取 16 位寄存器，那么在返回正常运行后它将从 16 位值的另一半中读取适当的值。在 BDM 模式下从 TPMxCnVH 和 TPMxCnVL 寄存器中读取的值是这些寄存器的值而不是它们的读取缓冲器的值。

在输出比较或 PWM 模式下，写入任何一个字节（无论是 TPMxCnVH 还是 TPMxCnVL）都会将该值锁入到缓冲器中。两个字节都被写入后，他们根据 CLKSb:CLKSA 位的值和所选模式作为连贯的 16 位值发送到定时器通道寄存器中，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 而且在 epwm 或 cpwm 模式下)，那么寄存器在第二个字节被写入后和 tpm 计数器下次发生变化（预分频器计数结束）时更新。
- 如果 (clksb:clksa not = 0:0 而且在 epwm 或 cpwm 模式下)，寄存器在两个字节被写入后以及 tpm 计数器从 (tpmxmodh:tpmxmodl - 1) 变为 (tpmxmodh:tpmxmodl) 时更新。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。

锁存机制可通过写入 TPMxCnSC 寄存器（不管 BDM 是否处于活动状态）来手动复位。这种锁定机制允许按从小到大或从大到小的顺序进行连贯的 16 位写入，这对各种编译器的编译都非常友好。

当 BDM 处于有效状态时，一致性机制被冻结，这样缓冲器中锁定的内容仍可保持 BDM 处于有效状态时的锁定状态，即使通道寄存器的一半或两个一半都在 BDM 处于有效状态时被写入。当处于有效状态时，向通道寄存器的任何写入操作都会绕过缓冲器锁定并直接写入到通道寄存器中。一旦恢复正常运行，BDM 处于有效状态时写入到通道寄存器的值用于 PWM 和输出比较操作。处于有效状态时，通道寄存器写入不会干扰一致性序列的部分完成。一致性机制全部运行完毕后，可通过用户写入的缓冲值（BDM 未处于活动状态时）更新通道寄存器。

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。

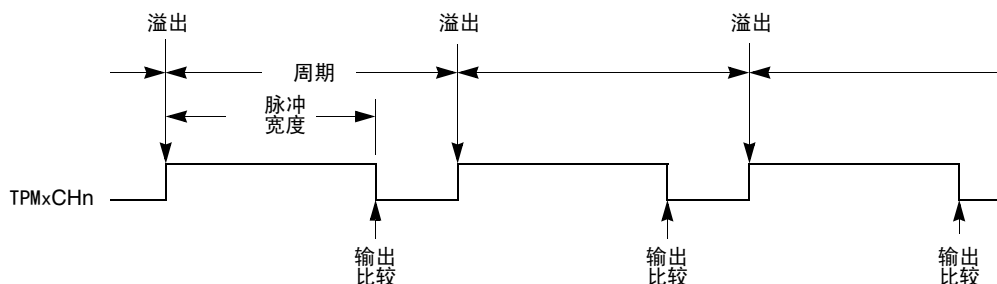


图 16-15. PWM 周期和脉冲宽度 (ELSnA=0)

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

图 17-3 显示主机从目标 HCS08 MCU 收到逻辑 1。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机保持低 BKGD 管脚足够长的时间，使目标识别它（至少两个目标 BDC 周期）。主机必须在目标 MCU 在其认为的位计时开始后驱动瞬时高态加速脉冲七个周期前，释放低电平驱动。主机应该在其启动位时间约 10 个周期后采样位电平。

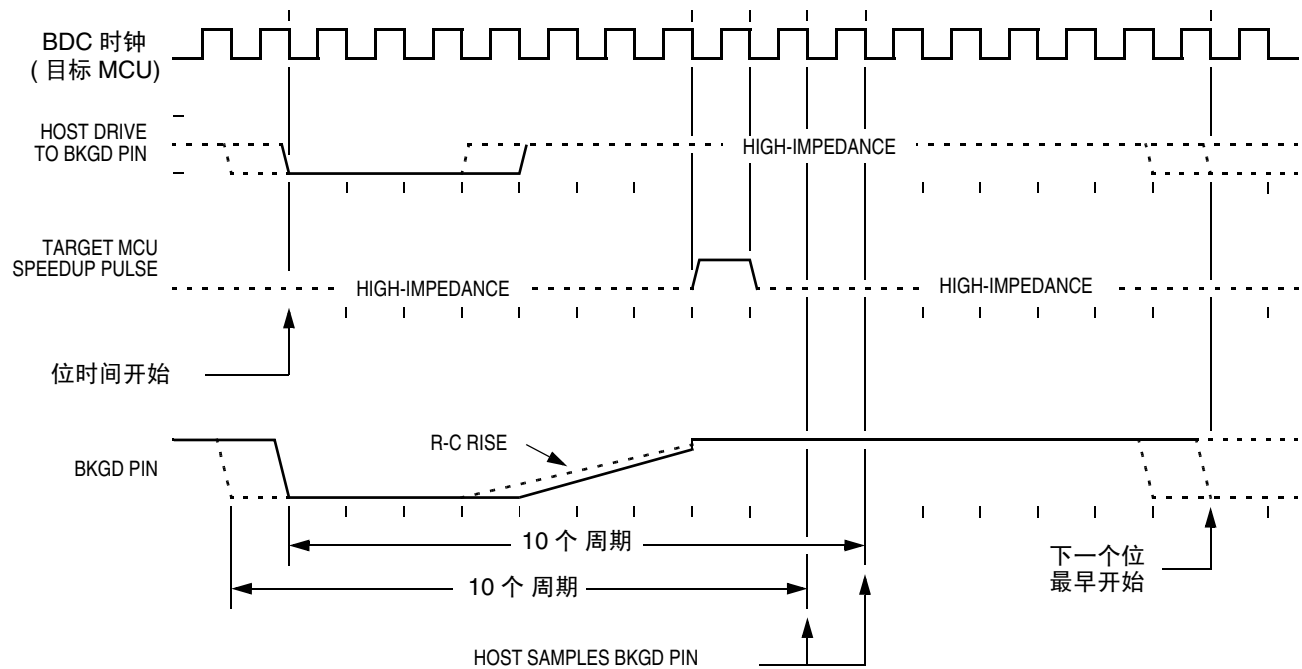


图 17-3. BDC 目标 - 到 - 主机串行位时序 (逻辑 1)

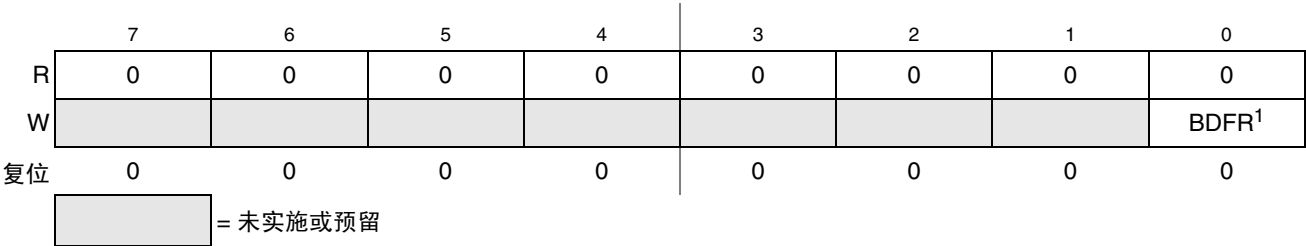
图 17-4 显示了主机从目标 HCS08 MCU 收到逻辑 0。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机启动位时间，但是目标 HCS08 MCU 完成它。由于目标希望主机接收逻辑 0，它保持低 BKGD 管脚 13 个 BKGD 管脚周期，然后驱动引脚置高，加速上升沿。主机在启动位时间约 10 个周期后采样位电平。

17.4.1.2 BDC 断点匹配寄存器 (BDCBKPT)

6- 位 寄存器保留 BDC 中的硬件断点的地址。BDCSCR 中的 BKPTEN 和 FTS 控制位用来使能和配置断点逻辑。专门的串行 BDC 命令 (READ_BKPT 和 WRITE_BKPT) 用来读和写 BDCBKPT 寄存器，但是用户程序不能存取它，因为它不位于 MCU 的普通存储器映射空间中。当目标 MCU 处于激活背景调试模式时，断点一般在运行用户应用程序前设置。关于建立和使用 BDC 中的硬件断点逻辑的更多信息，请参见 17.2.4，“BDC 硬件断点”。

17.4.2 系统背景调试强制复位寄存器 (SBDFR)

这个寄存器包含单个只写控制位。必须要用一个串行后台模式命令，如 WRITE_BYTE，来写 SBDFR。从用户程序写该寄存器的尝试被忽略。读总是返回 0x00。



¹ BDFR 只有通过串行后台模式调试命令才可写，不能通过用户程序来写。

图 17-6. 系统背景调试强制复位寄存器 (SBDFR)

表 17-3. 寄存器字段描述

字段	描述
0 BDFR	背景调试强制复位 — 一系列激活后台模式命令，如 WRITE_BYTE 等，允许外部调试主机强制目标系统复位。将 1 写到这个位，强制 MCU 复位。这个位 不能从用户程序写。

17.4.3 DBG 寄存器和控制位

这个调试模块包括 9 个字节的寄存器空间，用于三个 16- 位寄存器和三个 8- 位控制和状态寄存器。这些寄存器位于存储器空间的高地址空间中，这样它们可以存取正常的应用程序。普通用户应用程序几乎从不接入这些寄存器，除了使用断点逻辑的 ROM patching 机制。

17.4.3.1 调试比较器 A 高寄存器 (DBGCAH)

这个寄存器包含比较器 A 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。

17.4.3.2 调试比较器 A 低寄存器 (DBGCAL)

这个寄存器包含比较器 A 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。

A.9 ADC 特性

表 A-9. 12 位 ADC 操作条件

特性	条件	符号	最小值	典型值 ¹	最大值	单位	注释
电源电压	绝对	V_{DDAD}	2.7	—	5.5	V	
	Delta to V_{DD} ($V_{DD}-V_{DDAD}$) ²	DV_{DDAD}	-100	0	+100	mV	
接地电压	Delta to V_{SS} ($V_{SS}-V_{SSAD}$) ²	DV_{SSAD}	-100	0	+100	mV	
参考电压 高		V_{REFH}	2.7	V_{DDAD}	V_{DDAD}	V	仅在 64 管脚封装中适用 { $V_{REFH} < V_{DDAD}$ 描述性的, 未在生产中测试}
参考电压 低		V_{REFL}	V_{SSAD}	V_{SSAD}	V_{SSAD}	V	不适用于 64 管脚封装 (只适用于 32 和 48 管脚封装)
输入电压		V_{ADIN}	V_{REFL}	—	V_{REFH}	V	
输入电容		C_{ADIN}	—	4.5	5.5	pF	
输入电阻		R_{ADIN}	—	3	5	k Ω	
模拟信号源电阻	12 位模式 $f_{ADCK} > 4\text{MHz}$ $f_{ADCK} < 4\text{MHz}$	R_{AS}	— —	— —	2 5	k Ω	MCU 外部
	10 位模式 $f_{ADCK} > 4\text{MHz}$ $f_{ADCK} < 4\text{MHz}$		— —	— —	5 10		
	8 位模式 (所有有效 f_{ADCK})		—	—	10		
ADC 转换时钟频率	高速 (ADLPC=0)	f_{ADCK}	0.4	—	8.0	MHz	
	低速 (ADLPC=1)		0.4	—	4.0		

¹ 典型值假设 $V_{DDAD} = 5.0\text{V}$ 、温度 = 25°C、 $f_{ADCK} = 1.0\text{MHz}$ ，除非另有其他说明。典型值仅用于参考，并在生产中测试。

² DC 潜在差。

B.5.4 定时器通道 n 状态和控制寄存器 (TPMxCnSC)

TPMxCnSC 包含通道中断状态标记和控制位，用于配置中断启动、通道配置和管脚功能。

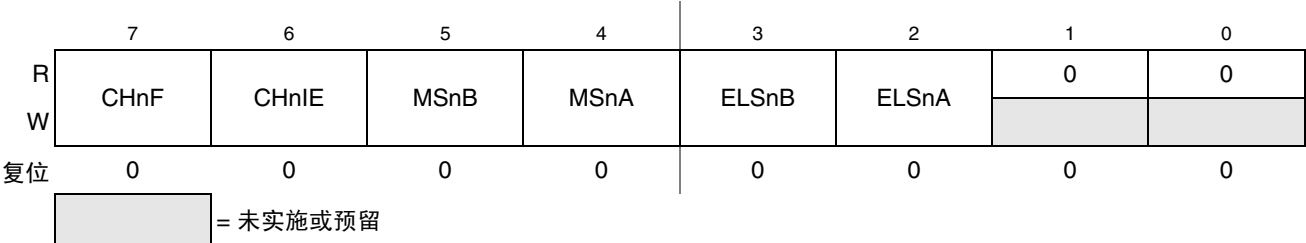


图 B-7. 定时器通道 n 状态和控制寄存器 (TPMxCnSC)

表 B-4. TPMxCnSC 寄存器字段描述

字段	描述
7 CHnF	通道 n 标记 — 通道 n 配置用于输入捕获的情况下，通道 n 管脚上发生活动边时设置该标记位。通道 n 为输出对比或边缘对齐 PWM 通道时，TPM 计数器寄存器中的值与 TPM 通道 n 值寄存器中的值匹配时，CHnF 被设置。此标记很少与中央对齐 PWM 一起使用，因为它是每次计数器与通道值寄存器（与活动工作循环周期的两个边相对应）相匹配时设置的。 当 CHnF 被设置而且中断（CHnIE = 1）被启动时，会请求相应的中断。在 CHnF 被设置时，您可以通过读取 TPMxCnSC 并将一个 0 写入 CHnF 中来清除 CHnF。如果清除序列完成前出现另一个中断请求，则序列将被复位，因此早先 CHnF 的清除序列完成后 CHnF 仍将被设置。这样做的目的是确保清除以前的 CHnF 不会导致 CHnF 中断请求的丢失。复位可清除 CHnF 位。将 1 写入 CHnF 是无效的。 0 通道 n 上没有输入捕获或输出对比事件 1 通道 n 上发生输入捕获或输出对比事件
6 CHnIE	通道 n 中断使能 — 这个读 / 写位从通道 n 中启动中断。复位可清除 CHnIE。 1 通道 n 中断请求使能
5 MSnB	TPM 通道 n 的模式选择 B — 当 CPWMS=0 时，MSnB=1 为边缘对齐 PWM 模式配置 TPM 通道 n。请参考表 B-5 中来查看通道模式和设置控制总结。
4 MSnA	TPM 通道 n 的模式选择 A — 当 CPWMS=0 而 MSnB=0 时，MSnA 为输入捕获模式或输出对比模式配置 TPM 通道 n。请参考表 B-5，查看通道模式和设置控制总结。
3:2 ELSn[B:A]	边缘 / 电平选择位 — 根据 CPWMS:MSnB:MSnA 设置的定时器通道运行模式（如表 B-5 所示），这些位选择触发输入捕获事件的输入边极性，选择根据输出对比匹配将驱动的电平，或者选择 PWM 输出的极性。 将 ELSnB:ELSnA 设置为 0:0 可把相关定时器管脚配置为与任何定时器通道功能无关的通用输入 / 输出管脚。当相关定时器通道被设置为不要求使用管脚的软件定时器时，本功能常用于临时关闭输入捕获通道或允许将定时器管脚用作通用输入 / 输出管脚。

如果通道值寄存器 TPMxCnVH:TPMxCnVL 为零或负数（位 15 被设置），工作周期将是 0%。如果 TPMxCnVH:TPMxCnVL 是正值（位 15 被清除）并大于（非零）模数设置，工作周期将为 100%，因为工作周期比较将不会发生。这意味着模数寄存器设置的可用周期范围为 $0x0001$ 至 $0x7FFE$ （如果不需要 100% 的工作周期，则为 $0x7FFF$ ）。这不是一个重要的限制条件，因为结果周期远远长于正常应用所需的周期。

$\text{TPMxMODH:TPMxMODL} = 0x0000$ 是不应与中央对齐 PWM 模式一同使用的特例。当 $\text{CPWMS}=0$ 时，这一情况与在 $0x0000$ 和 $0xFFFF$ 之间自由运行的计数器对应，然而当 $\text{CPWMS}=1$ 时，计数器需要与 $0x0000$ 以外的某个模数寄存器值有效匹配，以便将方向从向上计数改变为向下计数。

图 B-11 显示了 TPM 通道寄存器（乘以 2）中的输出比较值。该值决定 CPWM 信号的脉冲宽度（工作周期）。如果 $\text{ELSnA}=0$ ，向上计数时的比较匹配会强制 CPWM 输出信号降低；而向下计数时的比较匹配会强制输出升高。计数器向上计数，直到达到中的模数设置，然后向下计数，直到 0。这样就可将周期设置为 TPMxMODH:TPMxMODL 的两倍。

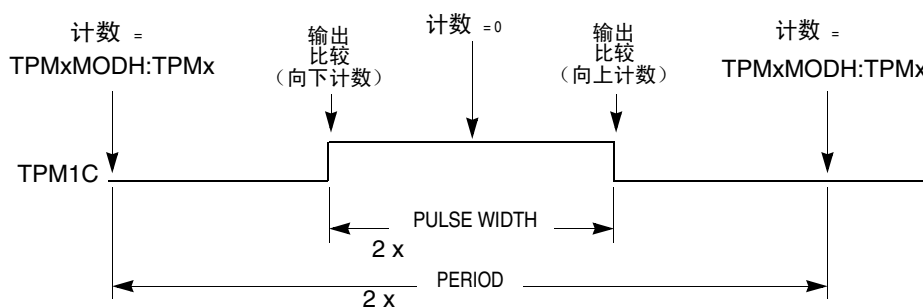


图 B-11. CPWM 周期和脉冲宽度 ($\text{ELSnA} = 0$)

中央对齐 PWM 输出生成的噪音一般比边缘对齐 PWM 小，因为相同系统时钟边上排列的输入 / 输出过渡更少。某些类型的电机也需要这类 PWM。

因为 HCS08 是一个 8 位 MCU 系列。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，不管是 TPMxMODH 、 TPMxMODL 、 TPMxCnVH 、还是 TPMxCnVL ，实际就是写入到缓冲器寄存器。只有在 16 位寄存器的两个 8 位字节都被写入而且定时器计数器溢出后（在模数寄存器上的终端计数结束后从向上计数改变为向下计数），值才被发送到相应的定时器通道寄存器中。这一 TPMxCNT 溢出要求只适用于 PWM 通道，而不是输出比较。

当 $\text{TPMxCNTH:TPMxCNTL} = \text{TPMxMODH:TPMxMODL}$ 时，TPM 可在计数结束时生成一个 TOF 中断。用户可以选择重新上载任何数量的 PWM 缓冲器，并且它们可以在新的周期开始时同时更新。

TPMxSC 写入操作会取消写入到 TPMxMODH and/or TPMxMODL 中的任何值，并且为模数寄存器复位一致性机制。 TPMxCnSC 写入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。