
MC9S08AW60

数据手册

苏州大学飞思卡尔嵌入式系统研发中心翻译

<http://sumcu.suda.edu.cn>

2009年11月

MC9S08AW60 特点

8 位 HCS08 中央处理单元 (CPU)

- 40 MHz 的 HCS08 的 CPU (中央处理单元)
- 20 MHz 的内部总线频率
- HC08 指令子集, 增加了 BGND 指令
- 单线后台调试模式接口
- 断点的能力, 允许单一的断点设置在线调试 (在片内调试模块加了多于两个的断点)
- 片内实时, 在线仿真 (ICE) 带有两个比较器 (在 BDM 中要加一) , 9 个触发模式, 以及片内总线捕获缓冲区。通常显示, 在触发点之前或之后大约有 50 条指令。
- 支持多达 32 个中断/复位源

存储器选项

- 高达 60 KB 的片内在线可编程 FLASH 存储器, 带有块保护和安全选项
- 高达 2 KB 的片内 RAM

时钟源选项

- 时钟源选项包括晶体, 谐振器, 外部时钟, 或内部产生的时钟与精密 NVM 切边

系统保护

- 可选的计算机正常操作 (COP) 复位
- 低电压检测与复位或中断
- 非法操作码检测与复位
- 非法地址检测与复位 (一些设备不具有非法地址)

省电模式

- Wait 另加两个 STOPS

外部设备

- ADC-多达 16 个通道, 10 位 AD 转换器与自动比较功能
- SCI-两个串行通信接口模块与可选的 13 位中断
- SPI 的-串行外设接口模块
- IIC -集成电路互连总线模块运作高达 100 kbps 的最高总线负载;拥有更高的传输速率与减少了负荷
- Timers-1 个 2 通道和 1 个 6 通道 16 位定时器/脉冲宽度调制器 (TPM) 模块: 可选输入捕捉, 输出比较, 和边沿分布的 PWM 能力, 在每个通道。每个定时器模块, 可配置为缓冲, 对所有通道以 PWM (CPWM) 为中心
- KBI -高达 8 引脚键盘中断模块

输入/输出

- 高达 54 通用输入/输出 (I/O) 管脚
- 输入时, 每个端口都有软件选择的上拉电阻
- 输出时, 每个端口都有软件选择的回转速率控制
- 输出时, 每个端口都有软件选择的驱动强度
- 主复位引脚和上电复位 (POR)
- 内部上拉复位 RESET , IRQ, 和 BKGD /MS 管脚, 以减少客户的系统成本

封装选择

MC9S08AW60/48/32

- 64 引脚的四方扁平封装 (QFP)
- 64 引脚的低轮廓四方扁平封装 (LQFP)
- 48 引脚的低轮廓四方扁平封装 (QFN)
- 44 引脚的低轮廓四方扁平封装 (LQFP)

MC9S08AW16

- 48 引脚的低轮廓四方扁平封装 (QFN)
- 44 引脚的低轮廓四方扁平封装 (LQFP)

第一章 导言

1.1 概述

MC9S08AW60，MC9S08AW48，MC9S08AW32 和 MC9S08AW16 是低成本、高性能 8 位微处理器单元（MCUs）HCS08 家族中的成员。家族中所有的 MCUs 使用增强型 HCS08 核，且使用不同的模块，存储大小，存储器类型和封装类型。请参考表 1-1 存储器的大小和封装类型。

表 1-2 总结了在 MC9S08AW60 系列中芯片的每个封装类型的可用外设。

表1-1 MC9S08AW60系列的芯片

芯片	FLASH	RAM	封装
MC9S08AW60	63,280	2048	64 QFP
MC9S08AW48	49,152		64 LQFP
MC9S08AW32	32,768		48 QFN
MC9S08AW16	16,384	1024	44 LQFP
			48 QFN
			44 LQFP

表1-2 每个封装的可用外设

特点	封装选项		
	64 引脚	48 引脚	44 引脚
ADC	16 通道	8 通道	8 通道
IIC	有	有	有
IRQ	有	有	有
KB11	8	7	6
SCI1	有	有	有
SCI2	有	有	有
SPI1	有	有	有
TPM1	6 通道	4 通道	4 通道
TPM1CLK	有	无	无
TPM2	2 通道	2 通道	2 通道
TPM2CLK	有	无	无
I/O 引脚	54	38	34

1.2 MCU框图

图 1-1 给出了 MC9S08AW60 MCU 的结构。

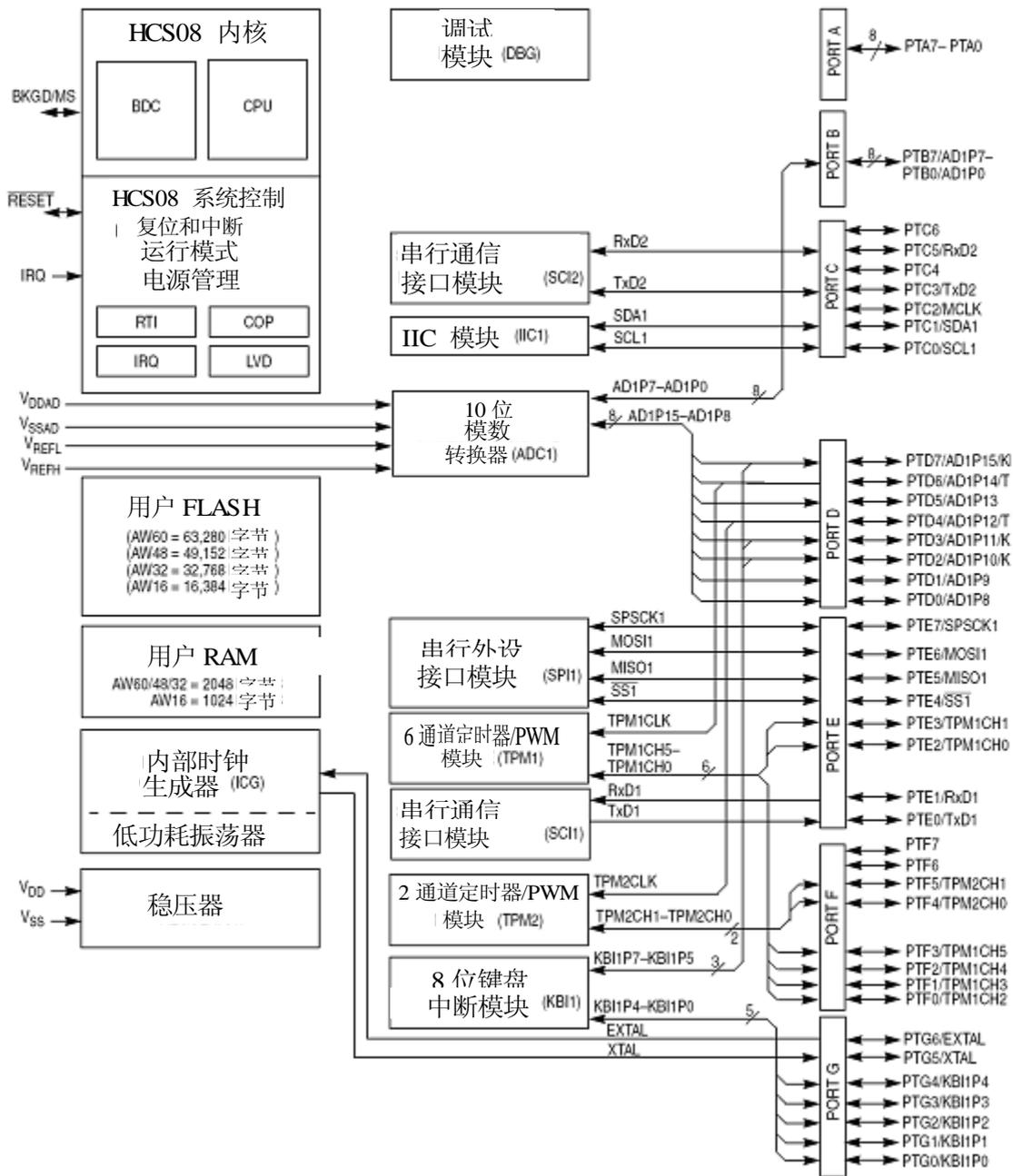


图1-1 MC9S08AW60 MCU框图

注:

1. 如果是输入端口，端口引脚被软件配置为上拉装置。
2. 如果 IRQ 启用 ($IRQPE = 1$)，引脚包含软件配置上拉/下拉式设备。如果上升沿检测被选定 ($IRQEDG = 1$)，下拉式启用。
3. IRQ 没有一个钳二极管电源 V_{DD} 。IRQ 不应该在 V_{DD} 上被驱动。
4. 引脚包含集成上拉装置。
5. 引脚 PTD7，PTD3，PTD2 和 PTG4 包含上拉和下拉式装置。下拉式启用，当 KBI 启用 ($KBIPEN = 1$) 和上升沿被选定 ($KBEDGN = 1$)。

表 1-3 列出了片内模块的功能版本。

表 1-3 模块的版本

模块	版本
AD 转换器 (S08 ADC10)	1
内部时钟生成器 (S08 ICG)	4
内部集成电路 (S08 IIC)	1
键盘中断 (S08 KBI)	1
串行通讯接口 (S08 SCI)	2
串行外设接口 (S08 SPI)	3
计时器脉冲宽度调制器 (S08TPM)	2
中央处理单元 (S08 CPU)	2
调试模块 (DBG)	2

1.3 系统时钟分配

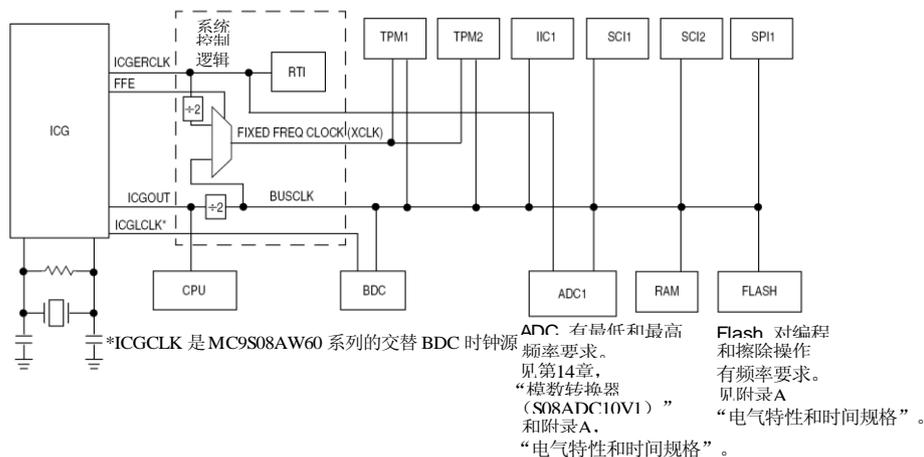


图1-2 系统时钟分配图

一些模块内的 MCU 有时钟源的选择。图 1-2 显示了一个简化的时钟接线图。ICG 提供了时钟来源：

- ICGOUT 是 ICG 模块的一个输出。它可以是下列之一：
 - 外部晶体振荡器
 - 外部时钟源
 - 频率锁相环子模块中的输出数位控制振荡器 (DCO)
 - ICG 内的控制位确定连接哪个时钟源。
- FFE 是一个控制信号，在 ICG 内部产生。如果 ICGOUT 的频率 $> 4 \times$ ICGERCLK 的频率，这个信号是一个逻辑 1，固定频率的时钟将为 ICGERCLK / 2 。否则，固定频率的时钟将为 BUSCLK 。
- ICGLCLK — 开发工具可以选择内部自时钟频率源 (\sim 8MHz) ，以加快 BDC 通信系统，在那里总线时钟很慢。
- ICGERCLK — 外部参考时钟可以选择作为实时中断时钟源。也可以用来作为

ALTCLK, 输入给 ADC 模块。

第二章 引脚和连接

2.1 简介

这一章描述了连接到各封装引脚的信号。内容包括引脚布局图，信号性能表，并对信号进行了详细地描述。

2.2 芯片引脚分配

图 2-1 和 2-2 分别给出了 MC9S08AW60 64 脚和 48 脚封装的引脚分配。

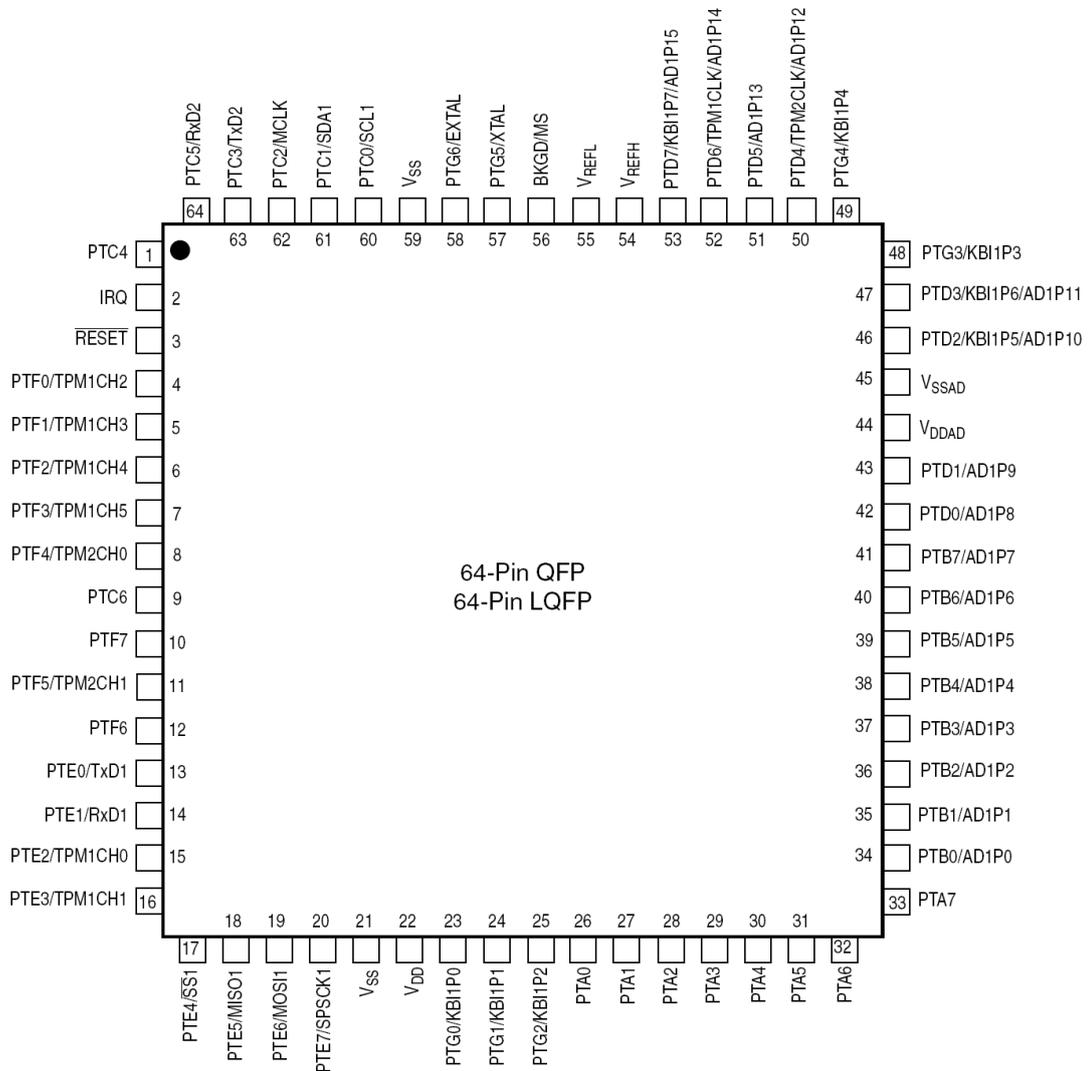


图2-1 MC9S08AW60 64脚QFP/LQFP封装图

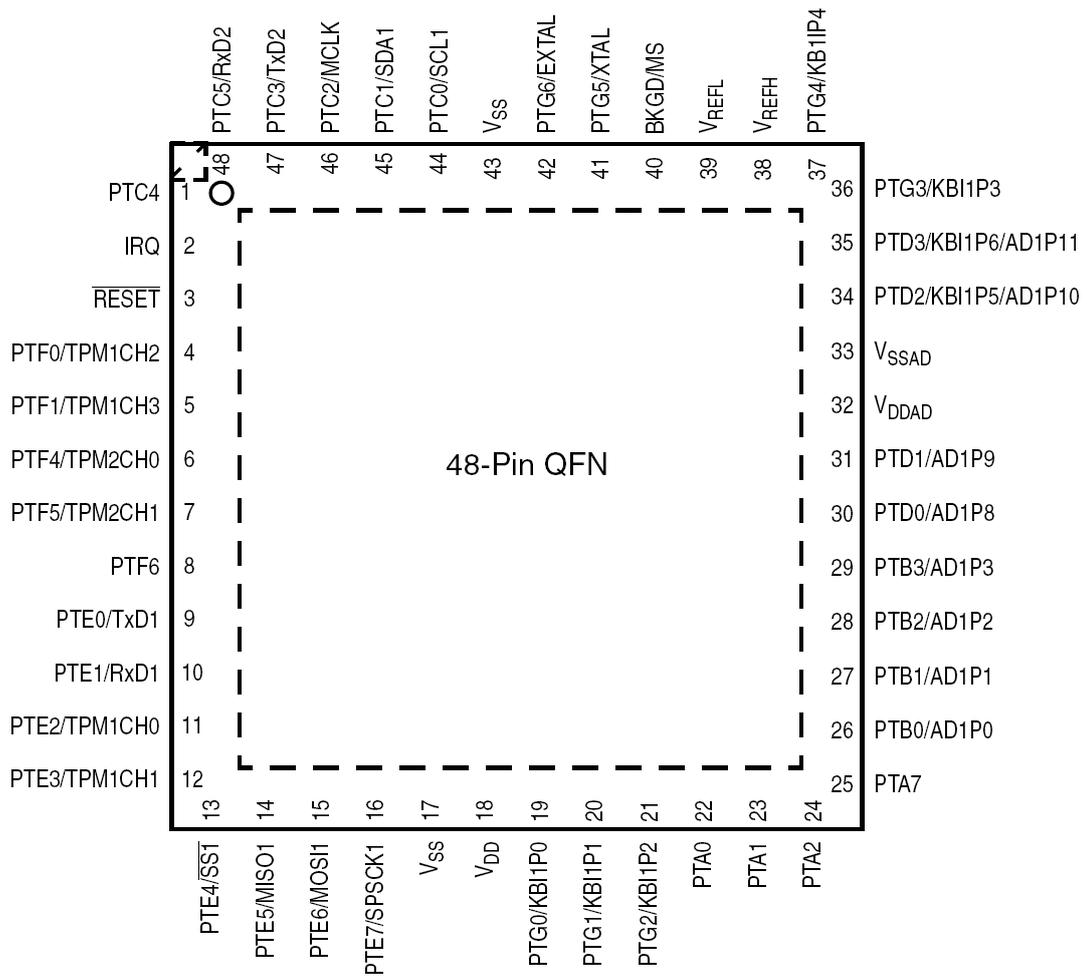


图2-2 MC9S08AW60 48脚QFN封装图

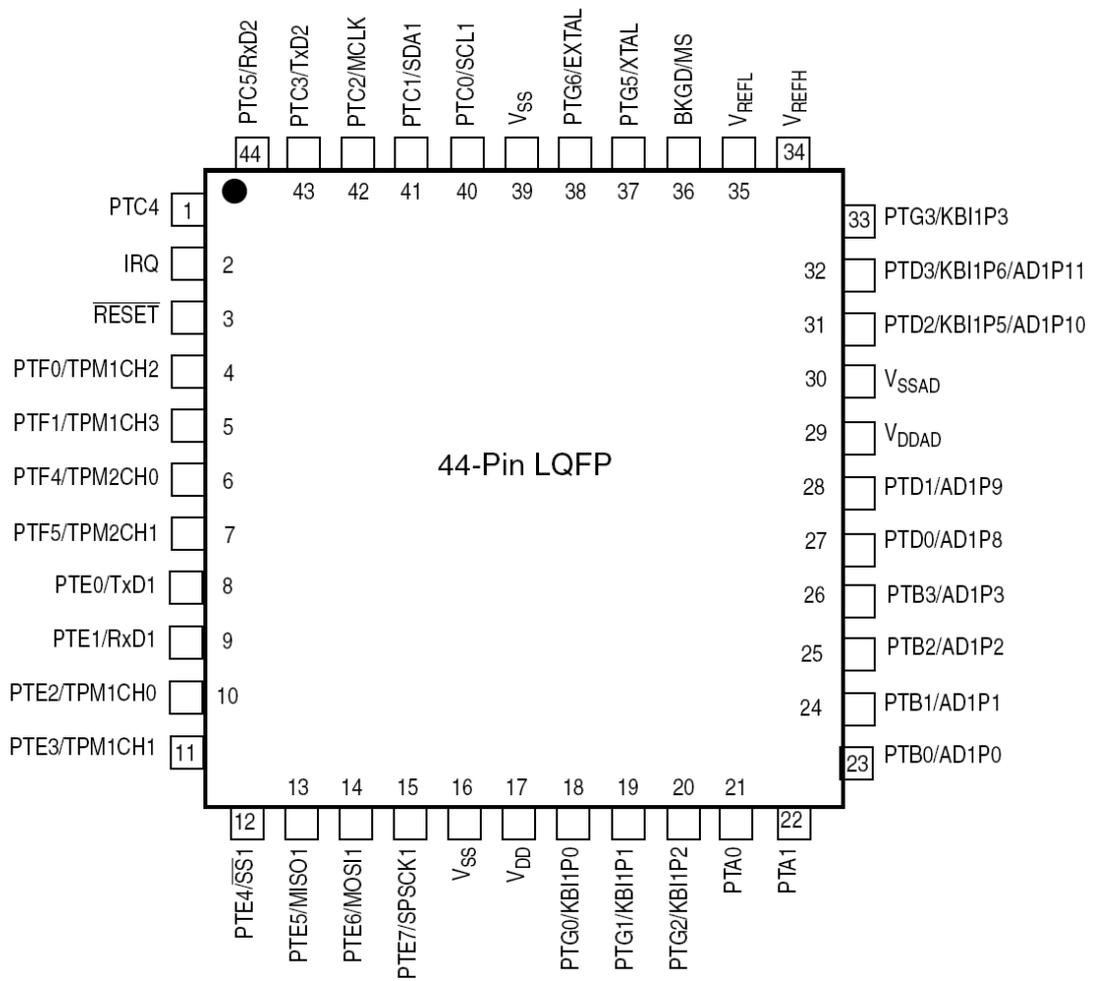
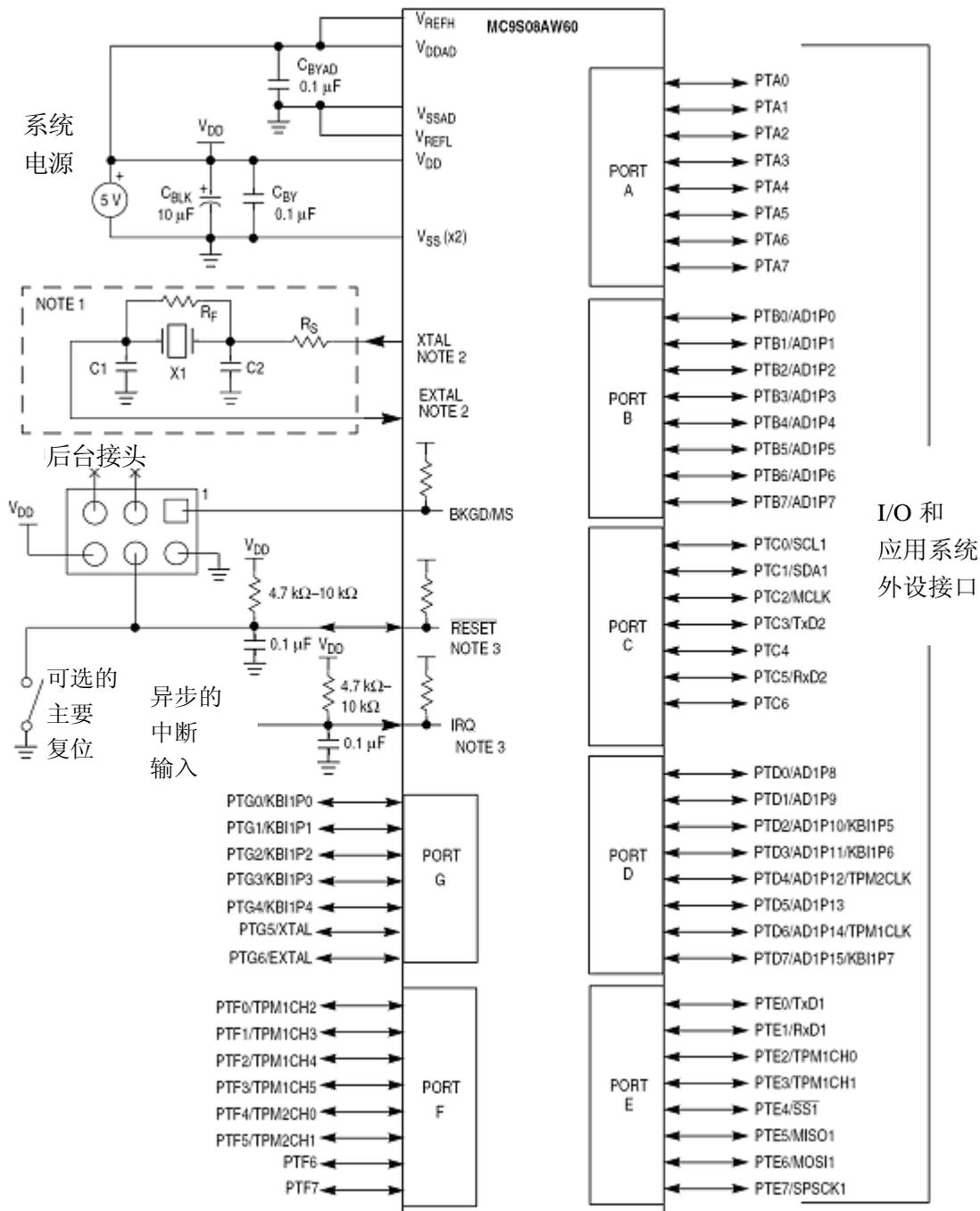


图2-3 MC9S08AW60 44脚LQFP封装图

2.3 推荐的系统连接

图 2-4 给出了几乎所有的 MC9S08AW60 系列应用系统的通用引脚连接。



注意:

1. 如果使用内部时钟, 则不需要外部晶体电路。
2. 类似 PTG5 和 PTG6 之类的是相同的引脚。
3. EMC 敏感应用中, 建议在 RESET 和 IRQ 引脚上使用 RC 滤波器。

图2-4 基础系统连接

2.3.1 电源 (V_{DD} , $2 \times V_{SS}$, V_{DDAD} , V_{SSAD})

V_{DD} 和 V_{SS} 是 MCU 主要电源供应引脚。该电源给所有 I/O 缓冲电路和一个内部电压调节器供电。内部电压调节器提供调节后的低压电源给 CPU 和 MCU 其它内部电路。

一般情况下, 实际应用系统中电源引脚接两个电容。在这种情况下, 应该一个大容量电解电容器, 如 $10\mu\text{F}$ 的钽电容, 用来为全体系统提供大电量存储, 和一个 $0.1\mu\text{F}$ 的陶瓷旁路电容, 尽可能的靠近配对的 V_{DD} 和 V_{SS} 电源引脚, 用以抑制高频噪声。该 MC9S08AW60 有第二个 V_{SS} 引脚。这个引脚应连接到系统的接地平面或到主要的 V_{SS} 引脚, 通过一个低阻抗连接。

这个电压源给 ADC 模块提供电源。一个 $0.1\mu\text{F}$ 的陶瓷旁路电容, 尽可能的靠近模拟电源引脚, 用以抑制高频噪声。

2.3.2 振荡器 (XTAL, EXTAL)

走出复位后, MCU 使用内部产生的时钟 (自时钟模式— $f_{\text{self_reset}}$) 相等于约 8 MHz 的晶体率。这个频率源是用来在上电复位启动和使能作为时钟源停止还原, 以避免需要一个长期的晶体启动时延。这个 MCU 也包含一个内部时钟发生器 (ICG) 模块, 可用于运行 MCU。想了解 ICG 的更多信息, 请参阅第 8 章, “内部时钟发生器 (S08ICGV4)”。

振荡器的振幅 XTAL 和 EXTAL 是有限的, 为低功率振荡。通常情况下, 这些引脚有一个 1-V 峰值到峰值信号。在噪声环境中, 高增益输出 (HGO) 位可设置为使轨至轨振荡。

振荡器在这个 MCU 里是一个皮尔斯振荡器, 可容纳一个晶体或陶瓷谐振器, 由 ICGC1 寄存器的 RANGE 位来选择两种频率范围的一种。而非晶体或陶瓷谐振器, 一个外部振荡器可以连接到 EXTAL 输入引脚。

以下讨论请参考图 2-4。 R_S (使用时) 与 R_F 应是低电感电阻, 如碳组成的电阻。丝伤口电阻和一些金属膜电阻器, 有太多的电感。 $C1$ 和 $C2$, 通常应是高品质的陶瓷电容器, 是专门为高频率应用设计的。

R_F 是用来提供一个偏的路径来保持 EXTAL 输入, 在其线性范围, 晶体启动和其值不是一般的密钥之间。典型的系统使用 $1\text{M}\Omega$ 至 $10\text{M}\Omega$ 。更高值是敏感的湿度和更低值减少增益和 (在极端情况下) 可以防止启动。

$C1$ 和 $C2$, 通常在 5pF 至 25pF 范围和选择相匹配的要求, 具体晶体或谐振器。一定要考虑到印刷电路板 (PCB) 电容和 MCU 引脚电容, 当测量 $C1$ 和 $C2$ 时。水晶制造商通常指定负载电容, 其中是一系列 $C1$ 和 $C2$ 的组合, 这通常是大小相同的。作为第一阶近似, 使用 10pF 作为一个估计, 结合引脚和 PCB 电容作为每个振荡器引脚 (EXTAL 和 XTAL)。

2.3.3 $\overline{\text{RESET}}$ 引脚

$\overline{\text{RESET}}$ 是一个专门引脚, 里面有一个上拉装置。它有输入滞后, 一个高电流输出驱动器, 没有输出转换率控制。内部上电复位和低电压复位电路通常不需要外部复位电路。这个引脚通常是连接到标准的 6 引脚后台调试连接器, 所以一个开发系统可以直接复位 MCU 系统。如果期望, 可以增加一本外部复位手册, 通过供应一个简单的开关到地面 (上拉复位引脚拉低来迫使一个复位)。

每当有复位启动 (无论是从外部信号或从内部系统), 这个复位引脚被驱动低于大约 34 个总线周期, 松一点说, 再次取样约 38 个总线周期以后。如果复位被一个内部源引起, 例如低电压复位或看门狗超时, 电路期望复位引脚的样本返回逻辑 1。这个复位电路解码复

位事件并记录下来，通过在系统控制复位状态寄存器（SRS）中设置相应的位。

在 EMC 敏感的应用中，一个外部 RC 滤波器被配置在复位引脚上。见图 2-4 就是一个例子。

2.3.4 后台/模式选择（BKGD/MS）

在复位期间，BKGD/MS 引脚功能作为一个模式选择引脚。复位提升引脚的功能后，该引脚立即作为后台引脚被用于后台调试通信。当作为后台/模式选择引脚时，有一个内部上拉电阻，输入滞后，一个标准输出驱动器，并没有输出转换率控制。

若这个引脚没有连接，则 MCU 在复位的上升沿进入正常的操作模式。如果调试系统被连接到引脚 6 的标准后台调试接口，它可以在复位上升沿时保持 BKGD/MS 为低电平，强制 MCU 进入后台模式。

BKGD 引脚主要被后台调试控制器(BDC)使用，它使用私有协议进行通信，该协议使用被调试 MCU 的 16 个 BDC 时钟周期来传送一个位。被调试 MCU 的 BDC 时钟频率使用最大总线时钟频率，因此不要将任何大的电容和 BKGD/MS 引脚相连，否则会干扰 BKGD 与后台的串行通信。

虽然 BKGD 引脚是一个准开漏引脚，但是后台调试通信协议提供了精简的、激活驱动、高速脉冲以确保快速上升沿。来自电缆的小电容和内部上拉电阻的绝对值对 BKGD 引脚上的上升沿和下降沿几乎不起任何作用。

2.3.5 ADC 的参考引脚（VREFH，VREFL）

该 V_{REFH} 和 V_{REFL} 引脚分别为 ADC 模块输入高参考电压和低参考电压。

2.3.6 外部中断引脚（IRQ）

IRQ 引脚是 IRQ 中断的输入源，又是 BIH 和 BIL 指令的输入。如果 IRQ 的功能没有确立，该引脚不执行任何功能。

当 IRQ 配置为 IRQ 的输入和设定为上升沿检测时，启用下拉电阻，而非上拉电阻。

在 EMC 敏感的应用中，一个外部 RC 滤波器被推荐在 IRQ 引脚上。见图 2-4 就是一个例子。

2.3.7 GPIO 及外设端口

其余引脚被通用 I/O 引脚、片内外设功能复用，比如定时器和串行 I/O 系统。复位后，所有的引脚立即被配置为高阻态通用输入引脚且没有内部上拉电阻。后立即复位，所有这些引脚配置为高阻抗一般用途的投入与内部拉设备停用。

注：为了避免来自浮空输入引脚额外的漏电流，应用程序中的复位初始化例程必须要么使能上拉或下拉电阻，要么改变不常用引脚的方向置为输出使该引脚不再浮空。

如需控制这些引脚作为通用 I/O 引脚的信息，见第 6 章“并行输入/输出”。如需有关如何以及何时对片内外设系统使用这些引脚，请参照表 2-1 中的适当章节。

表2-1引脚复用优先级

最低级←引脚功能优先级→最高级			参考 ¹
端口引脚	交替功能	交替功能	
PTB7—PTB0	AD1P7—AD1P0		第 14 章，“模数转换器（S08ADC10V1）”
PTC5, PTC3	RxD2—TxD2		第 11 章，“串行通信接口（S08SCIV2）”

PTC2	MCLK		第 5 章, “复位, 中断和系统控制”
PTC1-PTC0	SDA1-SCL1		第 13 章, “内部集成电路 (S08IICV1)”
PTD7	KBI1P7	AD1P15	第 14 章, “模数转换器 (S08ADC10V1)” 第 9 章, “键盘中断 (S08KBIV1)”
PTD6	TPM1CLK	AD1P14	第 14 章, “模数转换器 (S08ADC10V1)” 第 10 章, “计时器/PWM (S08TPMV2)”
PTD5		AD1P13	第 14 章, “模数转换器 (S08ADC10V1)”
PTD4	TPM2CLK	AD1P12	第 14 章, “模数转换器 (S08ADC10V1)” 第 10 章, “计时器/PWM (S08TPMV2)”
PTD3-PTD2	KBI1P6-KBI1P5	AD1P11-AD1P10	第 14 章, “模数转换器 (S08ADC10V1)” 第 9 章, “键盘中断 (S08KBIV1)”
PTD1-PTD0	AD1P9-AD1P8		第 14 章, “模数转换器 (S08ADC10V1)”
PTE7 PTE6 PTE5 PTE4	SPSCK1 MOSI1 MISO1 SS1		第 12 章, “串行外设接口 (S08SPIV3)”
PTE3-PTE2	TPM1CH1- TPM1CH0		第 10 章, “计时器/PWM (S08TPMV2)”
PTE1-PTE0	RxD1-TxD1		第 11 章, “串行通信接口 (S08SCIV2)”
PTF5-PTF4	TPM2CH1- TPM2CH0		第 10 章, “计时器/PWM (S08TPMV2)”
PTF3-PTF0	TPM1CH5-TPM1CH2		第 10 章, “计时器/PWM (S08TPMV2)”
PTG4-PTG0	KBI1P4-KBI1P0		第 9 章, “键盘中断 (S08KBIV1)”
PTG6-PTG5	EXTAL-XTAL		第 8 章, “内部时钟生成器 (S08ICGV4)”

1. 参见这些列出的章节来查看模块复用引脚的信息。

当一个片内外设系统来控制一个引脚，数据方向控制位仍然确定从端口数据寄存器中读出的是什么，即使外设模块通过控制启用引脚的输出缓存来控制引脚的方向。看第 6 章, “并行输入/输出”来了解更多详情。

上拉允许位控制每个输入引脚是否允许片内上拉电阻，无论何时该引脚作为输入，即使它被一个片内外设模块控制。当 PTD7，PTD3，PTD2 和 PTG4 引脚由 KBI 模块控制并且被配置为上升沿/高电平敏感度，上拉允许控制位允许下拉电阻，而非上拉电阻。

注:

当另一种功能首次启用，这有可能得到的一个**优势沿**模块，在启用中断前，用户软件应明确指出任何相关的标志。表 2-1 说明，如果有多个优先模块启用。最高优先模块将控制完引脚。选择较高的优先引脚功能与较低的优先功能已经启用，可对较低的优先模块产生优势沿。它是建议所有模块复用引脚，在开启另外一个模块之前要关闭。

第三章 操作模式

3.1 简介

本章将描述 MC9S08AW60 的操作模式。同时描述进入各种模式、退出各种模式和在各种模式中的功能。

3.2 特性

- 代码开发的激活后台模式
- 等待模式：
 - CPU 的关机，以节省电力
 - 系统时钟运行
 - 全电压调节保持
- 停止模式：
 - 系统时钟停止;稳压器在备用状态
 - STOP2—部分掉电内部电路，RAM 的内容保留
 - STOP3—为快速恢复，所有内部电路不断电

3.3 运行模式

这是 MC9S08AW60 系列在一般情况下的操作模式。在复位上升沿，若 BKGD/MS 为高电平，则进入此模式。在这种模式下，CPU 从内存执行代码，复位后，从内存 \$ FFFE : \$ FFFF 地址处开始执行。

3.4 激活后台模式

激活后台模式功能由 HCS08 内核中的后台调试控制器（BDC）管理。在软件开发期间 BDC 连同片内调试模块（DBG）一起提供各种方式来分析 MCU 操作。

进入激活后台模式是的五种方式：

- 当上升沿复位期间 BKGD /MS 引脚为低电平时
- 当通过 BKGD 引脚接收到 BACKGROUND 命令时
- 当 BGND 指令被执行时
- 当遇到 BDC 断点时
- 当遇到 DBG 断点时

在进入激活后台模式后，CPU 处于一个暂停状态，等待串行后台命令，而不是执行用户应用程序的指令。

后台调试命令的两种类型：

- 非插入命令，定义为可在用户程序运行时发出的命令。当 MCU 在运行模式下时，非插入命令可经由 BKGD 引脚发出。当 MCU 在后台调试模式下也可执行非插入命令。非插入命令包括：

- 存储器访问命令
- 存储器访问与状态命令
- BDC 寄存器访问命令
- BACKGROUND 命令
- 激活后台命令，只有当 MCU 在激活后台模式下才可以执行。激活的后台命令，包括命令：
 - 读/写 CPU 寄存器
 - 在一个时间单步跟踪用户程序指令
 - 退出激活后台模式，返回用户应用程序(GO)

激活后台模式通常用于 MCU 第一次在运行模式下运行之前，向 Flash 程序存储器写入引导程序或用户程序。当 MC9S08AW60 系列出 FREESCALE 半导体厂时，除非有特别指明，Flash 存储器默认为擦除的，因此在运行模式下没有程序可以执行，直到 Flash 存储器初次被编程。激活后台模式也可用于在 Flash 存储器已经写入程序后，对其擦除和重新写入程序。

有关激活后台模式的更多信息可见第 15 章“开发支持”。

3.5 等待模式

等待模式由执行 WAIT 指令进入。在程序中运行 WAIT 指令，使 CPU 处于低功耗状态，在该模式下，定时器没有关闭。当 CPU 的进入等待模式时，CCR 中的 I 位被清零，打开中断。当一个中断请求发生，CPU 退出等待模式，恢复进程，开始堆栈操作，产生了中断服务例程。

而 MCU 在等待模式，对可用的后台调试命令有一些限制。当 MCU 在等待模式下，只有 BACKGROUND 命令和存储器访问与状态命令可用。存储器访问与状态命令不容许存储器访问，但它们会报告一个错误，表明 MCU 处于停止或等待模式。BACKGROUND 命令可以用来将 MCU 从等待模式唤醒，进入激活后台模式。

3.6 停止模式

当系统选项寄存器中的 STOPE 位置位时，执行到 STOP 指令就进入两种停止模式的一种。在两种停止模式下，所有内部时钟被关闭。如果 STOPE 位在 CPU 执行 STOP 指令时没有置位，MCU 将不会进入停止模式，并强制产生一个非法指令复位。通过设置 SPMSC2 中的相应位来选择 STOP 模式。

HCS08 芯片是专为低电压操作（1.8 V 至 3.6 V 之间）设计的，还包括 STOP1 模式。MC9S08AW60 系列芯片，并不包括 STOP1 模式。

表 3-1 总结了 MCU 在每个停止模式下的行为。

表3-1 停止模式下的行为

模式	PPDC	CPU, 数字 外设, FLASH	RAM	ICG	ADC1	Regulator	I/O 引脚	RTI
STOP2	1	关闭	待命	关闭	禁止	待命	维持状态	可选
STOP3	0	待命	待命	关闭 ¹	可选	待命	维持状态	可选

¹ 晶体振荡器可以配置为运行在 stop3。请参阅 ICG 寄存器。

3.6.1 STOP2 模式

该 STOP2 模式提供了非常低的待机功耗、保持 RAM 的内容和所有 I/O 引脚的现状。进入 STOP2，用户必须执行 STOP 指令，选定 STOP2 (PPDC = 1) 并启用停止模式 (STOPE = 1)。此外，在停止模式 (LVDSE = 0 或 LVDE = 0) 下，LVD 绝不能允许操作。如果在停止模式下，LVD 被启用，则 MCU 执行 STOP 指令后就进入 STOP3，不论 PPDC 的状态。

进入 STOP2 模式之前，用户必须保存 I/O 端口寄存器的内容，以及任何其他映像存储寄存器，用户想要在离开 STOP2 后恢复它们，放在 RAM 中。退出 STOP2，引脚锁存开放之前，这些值可以由用户软件恢复。

当 MCU 处于 STOP2 模式，由稳压器供电的所有内部电路被关闭，除了 RAM。该稳压器处在一个低功耗待机状态，相当于 ADC。进入到 STOP2，I/O 引脚的状态被锁存。当进入 STOP2 模式和退出 STOP2 模式，该状态都被保持，直到逻辑 1 写入到 SPMSC2 的 PPDACK。

通过设置唤醒引脚： $\overline{\text{RESET}}$ 或 IRQ，或通过 RTI 中断来退出 STOP2。当 MCU 进入 STOP2，IRQ 的始终是低电平输入有效，不论在进入 STOP2 之前它是怎么配置的。

注：

虽然这 IRQ 引脚是自动设定为低电平输入有效，但与 IRQ 引脚相关的上拉是不会自动启用的。因此，如果一个外部上拉不使用，内部上拉必须通过设置 IRQSC 的 IRQPE 来启用。

唤醒了 STOP2 模式后，MCU 将启动作为上电复位 (POR)，除引脚状态外保持锁存。CPU 将采取复位向量。该系统 and 所有外设将在其预设的状态并必须初始化。

从 STOP2 唤醒后，SPMSC2 的 PPDF 位被置位。这个标志可能被用来指向用户代码进入一个 STOP2 恢复例程。PPDF 保持置位、I/O 引脚状态保持锁存，直到逻辑 1 写入 SPMSC2 的 PPDACK。

要保持被设定为通用 I/O 引脚的 I/O 状态，用户必须恢复 I/O 端口寄存器的内容，其已保存在 RAM 上，在写 PPDACK 位之前先写到端口寄存器。在写 PPDACK 之前，如果端口寄存器没有存储来自 RAM 的内容，当 I/O 引脚锁存开放时，寄存器位将假设其复位状态，且 I/O 引脚将改变其复位状态。

引脚被设定为外设 I/O，在写 PPDACK 位之前，用户必须重新设定外设模组接口引脚。在写 PPDACK 之前，如果外设模块未启用，当 I/O 锁存开放时，引脚将被其相关的端口控制寄存器控制。

3.6.2 STOP3 模式

在表 3-1 所示的条件下，执行 STOP 指令进入 Stop3 模式。所有的内部寄存器和逻辑的状态，RAM 的内容，和 I/O 引脚状态都被保持。

STOP3 可以退出，由断言 $\overline{\text{RESET}}$ ，或由以下某项中断源：实时中断 (RTI)，LVD，ADC，IRQ，或 KBI。

如果 STOP3 是以 $\overline{\text{RESET}}$ 引脚方式退出的，那么采取复位向量后，MCU 复位和操作会恢复。以某一内部中断源的方式退出，MCU 就采取适当的中断向量。

3.6.3 停止模式下激活的 BDM 使能

如果 BDCSCR 的 ENBDM 位被置位，则从运行模式进入激活后台模式被允许。本数据表的第 15 章“开发支持”中描述了该寄存器。当 CPU 执行 STOP 指令时，如果 ENBDM 被置位，这样当 MCU 进入到停止模式后后台调试逻辑的系统时钟仍然在工作，所以后台调试

通信仍然是可能的。此外，稳压器没有进入低功耗待机状态，而是保持满负荷工作中。如果用户试图进入 ENBDM 置位的 STOP2，MCU 将不会进入 STOP3。

大部分后台命令在停止模式下是无效的。存储器访问与状态命令不允许存储器存取，但它们会报告一个错误，表明 MCU 在停止或等待模式。如果 ENBDM 置位，BACKGROUND 指令可以用于将 MCU 从停止模式中唤醒进入激活后台模式。一旦进入后台模式后，所有激活后台命令都可用。表 3-2 总结了当允许进入后台模式时，在停止模式下的 MCU 的行为。

表3-2 停止模式下BDM使能时MCU行为

模式	PPDC	CPU, 数字外设, FLASH	RAM	ICG	ADC1	电压调节器	I/O 引脚	RTI
Stop3	0	待命	待命	激活	可选	激活	维持状态	可选

3.6.4 停止模式下 LVD 使能

当供应电压下降到 LVD 电压值以下时，LVD 系统能够产生中断或复位。如果通过设置 LVDE 和 LVDSE 位使得停止模式下的 LVD 有效，则在停止模式下电压调节器仍然保持工作。LVD 启用停止模式下如果用户试图进入 STOP2，MCU 将不会进入 STOP3。表 3-3 总结了 MCU 在启用 LVD 的停止模式下的行为。

表3-3 启用LVD的STOP模式下MCU的行为

模式	PPDC	CPU, 数字外设, FLASH	RAM	ICG	ADC1	电压调节器	I/O 引脚	RTI
Stop3	0	待命	待命	关闭 ¹	可选	激活	维持状态	可选

¹ 晶体振荡器可以配置为运行在 stop3。请参阅 ICG 寄存器。

3.6.5 停止模式下片内外设模块

当 MCU 进入任何 STOP 模式，系统时钟到内部外设模块都被关闭。甚至在例外情况下（ENBDM = 1），后台调试逻辑的时钟继续运作，外设系统时钟停止，以降低电力消耗。停止模式下系统行为的具体信息请参阅 3.6.1，“STOP2 模式“，和 3.6.2，“STOP3 模式“。表 3-4 给出了停止模式下的行为。

表3-4 STOP模式下MCU的行为

外设	模式	
	Stop2	Stop3
CPU	关闭	待命
RAM	待命	待命
FLASH	关闭	待命
并行端口寄存器	关闭	待命
ADC1	关闭	可选 ¹
ICG	关闭	可选 ²
IIC	关闭	待命
KBI	关闭	可选 ³
RTI	可选 ⁴	可选 ⁴
SCI	关闭	待命
SPI	关闭	待命
TPM	关闭	待命
稳压器	待命	待命
I/O 引脚	维持状态	维持状态

1、要求异步 ADC 的时钟和 LVD 必须启用，否则待命。

2、ICSC1 的 OSCSTEN 置位，否则待命。高频率范围(ICSC2 的 RANGE 置位)要求 LVD 也允许进入 STOP3。

3、STOP3 期间，KBI 引脚允许继续有中断源功能，能够从 STOP3 唤醒 MCU。

4、RTI 可以带有内部 RTI 的时钟源（SRTISC 的 RTICKS = 0，）运行在 STOP2 或 STOP3 下。RTI 也可以带有外部时钟源（RTICKS = 1 和 OSCSTEN = 1）运行在 STOP3 下。

第四章 存储器

4.1 MC9S08AW60系列存储器映像

图 4-1 给出了 MC9S08AW60 和 MC9S08AW48 MCUs 的存储器映像。图 4-2 给出了 MC9S08AW32 和 MC9S08AW16 MCUs 的存储器映像。MCUs 的 MC9S08AW60 系列产品中的片内存储器包括 RAM，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三类：

- 直接页寄存器（\$0000-\$006F）
- 高页寄存器（\$1800-\$185F）
- 非易失性寄存器（\$FFB0-\$FFBF）

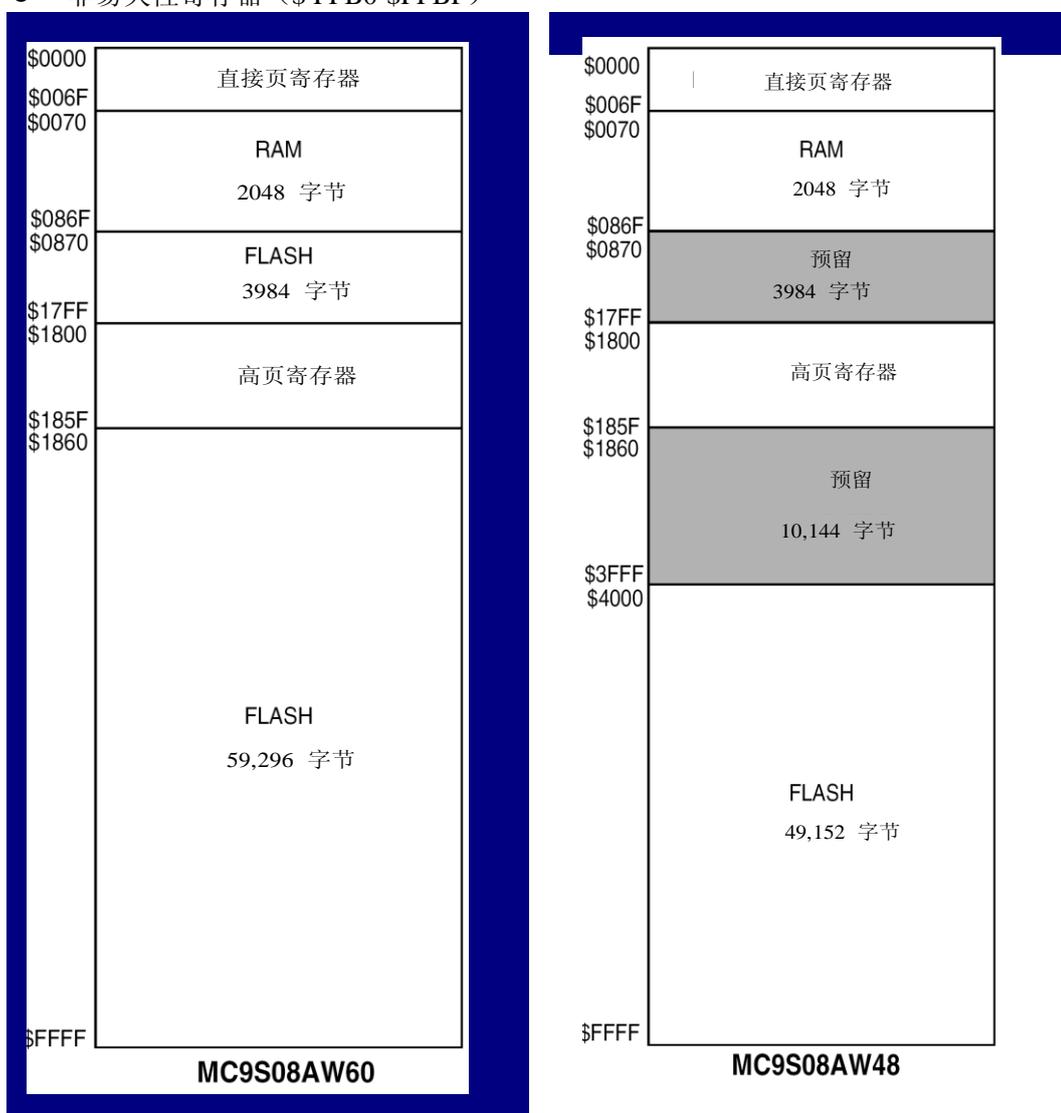


图4-1 MC9S08AW60和MC9S08AW48 MCUs的存储器映像

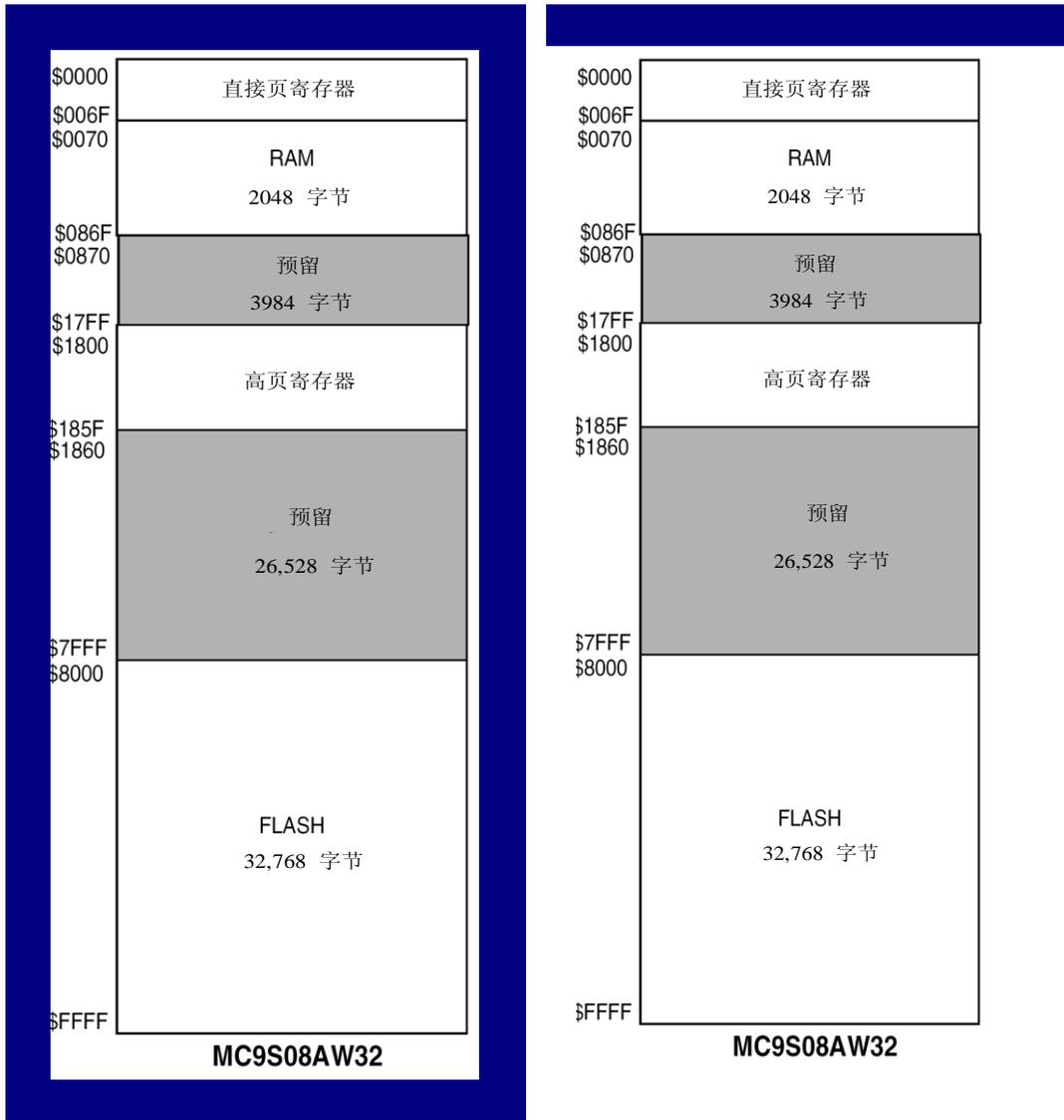


图4-2 MC9S08AW32和MC9S08AW16 MCUs的存储器映像

4.1.1 复位和中断向量分配

表 4-1 为复位和中断向量地址的分配情况。该表中使用的向量名为飞思卡尔半导体提供的 MC9S08AW60 系列通用文件中使用的标签。关于复位，中断，中断优先级，和本地掩码的更多细节，请参阅第 5 章，“复位，中断，系统配置”。

表4-1 复位和中断向量

地址（高/低）	向量	向量名
\$FFC0:FFC1		
↑	未使用的向量空间 (用于用户程序)	
\$FFCA:FFCB	RTI	Vrti
\$FFCC:FFCD	RTI	Vrti
\$FFCE:FFCF		
\$FFD0:FFD1	ADC1 转换	Vadc1

\$FFD2:FFD3	KBII	Vkeyboard1
\$FFD4:FFD5	SCI2 发送	Vsci2tx
\$FFD6:FFD7	SCI2 接收	Vsci2rx
\$FFD8:FFD9	SCI2 错误	Vsci2err
\$FFDA:FFDB	SCI1 发送	Vsci1tx
\$FFDC:FFDD	SCI1 接收	Vsci1rx
\$FFDE:FFDF	SCI1 错误	Vsci1err
\$FFE0:FFE1	SPI1	Vspi1
\$FFE2:FFE3	TPM2 溢出	Vtpm2ovf
\$FFE4:FFE5	TPM2 通道 1	Vtpm2ch1
\$FFE6:FFE7	TPM2 通道 0	Vtpm2ch0
\$FFE8:FFE9	TPM1 溢出	Vtpm1ovf
\$FFEA:FFEB	TPM1 通道 5	Vtpm1ch5
\$FFEC:FFED	TPM1 通道 4	Vtpm1ch4
\$FFEE:FFEF	TPM1 通道 3	Vtpm1ch3
\$FFF0:FFF1	TPM1 通道 2	Vtpm1ch2
\$FFF2:FFF3	TPM1 通道 1	Vtpm1ch1
\$FFF4:FFF5	TPM1 通道 0	Vtpm1ch0
\$FFF6:FFF7	ICG	Vicg
\$FFF8:FFF9	低电压检测	Vlvd
\$FFFA:FFFB	IRQ	Virq
\$FFFC:FFFD	SWI	Vswi
\$FFFE:FFFF	Reset	Vreset

4.2 寄存器地址和位分配

MC9S08AW60 系列产品中的寄存器可分为以下三组：

- 直接页寄存器，位于存储器映像的前 112 个位置上，这些寄存器可以通过高效的直接寻址方式指令访问。
- 高页寄存器不经常使用，因此位于存储器映像中\$1800 以上。在直接页中，为频繁使用的寄存器和变量留出了更多的空间。
- 非易失性寄存器，由 FLASH 存储器中\$FFB0-\$FFBF 之间 16 个位置的地址段组成。非易失性寄存器地址包括：
 - 在复位时，加载到工作寄存器的三个值
 - 一个 8 字节的后门对比密钥，可选择为用户拥有控制安全内存的访问权限。

由于非易失性寄存器地址在 FLASH 存储器中，它们必须像其它 FLASH 存储器地址一样进行擦除和编程。

直接页寄存器可以通过高效的直接寻址方式指令访问。位操纵指令可用于访问任何直接页寄存器中的任何位。表 4-2 总结了所有用户可访问的直接页寄存器和控制位。

表 4-2 所列的直接页寄存器可以使用更高效的直接寻址方式，只需要地址的较低字节。正因为如此，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。

表4-2 直接页寄存器汇总 (第1页, 共3页)

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$0000	PTAD	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
\$0001	PTADD	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
\$0002	PTBD	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
\$0003	PTBDD	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
\$0004	PTCD	0	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
\$0005	PTCDD	0	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
\$0006	PTDD	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
\$0007	PTDDD	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
\$0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
\$0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
\$000A	PTFD	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
\$000B	PTFDD	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
\$000C	PTGD	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
\$000D	PTGDD	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
\$000E– \$000F	预留	—	—	—	—	—	—	—	—
\$0010	ADC1SC1	COCO	AIEN	ADCO	ADCH				
\$0011	ADC1SC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
\$0012	ADC1RH	0	0	0	0	0	0	ADR9	ADR8
\$0013	ADC1RL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
\$0014	ADC1CVH	0	0	0	0	0	0	ADCV9	ADCV8
\$0015	ADC1CVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
\$0016	ADC1CFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
\$0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
\$0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
\$0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
\$001A– \$001B	预留	—	—	—	—	—	—	—	—
\$001C	IRQSC	0	0	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
\$001D	预留	—	—	—	—	—	—	—	—
\$001E	KBI1SC	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
\$001F	KBI1PE	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
\$0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
\$0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
\$0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
\$0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
\$0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
\$0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0

表4-2 直接页寄存器汇总（第3页，共3页）

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$0050	SPI1C1	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
\$0051	SPI1C2	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
\$0052	SPI1BR	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
\$0053	SPI1S	SPRF	0	SPTEF	MODF	0	0	0	0
\$0054	预留	0	0	0	0	0	0	0	0
\$0055	SPI1D	Bit 7	6	5	4	3	2	1	Bit 0
\$0056-	预留	—	—	—	—	—	—	—	—
\$0057		—	—	—	—	—	—	—	—
\$0058	IIC1A	ADDR							0
\$0059	IIC1F	MULT			ICR				
\$005A	IIC1C	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
\$005B	IIC1S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	FXAK
\$005C	IIC1D	数据							
\$005D-	预留	—	—	—	—	—	—	—	—
\$005F		—	—	—	—	—	—	—	—
\$0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
\$0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
\$0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8
\$0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
\$0065	TPM2C0SC	CHOF	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
\$0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
\$0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
\$0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
\$006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
\$006B-	预留	—	—	—	—	—	—	—	—
\$006F		—	—	—	—	—	—	—	—

表 4-3 中列出的高页寄存器的访问频率比其它 I/O 和控制寄存器低很多，因此存放在可直接寻址的内存空间外，从 \$1800 开始。

表4-3 高页寄存器汇总（第1页，共2页）

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$1800	SRS	POR	PIN	COP	ILOP	0	ICG	LVD	0
\$1801	SBD FR	0	0	0	0	0	0	0	BDFR
\$1802	SOPT	COPE	COPT	STOPE	—	0	0	—	—
\$1803	SMCLK	0	0	0	MPE	0	MCSEL		
\$1804— \$1805	预留	—	—	—	—	—	—	—	—
\$1806	SDIDH	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
\$1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
\$1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
\$1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0 ¹	BGBE
\$180A	SPMSC2	LVWF	LVWACK	LVDV	LVWV	PPDF	PPDACK	—	PPDC
\$180B— \$180F	预留	—	—	—	—	—	—	—	—
\$1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
\$1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
\$1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
\$1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
\$1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
\$1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
\$1816	DBG C	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
\$1817	DBG T	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
\$1818	DBG S	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
\$1819— \$181F	预留	—	—	—	—	—	—	—	—
\$1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
\$1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
\$1822	预留	—	—	—	—	—	—	—	—
\$1823	FCNFG	0	0	KEYACC	0	0	0	0	0
\$1824	FPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FDPIS
\$1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
\$1826	FCMD	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
\$1827— \$183F	预留	—	—	—	—	—	—	—	—
\$1840	PTAPE	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
\$1841	PTASE	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
\$1842	PTADS	PTADS7	PTADS6	PTADS5	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
\$1843	预留	—	—	—	—	—	—	—	—
\$1844	PTBPE	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
\$1845	PTBSE	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0

表4-3 高页寄存器汇总（第2页，共2页）

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$1846	PTBDS	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
\$1847	预留	—	—	—	—	—	—	—	—
\$1848	PTCPE	0	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
\$1849	PTCSE	0	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
\$184A	PTCDS	0	PTCDS6	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
\$184B	预留	—	—	—	—	—	—	—	—
\$184C	PTDPE	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
\$184D	PTDSE	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
\$184E	PTDDS	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
\$184F	预留	—	—	—	—	—	—	—	—
\$1850	PTEPE	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
\$1851	PTESE	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
\$1852	PTEDS	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
\$1853	预留	—	—	—	—	—	—	—	—
\$1854	PTFPE	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
\$1855	PTFSE	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
\$1856	PTFDS	PTFDS7	PTFDS6	PTFDS5	PTFDS4	PTFDS3	PTFDS2	PTFDS1	PTFDS0
\$1857	预留	—	—	—	—	—	—	—	—
\$1858	PTGPE	0	PTGPE6	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
\$1859	PTGSE	0	PTGSE6	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
\$185A	PTGDS	0	PTGDS6	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
\$185B– \$185F	预留	—	—	—	—	—	—	—	—

1 这个预留位必须总是被写入 0

表 4-4 列出的非易失性 FLASH 寄存器，位于 FLASH 中。这些寄存器包括 1 个 8 字节的后门密钥，可用于访问安全的内存资源。在复位过程中，FLASH 中非易失性寄存器区域的 NVPROT 和 NVOPT 内容被转移到高页寄存器中相应的 FPROT 和 FOPT 工作寄存器中，以控制安全性和块保护选项。

表4-4 非易失性寄存器汇总

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$FFB0 – \$FFB7	NVBACKKEY	8 字节比较密钥							
\$FFB8 – \$FFBB	预留	—	—	—	—	—	—	—	—
\$FFBC	预留存储 250kHz 的 ICGTRM 值	—	—	—	—	—	—	—	—
\$FFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
\$FFBE	预留存储 243kHz 的 ICGTRM 值	—	—	—	—	—	—	—	—
\$FFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

如果密钥启用（KEYEN）位为 1，那么 8 字节对比密钥可用于暂时脱离内存安全的限制。这种密钥机制只能通过安全内存中运行的用户代码访问。（安全密钥不能通过后台调试命令直接输入。）这个安全密钥可通过将 KEYEN 位设为 0 来完全禁用。如果这个安全密钥被禁用，那么脱离安全限制的唯一方式是整体擦除 Flash（通常通过后台调试接口）

并确认 Flash 是空的。为了避免在下一次复位后返回到安全模式，应该将安全位（SEC）设置为非安全状态（1: 0）。

4.3 RAM

MC9S08AW60 系列包括静态 RAM。RAM 中 \$0100 以下的地址可以使用更高效的直接寻址模式访问，而这一区域中的任何单一比特可以通过位操作指令（BCLR、BSET、BRCLR 和 BRSET）访问。首选的方式是在这一区域中查找 RAM 最频繁访问的程序变量。

在 MCU 处于低功耗等待、Stop2 或 Stop3 模式时，RAM 会保留数据。加电启动时，RAM 中的内容不会被初始化。如果电源电压没有降低到 RAM 保留 (VRAM) 的最低值以下，RAM 数据就不会受到复位的任何影响。

为了实现与 M68HC05 MCU 的兼容性，HCS08 会将栈指针复位为 \$00FF。在 MC9S08AW60 系列中，最好的方法通常是将栈指针重新初始化到 RAM 顶部，以便使经常被访问的 RAM 变量和位可寻址程序变量处于直接寄存器。复位初始化程序（其中的 RamLast 等于飞思卡尔半导体等同文件中 RAM 的最高地址）中包含以下两个指令序列。

```
LDHX    #RamLast+1    ;point one past RAM
TXS                                ;SP<-(H:X-1)
```

在启用了安全性的情况下，RAM 被认为是一种安全的内存资源，不能通过 BDM 或从非安全内存中执行代码来访问。若欲了解有关安全特性的更详尽描述，请参见 4.5 “安全性”。

4.4 FLASH

FLASH 存储器主要用于保存程序和数据。在线编程使正在运行的程序和数据可以在应用产品的最终组装完成后分别上载到 Flash 中。我们可以通过单线后台调试接口对阵列进行编程。由于 FLASH 擦除和编程操作不需要特殊的电压，所以也可以通过其他软件控制的通信路径来实现应用编程。有关在线和应用内编程的更详尽描述，请参见“HCS08 系列参考手册，第 1 卷”（飞思卡尔半导体文件编号 HCS08RMv1/D）。

4.4.1 特点

FLASH 存储器功能包括：

- FLASH 大小
 - MC9S08AW60- 63280 字节（124 页，每页 512 字节）
 - MC9S08AW48- 49152 字节（96 页，每页 512 字节）
 - MC9S08AW32- 32768 字节（64 页，每页 512 字节）
 - MC9S08AW16- 16384 字节（32 页，每页 512 字节）
- 单电源程序和擦除
- 命令界面的快速程序和擦除操作
- 用于快速编程和擦除操作的命令接口
- 灵活的块保护
- FLASH 和 RAM 的安全特性
- 自动关闭电源以**获得低频读取**

4.4.2 编程和擦除时间

在接受任何编程或擦除命令前，必须通过写 Flash 和 EEPROM 时钟分频寄存器 (FCDIV) 以将 Flash 和 EEPROM 模块的内部时钟设置为 150 kHz ~ 200 kHz 之间的频率 (fFCLK) (请参见 4.6.1 “FLASH 时钟分频寄存器 (FCDIV)”)。这个寄存器只能写入一次，因此这一写入操作通常是在复位初始化过程中执行的。如果 FSTAT 寄存器的 FACCERR 访问错误标志位被置位，FCDIV 不能被写入。用户必须确保在写入 FCDIV 寄存器之前没有设置 FACCERR。命令处理器使用最终时钟 (1/fFCLK) 的一个周期来对编程和擦除脉冲定时。命令处理器利用这些定时脉冲的一个整数来完成编程或擦除命令。

表 4-5 给出了编程和擦除时间。总线时钟频率和 FCDIV 决定 FCLK 的频率 (fFCLK)。一个 FCLK 周期为 $t_{FCLK} = 1/f_{FCLK}$ 。定时器显示为多个 FCLK 循环和一个绝对时间 ($t_{FCLK} = 5 \mu s$)。显示的编程和擦除时间包括命令状态机的开销及编程和擦除电压的启用及禁用的时间。

表4-5 编程和擦除时间

参数	FCLK 循环	FCLK = 200 kHz 时的时间
字节程序	9	45 μs
突发程序	4	20 μs^1
页擦除	4000	20 ms^2
整体擦除	20,00	100 ms^2

1 不包括开始/结束开销

2 因为页和整体擦除时间可长于 COP 看门狗超时时间，在任何软件擦除例程中，COP 应被服务。

4.4.3 编程和擦除命令的执行

下面列出执行任何命令的步骤。在执行命令之前任何错误标志都需要被清除。命令执行的步骤是：

1、将一个数据值写入到 Flash 阵列中的一个地址中。该地址和写入的数据信息被锁定到 Flash 接口上。这一写入操作是任何命令序列中要求的第一步。对于擦除和空白检查命令，这些数据的值并不重要。对于页擦除命令，地址可以是将要擦除的 Flash512 字节页面中的任何地址。对于整体擦除和空检查命令，地址可以是 Flash 存储器中的任何地址。512 字节的整个页面是 FLASH 能被擦除的最小块。在 60K 的版本，有两个实例，用户可用的块的大小少于 512 字节：RAM 以下的第一页，和在高页寄存器下的第一页。这些页分别被 RAM 和高页寄存器页重叠。

注：

在擦除操作成功之后，不要对 FLASH 中的任意字节编程超过一次。如果没有开始对 FLASH 存储器的整体擦除或擦除字节所在的页面，对已经编程的字节重新编程是不被允许的。编程前没先擦除可能会扰乱储存在 FLASH 中的数据。

2、将命令代码写入到 FCMD 中。6 个有效的命令分别是空白检查 (\$05)、字节编程 (\$20)、突发编程 (\$25)、分区擦除 (\$40)、整体擦除 (\$41) 和分区擦除终止 (\$47)。命令代码被锁定到命令缓冲器中。

3、写一将一个 1 写入到 FSTAT 中的 FCBEF 位上，以清除 FCBEF 并发起命令 (包括其地址和数据信息)。

在写内存阵列之后到写 1 用于清除 FCBEF 并发起完整命令之前的任何时候，可以通过向 FCBEF 中写入一个“0”，来手工终止部分命令顺序。以这种方式终止一个命令会设置 FACCERR 访问错误标记，而这个标记必须在开始一个新命令之前清除掉。

整个过程必须遵守严格监控的流程，否则命令将不会被接受。通过这种方式可以最大限度地降低无意中修改 FLASH 存储器内容的可能性。命令完整标记（FCCF）用于指示一条命令是否完整。要启动命令，必须通过清除 FCBEF 来使命令序列完整。图 4-3 是除突发编程和分区擦除终止以外的所有命令的执行流程。在使用任何 FLASH 命令之前，FCDIV 寄存器必须初始化。复位后，这只能执行一次。

4.4.4 执行突发编程

突发编程命令用来对连续的字节进行编程，相比较标准编程命令需要更少的时间。因为不需要屏蔽两次编程操作之间加在 Flash 阵列的高电压，突发编程命令是有可能的。通常情况下，当发起一个编程或擦除命令，必须启用与 Flash 相关的一个内部电荷泵用于为阵列提供高电压。命令执行完成后，该电荷泵会被关闭。发出突发编程命令后，电荷泵在以下两种条件下会被开启而且在突发编程操作完成后将保持开启状态：

- 在本次操作之前下次本次命令已经排队等候。
- 下一次连续地址选择的字节与本次编程的当前字节在同一物理行。Flash 存储器中一行由 64 个字节组成。一行中的一个字节由地址 A5 到 A0 选择。当 A5 到 A0 都为 0，则为一个新的行。

在突发模式中编程一系列连续字节中的第一个所花的时间与正常模式中编程的一个字节的时间是相同的，假设满足上述两个条件，后续字节将以突发模式编程。在下一个连续的地址是另一行的情况下，字节的编程时间将会是标准时间而不是突发时间。这因为加在阵列上的高电压将被屏蔽然后重新开启。如果在第一次命令结束之前新的命令没有排在队列中，电荷泵将被屏蔽，阵列上的高电压将被移除。

4.4.5 访问错误

无论何时只要违反了命令执行协议，将产生访问错误。

任何下列特别的动作将会导致 FSTAT 中的访问错误标志(FACCERR)置 1。在任何命令执行之前，FACCERR 位必须通过对 FSTAT 中的 FACCERR 写 1 的方式清除。

- 在通过写 FCDIV 寄存器设置内部 FLASH 时钟频率之前，写 FLASH 地址
- 写 FLASH 地址的同时 FCBEF 没有被设置(直到命令缓存清空,才能开始一个新命令.)
- 在开始上一次命令之前,第二次写 FLASH 地址(每一个命令只对 FLASH 写一次.)
- 在开始上一次命令之前,第二次写 FCMD(每一次命令只对 FCMD 写一次.)
- 写 FLASH 地址之后写任何 FLASH 控制寄存器而不是 FCMD
- 写(\$05, \$20, \$25, \$40, or \$41)之外的任何命令代码到 FCMD
- 在写命令到 FCMD 后访问(读或者写)任何 FLASH 控制寄存器而不是在写命令到 FCMD 之后写 FSTAT(为了清除 FCBEF 和开始命令)
- 当正在执行编程和擦除命令时 MCU 进入停止模式.(命令失败.)
- 当 MCU 处于保护状态,用背景调试命令写字节编程,突发编程或者页面擦除命令代码(\$20,\$25, or \$40). (当 MCU 处于保护状态,背景调试控制只能执行空检查和整体擦除命令.)

- 为了取消部分命令对 FCBEF 写 0。

4.4.6 FLASH 块保护

块保护特点防止了 FLASH 的保护块被编程和擦除改变。块保护由 FLASH 保护寄存器 (FPROT) 控制。当使能, 块保护从任何低于 FLASH 最后字节, \$FFFF, 的任何 512 个字节边界开始。(参看 4.6.4, “FLASH 保护寄存器(FPROT 和 NVPROT)”)。

退出复位后, FPROT 用于位于 FLASH 存储器中非易失寄存器块的 NVPROT 位置的内容加载。FPROT 不能从应用软件直接改变, 所以一个失控的程序不能改变块保护的设置。因为 NVPROT 在 FLASH 最后 512 字节之中, 如果任何数量的存储器被保护, NVPROT 自己也被保护, 并且不能被应用软件(有意的或无意的)改变。背景调试命令能够写 FPROT, 这是一种擦除和编程保护的 FLASH 存储器的方法。

图 4-5 中举例说明了块保护机制。FPS 位用作未保护存储区最后一个地址的高位。地址由如图所示串连的 FPS7:FPS1 和逻辑 1 组成。例如, 为了保护存储器中最后的 8192 个字节(地址从 \$E000 到 \$FFFF), FPS 位需设置为 1101 111, 产生的值 \$DFFF 是存储器中最后未保护的地址。除了对 FPS 位编程正确的值, FPDIS(NVPROT 中的 0 位)必须被编程为逻辑 0 以确保块保护。因此值 \$DE 必须编程到 NVPROT 以确保地址 \$E000 到 \$FFFF。

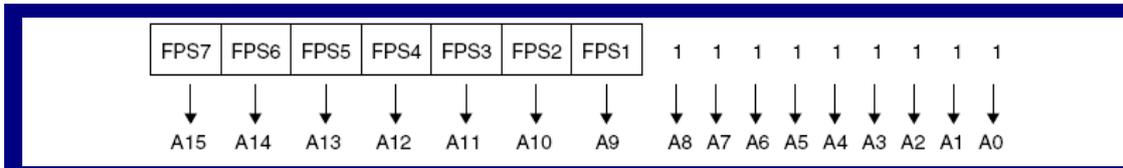


图4-5 块保护机制

块保护的一个用处是块保护在 FLASH 中用于 bootloader 程序的区域, 之后这个 bootloader 程序能够被用于擦除和重编程 FLASH 剩下的存储器。由于 bootloader 被保护, 即使如果 MCU 在一次擦除和重编程操作之中掉电, bootloader 仍然保持完整。

4.4.7 矢量变向

无论何时使能块保护, 复位和中断矢量将会被保护。矢量变向允许用户不用未保护 bootloader 和复位矢量空间就能修改中断矢量信息。矢量变向通过对位于 \$FFBF 地址的 NVOPT 中 FNORED 位编程为 0 使能。为产生变向, 通过对位于 \$FFBD 的 NVPROT 编程使至少一些而不是全部 FLASH 存储区受到块保护。所有的中断矢量(位于存储器 \$FFC0 - \$FFFD)变向, 而复位矢量(位于存储器 \$FFFE - \$FFFF)则不会变向。

例如, 如果 FLASH 的 512 字节受到保护, 保护的地址是从 \$FE00 到 \$FFFF。中断矢量(\$FFC0 - \$FFFD)变向到位于 \$FDC0 - \$FDFD。现在举例来说, 如果产生一个 SPI 中断, \$FDE0:FDE1 中的值而不是 \$FFE0:FFE1 中的值用于矢量。这允许用户用新的包括新的中断矢量值的代码重编程 FLASH 中的未保护部分, 同时让包括默认的矢量位置的保护区域不会改变。

4.5 保密性

MC9S08AW60 系列包括防止未经授权的对 FLASH 和 RAM 访问的电路。当采取保密措施, FLASH 和 RAM 被认为是保密资源。直接页寄存器, 高页寄存器和背景调试控制器被认为是非保密资源。在保密存储器中执行的程序可正常访问任何 MCU 存储器和资源。试图

用非保密存储空间中执行的程序或通过背景调试接口对一个保密存储器的访问将会被屏蔽(写被忽略,读返回0)。

是否采取保密措施取决于 FOPT 寄存器中两个非易失位(SEC01:SEC00)的状态。在复位过程中,非易失的 NVOPT 的内容从 FLASH 复制到在高页面寄存器空间的正在工作的 FOPT 寄存器。用户能通过编程 NVOPT 来采取保密措施,这也能在对 FLASH 存储器编程的同时完成。1:0 状态解除保密措施,其它三种组合则采取保密措施。注意擦除态(1:1)也让 MCU 保密。在开发过程中,无论何时只要 FLASH 被擦除,一个好的经验是立即对 NVOPT 中的 SEC00 位编程为 0,使 SEC01:SEC00 = 1:0。这让 MCU 在随后的复位后保持非保密状态。

当 MCU 处于保密态,片内调试模块将不能被使能。个别的背景调试控制器仍能用于背景存储器访问命令,但是除非在复位的上升沿保持 BKGD/MS 为低,MCU 不能进入活动的背景模式。

用户能通过一个 8 位的后门保密钥匙来选择允许或者不允许保密解锁机制。如果 NVOPT/FOPT 中的非易失 KEYEN 位为 0,后门钥匙被屏蔽,并且如果没有完全擦除所有的 FLASH 位置,则没有办法解除保密措施。如果 KEYEN 为 1,一个保密的用户程序可通过如下的方法暂时解除保密措施:

- 1.对 FCNFG 寄存器中 KEYACC 写 1。这让 FLASH 模块对后门比较钥匙位置(NVBACKKEY 到 NVBACKKEY+7)的写操作解释为与钥匙相比较的值,而不是 FLASH 编程或者擦除命令的第一步。

- 2.写用户进入钥匙的值到 NVBACKKEY 到 NVBACKKEY+7 的位置。这些写必须按照开始于 NVBACKKEY,结束于 NVBACKKEY+7 上的值这样的顺序。这些写中将不会用到 STHX,因为这些写不能在相邻总线周期中完成。用户软件通常从 MCU 系统外面通过一种通讯接口,例如串口,获得钥匙代码。

- 3.写 FCNFG 寄存器中的 KEYACC 为 0。如果刚写下的 8 字节钥匙与存储在 FLASH 中的钥匙相匹配,SEC01:SEC00 自动改变为 1:0,而且直到下次复位保密措施才会被解除。

保密钥匙只能从保密性的存储器(RAM 或者 FLASH)写入,所没有保密性的用户程序的配合就不能通过背景命令进入。

后门比较钥匙(NVBACKKEY 到 NVBACKKEY+7)位于 FLASH 存储位置,其在非易失寄存器空间,所以用户能正确的编程这些位置,同时他们能编程其他 FLASH 存储器的位置。作为复位和中断矢量的非易失寄存器在 FLASH 中的同一个 512 字节块,所以块保护的空間也被后门比较钥匙块保护。块保护不能从用户应用程序改变,所以如果矢量空间处于块保护,后门保密钥匙机制不能永久的改变块保护,保密性设置和后门钥匙。

保密措施总是按照如下步骤通过背景调试接口被解除:

- 1.通过写 FPROT 屏蔽任何块保护。FPROT 仅能用背景调试命令而不是应用软件写。
 - 2.如果必要,则整体擦除 FLASH。
 - 3.空检查 FLASH。如果 FLASH 被完全擦除,直到下次复位未采取保密措施。
- 为避免下次复位后返回保密模式,编程 NVOPT 使 SEC01:SEC00 = 1:0。

4.6 FLASH寄存器及控制位

FLASH 模块在高页面寄存器空间中有 9 个 8 位寄存器,两个位置(NVOPT, NVPROT)在 FLASH 存储器的非易失寄存器空间在复位中被复制到相对应的高页面控制寄存器(FOPT, FPROT)。FLASH 存储器中还有一个 8 字节比较钥匙。所有 FLASH 寄存器的绝对值地址分

配参考表 4-3 和表 4-4.本节仅仅通过它们的名字引用寄存器和控制位。飞思卡尔半导体提供等同的或者头文件用来翻译这些名字为适当的绝对地址。

4.6.1 FLASH 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 4-6 给出了 FLASH 时钟分频寄存器 FCDIV 的格式。表 4-6 为 FCDIV 寄存器的字段描述。

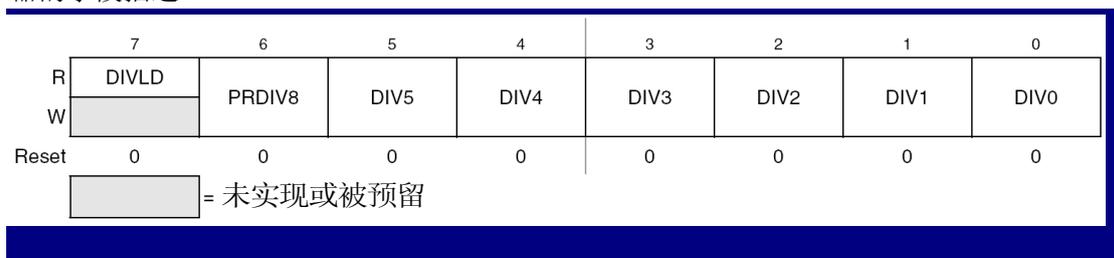


图4-6 FLASH时钟分频寄存器(FCDIV)

表4-6 FCDIV寄存器字段描述

字段	描述
7 DIVLD	<p>Divisor 加载状态标识 — 当置 1 时, 这个只读状态标记指出 FCDIV 寄存器自从复位后已有写入。复位会清除该位而且第一次写入该寄存器的操作将导致该位被置 1, 不管写入什么数据。</p> <p>0 FCDIV 自复位后没有写入; Flash 和 EEPROM 的擦除和编程操作被禁止。</p> <p>1 FCDIV 自复位后已写入; Flash 和 EEPROM 的擦除和编程操作已开启。</p>
6 PRDIV8	<p>Flash 时钟 8 预分频 (分频)</p> <p>0 Flash 时钟分频器的时钟输入为总线速率时钟。</p> <p>1 Flash 时钟分频器的时钟输入为总线速率时钟除以 8。</p>
5:0 DIV[5:0]	<p>Divisor Flash 时钟分频器 — Flash 时钟分频器用总线速率时钟 (如果 PRDIV8 = 1, 则用总线速率时钟除以 8) 除以 6 位 DIV5: DIV0 字段中的值再加 1。最后的内部 Flash 时钟的频率必须在 200 kHz 到 150 kHz 的范围内, 这样才能使 Flash 和 EEPROM 正常运行。编程/擦除定时脉冲为这个内部 Flash 时钟的一个循环, 这相当于 5 μs 到 6.7 μs。自动编程逻辑使用这些脉冲的整数来完成擦除或编程操作。请参见等式 4-1、等式 4-2 和表 4-6。</p>

$$\text{if PRDIV8} = 0 \text{ — } f_{\text{FCLK}} = f_{\text{Bus}} \div (\text{DIV} + 1) \quad \text{Eqn. 4-1}$$

$$\text{if PRDIV8} = 1 \text{ — } f_{\text{FCLK}} = f_{\text{Bus}} \div (8 \times (\text{DIV} + 1)) \quad \text{Eqn. 4-2}$$

表 4-7 说明了在所选总线频率时 PRDIV8 和 DIV 是什么值。

表4-7 FLASH时钟分频器设置

f_{BUS}	PRDIV8 (二进制)	DIV (十进制)	f_{FCLK}	编程/擦除时序脉冲 (最小 5 μs, 最大 6.7 μs)
20 MHz	1	12	192.3 kHz	5.2 μs
10 MHz	0	49	200 kHz	5 μs
8 MHz	0	39	200kHz	5 μs
4 MHz	0	19	200kHz	5 μs

2 MHz	0	9	200kHz	5 μs
1 MHz	0	4	200kHz	5 μs
200 kHz	0	0	200kHz	5 μs
150 kHz	0	0	150 kHz	6.7 μs

4.6.2 FLASH 选项寄存器 (FOPT 和 NVOPT)

在复位中，在非易失位置中 NVOPT 的内容从 FLASH 复制到 FOPT。第 2 至 5 位不使用并总是读出 0。在任何时间这个寄存器都可能被读出，但写入没有意义或影响。要改变这个寄存器的值，照常擦除和重编程 FLASH 存储器中 NVOPT 位置，然后发起一个新的 MCU 复位。



图4-7 FLASH选项寄存器(FOPT)

表4-8 FOPT寄存器字段描述

字段	描述
7 KEYEN	<p>使能后门钥匙机制 — 当这位为 0，后门钥匙机制不能使用解除保密措施。后门钥匙机制仅能从用户(保密的)固件访问。BDM 命令不能用来写将解除后门钥匙的钥匙比较值。更多关于后门钥匙机制的细节，参看 4.5，“保密性”。</p> <p>0 不允许后门钥匙访问。</p> <p>1 如果用户固件写了一个 8 字节的值，匹配非易失，后门钥匙(按顺序 NVBACKKEY 到 NVBACKKEY+7)，直到下次 MCU 复位，保密措施被暂时取消。</p>
6 FNORED	<p>屏蔽矢量变向 — 当此位为 1，屏蔽矢量变向。</p> <p>0 使能矢量变向。</p> <p>1 屏蔽矢量变向。</p>
5:0 SEC0[1 :0]	<p>保密状态代码 — 这两位域确定了如表 4-9 所示的 MCU 保密状态。当 MCU 处于保密，RAM 和 FLASH 存储器中的内容不能通过从任何非保密源的包括背景调试接口的指令来访问。关于保密的更多细节，参看 4.6，“保密性”。</p>

表4-9 保密状态

SEC01 :SEC00	描述
0:0	保密
0:1	保密
1:0	非保密
1:1	保密

在成功的后门钥匙进入或者成功的 FLASH 空检查之后,SEC01:SEC00 改变为 1:0。

4.6.3 FLASH 配置寄存器 (FCNFG)

在任何时间位 7:5 可以被读或写。位 4:0, 总是读出 0 但不能写。

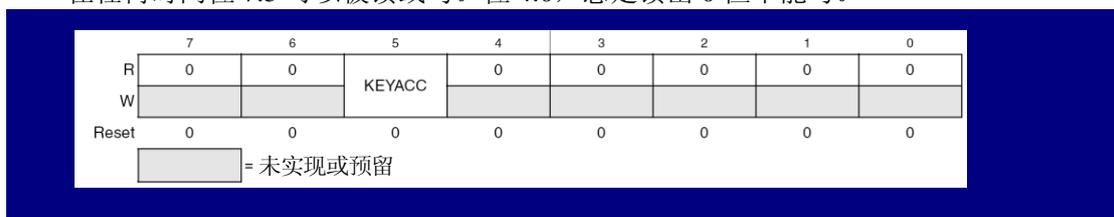


图4-8 FLASH配置寄存器(FCNFG)

表4-10 FCNFG寄存器字段描述

字段	描述
KEYACC	<p>使能写访问钥匙 — 这位使能写后门比较钥匙。关于后门钥匙机制的更多细节信息，参考 4.5, “保密性”。</p> <p>0 写 \$FFB0-\$FFB7 被解释为 FLASH 编程或擦除命令的开始。</p> <p>1 写 NVBACKKEY (\$FFB0-\$FFB7) 被解释为写比较钥匙。</p>

4.6.4 FLASH 保护寄存器 (FPROT 和 NVPROT)

在复位过程中，非易失位置 NVPROT 的内容从 FLASH 复制到 FPROT。在任何时间这个寄存器可被读。

- 如果 FPDIS = 0, 则保护可以增加 (换言之, FPS 的一个较小的值可被写入)。
- 如果 FPDIS = 1, 则写不改变保护。



图4-9 FLASH保护寄存器(FPROT)

表4-11 FPORT寄存器字段描述

字段	描述
FPS[7:1]	FLASH 保护选择位 — 当 FPDIS = 0, 这个 7 位的域确定了在 FLASH 的结束的高地址的非保护的 FLASH 位置的结束地址。保护的 FLASH 位置不能被擦除或者编程。屏蔽 FLASH 保护
FPDIS	<p>0 由 FPS[7:1] 确定的 FLASH 块为受保护的块 (不被允许的编程和擦除)。</p> <p>1 没有 FLASH 块被保护</p>

4.6.5 FLASH 状态寄存器 (FSTAT)

位 3, 1, 0 总是读出 0, 并写入操作没有意义或影响。其余 5 位是状态标志位可以在任何时间被读出。对这些位写入有特别的意义, 在位说明中被描述。

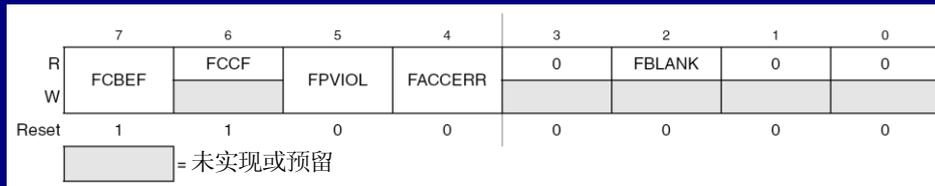


图4-10 FLASH状态寄存器(FSTAT)

表4-12 FSTAT寄存器字段描述

字段	描述
7 FCBEF	FLASH 命令缓存空标志 — FCBEF 位用来开始命令。当执行突发编程时，这表明了命令缓存为空以至后面的命令能被执行。FCBEF 通过对其写 1 或者为了编程当突发编程时把一个命令转移到阵列中来清除。仅仅突发编程命令能够缓存。 0 命令缓存满(没准备好另外命令) 1 一个新的突发编程命令能够被写到编程缓存
6 FCCF	FLASH 命令完成标志 — 当命令缓存为空和没有正在执行的命令，FCCF 自动置 1。当开始一个新的命令，FCCF 自动清零(当通过对 FCBEF 写 1 登记一个命令)。写 FCCF 没有意义和影响。 0 进行中的命令 1 所有命令结束
5 FPVIOL	保护侵犯标志 Flag — 当 FCBEF 为登记一个试图擦除或者编程一个受保护块位置的命令(错误的命令被忽略)而清零，FPVIOL 自动置 1。FPVIOL 通过对其写 1 被清除。 0 没有保护侵犯 1 试图擦除或者编程一个受保护的位置
4 FACCERR	访问错误 Flag—当没有遵守正确的命令顺序，如果在 FCDIV 寄存器初始化之前试图一个编程或者擦除操作，或者命令正在进行时 MCU 进入停止状态，FACCERR 自动设置。更多关于针对访问错误的正确的动作的细节讨论，参看 4.4.5，“访问错误”。FACCERR 通过对其写 1 被清除。对 FACCERR 写 0 没有意义或者影响。 0 没有访问错误 1 产生一个访问错误
2 FBLANK	FLASH 核对为所有为空(已擦除)标志 — 当全部 FLASH 被确认为擦除，在一个空检查完成时 FBLANK 自动置 1。FBLANK 通过清除 FCBEF 写新的有效的命令被清除。对 FBLANK 写 0 没有意义或者影响。 0 在一个空检查命令完成和 FCCF = 1，FBLANK = 0 表明 FLASH 阵列没有完成擦除 1 在一个空检查命令之后 FCCF = 1，FBLANK = 1 表明 FLASH 阵列完全擦除(都是\$FF)

4.6.6 FLASH 命令寄存器 (FCMD)

表 4-14 中说明了用户模式中仅仅能识别的五种命令代码。关于 FLASH 编程和擦除操作中的细节讨论，参考 4.4.3，“编程和擦除命令的执行”。

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
Reset	0	0	0	0	0	0	0	0

图4-11 FLASH命令寄存器(FCMD)

表4-13 FCMD寄存器字段描述

字段	描述
FCMD[7:0]	FLASH 命令位 — 见表 4-14

表4-14 FLASH命令

命令	FCMD	等同的文件标签
空检查	\$05	mBlank
字节编程	\$20	mByteProg
字节编程—突发模式	\$25	mBurstProg
页擦除(512 字节/页)	\$40	mPageErase
整体擦除(所有 FLASH)	\$41	mMassErase

所有其他命令代码为非法和产生一个访问错误。

在一个整体擦除操作之后没有必要执行一个空检查命令。只是要求空检查为保密解锁机制的一部分。

第五章 复位、中断和系统控制

5.1 引言

本章讨论了 MC9S08AW60 系列中基本的复位和中断机制以及复位和中断不同的源。一些源自外围模块的中断在本手册的其他章节中给出更加详细的讨论。为了便于查阅，本节在一个地方中集中关于所有复位和中断源的基本信息。一些复位和中断源，包括计算机运行正确(COP)看门狗和实时中断(RTI)，有它们本身的章节，不是片内外围设备系统的一部分，但是是系统控制逻辑的一部分。

5.2 特点

复位和中断功能包括：

- 为了灵活的系统控制和可靠运行的多个复位源：
 - 上电检测 (POR)
 - 启用低电压检测 (LVD)
 - 外部 RESET 引脚
 - 启用 COP 看门狗和两个超时的选择
 - 非法操作码
 - 来自后台调试主机的串行命令
- 复位状态寄存器(SRS)指出最近的复位源
- 每个模块独立的中断矢量(减少了轮询开销) (参看表 5-10)

5.3 MCU复位

对 MCU 进行复位提供了一种方法来从一组已知的初始条件开始处理.在复位期间中,大部分控制和状态寄存器强制置为初始值并从复位矢量处(\$FFFE:\$FFFF)加载程序寄存器。片内外围模块被屏蔽， I/O 管脚初始化配置成通用，高阻的输入，其下拉功能被屏蔽。为了屏蔽可屏蔽中断，条件代码寄存器(CCR) 中的 I 位置 1，以致用户程序有机会初始化堆栈指针(SP)和系统控制设置.复位中 SP 强制置为\$00FF。

MC9S08AW60 系列有如下的七个复位源：

- 上电复位 (POR)
- 低电压检测 (LVD)
- 计算机运行正确(COP)定时器
- 非法的操作码检测
- 后台调试强制复位
- 复位引脚 (RESET)
- 时钟发生器的锁丢失和时钟复位丢失

每一个这样的源，除了背景调试强制复位，在系统复位寄存器中有一个相关的位。无论何时 MCU 进入复位，内部时钟发生器(ICG)模块以选定的 $f_{\text{Self_reset}}$ 频率转向自时钟模式。复位引脚被驱动，低于 34 个总线周期，而内部总线频率是 ICG 频率的一半。34 个总线周期

完成后，引脚被释放，并由内部上拉电阻设置为上拉，除非它对外保持低电平。引脚释放后，另外 38 总线周期后引脚被采样，以确定是否复位引脚引起了 MCU 复位。

5.4 计算机正常运行（COP）看门狗

当应用软件与期望的运行不相符时，COP 看门狗试图强制系统复位。为防止系统从 COP 定时器复位，应用软件必须周期的重置 COP 计数器。如果应用程序丢失和在 COP 计数器到达中止时间之前重置 COP 计数器失败，将产生一个系统复位强制系统回到一个已知的起始点。SOPT1 中的 COPE 设置为使能 COP 看门狗(更多信息，参看 5.9.4, “系统选项寄存器(SOPT)”)。COP 计数器通过对 SRS 的地址写任何值被重置。这个写操作对只读的 SRS 没有任何影响，而是被解码为发送给 COP 计数器的一个复位信号。

任何复位之后，COP 计数器都会被激活。提供了一个可靠的方式来检查代码像预料地那样没有被执行。如果应用中没有使用 COP 看门狗，它可以通过清除只能写一次的 STOP 寄存器中是 COPE 位来被禁止。此外，COPT 位可以用来选择两个超时期之一（218 或 213 周期的总线率时钟）。即使应用将使用复位默认设置的 COPE 和 COPT，在复位初始化过程中，用户应写入只能写一次的 SOPT 来固定设置。这种方法下，如果应用程序丢失，它们不可能被偶然改变。

服务于（清除）COP 计数器的写 SRS 操作不应被放置在中断服务例程（ISR），因为即使主要的应用程序失败，ISR 也可能继续被周期地执行。

当 MCU 在激活后台模式，COP 计时器暂时停用。

5.5 中断

中断提供了一种方法来保存当前 CPU 状态和寄存器，执行一个中断服务子程序(ISR)，然后恢复 CPU 状态，以便返回执行中断之前的处理。与只是程序指令的软件中断(SWI)不同，中断由硬件事件触发，如 IRQ 管脚沿或者一个定时器溢出事件。调试模块在某种环境下也能产生 SWI。

如果在使能的中断源中发生一个事件，与之关联的只读的状态标志会被置位。CPU 当且仅当本地中断使能被置 1 来打开中断时才会响应。CCR 中的 I 位为 0 来允许中断。CCR 中的全局中断屏蔽(I 位)复位后初始化设置为 1，屏蔽（预防）了所有可屏蔽中断源。用户程序在清除 I 位以允许 CPU 相应中断之前初始化堆栈指针和完成其他系统设置。

当 CPU 收到一个许可的中断请求，它在相应中断之前要完成当前指令。中断顺序遵守与 SWI 指令相同的循环顺序，包括：

- 在堆栈中保存 CPU 寄存器
- CCR 中的 I 位置 1,屏蔽更多中断
- 在目前等待的中断中取出最高优先级中断的中断矢量
- 用从中断矢量位置中取出地址得到程序信息的前 3 个字节填满指令队列

当 CPU 响应中断，I 位自动置 1，以避免其他中断打断 ISR(这被叫做中断嵌套)。通常，当 CCR 从 ISR 入口的堆栈值恢复，I 位恢复成 0。在很少的情况下，I 位能在 ISR 中被清除(在清除能够产生中断地状态标志之后)，以便其他中断服务能够在第一个服务子程序没有完成时进入。这种方法只被推荐给具有丰富编程经验的人，因为这容易导致难于调试的隐蔽的错误。

中断服务子程序以一条中断返回(RTI)指令结束,其通过读先前存储在堆栈中的信息恢复 CCR, A, X 和 PC 寄存器为中断前的值。

注:

为了与 M68HC08 器件兼容, H 寄存器不被自动保存和恢复。好的编程经验是在中断服务子程序(ISR)开始时把 H 压到堆栈中,在用来从 ISR 中返回的 RTI 之前立即恢复 H。

当 I 位被清除,两个或更多中断等待时,最高优先级的源被首先服务(参看表 5-1)。

5.5.1 中断堆栈的帧

图 5-1 说明了一个堆栈帧的内容和组织。在中断之前,堆栈指针(SP)指向了堆栈中下一个有效的字节位置。CPU 寄存器的当前值存储在堆栈,其开始于程序计数器(PCL)的低位字节,结束于 CCR。压栈之后,SP 指向堆栈中的下一个有效位置,这是一个比保存 CCR 的地址小 1 的地址。压栈的 PC 值是主程序中在中断不发生时将执行的下一条指令所在地址。

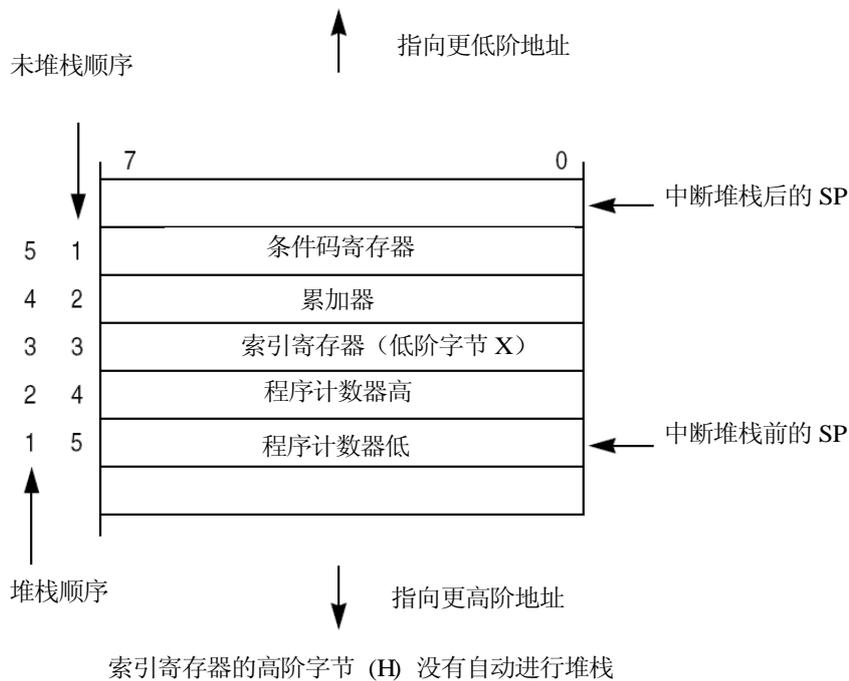


图5-1 中断堆栈帧

当执行一个 RTI 指令,这些值以相反的顺序恢复。作为 RTI 序列的一部分,CPU 用从堆栈恢复的 PC 地址得到程序信息的三个字节填充指令流水。

引起中断的状态标志在从 ISR 返回前一定要被拒绝(清除)。典型的是,在 ISR 开始的时候标志被清除,以便如果同样的源产生另一个中断,中断会被登记,直到当前 ISR 完成之后才会开始服务。

5.5.2 外部中断请求 (IRQ) 引脚

外部中断由 IRQ 状态和控制寄存器 IRQSC 管理。当使能 IRQ 功能,同步逻辑监测管脚的只边沿或者边沿和电平事件。当 MCU 处于停止状态和系统时钟关闭时,使用一个独立的异步通路以便 IRQ (如果使能)能唤醒 MCU。

5.5.2.1 引脚配置选项

为了 IRQ 管脚作为中断请求(IRQ)输入， $\overline{\text{IRQSC}}$ 中的 IRQ 管脚使能(IRQPE)控制位必须为 1。作为一个 IRQ 输入，用户可选择管脚为只边沿监测或者边沿电平检测(IRQMOD)，一个事件是否导致一个中断或者设置 IRQF 位，这能被软件轮询。

当 IRQ 引脚设定为上升边缘检测，一个可选的下拉式电阻器是可用的，而非一上拉电阻。BIH 和 BIL 指令能够在 IRQ 管脚被配置成 IRQ 输入时用来检测管脚上的电平。

注：

上拉 IRQ 引脚上测量出来的电压可低至 $V_{DD}-0.7V$ 。连接到此管脚的内部门始终保持 VDD。所有其他启用上拉电阻的引脚将有卸下测量电源。

5.5.2.2 边沿和电平敏感

IRQMOD 控制位重配置监测逻辑以便监测边沿时间和管脚电平。在这种边沿监测模式中，当一个边沿被监测到时(当 IRQ 管脚从未声明电平变为声明电平) IRQF 状态标志被置 1，但是标志当 IRQ 保持声明电平时一直被置 1(不能被清除)。

5.5.3 中断向量，源和本地掩码

表 5-1 总结了所有中断源。较高优先级的源位于表格下方。中断服务程序地址的高阶字节位于向量地址栏的第一个地址，中断服务程序地址的低阶字节位于下一个较高阶地址中。

当出现中断时，相关标记位被设置。如果相关的本地中断激活位是 1，中断请求会发送到 CPU。在 CPU 中，如果全球中断屏蔽 (CCR 中的 I 位) 是 0，CPU 将完成当前指令；堆栈 PCL、PCH、X、A 和 CCR CPU 寄存器；设置 I 位；然后为挂起的最高优先级中断获取中断向量。然后继续处理中断服务程序。

表5-1 向量摘要

向量优先级	向量号	地址 (高/低)	向量名	模块	源	使能	描述
较低	26	\$FFC0/FFC1					未使用向量空间 (用户程序有效)
	到	到					
↑	31	\$FFCA/FFCB					
	25	\$FFCC/FFCD	Vrti	系统空间	RTIF	RTIE	实时中断
	24	\$FFCE/FFCF	Viic1	IIC1	IICIF	IICIE	IIC1
	23	\$FFD0/FFD1	Vadc1	ADC1	COCO	AIEN	ADC1
	22	\$FFD2/FFD3	Vkeyboard1	KB11	KBF	KBIE	KB11 引脚
	21	\$FFD4/FFD5	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2 发送
	20	\$FFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 接收
	19	\$FFD8/FFD9	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2 错误
	18	\$FFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 发送
	17	\$FFDC/FFDD	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1 接收
	16	\$FFDE/FFDF	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 错误
	15	\$FFE0/FFE1	Vspi1	SPI1	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI1
	14	\$FFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 溢出
	13	\$FFE4/FFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 通道 1
	12	\$FFE6/FFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 通道 0
	11	\$FFE8/FFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 溢出

↓ 较高	10	\$FFEA/FFEB	Vtpm1ch5	TPM1	CH5F	CH5IE	TPM1 通道 5
	9	\$FFEC/FFED	Vtpm1ch4	TPM1	CH4F	CH4IE	TPM1 通道 4
	8	\$FFEE/FFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 通道 3
	7	\$FFF0/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 通道 2
	6	\$FFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 通道 1
	5	\$FFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 通道 0
	4	\$FFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG
	3	\$FFF8/FFF9	Vlvd	系统控制	LVDF	LVDIE	低压检测
	2	\$FFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ 引脚
	1	\$FFFC/FFFD	Vswi	内核	SWI 指令	-	软件中断
	0	\$FFFE/FFFF	Vreset	系统控制	COP LVD 引脚 非法代码	COPE LVDRE - -	看门构计时器 低压检测 外部引脚 非法代码

5.6 低电压检测（LVD）系统

MC9S08AW60 系列包括一个防止低电压的系统，以便在电源电压不稳时保护存储器内容、控制 MCU 系统状态。该系统由加电复位（POR）电路和 LVD 电路组成，其中 LVD 电路带脱扣电压用于警告和检测。当 SPMSC1 中的 LVDE 设置为 1 时，LVD 电路使能。当进入停止模式时，LVD 禁止，除非 SPMSC1 中设置了 LVDSE。如果同时设置了 LVDSE 和 LVDE，那么 MCU 不能进入 stop2（进入 stop3），LVD 激活的 stop3 模式更耗电。

5.6.1 上电复位操作

当首次接通 MCU 的电源时，或当电源电压低于加电复位准备电压 VPOR 时，POR 电路会发起复位。随着电源电压升高，LVD 电路让 MCU 保持复位状态，直到电源高于低压检测低阈值 VLVDL。POR 后，SRS 中 POR 位和 LVD 位同时被设置。

5.6.2 LVD 复位操作

通过把 LVDRE 设置为 1，可以在检测到低压情况时配置 LVD 以发起复位。低压检测阈值由 LVDV 位决定。在 LVD 复位后，LVD 系统会让 MCU 保持复位状态，直到电源电压高于低压检测阈值。LVD 复位或 POR 后都会在 SRS 寄存器中设置 LVD 位。

5.6.3 LVD 中断操作

当检测到低电压条件和 LVD 电路通过 SPMSC1 被配置成中断操作(LVDE 置位, LVDIE 置位和 LVDRE 清零), SPMSC1 中的 LVDF 将被置 1 和产生 LVD 中断请求。

5.6.4 低电压警告 (LVW)

LVD 系统有一个低压警告标记, 告知用户电源电压正接近 LVD 电压。LVW 没有相关的中断。LVW 有两个用户可选择的阈值电压, 一个高(V_{LVWH}) 和一个低(V_{LVWL})。通过 SPMSC3 中的 LVWV 选择阈值电压。设置 LVW 阈值电压等于 LVD 阈值电压是不被推荐的。典型的使用该 LVW 会选择 V_{LVWH} 和 V_{LVDL} 。

5.7 实时中断 (RTI)

实时中断功能可被用来产生周期性中断。RTI 可以接受两个时钟源, 1kHz 内部时钟或一个可用外部时钟。1kHz 内部时钟源与任何总线时钟源完全独立, 并只能被 RTI 模块和一些 MCUs, COP 看门狗使用。要使用外部时钟源, 它必须具备和激活。SRTISC 中的 RTICLKS 位用来选择 RTI 时钟源。

当 MCU 在运行, 等待或者停止 3 状态中, 两个 RTI 时钟源之一能被用到。当在停止 3 中用到外部振荡器, 必须在停止状态中使能(EREFSTEN = 1)和配置为低频操作(RANGE = 0)。只有内部 1kHz 时钟源可以被选择来 stop2 模式中唤醒 MCU。

SRTISC 寄存器包括一个只读状态标志, 一个只写请求位和用于禁止时钟源进入实时中断或选择七分之一唤醒周期的 3 位控制值 (RTIS2 : RTIS1 : RTIS0)。RTI 有一个局部中断使能, RTIE, 允许屏蔽实时中断。RTI 能通过写 RTI 的每一位为 0 被屏蔽, 这样不产生中断。关于这个寄存器的详细信息, 参看 5.9.7, “系统实时中断状态和控制寄存器(SRTISC)”。

5.8 MCLK 输出

PTA0 管脚共用于 MCLK 时钟输出。如果 MCSEL 位都是 0, MCLK 时钟禁止。若 MCSEL 的任意一个位被设置, 无论该管脚的端口数据方向控制位的状态如何, 都会导致 PTA0 管脚输出内部 MCU 总线时钟分频后时钟。分频比率由 MCSEL 位决定。管脚的斜率和驱动强度分别由 PTASE0 和 PTADS0 控制。如果斜率控制功能被打开, 最大时钟输出频率将会受到限制。如需了解不同情况下的最大频率, 请参见电气技术规范。

PTC2 引脚与 MCLK 时钟输出复用。设置引脚使能位, MPE, 导致 PTC2 引脚输出一个内部 MCU 总线时钟的分频版本。分频率由 MCSEL 位决定。当 MPE 被置位, PTC2 引脚被强制作为一个输出管脚, 不论该引脚的端口数据方向控制位的状态。如果 MCSEL 位全是 0, 该引脚打出低电平。引脚的转换率和驱动强度分别由 PTCSE2 和 PTCDS2 位控制。如果转换率控制功能被打开, 最大时钟的输出频率是有限的, 见附录 A, “电气特性和技术规范”, 了解转换率启用时引脚上升和下降时间。

5.9 复位，中断和系统控制寄存器和控制位

直接页面寄存器空间里的一个 8 位寄存器和高页寄存器空间的八个 8 位寄存器都与复位和中断系统有关。

如需了解各寄存器的绝对地址分配，请参见本产品说明第 4 章，“存储器”。本节只根据它们的名称提及寄存器和控制位。飞思卡尔提供的等式或头文件用来把这些名称转换为相应的绝对地址。

SOPT1 和 SPMSC2 寄存器中的有些控制位与运行模式有关。此处对这些位进行简要描述，更详细的描述参见第 3 章，“操作模式”。

5.9.1 中断管脚请求状态和控制寄存器（IRQSC）

直接页寄存器包括两个执行位，总是读出 0，4 个读/写位，1 个只读状态位，和 1 个只写位。这些页面配置 IRQ 功能、报告状态和确认 IRQ 事件的状态和控制位。

图 5-2 和表 5-2 分别为中断引脚请求状态和控制寄存器（IRQSC）的内容和字段描述。

	7	6	5	4	3	2	1	0
R	0	0	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
W								
Reset	0	0	0	0	0	0	0	0

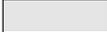
 = 未实现或预留

图5-2 中断引脚请求状态和控制寄存器（IRQSC）

表5-2 IRQSC寄存器字段描述

字段	描述
5 IRQEDG	<p>中断请求 (IRQ) 边沿选择 — 这个读 / 写控制位选择导致 IRQF 设置的 IRQ 管脚上的边沿或电平的极性。IRQMOD 控制位决定 IRQ 管脚对边沿和电平都敏感还是只对边沿敏感。当 IRQ 管脚作为 IRQ 输入被使能且配置来检测上升边沿时，它就下拉。当 IRQ 管脚作为 IRQ 输入被使能且配置来检测下降边沿时，它就上拉。</p> <p>0 IRQ 是下降边沿，或者对下降边沿 / 低电平敏感；</p> <p>1 IRQ 是上升边沿，或者对上升边沿 / 高电平敏感。</p>
4 IRQPE	<p>IRQ 管脚使能 — 这个读写控制位使能 IRQ 管脚。当设置了该位时，IRQ 管脚可以用作中断请求。</p> <p>0 IRQ 管脚禁止；</p> <p>1 IRQ 管脚使能。</p>
3 IRQF	<p>RQ 标志 — 该只读状态位显示中断请求事件发生的时间。</p> <p>0 无 IRQ 请求；</p> <p>1 检测到 IRQ 事件。</p>
2 IRQACK	<p>RQ 确认 — 这个只写位用来确认中断请求事件（写 1 来清除 IRQF）。写入 0 则没有任何意义或影响。读总是返回 0。如果选择了边沿和电平检测（IRQMOD = 1），则不能清除 IRQF，并且 IRQ 管脚位于断言级。</p>
1 IRQIE	<p>IRQ 中断使能 — 这个读 / 写控制位决定 IRQ 事件是否生成中断请求。</p> <p>0 当 IRQF = 1 时不产生中断请求（使用轮询）；</p> <p>1 当 IRQF = 1 时产生中断请求。</p>
0 IRQMOD	<p>IRQ 检测模式 — 这个读 / 写控制位选择只边沿检测还是边沿和电平检测。IRQEDG 控制位决定检测为中断请求事件的边沿和电平极性。如需了解更多的信息，请参见 5.5.2.2”边沿和电平敏感度”。</p> <p>0 只下降边沿或上升边沿的 IRQ 事件；</p> <p>1 下降边沿和低电平 IRQ 事件或上升边沿和高电平 IRQ 事件。</p>

5.9.2 系统复位状态寄存器（SR）

该寄存器包括 7 个只读状态标记，以显示最近复位源。将 1 写入 SBDFR 寄存器的 BDFR 时，调试主机强制完成复位，SRS 中不设置任何状态位。COP 使能时，在寄存器地址中写入任意值会清除 COP 看门狗定时器，但不会对寄存器内容造成影响。这些位的复位状态取决于导致 MCU 复位的原因。

图 5-3 和表 5-3 分别为系统复位状态寄存器（SR）的内容和字段描述。

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	0	ICG	LVD	0
W	写任何值到 SIMRS 地址清除 COP 看门狗计时器							
POR	1	0	0	0	0	0	1	0
LVR:	U	0	0	0	0	0	1	0
其它复位	0	(1)	(1)	(1)	0	(1)	0	0

U = 未受复位影响

1 任何这些复位源在复位期间被激活，将引起相应的位被设置；在复位期间没有被激活的复位源的相应位将被清 0。

图 5-3 系统复位状态寄存器 (SRS)

表5-3 SRS寄存器字段描述

字段	描述
7	加电复位 — 复位由电源开检测逻辑造成。由于此时内部电源电压在上升，所以还要设置低压复位 (LVD) 状态位以显示已进行的复位，这时内部电源低于 LVD 阈值。
POR	0 非 POR 造成的复位。 1 POR 造成的复位。
6	外部复位管脚 — 复位由外部复位管脚上的活动低电平造成。
PIN	0 非外部复位管脚造成的复位。 1 外部复位管脚造成的复位。
5	计算机正常操作 (COP) 看门狗 — 复位由 COP 看门狗定时器超时导致。该复位源可以由 COPE = 0 拦截。
COP	0 非 COP 超时造成的复位。 1 超时造成的复位。
4	非法操作码 — 复位由试图实施未实现或非法操作码导致。如果停止由 SOPT 寄存器里 STOPE = 0 禁止，STOP 指令被视为非法。如果主动后台模式由 BDCSC 寄存器里 ENBDM = 0 禁止，BGND 指令指令被视为非法。
ILOP	0 不是由非法操作码造成的复位。 1 非法操作码造成的复位。
2	内部时钟发生模块复位 — 由 ICG4 模块复位造成的复位。
ICG	0 不是由 ICG4 模块复位造成的复位。 1 由 ICG4 模块复位造成的复位
1	低压检测 — 如果设置了 LVDRE 位且电源降到 LVD 脱扣电压以下，就会发生 LVD 复位。该位也可以由 POR 设置。
LVD	0 不是由 LVD 门槛或 POR 造成的复位。 1 由 LVD 门槛或 POR 造成的复位。

5.9.3 系统后台调试强制复位寄存器 (SBD FR)

这个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 SBD FR。从用户程序写入寄存器的尝试被忽略。读总是返回 \$00。

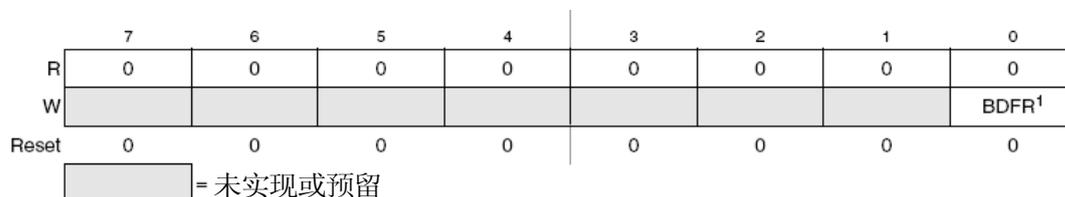


图5-4 系统后台调试强制复位寄存器 (SBD FR)

1 BDFR 是可写入的，只有通过串行后台调试命令，而不是从用户程序。

表5-4 SBD FR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 WRITE_BYTE ，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。

5.9.4 系统选项寄存器 (SOPT)

该寄存器可以在任何时候读取。位 3 和位 2 没有被落实并总是读出 0。这是一个只读寄存器，因此只有复位以后的第一次写入才被认可。任何后续 SOPT 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。SOPT 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。

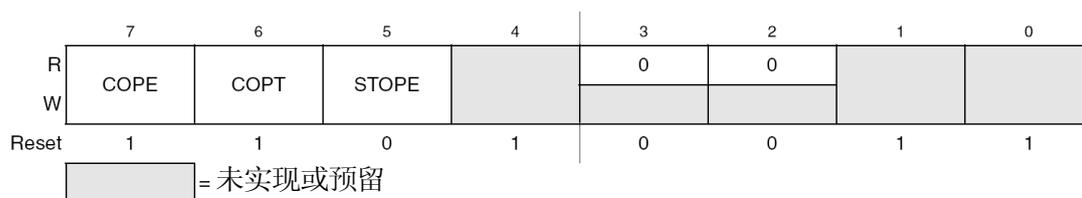


图5-5 系统选项寄存器 (SOPT)

表5-5 SOPT寄存器字段描述

字段	描述
7 COPE	COP 看门狗使能 — 复位后，只能写一次的位默认为 1 0 COP 看门狗计时器禁止 1 COP 看门狗计时器使能（超时强制复位）
6 COPT	COP 看门狗超时 — 复位后，只能写一次的位默认为 1 0 选择短超时期（ 2^{13} 个 BUSCLK 周期） 1 选择长超时期（ 2^{18} 个 BUSCLK 周期）
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。

5.9.5 系统 MCLK 控制寄存器（SMCLK）

这个寄存器册被用来控制 MCLK 时钟输出。

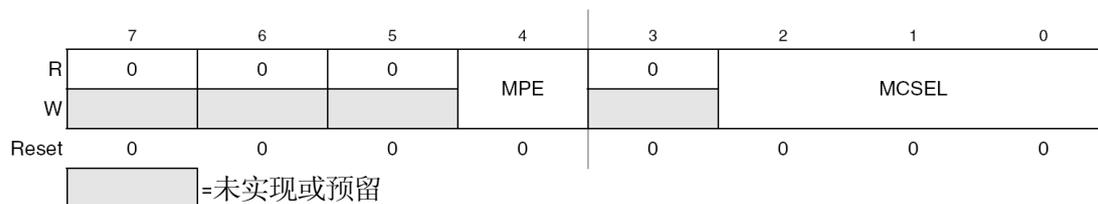


图5-6 系统MCLK控制寄存器（SMCLK）

表5-6 SMCLK寄存器字段描述

字段	描述
4 MPE	MCLK 引脚使能 — 该位被用来使能 MCLK 功能。 0 MCLK 输出禁止 1 PTC2 引脚上 MCLK 输出使能
2:0 MCSEL	MCLK 分频选择 — 这些位被用来选择 MCLK 输出的分频率，根据下面的公式，当 MCSEL 位不为全 0 时。MCSEL 位为全 0 的情况和 MPE 被置位，该引脚为低电平有效。见方程 5-1。

$$\text{MCLK 频率} = \text{总线时钟频率} \div (2 * \text{MCSEL})$$

Eqn. 5-1

5.9.6 系统器件识别寄存器（SDIDH，SDIDL）

这些只读寄存器也包括在内，这样主机开发系统就能够识别 HCS08 衍生和修定编号。这使得开发软件能够识别特定的存储器块、寄存器和控制位在目标 MCU 的什么位置。

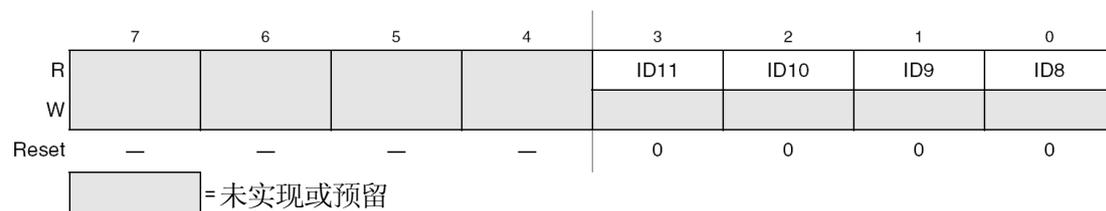


图5-7 系统器件识别寄存器—高（SDIDH）

表5-7 SDIDH寄存器字段描述

字段	描述
7:4	位 7:4 被预留。读这些位将会导致一个不确定值。写入操作没有意义。
预留的	
3:0	部件识别编号 — 每一个 HCS08 的派生物有一个独特的识别号。MC9S08AW60 固定的编号为值 \$008。也参考表 5-8 中的 ID 号。
ID[11:8]	

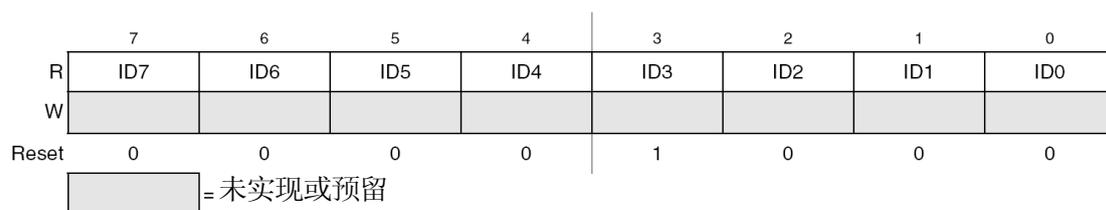


图5-8 系统器件识别寄存器—低（SDIDL）

表5-8 SDIDL寄存器字段描述

字段	描述
7:0 ID[7:0]	部件识别编号 — 每一个 HCS08 的派生物有一个独特的识别号。MC9S08AW60 固定的编号为值 \$008。也参考表 5-7 中的 ID 号。

5.9.7 系统实时中断状态和控制寄存器 (SRTISC)

该寄存器包含一个只读状态标志， 1 个只写应答位， 3 个读/写延迟选择， 3 个总读出 0 的未实现位。

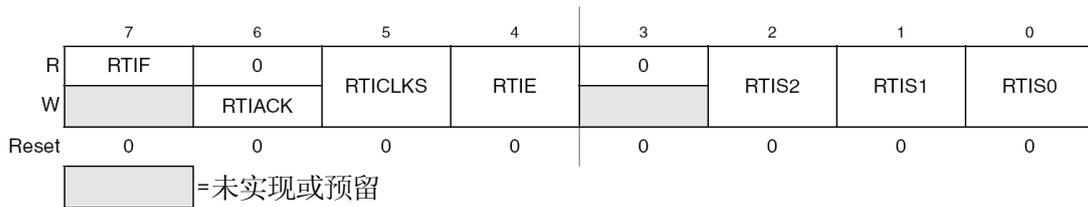


图5-9 系统RTI状态和控制寄存器 (SRTISC)

表5-9 SRTISC寄存器字段描述

字段	描述
7 RTIF	实时中断标志 — 这个只读状态位指示周期唤醒定时器时间溢出。 0 周期唤醒定时器还未时间溢出 1 周期唤醒定时器已经时间溢出
6 RTIACK	实时中断应答 — 这个只写位用来应答实时中断请求(写 1 清除 RTIF)。写 0 没有意义和影响。读总是返回 0。
5 RTICLKS	实时中断时钟选择 — 这个读/写位选择实时中断的时钟源 0 实时中断请求时钟源为内部 1kHz 振荡器 1 实时中断请求时钟源为外部时钟
4 RTIE	实时中断使能 — 这个读/写位使能实时中断。 0 屏蔽实时中断 1 使能实时中断
2:0 RTIS[2:0]	选择实时中断延迟 — 这些读/写位选择 RTI 的唤醒时延。实时中断的时钟源是一个自适应时钟的频率振荡源，约 1kHz，它独立于其它 MCU 时钟源。使用外部时钟源的延迟，将晶体的频率划分为值 RTIS2 : RTIS1 : RTIS0 。参看表 5-10。

表5-10 实时中断频率

RTIS2:RTIS1:RTIS0	1kHz 时钟源延迟 ¹	使用外部时钟源延迟 (晶体频率)
0:0:0	禁止周期唤醒时钟	禁止周期唤醒时钟
0:0:1	8 ms	256 分频
0:1:0	32 ms	1024 分频
0:1:1	64 ms	2048 分频
1:0:0	128 ms	4096 分频
1:0:1	256 ms	8192 分频
1:1:0	512 ms	16384 分频
1:1:1	1.024 s	32768 分频

1. 这个栏给出的正常值是建立在 $f_{RTI} = 1\text{kHz}$ 的基础上的。见附录 A，“电气特性和时间规格”， f_{RTI} 能容纳的值。

5.9.8 系统电源管理状态和控制 1 寄存器 (SPMSC1)

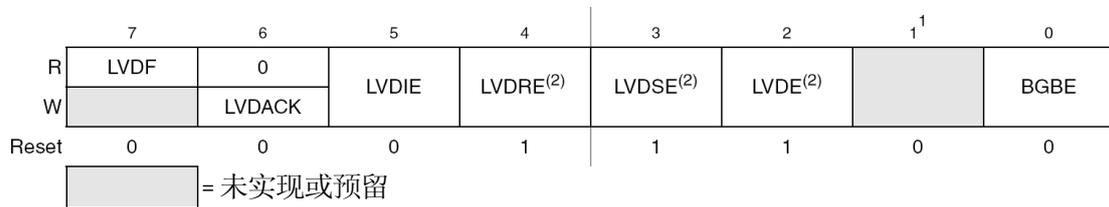


图5-10 系统电源管理状态和控制1寄存器 (SPMSC1)

1. 第 1 位保留，必须写为 0。
2. 这位复位后只能写一次。另外的写被忽略。

表5-11 SPMSC1寄存器字段描述

字段	描述
7 LVDF	低电压监测标志 — 如果 LVDE = 1,这个只读状态位指示一个低电压监测事件。
6 LVDACK	低电压监测应答 — 这个只写位用来应答低电压监测错误(写 1 清除 LVDF).读总是返回 0。
5 LVDIE	低电压监测中断使能 — 这位用来使能 LVDE 的硬件中断请求。 0 屏蔽硬件中断(使用轮询). 1 当 LVDF = 1,请求硬件中断.
4 LVDRE	低电压监测复位使能 — 这个只写位使能 LVDF 事件产生一个硬件复位(假设 LVDE = 1)。 0 LVDF 部产生硬件复位. 1 当 LVDF = 1 强制 MCU 复位.
3 LVDSE	低电压监测停止使能—如果 LVDE = 1, 当 MCU 在停止状态时, 这个读/写位确定是否运行低电压监测功能。 0 在停止状态屏蔽低电压监测. 1 在停止状态使能低电压监测.
2 LVDE	低电压监测使能 — 这个只写位使能低电压监测逻辑和准许寄存器其它位的操作。 0 屏蔽 LVD 逻辑. 1 使能 LVD 逻辑.
0 BGBE	能带隙缓存使能 —这位使能一个能带隙参考电压的内部缓存, 参考电压用于 ADC 其中一个内部通道或者 ACMP 模块。 0 屏蔽能带隙缓存. 1 使能能带隙缓存.

5.9.9 系统电源管理的地位和控制 2 寄存器 (SPMSC2)

这寄存器被用来报告低压警告功能的状态, 并配置 MCU 的停止模式行为。

	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV	LVWV	PPDF	0		PPDC ¹
W		LVWACK				PPDACK		
上电复位	0 ⁽²⁾	0	0	0	0	0	0	0
LVD 复位	0 ⁽²⁾	0	U	U	0	0	0	0
其它复位	0 ⁽²⁾	0	U	U	0	0	0	0

= 未实现或预留
 U = 未受复位影响

图5-11 系统电源管理状态和控制2寄存器 (SPMSC2)

1. 这位复位后只能写一次。另外的写被忽略。
2. 在 V_{supply} 过渡低于此行点或复位后并 V_{supply} 已经低于 V_{LVW} 这种情况下，LVWF 将被置位。

表5-12 SPMSC2 寄存器字段描述

字段	描述
7	低电压警告标志 — LVWF 位指示低压警告状态。
LVWF	0 低压警告未发生 1 低压警告发生或已发生
6	低电压警告应答 — LVWACK 位是低压警告电压。如果低压警告没有发生，写 1 到 LVWACK 清 LVWF 为 0。
LVWACK	
5	低压监测电压选择 — LVDV 位选择 LVD 行点电压 (V_{LVD})。
LVDV	0 低行点选择 ($V_{LVD}=V_{LVDL}$) 1 高行点选择 ($V_{LVD}=V_{LVDH}$)
4	低电压警告电压选择 — LVWV 位。
LVWV	0 低行点选择 ($V_{LVW}=V_{LVWL}$) 1 高行点选择 ($V_{LVW}=V_{LVWH}$) .
3	部分掉电标志 — 这个只读状态位指示 MCU 从停止 2 状态恢复。
PPDF	0 没有从停止 2 状态恢复。 1 从停止 2 状态恢复。
2	部分掉电应答 — 给 PPDACK 写 1 清除 PPDF 和 PDF 位..
PPDACK	
0	部分掉电控制 — 只可写一次的 PPDC 位控制选择 stop2 或 stop3 模式。
PPDC	0 stop3 模式使能 1 stop2 部分掉电模式使能。

第六章 并行输入/输出控制

6.1 介绍

本小节解释了与并行输入 / 输出(I/O)相关的软件控制。MC9S08AW60 系列有 7 个 I/O 端口，总共包含 54 个通用 I/O 引脚。如需了解这些引脚的逻辑和硬件分配的更多信息，请参见第 2 章，“引脚和连接”。

很多引脚都与片内外围设备复用，如定时器系统，通信系统或键盘中断。当这些其他的模块不控制端口引脚时，它们转向通用 I/O 控制。

在应用中没有被使用的引脚一定要被挂起。防止由浮动输入引起的额外电流并在噪声或瞬间事件中提高抗干扰力。终止方法包括：

- 配置未使用的引脚作为输出高电平或低电平
- 配置未使用的引脚作为输入并使用内部或外部上拉电阻

永远不要将未使用的引脚连接到 V_{DD} 或 V_{SS} 。表 6-1 为 KBI 和并行 I/O 的交互作用。

表 6-1 KBI 和并行 I/O 的交互作用

PTxPEn (上拉使能)	PTxDDn (数据方向)	KBIPEn (KBI 引脚使能)	KBEDGn (KBI 边沿选择)	上拉	下拉
0	0	0	X ¹	禁止	禁止
1	0	0	X	开启	禁止
X	1	0	X	禁止	禁止
1	X	1	0	开启	禁止
1	X	1	1	禁止	开启
0	X	1	X	禁止	禁止

6.2 特点

并行 I/O 和引脚控制功能，根据产品包装上的选择，包括：

- 在 7 个端口上共有 54 个通用 I/O 引脚
- 滞后输入缓冲器
- 每个输入引脚上软件控制上拉电阻
- 软件控制转换率输出缓冲器
- 8 个端口 A 引脚
- 8 个端口 B 引脚与 ADC1 复用
- 7 个端口 C 引脚共享与 SCI2， IIC1， MCLK

- 8 个端口 D 引脚与 ADC1 ， KBI1 ， TPM1 和 TPM2 外部时钟输入复用
- 8 个端口 E 引脚与 SCI1 ， TPM1 ， SPI1 复用
- 8 个端口 F 引脚与 TPM1 和 TPM2 复用
- 7 个端口 G 引脚与 XTAL ， EXTAL 和 KBI1 复用

6.3 引脚说明

该 MC9S08AW60 系列在 7 个端口（PTA—PTG）上共有 54 个并行 I/O 引脚。并不是所有的引脚在所有封装中被导出。查询可用并行 I/O 引脚的引脚分配在第 2 章，“引脚和连接”。所有这些引脚可用于通用 I/O，当它们没有被其它片内外围系统使用。

经过复位后，复用的外围模块功能禁止，I/O 配置为输入（PTxDDn = 0）。引脚控制功能，对每个引脚都是配置如下：转换速率禁止（PTxSEn = 0），低驱动强度选定（PTxDSn=0），以及内部上拉禁用（PTxPEN = 0）。

以下段落讨论每个端口和软件控制，确定每个引脚的使用。

6.3.1 端口 A

图 6-1 端口 A 引脚名：

端口 A	位 7	6	5	4	3	2	1	位 0
MCU 引脚	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0

图6-1 端口A引脚名

端口 A 引脚是通用 I/O 引脚。并行 I/O 功能是被端口 A 数据（PTAD）和数据方向（PTADD）寄存器控制，设在第 0 页寄存器空间。引脚控制寄存器，上拉使能（PTAPE），转换率控制（PTASE），和驱动强度选择（PTADS）是设在高页寄存器中。关于更多通用 I/O 控制的信息参见 6.4，“并行 I/O 控制”，关于更多引脚控制的信息参见 6.5，“引脚控制”。

6.3.2 端口 B

图 6-2 端口引脚 B 引脚名：

端口 B	位 7	6	5	4	3	2	1	位 0
MCU 引脚	PTB7/ AD1P7	PTB6/ AD1P6	PTB5/ AD1P5	PTB4/ AD1P4	PTB3/ AD1P3	PTB2/ AD1P2	PTB1/ AD1P1	PTB0/ AD1P0

图6-2 端口引脚B引脚名

端口 B 引脚是通用 I/O 引脚。并行 I/O 功能是被端口 B 数据（PTBD）和数据方向（PTBDD）寄存器控制，设在第 0 页寄存器空间。引脚控制寄存器，上拉使能（PTBPE），转换率控制（PTBSE），和驱动强度选择（PTBDS）是设在高页寄存器中。关于更多通用 I/O 控制的信息参见 6.4，“并行 I/O 控制”，关于更多引脚控制的信息参见 6.5，“引脚控制”。

端口 B 通用 I/O 与 ADC 复用。任何作为 ADC 输入的引脚将会禁止通用 I/O 功能。更多关于使用端口 B 作为模拟输入的信息，请参见第 14 章，“类比数位转换器（S08ADC10V1）”。

6.3.3 端口 C

图 6-3 端口 C 引脚名：

端口 C	位 7	6	5	3	3	2	1	位 0
MCU 引脚		PTC6	PTC5/ RxD2	PTC4	PTC3/ TxD2	PTC2/ MCLK	PTC1/ SDA1	PTC0/ SCL1

图6-3 端口C引脚名

端口 C 引脚是通用 I/O 引脚。并行 I/O 功能是被端口 C 数据（PTCD）和数据方向（PTCDD）寄存器控制，设在第 0 页寄存器空间。引脚控制寄存器，上拉使能（PTCPE），转换率控制（PTCSE），和驱动强度选择（PTCDS）是设在高页寄存器中。关于更多通用 I/O 控制的信息参见 6.4，“并行 I/O 控制”，关于更多引脚控制的信息参见 6.5，“引脚控制”。

端口 C 通用 I/O 与 SCI2、IIC 和 MCLK 复用。当任何这些复用功能使能，方向、输入或输出由复用功能控制而不是由并行 I/O 端口的数据方向寄存器控制。另外，被复用功能配置成输出的引脚，由复用功能控制输出数据而不是由端口数据寄存器控制。

更多关于使用端口 C 引脚作为 SCI 引脚的信息，请参见第 11 章，“串行通信接口（S08SCIV2）”。

更多关于使用端口 C 引脚作为 IIC 引脚的信息，请参见第 13 章“跨集成电路（S08IICV1）”。

更多关于使用端口 PTC2 作为 MCLK 引脚的信息，请参见第 5 章，“复位，中断，和系统配置”。

6.3.4 端口 D

图 6-4 端口 D 引脚名：

端口 D	位 7	6	5	4	3	2	1	位 0
MCU 引脚	PTD7/ AD1P15/ KBI1P7	PTD6/ AD1P14/ TPM1CLK	PTD5/ AD1P13/	PTD4/ AD1P12/ TPM2CLK	PTD3/ AD1P11/ KBI1P6	PTD2/ AD1P10/ KBI1P5	PTD1/ AD1P9	PTD0/ AD1P8

图6-4 端口D引脚名

端口 D 引脚是通用 I/O 引脚。并行 I/O 功能是被端口 D 数据（PTDD）和数据方向（PTDDD）寄存器控制，设在第 0 页寄存器空间。引脚控制寄存器，上拉使能（PTDPE），

转换率控制（PTDSE），和驱动强度选择（PTDDS）是设在高页寄存器中。关于更多通用 I/O 控制的信息参见 6.4，“并行 I/O 控制”，关于更多引脚控制的信息参见 6.5，“引脚控制”。

端口 D 通用 I/O 与 ADC、KBI 和 TPM1 和 TPM2 外部时钟输入复用。当任何这些复用功能使能，方向、输入或输出由复用功能控制而不是由并行 I/O 端口的数据方向寄存器控制。当一个引脚与 ADC 和一个数字外设功能复用，ADC 有更高的优先级。例如，ADC 和 KBI 被配置成使用 PTD7，则引脚被 ADC 模块控制。

更多关于使用端口 D 引脚作为 TPM 外部时钟输入的信息，请参见第 10 章“定时器/脉宽调制（S08TPMV2）”。

更多关于使用端口 D 引脚作为模拟输入的信息，请参见第 14 章“，类比数位转换器（S08ADC10V1）”。

更多关于使用端口 D 引脚作为键盘输入的信息，请参见第 9 章，“键盘中断（S08KBIV1）”。

6.3.5 端口 E

图 6-5 端口 E 引脚名：

端口 E	位 7	6	5	4	3	2	1	位 0
MCU 引脚	PTE7/ SPSCK1	PTE6/ MOSI1	PTE5/ MISO1	PTE4/ SS1	PTE3/ TPM1CH1	PTE2/ TPM1CH0	PTE1/ RxD1	PTE0/ TxD1

图6-5 端口E引脚名

端口 E 引脚是通用 I/O 引脚。并行 I/O 功能是被端口 E 数据（PTED）和数据方向（PTEDD）寄存器控制，设在第 0 页寄存器空间。引脚控制寄存器，上拉使能（PTEPE），转换率控制（PTESE），和驱动强度选择（PTEDS）是设在高页寄存器中。关于更多通用 I/O 控制的信息参见 6.4，“并行 I/O 控制”，关于更多引脚控制的信息参见 6.5，“引脚控制”。

端口 E 通用 I/O 与 SCI1、SPI 和 TPM1 时钟通道复用。当任何这些复用功能使能，方向、输入或输出由复用功能控制而不是由并行 I/O 端口的数据方向寄存器控制。另外，被复用功能配置成输出的引脚，由复用功能控制输出数据而不是由端口数据寄存器控制。

更多关于使用端口 E 引脚作为 TPM 外部时钟输入的信息，请参见第 11 章，“串行通信接口（S08SCIV2）”。

更多关于使用端口 E 引脚作为 SPI 引脚的信息，请参见第 12 章，“串行外围接口（S08SPIV3）”。

更多关于使用端口 E 引脚作为 TPM 通道引脚的信息，请参见第 10 章“定时器/脉宽调制（S08TPMV2）”。

6.3.6 端口 F

图 6-6 端口 F 引脚名：

端口 F	位 7	6	5	4	3	2	1	位 0
MCU 引脚	PTF7	PTF6	PTF5/ TPM2CH1	PTF4/ TPM2CH0	PTF3/ TPM1CH5	PTF2/ TPM1CH4	PTF1/ TPM1CH3	PTF0/ TPM1CH2

图6-6 端口F引脚名

端口 F 引脚是通用 I / O 引脚。并行 I / O 功能是被端口 F 数据 (PTFD) 和数据方向 (PTFDD) 寄存器控制, 设在第 0 页寄存器空间。引脚控制寄存器, 上拉使能 (PTFPE) , 转换率控制 (PTFSE) , 和驱动强度选择 (PTFDS) 是设在高页寄存器中。关于更多通用 I / O 控制的信息参见 6.4 , “并行 I / O 控制”, 关于更多引脚控制的信息参见 6.5 , “引脚控制”。

端口 F 通用 I / O 与 TPM1 和 TPM2 时钟通道复用。当任何这些复用功能使能, 方向、输入或输出由复用功能控制而不是由并行 I/O 端口的数据方向寄存器控制。另外, 被复用功能配置成输出的引脚, 由复用功能控制输出数据而不是由端口数据寄存器控制。

更多关于使用端口 F 引脚作为 TPM 通道引脚的信息, 请参见第 10 章“定时器/脉宽调制 (S08TPMV2) ”。

6.3.7 端口 G

图 6-7 端口 G 引脚名:

端口 G	位 7	6	5	4	3	2	1	位 0
MCU 引脚		PTG6/ EXTAL	PTG5/ XTAL	PTG4/ KBI1P4	PTG3/ KBI1P3	PTG2/ KBI1P2	PTG1/ KBI1P1	PTG0/ KBI1P0

图6-7 端口G引脚名

端口 G 引脚是通用 I / O 引脚。并行 I / O 功能是被端口 G 数据 (PTGD) 和数据方向 (PTGDD) 寄存器控制, 设在第 0 页寄存器空间。引脚控制寄存器, 上拉使能 (PTGPE) , 转换率控制 (PTGSE) , 和驱动强度选择 (PTGDS) 是设在高页寄存器中。关于更多通用 I / O 控制的信息参见 6.4 , “并行 I / O 控制”, 关于更多引脚控制的信息参见 6.5 , “引脚控制”。

端口 G 通用 I / O 与 KBI、XTAL 和 EXTAL 复用。当一个引脚使能作为 KBI 输入, 这个引脚功能就是输入, 无论相关的 PTG 数据方向寄存器位的状态。当外部振荡器使能, PTG5 和 PTG6 功能作为振荡器引脚。在这种情况下, 相关的并行 I / O 和引脚控制寄存器没有控制引脚。

更多关于使用端口 G 引脚作为 XTAL 和 EXTAL 引脚的信息, 请参见第 8 章, “内部时钟发生器 (S08ICGV4) ”。

更多关于使用端口 G 引脚作为键盘输入的信息, 请参见第 9 章, “键盘中断 (S08KBIV1) ”。

6.4 并行I/O控制

通过端口数据寄存器完成并行 I/O 的读和写。输入或输出的方向由端口数据方向寄存器控制。图 6-8 说明了一个单独引脚的并行 I/O 功能。

数据方向控制位 (PTxDDn) 决定是否启动引脚相关的输出缓冲器, 同时控制端口数据寄存器读取的源。每个端口引脚有一个数据方向寄存器位。当 PTxDDn=0, 相应的引脚作为输入, 并读取 PTxD, 返回该引脚的值。当 PTxDDn=1, 相应的引脚作为输出, 并读取 PTxD, 返回最后写入端口数据寄存器的值。当一个外设模块或系统功能控制端口引脚, 数据方向寄存器位仍然控制那些读取端口数据寄存器返回的值, 即使外设系统首要控制实际引脚的方向。

当引脚使能复用模拟功能时, 所有的数字位功能被禁止。任意端口数据寄存器位读值都为 0, 就有复用功能使能。总体来说, 每当数字功能和模拟功能共用一个引脚时, 模拟功能都优先。因此数字和模拟功能同时使能时, 引脚由模拟功能控制。

一个好的编程习惯是在把端口引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引脚。

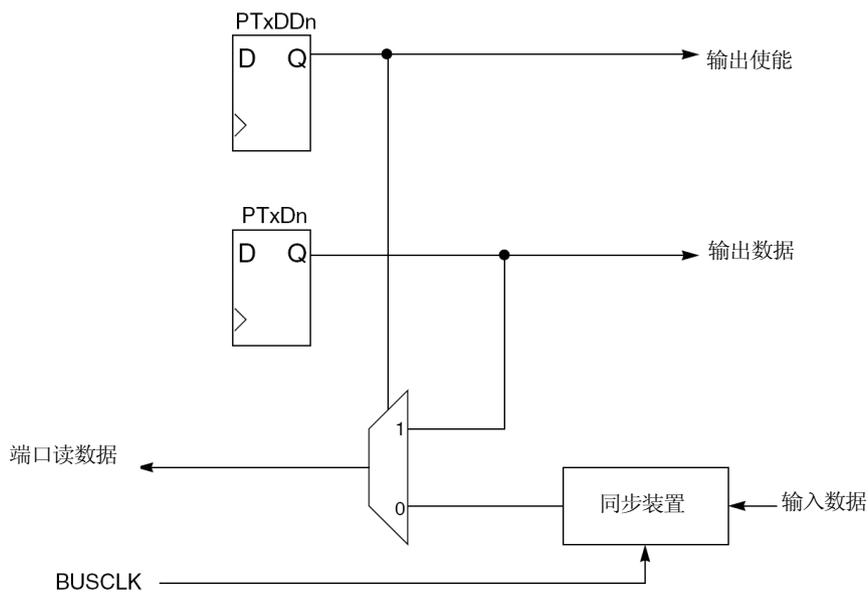


图6-8 并行I/O框图

6.5 引脚控制

引脚控制寄存器设在存储器的高页寄存器段块。这些寄存器被用来为 I/O 引脚控制上拉, 转换率, 和驱动强度。引脚控制寄存器操作独立于并行 I/O 寄存器。

6.5.1 内部上拉使能

每个端口引脚的内部上拉装置可通过设置某个上拉允许寄存器（PTXPEN）的相应位来使能。如果并行 I/O 控制逻辑或任何外围设备功能将引脚配置为输出，上拉器件就会被禁止，这与对应的上拉允许寄存器位的状态无关。如果引脚由模拟功能控制，上拉器件同样会被禁止。

6.5.2 输出转换率控制使能

每个端口引脚的转换率可通过设置某个转换率控制寄存器（PTXSEN）的相应位来使能。启动时，转换控制限制输出的转换速度，减少 EMC 发射。转换率控制对配置为输入的引脚没有任何影响。

6.5.3 输出驱动强度选择

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引脚。选定好高驱动后，引脚就能够送出和吸收更大的电流。即使可以将每个 I/O 引脚选择为高驱动，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 引脚的 DC 行为，然而，AC 行为也会受到影响。高驱动允许引脚以与低驱动使能引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引脚作为高驱动而受到影响。

6.6 停止模式下的引脚行为

在停止模式基础上，I/O 功能与执行一条 STOP 指令后的结果不同。下面解释不同停止状态的管脚行为：

- 停止 2 模式是局部断电模式，其中 I/O 寄存器保持执行 STOP 指令前的状态。执行 STOP 指令使 MCU 进入 STOP2 模式前，应将 CPU 寄存器状态和 I/O 寄存器状态保存在 RAM 中。在从 STOP2 模式恢复时，用户访问任何 I/O 前，都应检查 SPMSC2 寄存器里的 PPDF 位状态。如果 PPDF 位是 0，I/O 必须按发生过上电复位那样进行初始化。如果 PPDF 位是 1，外围设备可能要求通过初始化恢复为它们的停止前状态。如果执行 STOP 指令前数据已保留在 RAM 中，就可以完成上述操作。然后用户必须向 SPMSC2 寄存器中的 PPDACK 位写入 1。现在，用户应用程序再次被允许访问 I/O。
- 停止 3 模式保留了所有 I/O，因为内部逻辑电路处于通电状态。恢复时，用户可以使用正常的 I/O 功能。

6.7 并行 I/O 和引脚控制寄存器

本节提供了与并行 I/O 端口和引脚控制功能相关的寄存器的信息。这些并行 I/O 寄存器位于存储器映射的 0 页面中，引脚控制寄存器位于存储器的高页寄存器部分。

如需了解所有并行 I/O 第 4 章，“存储器”中的表格。本节只按照寄存器和控制位的名称查看寄存器和控制位。飞思卡尔提供的等式或头文件一般用来把这些名称转换到适当的绝对地址。

6.7.1 端口 A I/O 寄存器 (PTAD 和 PTADD)

端口 A 并行 I/O 功能由下面列出的寄存器控制。

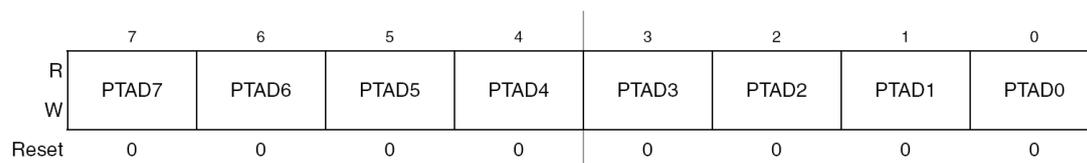


图6-9 端口A数据寄存器 (PTAD)

表 6-2 PTAD 寄存器字段描述

字段	描述
7:0 PTAD[7:0]	<p>端口 A 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。</p> <p>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。</p> <p>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。</p>

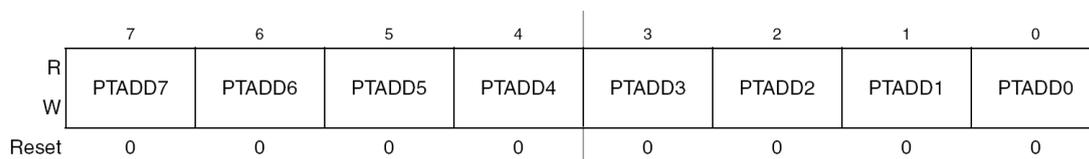


图 6-10 端口A数据方向寄存器 (PTADD)

表 6-3 PTADD 寄存器字段描述

字段	描述
7:0 PTADD[7:0]	<p>端口 A 数据方向位 — 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。</p> <p>0 输入 (输出驱动被禁止)，读数返回管脚值。</p> <p>1 A 端口位输出驱动使能，PTAD 读数返回 PTADn 内容。</p>

6.7.2 端口 A 引脚控制寄存器 (PTAPE , PTASE , PTADS)

I/O 控制以外，端口 A 引脚还由以下列出的寄存器控制。

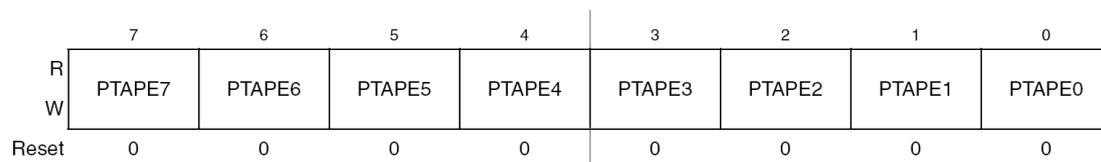


图6—11 端口A内部上拉允许

表 6—4 PTAPE 寄存器字段描述

字段	描述
7:0 PTAPE[7:0]	<p>端口 A 内部上拉使能位 — 对于 PTA 管脚，这些控制位决定着为相关 PTA 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 A 端口管脚，这些位不会产生影响，同时内部上拉器件被禁止。</p> <p>0 A 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 A 端口位内部上拉 / 下拉器件使能。</p>

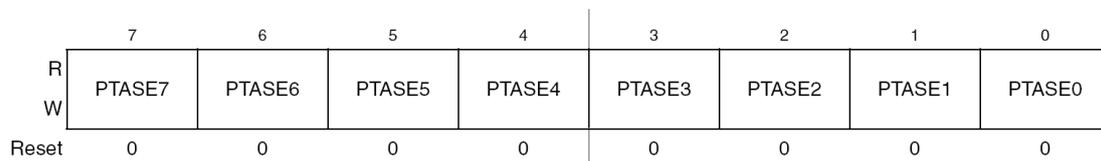


图6—12 端口A输出转换率控制允许 (PTASE)

表 6—5 PTASE 寄存器字段描述

字段	描述
7:0 PTASE[7:0]	<p>端口 A 输出转换率使能位 — 这些控制位决定着是否为相关 PTA 管脚使能输出转换率控制。对于配置为输入的 A 端口管脚，这些位不会产生任何影响。</p> <p>0 A 端口位输出转换率控制被禁止。</p> <p>1 A 端口位输出转换率控制使能。</p>

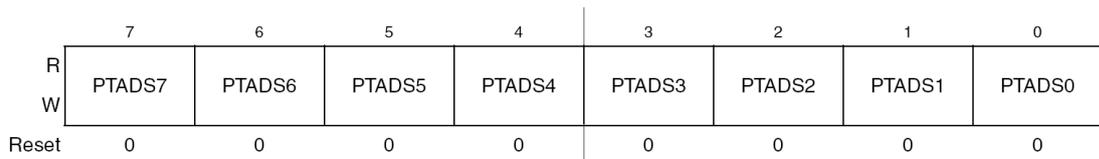


图6-13 端口A输出驱动强度选择 (PTADS)

表 6-6 PTADS 寄存器字段描述

字段	描述
7:0 PTADS[7:0]	<p>端口 A 输出驱动强度选择位 — 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。</p> <p>0 A 端口位选择的低输出驱动强度。</p> <p>1 A 端口位选择的高输出驱动强度。</p>

6.7.3 端口 B I/O 寄存器 (PTBD 和 PTBDD)

端口 B 并行 I/O 功能由下面列出的寄存器控制。

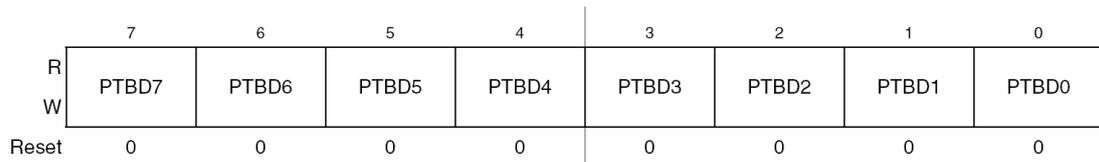


图6-14 端口B数据寄存器 (PTBD)

表 6-7 PTBD 寄存器字段描述

字段	描述
7:0 PTBD[7:0]	<p>端口 B 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。</p> <p>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。</p>

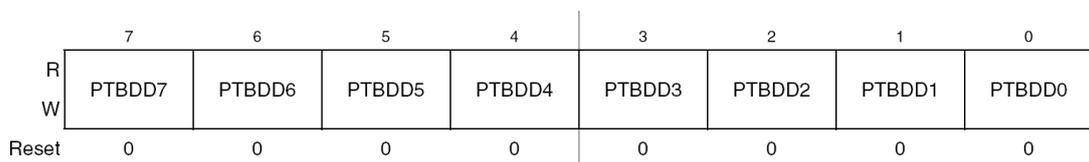


图6-15 端口B数据方向寄存器 (PTBDD)

表 6-8 PTBDD 寄存器字段描述

字段	描述
7:0 PTBDD[7:0]	<p>端口 B 数据方向位— 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。</p> <p>0 输入 (输出驱动被禁止), 读数返回管脚值。</p> <p>1 B 端口输出驱动使能, PTBD 读数返回 PTBDn 内容。</p>

6.7.4 端口 B 引脚控制寄存器 (PTBPE , PTBSE , PTBDS)

I/O 控制以外, 端口 B 引脚还由以下列出的寄存器控制。

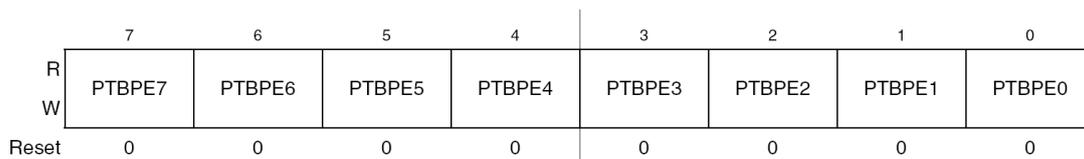


图6-16 端口B内部上拉允许

表 6-9 PTBPE 寄存器字段描述

字段	描述
7:0 PTBPE[7:0]	<p>端口 B 内部上拉使能位 — 对于 PTB 管脚, 这些控制位决定着为相关 PTB 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 B 端口管脚, 这些位不会产生影响, 同时内部上拉器件被禁止。</p> <p>0 B 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 B 端口位内部上拉 / 下拉器件使能。</p>

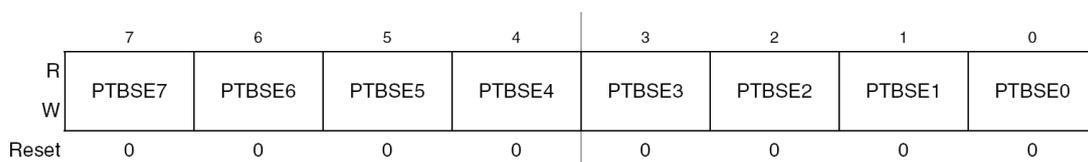


图6-17 端口B输出转换率控制允许（PTBSE）

表 6-10 PTBSE 寄存器字段描述

字段	描述
7:0 PTBSE[7:0]	<p>端口 B 输出转换率使能位 — 这些控制位决定着是否为相关 PTB 管脚使能输出转换率控制。对于配置为输入的 B 端口管脚，这些位不会产生任何影响。</p> <p>0 B 端口位输出转换率控制被禁止。</p> <p>1 B 端口位输出转换率控制使能。</p>

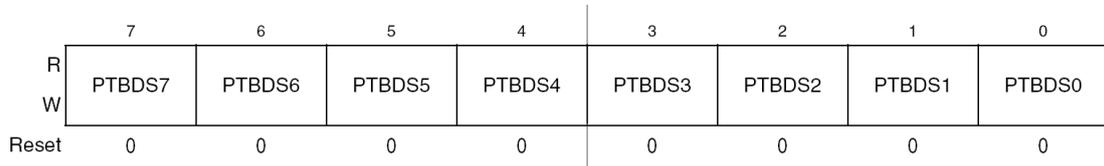


图6-18 端口B输出驱动强度选择（PTBDS）

表 6-11 PTBDS 寄存器字段描述

字段	描述
7:0 PTBDS[7:0]	<p>端口 B 输出驱动强度选择位 — 这些控制位为相关 PTB 管脚选择低输出驱动和高输出驱动。对于配置为输入的 B 端口管脚，这些位不会产生任何影响。</p> <p>0 B 端口位选择的低输出驱动强度。</p> <p>1 B 端口位选择的高输出驱动强度。</p>

6.7.5 端口 C I/O 寄存器（PTCD 和 PTCDD）

端口 C 并行 I/O 功能由下面列出的寄存器控制。

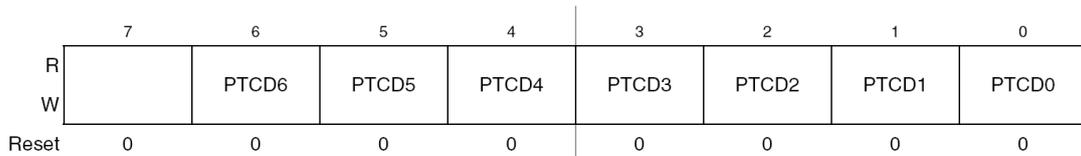


图6-19 端口C数据寄存器 (PTCD)

表 6-12 PTCD 寄存器字段描述

字段	描述
6:0 PTCD[6:0]	<p>端口 C 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。</p> <p>复位强制 PTCD 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。</p>

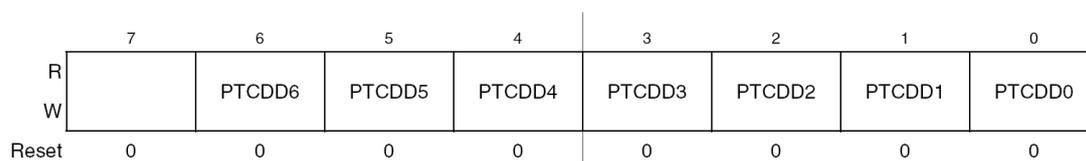


图6-20 端口C数据方向寄存器 (PTCDD)

表 6-13 PTCDD 寄存器字段描述

字段	描述
6:0 PTCDD[6:0]	<p>端口 C 数据方向位 — 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。</p> <p>0 输入（输出驱动被禁止），读数返回管脚值。</p> <p>1 C 端口输出驱动使能，PTCD 读数返回 PTCDDn 内容。</p>

6.7.6 端口 C 引脚控制寄存器 (PTCPE, PTCSE, PTCDS)

I/O 控制以外，端口 C 引脚还由以下列出的寄存器控制。

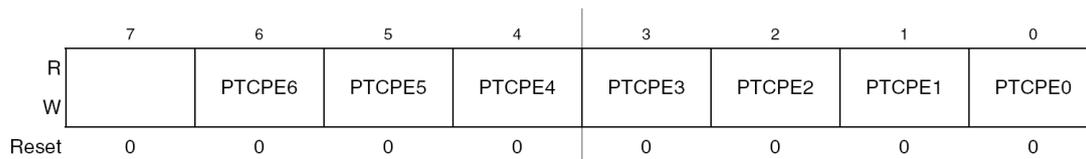


图6-21 端口C内部上拉允许

表 6-14 PTCPE 寄存器字段描述

字段	描述
6:0 PTCPE[6:0]	<p>端口 C 内部上拉使能位 — 对于 PTC 管脚, 这些控制位决定着为相关 PTC 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 C 端口管脚, 这些位不会产生影响, 同时内部上拉器件被禁止。</p> <p>0 C 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 C 端口位内部上拉 / 下拉器件使能。</p>

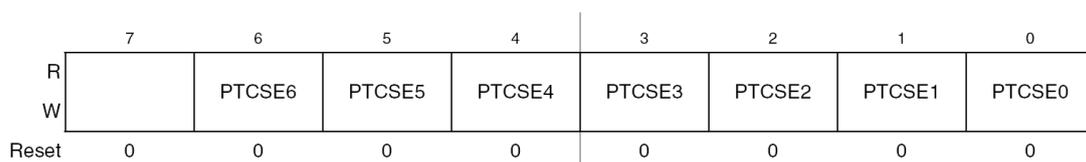


图6-22 端口C输出转换率控制允许 (PTCSE)

表 6-15 PTCSE 寄存器字段描述

字段	描述
6:0 PTCSE[6:0]	<p>端口 C 输出转换率使能位 — 这些控制位决定着是否为相关 PTC 管脚使能输出转换率控制。对于配置为输入的 C 端口管脚, 这些位不会产生任何影响。</p> <p>0 C 端口位输出转换率控制被禁止。</p> <p>1 C 端口位输出转换率控制使能。</p>

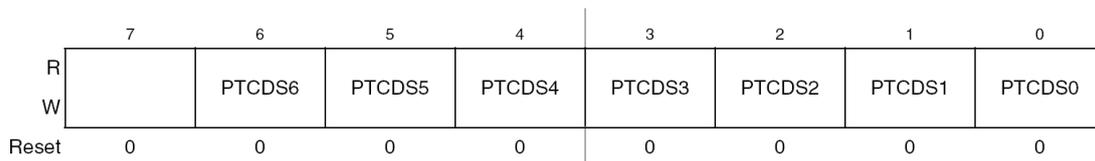


图6-23 端口C输出驱动强度选择 (PTCDS)

表 6-16 PTCDS 寄存器字段描述

字段	描述
6:0 PTCDS[6:0]	<p>端口 C 输出驱动强度选择位 — 这些控制位为相关 PTC 管脚选择低输出驱动和高输出驱动。对于配置为输入的 C 端口管脚，这些位不会产生任何影响。</p> <p>0 C 端口位选择的低输出驱动强度。</p> <p>1 C 端口位选择的高输出驱动强度。</p>

6.7.7 端口 DI/O 寄存器 (PTDD 和 PTDDD)

端口 D 并行 I/O 功能由下面列出的寄存器控制。

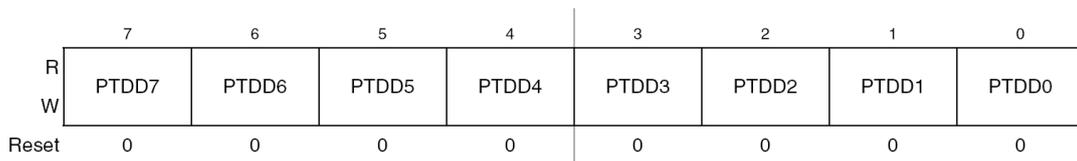


图6-24 端口D数据寄存器 (PTDD)

表 6-17 PTDD 寄存器字段描述

字段	描述
7:0 PTDD[7:0]	<p>端口 D 数据寄存器位 — 对于配置为输入的 D 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 D 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 D 端口管脚，逻辑电平被输出到相应的 MCU 管脚。</p> <p>复位强制 PTDD 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。</p>

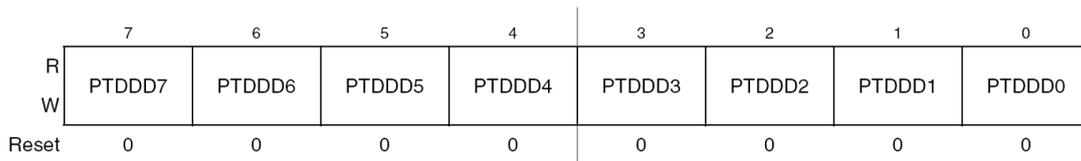


图6-25 端口D数据方向寄存器 (PTDDD)

表 6-18 PTDDD 寄存器字段描述

字段	描述
7:0 PTDDD[7:0]	<p>端口 D 数据方向位 — 这些读 / 写位控制着 D 端口管脚的方向以及为 PTDD 读数读取的内容。</p> <p>0 输入（输出驱动被禁止），读数返回管脚值。</p> <p>1 D 端口输出驱动使能，PTDD 读数返回 PTDDn 内容。</p>

6.7.8 端口 D 引脚控制寄存器（PTDPE，PTDSE，PTDDS）

I/O 控制以外，端口 D 引脚还由以下列出的寄存器控制。

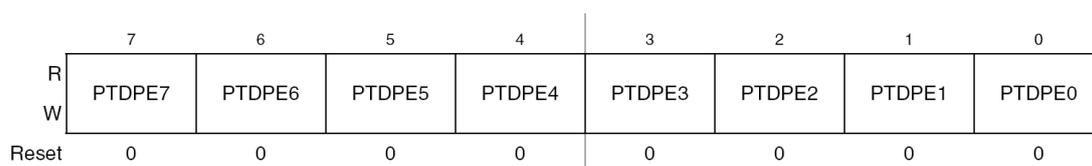


图6-26 端口D内部上拉允许

表 6-19 PTDPE 寄存器字段描述

字段	描述
7:0 PTDPE[7:0]	<p>端口 D 内部上拉使能位 — 对于 PTD 管脚，这些控制位决定着为相关 PTD 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 D 端口管脚，这些位不会产生影响，同时内部上拉器件被禁止。</p> <p>0 D 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 D 端口位内部上拉 / 下拉器件使能。</p>

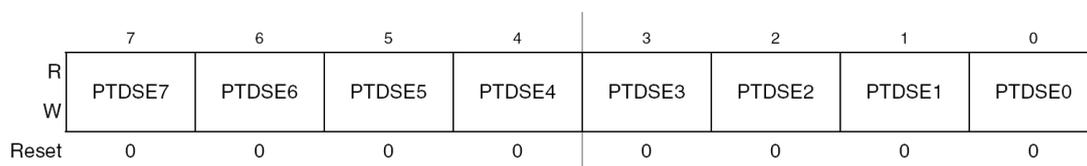


图6-27 端口D输出转换率控制允许（PTDSE）

表 6-20 PTDSE 寄存器字段描述

字段	描述
7:0 PTDSE[7:0]	<p>端口 D 输出转换率使能位 — 这些控制位决定着是否为相关 PTD 管脚使能输出转换率控制。对于配置为输入的 D 端口管脚，这些位不会产生任何影响。</p> <p>0 D 端口位输出转换率控制被禁止。</p> <p>1 D 端口位输出转换率控制使能。</p>

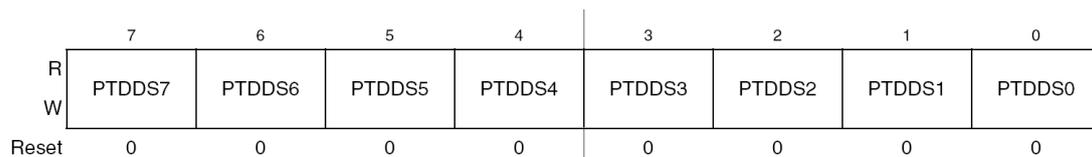


图6-26 端口D输出驱动强度选择 (PTDDDS)

表 6-21 PTDDDS 寄存器字段描述

字段	描述
7:0 PTDDDS[7:0]	<p>端口 D 输出驱动强度选择位 — 这些控制位为相关 PTD 管脚选择低输出驱动和高输出驱动。对于配置为输入的 D 端口管脚，这些位不会产生任何影响。</p> <p>0 D 端口位选择的低输出驱动强度。</p> <p>1 D 端口位选择的高输出驱动强度。</p>

6.7.9 端口 E I/O 寄存器 (PTED 和 PTEDD)

端口 E 并行 I/O 功能由下面列出的寄存器控制。

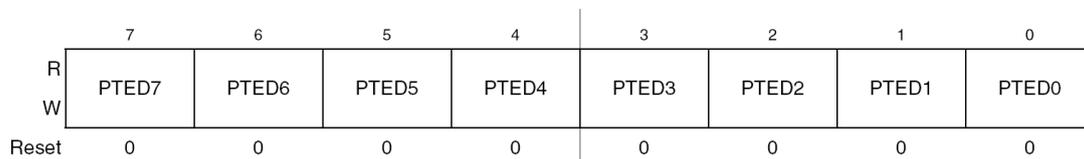


图6-29 端口E数据寄存器 (PTED)

表 6-22 PTED 寄存器字段描述

字段	描述
----	----

7:0 PTED[7:0]	<p>端口 E 数据寄存器位 — 对于配置为输入的 E 端口管脚，读数返回管脚上的逻辑电平。对于配置为输出的 E 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 E 端口管脚，逻辑电平被输出到相应的 MCU 管脚。</p> <p>复位强制 PTED 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。</p>
------------------	--

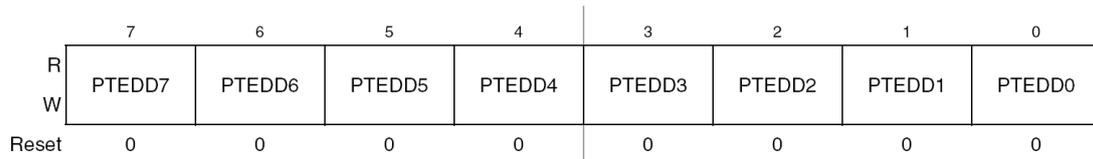


图6-30 端口E数据方向寄存器 (PTEDD)

表 6-23 PTEDD 寄存器字段描述

字段	描述
7:0 PTEDD[7:0]	<p>端口 E 数据方向位 — 这些读 / 写位控制着 E 端口管脚的方向以及为 PTED 读数读取的内容。</p> <p>0 输入 (输出驱动被禁止)，读数返回管脚值。</p> <p>1 E 端口输出驱动使能， PTED 读数返回 PTEDn 内容。</p>

6.7.10 端口 E 引脚控制寄存器 (PTEPE , PTESE , PTEDS)

I/O 控制以外，端口 E 引脚还由以下列出的寄存器控制。

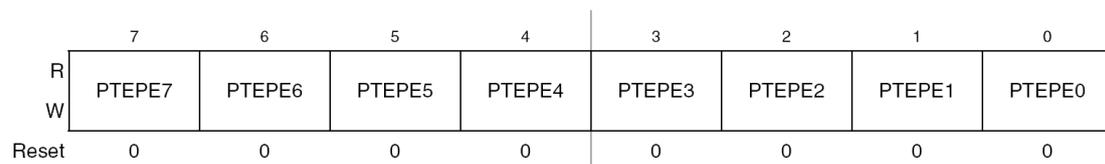


图6-31 端口E内部上拉允许

表 6-24 PTEPE 寄存器字段描述

字段	描述
7:0	端口 E 内部上拉使能位 — 对于 PTE 管脚，这些控制位决定着为相关 PTE 管脚使能内部

PTEPE[7:0]	<p>上拉还是使能下拉器件。对于配置为输出的 E 端口管脚，这些位不会产生影响，同时内部上拉器件被禁止。</p> <p>0 E 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 E 端口位内部上拉 / 下拉器件使能。</p>
------------	---

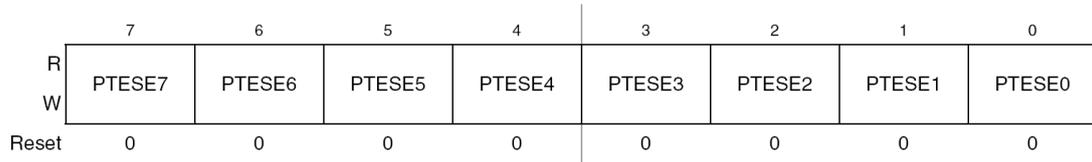


图6-32 端口E输出转换率控制允许 (PTESE)

表 6-25 PTESE 寄存器字段描述

字段	描述
7:0 PTESE[7:0]	<p>端口 E 输出转换率使能位 — 这些控制位决定着是否为相关 PTE 管脚使能输出转换率控制。对于配置为输入的 E 端口管脚，这些位不会产生任何影响。</p> <p>0 E 端口位输出转换率控制被禁止。</p> <p>1 E 端口位输出转换率控制使能。</p>

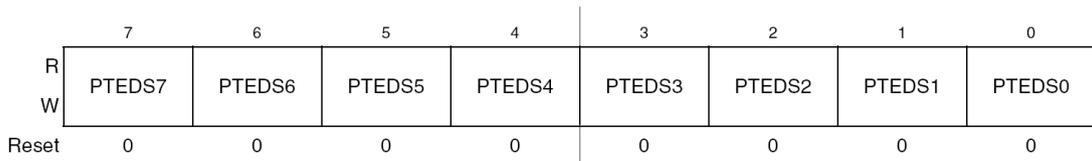


图6-33 端口E输出驱动强度选择 (PTEDS)

表 6-26 PTEDS 寄存器字段描述

字段	描述
7:0 PTEDS[7:0]	<p>端口 E 输出驱动强度选择位 — 这些控制位为相关 PTE 管脚选择低输出驱动和高输出驱动。</p> <p>0 E 端口位选择的低输出驱动强度。</p>

	1 E 端口位选择的高输出驱动强度。
--	--------------------

6.7.11 端口 F I/O 寄存器 (PTFD 和 PTFDD)

端口 F 并行 I/O 功能由下面列出的寄存器控制。

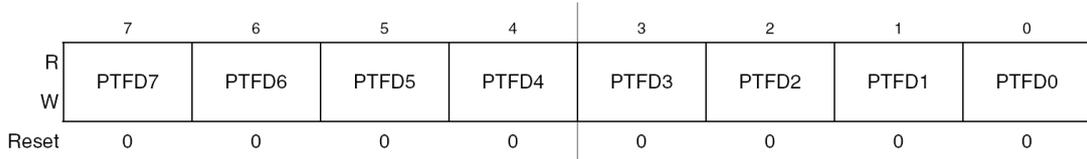


图6-34 端口F数据寄存器 (PTFD)

表 6-27 PTFD 寄存器字段描述

字段	描述
7:0 PTFD[7:0]	<p>端口 F 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。</p> <p>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，逻辑电平被输出到相应的 MCU 管脚。</p> <p>复位强制 PTFD 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。</p>

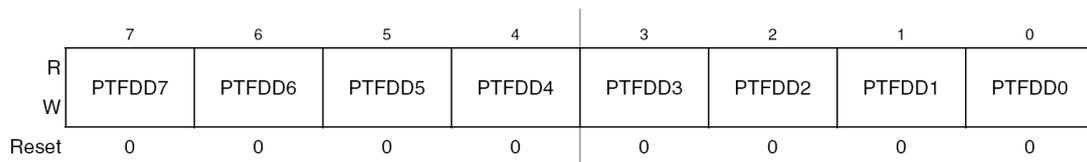


图6-35 端口F数据方向寄存器 (PTFDD)

表 6-28 PTFDD 寄存器字段描述

字段	描述
7:0 PTFDD[7:0]	<p>端口 F 数据方向位 — 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。</p> <p>0 输入 (输出驱动被禁止)，读数返回管脚值。</p>

	1 F 端口输出驱动使能，PTFD 读数返回 PTFDn 内容。
--	----------------------------------

6.7.12 端口 F 引脚控制寄存器 (PTFPE , PTFSE , PTFDS)

I/O 控制以外，端口 F 引脚还由以下列出的寄存器控制。

	7	6	5	4	3	2	1	0
R W	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
Reset	0	0	0	0	0	0	0	0

图6-36 端口F内部上拉允许

表 6-29 PTFPE 寄存器字段描述

字段	描述
7:0 PTFPE[7:0]	端口 F 内部上拉使能位 — 对于 PTF 管脚，这些控制位决定着为相关 PTF 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 F 端口管脚，这些位不会产生影响，同时内部上拉器件被禁止。 0 F 端口位内部上拉 / 下拉器件被禁止。 1 F 端口位内部上拉 / 下拉器件使能。

	7	6	5	4	3	2	1	0
R W	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
Reset	0	0	0	0	0	0	0	0

图6-37 端口F输出转换率控制允许 (PTFSE)

表 6-30 PTFSE 寄存器字段描述

字段	描述
7:0 PTFSE[7:0]	端口 F 输出转换率使能位 — 这些控制位决定着是否为相关 PTF 管脚使能输出转换率控制。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 端口位输出转换率控制被禁止。 1 F 端口位输出转换率控制使能。

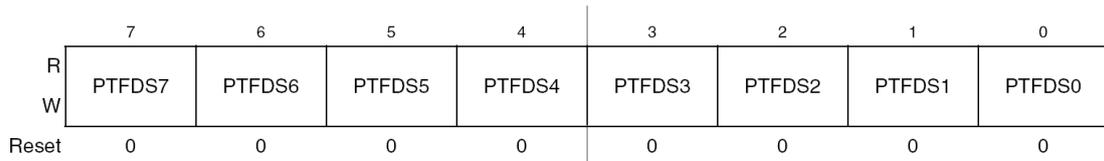


图6-38 端口F输出驱动强度选择 (PTFDS)

表 6-31 PTFDS 寄存器字段描述

字段	描述
7:0 PTFDS[7:0]	<p>端口 F 输出驱动强度选择位 — 这些控制位为相关 PTF 管脚选择低输出驱动和高输出驱动。</p> <p>0 F 端口位选择的低输出驱动强度。</p> <p>1 F 端口位选择的高输出驱动强度。</p>

6.7.13 端口 G I/O 寄存器 (PTGD 和 PTGDD)

端口 G 并行 I/O 功能由下面列出的寄存器控制。

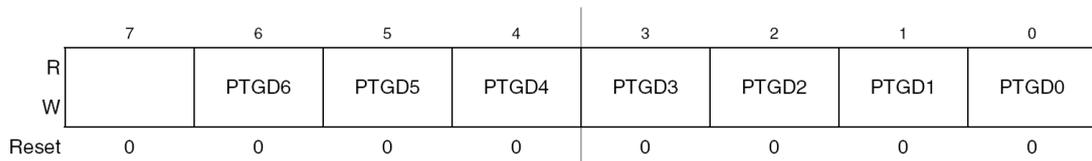


图6-39 端口G数据寄存器 (PTGD)

表 6-32 PTGD 寄存器字段描述

字段	描述
6:0 PTGD[6:0]	<p>端口 G 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。</p> <p>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。</p> <p>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。</p>

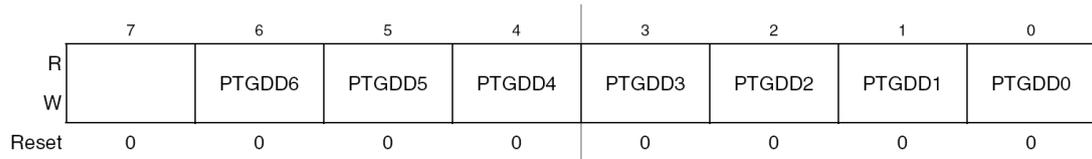


图6-40 端口G数据方向寄存器 (PTGDD)

表 6-33 PTGDD 寄存器字段描述

字段	描述
6:0 PTGDD[6:0]	<p>端口 G 数据方向位 — 这些读 / 写位控制着 G 端口管脚的方向以及为 PTGD 读数读取的内容。</p> <p>0 输入 (输出驱动被禁止), 读数返回管脚值。</p> <p>1 G 端口输出驱动使能, PTGD 读数返回 PTGD_n 内容。</p>

6.7.14 端口 G 引脚控制寄存器 (PTGPE , PTGSE , PTGDS)

I/O 控制以外, 端口 G 引脚还由以下列出的寄存器控制。

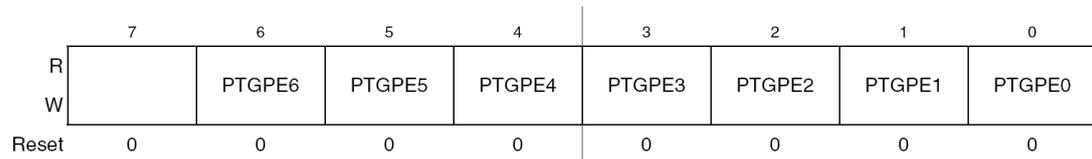


图6-41 端口G内部上拉允许

表 6-34 PTGPE 寄存器字段描述

字段	描述
6:0 PTGPE[6:0]	<p>端口 G 内部上拉使能位 — 对于 PTG 管脚, 这些控制位决定着为相关 PTG 管脚使能内部上拉还是使能下拉器件。对于配置为输出的 G 端口管脚, 这些位不会产生影响, 同时内部上拉器件被禁止。</p> <p>0 G 端口位内部上拉 / 下拉器件被禁止。</p> <p>1 G 端口位内部上拉 / 下拉器件使能。</p>

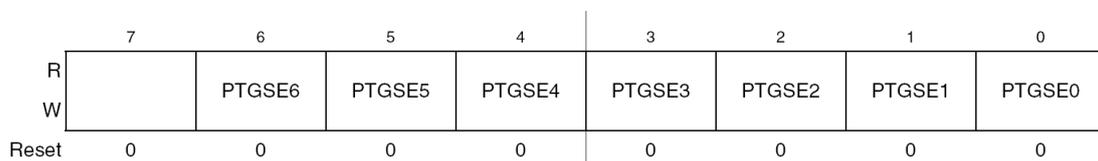


图6-42 端口G输出转换率控制允许（PTGSE）

表 6-35 PTGSE 寄存器字段描述

字段	描述
6:0 PTGSE[7:0]	<p>端口 G 输出转换率使能位 — 这些控制位决定着是否为相关 PTG 管脚使能输出转换率控制。对于配置为输入的 G 端口管脚，这些位不会产生任何影响。</p> <p>0 G 端口位输出转换率控制被禁止。</p> <p>1 G 端口位输出转换率控制使能。</p>

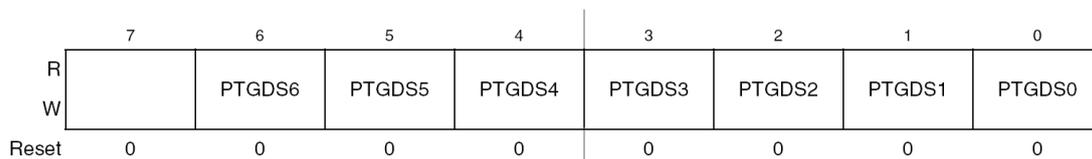


图6-43 端口G输出驱动强度选择（PTGDS）

表 6-36 PTGDS 寄存器字段描述

字段	描述
6:0 PTGDS[6:0]	<p>端口 G 输出驱动强度选择位 — 这些控制位为相关 PTG 管脚选择低输出驱动和高输出驱动。</p> <p>0 G 端口位选择的低输出驱动强度。</p> <p>1 G 端口位选择的高输出驱动强度。</p>

第七章 中央处理单元(S08CPUV2)

7.1 绪论

本章是 HCS08 系列 CPU 的寄存器、寻址方式和指令集等信息的概述。更详细的讨论请参考 HCS08 系列参考手册第一卷，飞思卡尔半导体文档单号 HCS08RMv1/D。

HCS08 CPU 的源代码和目标代码与 M68HC08 CPU 的完全兼容。一些指令和加强的寻址方式提高了 C 编译器的效率，并且支持一个新的背景调试系统，这个系统取代了较早的 M68HC08 MCU 的监控模式。

7.1.1 特性

HCS08 CPU 的主要特点是：

- 目标代码与 M68HC05、M68HC08 系列向上兼容
- 寄存器和存储器具有一个单独的 64K 字节的地址空间
- 16 位堆栈指针(不管堆栈多大都是在 64K 字节的地址空间内)
- 16 位变址寄存器(H:X)的强大寻址索引模式
- 许多指令把 X 看做第二个 8 位通用寄存器
- 7 种寻址方式：
 - 内在——操作数在内部寄存器
 - 相对——8 位带符号的偏移量指向跳转目的地址
 - 立即——操作数在下一个目标代码字节
 - 直接——操作数位于存储空间 0x0000~0x00FF
 - 扩展——操作数在 64K 字节地址空间的任何地址
 - 变址相对 H:X——包括自增加的五种模式
 - 变址相对 SP——显著增强 C 效率
- 存储器到存储器数据搬移指令通过 4 种寻址方式组合
- 溢出、半进位、负数、置零和进位进行条件代码支持有符号、无符号和二进制编码的十进制操作结果的条件分支。
- 高效率的位操作指令
- 快速 8 位与 8 位相乘和快速 16 位与 8 位相除指令
- STOP 和 WAIT 指令唤醒低功耗运行状态

7.2 编程模式和CPU寄存器

5 个 CPU 寄存器的结构示意图如图 7-1 所示。CPU 寄存器不属于存储器的部分。

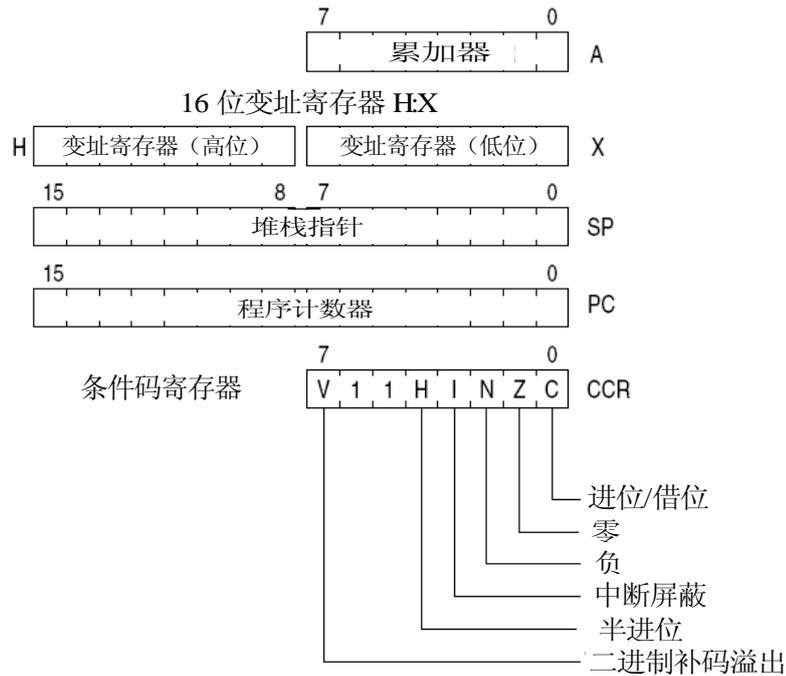


图 7-1 CPU 寄存器

7.2.1 累加器(A)

累加器 A 是 8 位通用寄存器。首先累加器存放一个写入到算术逻辑单元(ALU)的操作数，在执行算术、逻辑操作后，累加器存放操作结果。数据读取时，累加器用于存放从存储器读书的数据；数据写入时，累加器用于存放准备写入存储器的数据。复位时，累加器 A 中的内容不受影响。

7.2.2 变址寄存器(H:X)

16 位寄存器是两个分开的 8 位寄存器(H 和 X)，一起使用时作为一个 16 位的地址指针，H 是高 8 位，X 是低 8 位。所有变址寻址方式指令用 H:X 的 16 位作为一个变址参考指针；然而，鉴于与早前的 M68HC05 系列的兼容性，一些指令只运用低 8 位(X)。

许多指令把 X 作为第二个通用 8 位寄存器，可以用来存储 8 位数据。X 可以被清 0、递增、递减、求补、取反、转移或是跳转。转移指令允许把累加器 A 中的数据转移出来或是把数据转移到 A 中去执行数字逻辑运算。

鉴于与早前的 MC68HC05 系列的兼容性，复位时，H 被清零。但是复位时，X 的内容不受影响。

7.2.3 堆栈指针(SP)

这个 16 位地址指针寄存器指向下一个栈地址，堆栈是自动按照后进先出(LIFO)。这个堆栈可以位于带有 RAM 的 64K 字节地址空间的任一位置，而且大小可达到 RAM 的最高限度。堆栈是在子程序的调用时自动保存返回地址，在中断时保存返回地址和 CPU 寄存器，并且为变量。AIS(堆栈指针立即增加)指令是使堆栈指针 SP 加上一个 8 位带符号的立即数。对于在堆栈上的变量来说，最常用的是分配或是回收空间。

鉴于与 M68HC05 系列的兼容性，复位时 SP 的值是 0x00FF。在上电复位初始化时，HCS08 编程通常改变 SP 的值，以指向片内 RAM 最后的地址(最高地址)，用来释放直接页 RAM(从片上最后寄存器到 0x00FF)。

鉴于与 M68HC05 系列的兼容性，RSP(栈指针复位)指令包含在内。而且很少用在新的 HCS08 编程中，因为它只影响低位。

7.2.4 程序计数器(PC)

程序计数器是个 16 位的寄存器，它包含了下一条指令的地址或被取出来的操作。

正常运行时，程序计数器在每次取出指令或操作数之后，会自动增加以指向下一个要执行的时序的存储器位置。跳转、分支、中断和返回操作会使程序计数器加载另一个地址而不是下个时序存储器的位置，这被称为流程改变。

复位时，程序计数器是带有复位向量，位于 0xFFFFE 和 0xFFFF。在复位后，存储的向量是要执行的第一个指令地址。

7.2.5 条件码寄存器(CCR)

8 位条件码寄存器是包括中断屏蔽标志(I)和 5 个用于指示执行完指令的结果状态。第 6 位和第 5 位永远为 1。如图 7-2 所示，用专业术语介绍一下条件码的各个位。更详细的关于怎样设置每个位的讨论请参考 HCS08 系列参考手册第一卷，飞思卡尔半导体文档编号 HCS08RMv1。

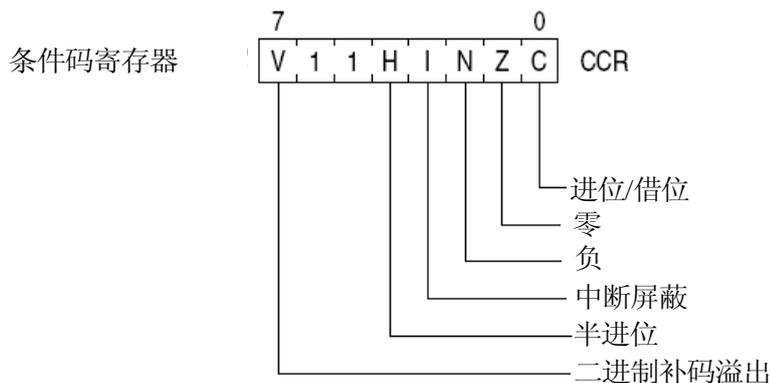


图 7-2 条件码寄存器

表 7-1 CCR 寄存器描述

域	描述
7	二进制补码溢出标志——当二进制补码溢出时 CPU 将溢出标志置位。有符号跳转指令 BGT、BGE、BLE 和 BLT 使用该标志。
V	0 无溢出 1 有溢出
4	半进位标志——执行加法指令(ADD)和带进位加法指令(ADC)时，当累加器 D3 向 D4 有进位时，CPU 将半进位置位。该标志要求二进制编码的十进制(BCD)数据运算。DAA 指令利用 H 和 C 的状态条件编码位来自动把一个正确的数据加入
H	

结果从早先的 BCD 型的 ADD 或是 ADC 运算到正确的结果，以得到有效的 BCD 值。

0 第 3 位和第 4 位之间没有进位

1 第 3 位和第 4 位之间有进位

3 I **中断屏蔽位**——当设置中断屏蔽时，所有的 CPU 禁止中断。当中断屏蔽位为 0 时，CPU 中断允许。中断响应时，CPU 将寄存器推入堆栈，然后执行中断服务子程序，最后自动设置中断屏蔽位。

任何清除 I(CLI 或是 TAP)的指令之后中断不能认出边界指令。这确保在 CLI 或是 TAP 后的下一个指令在没有内部中断的可能性下将要执行，I 已被设置。

0 中断使能

1 中断屏蔽

2 N **负标志**——在进行算数运算、逻辑运算或是数据处理时产生负结果，CPU 将设置负标志，即结果的第 7 位(D7)。载入或存储一个 8 位或 16 位数决定 N 的状态，即最高位是否是 1。

0 结果为正

1 结果为负

1 Z **零标志**——在进行算数运算、逻辑运算或是数据处理时结果为 0，CPU 将 0 标志置位。

载入或存储一个 8 位或 16 位数是否是 0，决定了 Z 的状态。

0 结果非 0

1 结果是 0

0 C **进位/借位标志**——当进行加法运算时，在 D7 上有进位；或在进行减法运算时 D7 需要向更高位借位，则 CPU 将进位/借位标志置位。一些指令如位测试、跳转、转移等指令也会影响该标志。

0 第 7 位没有执行

1 第 7 位执行

7.3 寻址方式

寻址方式是 CPU 存取运算单元和数据的方式。在 HCS08 中，所有的存储器、状态寄存器、控制寄存器和输入输出端口都共用一个 64K 字节的线性地址空间，所以一个 16 位二进制地址能识别任何一个存储位置。相同的指令可以访问 RAM 中的变量也可访问 I/O 和控制寄存器或是不易失性程序空间。

一些指令使用多种寻址方式。如转移指令用一种寻址方式来指定源操作，另一种寻址方式是指定目标地址。例如一些 BRCLR、BRSET、CBEQ 和 DBNZ 指令用一种寻址方式来指定测试的操作位置，当测试条件为真时，用相对寻址方式来指定跳转目的地址。对于 BRCLR、BRSET、CBEQ 和 DBNZ 的寻址方式，在指令表中设置一种寻址方式是需要访问将要测试操作的，还有一种相对寻址方式是指定跳转目的地址的。

7.3.1 内在寻址方式(INH)

这种寻址方式是在 CPU 内部寄存器之间进行指令操作的，所以 CPU 不需要访问存储器来操作。

7.3.2 相对寻址方式(REL)

相对寻址方式用来指定跳转指令的目的位置。一个带符号 8 位偏移量的数位于存储位置，紧跟存储器中的指令操作码。运行时，如果跳转指令为真，带符号偏移量扩展到 16 位有符号数，同时也把它加到当前程序计数器中去，这使得程序在跳转目的地址时能够继续运行。

7.3.3 立即数寻址方式(IMM)

立即寻址方式，需要完成的指令操作是包括在目标代码中，紧跟存储器中的指令操作码。要是 16 位立即数操作，高字节位于在操作码之后的下一个存储位置，低字节位于高字节之后下一个存储位置。

7.3.4 直接寻址方式(DIR)

直接寻址方式，指令包括一个地址的低 8 位在直接页(0x0000~0x00FF)中。运行时，一个 16 位的地址高 8 位被默认为 0x00，指令中的直接地址获得了 16 位地址，用来存放所需要的操作。这个比指定一个完整的 16 位操作地址速度更快，存储效率更高。

7.3.5 扩展寻址方式(EXT)

扩展寻址方式，操作的 16 位地址位于程序存储器中的两个字节，在操作码之后(首先是高字节)。

7.3.6 变址寻址方式

变址寻址方式有 7 种，包括 5 种用到 16 位的 H:X 变址寄存器，2 种用到堆栈指针作为参考基数。

7.3.6.1 无偏移量变址(IX)

变址寻址的变量是用 H:X 变址寄存器中 16 位数作为需要完成指令的操作地址。

7.3.6.2 无偏移量变址，变址加 1(IX+)

变址寻址的变量是用 H:X 变址寄存器中 16 位数作为需要完成指令的操作地址。操作被取出之后，变址寄存器加 1(H:X= H:X+1)。这个寻址方式只用在 MOV 和 CBEQ 指令中。

7.3.6.3 8 位偏移量变址(IX1)

变址寻址的变量用 H:X 变址寄存器中 16 位数加上在指令中的一个不带符号的 8 位偏移量，其和便是操作数地址。

7.3.6.4 8 位偏移量变址，变址加 1(IX1+)

变址寻址的变量用 H:X 变址寄存器中 16 位数加上在指令中的一个不带符号的 8 位偏移量，其和便是操作数地址。操作被取出之后，变址寄存器加 1(H:X= H:X+1)。这个寻址方式只用在 CBEQ 指令中。

7.3.6.5 16 位偏移量变址(IX2)

变址寻址的变量用 H:X 变址寄存器中 16 位数加上在指令中的一个不带符号的 16 位偏移量，其和便是操作数地址。

7.3.6.6 8 位偏移量堆栈(SP1)

变址寻址的变量用堆栈指针(SP)中 16 位数加上在指令中的一个不带符号的 8 位偏移量，其和便是操作数地址。

7.3.6.7 16 位偏移量变址(SP2)

变址寻址的变量用堆栈指针(SP)中 16 位数加上在指令中的一个不带符号的 16 位偏移量，其和便是操作数地址。

7.4 特殊操作

CPU 执行一些特殊操作，那些操作与指令相似，但是没有像其他 CPU 指令的操作码。除此之外，一些指令如 STOP 和 WAIT 会直接影响其他 MCU 的电路。本章提供了关于这些操作的额外信息。

7.4.1 复位顺序

复位有多种情况，上电复位(POR)、看门狗复位、低电压禁止复位。复位时，CPU 停止一切活动(在响应复位之前，MCU 不会等待一个边界指令)。更为详细的讨论关于 MCU 如何识别复位和确定来源，参考复位、中断和系统配置章节。

复位结束是在当时序完成时，即确定复位信号是否来自于内部电源的时序，或是当复位引脚不再报警。复位，CPU 用 6 个周期的时序从地址 0xFFFFE 和 0xFFFF 取出复位向量，并且把运行的第一个指令存进去。

7.4.2 中断顺序

有中断请求时，CPU 在响应中断之前完成当前的指令。同时，程序计数器指向下一个指令的开始端，当中断结束之后 CPU 放回到这个地方。CPU 通过执行操作时序来响应中断，这个操作是关于一个软件中断(SWI)指令。除了用来提取向量的地址，这个地址是按照最高优先级中断决定的。

CPU 中断的过程：

- 1.PCL、PCH、X、A、CCR 依次进栈。
- 2.设置 CCR 的 I 位。
- 3.取出中断向量的高字节。
- 4.取出中断向量的低字节。
- 5.延时一个空闲总线周期。
- 6.取出中断矢量指向地址程序信息的三个字节，填充指令队列准备执行中断服务子程序的第一条指令。

CCR 中的内容进栈后，CCR 中的 I 位置 1 以在中断服务子程序中防止其它的中断。尽管可能在中断服务子程序中用一条指令清除 I 位，这会导致中断嵌套。(因为这会导致程序很难运行和维护，这不推荐)。

为了保持与早期 M68HC05 系列 MCU 的兼容，H:X 变址寄存器高位没有进栈，未被保存。使用者在服务程序开始时用一个 PSHH 指令保存 H，然后在结束中断服务指令 RIT 之前用 PULH 指令。如果确知中断服务程序没有用到任何指令或是会改变 H 的自动增加寻址方式，就没必要保存 H。

软件中断(SWI)指令像硬件中断一样，除了它不用 CCR 中的 I 位标志外，它和程序内的一个指令操作码相关，所以程序执行时它不是异步的。

7.4.3 等待模式操作

等待指令是通过清除 CCR 的 I 位来中断的。停止 CPU 的时钟来减少总消耗，与此同时 CPU 等待中断或复位，那是将 CPU 从等待状态唤醒过来。运行中断或是复位时，CPU 时钟将恢复，中断和复位正常处理。

如果一个串口背景命令通过背景调试接口被发到 MCU 上，而 CPU 是在等待模式下，CPU 的时钟将恢复，CPU 将进入活跃的背景模式，在这种模式下其它串行背景命令可以得到处理。这可确保主机系统的发展仍然可以获得一个目标 MCU，即使是在等待模式。

7.4.4 停止模式操作

通常，所有的系统时钟包括晶体振荡器(使用时)，在停止模式时停止，尽量减少电源消耗。在这样的系统中，外部电路需要控制在停止模式下的时间，同时需要恢复处理时发出一个信号来唤醒目标 MCU。不像早前的 M68HC05 和 M68HC08 系列 MCU，通过设置 HCS08，可以使一组最小的时钟一直运行在停止模式中。这可随意地允许一个内部信号从停止模式中唤醒目标 MCU。

当一个主机调试系统连接到背景调试引脚(BKGD)，而 ENBDM 控制位用一个串口通过背景接口来设置(或因为 MCU 重新到活跃的背景模式)，当 MCU 进入停止模式时，振荡器仍保持活跃。在这种情况下，如果一个串口背景命令通过背景调试接口被发到 MCU 上，而 CPU 是在停止模式下，CPU 的时钟将恢复，CPU 将进入活跃的背景模式，在这种模式下

其它串行背景命令可以得到处理。这可确保主机系统的发展仍然可以获得一个目标 MCU，即使是在等待模式。

从停止模式中恢复过来是取决于特定的 HCS08 以及是否振荡器停在停止模式。更详细的介绍参考运行的模式章节。

7.4.5 BGND 指令

相比较 M68HC08，BGND 是 HCS08 中新增的指令。因为 BGND 强迫 CPU 停止处理用户指令和进入积极的背景状态，指令不会在正常用户中使用。从用户程序的执行重返回的方法仅有复位或者上位机调试系统通过背景调试接口发起 GO、TRACE1 或 TAGGO 串行命令。

使用 BGND 操作码能够在期望的断点位置放置一条操作码，从而能够设置基于软件的断点。当程序到达这个断点地址，CPU 强制为积极的背景状态而不是继续用户程序。

7.5 HCS08指令集摘要

指令集概括术语

这里列出的术语用于表 7-2 中的指令描述。

运算符

() = 圆括号内显示的是寄存器或存储单元的内容

← = 被载入(读：“得到”)

& = 逻辑与

| = 逻辑或

⊕ = 逻辑异或

× = 乘

÷ = 除

:

+

-

CPU 寄存器

A = 累加器

CCR = 条件码寄存器

H = 变址寄存器，高顺序(最高有效)8 位

X = 变址寄存器, 低顺序(最低有效)8 位

PC = 程序计数器

PCH = 程序计数器, 高顺序(最高有效)8 位

PCL = 程序计数器, 低顺序(最低有效)8 位

SP = 堆栈指针

存储器 and 地址

M = 存储单元或绝对数据, 取决于寻址模式

M: = 在两个连续的存储单元 16 位数值。高顺序(最高有效)8 位存

M+0x0001

在 **M** 地址处, 低顺序(最高有效)8 位存在下个更高的地址。

条件码寄存器(CCR)位

V = 二进制补码溢出标志, 位 7

H = 半进位, 位 4

I = 中断屏蔽, 位 3

N = 负标志, 位 2

Z = 零标志, 位 1

C = 进位/借位, 位 0 (通过位 7 执行)

CCR 的行为标记

- = 位不受影响

0 = 位强制清 0

1 = 位强制置 1

P = 依据操作结果, 置位或清零

U = 操作后未定义

机器码标记

-
- dd** = 一个直接地址 0x0000–0x00FF(高字节假定是 0x00)的低八位
- ee** = 16 位偏移量的高 8 位
- ff** = 16 或 8 位偏移量的低 8 位
- ii** = 一个立即数字节
- jj** = 一个 16 位立即数高字节
- kk** = 一个 16 位立即数低字节
- hh** = 16 位扩展地址的高字节
- ll** = 16 位扩展地址的低字节
- rr** = 相对偏移量

源形式

在源表格栏内的每个信息，是字面上的信息，它必须出现在精确显示的汇编源文件中。头 3 到 5 个字母助记符总是一个文字表达式。所有逗号，磅符(#)和加号(+)字符符号。

- n** — 任何标签或表达式用于表示一个在 0–7 范围内的单整型
- opr8i** — 任何标签或表达式用于表示一个 8 位立即数值。
- opr16i** — 任何标签或表达式用于表示一个 16 位立即数值。
- opr8a** — 任何标签或表达式用于表示一个 8 位数值。指令把这 8 位数值作为 64K 字节地址空间(0x00xx)的地址的低 8 位。
- opr16a** — 任何标签或表达式用于表示一个 16 位数值。指令把这格数值作为 64K 字节地址空间的一个地址。
- opr8** — 任何标签或表达式用于表示一个不带符号的 8 位数值。用在变址寻址中。
- opr16** — 任何标签或表达式用于表示一个 16 位数值。因为 HCS08 有一个 16 位地址总线，它要么是带符号的要么是不带符号的数值。

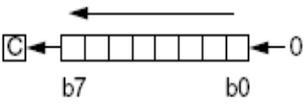
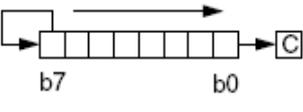
rel — 范围在与当前指令的目标代码的最后一个字节的下一个地址相差-128~127 单元的地址的标记或表达式。编译器将计算这 8 位有符号偏移地址，将其加到指令的目标代码中。

寻址方式

- INH** = 内在寻址（无操作数）
- IMM** = 8 位或 16 位立即寻址
- DIR** = 8 位直接寻址
- EXT** = 16 位扩展寻址
- IX** = 16 位无偏移量变址
- IX+** = 16 位无偏移量变址，变址加 1(只用于 CBEQ、MOV)
- IX1** = 16 位变址带 8 位偏移量来自于 H:X
- IX1+** = 16 位带 8 位偏移量变址，变址加 1(只用于 CBEQ)
- IX2** = 16 位变址带 16 位偏移量来自于 H:X
- REL** = 8 位相对偏移
- SP1** = 8 位偏移量堆栈
- SP2** = 16 位偏移量堆栈

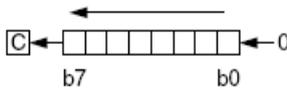
表 7-1 指令集

源格式	操作	描述											
			V	H	I	N	Z	C					
ADC #opr8i	带进位加	$A \leftarrow (A) + (M) + (C)$							IMM	A9	ii	2	
ADC opr8a									DIR	B9	dd	3	
ADC opr16a									EXT	C9	hh ll	4	
ADC oprx16,X									IX2	D9	ee ff	4	
ADC oprx8,X									IX1	E9	ff	3	
ADC ,X									IX	F9		3	
ADC oprx16,SP									SP2	9ED9	ee ff	5	
ADC oprx8,SP									SP1	9EE9	ff	4	
ADD #opr8i			不带进位加	$A \leftarrow (A) + (M)$						IMM	AB	ii	2
ADD opr8a								DIR	BB	dd	3		
ADD opr16a								EXT	CB	hh ll	4		

ADD oprx16,X ADD oprx8,X ADD ,X ADD oprx16,SP ADD oprx8,SP										IX2 IX1 IX SP2 SP1	DB EB FB 9EDB 9EEB	ee ff ff ff ee ff ff	4 3 3 5 4
AIS #opr8i	给堆栈加一个立即数值(有符号)	$SP \leftarrow (SP) + (M)$ M 是带符号扩展 16 位数值	—	—	—	—	—	—	—	IMM	A7	ii	2
AIX #opr8i	给变址寄存器(H:X)加一个立即数值(有符号)	$H:X \leftarrow (H:X) + (M)$ M 是带符号扩展 16 位数值	—	—	—	—	—	—	—	IMM	AF	ii	2
AND #opr8i AND opr8a AND opr16a AND oprx16,X AND oprx8,X AND ,X AND oprx16,SP AND oprx8,SP	逻辑与	$A \leftarrow (A) \& (M)$	0 —	— ↑	— ↓	— ↑	— ↓	—	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 B4 C4 D4 E4 F4 9ED4 9EE4	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
ASL opr8a ASLA ASLX ASL oprx8,X ASL ,X ASL oprx8,SP	算术左移(同 LSL)		↑ ↓	— —	↑ ↓	↑ ↓	— —	↑ ↓	—	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff ff ff	5 1 1 5 4 6
ASR opr8a ASRA ASRX ASR oprx8,X ASR ,X ASR oprx8,SP	算术右移		↑ ↓	— —	↑ ↓	↑ ↓	— —	↑ ↓	—	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff ff ff	5 1 1 5 4 6
BCC rel	进位为 0 跳转	if (C) = 0, 跳转	—	—	—	—	—	—	—	REL	24	rr	3
BCLR n,opr8a	清存储器单元第 n 位	$M_n \leftarrow 0$	—	—	—	—	—	—	—	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS rel	进位为 1 跳转(同 BLO)	if(C)=1, 跳转	—	—	—	—	—	—	—	REL	25	rr	3
BEQ rel	等于跳转	if(Z)=1, 跳转	—	—	—	—	—	—	—	REL	27	rr	3
BGE rel	大于或等于转跳	if(N ⊕ V)=1, 跳转	—	—	—	—	—	—	—	REL	90	rr	3
BGND	后台调试	进入后台调试模式	—	—	—	—	—	—	—	INH	82	rr	5+
BGT rel	大于跳转(有符号操作)	If (Z) (N ⊕ V)=0, 跳转	—	—	—	—	—	—	—	REL	92	rr	3
BHCC rel	半进位为 0 跳转	if (H) = 0, 跳转	—	—	—	—	—	—	—	REL	28	rr	3
BHCS rel	半进位为 1 跳转	if (H) = 1, 跳转	—	—	—	—	—	—	—	REL	29	rr	3
BHI rel	更高跳转	if (C) (Z) = 0, 跳转	—	—	—	—	—	—	—	REL	22	rr	3
BHS rel	更高或相同(BCC) 跳转	if (C) = 0, 跳转	—	—	—	—	—	—	—	REL	24	rr	3

BIH rel	IRQ 引脚高, 跳转	if IRQ 引脚 = 1, 跳转	—	—	—	—	—	REL	2F	rr	3	
BIL rel	IRQ 引脚低, 跳转	if IRQ 引脚 = 0, 跳转	—	—	—	—	—	REL	2E	rr	3	
BIT #opr8i BIT opr8a BIT opr16a BIT oprx16,X BIT oprx8,X BIT ,X BIT oprx16,SP BIT oprx8,SP	位测试	(A) & (M) (更新 CCR , 但是操作不改变)	0	—	—	↑	↑	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 B5 C5 D5 E5 F5 9ED5 9EE5	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
BLE rel	小于或等于转移 (带符号的操作)	if (Z) (N ⊕ V) = 1, 跳转	—	—	—	—	—	REL	93	rr	3	
BLO rel	低于或相同转移 (与 BCS 相同)	if (C) = 1, 跳转	—	—	—	—	—	REL	25	rr	3	
BLS rel	低于或相同转移	if (C) (Z) = 1 转移	—	—	—	—	—	REL	23	rr	3	
BLT rel	小于转移 (带符号的操作)	if (N ⊕ V) = 1	—	—	—	—	—	REL	91	rr	3	
BMC rel	中断屏蔽清零转移	if (I) = 0 转移	—	—	—	—	—	REL	2C	rr	3	
BMI rel	相减转移	if (N) = 1 转移	—	—	—	—	—	REL	2B	rr	3	
BMS rel	中断屏蔽置位转移	if (I) = 1 转移	—	—	—	—	—	REL	2D	rr	3	
BNE rel	不相等转移	if (Z) = 0, 转移	—	—	—	—	—	REL	26	rr	3	
BPL rel	相加转移	if (N) = 0, 转移	—	—	—	—	—	REL	2A	rr	3	
BRA rel	总跳转	不测试	—	—	—	—	—	REL	20	rr	3	
BRCLR n,opr8a,rel	若(opr8a)第 n 位为 0, 则跳转	if(Mn)=0, 跳转	—	—	—	—	↑ ↓	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5	
BRN rel	不跳转	用 3 个总线周期	—	—	—	—	—	REL	21	rr	3	
BRSET n,opr8a,rel	存储器中位 n 置位时转移	if (Mn) = 1, 跳转	—	—	—	—	↑ ↓	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5	
BSET n,opr8a	设置存储单元的位 n	$M_n \leftarrow 1$	—	—	—	—	—	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5	
BSR rel	子程序调用	PC ← (PC)+2 将 PC 压入影子 PC	—	—	—	—	—	REL	AD	rr	5	

		PC←(PC)+rel											
CBEQ opr8a,rel	相等比较跳转	if (A) = (M), 跳转 if (A) = (M), 跳转 if (X) = (M), 跳转 if (A) = (M), 跳转 if (A) = (M), 跳转 if (A) = (M), 跳转	—	—	—	—	—	—	—	DIR	31	dd rr	5
CBEQA#opr8i,rel										IMM	41	ii rr	4
CBEQX#opr8i,rel										IMM	51	ii rr	4
CBEQ opr8a,X+,rel										IX1+	61	ff rr	5
CBEQ ,X+,rel										IX+	71	rr	5
CBEQ oprx8,SP,rel										SP1	9E61	ff rr	6
CLC	清除进位位	C←0	—	—	—	—	—	0	INH	98		1	
CLI	清除中断屏蔽标志位	I←0	—	—	0	—	—	0	INH	9A		1	
CLR opr8a	清零	M←0x00	0	—	—	0	1	—	DIR	3F	dd	5	
CLRA		INH							4F		1		
CLR X		INH							5F		1		
CLR H		INH							8C		1		
CLR oprx8, X		IX1							6F	ff	5		
CLR ,X		IX							7F		4		
CLR oprx8,SP		SP1							9E6F	ff	6		
CMP #opr8i	累加器与存储器比较	(A)-(M) (更新 CCR, 但是不改变操作)	↑	—	—	↑	↑	↑	IMM	A1	ii	2	
CMP opr8a									DIR	B1	dd	3	
CMP opr16a									EXT	C1	hh ll	4	
CMP oprx16,X									IX2	D1	ee ff	4	
CMP oprx8,X									IX1	E1	ff	3	
CMP ,X									IX	F1		3	
CMP oprx16,SP									SP2	9ED1	ee ff	5	
CMP oprx8,SP									SP1	9EE1	ff	4	
COM opr8a	取反	M←(M) = 0xFF - (M)	0	—	—	↑	↑	1	DIR	33	dd	5	
COMA		INH							43		1		
COM X		INH							53		1		
COM oprx8,X		IX1							63	ff	5		
CIM ,X		IX							73		4		
COM oprx8,SP		SP1							9E63	ff	6		
CPHX opr16a	比较变址寄存器 (H:X) 和存储器	(H:X) - (M:M + 0x0001) (更新 CCR, 但操作不改变)	↑	—	—	↑	↑	↑	EXT	3E	hh ll	6	
CPHX #opr16i									IMM	65	jj kk	3	
CPHX opr8a									DIR	75	dd	5	
CPHX opr8a,SP									SP1	9EF3	ff	6	
CPX #opr8i	比较 X (变址寄存器低位) 和存储器	(X) - (M) (更新 CCR, 但操作不改变)	↑	—	—	↑	↑	↑	IMM	A3	ii	2	
CPX opr8a									DIR	B3	dd	3	
CPX opr16a									EXT	C3	hh ll	4	
CPX opr16,X									IX2	D3	ee ff	4	
CPX opr8,X									IX1	E3	ff	3	
CPX ,X									IX	F3		3	
CPX opr16,SP									SP2	9ED3	ee ff	5	
CPX opr8,SP	SP1	9EE3	ff	4									
DAA	对累加器进行十进制调整	(A)10	U	—	—				INH	72		1	
DBNZ opr8a,rel	减 1 不为 0 跳转	A/X/M-1 结果不为 0 跳转 影响 X 不影响 H	—	—	—	—	—	—	DIR	3B	dd rr	7	
DBNZ A,rel									INH	4B	rr	4	
DBNZ B,rel									INH	5	rr	4	
DBNZ opr8,X,rel									IX1	6	ff rr	7	
DBNZ ,X,rel									IX	7	rr	6	
DBNZ opr8,SP,rel									SP1	9E6B	ff rr	8	
DEC opr8a	减法	M←(M) - 0x01	↑	—	—	↑	↑	—	DIR	3A	dd	5	
DECA		INH							4A		1		
DECX		INH							5A		1		

DEC oprx8,X DEC ,X DEC oprx8,SP		$M \leftarrow (M) - 0x01$ $M \leftarrow (M) - 0x01$ $M \leftarrow (M) - 0x01$								IX1 IX SP1	6A 7A 9E6A	ff ff ff	5 4 6
DIV	除法	$A \leftarrow (H:A) \div (X)$ $H \leftarrow \text{Remainder}$	—	—	—	—	↑	↑	↓	INH	52		6
EOR #opr8i EOR opr8a EOR opr16a EOR oprx16,X EOR oprx8,X EOR ,X EOR oprx16,SP EOR oprx8,SP		$A \leftarrow (A \oplus M)$	0	—	—	↑	↑	↓	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 B8 C8 D8 E8 F8 9ED8 9EE8	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
INC opr8a INCA INCX INC opr8,X INC ,X INC oprx8,SP	加法	$M \leftarrow (M) + 0x01$ $A \leftarrow (A) + 0x01$ $X \leftarrow (X) + 0x01$ $M \leftarrow (M) + 0x01$ $M \leftarrow (M) + 0x01$ $M \leftarrow (M) + 0x01$	↑	—	—	↑	↑	↓	—	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff ff	5 1 1 5 4 6
JMP opr8a JMP opr16a JMP opr16,X JMP opr8,X JMP ,X	跳转	$PC \leftarrow \text{转移地址}$	—	—	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	3 4 4 3 3
JSR opr8a JSR opr16a JSR opr16,X JSR opr8,X JSR ,X	绝对转移	$PC \leftarrow (PC) + n$ ($n = 1, 2, \text{ or } 3$) Push (PCL); $SP \leftarrow (SP) - 0x0001$ Push (PCH); $SP \leftarrow (SP) - 0x0001$ $PC \leftarrow \text{绝对地址}$	—	—	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 6 5 5
LDA #opr8i LDA opr8a LDA opr16a LDA opr16,X LDA oprx8,X LDA ,X LDA opr16,SP LDA opr8,SP	从存储器取 数到累加器	$A \leftarrow (M)$	0	—	—	↑	↑	↓	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 B6 C6 D6 E6 F6 9ED6 9EE6	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
LDHX #opr16i LDHX opr8a LDHX opr16a LDHX ,X LDHX opr16,X LDHX opr8,X LDHX opr8,SP	从存储器取 数到变址寄 存器 (H:X)	$H:X \leftarrow (M:M + 0x0001)$	0	—	—	↑	↑	↓	—	IMM DIR EXT IX IX2 IX1 SP1	45 55 32 9EAE 9EBE 9ECE 9EFE	ij kk dd hh ll ee ff ff ff	3 4 5 5 6 5 5
LDX #opr8i LDX opr8a LDX opr16a LDX opr16,X LDX oprx8,X LDX ,X LDX oprx16,SP LDX oprx8,SP	从存储器取 数到 X (变址 寄存器低位)	$X \leftarrow (M)$	0	—	—	↑	↑	↓	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEE	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
LSL opr8a LSLA LSLX LSL oprx8,X LSL ,X	逻辑左移		↑	—	—	↑	↑	↓	↓	DIR INH INH IX1 IX	38 48 58 68 78	dd ff ff	5 1 1 5 4

LSL oprx8,SP									SP1	9E68	ff	6
LSR opr8a									DIR	34	dd	5
LSRA									INH	44		1
LSRX									INH	54		1
LSR oprx8,X	逻辑右移		↑	—	—	0	↑	↑	INH	64	ff	5
LSR ,X			↓				↓	↓	IX1	74		4
LSR oprx8,SP									SP1	9E64	ff	6
MOV opr8a,opr8a									DIR/DIR	4E	dd dd	5
MOV opr8a	移动	(M)destination ← (M)source	0	—	—	↑	↑	—	DIR/IX+	5E	dd	5
MOV#opr8i,opr8a		H:X ← (H:X) + 0x0001				↓	↓		IMM/DIR	6E	ii dd	4
MOV ,X+,opr8a		在 IX+/DIR 和 DIR/IX+ 模式中							IX+/DIR	7E	dd	5
MUL	不带符号的乘法	X:A ← (X) × (A)	—	0	—	—	—	0	INH	42		5
NEG opr8a		M ← -(M) = 0x00 - (M)							DIR	30	dd	5
NEGA		A ← -(A) = 0x00 - (A)							INH	40		1
NEGX	取反 (二进制补码)	X ← -(X) = 0x00 - (X)				↑	↑	↑	INH	50		1
NEG oprx8,X		M ← -(M) = 0x00 - (M)				↓	↓	↓	IX1	60	ff	5
NEG ,X		M ← -(M) = 0x00 - (M)							IX	70		4
NEG oprx8,SP		M ← -(M) = 0x00 - (M)							SP1	9E60	ff	6
NOP	无操作	一个总线周期	—	—	—	—	—	—	INH	9D		1
NSA	累加器中的高低4位交换	A ← (A[3:0]:A[7:4])	—	—	—	—	—	—	INH	62		1
ORA #opr8i									IMM	AA	ii	2
ORA opr8a									DIR	BA	dd	3
ORA opr16a									EXT	CA	hh ll	4
ORA oprx16,X	累加器和存储器进行或运算	A ← (A) (M)	0	—	—	↑	↑	—	IX2	DA	ee ff	4
ORA oprx8,X						↓	↓		IX1	EA	ff	3
ORA ,X									IX	FA		3
ORA oprx16,SP									SP2	9EDA	ee ff	5
ORA oprx8,SP									SP1	9EEA	ff	4
PSHA	A 进栈	Push (A); SP ← (SP) - 0x0001	—	—	—	—	—	—	INH	87		2
PSHH	H 进栈	Push (H); SP ← (SP) - 0x0001	—	—	—	—	—	—	INH	8B		2
PSHX	X 进栈	Push (H); SP ← (SP) - 0x0001	—	—	—	—	—	—	INH	89		2
PULA	A 出栈	SP ← (SP + 0x0001); Pull (A)	—	—	—	—	—	—	INH	86		3
PULH	H 出栈	SP ← (SP + 0x0001); Pull (H)	—	—	—	—	—	—	INH	8A		3
PULX	X 出栈	SP ← (SP + 0x0001); Pull (X)	—	—	—	—	—	—	INH	88		3
ROL opr8a									DIR	39	dd	5
ROLA									INH	49		1
ROLX	循环左移通过进位		↑	—	—	↑	↑	↑	INH	59		1
ROL oprx8,X			↓			↓	↓	↓	IX1	69	ff	5
ROL ,X									IX	79		4
ROL oprx8,SP									SP1	9E69	ff	6
ROR opr8a									DIR	36	dd	5
RORA									INH	46		1
RORX	循环右移通过进位		↑	—	—	↑	↑	↑	INH	56		1
ROR oprx8,X			↓			↓	↓	↓	IX1	66	ff	5
ROR ,X									IX	76		4
ROR oprx8,SP									SP1	9E66	ff	6
RSP	复位堆栈	SP ← 0xFF (不影响高字节)	—	—	—	—	—	—	INH	9C		1
RTI	从中断返回	SP ← (SP) + 0x0001; Pull (CCR) SP ← (SP) + 0x0001; Pull (A) SP ← (SP) + 0x0001; Pull (X) SP ← (SP) + 0x0001; Pull (PCH) SP ← (SP) + 0x0001; Pull (PCL)	↑	↑	↑	↑	↑	↑	INH	80		9
RTS	从绝对返回	SP ← SP + 0x0001; Pull (PCH) SP ← SP + 0x0001; Pull (PCL)	—	—	—	—	—	—	INH	81		6

SBC #opr8i SBC opr8a SBC opr16a SBC oprx16,X SBC oprx8,X SBC ,X SBC oprx16,SP SBC oprx8,SP	带进位减法	$A \leftarrow (A) - (M) - (C)$	↑ ↓	—	—	↑ ↓	↑ ↓	↑ ↓	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 B2 C2 D2 E2 F2 9ED2 9EE2	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
SEC	置进位位	$C \leftarrow 1$	—	—	—	—	—	1	INH	99		1
SEI	置中断屏蔽位	$I \leftarrow 1$	—	—	1	—	—	—	INH	9B		1
STA opr8a STA opr16a STA opr16,X STA opr8,X STA ,X STA opr16,SP STA opr8,SP	存存储器数到累加器	$M \leftarrow (A)$	0	—	—	↑ ↓	↑ ↓	—	DIR EXT IX2 IX1 IX SP2 SP1	B7 C7 D7 E7 F7 9ED7 9EE7	dd hh ll ee ff ff ff ee ff ff	3 4 4 3 2 5 4
STHX opr8a STHX opr16a STHX opr8,SP	存数到 H:X	$(M:M + 0x0001) \leftarrow (H:X)$	0	—	—	↑ ↓	↑ ↓	—	DIR EXT SP1	35 96 9EFF	dd hh ll ff	4 5 5
STOP	使能中断: 停止进程 参考 MCU 程序说明书	$I \text{ bit} \leftarrow 0; \text{ Stop Processing}$	—	—	0	—	—	—	INH	8E		2+
STX opr8a STX opr16a STX opr16,X STX opr8,X STX ,X STX opr16,SP STX opr8,SP	存到存储器中 X (变址寄存器的低 8 位)	$M \leftarrow (X)$	0	—	—	↑ ↓	↑ ↓	—	DIR EXT IX2 IX1 IX SP2 SP1	BF CF DF EF FF 9EDF 9EEF	dd hh ll ee ff ff ff ee ff ff	3 4 4 3 2 5 4
SUB #opr8i SUB opr8a SUB opr16a SUB opr16,X SUB oprx8,X SUB ,X SUB oprx16,SP SUB oprx8,SP	减法	$A \leftarrow (A) - (M)$	↑ ↓	—	—	↑ ↓	↑ ↓	↑ ↓	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 B0 C0 D0 E0 F0 9ED0 9EE0	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
SWI	软件中断	$PC \leftarrow (PC) + 0x0001$ Push (PCL); $SP \leftarrow (SP) - 0x0001$ Push (PCH); $SP \leftarrow (SP) - 0x0001$ Push (X); $SP \leftarrow (SP) - 0x0001$ Push (A); $SP \leftarrow (SP) - 0x0001$ Push (CCR); $SP \leftarrow (SP) - 0x0001$ $I \leftarrow 1;$ $PCH \leftarrow \text{Interrupt Vector High Byte}$ $PCL \leftarrow \text{Interrupt Vector Low Byte}$	—	—	1	—	—	—	INH	83		11
TAP	传递 A 到 CCR	$CCR \leftarrow (A)$	↑ ↓	↑ ↓	↑ ↓	↑ ↓	↑ ↓	↑ ↓	INH	84		1
TAX	传递 A 到 X (变址寄存器低位)	$X \leftarrow (A)$	—	—	—	—	—	—	INH	97		1
TPA	传递 CCR 到累加器 A	$A \leftarrow (CCR)$	—	—	—	—	—	—	INH	85		1
TST opr8a	测试是负数	$(M) - 0x00$	0	—	—	↑ ↓	↑ ↓	—	DIR	3D	dd	4

TSTA	或零	(A) - 0x00				↓	↓		INH	4D		1
TSTX		(X) - 0x00							INH	5D		1
TST oprx8,X		(M) - 0x00							IX1	6D	ff	4
TST ,X		(M) - 0x00							IX	7D		3
TST oprx8,SP		(M) - 0x00							SP1	9E6D	ff	5
TSX	传递 SP 到变址寄存器	$H:X \leftarrow (SP) + 0x0001$	-	-	-	-	-	-	INH	95		2
TXA	传递 X (变址寄存器低位) 到累加器 A	$A \leftarrow (X)$	-	-	-	-	-	-	INH	9F		1
TXS	传递变址寄存器到 SP	$SP \leftarrow (H:X) - 0x0001$	-	-	-	-	-	-	INH	94		2
WAIT	使能中断; 等待中断	$I \text{ bit} \leftarrow 0; \text{Halt CPU}$	-	-	0	-	-	-	INH	8F		2+

第八章 内部时钟发生器 (S08ICGV4)

内部时钟发生器(S08ICGV4)模块是用来为 MC9S08AW60 系列 MCU 生成系统时钟。模拟供电 V_{DD} 和 V_{SS} 完全来源于 MCU 的 V_{DD} 和 V_{SS} 引脚。在附录 A“电气特性和时间的规格”中可能有 ICG 电气参数数据。

注意: Freescale 半导体公司建议 FLASH 地址 \$FFBE 被用来存储一个易失性的版本。这将使调试器和编程器供应商去执行一个操作和存储 ICGTRM 的数值, 由此产生的值在稍后的时间为用户访问。

图 8-1 结构图突出 ICG 模块

注意:

1. 如果是输入端口, 端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用(IRQEDG=1), 引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测, 下拉式启用(IRQPE = 1)。
3. IRQ 没有一个二极管到 V_{DD} , IRQ 不应该超过 V_{DD} 。
4. 引脚包含完整的上拉装置。
5. 引脚 PTD7, PTD3, PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用(KBIPE_n = 1)并选定上升沿(KBEDG_n = 1)时, 下拉式启用时。

8.1 简介

ICG 提供了多种时钟源选择。这提供了用户极大的灵活性, 当在成本、精度、目前制定、以及绩效之间进行选择时。如图 8-3 所示, ICG 由四个功能模块组成。下面对每个模块进行简要的介绍, 更详细介绍见后一章。

- 振荡器模块——振荡器模块提供了连接外部晶体或谐振器的方法。两个频率范围是软件选择最优的启动与稳定。另外, 振荡器模块可以用来提供系统时钟的外部方波。外部源可以提供一个非常精确的时钟源。振荡器是能够被配置为低功耗模式或高振幅模式, 作为选定的 HGO。
- 内部参考发生器——内部参考发生器由两个控制时钟源组成。其中一个约 8 MHz, 作为当地时钟为背景调试控制器。另一个内部参考时钟源是 243KHz, 可以通过软件精简为更精细的准确性, 当一个确切的时间事件是输入到 MCU。这提供了一个高度可靠, 低成本的时钟源。
- 频率锁相环——频率锁相环(FLL)阶段, 用内部或外部时钟源乘以它得到更高的频率。该电路已取得锁存时, 状态位提供消息, 并适逢出锁。此外, 这个模块可以监控外部参考时钟和是否时钟是有效的信号。
- 时钟选择模块——时钟选择模块提供了几个开关选项为了连接不同的时钟源到系统时钟树。ICGDCLK 是 FLL 之外时钟频率的倍数, ICGDCLK 是参考时钟频率来于晶体或外部时钟源, FFE(固定频率使能)是一种控制信号用于控制系统的固定频率时钟(XCLK)。ICGDCLK 是时钟源为背景调试控制器(BDC)。

8.1.1 特征

这个模块使得用户与许多自动功能非常的友好, 在没有用户干预的情况下。快速配置模

块，请参考第 8.5 章节，“初始化/应用信息”，并挑选一个最适合的应用需求例子。

ICG 和时钟分布系统的特征

- 为主要时钟源允许的一个成本、频率和精度的范围选择。
 - 32kHz~100 kHz 石英晶体或谐振器
 - 1MHz~16 MHz 石英晶体或谐振器
 - 外部时钟
 - 内部参考发生器
- 默认为自我时钟模块，以尽量减少启动延误。
- 锁频环 (FLL)产生 8 MHz 到 40MHz(总线频率达 20 MHz)
外部或内部时钟作为参考频率
- 非运行时钟源的自动锁定
- 复位或中断对时钟或锁频环的损失
- 数字控制振荡器(DCO)保留了先前的频率设置，在恢复 stop3 模式时，进行快速频率锁存。
- DCO 将在丧失或取消参考时钟时保持运行频率
- 在频率锁频之后选择 1/8 分频
- 单独的自我时钟源为实时中断
- 内部时钟源支持 SCI 通信，没有额外的元件
- 锁存后获得自动频率锁存器
- 外部振荡器可供选择低能量或高增益

8.1.2 操作模块

这仅是一个高层描述。操作模块的详细介绍见第 8.4 章节，“功能描述”。

- 模块 1——关
输出时钟 ICGOUT 是静态的。在执行停止指令时进入这个模块。
- 模块 2——自我时钟频率器(SCM)
在复位后立即进入默认操作模块。IGC 的锁频环是开环的，数字控制振荡器(DCO)是在一个通过过滤位设置的频率下进行空操作。
- 模块 3——从事内部的 FLL (FEI)
这个模块中的 IGC 的频率锁存器用来创建可编程的频率，是内部参考频率的倍数。
 - 内部没锁的 FLL 是个过度状态，发生在 FLL 试图锁存的时候。FLL DCO 的频率远离目标，FLL 就调整 DCO 来匹配目标频率。
 - 内部锁存的 FLL 是个状态，发生在 FLL 发现 DCO 被锁定到内部参考的倍数。
- 模块 4——绕过外面的 FLL(FBE)
这个模块中 ICG 设置成避开 FLL，并用一个外部时钟作为时钟源。
- 模块 5——从事外部的 FLL(FEE)
ICG 的 FLL 用来产生可编程的频率，是内部时钟参考的倍数。
 - 内部没锁的 FLL 是个过度状态，发生在 FLL 试图锁存的时候。FLL DCO 的频率远离目标，FLL 就调整 DCO 来匹配目标频率。
 - 内部锁存的 FLL 是个状态，发生在 FLL 发现 DCO 被锁定到内部参考的倍数。

8.1.3 结构图

如图 8-3 所示，高层简图给出了内部时钟发生器(ICG)模块的实用组织。这章节给出了总体的介绍和特征目录。

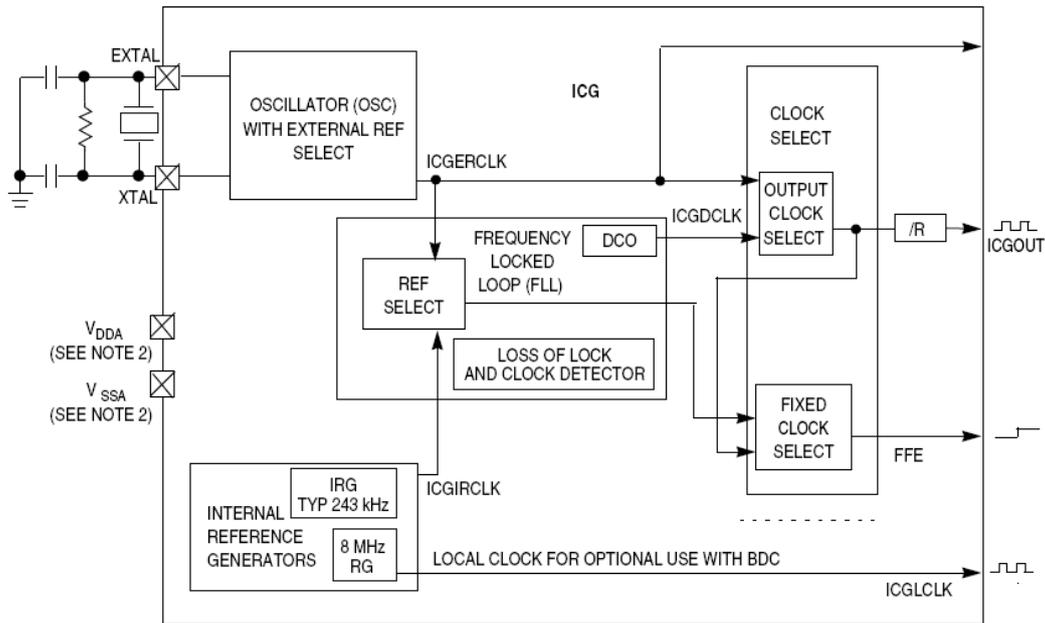


图 8-3 ICG 结构图

8.2 外部信号介绍

振荡器引脚用来给 MCU 提供一个外部时钟。振荡器引脚在低能量模式(默认)被控制。在低能量模式中振荡器的振幅被限制在大约 1V，高峰到高峰。

8.2.1 EXTAL——外部参考时钟/振荡输入

当首先写入 ICGC1，选择 FEE 模块或 FBE 模块时，这个引脚功能作为外部时钟输入或视 REFS 而定的振荡电路的输入。当写入 ICGC1，选择 FEI 模块或 SCM 模块时，这个引脚不被 ICG 所使用。

8.2.2 XTAL——振荡输出

当首先写入 ICGC1，选择 FEE 模块或 FBE 模块时，这个引脚功作为振荡电路输出。当首先写入 ICGC1，选择 FEI 模块或 SCM 模块时，这个引脚不被 ICG 所使用。振荡器能够被设置为提供更高振幅输出为改善抗干扰度。这种操作模式是选定 HGO=1。

8.2.3 外部时钟连接

若使用外部时钟，引脚如图 8-4 所示。

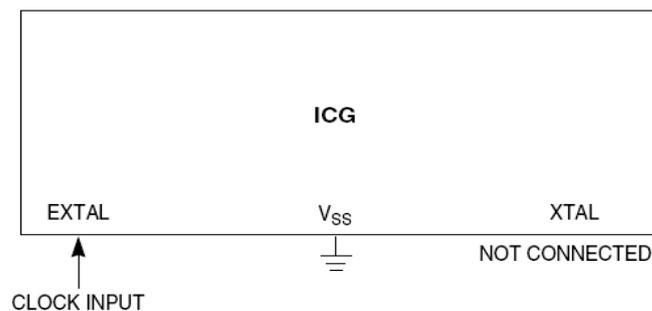


图 8-4 外部时钟连接

8.2.4 外部晶体/共振器连接

若使用外部晶体/共振器频率参数，引脚如图 8-5 所示。电气特性章节

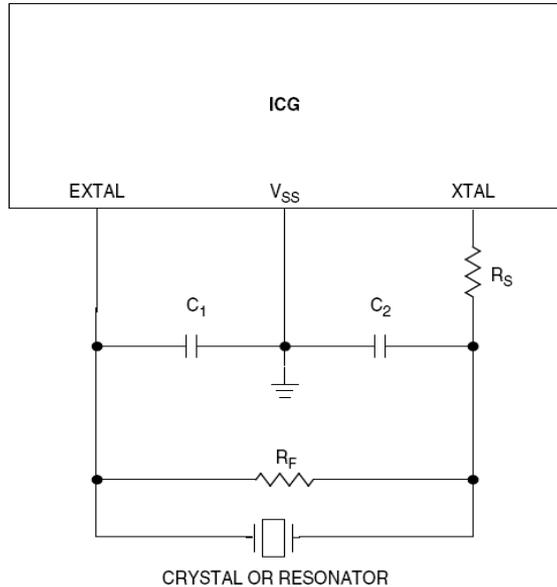


图 8-5 外部频率参数连接

8.3 寄存器定义

有关直接页寄存器概述在这个记录表的存储器章节，为绝对地址分配为所有 ICG 寄存器。本章节通过名字提到寄存器和控制位。飞思卡尔提供等同或头文件用来翻译这些名字到适当的绝对地址。

8.3.1 ICG 控制寄存器 1(ICGC1)

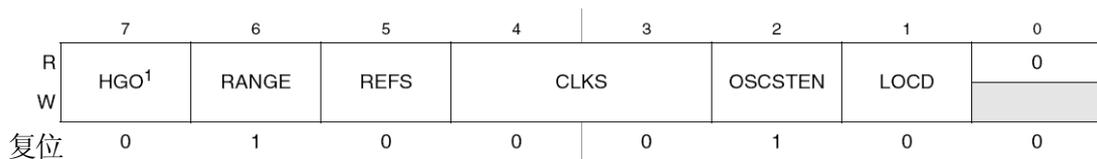


图 8-6 ICG 控制寄存器 1(ICGC1)

¹ 这个位在复位后仅能写入一次，另外的写入被忽略。

表 8-1 ICGC1 寄存器字段描述

字段	描述
7 HGO	高增益振荡选择 ——HGO 位用来选择低能量和高增益操作为改善抗干扰性。这个位只能在复位后写入一次。 0 振荡器设置为低能量操作 1 振荡器设置为高增益操作
6 RANGE	频率范围选择 ——RANGE 位控制振荡器、参考分频器和 FLL 环路分频器乘法因子(P)。它选择两个参考频率范围中的一个为 ICG。RANGE 位在复位后仅能

	<p>写入一次。RANGE 位仅影响从事外部的 FLL 或绕过外部的 FLL 模块。</p> <p>0 振荡器设置为低频率范围。FLL 精确因子 P 为 64。</p> <p>1 振荡器设置为高频率范围。FLL 精确因子 P 为 1。</p>
5 REFS	<p>外部参考选择——REFS 位控制外部参考时钟源为 ICGERCLK。REFS 在复位后仅能写入一次。</p> <p>0 外部时钟请求</p> <p>1 振荡器用晶体或共振器请求</p>
4: 3 CLKS	<p>时钟模块选择——CLKS 位控制时钟模块。若绕过外部的 FLL 被请求，它直到 ERCS=1 才被选择。若 ICG 进入关模块，CLKS 位将继续不变。若早前的写入不完整，写入 CLKS 位不影响。</p> <p>00 Self-clocked</p> <p>01 FLL engaged, internal reference</p> <p>10 FLL bypassed, external reference</p> <p>11 FLL engaged, external reference</p> <p>除了在复位后第一次写入 CLKS=0X，CLKS 位可在任何时候写入。CLKS 位不能写到 1X 直到在下次复位后(因为没有预留 EXTAL 引脚)。</p>
2 OSCSTEN	<p>在关模式下使能振荡器——OSCSTEN 位控制控制振荡电路继续使能，当 ICG 进入关模式。若 HGO=1、RANGE=1，该位没有影响。</p> <p>0 振荡器失去当 ICG 进入关模式，除非使能端为高，CLKS = 10, and REFST = 1。</p> <p>1 振荡器使能当 ICG 进入关模式，CLKS = 1X and REFST = 1。</p>
1 LOCD	<p>时钟的损失禁用</p> <p>0 时钟的损失检测启动</p> <p>1 时钟的损失检测禁用</p>

8.3.2 ICG 控制寄存器 2(ICGC2)

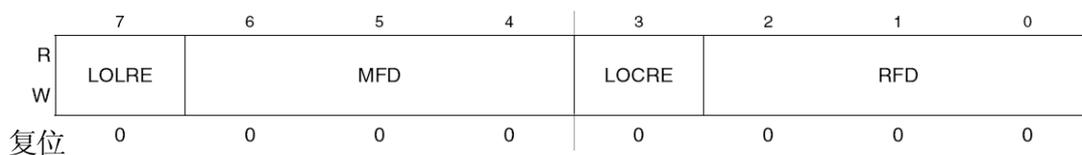


图 8-7 ICG 控制寄存器 2(ICGC2)

表 8-2 ICGC2 寄存器字段描述

字段	描述
7 LOLRE	<p>锁存丢失复位使能——LOLRE 位决定由 ICG 制作的请求的类型，跟着一个锁存丢失指示。LOLRE 位仅有一个影响是当设置 LOLS 时。</p> <p>0 在锁存丢失发出一个中断请求</p>

	1 在锁存丢失发出一个复位请求
6:4 MFD	<p>倍增因数——MFD 位控制可编程的倍增因数在 FLL 回路。用 MFD 位表示的数值设置倍增因数(N)用于参考频率。若先前的写入不完整不影响 MFD 位的写入。选择一个足够小的数值 N，$f_{ICGDCLK}$ 不超过给定值的最大值。</p> <p>000 倍增因数=4 001 倍增因数=6 010 倍增因数=8 011 倍增因数=10 100 倍增因数=12 101 倍增因数=14 110 倍增因数=16 111 倍增因数=18</p>
3 LOCRES	<p>时钟丢失复位使能——LOCRES 位决定了该系统如何管理一个时钟丢失条件。</p> <p>0 在时钟丢失发出一个中断请求 1 在时钟丢失发出一个复位请求</p>
2:0 RFD	<p>简化的频率分频器——RFD 位控制分频器的数值，随着时钟选择电路。用 RFD 位表示数值设置分频因子(R)用于时钟源输出。若先前的写入不完整不影响 RFD 位的写入。</p> <p>000 分频因子=1 001 分频因子=2 010 分频因子=4 011 分频因子=8 100 分频因子=16 101 分频因子=32 110 分频因子=64 111 分频因子=128</p>

8.3.3 ICG 状态寄存器 1(ICGS1)

	7	6	5	4	3	2	1	0
R	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
W								1
复位	0	0	0	0	0	0	0	0

图 8-8 ICG 状态寄存器 1(ICGS1)

表 8-3 ICGS1 寄存器字段描述

字段	描述
7: 6 CLKST	<p>时钟模块状态——CLKST 位表明了当前的时钟模块。在写入 CLKST 位之后，CLKST 位不立即更新。由于在时钟域内部同步。</p> <p>00 Self-clocked 01 FLL engaged, internal reference 10 FLL bypassed, external reference 11 FLL engaged, external reference</p>
5 REFST	<p>参考时钟状态——REFST 位指名哪个时钟参数是被当前参选择考电路选择的。</p> <p>0 选择外部时钟 1 选择晶振/共振器</p>
4 LOLS	<p>FLL 锁存失去状态——LOLS 位是表明 FLL 锁存状态的。</p> <p>0 FLL 没有意外地丢失锁存，自 LOLS 最后清理。 1 FLL 有意外地丢失锁存，自 LOLS 最后清理，LOLRE 决定执行。</p>
3 LOCK	<p>FLL 锁存状态——LOCK 位表明 FLL 是否锁存。LOCK 位是清除、自我时钟频率和 FLL 绕过模式。</p> <p>0 FLL 目前没锁存 1 FLL 目前锁定</p>
2 LOCS	<p>时钟失去状态——LOCS 位表明 ICG 时钟丢失锁存状态。</p> <p>0 ICG 没有丢失锁存，自 LOCS 最后清理。 1 ICG 有意外地丢失锁存，自 LOLS 最后清理，LOLRE 决定执行。</p>
1 ERCS	<p>外部参考时钟状态——ERCS 位指明外部参考时钟(ICGERCLK)满足最小频率需求。</p> <p>0 外部参考时钟不合适，频率要求不满足。 1 外部参考时钟合适，频率要求满足。</p>
0 ICGIF	<p>ICG 中断标志——ICGIF 读/写标志被设置，当 ICG 中断请求时。它通过复位或读 ICG 状态寄存器被清除，在 ICGIF 被设置，写入逻辑 1 到 ICGIF 时。若另一个 ICG 中断在清除时序完成之前产生，时序复位，所以 ICGIF 将仍然在清除时序完成之后为早期的中断置位。写入逻辑 0 到 ICGIF 没有影响。</p> <p>0 没有中断请求正在等待 1 一个 ICG 中断请求正在等待</p>

8.3.4 ICG 状态寄存器 2(ICGS2)

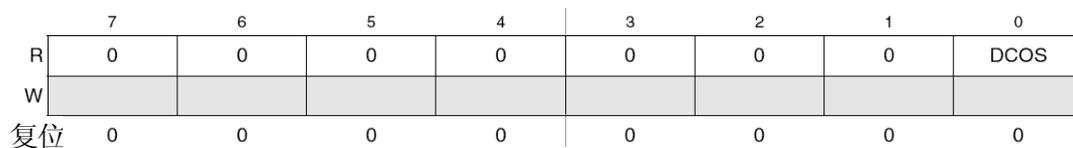


图 8-9 ICG 状态寄存器 2(ICGS2)

表 8-4 ICGS2 寄存器字段描述

字段	描述
0 DCOS	<p>DCO 时钟稳定——DCOS 位在 DCO 时钟(ICG2DCLK)稳定时被设置。计数误差并没有改变, 通过大于 n_{unlock} 为 2 个连续的样本, DCO 时钟不是静态的。该位是用于在退出关状态时, 是否 $CLKS=X1$, 以确定何时切换到所要求的时钟模块。它还可被用来在自我的时钟频率的模式, 以确定何时开始监测 DCO 时钟。该位是进入关状态后清除。</p> <p>0 DCO 时钟不稳定</p> <p>1 DCO 时钟稳定</p>

8.3.5 ICG 过滤寄存器(ICGFLTU, ICGFTL)

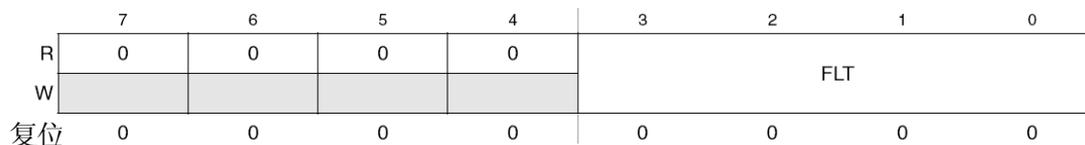


图 8-10 ICG 高过滤器的寄存器(ICGFLTU)

表 8-5 ICGFLTU 寄存器字段描述

字段	描述
3: 0 FLT	<p>过滤器的数值——FLT 位指明了当前过滤器的数值, 控制 DCO 频率。FLT 位仅能读除了当 $CLKS$ 位安排到自我时钟模块($CLKS=00$)。在自我时钟模块中, 任何写入到 ICGFLTU 更新当前的 12 位过滤器的价值。写入 ICGFLTU 寄存器将不会影响 FLT, 若先前锁存序列是不完整的。</p>

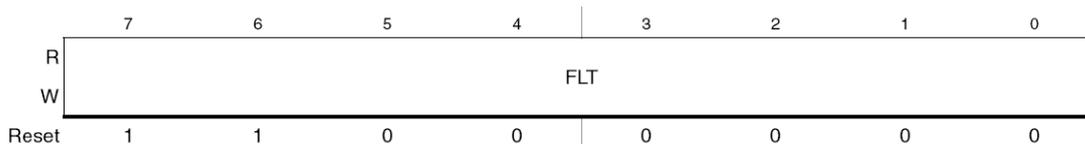


图 8-11 ICG 低过滤器的寄存器(ICGFTL)

表 8-6 ICGFTL 寄存器字段描述

字段	字段

7: 0 FLT	过滤器数值 ——FLT 位指明了当前过滤器的数值，控制 DCO 频率。FLT 位仅能读除了当 CLKS 位安排到自我时钟模块(CLKS=00)。在自我时钟模块中，任何写入到 ICGFLTU 更新当前的 12 位过滤器的价值。写入 ICGFLTU 寄存器将不会影响 FLT，若先前锁存序列是不完整的。过滤寄存器显示过滤数值(FLT)。
---------------------------	---

8.3.6 ICG 修正寄存器(ICGTRM)

	7	6	5	4	3	2	1	0
R	TRIM							
W	TRIM							
POR	1	0	0	0	0	0	0	0
复位	U	U	U	U	U	U	U	U

图 8-12 ICG 修正寄存器(ICGTRM)

表 8-7 ICGTRM 寄存器字段描述

字段	描述
7 TRIM	ICG 修正设置 ——TRIM 位控制内部参考发生器频率。允许 $\pm 25\%$ 调整周期。影响该位周期的是二进制加权(例如，位 1 比位 0 调整多 2 次)。在 TRIM 中增加二进制值将增加周期，降低的数值将减少周期。

8.4 实用的描述

本章节提供了 ICG 的五个操作模块的每个功能描述。也讨论了时钟的损失、锁存的损失和进入每个模块的要求。ICG 非常灵活，在某些配置中可能超过某些时钟规格。在使用 FLL 时，配置 ICG 以至于 ICGDCLK 的频率不超过其最大值以确保适合 MCU 运作。

8.4.1 关模块 (Off)

通常当 CPU 进入停止模式时，ICG 将停止一切活动并在关状态。不过当时钟继续运作，而 CPU 在停止模式时要考虑两种情况。

8.4.1.1 BDM 使能

激活 BDM，ICG 像原先的程序继续活动。允许通过 BDC 控制器访问存储器和控制寄存器。

8.4.1.2 OSCSTEN 位设置

在停止模式下启用振荡器(OSCSTEN=1)，个别时钟发生器启动但是 MCU 剩余的时钟被关闭了。这个选项用来避免振荡器长时间启动，或在 stop3 模式下进行振荡器 RTI 里运行。

8.4.1.3 停止/关模块恢复

当 CPU 由于中断进入停止模式，先前设置的控制位有效，且系统时钟恢复。若选择 IEEE，

ICG 将源于内部参考直到外部时钟稳定。若选择 FBE, ICG 将在 ICGOUT 之前等外部时钟稳定。

当 CPU 由于复位进入停止模式, 先前设置的 ICG 控制位被忽略, 且应用默认复位值。ICG 将在 SCM 模式中退出停止, 配置大约 8MHz DCO 输出(总线时钟 4MHz)与修正值维持不变。若使用晶体, 在使用 ICGERCLK 之前先检测 4096 时钟表。这成为晶体开始时间的一部分。

8.4.2 自我锁存模块(SCM)

自我时钟模块(SCM)是默认的操作模式, 并在产生下面任一种情况的时候输入:

- 任何复位后。
- 当 CLKS 不等于 10, 从关模式进入。若 CLKS=X1, ICG 进入临时状态直到 DCO 稳定时(DCOS=1)。
- CLKS 位写入 X1 到 00。
- CLKS=1X, 不检测 ICGERCLK(ERCS=0 和 LOCS=1)。

在这种状态, FLL 回路是打开的。打开 DCO, 输出时钟信号 ICGOUT 频率由 fICGDCLK/R 给出。ICGDCLK 频率是 8 MHz 到 40 MHz 由写入到过滤寄存器(ICGFLTH 和 ICGFLTL)的新数值决定。

若由于复位进入到这个模块, fICGDCLK 将默认 fSelf_reset, 通常是 8MH。若这个模块是从 FLL 外部使能进入, fICGDCLK 将维持原先的频率。若这个模块是从 FLL 外部使能(通过设置 CLKS 或由于失去外部参考时钟)进入, fICGDCLK 将维持原先的频率, 但若 FLL 不被锁存时 ICGOUT 将双倍。若这个模块是从关模块进入, fICGDCLK 将等于 ICGDCLK 的频率是在进入关状态之前。若在离开关状态时 CLKS 位设置为 01 到 11, ICG 进入这个模式直到 ICGDCLK 稳定, 由 DCOS 位决定。在认为 ICGDCLK 稳定之后, ICG 自动关闭回路通过选择 CLKS 位到 FLL 使能(内部或外部)。

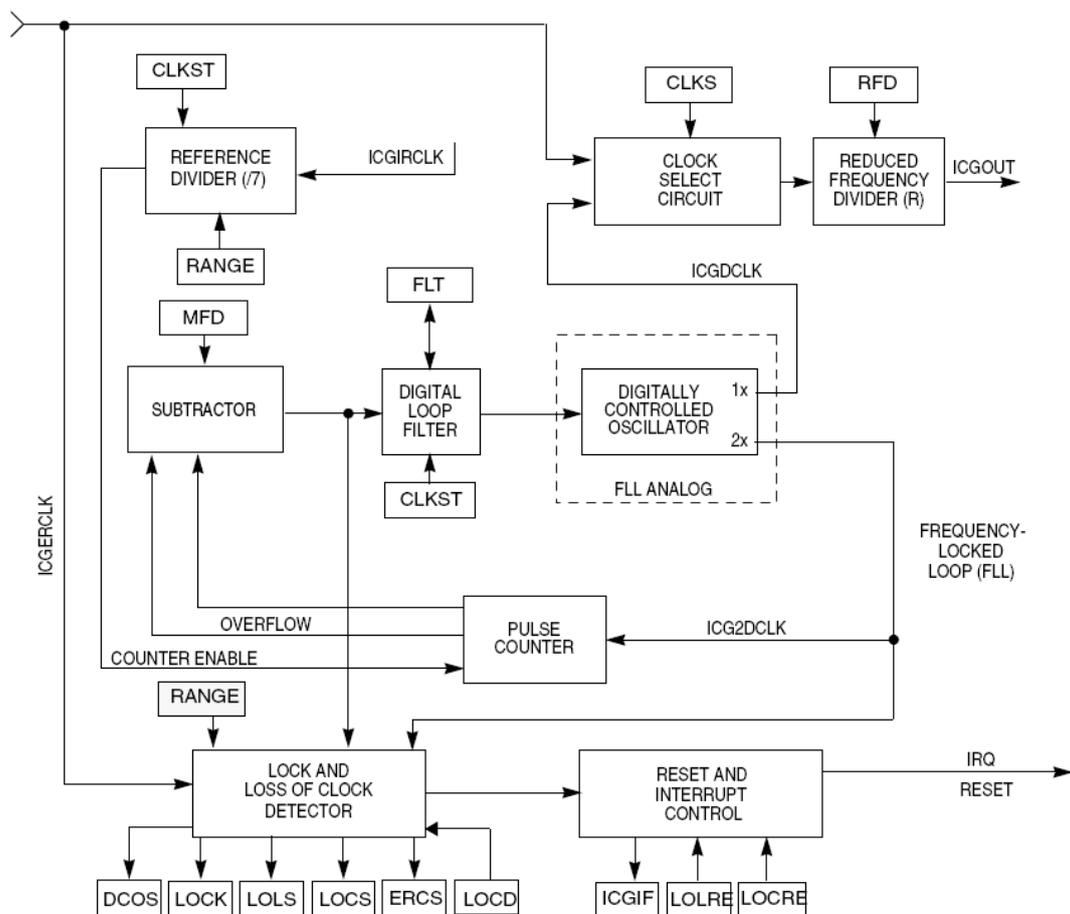


图 8-13 频率锁存板模块

8.4.3 FLL 内部时钟(FEI)模块

当产生下面任一种情况的时候，进入 FLL 内部使能(FEI)：

- CLKS 位写入 01
- DCO 时钟稳定(DCOS=1) 而在 SCM 中正进入关状态 CLKS=01。

FLL 内部使能模块，参考时钟源于内部参考时钟 ICGIRCLK，FLL 回路将试图锁存 ICGDCLK 频率到预期的数值，通过 MFD 位选择。

8.4.4 内部未锁的 FLL

FEI 未锁存是一个临时状态，进入这个状态是在当 FEI 进入时，来于 subtractor 技术误差(Δn)，即大于最大值 $nunlock$ 或小于最小值 $nunlock$ 给出一个数，如要求的那样通过锁存检测器去检测解锁的条件。

ICG 将仍然在这个模块中，当计数误差(Δn)大于最大值 $nlock$ 或小于最小值 $nlock$ ，像所要求的那样通过锁存检测器去检测解锁的条件。

在这种状态中输出时钟信号 ICGOUT 频率由 $fICGDCLK/R$ 给出。

8.4.5 内部锁存的 FLL

从 FEI 未锁到 FLL 内部锁存，是当来于 subtractor 的技术误差(Δn)，即小于 $nlock(max)$ 或大于 $nlock(min)$ 给出一个数，如要求的那样通过锁存检测器去检测解锁的条件。输出时钟信号 ICGOUT 频率由 $fICGDCLK/R$ 给出。在 FEI 被锁存时，过滤数值每四个周期的比较仅

更新一次。

8.4.6 旁路 FLL 外部时钟(FBE)模块

当产生下面任一种情况的时候，进入 FLL 旁路外部 (FBE)：

- 来于 SCM 当 CLKS=10, ERCS 为高时。
- 当 CLKS=10, ERCS=1 进入关模式，然后进入关模式
- 来于 FLL 外部使能模块，若发生 DCO 时钟损失且外部参考仍然有效(LOCS=1 且 ERCS=1)

在这种状态中，关闭 DCO 和 IRG，参考时钟源于外部参考时钟 ICGERCLK。输出时钟信号 ICGOUT 频率由 fICGDCLK/R 给出。若使用外部时钟源(REFS=0)，在 EXTAL 引脚输出频率的范围是 0MHz 到 40MHz。若使用晶体或共振器，频率是低 RANGE=0 或高 RANGE=1。

8.4.7 FLL 外部时钟使能(FEE)模式

当产生下面任一种情况的时候，进入 FLL 旁路外部 (FBE)：

- CLKS=11, ERCS 和 DCOS 都为高。
- DCO 稳定(DCOS=1)同时在 SCM 中进入关状态 CLKS=11。

在 FEE 模块中，参考时钟源于外部参考时钟 ICGERCLK，FLL 回路将试图锁存 ICGDCCLK 频率到预期值，通过 MFD 位选择。在 FEE 模式中运行，必须有一个工作在 32kHz~100kHz 或 2MHz ~10MHz 的外部时钟源。最大外部时钟频率在 FEE 模块中限制在 10 MHz，以防止超过时钟 DCO。如表 8-12 所示，最小乘数是 4。因为 $4 \times 10\text{MHz}$ 是 40MHz，这是 DCO 所限制的，参考时钟频率不能大于 10MHz。

8.4.7.1 FLL 外部未锁使能

FEE 锁存是从 FEE 未锁中进入，是当来于 subtractor 的技术误差(Δn)，即小于 nlock(max) 或大于 nlock(min)给出一个数，像所要求的那样通过锁存检测器去检测解锁的条件。

ICG 将仍然在这种状态中，当计数误差(Δn)大于最大值 nlock 或小于最小值 nlock，如所要求的那样通过锁存检测器去检测解锁的条件。

将在这章的下面介绍在这种状态中，脉冲计数器、电子减法器、数字滤波器、源于闭环的 DCO 和试图根据操作来锁存的描述。进入状态直到 FLL 被锁存，输出时钟信号 ICGOUT 频率由 $fICGDCLK/(2 \times R)$ 给出。除以 2 是防止频率过大

8.4.7.2 FLL 外部锁存使能

FEE 锁存是从 FEE 未锁中进入，是当来于 subtractor 的技术误差(Δn)，即小于 nlock(max) 或大于 nlock(min)给出一个数，像所要求的那样通过锁存检测器去检测解锁的条件。输出时钟信号 ICGOUT 频率由 fICGDCLK/R 给出。在 FLL 被锁存时，过滤数值每四个周期的比较仅更新一次。更新的是一个错误测量的平均在原先四个的比较。

8.4.8 FLL 锁存和锁亏检测

为确定 FLL 锁定和锁亏的条件，脉冲计数器计数 DCO 的脉冲为一个周期的比较(见表 8-9 解释了比较周期)，把此数字存至 subtractor。Subtractor 比较这个值和 MFD 中的值，比较后产生一个计数误差 Δn 。为达到锁定状态， Δn 必须介于 nlock (min)和 nlock (max)。

在 FLL 已被锁定后, Δn 必须介于 $n_{unlock(min)}$ 和 $n_{unlock(max)}$, 以保持锁定。若 Δn 出人意料的超越这个范围, 设置 LOLS 状态位, 直到通过软件清零或 MCU 复位。LOLS 清理是通过读 ICGS1, 然后向 ICGIF 写 1 (LOLRE = 0), 或由一个锁亏引起复位 (LOLRE = 1), 或以任何 MCU 复位。

如果 ICG 进入关状态, 由于停止模式即 ENBDM = OSCSTEN = 0 时, FLL 失去锁定状态 (LOCK 被清除), 但 LOLS 维持不变, 因为这不是一个意想不到的锁亏条件。虽然它不寻常, 如果 ENBDM 被清除 0, 而 MCU 是停止的, ICG 进入该关断状态。因为这是一个意想不到的时钟停止, LOLS 将被置位, MCU 从停止中被唤醒。

预期锁亏发生, 是在改变 MFD 或 CLKS 位; 或仅在 FEI 模式下当 TRIM 位改变时。在这些情况下, LOCK 位将被清除, 直到 FLL 恢复锁存, 但将不会设置 LOLS。

8.4.9 FLL 时钟亏损检测

参考时钟和 DCO 时钟在不同条件下的监测(见表 8-8)。提供的参考频率是受到严密监控的, ERCS = 1 表明该参考时钟达到最低频率的要求。当参考和/或 DCO 时钟都受到严密监控, 如果其中一项属于低于某个频率, fLOR 和 fLOD, LOCS 状态位将被设置为显示错误。LOCS 将继续设置, 直到它被认知或直至 MCU 被复位。LOLS 清理是通过读 ICGS1, 然后向 ICGIF 写 1 (LOCRE = 0), 或由一个时钟亏损引起复位 (LOCRE = 1), 或以任何 MCU 复位。

如果 ICG 在 FEE 模式中, 参考时钟损失使得 ICG 进入 SCM, DCO 时钟亏损使得 ICG 进入 FBE 模式。如果 ICG 在 FBE 模式中, 参考时钟损失将使得 ICG 进入 SCM。在每种情况下, CLKST 和 CLKS 位将被自动改变, 以反映出新的状态。

如果 ICG 在 FEE 模式中, 当时钟损失产生时, ERCS 仍被设置为 1, CLKST 位设置为 10, ICG 回到 FBE 模式中。

时钟亏损也将造成锁亏, 在 FEE 或 FEI 模式下。因为清除 LOCS 和 LOLS 位是用一样的方法, 这只会是在极不可能的情况下即 LOLRE=1 和 LOCRE = 0 的一个问题。在这种情况下, 为锁亏的复位中断优先。

表 8-8 时钟监控(当 LOCD=0)

Mode	CLKS	REFST	ERCS	External Reference Clock Monitored?	DCO Clock Monitored?
Off	0X or 11	X	Forced Low	No	No
	10	0	Forced Low	No	No
	10	1	Real-Time ¹	Yes ⁽¹⁾	No
SCM (CLKST = 00)	0X	X	Forced Low	No	Yes ²
	10	0	Forced High	No	Yes ⁽²⁾
	10	1	Real-Time	Yes	Yes ⁽²⁾
	11	X	Real-Time	Yes	Yes ⁽²⁾
FEI (CLKST = 01)	0X	X	Forced Low	No	Yes
	11	X	Real-Time	Yes	Yes
FBE (CLKST = 10)	10	0	Forced High	No	No
	10	1	Real-Time	Yes	No
FEE (CLKST = 11)	11	X	Real-Time	Yes	Yes

¹ ENABLE 是高(等外部晶振在停止后开始)。

² DCO 时钟将不被监控直到 DCOS=1 从关闭或旁路 FLL 外部模式进入 SCM。

8.4.10 时钟模块要求

要求时钟模式是通过写入到 CLKST1: CLKST0，显示实际时钟模式通过 CLKST1: CLKST0。提供的满足最起码的条件，显示在 CLKST1: CLKST0 的状态应和在 CLKST1: CLKST0 要求的模式一样。表 8-9 显示了 CLKS、CLKST 和 ICGOUT 之间的关系，也显示了 CLKS=CLKST 的条件或 CLKS≠CLKST 的原因。

注意：若在下次复位之前使用晶振，确保设置 REFS=1、CLKS=1x 在首次写入到 ICGC1 寄存器中。失败是由于锁存 REFS=0，这将阻止振荡器放大直到下次复位产生。

表 8-9 ICG 状态表

Actual Mode (CLKST)	Desired Mode (CLKS)	Range	Reference Frequency (f _{REFERENCE})	Comparison Cycle Time	ICGOUT	Conditions ¹ for CLKS = CLKST	Reason CLKS1 ≠ CLKST
Off (XX)	Off (XX)	X	0	—	0	—	—
	FBE (10)	X	0	—	0	—	ERCS = 0
SCM (00)	SCM (00)	X	f _{ICGIRCLK} /7 ²	8/f _{ICGIRCLK}	ICGDCLK/R	Not switching from FBE to SCM	—
	FEI (01)	0	f _{ICGIRCLK} /7 ⁽¹⁾	8/f _{ICGIRCLK}	ICGDCLK/R	—	DCOS = 0
	FBE (10)	X	f _{ICGIRCLK} /7 ⁽¹⁾	8/f _{ICGIRCLK}	ICGDCLK/R	—	ERCS = 0
	FEE (11)	X	f _{ICGIRCLK} /7 ⁽¹⁾	8/f _{ICGIRCLK}	ICGDCLK/R	—	DCOS = 0 or ERCS = 0
FEI (01)	FEI (01)	0	f _{ICGIRCLK} /7	8/f _{ICGIRCLK}	ICGDCLK/R	DCOS = 1	—
	FEE (11)	X	f _{ICGIRCLK} /7	8/f _{ICGIRCLK}	ICGDCLK/R	—	ERCS = 0
FBE (10)	FBE (10)	X	0	—	ICGERCLK/R	ERCS = 1	—
	FEE (11)	X	0	—	ICGERCLK/R	—	LOCS = 1 & ERCS = 1
FEE (11)	FEE (11)	0	f _{ICGERCLK}	2/f _{ICGERCLK}	ICGDCLK/R ³	ERCS = 1 and DCOS = 1	—
		1	f _{ICGERCLK}	128/f _{ICGERCLK}	ICGDCLK/R ⁽²⁾	ERCS = 1 and DCOS = 1	—

- 1 在写入 CLKST 后 CLKST 将不会立即更新。在 CLKST 更新数值之前，需要几个总线周期。
- 2 在 SCM 中参考频率不影响 ICGOUT，但是参考频率仍用来做比较以决定 DCOS 位。
- 3 在最初的 LOCK 之后；ICGDCLK/2R 在最初锁存处理而在 MFD 位改变时 FLL 进行重新锁存。

8.4.11 时钟模块要求

ICG 提供了固定的频率时钟输出 XCLK 是通过片上外设来用的。这个输出相当于内部总线时钟，BUSCLK，在所有的模式中除了 FEE。在 FEE 模式中，XCLK 等于 ICGERCLK ÷ 2，当满足以下条件时：

- $(P \times N) \div R \geq 4$ 其中 P 是通过 RANGE 来确定的(见表 8-11), N 和 R 分别通过 MFD 和 RFD 来确定(见表 8-12)。
- LOCK=1。

如果上述条件不属实，那么 XCLK 等于 BUSCLK。

当 ICG 在 FEI 或 SCM 模式中时，关闭 XCLK。可用 XCLK 作为时钟源的外设绝不能这样做，当 ICG 在 FEI 或 SCM 模式中时。

8.4.12 高增益振荡器

振荡器选择运行在一个高增益振荡器(HGO)模式,这提高了振荡器对 EMC 噪音的阻力,当运行在 FBE 或 FEE 模式中时。即给 ICGC1 寄存器中的 HGO 位写入 1。高、低各种振荡器都能用 HGO,仅有一个有效的是当 ICGC1 寄存器中 REFS=1。当 HGO = 0,选定标准低功率振荡器。该位在复位后仅能写一次。

8.5 初始化/应用信息

8.5.1 绪论

本章节是给出一些基本的方向,在初始化 ICG 时,选择适合的配置。对于某些应用,串行通信链路可支配时钟参考的准确性。在其他的应用中,功耗最低的消费考虑总的时钟消耗。还有一些人可能以最低的成本作为首要目标。ICG 有极大弹性选择性,这对于任何应用是最好的。

表 8-10 ICG 配置考虑

	Clock Reference Source = Internal	Clock Reference Source = External
FLL Engaged	FEI $4 \text{ MHz} < f_{\text{Bus}} < 20 \text{ MHz}$. Medium power (will be less than FEE if oscillator range = high) Good clock accuracy (After IRG is trimmed) Lowest system cost (no external components required) IRG is on. DCO is on. ¹	FEE $4 \text{ MHz} < f_{\text{Bus}} < 20 \text{ MHz}$ Medium power (will be less than FEI if oscillator range = low) High clock accuracy Medium/High system cost (crystal, resonator or external clock source required) IRG is off. DCO is on.
FLL Bypassed	SCM This mode is mainly provided for quick and reliable system startup. $3 \text{ MHz} < f_{\text{Bus}} < 5 \text{ MHz}$ (default). $3 \text{ MHz} < f_{\text{Bus}} < 20 \text{ MHz}$ (via filter bits). Medium power Poor accuracy. IRG is off. DCO is on and open loop.	FBE $f_{\text{Bus}} \text{ range} \leq 8 \text{ MHz}$ when crystal or resonator is used. Lowest power Highest clock accuracy Medium/High system cost (Crystal, resonator or external clock source required) IRG is off. DCO is off.

¹ IRG 通常消耗 100 μA 。FLL 和 DCO 通常会消耗 0.5 至 2.5mA,取决于输出频率。对于最低功耗和最低的抖动,选择 N 和 R 尽可能小。

下面是配置初始化的例子。

注意:十六进制数前面是\$,二进制数前面是%,十进制数前面没有符号。最重要的配置信息在这里重复作为参考。

表 8-11 ICGOUT 频率计算选项

Clock Scheme	f_{ICGOUT}^1	P	Note
SCM — self-clocked mode (FLL bypassed internal)	f_{ICGDCLK} / R	NA	Typical $f_{\text{ICGOUT}} = 8 \text{ MHz}$ immediately after reset
FBE — FLL bypassed external	f_{ext} / R	NA	
FEI — FLL engaged internal	$(f_{\text{IRG}} / 7) * 64 * N / R$	64	Typical $f_{\text{IRG}} = 243 \text{ kHz}$
FEE — FLL engaged external	$f_{\text{ext}} * P * N / R$	Range = 0; P = 64 Range = 1; P = 1	

¹ 确保 f_{ICGDCLK} , 这个等于 $f_{\text{ICGOUT}} * R$, 不超过 $f_{\text{ICGDCLKmax}}$ 。

表 8-12 MFD 和 RFD 译码表

MFD Value	Multiplication Factor (N)
000	4
001	6
010	8
011	10
100	12
101	14
110	16
111	18

RFD	Division Factor (R)
000	+1
001	+2
010	+4
011	+8
100	+16
101	+32
110	+64
111	+128

8.5.2 例 1: 外部晶振=32kHz, 总线频率=4.19MHz

在这个例子中, 用 FLL(在 FEE 模式中)增加外部 32kHz 振荡达到 8.38MHz 用来实现 4.19MHz 总线频率。

在 MCU 从复位解脱之来后, ICG 处于自我时钟模式(SCM), 供给 ICGOUT 大约 8MHz, 这相当于 4MHz 总线频率(fBUS)。

时钟策划为 FLL 外部使能(FEE)。所以

$$f_{ICGOUT} = f_{ext} * P * N / R; P=64, f_{ext}=32kHz \quad \text{公式.8-1}$$

N/R 给出:

$$N/R = 8.38MHz / (32kHz * 64) = 4; \text{ 我们可以选择 } N=4, R=1 \quad \text{公式.8-2}$$

每个寄存器设置的数值:

ICGC1=\$38(%00111000)

- 位 7 HGO 0 设定振荡器的低功耗
- 位 6 RANGE 0 设定振荡器的低频率范围; FLL 调整因子是 64
- 位 5 REFS 1 振荡用晶体或共振器
- 位 4: 3 CLKS 11 FLL 外部参考时钟使能模块
- 位 2 OSCSTEN 0 振荡器不起作用
- 位 1 LOCD 0 时钟损失检测使能
- 位 0 0 未生效的或预留的, 总读入 0

ICGC2=\$00(%00000000)

- 位 7 LOLRE 0 产生中断请求在锁存损失时
- 位 6:4 MFD 000 设置 MFD 倍增因子为 4
- 位 3 LOCRE 0 产生中断请求在时钟损失时
- 位 2:0 RFD 000 设置 RFD 分频因子为 ÷1

ICGS1=\$xx

这仅能被读除非清除中断标志

ICGS2=\$xx

这仅能被读;应该读 DCOS=1 在执行任何关键性的任务

ICGFLTU/L=\$xx

仅需要在自我时钟模块; FLT 将被调整由回路给 8.38MHz DCO 时钟

位 15:12 未使用 0000

位 11:0 FLT 不需要为用户初始化

ICGTRM=\$xx

位 7:0 TRIM 当修正内部振荡器, 只需要写入; 当外部晶振是时钟源时, 不使用

图 8-14 显示 3 个条件的流程图, 需要 ICG 的初始化。

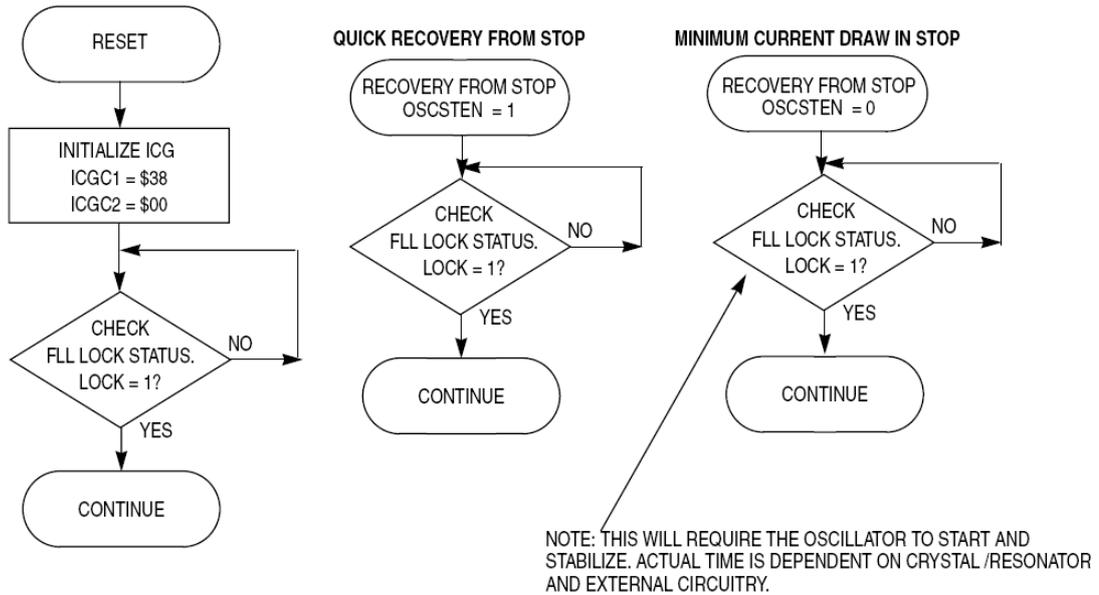


图 8-14 ICG 初始化 FEE 在例 1 中

8.5.3 例 2: 外部晶振=4MHz, 总线频率=20MHz

在这个例子中,用 FLL(在 FEE 模式中)增加外部 4kHz 振荡达到 40MHz 用来实现 20MHz 总线频率。

在 MCU 从复位解脱之来后, ICG 处于自我时钟模式(SCM), 供给 ICGOUT 大约 8MHz, 这相当于 4MHz 总线频率(fBUS)。

在复位初始化软件, 时钟策划将为 FLL 外部使能(FEE)。所以

$$f_{ICGOUT} = f_{ext} * P * N / R; P=1, f_{ext}=4.00\text{MHz} \quad \text{公式.8-1}$$

N/R 给出:

$$N/R = 40\text{MHz} / (4\text{MHz} * 1) = 10; \text{ 我们可以选择 } N=10, R=1 \quad \text{公式.8-2}$$

每个寄存器设置的数值:

ICGC1=\$78(%01111000)

位 7	HGO	0	设定振荡器的低功耗
位 6	RANGE	1	设定振荡器的低频率范围; FLL 调整因子是 1
位 5	REFS	1	振荡用晶体或共振器
位 4:	3 CLKS	11	FLL 外部参考时钟使能模块
位 2	OSCSTEN	0	振荡器不起作用
位 1	LOCD	0	时钟损失检测使能
位 0		0	未生效的或预留的, 总读入 0

ICGC2=\$30(00110000)

位 7	LOLRE	0	产生中断请求在锁存损失时
位 6:4	MFD	011	设置 MFD 倍增因子为 10
位 3	LOCRE	0	产生中断请求在时钟损失时
位 2:0	RFD	000	设置 RFD 分频因子为 ÷1

ICGS1=\$xx

这仅能被读除非清除中断标志

ICGS2=\$xx

这仅能被读;应该读 DCOS=1 在执行任何关键性的任务

ICGFLTU/L=\$xx

在这个例子中没用到

ICGTRM

在这个例子中没用到

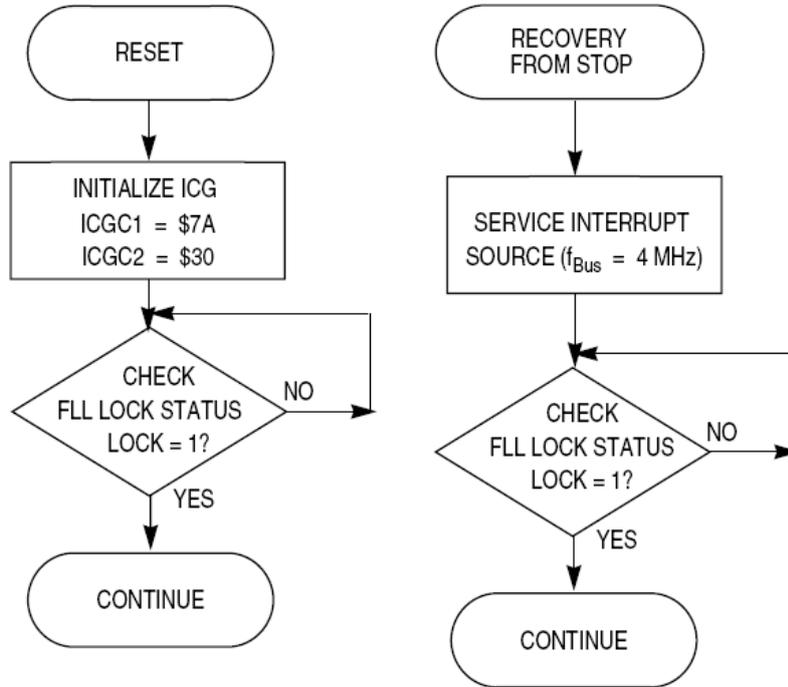


图 8-15 ICG 初始化和停止恢复例

8.5.4 例 3: 无外部晶振连接，总线频率=5.4MHz

在这个例子中，用 FLL(在 FEI 模式中)增加外部 243kHz 振荡达到 10.8MHz 用来实现 5.4MHz 总线频率。

在 MCU 从复位解脱之来后，ICG 处于自我时钟模式(SCM)，供给 ICGOUT 大约 8MHz，这相当于 4MHz 总线频率(fBUS)。

时钟策划为 FLL 外部使能(FEI)。所以

$$f_{ICGOUT} = (f_{IRG}/7) * P * N/R; P=64, f_{IRG}=243kHz \tag{公式.8-1}$$

N/R 给出:

$$N/R = 10.8MHz / (243/7kHz * 64) = 4.86; \text{ 我们可以选择 } N=10, R=2 \tag{公式.8-2}$$

每个寄存器设置的数值:

ICGC1=\$28(%00101000)

- 位 7 HGO 0 设定振荡器的低功耗
- 位 6 RANGE 0 设定振荡器的低频率范围; FLL 调整因子是 64
- 位 5 REFS 1 振荡用晶体或共振器
- 位 4: 3 CLKS 01 FLL 内部参考时钟使能模块
- 位 2 OSCSTEN 0 振荡器不起作用
- 位 1 LOCD 0 时钟损失检测使能
- 位 0 0 未生效的或预留的, 总读入 0

ICGC2=\$31(%00110001)

- 位 7 LOLRE 0 产生中断请求在锁存损失时
- 位 6:4 MFD 011 设置 MFD 倍增因子为 10
- 位 3 LOCRE 0 产生中断请求在时钟损失时
- 位 2:0 RFD 001 设置 RFD 分频因子为 ÷2

ICGS1=\$xx

这仅能被读除非清除中断标志

ICGS2=\$xx

这仅能被读;应该读 DCOS=1 在执行任何关键性的任务

ICGFLTU/L=\$xx

在这个例子中没用到

ICGTRM=\$xx

位 7:0 TRIM 当修正内部振荡器, 只需要写入; 单独运行(见例 4)

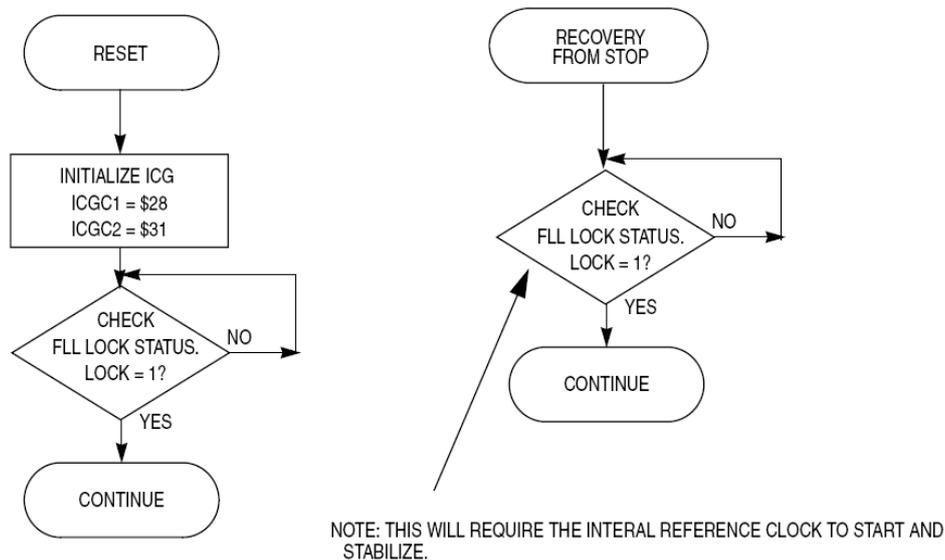


图 8-16 ICG 初始化和停止恢复例 3

8.5.5 例 4: 内部时钟发生器修正

该内部产生的时钟源是保证有一个周期数值±25%。在一些情况下,这可能是足够的精度。作其他用途,需要严格的频率偏差,修剪的程序规定,将允许非常准确源。本节概述的一个修正内部振荡器的例子。还可能其他多种修正程序是有效的和可以使用。

初始条件:

从 ATE 提供的时钟有 500µs 周期

ICG 配置 4MHz 总线内部参考

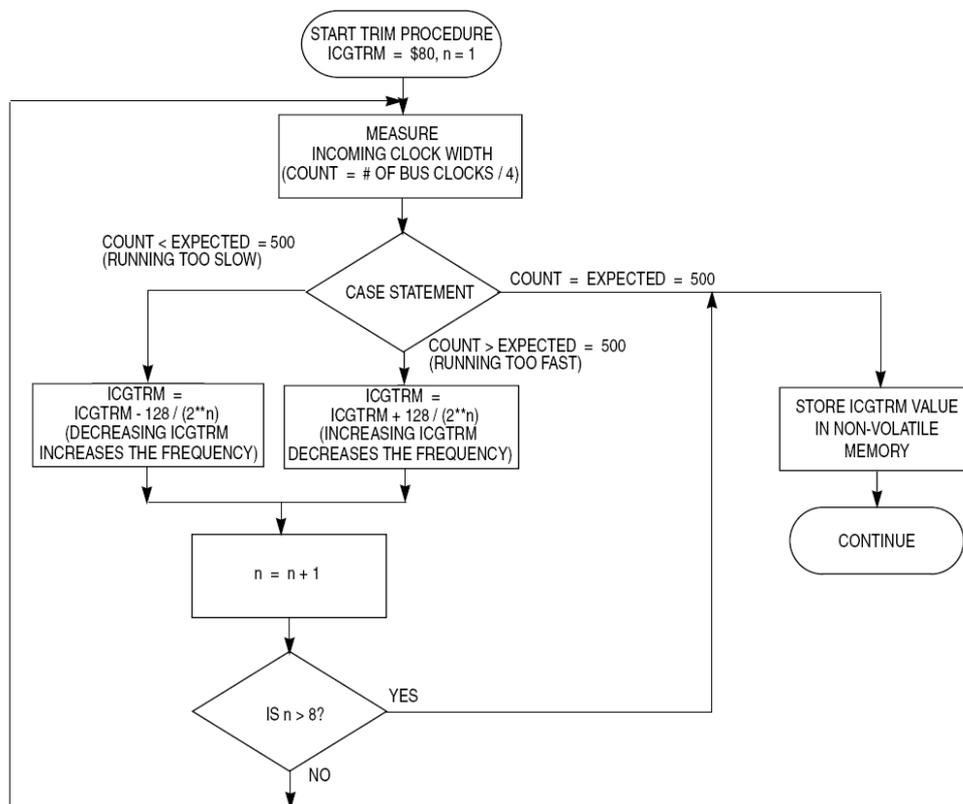


图 8-17 修正程序

在此特定情况下，MCU 已连接到印刷电路板和整个是通过自动测试设备正在进行最后测试。一个单独的信号或讯息提供给 MCU 的运行是在用户提供的软件控制下。MCU 启动修正程序，所图 8-17 所示的数据，而测试仪提供了精确的参考信号。

如果总线是在所允许的设备的最高值附近，使用减少除数 (R)，最后值的两倍。在修正程序完成之后，减少的除数能够恢复。这将防止意外超过最大时钟频率。

第九章 键盘中断(S08KBIV1)

9.1 介绍

MC9S08AW60 系列有一个带 8 个键盘中断输入的 KBI 模块，与端口 D 和端口 G 引脚合用。见第 2 章节，“引脚和连接”，更详细的介绍了关于这些引脚的逻辑和硬件方面。

9.2 键盘引脚

KBI 输入 KBIP7 共用通用引脚 PTD7 和 AD15。KBIP7 启用时，引脚是输入状态不管端口 D 数据方向位的值。端口 D 上拉使能仍然用来控制上拉寄存器，引脚状态通过读端口 D 数据寄存器来识别(要求端口 D 的第 7 位方向寄存器为 0)。在启用引脚作为一个 ADC 输入时，PTD7 和 KBIP7 都不起作用，包括上拉寄存器。

KBI 输入 KBIP6 共用引脚 PTD3 和 AD11，KBI 输入 KBIP5 共用引脚 PTD2 和 AD10。这些输入引脚的共用功能与上面讲的 KBIP7 一样。

KBI 输入 KBIP4~ KBIP0 共用通用引脚 PTG4~ PTG0。这些引脚功能与上面讲的 KBIP7 一样，但它们都不与 ADC 输入共用引脚。

KBIP3~ KBIP0 总是受到下降沿/低电平影响。KBIP7~ KBIP4 设置为受上升沿/高电平或下降沿/低电平影响。当 KBIP7~ KBIP0 任何一个输入启用时，设置去检测上升沿/高电平，引脚上拉启用是通过相应的端口上拉启用那个引脚的位，启用引脚的上拉寄存器而不是下拉寄存器。

表 9-1 KBI 和相应 I/O 的作用

PTxPE _n (上拉使能)	PTxDD _n (数据方向)	KBIPEn (KBI 引脚使能)	KBEDG _n (KBI 边缘选择)	上拉	下拉
0	0	0	X ¹	不启用	不启用
1	0	0	x	启用	不启用
x	1	0	x	不使用	不启用
1	x	1	0	启用	不启用
1	x	1	1	不启用	启用
0	x	1	x	不启用	不启用

¹ X=不要注意

9.3 特征

键盘中断(KBI)模块特征包括：

- 4 个下降沿/低电平敏感
- 4 个下降沿/低电平或上升沿/高电平敏感
- 选择仅边沿或边沿和电平敏感
- 通用中断标志和中断使能控制
- 能够从 stop3 或停止模式中唤醒

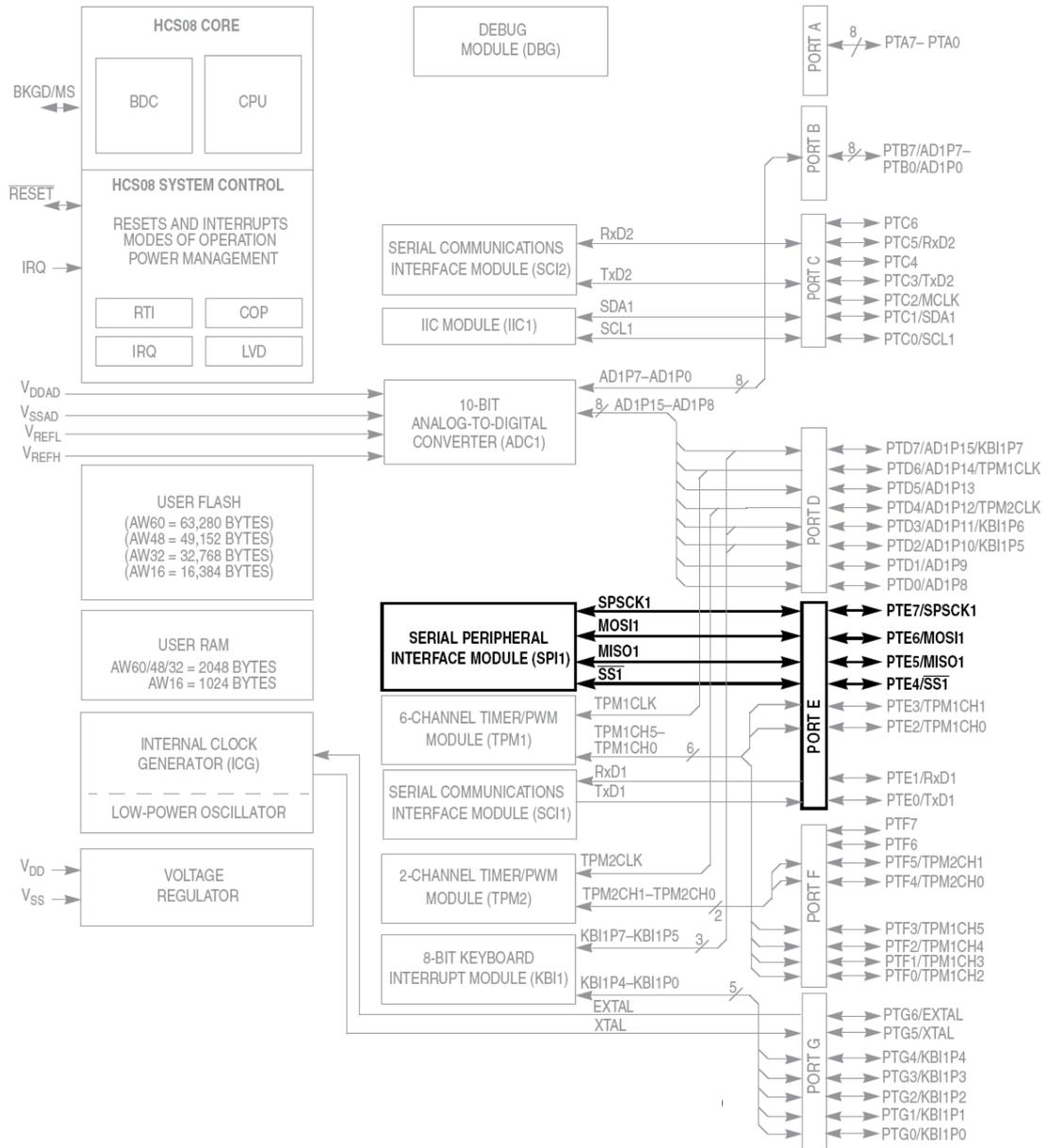


图 9-1 结构图突出 KBI 模块

注意:

1. 如果是输入端口，端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用 (IRQEDG=1)，引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测，下拉式启用 (IRQPE = 1)。
3. IRQ 没有一个二极管到 V_{DD} ，IRQ 不应该超过 V_{DD} 。

4. 引脚包含完整的上拉装置。
5. 引脚 PTD7, PTD3, PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用(KBIPEn = 1)并选定上升沿 (KBEDGn = 1)时, 下拉式启用时。

9.3.1 KBI 块框图

图 9-2 显示了 KBI 模块的结构图。

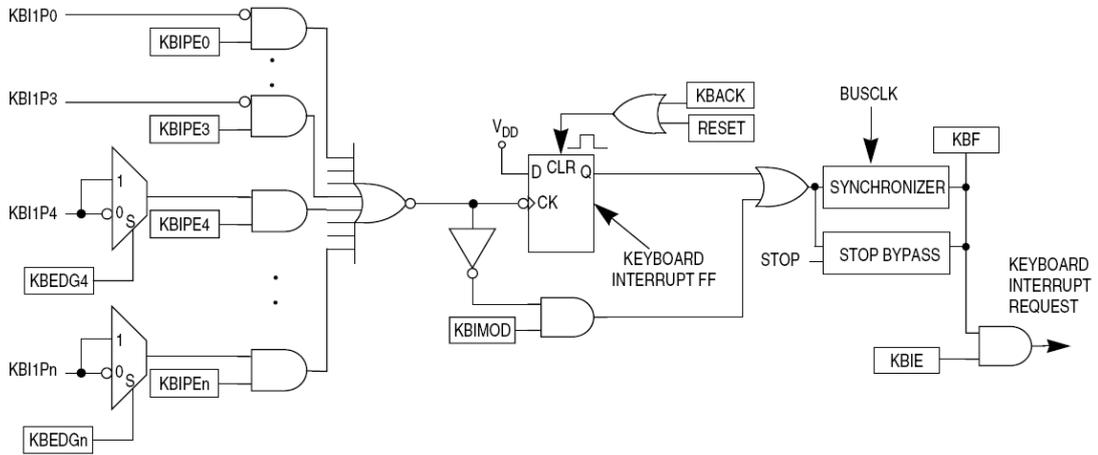


图 9-2 KBI 块框图

9.4 寄存器定义

图 9-2 显示了 KBI 模块的结构图。本章节提供了所有与 KBI 模块有关的寄存器和控制位。

如需了解所有 KBI 寄存器的绝对地址分配, 参见本文“寄存器”章的直接页面寄存器概述。本小节只提及了寄存器和控制位的名称, 飞思卡尔提供的对照表或头文件用来把这些名称转换成适当的绝对地址。

9.4.1 KBI 状态和控制寄存器(KBI1SC)

	7	6	5	4	3	2	1	0
R	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
W						KBACK		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

图 9-3 KBI 状态和控制寄存器(KBI1SC)

表 9-2. KBI1SC 寄存器字段描述

字段	描述
7:4 KBEDG[7:4]	<p>键盘的边缘选择为KBI端口位—每个读/写位选择边缘的方向和/或电平, 是用来作为启动相应的KBI端口引脚, 当它被配置作为一个键盘输入时 (KBIPEn=1)。也可看到是KBIMOD控制位, 来决定引脚是否对边沿或边沿和电平敏感。</p> <p>0 下降沿/低电平</p> <p>1 上升沿/高电平</p>

3 KBF	<p>键盘中断标志—设置这个仅能读的标志是在被选边沿已被检测在任何使能的KBI端口引脚。该标志是通过写1到KBACK控制位来清除的。若KBIMOD=1来选择边沿和水平操作时，标志将仍然被设置，任何使能KBI端口引脚仍然在定义的电平。</p> <p>KBF可作为软件可查询的标志（KBIE=0）或是能发出一个硬件中断请求到CPU（KBIE=1）。</p> <p>0 没有KBI中断</p> <p>1 KBI中断</p>
2 KBACK	<p>键盘中断确认—该只读位通过写1到KBACK来清除KBF状态标志。KBIMOD=1用来选择边沿和电平操作，任何使能的KBI端口引脚仍然在定义的电平，KBF将继续被设置，所以写入1到KBACK不清除KBF标志。</p>
1 KBIE	<p>键盘中断使能—读写控制位决定是否软件中断发生在KBF状态标志是1。当KBIE=0，没有产生软件中断，但是KBF仍被用作软件投票。</p> <p>0 KBF没有产生硬件中断（用投票）</p> <p>1 KBI请求硬件中断当KBF=1</p>
KBIMOD	<p>键盘检测模块—读写控制位选择仅边沿检测或边沿和电平检测。KBI端口位3到0能检测下降沿或下降沿和低电平。KBI端口位7到4设置去检测下面情况：</p> <ul style="list-style-type: none"> •仅上升沿或上升沿和高电平(KBEDGn=1) •仅下降沿或下降沿和低电平（KBEDGn=0） <p>0 仅边沿检测</p> <p>1 边沿和电平检测</p>

9.4.2 KBI 管脚使能寄存器 (KBI1PE)



图 9-4. KBI 管脚使能寄存器

表 9-3. KBIPE 寄存器字段描述

字段	描述
7:0 KBIPE[7:0]	<p>键盘管脚使能为KBI端口位—每个读写位选择是否作为键盘中断输入与KBI端口引脚有关的使能或作为一个通用的I/O引脚功能。</p> <p>0 KBI端口位n是一个通用I/O引脚与KBI无关</p> <p>1 KBI端口位n作为一个键盘中断输入使能</p>

9.5 功能描述

9.5.1 引脚使能

在 KBI1PE 寄存器中的 KBIPE_n 控制位允许使用者启用(KBIPE_n=1)与 KBI 有关的端口引脚的任何组合连接到 KBI 模块。相当于 KBI1PE 中 0s 的引脚是通用 I/O 引脚，那与 KBI 模块无关。

9.5.2 边沿和电平敏感性

同步逻辑是用来检测边沿。在检测一个边沿之前，在一个 KBI 模块的使能键盘的输入必须是在未宣布的逻辑电平。

检测下降沿是在一个使能键盘输入信号看做是在一个总线周期内的逻辑 1（未宣布的电平），接着是在下一个总线周期内的逻辑 0（宣称的电平）。

检测上升沿是在输入信号看做是在一个总线周期内的逻辑 0，接着是在下一个总线周期内的逻辑 1。

设置 KBIMOD 控制位去重新证实检测的逻辑，这样它检测边沿和电平。在 KBIMOD=1 模块中，当检测一个边沿时，KBF 状态标志适合设置（当一个或更多的使能引脚从未宣布的到宣称的电平改变，然而所有其他引脚仍然在未宣布的电平），但是继续设置标志（不能被清除）只要任何使能键盘输入引脚仍然在宣称的电平。当 MCU 进入 stop3 模式时，同步边沿检波逻辑是避开的（因为时钟停止了）。在 stop3 模式中，KBI 输入担当异步电平敏感的输入，所以他们能从 stop3 模式中唤醒 MCU。

9.5.3 KBI 中断控制

KBF 状态标志适合置位（1）是在任何 KBI 输入引脚检测一个边沿时。如果在 KBI1SC 寄存器中的 KBIE=1，将有一个硬件中断请求使得 KBF=1。KBF 标志是通过写 1 到键盘确认（KBACK）来清除。

当 KBIMOD=0（选择仅边沿操作），KBF 总是通过写 1 到 KBACK 来清除。当 KBIMOD=1（选择边沿和电平操作），KBF 不能被清除只要任何键盘输入的是在宣称的电平。

第十章 定时器/脉冲调制器(S08TPMV2)

10.1 介绍

MC9S08AW60 系列包括两个独立的定时器/脉宽调制(TPM)模块,支持传统的输入捕捉、输出比较、或已缓冲的优势,对每个通道的脉冲宽度调制 (PWM)。在每个 TPM 控制位配置所有的通道,定时器作为中心 PWM 职能进行操作。对于两个 TPMS 中的每一个,时机职能是基于一个单独的 16 位计数器分频器,并以特点为模来控制频率和时间参考的范围(溢出周期内)。这个计时系统非常适合范围广泛的控制等方面的应用,以及中心不结盟的 PWM 能力,对 3 个通道 TPM 的扩展应用领域,以电机控制小家电。

固定系统时钟的使用, XCLK , 作为时钟源为 TPM 模块允许 TPM 的分频器运行,使得振荡器速率除以 2 (ICGERCLK/ 2)。此选项仅适用如果 ICG 设定在 FEE 模式和适当的条件得到满足时(见第 8.4.11 章节,“固定高频时钟”)。在所有 ICG 模式中,这个选项是多余的,因为 XCLK 相当于 BUSCLK。

10.2 特性

MC9S08AW60 系列中定时器系统包括一个 6 通道的 TPM1 和一个独立的 2 通道 TPM2。定时器系统的功能包括:

- 总共 8 个通道:
 - 每个通道可以是输入捕捉、输出比较或边缘对齐 PWM 模式
 - 上升边、下降边或任何边输入捕捉触发
 - 设置、清除、切换输出比较操作
 - PWM 输出上的可选极性
- 在每个通道上,模块可配置为缓冲、中央对齐脉冲宽度调制 (CPWM)
- 时钟源为每个 TPM 分频单独可选作为总线时钟,固定式系统时钟,或外部引脚:
 - 除以 1、2、4、8、16、32、64 或 128 的预分频器值相位
 - 外部时钟输入 TPM1 的 TPM1CLK 和 TPM2 的 TPM2CLK (只适用于 64 引脚封装)
- 16 位自由运行或模数向上/向下(CPWM)计数操作
- 16 位模寄存器来控制计数范围
- 定时器系统启动
- 每个通道一个中断以及终端计数中断为每个模块的 TPM

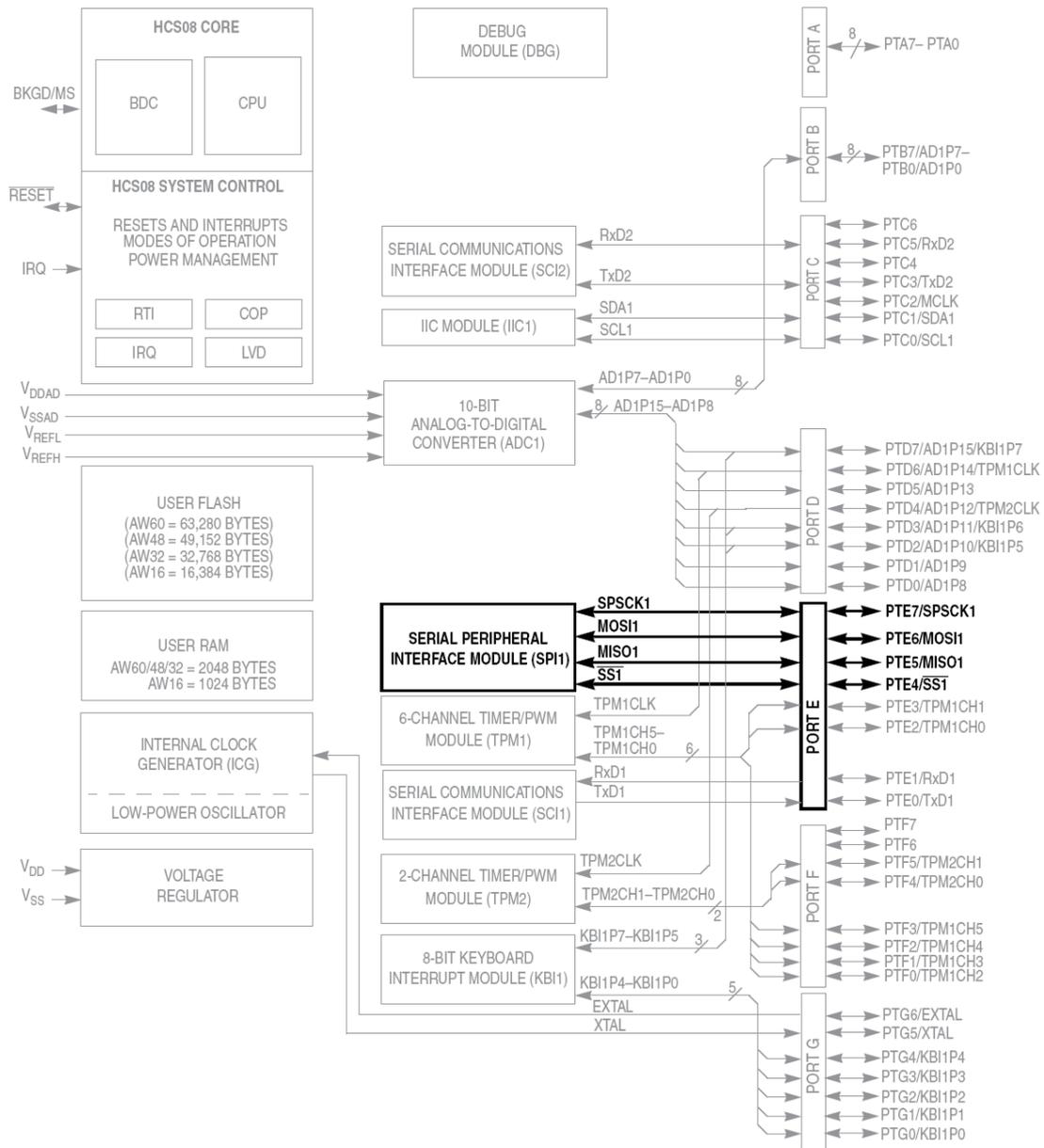


图 10-1. 结构图突出 TPM 模块

注意:

1. 如果是输入端口，端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用(IRQEDG=1)，引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测，下拉式启用(IRQPE = 1)。
3. IRQ 没有一个二极管到 V_{DD}，IRQ 不应该超过 V_{DD}。
4. 引脚包含完整的上拉装置。
5. 引脚 PTD7，PTD3，PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用(KBIPE_n = 1)并选定上升沿(KBEDG_n = 1)时，下拉式启用时。

10.2.1 特性

TPM 有如下的特性:

- 在每个通道上, 模块可配置为缓冲、中央对齐脉冲宽度调制 (CPWM)
- 时钟源为每个 TPM 分频单独可选作为总线时钟, 固定式系统时钟, 或外部引脚:
- 可选的时钟源 (依赖装置): 总线时钟, 固定的系统时钟, 外部引脚
- 除以 1、2、4、8、16、32、64 或 128 的预分频器值相位
- 16 位自由运行或模数向上/ 向下(CPWM)计数操作
- 16 位模寄存器来控制计数范围
- 定时器系统启动
- 每个通道一个中断以及终端计数中断为每个模块的 TPM(多个 TPMs 装置)
- 通道特征:
 - 每个通道可以是输入捕捉、输出比较或边缘对齐 PWM 模式
 - 上升边、下降边或任何边输入捕捉触发
 - 设置、清除、切换输出比较操作
 - PWM 输出上的可选极性

10.2.2 结构图

图 10-2 显示了 TPM 的结构。一些 MCUs 有一个以上的 TPM, 与各种通道。

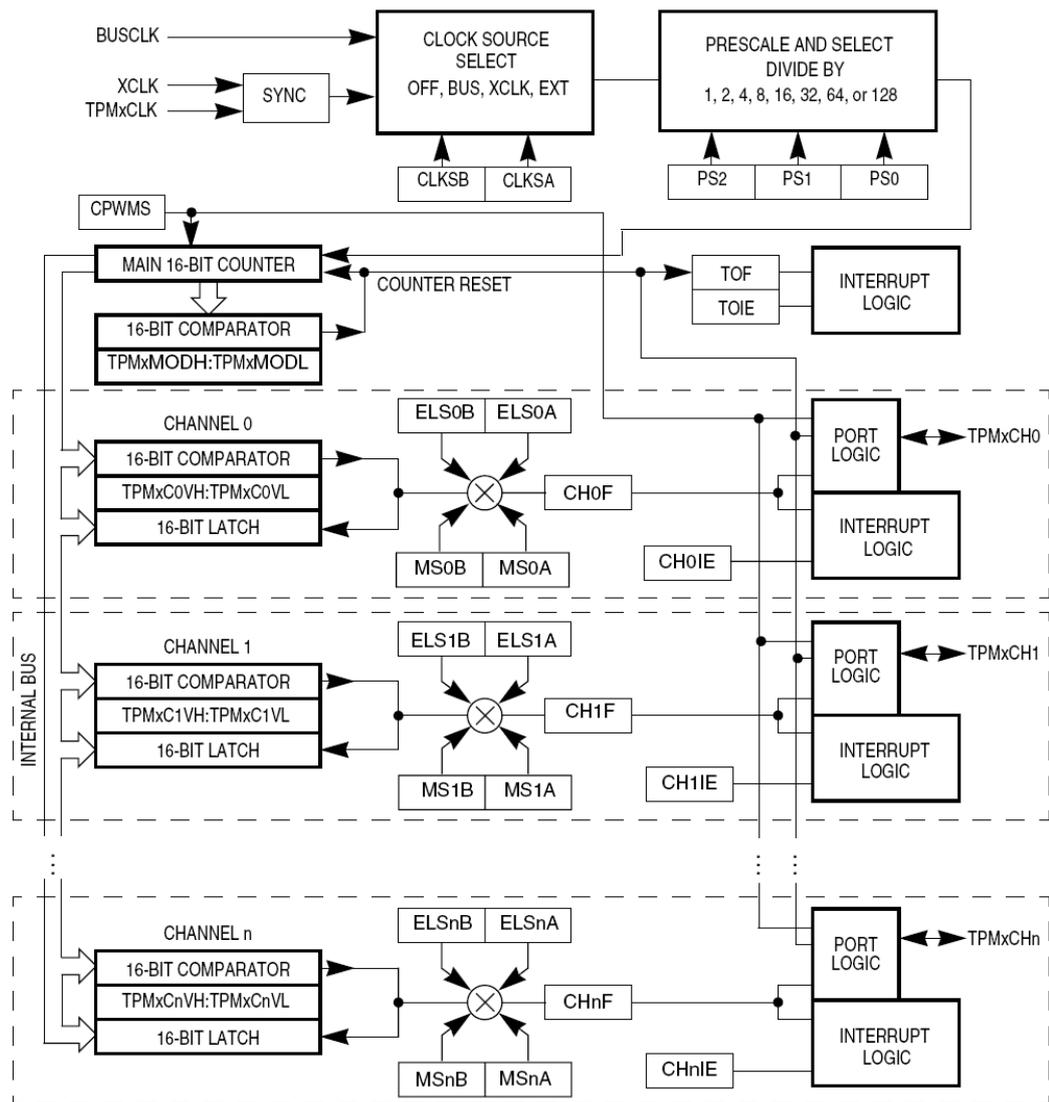


图 10-2. TPM 结构图

TPM 的中心部件是 16 位计数器，它可以作为一个自由运行计数器来操作，一个模计数器，或一个向上/ 向下(CPWM)计数器。当 TPM 被配置为中心成一线的 PWM。TPM 计数器（当操作在正常向上计数模式时）提供了参考时间为输入捕捉、输出比较和边缘对齐的 PWM 功能。计时模寄存器，TPMMODH:TPMMODL，控制计数器的模值。（值 0x0000 或 0xFFFF 有效地使计数器自由运行）软件可以在任何时候读取计数器数值而不会影响计数顺序。无论写入的数值是什么，任何写入到 TPMCNT 计数器的字节都会复位计数器。所有的 TPM 通路都可以独立编程为输入捕捉、输出比较或者缓冲的边沿排列 PWM 信道。

10.3 外部信号描述

当与定时器关联的任一管脚被配置为一个定时器输入时，被动上拉启用。复位后，TPM 模块被禁用并且所有管脚默认为通用输入端口，但是被动上拉被禁用。

10.3.1 外部 TPM 时钟源

当定时器状态和控制寄存器的控制位 CLKSB:CLKSA 被设置为 1:1 时，预定标器和 TPM 的 16 位计数器被外部时钟源 TPMxCLK 驱动，并连接到 I/O 管脚。在外部时钟和 TPM 的其

余部分之间，需要同步装置。这种同步装置根据总线时钟定时，所以外部时钟源的频率必须小于总线速率时钟频率 1/2。为了适当地调节工作负载循环和锁相环(PLL)或频率锁定环(FLL)频率抖动影响，该外部时钟源的频率上限被指定为 1/4 总线频率。

在某些器件内，其中一个 TPM 通道可共享为外部时钟输入。当一个 TPM 通道被共享作为外部时钟输入时，相关的 TPM 通道则不能使用这个管脚。（该通道可依然在输出比较模式中作为软件定时器）同时，如果 TPM 通道中某一个用于外部时钟输入，那么相应的 ELSnB:ELSnA 控制位则必须被设置为 0:0，而这个通道也不再使用同一管脚。

10.3.2 TPMxCHn—TPMx 通道 n I/O 管脚

每一个 TPM 通道都与 MCU 上的一个 I/O 管脚关联。这个管脚的功能依赖于该信道的配置。在某些情况下，有的管脚功能是不需要的，所以这个管脚重新被通用 I/O 控制。当一个定时器能控制端口管脚时，这个端口数据和数据方向寄存器不会影响相关（多个）管脚。关于共享管脚功能的更多信息，请参考“管脚与连接”一章。

10.4 寄存器定义

TPM 包括：

- 一个 8 位状态和控制寄存器（TPMSC）
- 一个 16 位计数器（TPMCNTH:TPMCNTL）
- 一个 16 位模寄存器（TPMMODH:TPMMODL）

每个定时器通道包含：

- 一个 8 位状态和控制寄存器（TPMCnSC）
- 一个 16 位通道值寄存器（TPMxCnVH:TPMxCnVL）

所有 TPM 寄存器的绝对地址分配请参考本技术手册中“存储器”一章的“直接页面寄存器总结”，在本节中仅通过它们的名字对寄存器和控制位进行引用。飞思卡尔提供的头文件用于将这些名字解释为适当的绝对地址。

一些 MCU 系统有一个以上的 TPM，所以寄存器的姓名包括占位符字符识别其中 TPM 和哪个通道正在被参照。举例来说，TPMxCnSC 指的是定时器(TPM)x, 通道 n 和 TPM1C2SC 状态和控制寄存器为定时器 1，通道 2。

10.4.1 定时器状态和控制寄存器（TPMxSC）

TPMxSC 包含用于配置中断使能、TPM 配置、时钟源和预定标因子的溢出状态标志和控制位。这些位涉及该定时器模块中的所有通道。

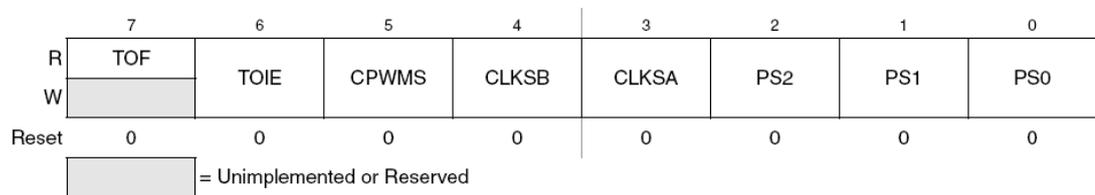


图 10-3. 定时器状态和控制寄存器（TPMxSC）

表 10-1. TPMxSC 寄存器字段描述

字段	描述
7	定时器溢出标志—当TPM计数器在达到TPM计数模寄存器编程的模值后变为0x0000时，该

TOF	<p>标志被置位。当TPM被配置为CPWM时，在计数器达到模寄存器中的值后，将要过渡为一个更低的计数值时，TO被置位。当TOF被置位时，通过读取TPM状态和控制寄存器再向TOF写0，可清TOF。如果在清除操作完成之前另一个TPM 发生溢出，那么，清除操作重新执行，因此，在对前一个TOF的清除操作完成后，TOF将保持置位状态。复位清除TOF。向TOF写1无效。</p> <p>0 TPM计数器没有达到模值或溢出</p> <p>1 TPM计数器溢出</p>
6 TOIE	<p>定时器溢出中断使能—该读/写位使能TPM溢出中断。如果TOIE被置位，当TOF等于1时，产生一个中断。复位可清除 TOIE。</p> <p>0 TOF 中断禁用（使用软件轮询）</p> <p>1 TOF 中断使能</p>
5 CPWMS	<p>中心排列PWM选择—该读/写位选择CPWM运行方式。复位可清除此位，因此，对于输入俘获、输出比较和边沿排列PWM功能，TPM以增计数模式运行。对于CPWM 功能，设置 CPWMS 可将TPM重新配置为以增/减计数模式运行。复位可清除 CPWMS。</p> <p>0 所有TPM通道运行于输入捕捉、输出比较或边沿排列PWM模式，通过对每个通道的状态和控制寄存器的MSnB:MSnA进行选择。</p> <p>1 所有TPM通道以中心排列PWM模式运行。</p>
4:3 CLKS[B:A]	<p>时钟源选择——如表 10-2 所示，这两位用于禁用TPM系统，或选择三个时钟源中的一个用于驱动计数预定标器。通过一个片上同步电路，外部时钟源和 XCLK 与总线时钟进行同步。</p>
2:0 PS[2:0]	<p>预定标因子选择—此三位用于为TPM时钟输入选择八个因子中的其中一个，如表10-3所示。在任意时钟源同步或时钟源选择后，可定位预定标器，因此，它对选择什么时钟源来驱动TPM系统有影响。</p>

表 10-2. TPM 时钟源选择

CLKSB:CLKSA	预置输入的TPM时钟源
0: 0	没有选择的时钟 (TPM屏蔽)
0: 1	总线速率时钟(BUSCLK)
1: 0	固定系统时钟(XCLK)
1: 1	外部源(TPMCLK) ^{1,2}

¹ 允许作为外部时钟的最大频率为总线频率 1/4。

² 外部时钟输入与通道 n 共享，并被选为 TPM 时钟源，相应的 ELSnB:ELSnA 控制位应该被设置为 0:0，因此，通道 n 不会因为一个相互冲突的功能而使用同一管脚。

表 10-3. 选择预置分频器

PS2:PS1 :PS0	TPM Clock Source Divided-By
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

10.4.2 定时器 x 计数寄存器 (TPMxCNTH:TPMxCNTL)

这两个只读 TPM 计数寄存器包含 TPM 计数器中的值的高字节和低字节。读取任一字节 (TPMCNTH 或 TPMCNTL) 可将两字节中的内容锁入一个缓存器, 在该缓存器中这两个字节的内容保持被锁状态直到另一字节也被读取。这允许相参的 16 位数据以两种顺序中任何一种被读取。通过 MCU 复位、向 TPMCNTH 或 TPMCNTL 写入一个任意值、或向定时器状态/控制寄存器 (TPMSC) 任意写操作可自动重启相参机制。

复位可清除 TPM 计数寄存器。

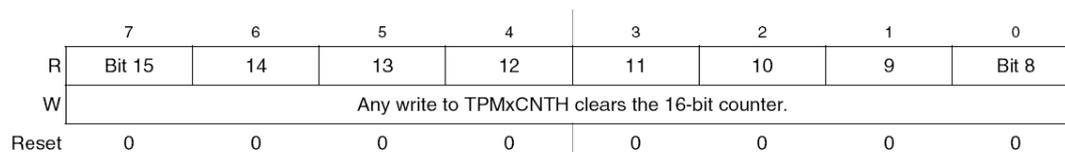


图 10-4. 定时 x 计时寄存器高 (TPMxCNTH)

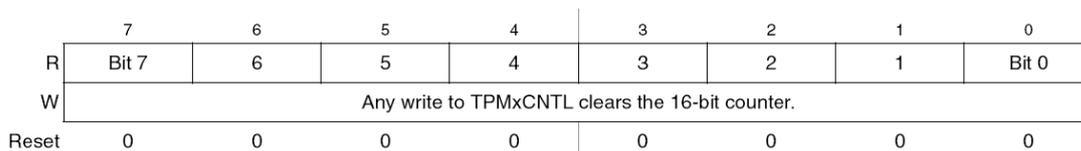


图 10-5. 定时 x 计时寄存器低 (TPMxCNTL)

当后台方式处于活动状态时, 计时器和相参机制被冻结, 从而使得缓存器锁存器保持后台方式变为活动时的状态, 即使在后台方式处于活动状态时计数器的一个或两个字节被读取。

10.4.3 定时器 x 计时模寄存器 (TPMMODH:TPMMODL)

读/写 TPM 模寄存器包含 TPM 计数器的模值。在 TPM 计数器达到模值后, TPM 计数器在下一时钟周期 (CPWMS=0) 重新从 0x0000 开始计数或开始递减计数 (CPWMS=1), 且溢出标志 (TOF) 被置位。写 TPMMODH 或 TPMMODL 可禁止 TOF 和溢出中断直至另一字节被写。复位将 TPM 计数器模寄存器设置为 0x0000, 这将产生一个不同步计时器 (模禁用时)。

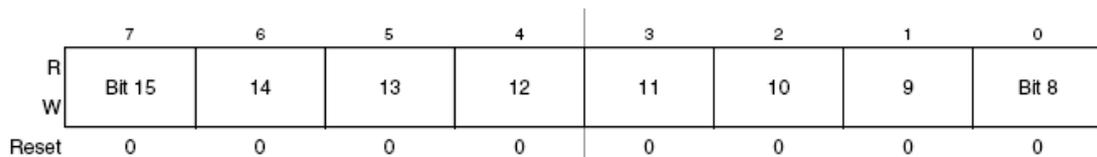


图 10-6. 计时器模寄存器高 (TPMxMODH)

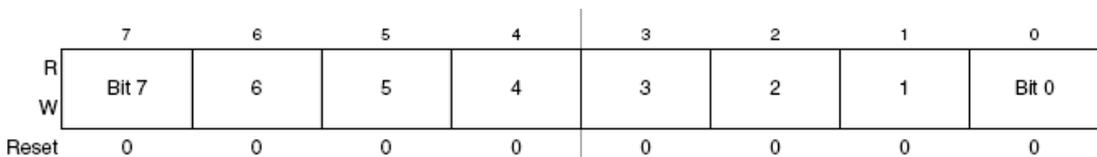


图 10-7. 计时器模寄存器低 (TPMxMODL)

在一个新的溢出出现前，最好是等待一个溢出中断，从而使得模寄存器的两个字节易于进行写操作。另一种方法是，在写 TPM 模寄存器之前对 TPM 计数器进行复位，以避免对何时将会发生第一个计数器溢出产生混淆。

10.4.4 定时器 x 通道 n 状态和控制寄存器 (TPMxCnSC)

TPMxCnSC 包含用于配置中断使能、通道配置和管脚功能的通道中断状态标志和控制位。

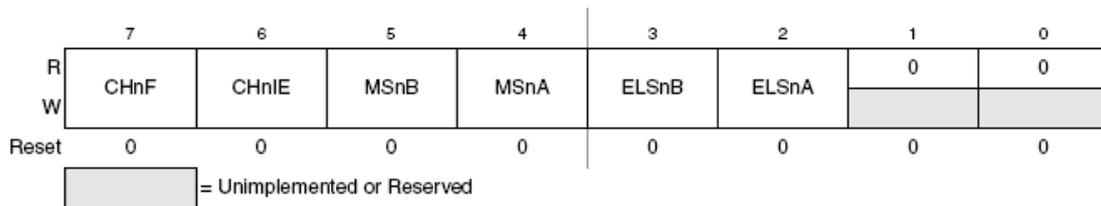


图 10-8. 定时器 x 通道 n 状态和控制寄存器 (TPMxCnSC)

表 10-4. TPMxCnSC 寄存器字段描述

字段	描述
7 CHnF	<p>通道n标志—通道n被配置为输入捕获时，当一个活动边沿在通道n管脚出现时，此标志位被置位。通道n为一个输出比较或边沿排列PWM通道时，当TPM计数寄存器中的值与TPM通道n数值寄存器中的植匹配时，CHnF被置位。此标志很少用于中心排列 PWM，因为每当计数器与通道数值寄存器匹配时，对应于活动周期的两个沿，该标识都将被置位。当CHnF被置位且中断被使能 (CHnIE=1) 时，会进行相应的中断请求。当CHnF被置位时，通过读TPMCnSC，然后向CHnF写0，可清除CHnF。如果在清除操作完成之前出现另一个中断请求，清除操作被复位，因此，CHnF将保持置位状态直到对前一CHnF的清除操作完成。这样可以使得一个 CHnF中断请求不会因为清除前一个CHnF而丢失。复位可清除CHnF。对CHnF写1无效。</p> <p>0 在通道n无输入捕获或输出比较事件发生。</p> <p>1 在通道n发生输入捕获或输出比较事件。</p>
6 CHnIE	<p>通道n中断使能—该读/写位使能来自通道 n 的中断。复位可清除 CHnIE。</p> <p>0 通道n中断请求禁止 (使用软件轮询)</p>

	1 通道n中断请求使能
5 MSnA	为TPM通道n模式选择B —当CPWMS=0时，MSnB=1配置TPM通道n为边沿排列PWM模式。通道模式和设置控制的概述，参照表10-5。
3:2 ELSn[B:A]	边沿/电平选择位 —根据CPWMS:MSnB:MSnA设置的定时器通道的运行方式，如表10-5所示，这些位可选择触发输入捕获事件的输入边沿极性、选择响应输出比较匹配的驱动电平、或选择PWM输出的极性。 将ELSnB:ELSnA设置为0:0可将相关定时器管脚配置为通用I/O管脚，而与定时器通道功能无关。当相关定时器通道设置为无需使用管脚的软件定时器时，此功能典型应用于暂时禁止一个输入捕获通道，或使得定时器管脚成为可用的通用I/O管脚。当TPMCH0管脚用作外部时钟输入时，对于通道0，这也是必须设置的。

表 10-5. 模式、边沿和电平选择

CPWMS	MSnB:MSnA	ELSnB:ELSnA	模式	配置	
X	XX	00		不用于TPM通道的管脚；作为TPM的外部时钟或者返回为通用I/O	
0	00	01	输入捕获	仅在上升沿捕获	
		10		仅在下降沿捕获	
		11		在上升或下降沿捕获	
	01	00	输出比较	仅软件比较	
				01	连接输出到比较
				10	清除输出到比较
11				设置输出到比较	
1X	10	边沿对齐PWM	高为真脉冲(清除输出到比较)		
			X1	低为真脉冲(设置输出到比较)	
1	XX	10	中心对其PWM	高为真脉冲(清除输出到向上比较)	
				X1	低为真脉冲(设置输出到向上比较)

在变成输入捕获模式前，如果相关端口管脚在至少两个总线时钟周期内不稳定，那可能是得到了一个边沿触发的意外指示。典型的，在改变通道配置位之后，且在使能通道中断或

使用状态标志之前，程序将清除状态标志以避免出现任何意外行为。

10.4.5 定时器 x 通道数值寄存器 (TPMxCnVH:TPMCnVL)

这些读/写寄存器包括输入捕获功能的捕获 TPM 计数器数值，输出比较或 PWM 功能时的输出比较值。通过复位可清除通道数值寄存器。

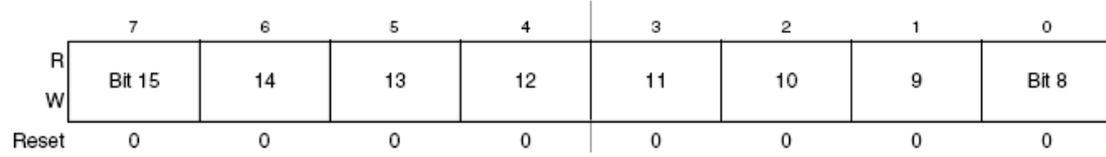


图 10-9. 定时器 x 通道数值寄存器高 (TPMxCnVH)

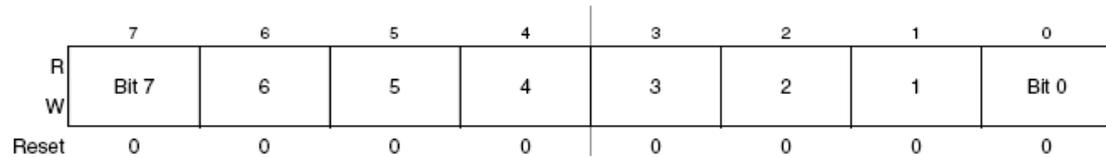


图 10-10. 定时器 x 通道数值寄存器低 (TPMxCnVL)

在输入捕获模式下，读取任一字节 (TPMCnVH 或 TPMCnVL) 可将两个字节中的内容都锁存至缓存器内，在缓存器中，这两个字节将保持锁存状态直到另一个字节也被读取。当 TPMxCnSC 寄存器被写操作时，该锁存机制也将复位 (成为非锁存状态)。

在输出比较或 PWM 模式下，写任一字节 (TPMxCnVH 或 TPMCnVL) 可将数值锁存至缓存器内。当两个字节都已被写入时，它们作为一个相连的 16 位数值传送到定时器通道数值寄存器内。该锁存机制可通过写 TPMCnSC 寄存器进行手控复位。

该锁存机制允许相连 16 位数据的两个字节以任何一种顺序写入，这有利于各种编译器实现。

10.5 功能描述

所有的 TPM 功能都与一个主 16 位计数器相关，该计数器允许对时钟源和预定标因子进行灵活选择。在 TPM 内，一个 16 位模寄存器也与该主 16 位计数器关联。每个 TPM 通道可随意与一个 MCU 管脚和一个可屏蔽的中断功能关联。

TPM 具有中心排列 PWM 性能，由 TPMSC 内的 CPWMS 控制位进行控制。当 CPWMS 被设置为 1 时，计时器 TPMCNT 变为一个增/减计数器，且所有关联 TPM 内的通道成为中心排列 PWM 通道。当 CPWMS=0 时，每个通道可被独立配置为输入捕获、输出比较或缓冲的中心排列 PWM 模式。

接下来的一节内容描述了该主 16 位计数器和每种定时器运行模式 (输入捕获、输出比较边缘排列 PWM 和中心排列 PWM)。因为详细的管脚行为和中断行为取决于运行模式，上述专题将在相关模式小节中进行描述。

10.5.1 计数器

所有的定时器功能都基于该主 16 位计数器 (TPMCNTH:TPMCNTL)。本节讨论时钟源的选择、增计数与增/减计数、计数结束溢出和手动计数器复位。

在任意 MCU 复位后，CLKSB:CLKSA=0:0，因此没有选择时钟源，TPM 为非活动状态。通常，CLKSB:CLKSA 会被设置为 0:1，由总线时钟驱动计时器。TPM 的时钟源可选为关闭、总线时钟 (BUSCLK)、固定的系统时钟 (XCLK)、或一个外部输入。可选外部时钟的

最大频率为总线速率的 1/4。有关时钟源选择的更多信息，参照 10.4.1 节“定时器 x 状态和控制寄存器 (TPMxSC)”，和表 10-2。

当微控制器处于后台活动状态时，TPM 会暂时中止所有计数直到微处理器返回正常的用户操作模式。在处于停止模式期间，所有的 TPM 时钟都被停止；因此，TPM 被有效禁用直到时钟恢复。在等待模式期间，TPM 正常运行。

该主 16 位计数器有两种计数方式。当选择中心排列 PWM 模式 (CPWMS=1) 时，计数器以增/减计数模式运行。否则，该计数器为简单的增计数器。作为增计数器，该主 16 位计数器从 0x0000 开始计数，一直计数到终止计数值，然后又从 0x0000 开始计数。终止计数值是 0xFFFF 或 TPMMODH:TPMMODL 中的模值。

当指定了中心排列 PWM 运行方式时，计数器从 0x0000 向上计数至终止计数值，然后又向下计数至 0x0000，在值 0x0000 时返回向上计数。0x0000 和终止计数值 (TPMMODH:TPMMODL 中的值) 都是正常长度计数 (一个定时器时钟周期长度)。

有一个中断标志和使能标志与该主 16 位计数器关联。定时器溢出标志是软件可访问的，用于指示计时器益处。使能信号在软件轮询 (TOIE = 0) 与中断驱动运行方式之间进行选择，软件轮询时没有产生硬件中断，而中断驱动运行时只要 TOF 标志为 1，即可自动产生一个静态硬件中断。

使得 TOF 置位的条件取决于计数方式 (增或增/减)。在增计数方式下，主 16 位计数器从 0x0000 一直计数至 0xFFFF，溢出时在下一计数时钟又从 0x0000 开始计数。计数值从 0xFFFF 转变为 0x0000 时，TOF 置位。当设置了一个模值上限时，计数值从模寄存器中的模值上限转变为 0x0000 时，TOF 置位。主 16 位计数器以增/减计数方式运行，当计数器从模寄存器中设置的值与其下一个更小的计数值之间改变计数方向时，TO 标志置位。这对应一个 PWM 周期的结束位置。(计数值 0x0000 对应一个周期的中心。)

因为 HCS08 的 MCU 是 8 位结构，相关机制内建于计时器用于读操作。只要计数器的任一字节 (TPMCNTH 或 TPMCNTH) 被读取，两个字节均被捕获至缓冲器中，因此，当另一字节被读取时，其值将代表第一个字节被读取时的计数值。计数器继续正常计数，但是两个字节中的新值都不能读取直到旧计数值的两个字节都已经被读取。

主定时计数器可以在任何时候，通过向定时器计数值的任何一个字节 TPMCNTH 或 TPMCNTH 中写入任意值，进行手动复位。用这种方式复位计数器也可以复位相关部分，以防在复位计数值前仅有计数器的一个字节被读取。

10.5.2 通道模式选择

如果 CPWMS=0 (未指定中心排列 PWM 运行模式)，通道 n 状态和控制寄存器中的 MSnB 和 MSnA 控制位决定相应通道运行的基本方式。可选方式包括；输入捕获、输出比较和缓冲边沿排列 PWM。

10.5.2.1 输入捕获模式

输入捕获功能下，TPM 可以捕获外部事件发生的时刻。当一个活动边沿出现在输入捕获通道管脚时，TPM 将 TPM 计数器的内容锁存至通道数值寄存器 (TPMCnVH:TPMCnVL)。上升沿、下降沿或任意边沿可被选为活动边沿用于触发输入捕获。

当 16 位捕获寄存器的两个字节中的任何一个被读取时，两个字节都会被锁存至缓冲器内以支持相关 16 位访问而无需考虑访问顺序。关联顺序可通过写通道状态/控制寄存器 (TPMCnSc) 操作进行手动复位。

一个输入捕获事件可设置一个标志位 (CHnF)，该标志位可随意产生一个 CPU 中断请求。

10.5.2.2 输出比较模式

输出比较功能时，TPM 可产生计时脉冲，这些计时脉冲是位置、极性、持续时间和频率可编程的。当计数器达到一个输出比较通道的通道数值寄存器内的值时，TPM 能设置、清除或连接通道管脚。

在输出比较模式下，只有在一个 16 位寄存器的两个 8 位字节都被写入后，数值才会被转移至相应的定时器通道数值寄存器。这一关联顺序操作可通过写通道状态/控制寄存器（TPMCnSn）进行手动复位。

一个输出比较事件可设置一个标志位（CHnF），该标志位可随意产生一个 CPU 中断请求。

10.5.2.3 边沿排列 PWM 模式

这一 PWM 输出类型使用计时器的正常增计数模式（CPWMS = 0），且当同一 TPM 中的其它通道配置为输入捕获或输出比较功能时，可用。该 PWM 信号的周期由模寄存器（TPMMODH:TPMMODL）的设置情况确定。占空系数由定时器通道数值寄存器（TPMCnVH:TPMCnVL）的设置情况确定。PWM 信号的极性由 ELSnA 控制位的设置情况确定。占空系数的可能值在 0% 与 100% 之间。

如图 10-11 所示，TPM 通道寄存器中的输出比较数值确定了 PWM 信号的脉冲宽度（占空系数）。溢出和输出比较之间的时间是脉冲宽度。如果 ELSnA=0，计数器溢出强制 PWM 信号为高电平，输出比较强制 PWM 信号为低电平。如果 ELSnA=1，计数器溢出强制 PWM 信号为低电平，输出比较强制 PWM 信号为高电平。

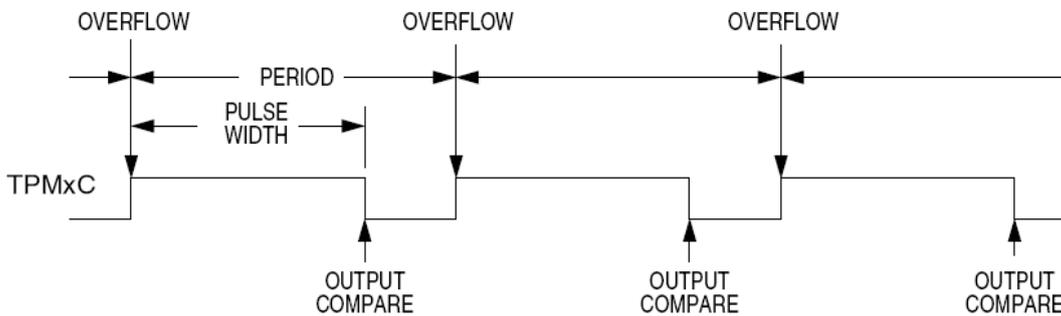


图 10-11. PWM 周期和脉宽（ELSnA = 0）

当通道数值寄存器设置为 0x0000 时，占空系数为 0%。通过将定时器通道数值寄存器（TPMCnVH:TPMCnVL）设置为一个大于所设模值的数值，可使占空系数达到 100%。这表示，为了获得 100% 的占空系数，设置的模值必须小于 0xFFFF。

因为 HCS08 是 8 位 MCU 系列，定时器通道寄存器的设置被缓存起来，以确保连续 16 位数据更新，并避免出现意外的 PWM 脉冲宽度。写 TPMCnVH 或 TPMCnVL 中的任意一个寄存器，也就是写缓冲寄存器。在边沿排列模式下，只有在一个 16 位寄存器的两个 8 位字节都被写入后，且 TPMCNTH:TPMCNTL 计数器中的值为 0x0000，计数值被转移至相应的定时器通道寄存器。（直到下一个整周期新的占空系数才有效。）

10.5.3 中心排列 PWM 模式

此类 PWM 输出使用计时器的增/减计数模式（CPWMS=1）。TPMCnVH:TPMCnVL 中的输出比较值决定 PWM 信号的脉冲宽度（占空系数），周期由 TPMMODH:TPMMODL 中的值确定。

TPMMODH:TPMODL 中的值应该保持在范围 0x0001 至 0x7FFF 内，因为在此范围外的值会产生模糊的结果。ELSnA 将确定 CPWM 输出的极性。

$$\text{pulse width} = 2 \times (\text{TPMCnVH}:\text{TPMCnVL}) \quad \text{Eqn. 16-1}$$

$$\text{period} = 2 \times (\text{TPMMODH}:\text{TPMODL});$$

$$\text{for TPMMODH:TPMODL} = 0\text{x}0001\text{--}0\text{x}7\text{FFF} \quad \text{Eqn. 16-2}$$

如果通道数值寄存器 TPMCnVH:TPMCnVL 为 0 或负值（第 15 位被置位），占空系数将为 0%。如果 TPMCnVH:TPMCnVL 为正值（第 15 位被清 0）且大于设置的模值（非零），占空系数为 100%，因为不会发生占空系数比较。这表明，通过模寄存器设置的可用周期范围在 0x0001 与 0x7FFE（如果不必产生 100% 的占空系数可为 0x7FFF）之间。这不是一个重要的限制，因为结果周期远远比正常应用所需要的周期长。

TPMMODH:TPMODL=0x0000 为一不能被用于中心排列 PWM 模式的特殊情况。当 CPWMS=0 时，这种情况对应计数器从 0x0000 计数至 0xFFFF；但是当 CPWMS=1 时，计数器需要一个与模寄存器匹配的有效值，用于在非 0x0000 的某一位置从增计数转向为减计数。

图 10-12 所示为 TPM 通道寄存器中的输出比较值（乘以 2），该值确定了 CPWM 信号的脉冲宽度（占空系数）。如果 ELSnA=0，当向上计数强制 CPWM 输出信号为低时或者当向下计数强 CPWM 输出信号为高时，比较得到匹配。计数器向上计数至 TPMMODH:TPMODL 中设置的值，然后向下计数至 0。这将周期设置等于 TPMMODH:TPMODL 的两倍。

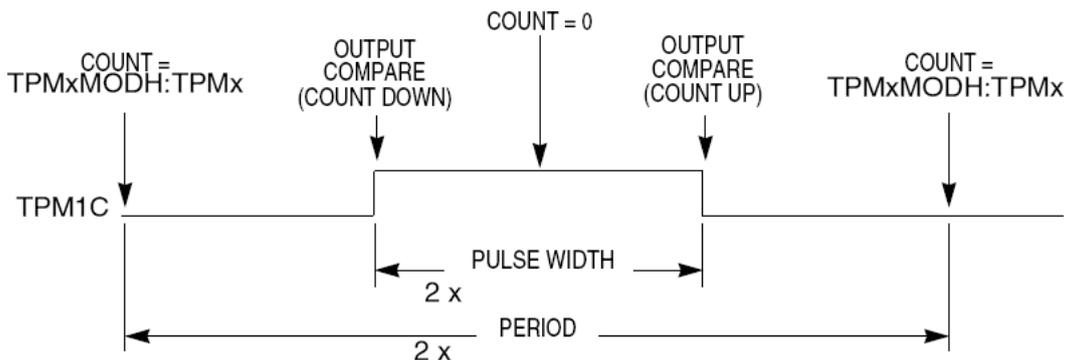


图 10-12. CPWM 周期和脉冲宽度 (ELSnA = 0)

中心排列 PWM 输出产生的噪声小于各种边沿排列 PWM，因为在同一系统时钟边沿，排列的 I/O 传送管脚更少。对于一些类型的电动机，此类 PWM 也是需要的。

因为 HCS08 为 8 位 MCU 系列，定时器通道寄存器中的设置被缓存下来，以保证连续 16 位数据更新，并避免意外的 PWM 脉冲宽度出现。对寄存器 TPMMODH、TPMODL、TPMCnVH 和 TPMCnVL 中的任意一个进行写操作，实际是对缓冲寄存器进行写操作。

只有在一个 16 位寄存器的两个 8 位字节都被写入后，数值被传送至对应的定时器通道寄存中，且计时器溢出（在模寄存器中终止计数值的结尾，由增计数转向减计数）。此 TPMCNT 溢出需要仅应用于 PWM 通道，而不应用于输出比较。

另外，当 TPMCNTH:TPMCNTL=TPMMODH:TPMODL 时，TPM 能在此次计数的结尾产生一个 TOF 中断。用户可以选择重载任意数量的 PWM 缓冲器，这些缓冲器将在一个新周期开始时同时更新。

写 TPMSC 可取消写入 TPMMODH 和/或 TPMODL 中的任意值，并为模寄存器中的关联装置复位。写 TPMCnSC 可取消写入通道数值寄存器中的任意值，并为 TPMCnVH:TPMCnVL 中的关联装置复位。

10.6 TPM 中断

TPM 为主寄存器一处产生一个可选中断，并为每个通道产生一个中断。通道中断的意义取决于每个通道采用的运行模式。如果通道配置为输入捕获，每当所选输入捕获边沿被识别时，中断标志被置位。如果通道配置为输出比较或 PWM 模式，每当主计时器与 16 位通道数值寄存器中的值匹配时，中断标志被置位。对于绝对中断向量地址、极性和局部中断屏蔽控制位，见“复位、中断和系统配置”一章。

对 TPM 中的每个中断源，当中断条件，如定时器溢出、通道输入捕获或输出比较事件，被识别时，标志位被置位。该标志位可通过软件读取（查询）以验证操作是否已经发生，或者，置位一个相关使能位（TOIE 或 CHnIE）来使能硬件中断产生。当中断使能位被置位时，只要相关中断标志等于 1，就将会产生一个静态中断。从中断服务程序返回前，执行一序列用以清除中断标志的步骤是用户软件需要完成的。

10.6.1 清除定时器中断标志

TPM 中断标志清除需要两个步骤：首先，当标志位被置位为 1 时，对标志位进行读操作；然后，向该标志位写 0。如果在这两个步骤之间检测到一个新事件，操作重新执行，且在第二步之后中断标志仍保持置位，以避免可能丢失新事件。

10.6.2 定时器溢出中断描述

引起 TOF 置位的条件取决于计数模式（增或增/减）。在增计数模式下，16 位计时器从 0x0000 计数至 0xFFFF，溢出后，在下一个计数时钟又从 0x0000 开始计数。计数至从 0xFFFF 转变为 0x0000 时，TOF 置位。当设置了一个模界限时，在计数值从模寄存器中设置的模限值转变为 0x0000 时，TOF 置位。计数器以增/减计数模式运行时，当计数器从模寄存器中设置的值和下一个更低的计数值之间转向时，TOF 标志置位。这对应一个 PWM 周期结束。（计数值 0x0000 对应周期中心。）

10.6.3 通道事件中断描述

通道中断的意义取决于通道的当前模式（输入捕获、输出比较、边沿排列 PWM 或中心排列 PWM）。

当一个通道被配置为输入捕获通道时，ELSnB:ELSnA 控制位选择上升沿、下降沿、任意边沿、或无边沿（关闭）作为触发一个输入捕获事件的边沿。当所选边沿被检测到时，中断标志被置位。该标志可通过 2 个步骤清除，见 10.6.1 节，“清除定时器中断标志”。

一个通道被配置为输出比较通道时，每当主计时器与通道数值寄存器中的值匹配时，中断标志被置位。该标志可通过 2 个步骤清除，见 10.6.1 节，“清除定时器中断标志”。

10.6.4 PWM 工作周期结束事件

对于配置为 PWM 操作的通道，存在两种可能：

- 通道配置为边沿排列 PWM 时，当计时器与屏蔽当前工作循环周期结束的通道数值寄存器匹配时，通道标志被置位。
- 通道配置为中心排列 PWM 时，在每个 PWM 周期内，计时器与通道数值寄存器匹配两次。在这种 PWM 情况下，在当前工作循环的开始和结束时，也就是在计时器与通道数值寄存器匹配的时候，通道标志被置位。

该标志可通过 2 个步骤清除，见 10.6.1 节，“清除定时器中断标志”。

第十一章 串行通信接口(S08SCIV2)

11.1 介绍

MC9S08AW60 系列包括两个独立的串行通信接口(SCI)的模块，这有时也称为通用异步接收/发射器(UARTs)。通常情况下，这些系统主要是用于连接到个人电脑或工作站的 RS232 串行输入/输出(I/O)的端口，但也可以用来与其他嵌入式控制器通信。

一个灵活的 13 位以模为基础的波特率发生器支持范围广泛的标准波特率超过 115.2 kbaud。在同一个 SCI 中发送和接收使用共同的波特率，每个 SCI 模块有一个单独的波特率发生器。

SCI 系统提供了许多先进的功能并不普遍发现在其他嵌入式控制器的其他异步串行 I/O 外设。接收器采用先进的数据采集技术确保可靠的通信和噪声检测。硬件平价、接收唤醒和双缓冲对发送和接收也包括在内。

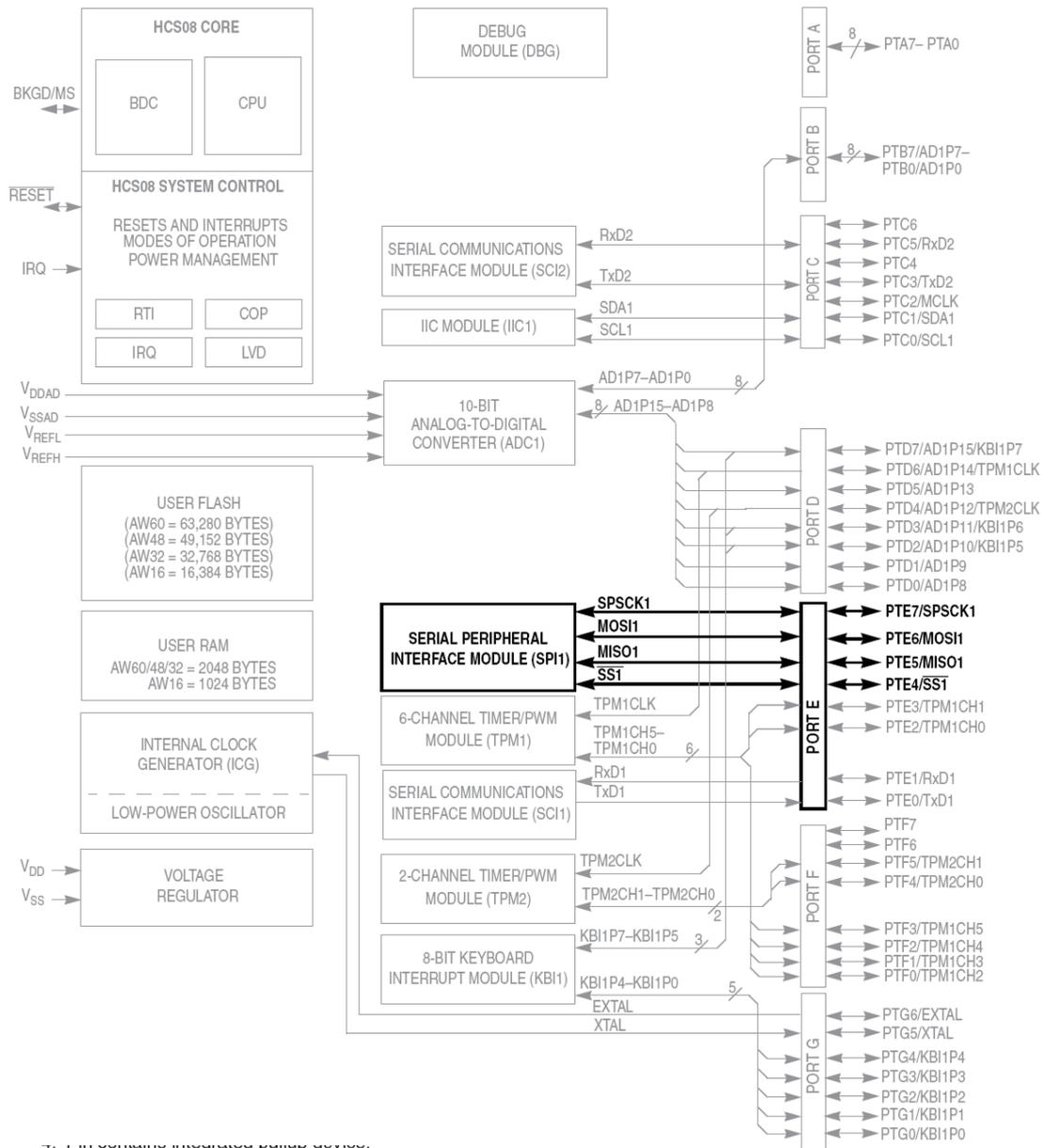


图 11-1. MC9S08DZ60 结构图

注意:

1. 如果是输入端口，端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用($IRQEDG=1$)，引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测，下拉式启用($IRQPE=1$)。
3. IRQ 没有一个二极管到 V_{DD} ，IRQ 不应该超过 V_{DD} 。
4. 引脚包含完整的上拉装置。
5. 引脚 PTD7，PTD3，PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用($KBIPE_n=1$)并选定上升沿($KBEDG_n=1$)时，下拉式启用时。

11.1.1 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

11.1.2 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 11.3, “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

11.1.3 结构图

图 11-2 显示了 SCI 的发射器部分。

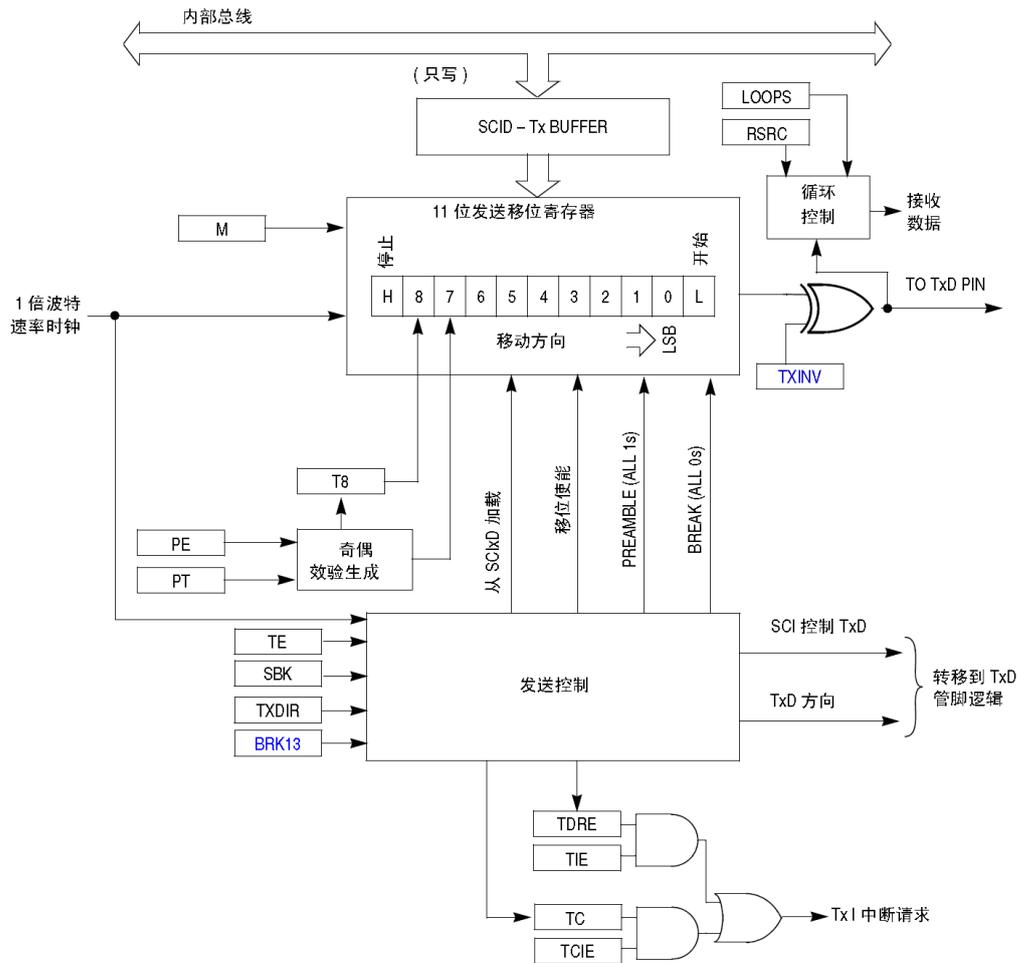


图 11-2 SCI 发射器结构图

图 11-3 SCI 接收器结构图。

[SBR12:SBR0], 首先写入 SCIxBDH, 缓存生成的高位字节, 然后再写入 SCIxBDL。SCIxBDH 中的值在写入 SCIxBDL 前不会变化。

SCIxBDL 复位为非零值, 因此复位后, 波特率发生器一直保持禁止, 直到第一次使能接收器或发射器 (SCIxC2 中的 RE 或 TE 位写为 1)。

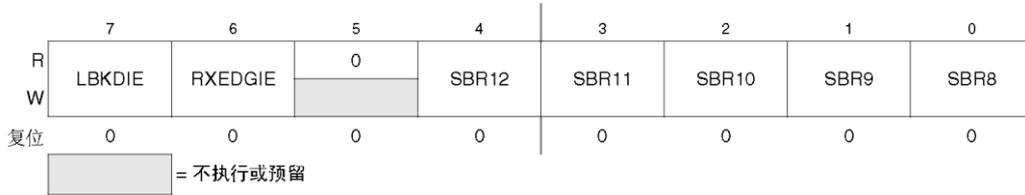


图 11-4. SCI 波特率寄存器(SCIxBDH)

表 11-1 SCIxBDH 字段描述

字段	描述
4:0 SBR[12:8]	波特率模数除数 — SBR[12:0] 中的13个位统称为BR, 它们为SCI 波特率发生器设置模数除数系数。当BR =0, SCI 波特率发生器禁止, 以降低电源电流。当BR = 1~8191时, SCI波特率= BUSCLK/ (16xBR)。也可参见中的BR位。表11-2中的BR位。

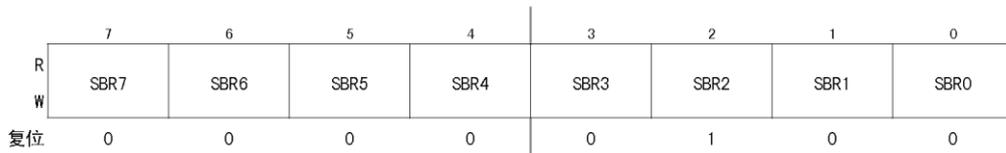


图 11-5 波特率寄存器 (SCIxBDL)

表 11-2 SCIxBDL 字段描述

字段	描述
4:0 SBR[12:8]	波特率模数系数 — SBR[12:8] 中的这13 个位统称为BR, 它们为SCI 波特率发生器设置模数除数系数。当BR= 0, SCI 波特率发生器禁止, 以降低电源电流 。当BR = 1~8191时, SCI 波特率= BUSCLK/ (16xBR) 。也可参见表11-1中的BR 位。

11.2.2 SCI 控制寄存器 1(SCIxC1)

该读/ 写寄存器用于控制 SCI 系统的各种可选功能。

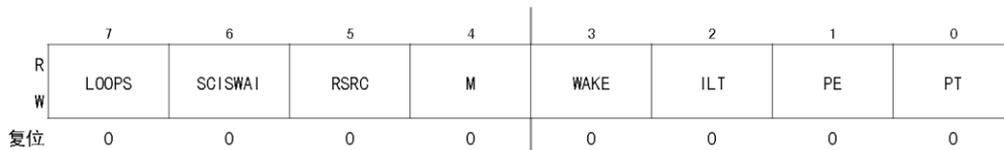


图 11-6. SCI 控制寄存器 1 (SCIxC1)

表 11-3. SCIxC1 字段描述

字段	描述
7 LOOPS	循环模式选择 —在环回模式和正常的2 管脚全双工模式之间选择。当LOOPS = 1, 发射器输出内部连接到接收器输入。

	<p>0 正常运行 — RxD 和TxD 使用独立管脚。</p> <p>1 循环模式或单线模式，发射器输出内部连接到接收器输入（见<st-blue>RSRC 位）。SCI 不使用 RxD 管脚。</p>
6 SCISWAI	<p>等待模式中的SCI 停止</p> <p>0 SCI 时钟继续在等待模式中运行，因此SCI 可以是唤醒CPU 的中断源。</p> <p>1 SCI 时钟在 CPU 处于等待模式时冻结。</p>
5 RSRC	<p>接收器源选择—该位没有任何意义或影响，除非LOOPS 位设置为1。当LOOPS = 1 时，接收器输入内部连接到TxD 管脚， RSRC 决定该连接是否也连接到发射器输出。</p> <p>0 假设LOOPS = 1， RSRC = 0 选择内部环回模式， SCI 不使用RxD 管脚。</p> <p>1 单线 SCI 模式，其中 TxD 管脚连接到发射器输出和接收器输入。</p>
4 M	<p>9位或8位模式选择</p> <p>0 正常 — 启动+ 8个数据位(LSB 先发)+停止</p> <p>1 接收器和发射器使用 9 位数据字符启动+ 8 个数据位(LSB 先发)+第 9 个数据位+停止</p>
3 WAKEI	<p>接收器唤醒方法选择—详情请见11.3.3.2，“接收器唤醒操作”</p> <p>0 闲置线路唤醒</p> <p>1 地址标记唤醒</p>
2 ILT	<p>闲置线路类型选择—将该位设置为1，确保字符末端的停止位和逻辑1位不会计数闲置线路检测逻辑所需的逻辑高电平的10或11个位时间。如需了解更多信息，14.3.3.2.1，“闲置线路唤醒”</p> <p>0 开始位后闲置字符位计数开始。</p> <p>1 停止位后闲置字符位计数开始。</p>
1 PE	<p>奇偶效验使能—使能硬件奇偶效验生成和检查。当使能奇偶效验时，数据字符(第8或第9数据位)的最高位(MSB)视为奇偶校验位。</p> <p>0 无硬件奇偶效验生成或检查。</p> <p>1 奇偶效验使能。</p>
0 PT	<p>奇偶效验类型—如果使能奇偶效验(PE = 1)，该位选择奇或偶效验。奇效验表示数据字符中1的总数(包括奇偶校验位)是奇数。偶数表示数据字符中1的总数(包括奇偶校验位)是偶数。</p> <p>0 偶效验</p> <p>1 奇效验</p>

11.2.3 SCI 控制寄存器 2 (SCIxC2)

该寄存器可以随时读取或写入。

	7	6	5	4	3	2	1	0
R								
W	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
复位	0	0	0	0	0	0	0	0

图 11-7. SCI 控制寄存器 2 (SCIxC2)

表 11-4. SCIx2 字段说明

字段	描述
7 TIE	发送中断使能(用于TDRE) 0 来自TDRE 的硬件中断禁止(使用轮询)。 1 当TDRE 标记为1 时允许硬件中断。
6 TCIE	发送完成中断使能(用于TC) 0 来自TC 的硬件中断禁止(使用轮询)。 1 当 TC 标记为 1 时允许硬件中断。
5 RIE	接收器中断使能(用于RDRF) 0 来自RDRF 的硬件中断禁止 (使用轮询)。 1 当 RDRF 标记为 1 时允许硬件中断。
4 ILIE	闲置线路中断使能 (用于IDLE) 0 来自IDLE 的硬件中断禁止 (使用轮询)。 1 当 IDLE 标记为 1 时允许硬件中断。
3 TE	发射器使能 0 发射器关闭。 1 发射器打开。 要使用SCI 发射器， TE 必须是1。当TE = 1 时， SCI 强制TxD 管脚作为SCI 系统的输出。 当SCI 配置用于单线运行 (LOOPS = RSRC = 1) 时， TXDIR 控制单SCI 通信线路 (TxD 管脚) 上的流量方向。 当正在进行发送时通过写TE = 0， 然后写TE=1， TE 也可以用来排队闲置字符。详情请 14.3.2.1， “发送中断和排队闲置”。 当TE 写入0， 发射器保持对端口TxD 管脚的控制， 直到任何数据、 队列闲置或队列中止字符在允许管脚恢复为通用I/O 管脚前完成传输。
2 RE	接收器使能 — 当SCI 接收器关闭时， RxD 管脚恢复为通用端口I/O 管脚。如果 LOOPS = 1， RxD 管脚恢复为通用I/O 管脚， 即使RE = 1。 0 接收器关闭。 1 接收器打开。
1 RWU	接收器唤醒控制 — 该位可以写入1， 将SCI 接收器置于待机状态， 等待所选唤醒条件的自动硬件检测。唤醒条件既可以是信息间的闲置线路 (WAKE = 0， 闲置线路唤醒)， 也可以是某个字符中最高数据位中的逻辑1 (WAKE = 1， 地址标记唤醒)。应用软件设置RWU， (正常情况下) 且所选的硬件条件自动清除RWU。如需了解更多信息， 11.3.3.2， “接收器唤醒操作”。 0 正常的SCI 接收器运行。 1 处于待机状态的 SCI 接收器等待唤醒条件。
0 SBK	发送中止字符 — 先将1 和0 写入SBK， 即在发送数据流中排入了一个中止字符。只要SBK=1， 多达10 或11 (如果BRK13 = 1， 则为13 或14 位) 位时间的逻辑0中止字符被加入队列。根据当前正在发送信息有关的SBK 的设置和清除时间， 第二个中止字符可以在软件清除SBK 前排队。如需了解更多信息， 11.3.2.1， “发送中断和排队闲置”。 0 正常的发射器运行。

	1 将发送的队列中止字符。
--	---------------

11.2.4 SCI 状态寄存器 1 (SCIxS1)

该寄存器有 8 种只读状态标记。写没有影响，特殊软件顺序（不包括写入该寄存器）用来清除这些状态标记。

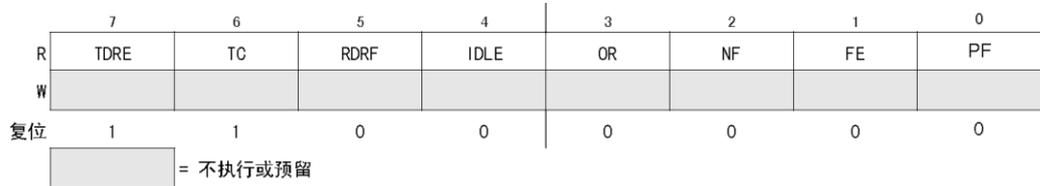


图 11-8. SCI 状态寄存器 1(SCIxS1)

表 11-5. SCIxS1 字段描述

字段	描述
7 TDRE	<p>发送数据寄存器空标记 — TDRE 设置于复位，当发送数据值从发送数据缓冲器传输到发送移位器时，就在缓冲器中为新字符留出空间。要清除TDRE，当TDRE = 1时读SCIxS1，然后写入SCI 数据寄存器（SCIxD）。</p> <p>0 发送数据寄存器（缓冲器）已满。 1 发送数据寄存器（缓冲器）为空。</p>
6 TC	<p>发送完成标记 — TC 设置于复位，当TDRE = 1 时，无数据、前导信号或中止字符在发送。</p> <p>0 发射器活动（发送数据、前导信号或中止字符）。 1 发射器闲置（发送活动完成）</p> <p>当TC = 1 时读取SCIxS1 时可以自动清除TC，然后进行以下三种操作中一种：</p> <ul style="list-style-type: none"> ● 写入SCI 数据寄存器（SCIxD），以发送新数据 ● 通过把TE 从0 变为1，排队前导信号 ● 将1 写入 SCIxC2 中的 SBK，排队中止字符。
5 RDRF	<p>接收数据寄存器已满标记 — 当字符从接收移位器传输到接收数据寄存器（SCIxD）时，设置RDRF。要清除RDRF，当RDRF = 1 时读SCIxS1，然后读SCI 数据寄存器（SCIxD）。</p> <p>0 接收数据寄存器空。 1 接收数据寄存器已满。</p>
4 IDLE	<p>闲置线路标记— 在一段时间的活动后，当SCI 接收线路已经闲置了一个全字符时间时，就设置IDLE。当ILT =0，接收器在起始位后开始计数闲置位时间。因此，如果接收字符都为1，这些位时间和停止位时间计数入接收器用于探测一个闲置线路所需逻辑高态（10 或11 个位时间，取决于M 控制位）的全字符时间。当ILT = 1，接收器直到停止位后才开始计数闲置位时间。因此，停止位和前一字符末端的任何逻辑高态位时间不会计数入接收器用于探测一个闲置线路所需逻辑高态的全字符时间。</p> <p>要清除IDLE，当IDLE = 1 时读取SCIxS1，然后读取SCI 数据寄存器（SCIxD）。清除IDLE 后，不能再次进行设置，直到接收到新字符且已设置了RDRF。IDLE 只设置一次，即便接收线路闲置了很长一段时间。</p> <p>0 没有检测到闲置线路 1 检测到闲置线路</p>
3	<p>接收器溢出标记— 当新的串行字符做好了传输到接收数据寄存器（缓冲器）的准备时，但原来接收的字符还没有从SCIxD 读取，设置OR。在这种情况下，新字符（和</p>

OR	<p>所有相关错误信息) 丢失, 因为没有空间将它们移到SCIxD。要清除OR, 当OR = 1 时读SCIxS1, 然后读SCI 数据寄存器 (SCIxD)。</p> <p>0 没有溢出</p> <p>1 接收溢出 (新SCI 数据丢失)</p>
2 NF	<p>噪音标记— 接收器中采用的先进的采样技术在起始位中提取7 个样本, 在每个数据位和停止位中提取3 个样本。如果这些样本中任何一个样本与帧中任何位时间内的其余样本不一致, 就要在RDRF 为这个字符而置1 的同时设置标记NF。要清除NF, 读SCIxS1, 然后读SCI 数据寄存器 (SCIxD)。</p> <p>0 没有检测到噪音</p> <p>1 SCIxD中的已接收字符中检测到噪音</p>
1 FE	<p>成帧错误标记— 当接收器在应该是停止位的时候检测到逻辑0 时, 同时设置FE 和RDRF。这表示接收器与字符帧没有完全统一。要清除FE, 当FE = 1 时读SCIxS1, 然后读SCI 数据寄存器 (SCIxD)。</p> <p>0 未检测到成帧错误, 这不能保证成帧正确。</p> <p>1 成帧错误。</p>
0 PF	<p>奇偶效验错误标记— 当奇偶效验使能 (PE = 1) 且已接收字符中的奇偶校验位与预期奇偶效验值不一致时, 同时设置PF 和RDRF。要清除PF, 读SCIxS1, 然后读SCI 数据寄存器 (SCIxD)。</p> <p>0 没有奇偶效验错误</p> <p>1 奇偶效验错误</p>

11.2.5 SCI 状态寄存器 2 (SCIxS2)

该寄存器有一个只读状态标记。

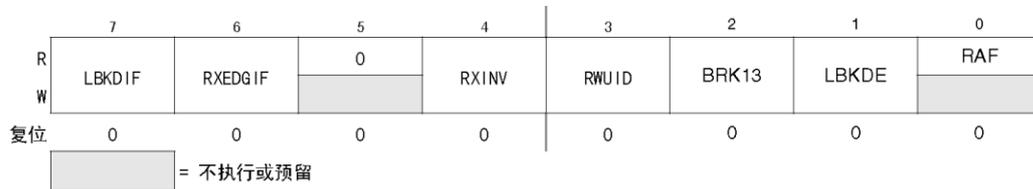


图 11-9. SCI 状态寄存器 2 (SCIxS2)

表 11-6. SCIxS2 字段描述

字段	描述
2 BRK13	<p>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。</p> <p>0 中止字符用10位时间 (如果M = 1, 则是11位时间) 长度发送</p> <p>1 中止字符用13位时间 (如果M = 1, 则是14位时间) 长度发送</p>
0 RAF	<p>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。</p> <p>0 SCI 接收器闲置, 正在等待起始位。</p>

1 SCI 接收器活动 (RxD 输入不闲置)。

11.2.6 SCI 控制寄存器 3 (SCIxC3)

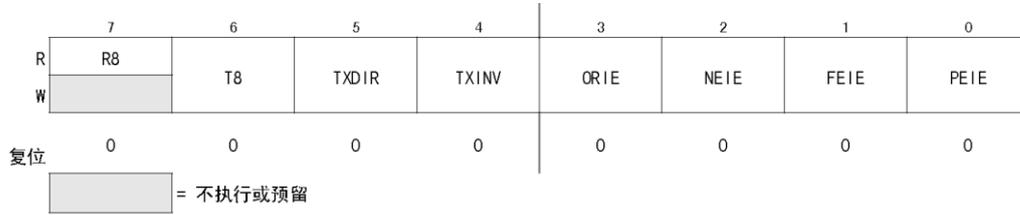


图 11-10. SCI 控制寄存器 3 (SCIxC3)

表 14-8. SCIxC3 字段描述

字段	描述
7 R8	接收器的第9个数据位 —当SCI 配置用于9 位数据 (M = 1) 时, R8可以视为SCIxD 寄存器中缓冲数据的MSB 左侧的第9 个接收数据位。读9位数据时, 读SCIxD前读取R8, 因为读SCIxD 能够完成自动的标记清除顺序, 允许R8和SCIxD被新数据覆盖。
6 T8	9个数据位发射器 —当SCI 配置用于9位数据 (M = 1) 时, T8可以视为SCIxD 寄存器中缓冲数据的MSB左侧的第9个接收数据位。写9 位数据时, 整个9位值在SCIxD 写入后被传输到SCI移位寄存器, 因此T8应在SCIxD写入前写入 (如果它需要从它的原来值中修改)。如果T8不需要在新值 (例如当它用于生成标记或空间奇偶效验) 中修改, 它就不需要在每次写SCIxD 时写入。
5 TXDIR	单线模式中的TxD管脚方向 —当SCI配置用于单线半双工运行 (LOOPS = RSRC = 1) 时, 该位决定TxD管脚上数据的方向。 0 TxD管脚是单线模式中的输入。 1 TxD管脚是单线模式中的输出。
4 TXINV1	发送数据反转 —设置该位反转已发送数据输出的极性。 0 发送数据未被反转 1 发送数据被反转
3 ORIE	溢出中断使能 —该位使能溢出标记 (OR) 以生成硬件中断请求。 0 OR中断禁止 (使用轮询) 1 当OR = 1 时允许硬件中断
2 NEIE	噪音错误中断使能 — 该位使能噪音标记 (NF) 以生成硬件中断请求。 0 NF中断禁止 (使用轮询) 1 当NF = 1 时允许硬件中断
1 FEIE	成帧错误中断使能 — 该位使能成帧错误标记 (FE) 以生成硬件中断请求。 0 NF中断禁止 (使用轮询)

	1 当NF = 1 时允许硬件中断
0 PEIE	奇偶效验错误中断使能 — 该位使能奇偶错误标记（PF）以生成硬件中断请求。 0 PF中断禁止（使用轮询） 1 当PF = 1 时允许硬件中断

¹ 设置 TXINV 会反转所有情况下的 TxD 输出：数据位、起始位和停止位、中止符、闲置。

11.2.7 SCI 数据寄存器 (SCIxD)

该寄存器实际上是两个独立寄存器。读返回只读接收数据缓冲器的内容，写进入只写发送数据缓冲器。该寄存器的读写还涉及 SCI 状态标记的自动标记清除机制。

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
复位	0	0	0	0	0	0	0	0

图 11-11. SCI 数据寄存器 (SCIxD)

11.3 功能描述

SCI 允许在 MCU 和远程器件（包括其他 MCU）间进行全双工、异步、NRZ 串行通信。SCI 由波特率发生器、发射器和接收时钟组成。发射器和接收器独立运行，尽管它们使用同一波特率发生器。在正常运行期间，MCU 监控 SCI 的状态，写将要发送的数据，处理已接收的数据。下面就简要地介绍一下 SCI 的每个块。

11.3.1 波特率生成

如图 11-12 所示，SCI 波特率发生器的时钟源是总线速率时钟。

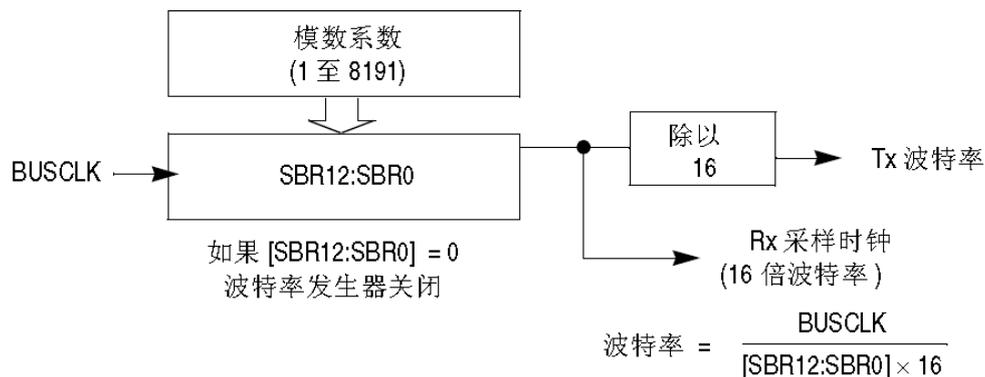


图 11-12. SCI 波特率生成

SCI 通信要求发射器和接收器（一般情况下从独立时钟源中获取波特率）使用相同的波特率。该波特率的允许容限取决于接收器与起始位的前沿以及位采样执行的具体情况。

每次从高态转换到低态时，MCU 都重新同步位边界，但在最坏的情况下，整个 10 或

11 位时间字符帧中没有这种转换，所以波特率中的任何不匹配都累积到整个字符时间。对于总线频率由晶振驱动的飞思卡尔半导体 SCI 系统，允许的波特率不匹配对 8 位数据格式来说大约为 4.5%、对 9 位数据格式来说大约 4%。尽管波特率模数除数设置不会永远生成与标准速率严格匹配的波特率，但一般情况下都在一个很小的百分比内，是可靠通信可以接受的。

11.3.2 发射器功能描述

本小节描述 SCI 发射器的整体结构图，以及发送中断和闲置字符的一些专用功能。发射器结构图如图 11-2。

发射器输出 (TxD) 闲置状态默认为逻辑高态 (复位后 TXINV = 0)。如果 TXINV = 1，发射器输出就被颠倒。通过在 SCIxC2 中设置 TE 位，发射器被使能。这会排队前导信号字符，前导信号字符是闲置状态的一个完整字符帧。发射器然后保持闲置状态，直到发送数据缓冲器中出现数据。通过把数据写入 SCI 数据寄存器 (SCIxD)，程序把数据保存到发送数据缓冲器。

SCI 发射器的中心元件是长度为 10 或 11 位 (取决于 M 控制位中的设置) 的发送移位寄存器。对于本小节的剩余部分，我们假设 M = 0，选择正常的 8 位数据模式。在 8 位数据模式中，移位寄存器中有 1 个起始位、8 个数据位和 1 个停止位。当发送移位寄存器可以用于新 SCI 字符时，在发送数据寄存器中等待的值被传输到移位寄存器 (与波特率时钟同步)，同时设置发送数据寄存器空 (TDRE) 状态标记，显示另外一个字符可以写入 SCIxD 的发送数据缓冲器。

如果停止位移出 TxD 管脚后发送数据缓冲器中没有新字符在等待，发射器设置发送完成标记，进入闲置模式，TxD 处于高态，等待发送更多字符。

将 0 写入 TE 不会立即释放管脚使其成为通用 I/O 管脚。正在进行的任何发送活动必须首先完成，这包括正在发送的数据字符、已进入队列的闲置字符和已进入队列的中止字符。

11.3.2.1 发送中断和排队闲置

SCIxC2 中的 SBK 控制位用来发送中止字符，中止字符最初用来引起旧式电传打字接收器的注意。中止字符是逻辑 0 (10 位时间，包括启动和停止位) 的全字符时间。13 位时间的较长中止符可以通过设置 BRK13 = 1 进行使能。一般来说，程序要等待 TDRE 进行设置，以显示信息的最后一个字符已经移动到发送移位器，然后依次把 1 和 0 写入 SBK 位。一旦移位器可用，该操作就立即对将发送的中止字符进行排队。如果当已进入队列的中止符进入移位器 (与波特率时钟同步) 时 SBK 仍然为 1，额外的中止字符会进入队列。如果接收器件是另一个飞思卡尔半导体 SCI，中止字符将作为所有 8 个数据位中的 0 进行接收，并出现成帧错误 (FE = 1)。

当使用闲置线路唤醒时，信息之间就需要闲置 (逻辑 1) 的全字符时间，以唤醒正处于睡眠状态的任何接收器。在正常情况下，程序会等待 TDRE 进行设置，显示信息的最后字符已经移动到发送移位器，然后依次把 0 和 1 写入 TE 位。一旦移位器可用，该操作就立即对将发送的闲置字符进行排队。只要 TE = 0 时移位器中的字符没有完成，SCI 发射器永远不会真正放弃 TxD 管脚的控制。如果移位器在 TE = 0 时有完成的可能，则设置通用 I/O 控制，这样与 TxD 共享的管脚就是驱动逻辑 1 的输出。这确保了 TxD 线路看起来像是正常闲置线路，即便在向 TE 写入 0 和 1 的过程中 SCI 失去对端口管脚的控制。

表 11-8. 中止字符长度

BRK13	M	中止字符长度

0	0	10 位时间
0	1	11 位时间
1	0	13 位时间
1	1	14 位时间

11.3.3 接收器功能描述

在本小节，首先把接收器结构图(图 11-3)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 11.3.5.1，“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 11.3.4，“中断和状态标记”。

11.3.3.1 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。

11.3.3.2 接收器唤醒操作

接收器唤醒是一种硬件机制，允许 SCI 接收器忽略用于不同 SCI 接收器的信息中的字符。在这种系统中，所有接收器都估计每条信息的第一个字符，一旦确定该信息旨在用于不同接收器，它们就立即将逻辑 1 写入 SCIxC2 中的接收器唤醒 (RWU) 控制位。当设置了 RWU 位时，禁止设置与接收器有关的状态标记（当设置了 RWUID 位时，闲置位 IDLE 除外），因此消除了处理不重要信息字符的软件开销。在信息结束或在下一条信息开始时，所有接收器自动强制 RWU 清零，这样所有接收器及时唤醒，以查看下一条信息的首字符。

11.3.3.2.1 闲置西安路唤醒

当 WAKE = 0 时，接收器配置用于闲置线路唤醒。在该模式中，当接收器检测到闲置线路级的某个全字符时间时，RWU 被自动清除。M 控制位选择 8 位或 9 位数据模式确定构成全字符时间所需的闲置位时间（10 或 11 位时间，由于起始和停止位）。

当 RWU 为 1、RWUID 为 0 时，唤醒接收器的闲置条件不会设置 IDLE 标记。接收器唤醒并等待下一条信息的第一个数据字符，这将设置 RDRF 标记并生成中断（如使能的话）。当 RWUID 为 1 时，任何闲置条件都设置闲置标记并生成中断（如使能的话），无论 RWU 是 0 还是 1。

闲置线路类型 (ILT) 控制位选择以下两种方式中的一种来检测闲置线路。当 ILT = 0 时，闲置位计数器在起始位后启动，这样停止位和字符末端的任何逻辑 1 计数闲置的全字符时间。当 ILT = 1 时，闲置位计数器直到停止位时间结束后才启动，这样闲置检测不受上一条信息的最后一个字符中的数据的影响。

11.3.3.2.2 地址标记唤醒

当 WAKE = 1 时，接收器配置用于地址标记唤醒。在该模式中，当接收器检测到已接收字符的最高位（在 M = 0 模式中是第 8 个位；在 M = 1 模式中是第 9 个位）中的逻辑 1 时，RWU 被自动清除。

地址标记唤醒允许信息包含闲置字符，但要求预留 MSB，以便在地址帧中使用。在收到停止位前，地址帧中 MSB 的逻辑 1 会清除 RWU 位，并设置 RDRF 标记。在这种情况下，会收到设置了 MSB 的字符，即便接收器在该字符时间的大部分时间中处于睡眠状态。

11.3.4 中断和状态标记

SCI 系统有三种独立的中断向量，以减少隔离中断原因所需的软件数量。一个中断向量与 TDRE 和 TC 事件的发射器相关，一个中断向量与 RDRF、IDLE、RXEDGIF 和 LBKDIF 事件的接收器相关，第三个向量用于 OR、NF、FE 和 PF 错误情况。这 10 个中断源的每个都可以由本地中断使能分别进行屏蔽。当清除本地使能以禁止生成硬件中断请求时，标记仍然可以用软件进行轮询。

SCI 发射器有两种状态标记，它们都可以生成硬件中断请求。发送数据寄存器空 (TDRE) 显示发送数据缓冲器何时空间将其他发送字符写入 SCIxD。如果设置了发送中断使能 (TIE) 位，每当 TDRE = 1 时都请求硬件中断。发送完成 (TC) 表示发射器完成发送所有数据、前导信号和中止字符，且它处于闲置状态，TxD 不活动。该标记通常用于带有调制解调器的系统，以决定何时可以安全关闭调制解调器。如果设置了发送完成中断使能 (TCIE) 位，每当 TC = 1 时请求硬件中断。如果相应 TIE 或 TCIE 本地中断允许位为 0，那么就使用软件轮询来监控 TDRE 和 TC 状态标记，而不是发生软件中断。

当程序检测到接收数据寄存器已满 ($RDRF = 1$) 时, 它通过读 $SCIxD$ 从接收数据寄存器获取数据。 $RDRF = 1$ 时读 $SCIxS1$, 这样 $RDRF$ 标记就被清除, 然后再读 $SCIxD$ 。

当使用轮询时, 该顺序自然在用户程序的正常过程中得到满足。如果使用硬件中断, 就必须在中断服务程序 (ISR) 中读 $SCIxS1$ 。在正常情况下, 这无论如何都要在 ISR 中完成, 以检查接收错误, 这样该顺序就自动满足了

当 RxD 线路在很长一段时间内保持闲置时, $IDLE$ 状态标记包括可以防止其进行重复设置的逻辑。当 $IDLE = 1$ 时, 读 $SCIxS1$ 可以清除 $IDLE$, 然后再读 $SCIxD$ 。在已经清除 $IDLE$ 后, 它不能再次进行设置, 直到接收器已经收到至少一个新字符并已设置 $RDRF$ 。

如果在造成设置 $RDRF$ 的已接收字符中检测到有关错误, 就在设置 $RDRF$ 的同时设置错误标记, 如噪音标记 (NF)、成帧错误 (FE) 和奇偶效验错误标记 (PF)。这些标记不会在溢出情况下设置。

如果当一个新字符准备好从接收移位器传输到接收数据缓冲器时已经设置了 $RDRF$, 就设置溢出 (OR) 标记, 而数据及任何有关的 NF 、 FE 或 PF 条件丢失。

任何时候, RxD 串行数据输入管脚上的活动边沿都会引起 $RXEDGIF$ 标记设置。把 1 写入 $RXEDGIF$ 会清除该标记。该功能取决于正被使能 ($RE = 1$) 的接收器。

11.3.5 其他 SCI 功能

以下几节描述其他 SCI 功能。

11.3.5.1 8 位和 9 位数据模式

通过在 $SCIxC1$ 中设置 M 控制位, SCI 系统 (发射器和接收器) 在经过配置后可以运行在 9 位数据模式中。在 9 位模式中, 在 SCI 数据寄存器的 MSB 的左侧有一个第 9 数据位。对发送数据缓冲器来说, 该位保存在 $SCIxC3$ 中的 $T8$ 。对接收器来说, 第 9 位保存在 $SCIxC3$ 中的 $R8$ 。

为了连贯写入发送数据寄存器, 写入 $SCIxD$ 前请先写入 $T8$ 位。

如果作为新字符第 9 位发送的位值和上一字符的位值相同, 则不需要重新写入 $T8$ 。当数据从发送数据缓冲器传输到发送移位器时, 在数据从 $SCIxD$ 传输到移位器的同时 $T8$ 中的值被复制。

9 位数据模式通常和奇偶效验一起使用, 以允许数据的 8 个位加第 9 位中的奇偶效验位。或者与地址标记唤醒一起使用, 这样第 9 数据位可以作为唤醒位。在自定义协议中, 第 9 位还可以作为软件控制标记。

11.3.5.2 停止模式运行

在所有停止模式中, SCI 模块的时钟都被暂停。

在 $STOP1$ 和 $STOP2$ 模式中, 所有 SCI 寄存器数据丢失, 当从这两种停止模式恢复时必须重新初始化。任何 SCI 模块寄存器在 $STOP3$ 模式中都不受影响。

接收输入活动边沿检测电路在 $STOP3$ 模式中仍然是活动的, 但在 $STOP2$ 模式中不活动。如果中断未屏蔽 ($RXEDGIE = 1$), 接收输入上的活动边沿将把 CPU 带离 $STOP3$ 模式。

注意, 由于时钟被暂停, 当从停止模式 (仅在 $STOP3$ 模式) 退出时, SCI 模块会重新开始运行。当 ISC 模块正在发送或接收字符时, 软件应确保不会进入停止模式。

11.3.5.3 循环模式

当 $LOOPS = 1$ 时，相同寄存器中的 $RSRC$ 位选择循环模式 ($RSRC = 0$) 或单线模式 ($RSRC = 1$)。循环模式独立于外部系统连接，有时用于检查软件，以帮助隔离系统问题。在该模式中，发射器输出内部连接到接收器输入，且 SCI 不使用 RxD 管脚，因此它恢复为通用端口 I/O 管脚。

11.3.5.4 单线运行

当 $LOOPS = 1$ 时，相同寄存器中的 $RSRC$ 位选择循环模式 ($RSRC = 0$) 或单线模式 ($RSRC = 1$)。单线模式用来执行半双工串行连接。接收器内部连接到发射器输出和 TxD 管脚。 RxD 管脚不使用并恢复为通用端口 I/O 管脚。

在单线模式中， $SCIxC3$ 中的 $TXDIR$ 位控制着 TxD 管脚上的串行数据方向。当 $TXDIR = 0$ 时， TxD 管脚是 SCI 接收器的输入，发射器与 TxD 管脚的连接被暂时断开，因此外部器件就可以向接收器发送串行数据。当 $TXDIR = 1$ 时， TxD 管脚是一个由发射器驱动的输出。在单线模式中，辅发射器到接收器的内部环回连接使接收器接收由发射器发送出来的字符。

第十二章 串行外围设备接口(S08SPIV3)

该 MC9S08AW60 系列有一个串行外设接口 (SPI) 模块。与 SPI 功能相关的四个引脚共享端口 E 引脚 4-7。见附录 A, “电气特性和时间规格”, SPI 电器参数信息

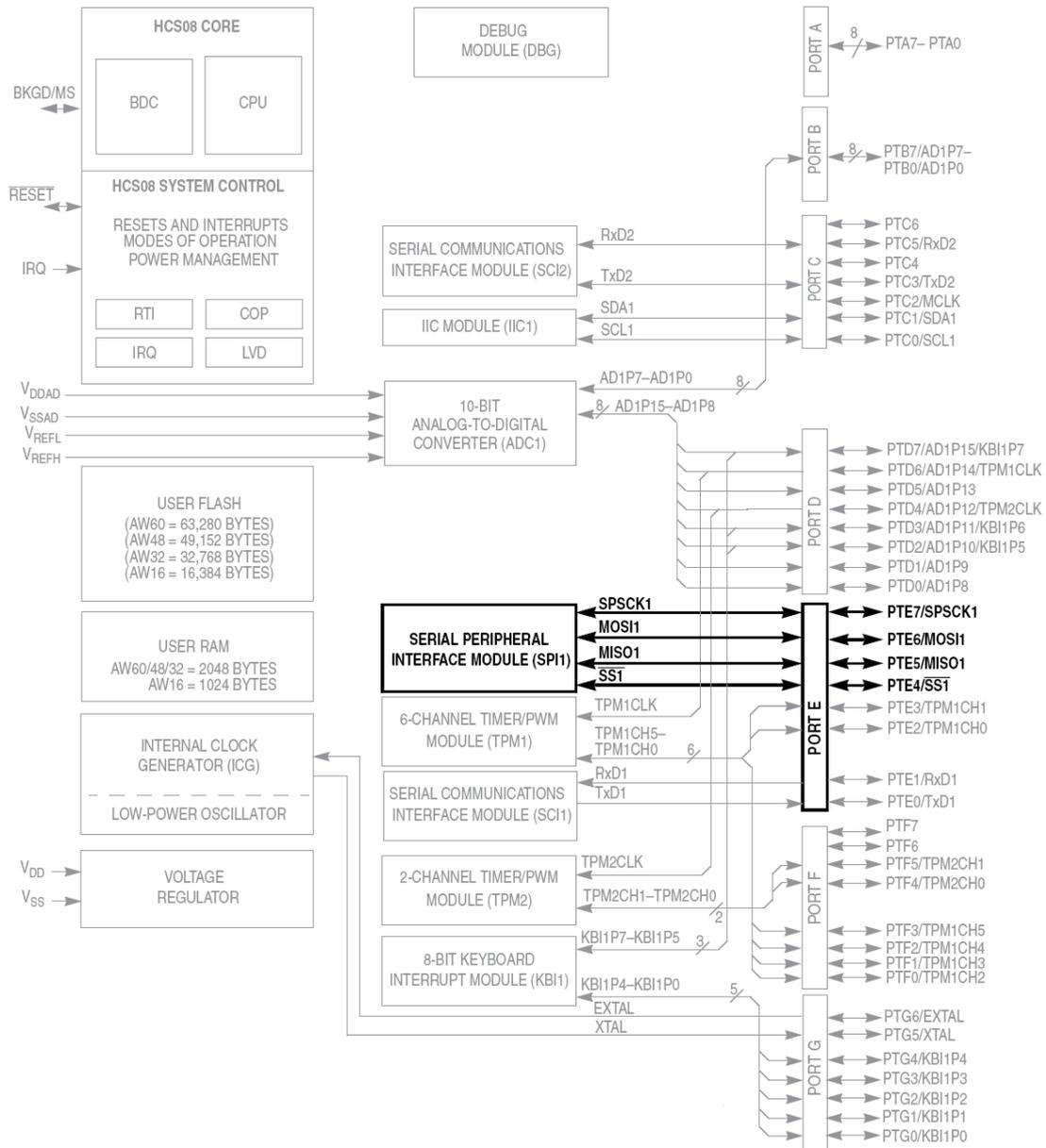


图 12-1 框图突出 SPI 模块

注意:

1. 如果是输入端口, 端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用 (IRQEDG=1), 引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测, 下拉式启用 (IRQPE = 1)。
3. IRQ 没有一个二极管到 V_{DD} , IRQ 不应该超过 V_{DD} 。

4. 引脚包含完整的上拉装置。
5. 引脚 PTD7, PTD3, PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用(KBIPEn = 1)并选定上升沿(KBEDGn = 1)时, 下拉式启用时。

12.0.1 特性

SPI 模块的特性包括:

- 主或辅模式运行
- 全双工或单线双向选项
- 可编程发送波特率
- 双缓冲发送和接收
- 串行时钟相位和极性选项
- 辅选择输出
- 可选择的MSB在先或LSB在先转换

12.0.2 结构图

本节中的内容包括: 显示了 SPI 系统连接的结构图, SPI 模块的内部组织, 和用于控制主模式位速率的 SPI 时钟分频器。

12.0.2.1 SPI 系统结构图

图 12-2 所示为两个以主从分布方式连接的 MCU 构成的 SPI 模块。主器件启动所有的 SPI 数据传输。在一次传输过程中, 主器件把数据移出(在 MOSI 管脚)到从器件同时从从器件传送数据进入主器件(在 MISO 管脚)。这种传输有效的交换了两个 SPI 系统中 SPI 移位寄存器里面的数据。SPSCK 信号是从主器件的一个时钟输出, 也是从器件的时钟输入。从器件必须通过一个从选择输入管脚(SS 管脚)上的低电平选中。在这个系统中, 主器件已经配置它的 SS 管脚为一个可选的从选择输出。

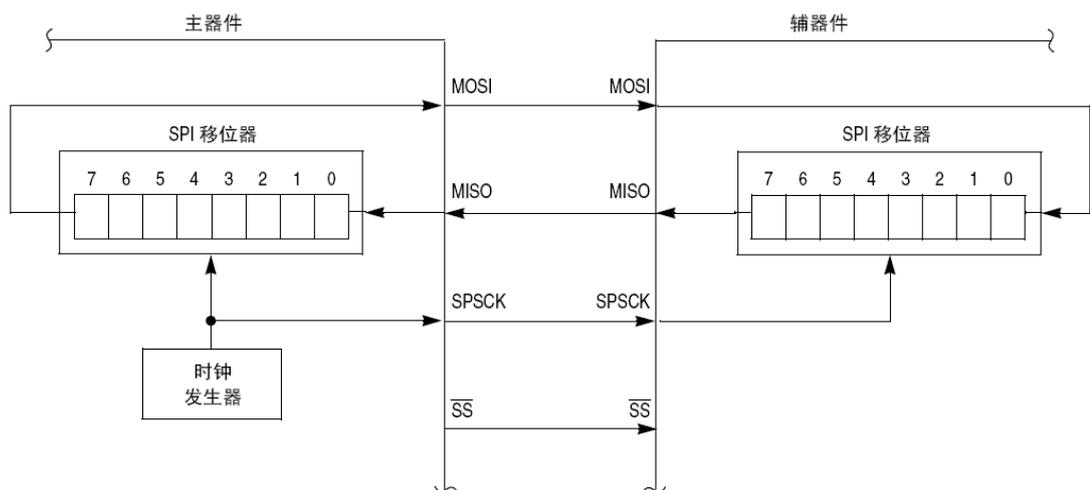


图 12-2. SPI 系统连接

SPI 系统最常见的使用包括连接简单移位寄存器, 以增加输入或输出端口, 或者连接小型外围器件, 如串行 A/D 或 D/A 移位器。尽管图 12-2 显示了一个在两 MCU 间交换数据的系统, 但许多实际的系统的连接更简单, 数据要么从主 MCU 单向传输到从 MCU, 要么从从 MCU 单向传输到主 MCU。

12.0.2.1 SPI 模块结构图

图 12-3 是 SPI 模块的结构图。SPI 的中心元件是 SPI 移位寄存器。数据写入双缓冲发送器（写入 SPID），然后转移到位于数据传输起点的 SPI 移位寄存器。在数据字节中转换后，数据被传输到双缓冲接收器，在这里数据可以被读取（从 SPID 读取）。管脚复用逻辑控制着 MCU 管脚和 SPI 模块间的连接。

当 SPI 配置为主 SPI 时，时钟输出被路由到 SPSCCK 管脚，移位器输出被路由到 MOSI，移位器输入从 MISO 管脚路由出来。

当 SPI 配置为从 SPI 时，SPSCCK 管脚为时钟输出。MOSI 为移位输出，MISO 管脚为移位器输入。

在外部 SPI 系统，只需将所有 SPSCCK 管脚彼此连接，所有 MISO 管脚连接起来，所有 MOSI 管脚连接起来就可以来。外围器件通常为这些管脚使用略有不同的名称。

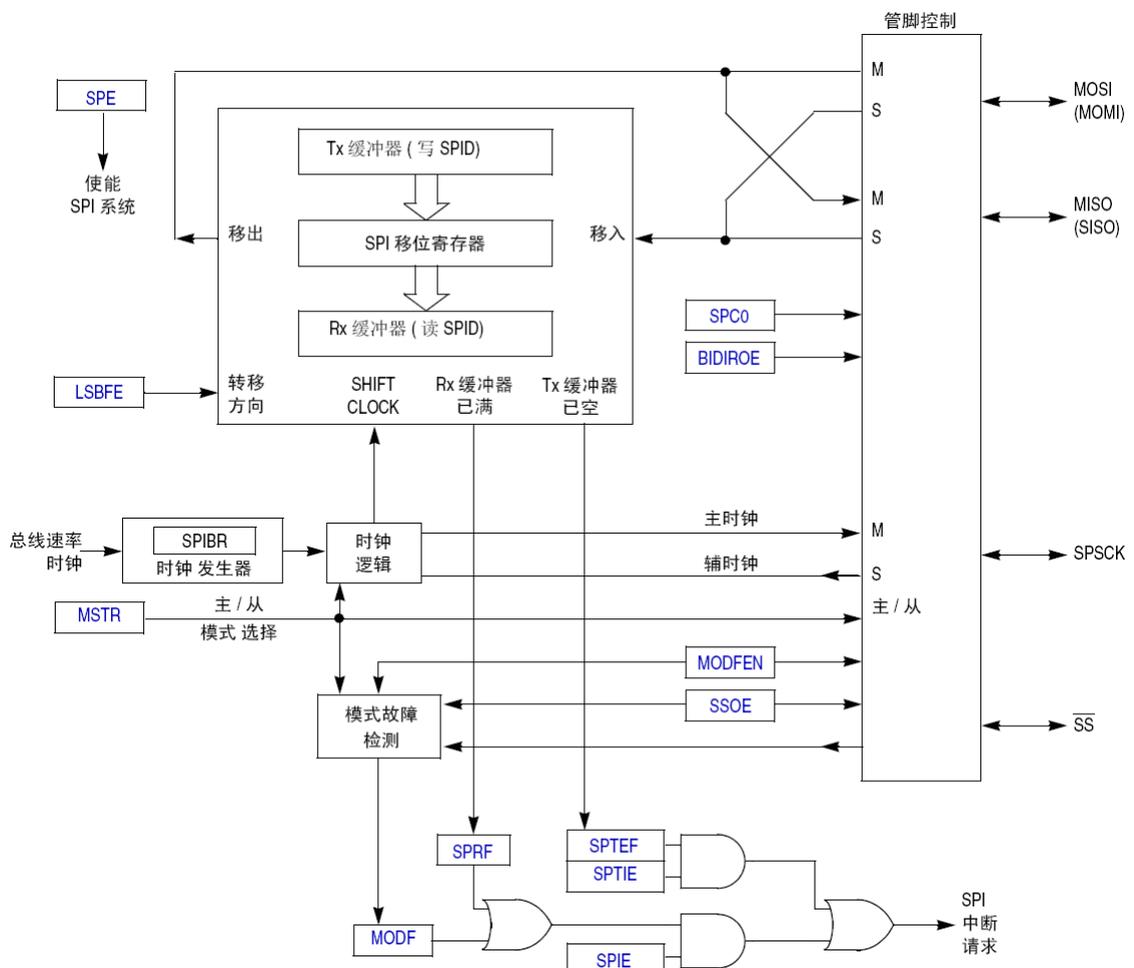


图 12-3. SPI 模块结构图

12.0.3 SPI 波特率生成

如图 12-4 所示，SPI 波特率发生器的时钟源为总线时钟。这三个调节位（SPPR2:SPPR1:SPPR0）从 1、2、3、4、5、6、7 或 8 中选择一个调节系数。这三个波特率选择位（SPR2:SPR1:SPR0）除以预定标因子输出的 2、4、8、16、32、64、128 或 256 次幂，便得到内部 SPI 主模式比特率时钟。

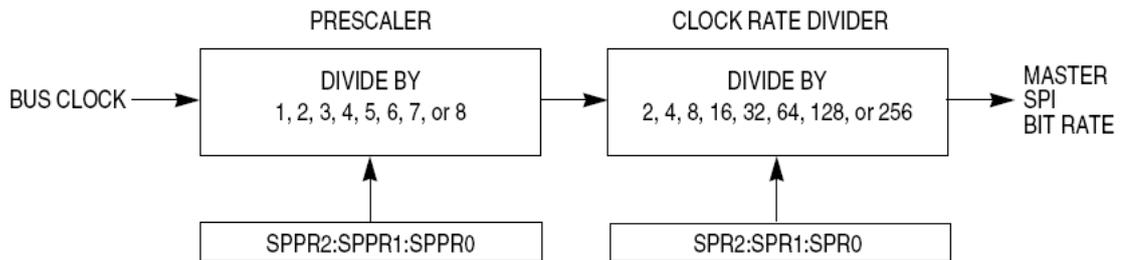


图 12-4. SPI 波特率生成

12.1 外部信号描述

SPI 共享 4 个端口管脚。这些管脚的功能取决于 SPI 控制位的设置。当 SPI 禁止 ($SPE = 0$) 时，这 4 个管脚恢复为不受 SPI 控制的通用端口 I/O 管脚。

12.1.1 SPSCCK — SPI 串行时钟

当 SPI 作为从 SPI 被使能时，该管脚是串行时钟输入。当 SPI 作为主 SPI 被使能时，该管脚是串行时钟输出。

12.1.2 MOSI — 主数据输出，从数据输入

当 SPI 作为主 SPI 被使能且 SPI 管脚控制零 ($SPC0$) 为 0 (不是双向模式) 时，该管脚是串行数据输出。当 SPI 作为从 SPI 被使能且 $SPC0 = 0$ 时，该管脚为串行数据输入。如果 $SPC0 = 1$ ，选择单线双向模式并选择了主模式，该管脚就成为双向数据 I/O 管脚 ($MOMI$)。同样，双向模式输出使能位决定该管脚作为输入 ($BIDIROE = 0$) 管脚还是输出管脚 ($BIDIROE = 1$)。如果 $SPC0 = 1$ 且选择了辅模式，该管脚不会被 SPI 使用，并恢复为通用端口 I/O 管脚。

12.1.3 MISO — 主数据输入，从数据输出

当 SPI 作为主 SPI 被使能且 SPI 管脚控制零 ($SPC0$) 为 0 (不是双向模式) 时，该管脚是串行数据输入。当 SPI 作为从 SPI 被使能且 $SPC0 = 0$ 时，该管脚是串行数据输出。如果 $SPC0 = 1$ ，选择单线双向模式并选择了辅模式，该管脚变成双向数据 I/O 管脚 ($SISO$)，双向模式输出使能位决定该管脚作为输入 ($BIDIROE = 0$) 管脚还是输出管脚 ($BIDIROE = 1$)。如果 $SPC0 = 1$ 且选择了辅模式，该管脚不会被 SPI 使用，并恢复为通用端口 I/O 管脚。

12.1.4 \overline{SS} — 从模式选择

当 SPI 作为从 SPI 被使能时，该管脚是低电平有效的从选择输入。当 SPI 作为主 SPI 被使能且模式默认使能关闭 ($MODFEN = 0$) 时，该管脚不被 SPI 使用，并恢复为通用端口 I/O 管脚。当 SPI 作为主 SPI 被使能时， $MODFEN = 1$ ，辅选择输出使能位决定该管脚是作为模式默认输入 ($SSOE = 0$) 还是辅选择输出 ($SSOE = 1$)。

12.2 运行模式

12.2.1 处于停止模式的 SPI

SPI 在所有停止模式禁止，无论在执行 STOP 指令前的设置如何。在 STOP1 或 STOP2

模式期间，SPI 模块将完全关闭。一旦从 STOP1 或 STOP2 模式唤醒，SPI 模块将进入复位状态。在 STOP3 模式期间，SPI 模块的时钟暂停。寄存器没受到影响。如果通过复位退出 STOP3，SPI 被置复位状态。如果通过中断退出 STOP3，SPI 继续保持其进入 STOP3 时所处的状态。

12.3 寄存器定义

SPI 有 5 个 8 位寄存器用于选择 SPI 选项、控制波特率、报告 SPI 状态和用来传送/接收数据。所有 SPI 寄存器的绝对地址分配请参考本技术手册中“存储器”一章的“直接页面寄存器总结”。在本节中仅通过它们的名字对寄存器和控制位进行引用。一个飞思卡尔提供的等价的或者头文件用来映射这些名字为适当的绝对地址。

12.3.1 SPI 控制寄存器 1 (SPIIC1)

该读/写寄存器包括 SPI 使能控制、中断使能和配置选项。

	7	6	5	4	3	2	1	0
R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
W								
复位	0	0	0	0	0	1	0	0

图 12-5. SPI 控制寄存器 1 (SPIIC1)

表 12-1. SPIIC1 字段描述

字段	描述
7 SPIE	SPI 中断使能 （用于 SPRF 和 MODF）—— 这是 SPI 接收缓冲器已满 (SPRF) 和模式故障 (MODF) 事件的中断使能。 0 禁止从 SPRF 和 MODF 中中断（使用轮询） 1 当 SPRF 或 MODF 为 1，请求硬件中断
6 SPE	SPI 系统使能 —— 禁止 SPI 将暂停正在进行的任何传输、清除数据缓冲器、初始化内部状态设备。SPRF 被清除，并设置 SPTEF 来显示 SPI 发送数据缓冲器空。 0 SPI 系统禁止 1 SPI 系统使能
5 SPTIE	SPI 发送中断使能 —— 这是 SPI 发送缓冲器空 (SPTEF) 的中断使能位。 0 禁止从 SPTEF 中中断（使用轮询） 1 当 SPTEF 为 1 时，请求硬件中断
4 MSTR	主/从模式选择 0 SPI 模块配置为从 SPI 器件 1 SPI 模块配置为主 SPI 器件
3	时钟极性 ——这个位有效地对从主 SPI 到从 SPI 器件的时钟信号串联放置

CPOL	一个反相器。更多信息请参考 12.4.1 一节，“SPI 时钟格式”。 0 高有效 SPI 时钟（低无效） 1 低有效 SPI 时钟（高无效）
2 CPHA	时钟相位 — 该位选择两种时钟格式的一种用于不同类型的同步串行外围器件。详情请参见 12.4.1，“SPI 时钟格式”。 0 SPSCCK 上的第一个边沿出现在 8 周期数据传输的第一个周期的中央 1 SPSCCK 上的第一个边沿出现在 8 周期数据传输的第一个周期的起点
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 12-2 所示。
0 LSBFE	LSB 优先（移位器方向） 0 SPI 串行数据传输以最高有效位开始 1 SPI 串行数据传输以最低有效位开始

表 12-2. \overline{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。

12.3.2 SPI 控制寄存器 2（SPI1C2）

该读/写寄存器用来控制 SPI 系统的任选功能。第 7、6、5 和 2 位无效，总为 0。

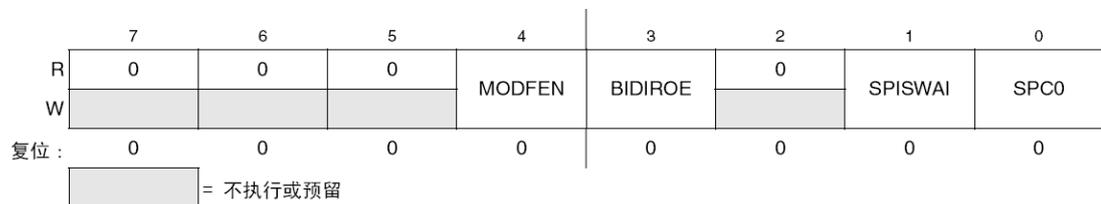


图 12-6. SPI 控制寄存器 2 (SPI1C2)

表 12-3. 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 12-2）。

	<p>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</p> <p>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</p>
<p>3</p> <p>BIDIROE</p>	<p>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。</p> <p>0 输出驱动器禁止，因此 SPI 数据 I/O 管脚作为输入</p> <p>1 SPI I/O 管脚作为输出使能</p>
<p>1</p> <p>SPISWAI</p>	<p>SPI 停止在等待模式中</p> <p>0 在等待模式中 SPI 时钟继续运行</p> <p>1 当 MCU 进入等待模式时 SPI 时钟停止</p>
<p>0</p> <p>SPC0</p>	<p>SPI 管脚控制 0—SPC0 位用于选择单线双向模式。如果 MSTR=0（从模式），SPI 将 MISO (SISO) 管脚用于双向 SPI 数据传输。如果 MSTR=1（主模式），SPI 将 MOSI (MOMI) 管脚用于双向 SPI 数据传输。当 SPC0=1 时，BIDIROE 被用于为单个双向 SPI I/O 管脚使能或禁止输出驱动器。</p> <p>0 SPI 使用单独管脚用于数据输入或输出</p> <p>1 SPI 配置为单线双向操作</p>

12.3.3 SPI 波特率寄存器 (SPI1BR)

该寄存器用于为一个 SPI 主机设定预定标器和位速率分频因子。它可以在任何时间被读取或写入。

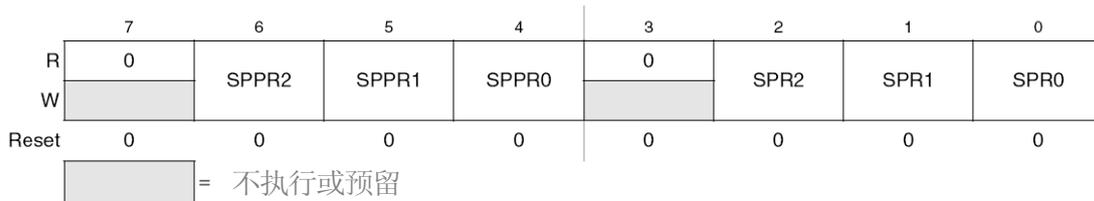


图 12-7. SPI 波特率寄存器 (SPI1BR)

表 12-4. SPI1BR 寄存器字段描述

字段	描述
<p>6:4</p> <p>SPPR[2:0]</p>	<p>SPI 波特率预分频系数 — 该 3 位字段为 SPI 波特率预分频器选择 8 个系数中的一个，如表 12-5 所示。该预分频器的输入是总线速率时钟 (BUSCLK)。该预分频器的输出驱动 SPI 波特率系数的输入 (图 12-4)。</p>
<p>2:0</p> <p>SPR[3:0]</p>	<p>SPI 波特率系数 — 该 4 位字段为 SPI 波特率系数选择 8 个系数中的一个，如表 12-6 所示。该被除数的输入来自 SPI 波特率预分频器 (见图 12-4)。该被除数的输出是主模式的 SPI 波特率时钟。</p>

表 12-5. SPI Baud 波特率预分频器系数

SPPR2:SPPR1:SPPR0	预分频器系数
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

表 12-6. SPI 波特率系数

SPR2:SPR1:SPR0	速度系数
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

12.3.4 SPI 状态寄存器 (SPI1S)

该寄存器有三个只读状态位，位 6、3、2、1 和 0 没有定义，总是读 0，写入无效。

	7	6	5	4	3	2	1	0
R	SPRF	0	SPTEF	MODF	0	0	0	0
W								
复位:	0	0	1	0	0	0	0	0

= 不执行或预留

图 12-8. SPI 状态寄存器 (SPI1S)

表 12-7. SPI1S 寄存器字段描述

字段	描述
----	----

7 SPRF	<p>SPI 接受缓冲器满标志—在一次 SPI 传输完成时, SPRF 被置位, 表明接收到的数据可以从 SPI 数据寄存器 (SPID) 读取。当 SPRF 被置位时, 通过读 SPRF, 然后读取 SPI 数据寄存器, 可将其清除。</p> <p>0 在接收数据缓冲器中无可用数据 1 在接收数据缓冲器中的数据可用</p>
5 SPTEF	<p>SPI 发送缓冲器空标志—当发送数据缓冲区空时, 这个位被置位。PPTEF 被置位时, 通过读取 SPI1S, 然后向 SPI1D 的发送缓冲器写入一个数值, 可将其清除。SPTEF=1 时, SPI1S 必须在向 SPI1D 写入数据之前被读取, 否则 SPI1D 写操作将被忽略。如果 SPIC1 中的 SPTIE 位也被置位, SPTEF 会产生一个 SPTEF, CPU 中断请求。当一个数据字节从发送缓冲器传输进入发送移位寄存器时, SPTEF 被自动置位。对于一个空闲 SPI (在发送缓冲器或移位寄存器中没有数据, 且无进行中的传输), 写入 SPID 中的数据几乎立即被传输至移位器, 因此, SPTEF 在两个总线周期内被置位, 从而允许第二个 8 位数据值被排入发送缓冲器。在将移位寄存器中的值传输完成后, 从发送缓冲器排列的值将自动转移至移位器, 且 SPTEF 将被置位以表明在发送缓冲器中有空间可用于新数据传输。如果没有新数据在发送缓冲器中等待, SPTEF 只是保持置位状态, 而不会有数据从缓冲器转移至移位器。</p> <p>0 SPI 发送缓冲器非空 1 SPI 发送缓冲器空</p>
4 MODF	<p>主模式故障标志—如果 SPI 被设置为主模式, 且从模式选择输入变为低电平时, MODF 被置位, 这表明某个其它的 SPI 器件也被设置成了主模式。只有 MSTR=1, MODFEN=1, 且 SSOE=0 时, SS 管脚才作为为模式故障错误输入; 否则, MODF 将不会被置位。MODF 为 1 时, 通过读 MODF, 然后写入 SPI 控制寄存器 1 (SPIC1), 可将其清空。</p> <p>0 无模式故障错误 1 检测到模式故障错误</p>

12.3.5 SPI 数据寄存器 (SPI1D)

	7	6	5	4	3	2	1	0
R								
W	Bit 7	6	5	4	3	2	1	Bit 0
复位	0	0	0	0	0	0	0	0

图 12-9. SPI 数据寄存器 (SPI1D)

读取该寄存器将返回从接收数据缓冲器中读取的数据。写该寄存器将会把数据写入发送数据缓冲器。当 SPI 被配置为主模式时, 写入数据到传输数据缓冲器发起一次 SPI 传输。

除非 SPI 发送缓冲器空标志 (SPTEF) 被置位, 数据不应被写入到发送数据缓冲器, 表明发送缓冲器内有空间来排列一个新的发送字节。

在 SPRF 置位后且另一传输完成之前的任意时刻, 数据都可以从 SPID 中被读取。在一个新的传输完成前, 从接收数据缓冲器读出数据失败, 将会引起一个接收丢包状态, 并且新传输的数据也会丢失。

12.4 功能描述

通过检查 SPI 发送缓冲器空标志(SPTEF=1), 然后向主 SPI 器件 SPI 数据寄存器(SPID) 写入一个字节的的数据, 可以发起一个 SPI 传输。当 SPI 移位寄存器可用时, 这个字节的数据从发送数据缓冲器被移动到移位器中, 置位 SPTEF 用以表明缓冲器内有空间来排列另一个需要发送的字符, 并启动 SPI 串行传输。

在 SPI 传输期间, 在 MISO 管脚的某个 SPSCK 边沿, 数据会被采样(读取)和移位, 半个 SPSCK 周期后, 改变 MOSI 管脚的位值。8 个 SPSCK 周期后, 位于主器件移位寄存器中的数据已经被移出 MOSI 管脚, 转移到从器件, 同时, 8 位数据从 MISO 管脚的移位器移位至主器件移位寄存器中。在传输结束时, 接收到的数据字节从移位器转移到接收数据缓冲器内, 且 SPRF 被置位, 以表示数据可以通过读 SPID 被读取。如果在传输结束时有另一个字节的数据在发送缓冲器内等待, 就会转移到移位器中, SPTEF 被置位, 新的一次传输便开始了。

通常, SPI 数据从最高有效位(MSB)开始被传送。如果最低有效位优先使能位(LSBFE=1) 被置位, 则 SPI 数据从 LSB 开始移位。

当 SPI 被配置为从模式时, 它的 \overline{SS} 管脚在传输开始之前必须被驱动为低电平, 且在整个传输过程中 \overline{SS} 必须保持低平。如果所选时钟格式的位置有 CPHA=0, 在连续传输之间, \overline{SS} 必须被驱动为逻辑 1。如果 CPHA=1, \overline{SS} 可以在连续传输中保持低平。更多信息, 请参考 12.4.1 “SPI 时钟格式” 一节。

因为发送器和接收器是双缓冲器, 所以, 除当前正在被移出的字节外, 另一个字节, 可以被排列进入发送数据缓冲器, 并且, 当一个新的字符被移入时, 先前接收到的字符可以在接收数据缓冲器内。SPTEF 标志指示发送缓冲器内何时有空间接受一个新的字符。SPRF 标志指示在接收数据缓冲器内一个已接收的字符何时是可用的。这个接收到字符必须在下一个传输完成或接收丢包错误发生之前, 从接收缓冲器中被读出(读取 SPID)。

在接收丢包的情况下, 因为接收缓冲器仍然保持先前的字符, 尚未准备好接收新数据, 所以, 新的数据会丢失。对于这种丢包情况, 不寄存器指示, 所以, 对于应用系统设计者来说, 必须确保先前数据在新的传输开始之前已经从接收缓冲器中被读取。

12.4.1 SPI 时钟格式

为了适应不同厂商的各种同步串行外设, SPI 系统具有一个时钟极性(CPOL)位和一个时钟相位(CPHA)控制位, 用以选择为数据传输选择 4 种时钟格式的其中一种。CPOL 选择性的插入一个与时钟串联的反相器。CPHA 在时钟与数据之间可选择两种不同的时钟相位关系。

图 12-10 所示为当 CPHA=1 时的时钟格式。在图的顶端, 显示了用于参考的 8 位, 第一位以第一个 SPSCK 沿开始, 而第 8 位以第 16 个 SPSCK 沿之后的半个 SPSCK 周期处结束。MSB 优先和 LSB 优先线显示了取决于 LSBFE 的设置情况的 SPI 数据位的顺序。SPSCK 极性的两种变化情况都得到了显示, 但是只有其中一种波形应用于一种特殊的传输, 这取决于 CPOL 的值。SAMPLE IN 波形应用于从模式的 MOSI 输入或主模式的 MISO 输入。MOSI 波形应用于主器件的 MOSI 输出管脚, 而 MISO 波形应用于从器件 MISO 的输出管脚。SS OUT 波形应用于主器件的从模式选择输出(如果 MODFEN 和 SSOE=1)。主器件的 SS 输出在传输开始之前变为低电平有效, 且保持半个 SPSCK 周期; 并在发送的第 8 位结束时返回高电平。SS IN 波形应用于从器件的从模式选择输入。

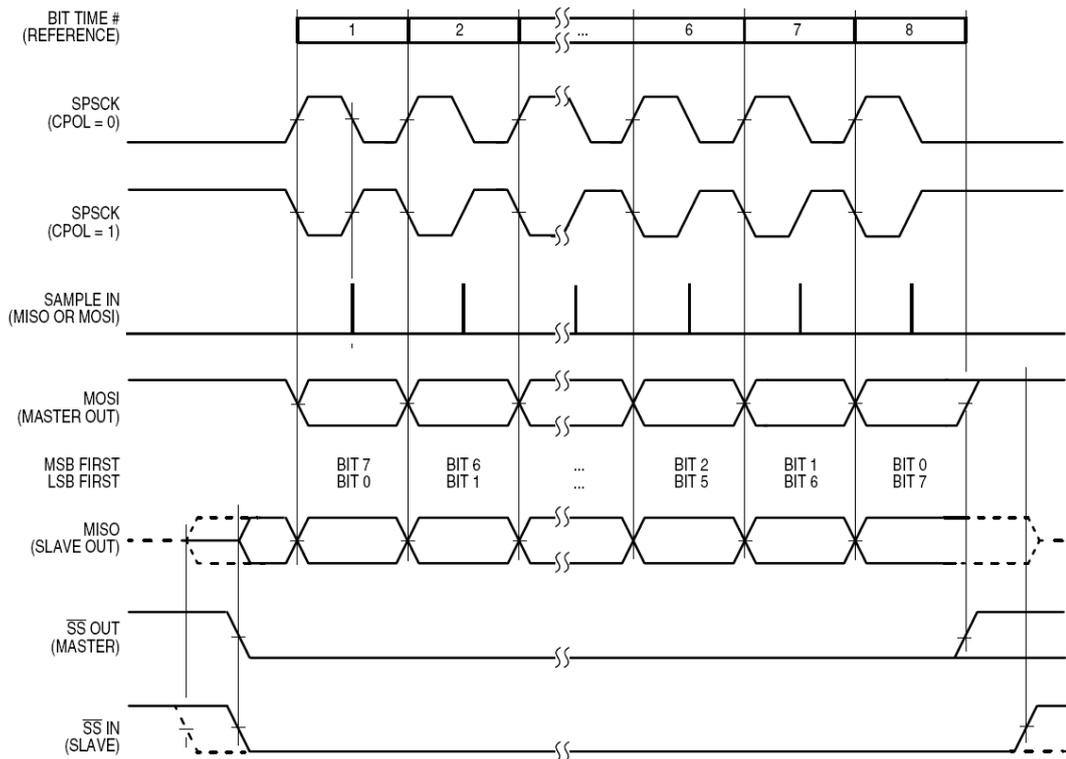


图 12-10. SPI 时钟格式 (CPHA = 1)

当 CPHA=1, \overline{SS} 进入低电平有效时, 从器件开始驱动它的 MISO 输出, 但是直到第一个 SPSCK 边沿时数据才被定义。第一个 SPSCK 沿将数据的第一位从移位器向上转移至主器件的 MOSI 输出端和从器件的 MISO 输出端。下一个 SPSCK 沿引起主器件和从器件分别在它们的 MISO 和 MOSI 输入端同时对数据位值进行采样。在第三个 SPSCK 沿, SPI 移位器在刚才被采样过的数值内移位一个位的位置, 并且将第二个数据位的值从移位器的另一端移出, 分别转移至主器件和从器件的 MOSI 和 MISO 输出端。当 CPHA=1 时, 从器件的 SS 输入端在两次传输之间不需要进入它的无效高电平状态。

图 12-11 所示为当 CPHA=0 时的时钟格式。在图的顶端, 显示了用于参考的 8 位, 当从模式被选择好时 ($\overline{SS IN}$ 进入低电平) 第一位开始; 第 8 位在最后一个 SPSCK 沿结束。MSB 优先和 LSB 优先线显示了取决于 LSBFE 设置情况的 SPI 数据位的顺序。SPSCK 极性的两种变化情况都得到了显示, 但是只有其中一种波形应用于一种特殊的传输, 这取决于 CPOL 的值。SAMPLE IN 波形应用于从器件的 MOSI 输入或主器件的 MISO 输入。MOSI 波形应用于主器件的 MOSI 输出管脚, 而 MISO 波形应用于从器件 MISO 输出管脚。SS OUT 波形应用于主器件的从模式选择输出 (如果 MODFEN 和 SSOE=1)。主器件的 SS 输出在第一位传输开始时变为低电平有效; 并在发送第 8 位结束后返回高电平, 且保持半个 SPSCK 周期。SS IN 波形应用于从器件的从模式选择输入。

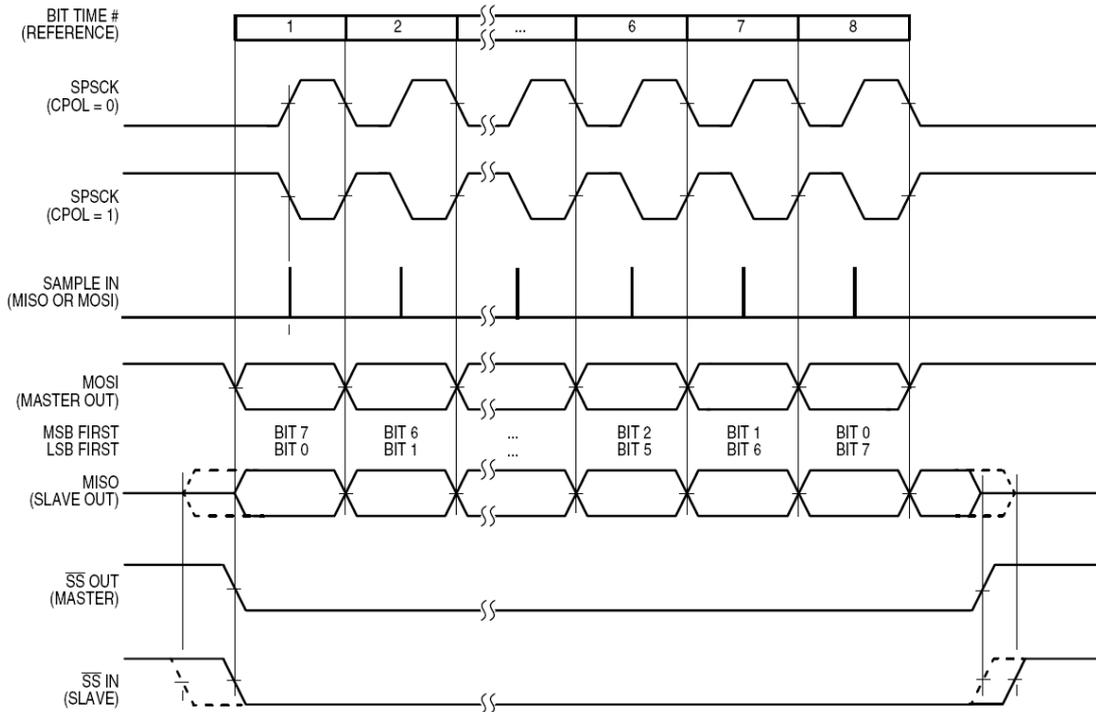


图 12-11. SPI 时钟格式(CPHA = 0)

当 CPHA=0, \overline{SS} 进入低电平有效时, 从器件从第一个数据位值开始驱动它的 MISO 输出 (MSB 还是 LSB 取决于 LSBFE)。第一个 SPSCK 沿引起主器件和从器件分别在它们的 MISO 和 MOSI 输入端对数据位值进行采样。在第二个 SPSCK 沿, SPI 移位器在刚才被采样过的数值内移位一个位的位置, 并且将第二个数据位的值从移位器的另一端移出, 分别转移至主器件和从器件的 MOSI 和 MISO 输出端。当 CHPA=0 时, 从器件的 SS 输入端在两次传输之间必须进入它的无效高电平状态。

12.4.2 SPI 中断

存在三个标志位, 其中两个为中断屏蔽位, 另一个为与 SPI 系统相关的中断向量。SPI 中断使能蔽位 (SPIE) 使能来自 SPI 接收器满标志位 (SPRF) 和模式故障标志位 (MODF) 的中断。SPI 发送中断使能屏蔽位 (SPTIE) 使能来自 SPI 发送缓冲器空标志 (SPTEF) 的中断。当某一个标志位被置位时, 相关的中断屏蔽位也置位, 一个硬件中断请求即被发送到 CPU。如果这个中断屏蔽位被清空, 软件则会轮询相关的标志位而不使用中断。SPI 中断服务程序 (ISR) 应该检查该标志位以确定为何引起中断。服务程序也应该在从 ISR 返回之前清空标志位 (通常接近 ISR 开始时)。

12.4.3 模式故障检测

当主 SPI 器件在 \overline{SS} 管脚 (如果 \overline{SS} 管脚被配置为模式故障输入信号) 检测到一个错误时, 一个模式故障将发生, 且模式故障标志 (MODF) 被置位。当 MSTR=1, 模式故障使能被置位 (MODFEN=1), 且从模式选择输出使能位被清空 (SSOE=0) 时, \overline{SS} 管脚即被配置为模式故障输入信号。

模式故障检测功能可用于可能有多个 SPI 器件同时成为主器件的系统中。当主器件 \overline{SS} 管脚低平时, 错误被检测到, 这表明某些其它的 SPI 器件就像从器件一样正在试图寻址该主器件。这也许指示了一个不利的输出驱动器冲突, 所以, 当这样的错误出现时, 模式故障逻辑

辑被设计为禁用所有 SPI 输出驱动。

当检测到某个模式故障时，MODF 被置位且 MSTR 被清空，从而将 SPI 的配置变回从模式。位于 SPSCK、MOSI 和 MISO（如果为非双向模式）的输出驱动也被禁用。

当 MODF 被置位时，通过读取 MODF，然后对 SPI 控制寄存器 1（SPIC1）进行写操作，可将 MODF 清除。用户软件应该在将 SPI 改变回主模式前确认错误状态已经得到纠正。

第十三章 内部集成电路(S08IICV1)

13.1 绪论

MC9S08AW60 系列控制器有一个内部集成电路（IIC）模块为与其他集成电路通信。两个引脚与此模块相关，SCL 和 SDA，是漏极开路输出，分别共享端口 C 引脚 0 和 1。

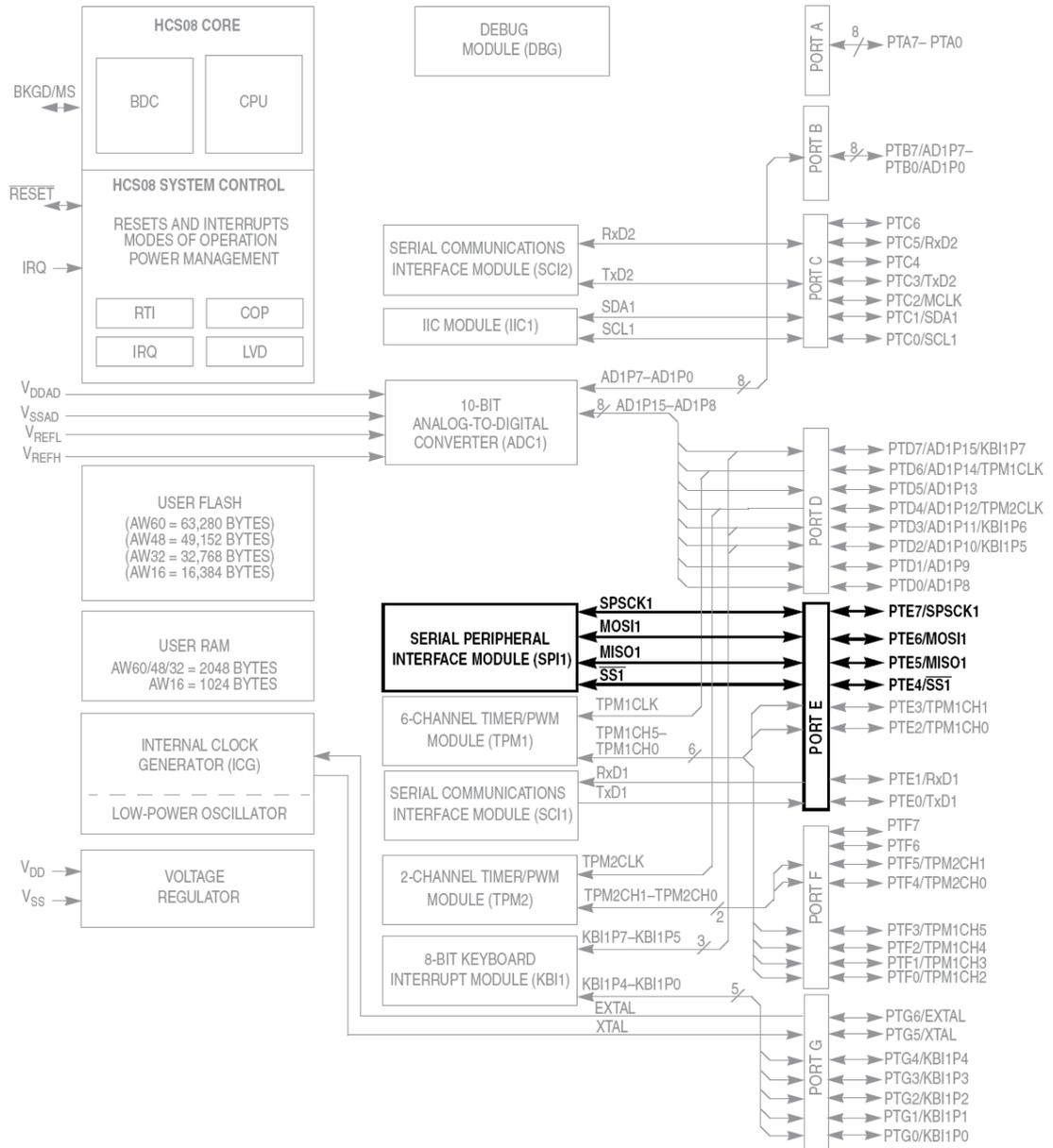


图 13-1 框图突出 IIC 模块

注意：

1. 如果是输入端口，端口引脚是软件配置与上拉装置。
2. 如果 IRQ 启用(IRQEDG=1)，引脚包含软件配置上拉/下拉式设备。如果选定上升沿检测，下拉式启用(IRQPE = 1)。

-
3. IRQ 没有一个二极管到 V_{DD} , IRQ 不应该超过 V_{DD} 。
 4. 引脚包含完整的上拉装置。
 5. 引脚 PTD7, PTD3, PTD2 和 PTG4 包含上拉和下拉式装置。当 KBI 启用($KBIPE_n = 1$)并选定上升沿($KBEDG_n = 1$)时, 下拉式启用时。

13.1.1 特点

IIC 独有的特点包括:

- 兼容 IIC 总线标准
- 多主控操作
- 可软件编程选择 64 种不同的串行时钟频率
- 软件可选应答位
- 中断驱动的逐字节数据传输
- 仲裁丢失中断的同时, 自动从主机模式切换到从机模式
- 可从地址识别中断
- 启停信号的产生和检测
- 重复产生开始信号
- ACK 信号的产生和检测
- 总线忙检测

13.1.2 工作模式

在常规和监控模式下 IIC 功能相同。以下给出 IIC 模块在各种 MCU 模式下简短的描述。

- 运行模式—这是工作的基本模式。要降低该模式下的功耗, 只需禁止 IIC 模块。
- 等待模式—当 MCU 在等待模式中, 该模块继续运行并且可以提供唤醒中断。
- 停止模式—在 stop3 模式下为了减少功率消耗, IIC 模块未激活。STOP 指令不影响 IIC 寄存器状态。Stop2 复位 IIC 寄存器内容。

13.1.3 结构图

IIC 模块的结构图如图 13-2 所示。

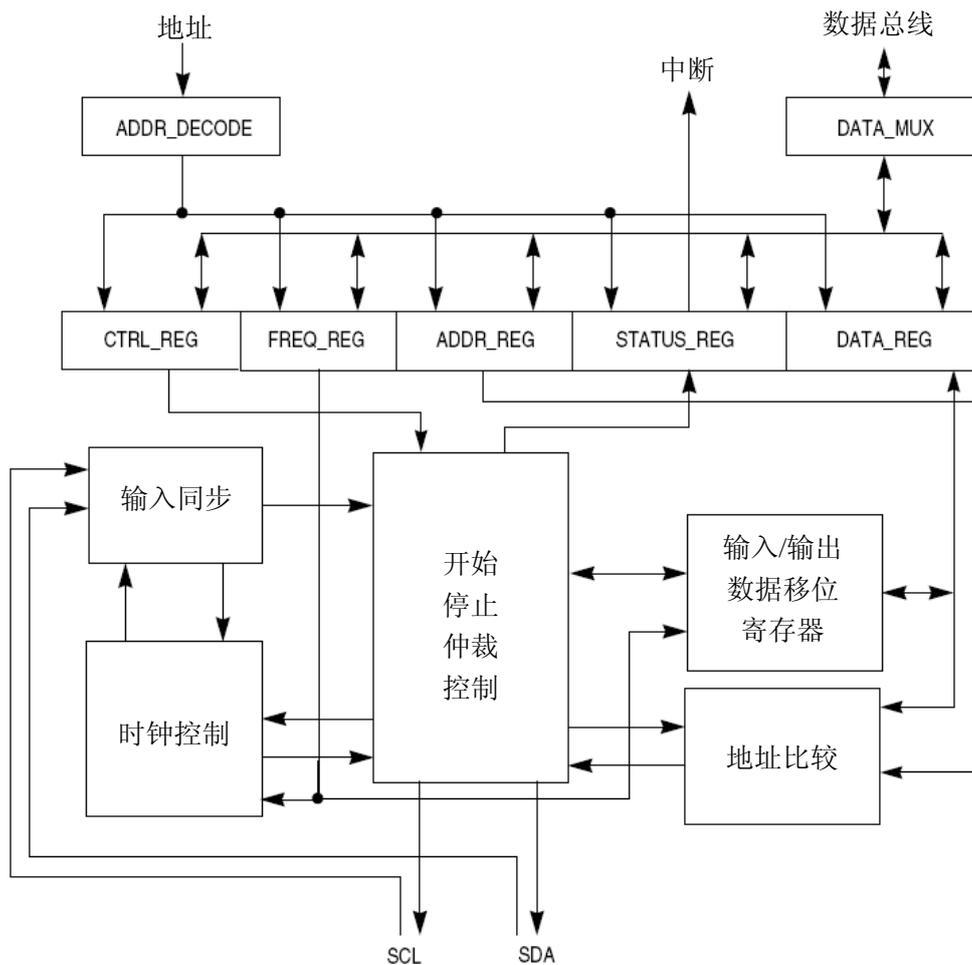


图 13-2 IIC 功能结构图

13.2 外部信号描述

这部分描述了各个用户可访问引脚信号。

13.2.1 SCL-串行时钟线

双向的 SCL 是 IIC 系统的串行时钟线。

12 SDA-串行数据线

双向的 SDA 是 IIC 系统的串行数据线。

13.3 寄存器定义

这部分按照寻址顺序对 IIC 寄存器进行描述。

对于所有的 IIC 寄存器都涉及到直接页寄存器的绝对地址分配，直接页寄存器概要在本数据手册的存储器一章中。这部分中的寄存器和控制位仅与他们的名

称相关。飞思卡尔提供的等式或者标头文件是用来将这些名称转换成适当的绝对地址。

13.3.1 IIC 地址寄存器(IIC1A)

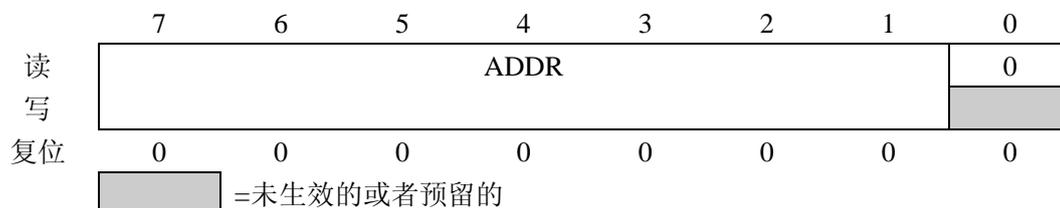


图 13-3. IIC 地址寄存器(IIC1A)

表 13-1. IIC1A 寄存器域描述

域	描述
7:1 ADDR[7:1]	IIC 地址寄存器—ADDR 包含 IIC 模块中要使用的特殊从机地址。当被设为从机时，该模块会响应寻址。

13.3.2 IIC 分频寄存器(IIC1F)

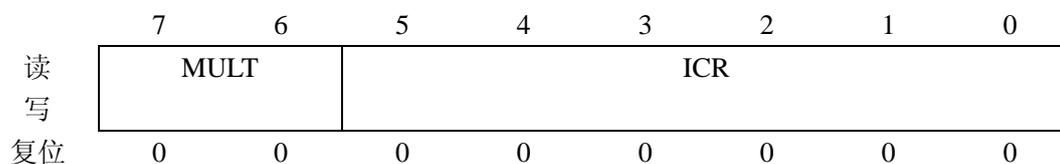


图 13-4. IIC 分频寄存器(IIC1F)

表 13-2. IIC1A 寄存器域描述

域	描述
7:6 MULT	IIC 增频因子—MULT 位定义增频因子的值。该值与 SCL 的分频因子配合使用，产生 IIC 波特率。由 MULT 位定义的分频因子值定义如下。 MULT=00、01、10 时，分频因子值分别为 1、2、4；MULT=11，不使用。
5:0 ICR	IIC 时钟速率—ICR 位用于为位速率选择预分配总线时钟。这 6 位用来定义 SCL 分频因子和 SDA 保持值。SCL 分频因子乘以由 MULT 寄存器提供的增频因子的值产生 IIC 波特率。 IIC 总线波特率=CPU 总线频率(Hz)/(增频因子值×分频因子) SDA 保持时间是指从 SCL 线上时钟的下降沿开始到 SDA 线上数据稳定这段时间。ICR 就是用来确定 SDA 保持时间的。 SDA 保持时间=CPU 总线周期×SDA 保持时间 表格 13-3 提供了 ICR 中不同的值所对应的 SCL 分频因子和 SDA 保持时间。这些值可用于设置 IIC 波特率和 SDA 保持时间。例如： CPU 总线频率=8MHz MULT 设置为 01(增频因子值=2)

	<p>要想得到的 IIC 总线波特率=100kbps</p> <p>IIC 总线波特率=CPU 总线频率(Hz)/(增频因子值×分频因子) 100000=8000000/(2×分频因子) SCL 分频因子=40</p> <p>表格 13-3 说明了 SCL 分频因子为 40 时 ICR 必须被设为 0B, 此时将会得到 SDA 保持值为 9。</p> <p>SDA 保持时间=CPU 总线周期×SDA 保持时间 SDA 保持时间=1/800000×9=1.125 μs</p> <p>如果产生的 SDA 保持值不可靠, MULT 用来改变 ICR。这将导致产生一个不同的 SDA 保持值。</p>
--	--

表格 13-3. IIC 分频因子和保持值

ICR(hex)	SCL 分频因子	SDA 保持值	ICR(hex)	SCL 分频因子	SDA 保持值
00	20	7	20	160	17
01	22	7	21	192	17
02	24	8	22	224	33
03	26	8	23	256	33
04	28	9	24	288	49
05	30	9	25	320	49
06	34	10	26	384	65
07	40	10	27	480	65
08	28	7	28	320	33
09	32	7	29	384	33
0A	36	9	2A	448	65
0B	40	9	2B	512	65
0C	44	11	2C	576	97
0D	48	11	2D	640	97
0E	56	13	2E	768	129
0F	68	13	2F	960	129
10	48	9	30	640	65
11	56	9	31	768	65
12	64	13	32	896	129
13	72	13	33	1024	129
14	80	17	34	1152	193
15	88	17	35	1280	193
16	104	21	36	1536	257
17	128	21	37	1920	257
18	80	9	38	1280	129
19	96	9	39	1536	129
1A	112	17	3A	1792	257
1B	128	17	3B	2048	257
1C	144	25	3C	2304	385

1D	160	25	3D	2560	385
1E	192	33	3E	3072	513
1F	240	33	3F	3840	513

13.3.3 IIC 控制寄存器(IIC1C)

	7	6	5	4	3	2	1	0
读	IICEN	IICIE	MST	TX	TXAK	0	0	0
写						RSTA		
复位	0	0	0	0	0	0	0	0

=未生效的或者预留的

图 13-5. IIC 控制寄存器(IIC1C)

表 13-4. IIC1C 寄存器域描述

域	描述
7 IICEN	IIC 总线使能—IICEN 位决定 IIC 模块是否可激活。 0 IIC 总线模块不可用。 1 IIC 总线模块可使用。
6 IICIE	IIC 中断使能—IICIE 位决定 IIC 总线中断是否可请求。 0 IIC 总线模块的中断不可使用 1 IIC 总线模块的中断可使用
5 MST	主机模式选择—当 MST 位从 0 变为 1 时，在总线上将产生一个开始信号，而且主机模式被选择。当该位从 1 变为 0 时，在总线上将产生一个停止信号，同时工作模式由主动变为被动。 0 从机模式。 1 主机模式。
4 TX	发送模式选择—TX 位选择主动或者被动传输方向。在主机模式下，该位应根据要求的传输模式来设置。因此在地址传送过程中，该位将被置高。如果设备被设置为被动，则根据状态寄存器的 SRW 位可以对该位进行软件设置。 0 表示接收。 1 表示发送。
3 TXAK	传输确认使能—无论是主动还是被动接收，在数据确认周期内，该位能识别数据线 SDA 的值。 0 在接收完一个字节的的数据后设备将会向总线发送一个确认信号。 1 设备不会向总线发送确认信号。
2 RSTA	重新开始—如果当前总线是主机模式，则向该位写 1 将会在总线上产生一个重新开始条件。读取该位，该位总为 0。在错误的时间尝试重新开始，则该设备将丧失仲裁能力。

13.3.4 IIC 状态寄存器(IIC1S)

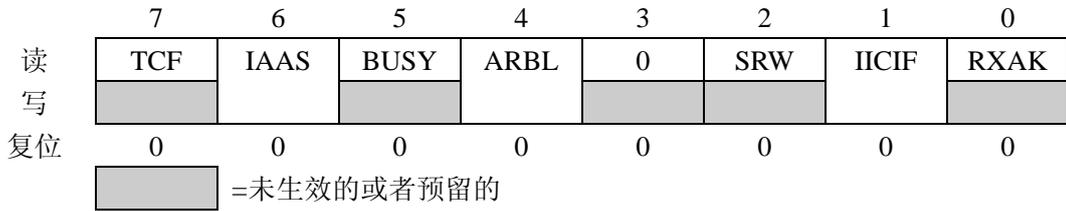


图 13-6. IIC 状态寄存器(IIC1S)

表 13-5. IIC1S 寄存器域描述

域	描述
7 TCF	<p>传输完成标志—该位在一个字节数据传输完成后被设置。注意该位只有在 IIC 模块进行输入或输出时才有效。当在接收模式时读 IIC1D 寄存器或者在发送模式时写 IIC1D 寄存器，TCF 位被清零。</p> <p>0 正在传输。 1 传输完成。</p>
6 IAAS	<p>被编址为从机—当 MCU 作为从机，并且他的地址与 IIC 总线上由主机发送的寻址地址一致时，IAAS 位被设置。当写 IIC1C 寄存器时，将清除该位。</p> <p>0 未编址 1 编址为从机</p>
5 BUSY	<p>总线忙—不论在主机模式还是从机模式下，BUSY 位都显示了总线的状态。当检测到一个开始信号时，BUSY 位被置 1。当检测到一个停止信号时，该位被清除。</p> <p>0 总线空闲状态。 1 总线忙。</p>
4 ARBL	<p>仲裁丢失—当仲裁处理丢失，该位被硬件置位。必须通过软件的方法向 ARBL 写 1 来清除该位。</p> <p>0 正常总线运行。 1 仲裁丢失。</p>
2 SRW	<p>从机读/写—当被设为从机时，SRW 位表明了从主机发送的地址中的 R/W 位的对应值。</p> <p>0 从机接收，主机写数据到从机。 1 从机发送，主机从从机读数据。</p>
1 IICIF	<p>IIC 中断标志—当中断操作被悬停 IICIF 位被设置。该位必须软件清除，在中断程序中通过写 1 来清除。当下面的情况发生时，IICIF 位置位：</p> <ul style="list-style-type: none"> ● 一字节完成传输 ● 编址为从机 ● 仲裁丧失 <p>0 无中断悬停 1 中断悬停</p>
0 RXAK	<p>接收应答—当 RXAK 位是低电平，则表明在总线上完整的 8 位数传输完毕后，应答信号已经被接收。如果 RXAK 位是高电平，则表明非应答信号被探测到。</p>

	0 接收应答信号。
	1 接收非应答信号。

13.3.5 IIC I/O 数据寄存器(IIC1D)

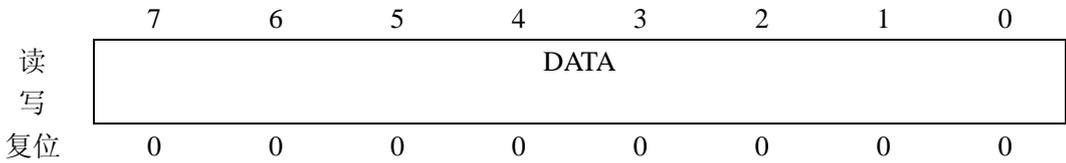


图 13-7. IIC I/O 数据寄存器(IIC1D)

表 13-6. IIC1D 寄存器域描述

域	描述
7:0 DATA	数据—在主机传输模式下，当数据写入 IIC1D 时，数据传输启动。最高位首先被发送出去。在主机接收模式下，读这个寄存器将启动下一字节数据的接收。

注：当主机接收模式传输结束时，应在读 IIC1D 寄存器前切换 IIC 模式，以便防止意外启动主机接收数据的传输。

在从机模式中，地址匹配发生后提供相同的功能。

注意 IIC1C 中的 TX 位必须正确的反映主机和从机所要求的转换方向，以便开始传输。例如，假如 IIC 已经设置为主动传输，这时又要求主动接收，那么读 IIC1D 就不会启动接收了。

无论 IIC 设置为主动接收还是被动接收模式，读 IIC1D 将返回最后一个接收的字节。IIC1D 不会反映出在 IIC 总线上传输的每一字节，没有一种软件方法能够通过读回的方式确认字节已经被正确的写入 IIC1D 中。

在主动传输模式下，第一字节的数据被写入 IIC1D，由 MST 来确定地址转换，并且是由调用地址(位于 7 位- 1 位)连接要求的 R /W 位(位于 0 位)组成。

13.4 功能描述

这部分提供了 IIC 模块的一个完整的功能描述。

13.4.1 IIC 协议

IIC 总线系统使用一条串行数据线(SDA)和一条串行时钟线(SCL)进行数据传输。所有设备连入必须打开对外接口或者输出口。两根线被接连外部上拉电阻通过一个逻辑与函数进行测试。这些电阻的阻值由系统决定。

一般情况下，一个标准的 IIC 通讯包含以下四部分：

- 开始信号
- 从机地址传输
- 数据传输
- 停止信号

停止信号不应与 CPU 的停止命令相混淆。IIC 总线系统通讯在如下部分简明的描述并且图 13-8 进行图示举例。

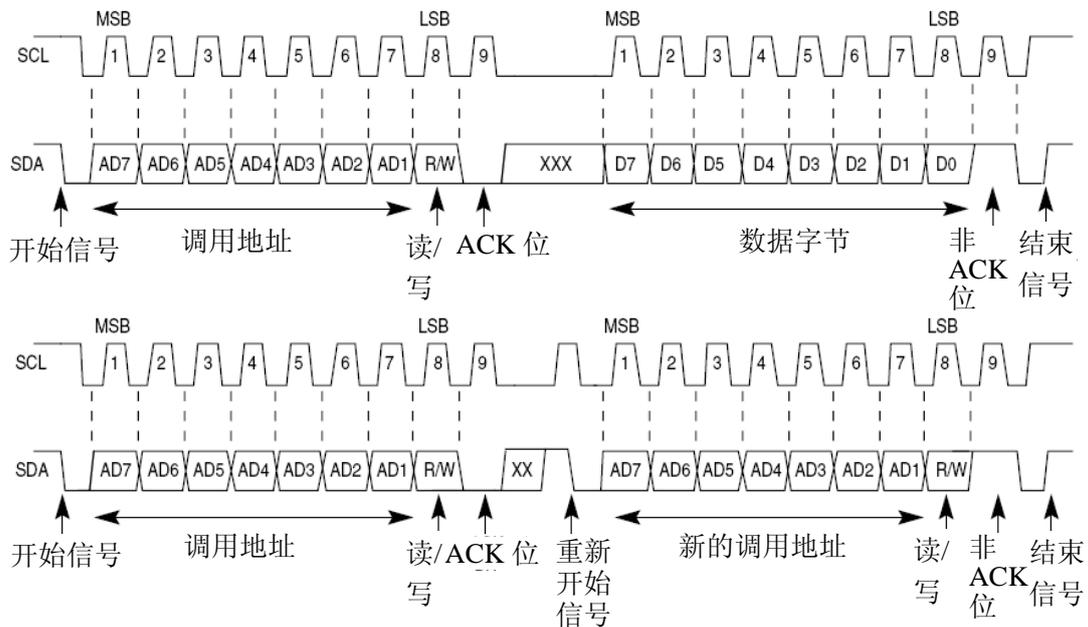


图 13-8. IIC 总线传输信号

13.4.1.1 开始信号

当总线空闲时，例如没有主机设备在使用总线(SCL 和 SDA 都处于高电平)，主机通过发送开始(START)信号建立通信。当 SCL 为高电平时，SDA 由高电平向低电平跳变，产生开始信号，如图 13-8 所示。该信号表示一个新的数据传输开始(各个数据传输可能包含一些字节的数据)，并且将所有的从机从空闲状态切换出来。

13.4.1.2 从机地址传输

在开始信号后立即传输的数据的第一个字节是由主机发送的从机地址。该字

节由从机的 7 位地址位后跟一位 R/W 位。R/W 位告诉从机想要的数据传输方向。

1=读传输，主机从从机读取数据。

0=写传输，逐级将数据写入从机。

仅当从机的调用地址与主机发送的地址相匹配时，从机才发送回 ACK 位来做响应。通过在第九个时钟周期拉低 SDA 来实现(见图 13-8)。

在 IIC 总线系统中，没有两个从机的地址是相同的。如果 IIC 模块为主机，主机不应该传送一个与其本身的从机地址相同的地址。IIC 不能同时充当主机和从机。然而，如果在寻址周期中仲裁丢失，IIC 将恢复从机模式，并且即使正在被另一个主机寻址，也能够正常运作。

13.4.1.3 数据传输

在从机寻址完成前，数据传输按照由主机发送 R/W 位要求的指定方向逐字节传输。

在一个寻址周期后所有的传输都涉及到数据转换，即使对从机设备来说这些数据带有附属地址。

一个字节数据有 8 位。仅当 SCL 为低电平时，数据才能改变，并且当 SCL 为高电平时，数据必须保持稳定的电平状态见图 13-8 所示。每一个数据位在 SCL 上都有一个时钟脉冲相对应，从数据字节的最高位开始传输。每个数据字节跟在由接收设备发出的八位后的第九位开始。通过在第九个时钟周期时拉低 SDA 位来产生一个应答信号。总而言之，一个完整数据传输需要九个时钟脉冲。

如果作为接收方的从机没有在第九位的时间响应主机，SDA 线必须被从机拉高。主机识别失败的应答作为一次不成功的数据传输。

如果作为接收方的主机在一个字节数据的传输后没有应答作为发送方的从机，从机认为此时数据传输结束，并且释放 SDA 线。

以上情况任意种发生时，数据传输失败并且主机会做以下操作之一：

- 通过产生一个停止信号来放弃总线
- 通过产生重新开始信号来开始一个新的调用

13.3.1.4 停止信号

主机可以通过产生一个停止信号到空闲总线上来停止通讯。然而，主机可能

通过命令调用直接产生一个开始信号，而不事先产生一个停止信号。这样的操作称为重新开始。一个停止信号被定义为当 SCL 在逻辑 1 时，SDA 由低电平到高电平跳变。

甚至在从机产生应答信号时，主机可以产生停止信号，关键点是从机必须释放总线。

13.4.1.5 重新开始信号

如图 13-8 所示，重新开始信号是主机在首次发送停止信号结束通讯前，发送一个开始信号。主机通过发送重新开始信号，可以转换当前从机的通讯模式(发送/接收模式)无需释放总线，或是切换到与另一个从机通信。

13.4.1.6 仲裁程序

IIC 总线是多主机总线，允许多个主机连接到该总线上。如果 2 个或者 2 个以上主机试图在同一时刻控制总线，则通过一个时钟同步的过程决定总线的时钟信号。时钟信号低电平时间与主机中最长的低电平时间相等，时钟信号高电平时间与主机中最短的高电平时间相等。多个主机的相关优先权由数据仲裁程序决定，当其他的主机发送逻辑 0 时，有一个主机发送逻辑 1，那么它将失去仲裁。失去仲裁的主机立刻转入被动接收模式，并且停止数据和时钟信号的输出。在这种情况下，传输模式从主控到被控的转变不会产生一个停止条件。同时，状态位由硬件来置位，表明该主机已经丧失了仲裁。

13.4.1.7 时钟同步

由于 SCL 线具有 AND 功能，因此 SCL 由高电平到低电平的跳变将影响到总线上的所有设备。这样设备开始计数它们的低电平周期，一旦某个设备的时钟脉冲变为低电平，它就控制了 SCL 线，直到时钟脉冲回到高电平状态。但是，如果其他的设备仍然在低电平周期，那么该设备的时钟脉冲从低到高的改变并不改变 SCL 的状态。因此，同步时钟线 SCL 在所有设备的最长低电平周期内持续低电平。在这段时间内，低电平周期相对较短的设备进入了高电平等待状态(见图 13-9)。当所有相关设备退出低电平时，同步时钟线 SCL 就被释放，并

且进入高电平。然后，所有的设备和 SCL 都开始计数高电平周期。第一个完成高电平周期的设备再次将 SCL 拉低。

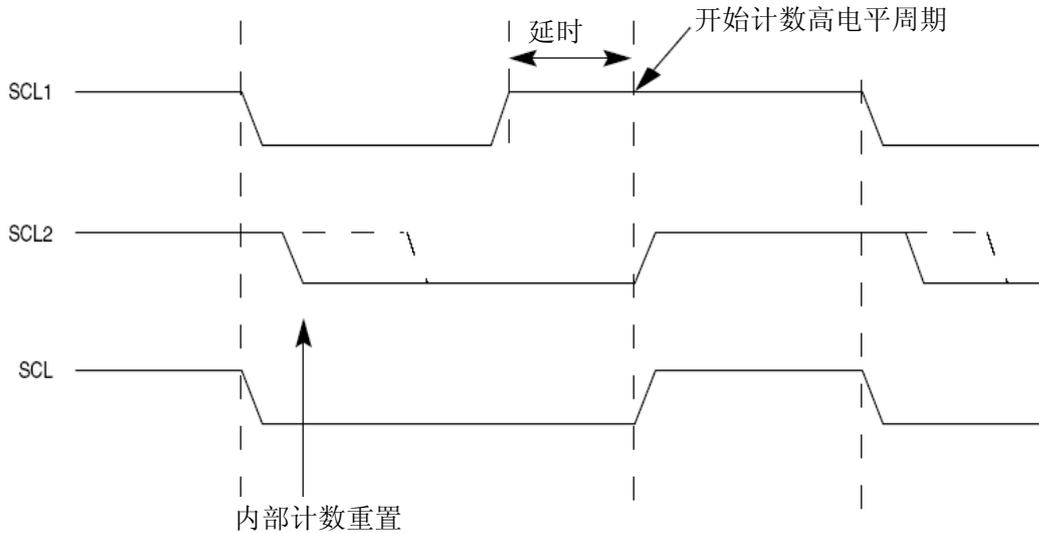


图 13-9. IIC 时钟同步

13.4.1.8 握手

时钟同步机制在数据传输时可以起到握手的作用。在完成 1 字节(9 位)的传输后，从机可以将 SCL 拉低。在这种情况下，总线时钟停止，并且强迫主控时钟进入等待状态，直到从机释放 SCL 线。

13.4.1.9 时钟周期延长

时钟同步机制可被从机用来降低传输速率。在主机将 SCL 拉低以后，从机将 SCL 拉低至所求的周期，并且释放它。如果从机的 SCL 低电平周期大于主机的 SCL 低电平周期，那么 SCL 的低电平周期就可以延长了。

13.5 重启

在系统重启后 IIC 总线是禁止的。IIC 总线无法产生 MCU 重置。

13.6 中断

IIC 总线产生一个个别的中断。

当表 13-7 中任意事件发生导致 IICIE 位被设置，那么 IIC 总线会产生一个中

断。该中断由 IIC 状态寄存器中的 IICIF 位驱动, 并且由 IIC 控制寄存器中的 IICIE 位进行控制。在中断事务中, IICIF 位必须通过软件方式写入 1 来清除。用户可以通过读出状态寄存器来测定中断类型。

表 13-7. 中断概要

中断来源	状态寄存器	标志位	本地始能
1 字节完全传输	TCF	IICIF	IICIE
与接收端地址匹配	IAAS	IICIF	IICIE
仲裁丧失	ARBL	IICIF	IICIE

13.6.1 字节传输中断

TCF(传输完成标志)位在第九个时钟周期的下降沿被置位, 表明字节传输完成。

13.6.2 地址检测中断

当从机中的调用地址与主机编程的从机地址(IIC 地址寄存器)相匹配时, 状态寄存器中的 IAAS 位被置位。如果 IICIE 被置位则 CPU 被中断。CPU 必须检查 SRW 位并且根据 SRW 位来设置其收发模式。

13.6.3 仲裁丧失中断

IIC 总线是多主机总线, 允许多个主机连接到该总线上。如果 2 个或者 2 个以上主机试图在同一时刻控制总线, 则通过一个数据仲裁的过程决定多个主机的相关优先权。当 IIC 模块丧失其数据仲裁过程时, IIC 模块声明仲裁丧失中断并且置位状态寄存器中的 ARBL 位。

在如下情况下, IIC 仲裁丧失:

- 当主机在寻址或者数据传输周期驱动为高电平, SDA 采样为低电平。
- 当主机在数据接收的应答位周期驱动为高电平, SDA 采样为低电平。
- 在总线忙时试图产生一个开始信号。
- 在从机模式下请求重新开始信号周期。
- 当主机未发出请求时, 监测到停止信号条件。

这些位必须通过软件写 1 来清除。

13.7 初始化/应用资料

模块初始化(从机)

1. 写: IICA
 - 设置从机地址
2. 写: IICC
 - 始能 IIC 和中断
3. 为传输数据初始化 RAM 变量(IICEN=1 和 IICIE=1)
4. 初始化 RAM 变量用于完成如图 13-11 所示的事务

模块初始化(主机)

1. 写: IICF
 - 设置 IIC 波特率(本章中提供的例子)
2. 写: IICC
 - 始能 IIC 和中断
3. 为传输数据初始化 RAM 变量(IICEN=1 和 IICIE=1)
4. 初始化 RAM 变量用于完成如图 13-11 所示的事务
5. 写: IICC
 - 始能收发模式位
6. 写: IICC
 - 始能主机模式选择位(主机模式)
7. 写: IICD
 - 和目标从机地址相一致。(这个字节的最低位决定通讯是主机接收还是主机发送。)

模块使用

如图 13-11 所示的事务既可以处理主机的 IIC 操作, 也可以处理从机的 IIC 操作。对于从机的操作, 一个包含了正确地址的 IIC 入口信息将会开始 IIC 通讯。对于主机的操作, 必须通过写 IICD 寄存器来初始化通讯。

寄存器模式

IICA	ADDR	0
------	------	---

当作为从机寻址时(从机模式), 寻址到模块将响应处

IICF	MULT	ICR
------	------	-----

波特率=总线时钟频率/(2×增频因子值×分频因子)

IICC	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
------	-------	-------	-----	----	------	------	---	---

模块配置

IICS	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
------	-----	------	------	------	---	-----	-------	------

模块状态标志

IICD	DATA
------	------

数据寄存器; 写操作时写入 IIC 传输数据, 读操作时读出 IIC 数据

图 13-10. IIC 模块快速启动

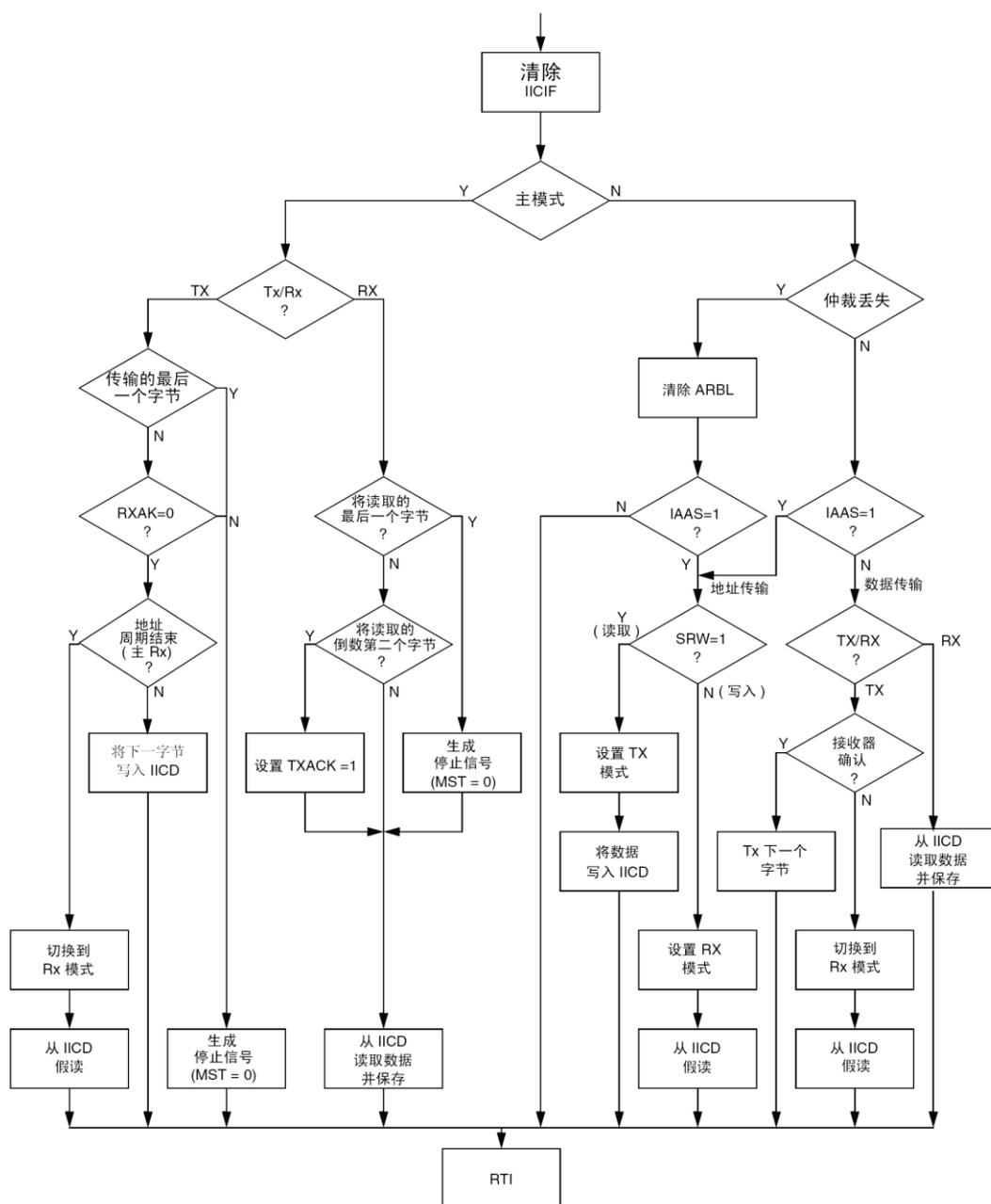


图 13-11. 典型的 IIC 模块中断程序

第十四章 模数转换器(S08ADC10V1)

14.1 概述

10 位的模数转换器(ADC)是一种逐次逼近的模数转换器,为集成于单片微控制器片上系统设计。ADC 模块设计支持最多到 28 个单独的模拟量输入

(AD0-AD27)。在 MC9S08AW60 系列的 MCU 中，只有 18 个(AD0-AD15, AD26 和 AD27)可能输入端可执行。这些输入由 ADCH 位来选择。一些输入端与 I/O 引脚复用，详见图 14-1。MC9S08AW60 系列设备的所有 ADC 通道分配概述见表 14-1。

14.2 通道分配

MC9S08AW60 系列设备的 ADC 通道分配入下表所示。未生效的通道连接在 V_{REFL} 内。保留的通道转换成一个未知值。连接到 I/O 引脚的通道显示一个关联的引脚控制。

表 14-1. ACD 通道分配

ADCH	通道	输入	引脚控制	ADCH	通道	输入	引脚控制
00000	AD0	PTB0/ADC1P0	ADPC0	10000	AD16	V_{REFL}	N/A
00001	AD1	PTB1/ADC1P1	ADPC1	10001	AD17	V_{REFL}	N/A
00010	AD2	PTB2/ADC1P2	ADPC2	10010	AD18	V_{REFL}	N/A
00011	AD3	PTB3/ADC1P3	ADPC4	10011	AD19	V_{REFL}	N/A
00100	AD4	PTB4/ADC1P4	ADPC3	10100	AD20	V_{REFL}	N/A
00101	AD5	PTB5/ADC1P5	ADPC5	10101	AD21	V_{REFL}	N/A
00110	AD6	PTB6/ADC1P6	ADPC6	10110	AD22	预留	N/A
00111	AD7	PTB7/ADC1P7	ADPC7	10111	AD23	预留	N/A
01000	AD8	PTD0/ADC1P8	ADPC8	11000	AD24	预留	N/A
01001	AD9	PTD1/ADC1P9	ADPC9	11001	AD25	预留	N/A
01010	AD10	PTD2/ADC1P10 /KBI1P5	ADPC10	11010	AD26	温度传感器 ¹	N/A
01011	AD11	PTD3/ADC1P11 /KBI1P6	ADPC11	11011	AD27	内部隙带	N/A
01100	AD12	PTD4/ADC1P12 /TPM2CLK	ADPC12	11100	-	预留	N/A
01101	AD13	PTD5/ADC1P13	ADPC13	11101	V_{REFH}	V_{REFH}	N/A
01110	AD14	PTD6/ADC1P14 /TMP1CLK	ADPC14	11110	V_{REFL}	V_{REFL}	N/A
01111	AD15	PTD7/ADC1P15 /KBI1P7	ADPC15	11111	模块禁止	无	N/A

¹ 更多的资料见 14.2.3 节，“温度传感器”。

注：选择内部隙带通道要求 SMPSC1 寄存器中 BGBE=1，见 5.9.8 节，“系统电源管理状态和控制寄存器 1(SMPSC1)”。隙带参考电压值见表 A-7.部分，“DC 特性”。

14.2.1 替代时钟

ADC 模块的可以转换使用 MCU 总线时钟，总线时钟可为模块内的本地异

步时钟(ADACK)或替代时钟 ALTCLK。MC9S08AW60 系列 MCU 设备的替代时钟是由内部时钟发生器(ICG)模块产生的外部参考时钟(ICGERCLK)。

由于 ICGERCLK 仅当外部时钟源使能时才可用，所以 ICG 必须配置为 FEB 或者 FEE 模式(CLKS1=1)。ICGERCLK 所选的时钟源必须运行在一定的频率范围内，ADC 转换时钟(ADCK)就可以通过 ADIV 位的设置，在从 ALTCLK 输入分频后，运行在指定频率范围(f_{ADCK})内。例如，假设 ADIV 位设置为分频是 4，那么 ALTCLK(ICGERCLK)的最小频率是 f_{ADCK} 最小值的 4 倍，最大频率是 f_{ADCK} 的最大值的 4 倍。由于最小频率要求，所以当使用振荡器电路时必须设置为高范围运行(RANGE=1)。

如果满足以上条件，当 MCU 处于等待模式时，ALTCLK 是使能的。这使得当 MCU 处于等待模式时，ALTCLK 可以用作 ADC 转换时钟源。当 MCU 处于 stop3 时，ALTCLK 不能用作 ADC 转换时钟源。

14.2.2 硬件触发器

ADC 硬件触发器，ADHWT，是实时时钟中断计数器(RTI)的输出。RTI 计数器可以使用 ICGERCLK 或 RTI 时钟内标称的 1kHz 时钟源。1kHz 时钟源在 MCU 运行，等待和 stop3 模式下可用。在 ICG 配置为 FBE 或 FEE 模式时，ICGERCLK 可用于 MCU 运行和等待模式。

RTI 的周期由输入时钟频率和 RTIS 位决定。当 ADC 硬件触发器使能时，RTC 计数器溢出触发 ADC 转换。RTI 计数器是一个自由运行计数器，按照 RTIS 位决定的 RTI 率产生一个溢出。

14.2.2.1 模拟引脚使能

MC9S08AW60 系列的 ADC 模块只包含两个模拟引脚使能寄存器：APCTL1 和 APCTL2。

14.2.2.2 低功耗模式操作

ADC 可在 stop3 模式下运行，但是需 SPMSC1 中的 LVDSE 和 LVDE 被置位。

14.2.3 温度传感器

ADC1 模块包含一个温度传感器，其输出与 ADC 模拟通道输入连接。等式 14-1 提供了一个温度传感器的近似传递函数。

$$\text{Temp} = 25 - ((V_{\text{TEMP}} - V_{\text{TEMP25}}) \div m) \quad \text{公式. 14-1}$$

其中：

— V_{TEMP} 是在当前环境温度下温度传感器通道的电压。

— V_{TEMP25} 是 25°C 时的温度传感器通道的电压。

— m 是以 V/°C 表示的热或冷电压与温度斜率比较值。

如需进行温度计算，请使用 ADC Electricals 表中的 V_{TEMP25} 值和 m 值。

在应用代码中，用户读取温度传感器通道，计算 V_{TEMP} 值并与 V_{TEMP25} 进行比较。如果 V_{TEMP} 大于 V_{TEMP25} ，冷端斜率就应用于等式 14-1。如果 V_{TEMP} 小于 V_{TEMP25} ，热端斜率就应用于等式 14-1。

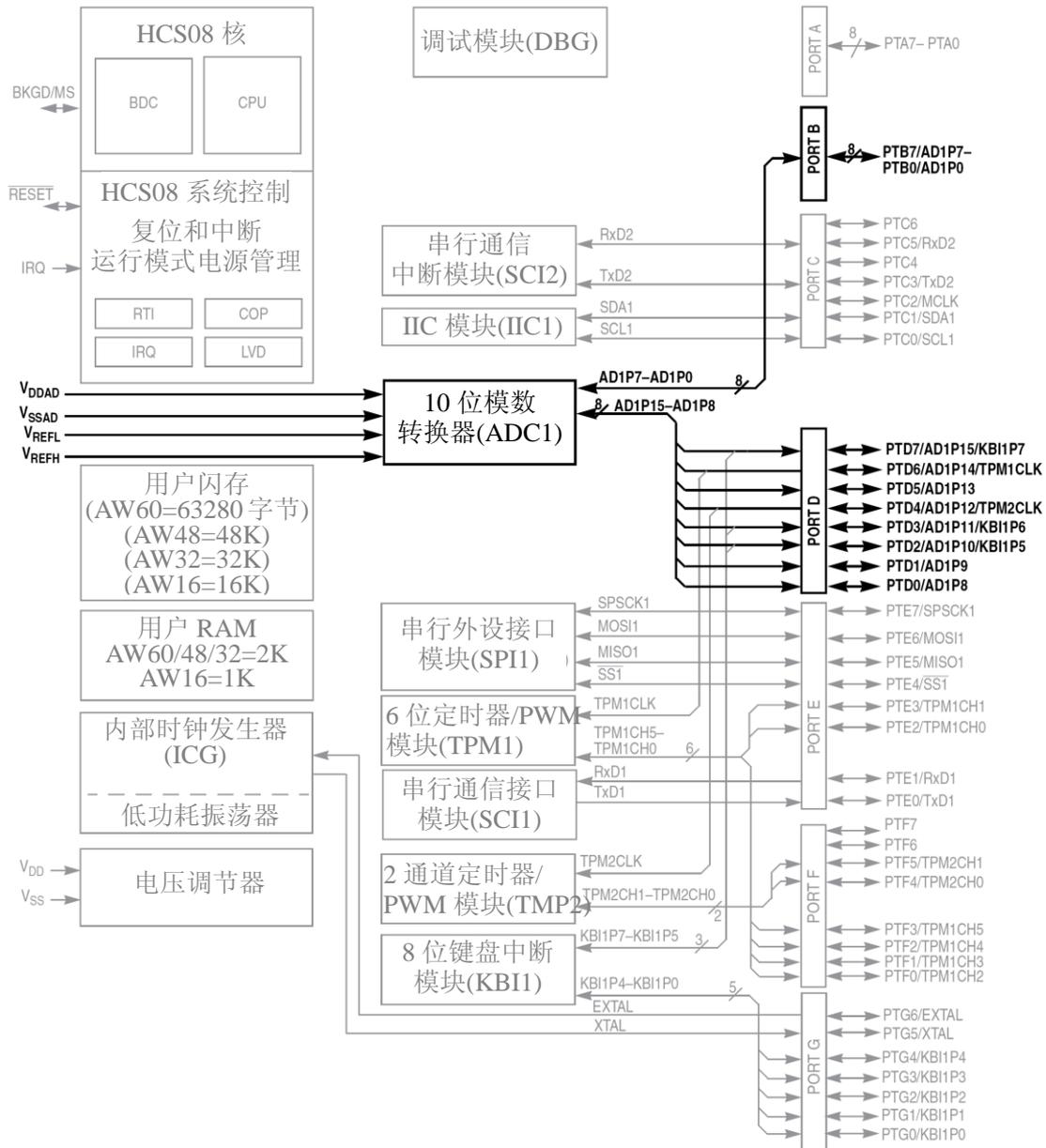


图 14-1. 高亮显示了 ADC 模块引脚的 MC9S08AW60 结构图

注:

1. 端口引脚如果作为输入口，可以软件设置为拉高装置。
2. 如果 IRQ 使能(IRQPE=1)，引脚可软件设置为拉高/拉低装置。如果上升沿检测被选择(IRQEDG=1)使能拉高设备。
3. IRQ 没有箝位二极管连接到 V_{DD}。IRQ 低于 V_{DD} 不应该被驱动。
4. 引脚包含完整的拉高装置。
5. 引脚 PTD7, PTD3, PTD2 和 PTG4 包含拉高和拉低装置。当 KBI 使能(KBIPE_n=1)并且选择上升沿(KBEDG_n=1)时，使能拉低设备。

14.2.4 特性

ADC 模块的特性包括:

- 具有 10 位分辨率的线性逐次逼近算法。
- 高达 28 个模拟输入。
- 10 位或八位右对齐输出格式。
- 单次转换或连续转换(单次转换后自动返回空闲状态)。
- 采样时间和转换速度/功率可配置。
- 最多可选择 4 个输入时钟源。
- 在等待或 stop3 模式下实现了低噪音运行。
- 异步时钟源实现了低噪音运行。
- 可选的异步硬件转换触发。
- 与小于, 大于或等于可编程值自动比较的中断。

14.2.5 结构图

图 14-2 是 ADC 模块的结构图。

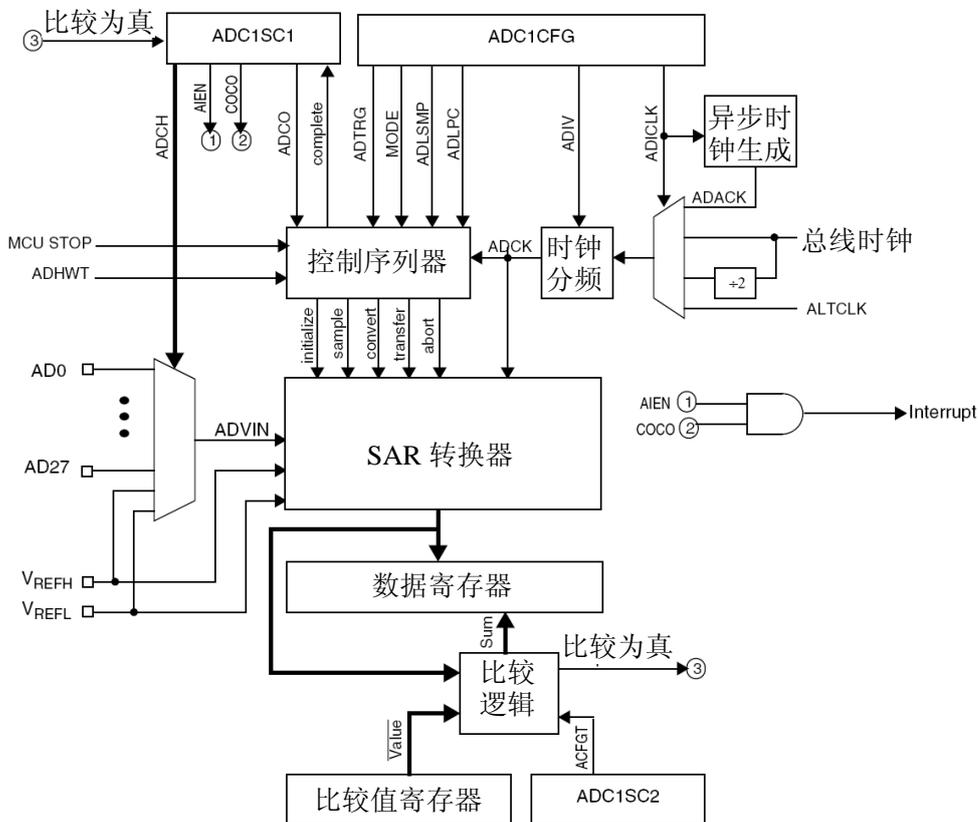


图 14-2. ADC 结构图

14.3 外部信号描述

ADC 模块支持高达 28 个独立模拟输入。它还需要 4 个电源/参考/接地连接。

表 14-2. 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}	地参考电压
V_{DDAD}	模拟电源
V_{SSAD}	模拟接地

14.3.1 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} 在内部连接到 V_{DD} 。如果外部电压可用, 则连接到 V_{DDAD} 引脚的外部电压作为 V_{DD} 。为了确保干净的 V_{DDAD} 信号, 可能还需要外部滤波。

14.3.2 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} 在内部连接到 V_{SS} 。如果外部电压可用, 则连接到 V_{SSAD} 引脚的外部电压作为 V_{SS} 。

14.3.3 参考电压高(V_{REFH})

V_{REFL} 是 AD 转换的高参考电压。在一些封装中, V_{REFH} 在内部连接到 V_{DDAD} , 如果外部电压可用, 则连接到 V_{REFH} 引脚的外部电压作为 V_{DDAD} , 或者连接到在最小 V_{DDAD} 和可能的 V_{DDAD} 之间(V_{REFH} 决不能超过 V_{DDAD})。

14.3.4 参考电压低(V_{REFL})

V_{REFL} 是转换器的低参考电压, 在相同的封装中, V_{REFL} 在内部连接到 V_{SSAD} , 如果外部电压可用, 连接到 V_{REFL} 的引脚和 V_{SSAD} 有相同电压。

14.3.5 模拟通道输入(V_{DDAD})

ADC 模块支持高达 28 个独立的模拟输入。通过 ADCH 通道选择位，来选择被转换的输入通道。

14.4 寄存器定义

这些内存映像寄存器控制和管理着 ADC 的操作：

- 状态和控制寄存器，ADC1SC1
- 状态和控制寄存器，ADC1SC2
- 数据结果寄存器，ADC1RH 和 ADC1RL
- 比较值寄存器，ADC1CVH 和 ADC1CVL
- 配置寄存器，ADC1CFG
- 引脚使能寄存器，APCTL1, APCTL2, APCTL3

14.4.1 状态和控制寄存器 1(ADC1SC1)

本节描述 ADC 状态和控制寄存器(ADC1SC1)的功能。写 ADCSC1 可以终止当前的转换并初始化一个新的转换(如果 ADCH 等于一个不是全 1 的值)。

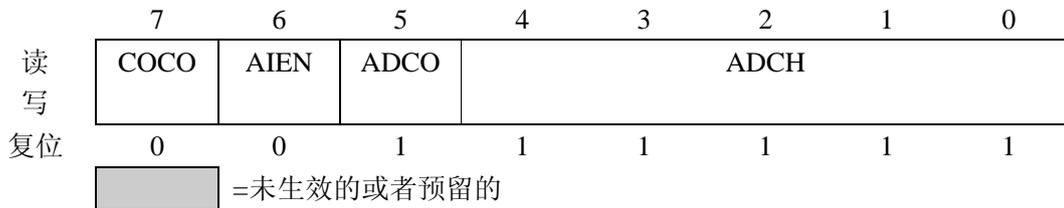


图 14-3. 状态和控制寄存器 1(ADC1SC1)

表 14-3. ADC1SC1 寄存器域描述

域	描述
7 COCO	转换完成标志—COCO 标志是一个只读位。当比较功能关闭(ACFE=0)时，每次转换完成时置位。当比较功能允许(ACFE=1)时，转换完成后，只要比较结果为真，则 COCO 置位。只要写 ADC1SC1 或读 ADC1RL，该位清零。 0 转换未完成。 1 转换完成。
6 AIEN	中断使能—AIEN 用于使能转换完成中断。当 AIEN 为高，COCO 置位时，确认一个中断。 0 关闭转换完成中断。 1 使能转换完成中断。

5 ADCO	连续转换使能—ADCO 用于使能连续转换。 0 在一个触发操作 1 之后产生一个转换。 1 在一个触发操作被选择后初始化连续转换。
4: 0 ADCH	输入通道选择—ADCH 包括 5 位，用于选择输入通道中的一个。输入通道在图 14-4 中描述。 当通道选择位设置为全 1 时，逐次逼近转换器子系统关闭。这个特点允许禁止 ADC 和从所有的输入源中隔离输入通道。 使用这种方法来终止连续转换可以防止在转换完成时产生一个额外的信号转换。当连续转换被关闭时，不必通过设置通道位为全 1 来使 AD 转换进入到低功耗模式，因为此模块在转换完成时自动进入低功耗模式。

ADCH	输入选择	ADCH	输入选择
00000	AD0	10000	AD16
00001	AD1	10001	AD17
00010	AD2	10010	AD18
00011	AD3	10011	AD19
00100	AD4	10100	AD20
00101	AD5	10101	AD21
00110	AD6	10110	AD22
00111	AD7	10111	AD23
01000	AD8	11000	AD24
01001	AD9	11001	AD25
01010	AD10	11010	AD26
01011	AD11	11011	AD27
01100	AD12	11100	保留
01101	AD13	11101	VREFH
01110	AD14	11110	VREFL
01111	AD15	11111	模块被关闭

图 14-4. 输入通道选择

14.4.2 状态和控制寄存器 2(ADC1SC2)

	7	6	5	4	3	2	1	0
读	ADACT	ADTRG	ACFE	ACFGT	0	0	R ¹	R ¹
写								
复位	0	0	0	0	0	0	0	0

■ =未生效的或者预留的

¹ 位 1 和位 0 为保留位，它们必须一直为 0。

图 14-5. 状态和控制寄存器 2(ADC1SC2)

表 14-4. ADC1SC2 寄存器域描述

域	描述
7 ADACT	转换行为—ADACT 表示转换正在进行中。当初始化转换时，ADACT 置位；当转换完成或终止时，ADACT 清零。 0 转换不在进行。 1 转换处理中。
6 ADTRG	转换触发选择—ADTRG 用于选择初始化转换的触发的类型。两种触发类型可选：软件触发和硬件触发。当选择软件触发时，写入 ADC1SC1 就能发起转换。当选择硬件触发时，ADHWT 触发后就能发起转换。 0 选择软件触发。 1 需安装硬件触发。
5 ACFE	比较功能使能—ACFE 用于使能比较功能。 0 关闭比较功能。 1 使能比较功能。
4 ACFGT	比较功能更大使能—ACFGT 用于设置比较功能的触发。默认的比较功能为在当前的转换结果小于比较值时触发。 0 当输入小于比较电平时，比较触发。 1 当输入大于或等于比较电平时，比较触发。

14.4.3 数据结果高位寄存器(ADC1RH)

在 10 位模式中，ADC1RH 包含 10 位转换结果中的高 2 位。在配置 8 位模式时，ADR8 和 ADR9 等于 0。每次完成转换时 ADCRH 都被更新，除非使能了自动比较但不满足比较条件。在 10 位模式中，读取 ADC1RH 能够防止在读取 ADC1RL 之前 ADC 将后续转换结果传输到结果寄存器。如果 ADC1RL 是在下一次转换完成后才被读取，那么中间转换结果会被丢失。在 8 位模式中，没有与 ADC1RL 的互锁。当 MODE 位被更改时，ADC1RH 中的任何数据都将无效。

	7	6	5	4	3	2	1	0
读	0	0	0	0	0	0	ADR9	ADR8
写								
复位	0	0	0	0	0	0	0	0

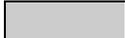
 =未生效的或者预留的

图 14-6. 数据结果高位寄存器(ADC1RH)

14.4.4 数据结果低地址寄存器(ADC1RL)

ADC1RL 包含 10 位转换结果的低 8 位，或者是一个 8 位转换的所有位。每次转换完成，这个寄存器都被更新，除非自动比较被允许而且不满足比较结果。在 10 位数据模式下，读取 ADC1RH 能够防止在读取 ADC1RL 之前 ADC 将后续转换结果传输到结果寄存器。如果 ADC1RL 是在下一次转换完成后才被读取，那么中间转换结果会被丢失。在 8 位模式中，没有与 ADC1RH 的互锁。当 MODE 位被更改时，ADC1RL 中的任何数据都将无效。

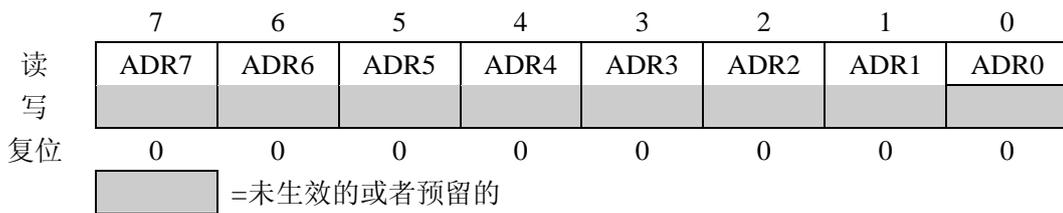


图 14-7. 数据结果低位寄存器(ADC1RL)

14.4.5 比较值高地址寄存器(ADCCVH)

该寄存器包含了 10 位比较值的高 2 位。当使能比较功能时，这些位和 10 位模式中的转换结果的高 2 位比较。在 8 位操作中，ADCCVH 在比较过程中不使用。

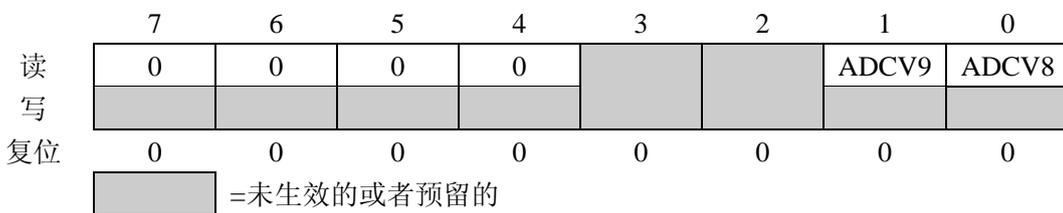


图 14-8. 比较值高地址寄存器(ADCCVH)

14.4.6 比较值低位寄存器(ADCCVL)

该寄存器包含了 10 位比较值的低 8 位，或者 8 位比较值的所有位。在 10 位或 8 位模式中，ADCV7:ADCV0 与转换结果的低 8 位比较。

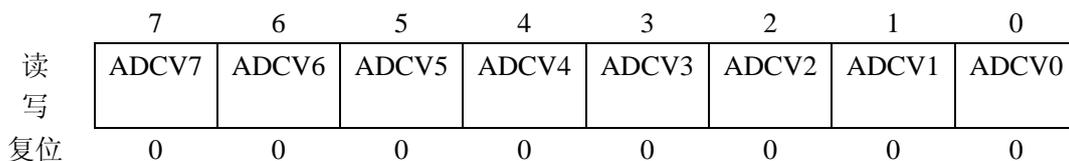


图 14-3. 状态和控制寄存器 1(ADC1SC1)

14.4.7 配置寄存器(ADC1CFG)

ADC1CFG 用于选择操作模式，时钟源，时钟分频和低功耗或长采样时间的设置。

	7	6	5	4	3	2	1	0
读	ADLPC	ADIV		ADLSMP	MODE		ADICK	
写								
复位	0	0	0	0	0	0	0	0

图 14-3. 状态和控制寄存器 1(ADC1CFG1)

表 14-4. ADC1CFG2 寄存器域描述

域	描述
7 ADLPC	低功耗配置—ADLPC 控制逐次逼近转换器的速度和功率配置。当不需要更高采样率时，ADLPC 可以用来优化功耗。 0 高速配置 1 低功耗配置：{FC31}功率的降低以最大时钟速度为代价。
6:5 ADIV	时钟分频选择—ADIV 选择 ADC 生成内部时钟 ADCK 所使用的分频率。 表 14-6 显示了可用时钟配置。
4 ADLSMP	长采样时间配置—ADLSMP 选择长采样时间和短采样时间。这样可以调节采样时间，实现对高抗阻输入的精确采样，或者最大限度地提高对低抗阻输入的转换速度。如果在连续转换模式下不需要高转换速率，更长的采样时间还可以用来降低整体功耗。 0 短采样时间 1 长采样时间
3:2 MODE	转换模式选择—MODE 位用来选择是 10 或 8 位转换。见表 14-7。
1:0 ADICK	输入时钟选择—ADICK 位选择生成内部时钟 ADCK 的输入时钟源。见表 14-8。

表 14-6. 时钟分频选择

ADIV	分频因子	时钟
00	1	输入时钟
01	2	2 分频输入时钟
10	4	4 分频输入时钟
11	8	8 分频输入时钟

表 14-7. 转换模式

模式	模式描述
00	8 位转换(N=8)
01	保留

10	10 位转换(N=10)
11	保留

表 14-8. 转换模式

ADICK	时钟源选择
00	总线时钟
01	二分频总线时钟
10	交替时钟(ALTCLK)
11	异步时钟(ADACK)

14.4.8 引脚控制寄存器 1(APCTL1)

引脚控制寄存器用于关闭对模拟输入的 MCU 引脚作为 I/O 口。APCTL1 用于控制这些引脚和 ADC 模块通道 0-7 的连接。

	7	6	5	4	3	2	1	0
读	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
写								
复位	0	0	0	0	0	0	0	0

图 14-11. 引脚控制寄存器 1(APCTL1)

表 14-9. APCTL1 寄存器域描述

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

1 ADPC1	ADC 引脚控制 1—ADPC1 用来控制与通道 AD1 连接的引脚。 0 AD1 引脚 I/O 控制使能 1 AD1 引脚 I/O 控制禁止
0 ADPC0	ADC 引脚控制 0—ADPC0 用来控制与通道 AD0 连接的引脚。 0 AD0 引脚 I/O 控制使能 1 AD0 引脚 I/O 控制禁止

14.4.9 引脚控制寄存器 2(APCTL2)

APCTL2 用于控制和 ADC 模块的通道 8-15 相关的位。

	7	6	5	4	3	2	1	0
读	ADPC	ADPC	ADPC	ADPC	ADPC	ADPC	ADPC9	ADPC8
写	15	14	13	12	11	10		
复位	0	0	0	0	0	0	0	0

图 14-12. 引脚控制寄存器 2(APCTL2)

表 14-10. APCTL2 寄存器域描述

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

14.4.10 引脚控制寄存器 3(APCTL3)

APCTL3 用于控制和 ADC 模块的通道 16-23 相关的位。

	7	6	5	4	3	2	1	0
读	ADPC							
写	23	22	21	20	19	18	17	16
复位	0	0	0	0	0	0	0	0

图 14-13. 引脚控制寄存器 3(APCTL3)

表 14-11. APCTL1 寄存器域描述

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

14.5 功能描述

当复位或 ADCH 位全高时，ADC 模块被关闭。当一个转换完成而另一个转

换还未开始时，该模块是空闲的。空闲时，模块处于最小功耗状态。

ADC 可以通过软件选择任何一个通道进行模数转换。选择的通道电压可以通过逐次渐进算法转换成 11 位数字的结果。在 8 位模式中，选择的通道电压可以被逐次渐进算法转换成 9 位数字的结果。

当转换完成，结果放在数据寄存器中(ADC1RH 和 ADC1RL)。在 10 位模式中，结果四舍五入成 10 位放在 ADCRH 和 ADCRL 中。在 8 位模式中，结果四舍五入成 8 位放在 ADCRL 中。转换完成标志置 1，同时使能转换完成中断位(AIEN=1)，则产生一个中断。

ADC 模块能够自动比较转换结果和比较寄存器的内容。通过设置 ACFE 位并结合任意一种转换模式和配置一起运行，就使能了比较功能。

14.5.1 时钟选择和分频控制

可以选择 4 个时钟源中的一个作为 ADC 模块的时钟源。这个时钟源通过一个设置值分频后就可以产生转换器的输入时钟(ADCK)。时钟选择下面源中的一个，由 ADICLK 位决定。

- 总线时钟，等于软件运行的频率。这是复位后的默认选择。
- 总线时钟 2 分频，如果总线时钟很高，允许总线时钟最大为 16 分频。
- ALTCLK，由 MCU 定义(参见相应模块部分的介绍)。
- 异步时钟(ADACK)—该时钟由 ADC 模块内部的时钟源产生。当选择这个时钟源时，若 MCU 处于等待或停止模式时，该时钟仍有效，允许在这些模式中以更低的噪音操作来进行转换。

无论选择哪个时钟，其频率都必须在规定的 ADCK 频率范围内。如果可用时钟太慢，ADC 将无法保证正常工作。如果可用时钟太快，时钟必须分频到适当的频率。分频因子由 ADIV 位决定，可以进行 1、2、4、8 分频。

15.5.2 输入选择和引脚控制

引脚控制寄存器(APCTL3、APCTL2、APCTL1)用来禁止作为模拟输入的引脚的 I/O 控制功能。当置位引脚控制寄存器相应位时，对应的 MCU 引脚将会服从接下来的条件：

- 输出缓冲器进入高阻抗状态。

- 输入缓冲器禁止。对于其输入缓冲器被禁止的任何引脚，I/O 端口读数均返回 0。

- 禁止上拉电阻。

14.5.3 硬件触发

ADC 模块有一个可选的异步硬件转换触发器，ADHWT，当 ADTRG 位被置位时，它被使能。这个源并不是在所有的 MCU 都有效。关于该 MCU 的 ADHWT 源的具体细节在参考模块介绍。

当 ADHWT 源可用并且硬件触发被使能(ADTRG=1)，在 ADHWT 的上升沿初始化转换。如果转换进行过程中再出现上升沿，该上升沿就被忽略。在连续转换模式下，只是触发连续转换的初始上升沿有效。硬件触发功能可以结合任意转换模式和配置一起运行。

14.5.4 转换控制

可以使用 8 位或 10 位模式转换，由 MODE 位决定。一个软件或硬件触发可以初始化转换。另外，ADC 模块可以设置为低功耗操作，长采样时间，连续采样，自动比较转换值和软件决定的比较值。

14.5.4.1 初始化转换

一个转换被初始化：

- 如果选择软件触发操作，在写 ADC1SC1 之后(ADCH 不是全 1)。
- 如果选择硬件触发操作，在一个硬件触发(ADHWT)事件之后。
- 当允许连续转换时，在将数据传到数据寄存器之后。

如果使能为连续转换，当前转换完成后，一个新的转换可以自动初始化。在软件触发操作，连续转换在写 ADC1SC1 后开始，并继续直到终止。在硬件触发操作，连续转换在硬件触发事件后开始，并继续直到终止。

14.5.4.2 完成转换

当转换的结果传到数据结果寄存器(ADC1RH 和 ADC1RL)后，转换完成。通

过置位 COCO 表示。如果 AIEN 是高，在 COCO 置位时会产生一个中断。

在 10 位模式中，如果数据正在被读(ADC1RH 已经被读但是 ADC1RL 还未被读)，拦截机制防止新数据不会覆盖 ADC1RH 和 ADC1RL 中原有的数据。当拦截机制处于工作状态时，数据传送被拦截，COCO 不能置位，新转换数据丢失。在允许比较功能的单个转换的情况下，并且比较条件为假，拦截不起作用，ADC 操作终止。在其他情况下，当数据传送被拦截，无论 ADCO 处于何种状态(使能单个或连续转换)，都会初始化另一个转换。

如果使能单个转换，拦截机制可能导致丢弃几个转换并且有额外的功耗。为了避免这种情况，在初始化一个单个转换后，数据寄存器直到转换完成才能读。

14.4.4.3 终止转换

当出现下列情况时，进行中的任何转换都将中止：

- ADC1SC1 写入(当前转换被中止，如果 ADCH 不都是 1，则会初始一个新转换。)
- ADC1SC2、ADC1CFG、ADC1CVH 或 ADC1CVL 写入。这表明出现运行模式更改，当前转换因此无效。
- MCU 复位。
- MCU 进入停止模式，ADACK 被禁止。

当一个转换终止，数据寄存器(ADCRH 和 ADCRL)的内容不会改变，而是上次转换后完成后的传送的值。在因复位导致的转换终止情况中，ADCRH 和 ADCRL 返回到它们的复位值。

14.5.4.4 电源控制

直到初始化一个转换，ADC 模块都保持空闲。如果 ADACK 被选作转换时钟源，ADACK 时钟产生器也被使能。

通过设置 ADLPC 可以减小功耗。这导致更小的 f_{ADCK} 最大值(参考电气描述)。

14.5.4.5 总转换时间

总转换时间依赖于采样时间(由 ADLSMP 决定)、MCU 总线频率、转换模式(8 位或 10 位)和转换时钟的频率(f_{ADCK})。模块有效后,输入的采样开始。ADLSMP 用于选择长还是短采样时间。当转换完成,转换器和输入通道隔离,用逐次渐进算法将模拟信号转换成数字信号。转换算法完成后,转换结果传送到 ADC1RH 和 ADC1RL。

如果总线频率小于 f_{ADCK} 频率,在使能短采样时间(ADLSMP=0)时,不能保证连续转换模式下采样时间的准确性。如果总线频率小于 f_{ADCK} 频率的 1/11,在使能长采样时间(ADLSMP=1)时,不能保证连续转换模式下采样时间的准确性。

表 14-12 概括介绍了不同条件下的最长总转换时间。

表 14-12. 总转换时间与控制条件之比较

转换类型	ADICLK	ADLSMP	最长总转换时间
单个或第一个连续转换 8 位	0x, 10	0	20ADCK 周期+5 总线时钟周期
单个或第一个连续转换 10 位	0x, 10	0	23ADCK 周期+5 总线时钟周期
单个或第一个连续转换 8 位	0x, 10	1	40ADCK 周期+5 总线时钟周期
单个或第一个连续转换 10 位	0x, 10	1	43ADCK 周期+5 总线时钟周期
单个或第一个连续转换 8 位	11	0	5 μ s+20ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	0	5 μ s+23ADCK +5 总线时钟周期
单个或第一个连续转换 8 位	11	1	5 μ s+40ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	1	5 μ s+43ADCK +5 总线时钟周期
后来的连续转换 8 位 $f_{BUS} \geq f_{ADCK}$	xx	0	17ADCK 周期
后来的连续转换 10 位 $f_{BUS} \geq f_{ADCK}$	xx	0	20ADCK 周期
后来的连续转换 8 位 $f_{BUS} \geq f_{ADCK}/11$	xx	1	37ADCK 周期
后来的连续转换 10 位 $f_{BUS} \geq f_{ADCK}/11$	xx	1	40ADCK 周期

最长的总转换时间由所择时钟源和分频率决定。时钟源由 ADICLK 位决定,分频率由 ADIV 制定。例如,在 10 位模式中,总线时钟选为输入时钟源、输入时钟分频率选为除以 1、总线频率为 8MHz,那么单转换的转换时间为:

$$\text{转换时间} = \frac{23\text{ADCK 周期}}{8\text{MHz}/1} + \frac{5\text{个总线周期}}{8\text{MHz}} = 3.5\mu\text{s}$$

$$\text{总线周期数} = 3.5\mu\text{s} \times 8\text{MHz} = 28\text{周期}$$

注：ADCK 频率必须介于 f_{ADCK} 最小值和最大值之间才符合 ADC 技术规范。

14.5.5 自动比较功能

比较功能可以设置为检测上限或下限。采样和转换输入后，结果和比较值 (ADC1CVH 和 ADC1CVL) 的补数相加。比较上限时 (ACFGT=1)，如果结果大于或等于比较值，COCO 置位。比较下限时 (ACFGT=0)，如果结果小于比较值，COCO 置位。转换结果和比较值的补数相加后产生的值传送到 ADC1RH 和 ADC1RL。

使能比较功能时，当一次转换完成，如果比较条件非真，COCO 不置位并且没有数据传输到结果寄存器。如果使能 ADC 中断 (AIEN=1) 当 COCO 置位时，产生一次 ADC 中断。

注：这个比较功能可以用来监控通道的电压，这时 MCU 可能处于等待模式或 STOP3 模式。在满足比较条件时，ADC 中断会唤醒 MCU。

14.5.6 MCU 等待模式运行

WAIT 指令使 MCU 处于更低功耗的待机模式，待机模式的恢复非常快，因为时钟源保持有效状态。如果正在进行转换时 MCU 进入等待模式，那么转换会继续，直到完成。在 MCU 处于等待模式时，通过硬件触发或使能连续转换，可以初始转换。

处于等待模式时，总线时钟、总线时钟的二分频和 ADACK 可以作为转换时钟源。在等待模式时，ALTCLK 作为转换时钟源是由该 MCU 的 ALTCLK 的定义决定的。参考该 MCU 中模块说明中关于 ALTCLK 说明的信息。

如果 ADC 中断使能 (AIEN=1)，转换完成事件就会设置 COCO，生成 ADC 中断，把 MCU 从等待模式中唤醒。

14.5.7 MCU Stop3 模式运行

STOP 指令使 MC 进入低功耗待机模式，在该模式中，MCU 上的大多数甚至所有时钟源都被禁止。

14.5.7.1 ADACK 禁止的 Stop3 模式

如果异步时钟 ADACK 未被选为转换时钟, 执行 STOP 指令将中止当前转换, 并把 ADC 置入空闲状态。ADC1RH 和 ADC1RL 内容不受 stop3 模式的影响。从 stop3 模式退出后, 需要软件或硬件触发来重新开始转换。

14.5.7.2 SDACK 使能的 Stop3 模式

如果异步时钟 ADACK 已被选为转换时钟, ADC 会在 stop3 模式中继续运行。为了保证 ADC 操作, MCU 的电压调节器在 stop3 模式中必须保持工作状态。参考该 MCU 模块介绍中的配置信息。

若 MCU 进入停止模式时, 有一个转换正在处理, 则等待转换结束。若 MCU 已处于停止模式, 通过硬件触发的方式或者使能连续转换模式, 可以重新开始转换。

如果 ADC 中断使能(AIEN=1), 转换完成事件就会设置 COCO, 生成 ADC 中断, 把 MCU 从 stop3 模式中唤醒。

注: ADC 可能将系统从低功率停止中唤醒, 导致 MCU 开始强烈的运行电平电流而没有产生一个系统级的中断。为了避免这种情况, 当进入 stop3 模式并继续 ADC 转换时, 软件应该确保数据传输拦截机制(在 14.5.4.2 节“完成转换”中作过介绍)已经清除。

14.5.8 MCU Stop1 和 Stop2 模式运行

当 MCU 进入 stop1 或 stop2 模式时, ADC 模块会被自动禁止。所有模块寄存器在退出 stop1 或 stop2 模式后都包含它们的复位值。因此, 在从 stop1 或 stop2 退出后, 模块必须重新使能和重新配置。

14.6 初始化信息

本节给出了一个为用户提供如何初始化和配置 ADC 模块的一些基本指导示例。用户可以从众多选项中灵活选择配置模块的 8 位或 10 位分辨率、单或连续转换、查询或中断方式。在该例中的设置信息可以参见表 14-6, 表 14-7 和表 14-8。

注: 十六进制的前缀是 0x, 二进制的前缀是%, 十进制没有前缀。

14.6.1 ADC 模块初始化举例

14.6.1.1 初始化顺序

在 ADC 模块完成转换之前，必须对其进行初始化操作。一个典型的初始化顺序是：

1. 更新配置寄存器(ADCCFG)选择输入时钟源和产生内部时钟(ADCK)的分频因子。这个寄存器也可以用来选择采样时间和低功耗配置。

2. 更新状态和控制寄存器 2(ADCSC2)选择转换触发器(硬件或软件)和比较功能选项，如使能的话。

3. 更新状态和控制寄存器 1(ADCSC1)可以选择转换是否是连续的还是仅一次完成，并且使能或关闭转换完成中断。同时还选择执行转换的输入通道。

14.6.1.2 伪代码举例

在该例中，ADC 模块将设置为：允许中断产生一个单个 10 位转换，低功耗，输入通道为 1，长采样时间，内部的 ADCK 时钟将由总线时钟 1 分频得到。

ADCCFG=0x98(%10011000)

Bit 7	ADLPC	1	配置为低功耗(降低最大的时钟速率)
Bit 6:5	ADIV	00	设置 ADCK 为输入时钟/1
Bit 4	ADLSMP	1	设置为长采样时间
Bit 3:2	MODE	10	设置位 10 位转换模式
Bit 1:0	ADICLK	00	选择总线时钟作为输入时钟

ADCSC2=0x00(%00000000)

Bit 7	ADACT	0	标志表示转换是否在处理中
Bit 6	ADTRG	0	选择软件触发器
Bit 5	ACFE	0	关闭比较功能
Bit 4	ACFGT	0	在该例中未使用
Bit 3:2		00	未实现或保留，读为 0
Bit 1:0		00	保留为 Freescale 使用；写为 0

ADCSC1=0x41(%01000001)

Bit 7	COCO	0	只读标志当转换完成时置位
Bit 6	AIEN	1	转换完成中断使能
Bit 5	ADCO	0	仅一次转换(关闭连续转换)
Bit 4:0	ADCH	00001	选择通道 1 作为 ADC 输入通道

ADCRH/L=0xxx

保存转换结果。在低字节前读高字节，所以转换数据不会被下一次转换的数据重写。

ADCCVH/L=0xxx

当允许比较功能时，保存比较值。

APCTL1=0x02

关闭 AD1 引脚 I/O 控制。其他 AD 引脚仍然位通用 I/O 引脚。

APCTL2=0x00

其他 AD 引脚仍然时通用 I/O 引脚。

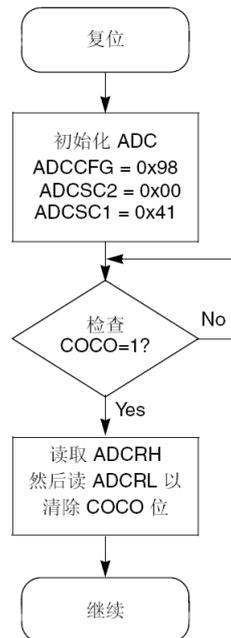


图 14-14. 初始化流程图示例

14.7 应用信息

该节包含了在应用中使用 ADC 模块的信息。ADC 被设计集成了一个微控制器可以使用在需要 A/D 转换器的嵌入式控制应用中。

14.7.1 外部引脚和布线

以下几节讨论与 ADC 模块相关的外部引脚以及为获得最佳结果，应该如何使用它们。

14.7.1.1 模拟电源引脚

ADC 模块有模拟电源和模拟地(V_{DDAD} 和 V_{SSAD})，在一些设备中有独立的引脚。在其他设备， V_{SSAD} 和 MCU 的 V_{SS} 复用相同的引脚。在一些设备中， V_{SSAD} 和 V_{DDAD} 共享数字信号供电引脚。有分离的部分来为模拟信号供电，同时和相应的数字信号电源绑定在相同的引脚。所以这两个电源在一定程度上保持分离。

当作为独立的引脚时， V_{DDAD} 和 V_{SSAD} 必须和相应的 MCU 数字电源(V_{DD} 和 V_{SS})连接相同的电压，谨慎布线防避免最大干扰度，旁路电容离封装要尽可能近。

在模拟和数字电源单独供电时的情况，这两个电源的接地连接必须是 V_{SSAD} 引脚。如果可能这应该是这两个电源的唯一接地连接。 V_{SSAD} 接单独的地位置。

14.7.1.2 模拟参考引脚

除了模拟电源，ADC 模块连接了两个参考电压输入。高参考电压是 V_{REFH} ，在有些设备中，可能和 V_{DDAD} 复用相同的引脚。地参考电压是 V_{REFL} ，在有些设备中和 V_{SSAD} 复用相同的引脚。

当作为单独引脚时， V_{REFH} 可能和 V_{DDAD} 连到相同的电压。或可能是由外部得到(介于 V_{DDAD} 最小值和 V_{DDAD} 之间， V_{REFH} 绝对不能大于 V_{DDAD})。当作为单独引脚时， V_{REFH} 可能和 V_{DDAD} 连到相同的电压。 V_{REFH} 和 V_{REFL} 必须谨慎布线防止最大干扰度和旁路电容离封装尽可能近。

在每次逐次渐进步骤，电流尖峰组成的交流(AC)电流通过 V_{REFH} 和 V_{REFL} 循环位电容阵列提供电荷。满足这个电流要求的最好的外部元件是 $0.1\mu\text{F}$ 的电容(高频特性)。这个电容连接到 V_{REFH} 和 V_{REFL} 之间，离封装尽可能近。不推荐使用电阻，因为电流导致电压泄露，这可能导致转换错误。该路径上的电磁感应应最小化(仅伴随而生的)。

14.7.1.3 模拟输入引脚

外部模拟输入引脚通常和 MCU 设备的开关量 I/O 引脚复用。通过置位引脚控制寄存器的相应位可以关闭引脚 I/O 功能。相应的引脚控制寄存器位没有置位，转换可以操作。当引脚作为模拟输入时，推荐置位引脚控制寄存器位。这避免了连接问题，因为输入缓冲区处于高阻抗状态并且关闭上拉电阻。而且，当输入既

不是 V_{DD} 也不是 V_{SS} ，输入缓冲吸收直流(DC)电流。当置位引脚控制位，使所有的引脚都作为模拟输入时，必须要保证达到最低工作电流。

试验数据表明，当存在噪音或源阻抗高时，模拟输入上的电容可以提高性能。使用 $0.01\mu\text{F}$ 的电容(有高频特性)完全可以满足。这些电容并不是在所有的情况下都需要，但是它们必须放在离封装尽可能近的地方，作为 V_{SSA} 的参考。

为正确转换，输入的模拟电压必须在 V_{REFH} 和 V_{REFL} 之间。如果输入等于或大于 V_{REFH} ，转换电路会把模拟信号转换成 $3FF$ (10 位转换时的最大值)或 FF (8 位转换时的最大值)，如果输入等于或小于 V_{REFH} ，转换电路把模拟信号转换成 000 ，在 V_{REFH} 和 V_{REFL} 之间的输入电压是线性转换。当采样电容正在充电时，将会有有一个和 V_{REFL} 相关的短暂电流。当 $ADLSMP$ 低，将会采样以 $ADCK$ 为源的 3.5 个周期，当 $ADLSMP$ 高，则是 23.5 个周期。

为了减少因电流进入而引起的正确度减小，连接到模拟输入的引脚在转换期间不应该传输。

14.7.2 错误源

A/D 转换中存在几种错误源。将在本节的后面来讨论它们。

14.7.2.1 采样错误

为正确转换，输入必须被采样足够长时间才能达到合适的精度。如果最大输入电阻 $7\text{k}\Omega$ 和输入电容 5.5pF ，外部模拟源(RAS)的电阻小于 $5\text{k}\Omega$ ，则采样 $1/4\text{LSB}$ (10 位精度)可以在最小的采样窗口完成(8MHz 的最大 $ADCK$ 频率时，3.5 个周期)。

更高电阻或更高精度的采样可以通过置位 $ADLSM$ (增加采样窗口到 23.5 个周期)或通过减少 $ADCK$ 频率来增加采样时间。

14.7.2.2 引脚泄露误差

如果外部模拟源(RAS)为高，I/O 引脚上的泄露导致转换错误。如果在应用中不能容忍这个错误，为了少于 $1/4\text{LSB}$ 泄露错误(8 位模式 $N=8$ ，10 位模式 $N=10$)，应保持 RAS 小于 $V_{DDAD}/(2^N \times I_{LEAK})$ 。

14.7.2.3 减少噪音错误

在采样或转换过程中产生的系统噪音会影响转换的正确性。只有满足下面指定的条件才能保证 ADC 采样的正确性。

- V_{REFH} 和 V_{REFL} 之间有一个 $0.1\mu\text{F}$ 的低等效串联电阻的电容。
- V_{DDAD} 和 V_{SSAD} 之间有一个 $0.1\mu\text{F}$ 的低等效串联电阻的电容。
- 如果电源使用感应隔离, V_{DDAD} 和 V_{SSAD} 之间需要放一个 $1\mu\text{F}$ 的电容。
- V_{SSAD} (和 V_{REFL} , 如果连接)连接到 V_{SS} (连着地平面的点)。
- 初始化(硬件触发转换)前或刚初始化(软件或硬件触发转换)ADC 转换, 使 MCU 处于等待或停止模式下,
 - 对于软件触发转换, 用等待或停止指令写 ADC1SC1 后。
 - 对于停止模式操作, 选择 ADACK 作为时钟源。在停止模式的操作减少 V_{DD} 噪音但是因为停止复原增加了有效转换时间。

- MCU 处于转换时, 没有 I/O 转换开关, 输入或输出。

在一些情况, 外部系统行为导致辐射或噪音发射或伴随 ADC 的过多 V_{DD} 噪音。在这些情况, 或当 MCU 不能在等待状态或 I/O 行为停止, 这些推荐的行为可能减少影响正确性的噪音:

- 在选择输入通道和 V_{REFH} 或 V_{REFL} 之间放置一个 $0.01\mu\text{F}$ 的电容 (C_{AS})(这将改善噪音问题但是影响基于外部的模拟源电阻的采样率)。
- 求多次连续转换模拟值的平均。每四次采样都需要消除 1LSB 的影响, 一个时钟的错误。
- 通过关闭异步时钟(ADACK)和求平均, 减少同步噪音的影响。和 ADCK 同步的噪音无法达到平均数。

14.7.2.4 编码宽度和量化误差

ADC 在理想化的情况下可以将输入的线性值量化成 1024 个块(在 10 位模式)。每块有相同的高度(1 个代码)和宽度。宽度定义为 d_{leta} , 在一个代码到下一个之间。N 位转换器(N 是 8 或 10)的理想代码宽度, 定义为 1LSB, 为:

$$1\text{LSB} = \frac{(V_{REFH} - V_{REFL})}{2^N} \quad \text{公式. 14-2}$$

数字化结果存在固有的量化误差。对于 8 位或 10 位转换，在从当前的转换函数到直线转换函数这两点的中间点时，代码转换将发生。因此在 8 位或 10 位模式中，量化误差为 $\pm 1/2\text{LSB}$ 。因此，第一个转化的编码宽度(\$000)只有 $1/2\text{LSB}$ 并且最后一个转化的宽度(\$FF 或\$3FF)是 1.5LSB 。

14.7.2.5 线性误差

ADC 可能存在几种非线性的情况。尽管已经用了各种方法减少这些错误，但是应用程序应该要知道错误的存在，因为它们影响全局的精度。这些错误是：

- 归零错误(EZS)(也称作偏移量)—这个错误定义为第一个转换的实际编码宽度和理想编码宽度的不同($1/2\text{LSB}$)。注意如果第一个转换是\$001，那么实际的\$001 编码宽度和理想情况(1LSB)的不同被使用了。

- 满标错误(EFS)—这个错误定义为最后一个转换的实际编码宽度和理想编码宽度的不同(1.5LSB)。注意如果最后一个转换是\$3FE，那么实际的\$001 编码宽度和理想情况(1LSB)的不同被使用了。

- 微分非线性(DNL)—这个错误被定义为所有转换中实际编码宽度和理想编码宽度最大误差。

- 积分非线性(INL)—这个错误定义为 DNL 总和所能达到的最大值。更简单的，所有编码中，编码所给的实际的转换电压和和相应的理想电压的最大误差。

- 总非校准误差(TUE)—这个错误定义为实际转换函数和理想线性转换函数的不同，因此包含所有形式的错误。

14.7.2.6 编码抖动，非单调性和丢码

ADC 易受三种特殊形式的错误影响。它们是编码抖动，非单调性，丢码。

编码抖动—在某一点时，当重复采样时，一个确定的输入电压转化成两个值中的一个。理想的，当输入电压比转换电压只小很少时，转换器产生更低的编码(反之亦然)。然而，即使很小的系统噪音也会导致转换器对于转换电压周围一定范围的输入电压不确定(在两个编码之间)。这个范围通常是 $1/2\text{LSB}$ ，并且随噪音而增加。这个错误可以通过重复采样和对结果求均值来减少。另外在 14.7.2.3 节噪音错误中的技巧可以减少该错误。

非单调性—可以定义为，除了代码抖动，转换器可能转换一个较高电压位较

低编码。丢码是那些对于任何输入都不会转换的值。

丢码一输入任何数值都不能改变其原有的数值。

在 8 位或 10 位模式中，ADC 将确保单调并且不遗失编码。

第十五章 开发支持

15.1 简介

HCS08 中的开发支持系统包括后台调试控制(BDC)和片上调试模块(DBG)。BD 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，允许以非介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，外部引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后台调式接口向目标 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。

MC9S08AW60 系列的交替 BDC 时钟源是 ICGLCLK。更多 ICGLCLK 信息和如何选用时钟源，请参见第八章，“内部时钟发生器(S08ICGV4)”。

15.1.1 特性

BDC 模块的特性包括：

- 单引脚进行模式选择和后台调试通信
- BDC 的寄存器不位于存储器地址中
- SYNC 命令确定目标通信速率
- 非介入式命令进行存储器存取
- 供 CPU 寄存器存取的激活后台调试模式命令
- GO 和 TRACE1 命令
- 后台调试命令可以将 CPU 从停止模式或等待模式中唤醒
- BDC 内置一个硬件地址断点

-
- 如果 BDC 使能，则振荡器运行在停止模式
 - 处于激活后台调试模式时，COP 看门狗禁止

ICE 系统的特性包括：

- 两个触发比较器：两个地址+读/写(R/W)或一个完整地址+数据+R/W
- 灵活的 8 字 16 位 FIFO(先进先出)缓存，用于捕获信息：
 - 流程变化的地址或
 - 纯事件数据
- 两个类型的断点：
 - 指令操作码的标记断点
 - 任何地址存取的强制断点
- 九个触发模式：
 - 基本：只有 A，A 或 B
 - 顺序：A 然后 B
 - 全部：A 和 B 数据，A 和非 B 数据
 - 事件(存储数据)：纯事件 B，A 然后纯事件 B
 - 范围：在范围以内($A \leq \text{地址} \leq B$)，在范围以外(地址 < A 或地址 > B)

15.2 后台调试控制器(BDC)

HCS08 系列中的所有 MCU 都包含一个单线后台调试接口，它支持片上非易失性存储器的在线编程和先进的非介入式调试功能。与早期 8 位 MCU 的调试接口不同，这个系统不干扰正常的应用资源。它不使用任何用户存储器或存储器映射中的地址，也不分享任何片上外设。

BDC 命令分为两大组：

- 激活后台调试模式命令要求目标 MCU 处于激活后台调试模式(用户程序未运行)。激活后台调试模式命令允许读写 CPU 寄存器，允许用户一次跟踪一个用户指令，或从激活后台调试模式进入用户程序。

- 非介入式命令可以随时执行，即使用户的程序正在运行。非介入式命令允许用户在后台调试控制器中读写 MCU 存储器位置或存取状态和控制寄存器。

一般地，可以用相当简单的接口盒将来自主机的命令转换为与单线后台调试系统连接所需的串行命令。根据开发工具供应商的不同，这个接口盒可以采用标准 RS-232 串行端口，或是并行打印端口，或是其它类型的通信端口，如用来与

PC 通信的 USB 接口。这个接口盒一般通过接地、BKGD 引脚、RESET，有时还有 V_{DD} 与目标系统连接。RESET 引脚的开漏连接允许主机强制目标系统复位，这有助于重新获得对已失目标系统的控制，或在片上非易失性存储器重新编程之前，控制目标系统的启动。有时可以用 V_{DD} 来允许接口盒使用目标系统的电源，避免再使用一个电源。但是，如果单独对接口盒供电，它可以连接到一个正在运行的目标系统，而不必强制目标系统复位，否则会干扰正在运行的应用程序。

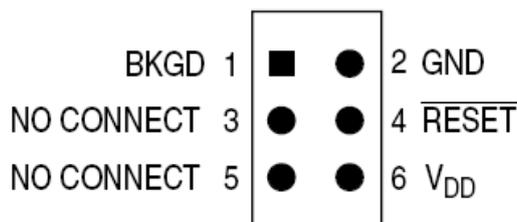


图 15-1. BDM 工具接口

15.2.1 BKGD 引脚描述

BKGD 是单线后台调试接口引脚。这个引脚的主要功能是实现激活后台调试模式命令和数据的双向串行通信。在复位过程中，这个引脚用来选择激活后台调试模式启动或启用用户的应用程序。这个引脚还用来请求定时同步响应脉冲，允许主机开发工具确定后台调试串行通信的正确时钟频率。

BDC 串行通信采用首先引入在微处理器 M68HC12 系列上的定制串行协议。这个协议假定主机知道通信时钟速率，这个速率按目标 BDC 所有通信通过主机启动和控制所有通信，主机驱动高到低边沿发出每个位时间开始信号。命令和数据以最重要的位先发(MSB 先发)的方式发送。有关通信协议的详细信息，请参见 15.2.2，“通信详细介绍”。

如果主机尝试与 BDC 时钟速率未知的目标 MCU 沟通，可以发送 SYNC 命令给目标 MCU，请求定时同步响应信号，通过这个信号，主机可以判断正确的通信速率。

BKGD 是伪开漏引脚，有一个片上上拉，因此不需要外部上拉电阻。与典型的开漏引脚不同，引脚上的外部 RC 时间常数(受外部容性的影响)，在信号上升时间上几乎不起作用。定制协议提供瞬态加速脉冲，强制提高这个引脚的上升时间，而没有驱动电平冲突风险。参见 15.2.2，“通信详细介绍”，了解更多详情。

当没有调试盒连接 6 引脚的 BDM 接口连接器时，BKGD 的内部上拉会选择

正常的操作模式。当调试盒连接到 BKGD 时，可以在 MCU 复位后强制它进入激活后台调试模式。强制激活后台调试的具体条件取决于 HCS08 衍生产品(参见“开发支持”小节的介绍)。不必复位目标 MCU 来通过后台调试接口来与之通信。

15.2.2 通信详细介绍

BDC 串行接口需要外部控制器来生成 BKGD 引脚上的下降沿，指示每个位时间的开始。无论数据是发送或接收，外部控制器都会提供这个下降边沿。

BKGD 是伪开漏引脚，可以被外部控制器或 MCU 来驱动。数据以 MSB 先发的形式且以每位 16 个 BDC 时钟周期的速率(标定速率)发送。如果来自主机的下降边沿之间产生 512 个 BDC 时钟周期，则该接口超时。如果出现超时，任何正在进行的 BDC 命令被中止，对目标 MCU 系统的存储器或操作模式没有影响。

定制串行协议要求调试盒知道目标 BDC 通信时钟速率。

BDC 状态和控制寄存器中的时钟开关(CLKSW)控制位允许用户选择 BDC 时钟源。BDC 时钟源可以是总线，或备用的 BDC 时钟源。

BKGD 引脚可以接收高或低电平，或发送高或低电平。下图显示了每种情况的时序。接口时序与目标 BDC 中的时钟同步，但是与外部主机异步。显示的内部 BDC 时钟信号是计数周期的参考。

图 15-2 显示了外部主机将逻辑 1 或 0 发送到目标 HCS08MCU 的 BKGD 引脚。主机与目标异步，因此主机生成的 BKGD 下降边沿与目标所认为的位时间起始点有 0 到 1 周期的延迟。10 个目标 BDC 时钟周期后，目标获得 BKGD 引脚的电平。一般地，主机在主机到目标方向的传输过程中驱动 BKGD 引脚，以加快上升边沿。由于目标在主机至目标方向的传输周期中不驱动 BKGD 引脚，因此没有必要在此期间将线路作为开漏信号。

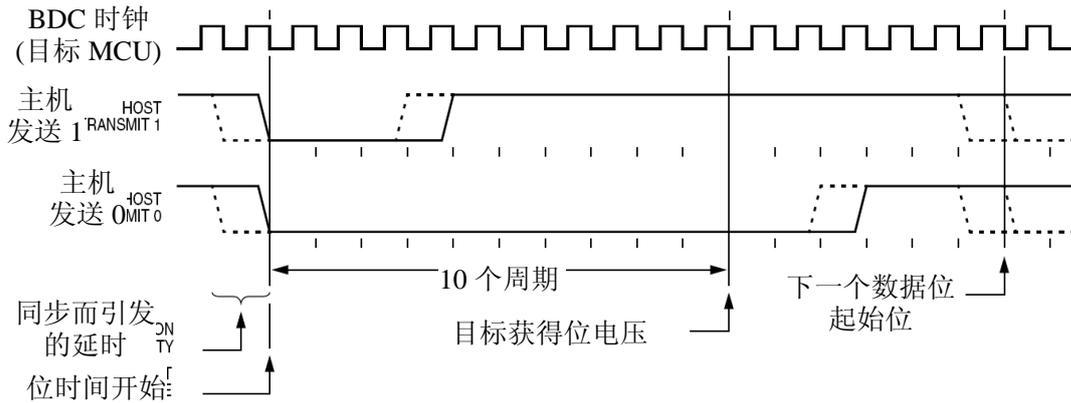


图 15-2. BDC 主机到目标机串行位时序

图 15-3 显示主机从目标 HCS08MCU 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 引脚足够长的时间，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。

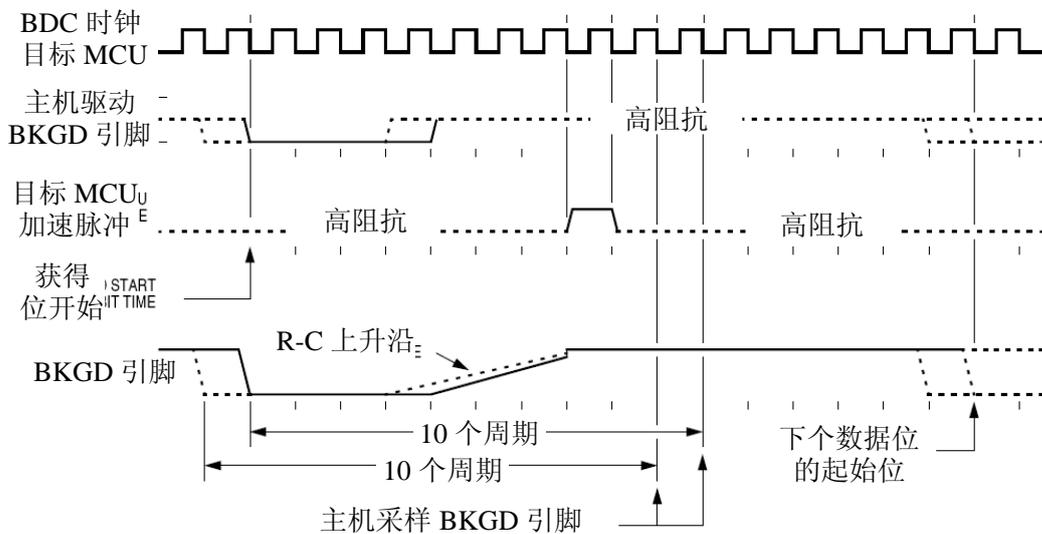


图 15-3. BDC 目标机到主机串行位时序(逻辑 1)

图 17-4 显示了主机从目标 HCS08MCU 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 HCS08MCU 完成它。由于目标希望主机接收逻辑 0，它保持低 BKGD 引脚 13 个 BKGD 引脚周期，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。

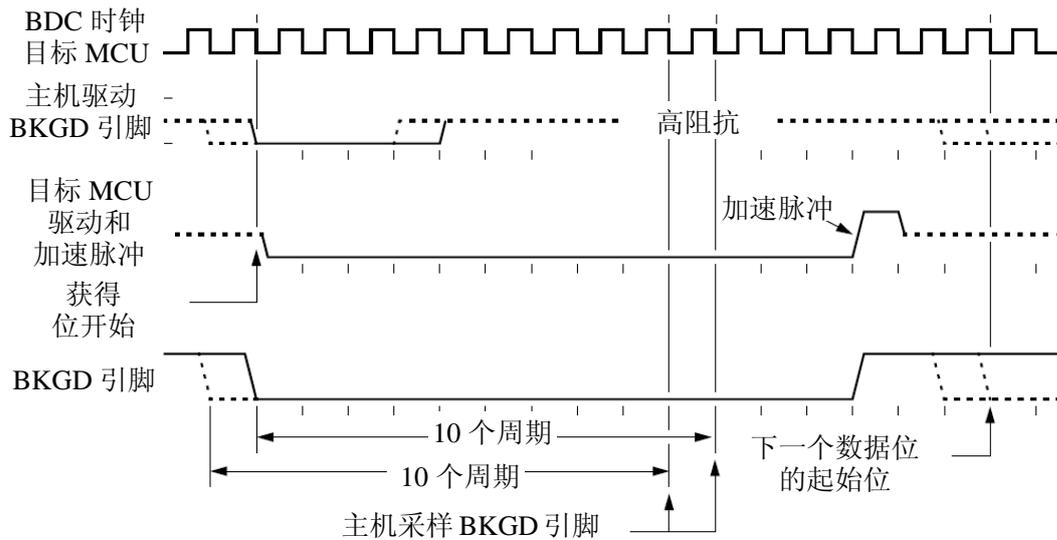


图 15-3. BDC 目标机到主机串行位时序(逻辑 0)

15.2.3 BDC 命令

BDC 指令由主机串行发向目标 MCU 的 BKGD 引脚。所有的命令和数据都通过私有的 BDC 通信协议先发送最高位。激活后台模式命令需要目标 MCU 当前处在激活后台模式，而非介入式命令可在任何时候发出，无论目标 MCU 是处在激活后台模式还是在运行用户应用程序。

表 15-1 显示了所有 HCS08BDC 命令，并简要描述了它们的编码结构，以及每个命令的含义。

代码别名术语

这些术语在表 15-1 中使用来描述 BDC 命令的代码结构。

指令开始于主机到目标机方向(先发送最高位)的 8 位指令代码

/	==	命令分隔符
d	==	延时 16 个目标机 BDC 时钟周期
AAAA	==	16 位地址，主机向目标机传送
RD	==	8 位读取的数据，目标机向主机发送
WD	==	8 位写入的数据，主机向目标机发送
RD16	==	16 位读取数据，主机向目标机发送
WD16	==	16 位写入数据，主机向目标机发送
SS	==	BDCSCR 的内容，目标机向主机发送时(状态)
CC	==	对 BDCSCR 写入的 8 位数据，主机向目标机发送(控制)
RBKP	==	16 位读取数据，目标机向主机传送(来自 BDCBKPT 断点寄存器)
WBKP	==	16 位写入的数据，主机向目标机传送(写入 BDCBKPT 断点寄存器)

表 15-1. BDC 命令概要

命名	激活 DBM /非介入式	指令结构	描述
SYNC	非介入式	n/a ¹	请求定时参考脉冲以决定目标 BDC 通信速率。
ACK_ENABLE	非介入式	D5/d	使能响应协议。参见飞思卡尔文档编号 HCS08RMv1/D。
ACK_DISABLE	非介入式	D6/d	禁止响应协议。参见飞思卡尔文档编号 HCS08RMv1/D。
BACKGROUND	非介入式	90/d	如果允许，则进入后台模式(如果 ENBDM=0，则忽略)
READ_STATUS	非介入式	E4/SS	从 BDCSCR 读取 BDC 状态
WRITE_CONTROL	非介入式	C4/CC	写 BDC 控制位到 BDCSCR 中
READ_BYTE	非介入式	E0/AAAA/d/RD	从目标内存中读一个字节
READ_BYTE_WS	非介入式	E1/AAAA/d/SS/RD	读一个字节并报告状态
READ_LAST	非介入式	E8/SS/RD	从地址重新读字节，仅读和报告状态
WRITE_BYTE	非介入式	C0/AAAA/WD/d	写一个字节到目标内存
WRITE_BYTE_WS	非介入式	C1/AAAA/WD/d/SS	写一个字节并报告状态
READ_BKPT	非介入式	E2/RBKP	读 BDCBKPT 断点寄存器
WRITE_BKPT	非介入式	C2/WBKP	写 BDCBKPT 断点寄存器
GO	激活 BDM	08/d	执行开始于当前 PC 中地址处的用户应用程序
TRACE1	激活 BDM	10/d	跟踪 PC 中地址处的 1 条用户指令，然后回到激活后台模式
TAGGO	激活 BDM	18/d	与 GO 相同，但激活外部标签(HCS08 器件没有外部标签引脚)
READ_A	激活 BDM	68/d/RD	读累加器(A)
READ_CCR	激活 BDM	69/d/RD	读条件代码寄存器(CCR)
READ_PC	激活 BDM	6B/d/RD16	读程序计数器(PC)
READ_HX	激活 BDM	6C/d/RD16	读 H 和 X 寄存器(H:X)
READ_SP	激活 BDM	6F/d/RD16	读堆栈指针(SP)
READ_NEXT	激活 BDM	70/d/RD	以 1 为基数递增 H:X，然后读位于 H:X 的存储器字节
READ_NEXT_WS	激活 BDM	71/d/SS/RD	以 1 为基数递增 H:X，然后读位于 H:X 的存储器字节。报告状态和数据
WRITE_A	激活 BDM	48/WD/d	写累加器(A)
WRITE_CCR	激活 BDM	49/WD/d	写条件代码寄存器(CCR)
WRITE_PC	激活 BDM	4B/WD16/d	写程序计数器(PC)
WRITE_HX	激活 BDM	4C/WD16/d	写 H 和 X 寄存器(H:X)
WRITE_SP	激活 BDM	4F/WD16/d	写堆栈指针(SP)
WRITE_NEXT	激活 BDM	50/WD/d	以 1 为基数递增 H:X，然后写位于 H:X 的存储器字节

WRITE_NEXT_WS	激活 BDM	51/WD/d/SS	以 1 为基数递增 H:X, 然后写位于 H:X 的存储器字节。报告状态和数据
---------------	--------	------------	---

¹.SYNC 指令是一种特殊的操作, 它没有指令代码。

SYNC 命令与其它 BDC 命令不同, 因为主机不必要知道 BDC 通信的正确通信速率, 直到它分析完 SYNC 命令的响应后。

要发出 SYNC 命令, 主机:

- 保持 BKGD 引脚为低电平至少 128 个周期, 而且是最慢的 BDC 时钟来计(最慢的时钟一般是参考振荡器 64 分频或自时钟速率 64 分频。)

- 驱动 BKGD 达到高电平, 实现瞬态加速, 快速上升时间(这个加速脉冲一般是系统中最快时钟的一个周期)。

- 去除 BKGD 引脚的所有驱动, 这样它可回复到高阻抗。

- 监视 BKGD 引脚得到同步响应脉冲

当检测到主机的 SYNC 请求(比在正常 BDC 通信过程中发生的慢时钟要长), 则目标:

- 等待 BKGD 返回到逻辑高电平

- 延迟 16 个周期, 允许主机停止驱动高电平加速脉冲

- 驱动 BKGD 低态 128 个 BDC 时钟周期

- 驱动一个周期的高电平加速脉冲, 在 BKGD 上实现快速上升时间

- 去除 BKGD 引脚的所有驱动, 这样它可回复到高阻抗。

主机测量这个 128 个周期的响应脉冲的低电平时间, 判断速率, 进行后续的 BCD 通信。主机一般可以确定正确的通信速率, 与实际目标速率的误差只有百分之几, 通信协议能够接受百分之几的速率误差。

15.2.4 BDC 硬件断点

BDC 包括一个相对简单的硬件断点, 将 CPU 地址总线与 BDCBKPT 寄存器中的 16 位匹配值进行比较。这个断点可以生成强制断点或标记断点。强制断点使 CPU 在存取断点地址后的第一个指令边界进入激活后台调试模式。标记的断点使指令操作码在断点地址被标记, 这样当 CPU 到达指令队列的终点时, 将进入激活后台模式, 而不是执行该指令。这意味着标记的断点可能放置在指令操作代码的地址上, 而强制断点可以设置在任何地址。

BDC 状态和控制寄存器(BDCSCR)中的断点使能(BKPTEN)控制位用来激活

断点逻辑(BKPTEN=1)。当 BKPTEN=0(复位后它的默认值), 断点逻辑禁止, 无论其它 BDC 断点中的值是多少, 也不管控制位如何, 均不请求断点。BDCSCR 中的强制/标记选择(FTS)控制位用来选择强制(FTS=1)或标记(FTS=0)类型断点。

片上调试模块(DBG)包括两个额外的硬件断点的电路, 这两个硬件断点比 BDC 模块中的简单断点更灵活。

15.3 片上调试系统(DBG)

由于 HCS08 器件没有外部地址和数据总线, 在线仿真器最重要的功能已经构建在 MCU 的芯片上。这种调试系统包含可以灵活地存储地址或数据总线信息的 8 级 FIFO, 和一个确定何时捕获总线信息以及捕获哪些总线信息的灵活触发系统。这个系统依赖单线后台调试系统来存取调试控制寄存器, 读取 8 级 FIFO 的结果。

调试模块包括控制和状态寄存器, 可以在用户存储器映射中存取。这些寄存器位于高地址寄存器空间中, 避免使用宝贵的直接页面存储器空间。

大多数调试模块的功能在开发过程使用, 用户程序很少存取调试模块的任何控制和状态寄存器。一个例外就是调试系统可以提供一种手段来实施某种形式的 ROM 补丁。15.3.6, “硬件断点” 中对此有更详细的描述。

15.3.1 比较器 A 和 B

两个 16 位比较器(A 和 B)可以选择用 R/W 信号或一个操作码跟踪电路来鉴定。单独的控制位允许忽略每个比较器的 R/W 位。操作码跟踪电路可选地允许被规定, 如果操作码在规定的地址实际执行, 而不是只从存储器读到指令队列中, 则触发将发生。比较器还能够进行庞大的比较, 支持范围内和范围外触发模式。在所有 BDC 存取过程中, 比较器被临时禁止。

比较器 A 总是与 16 位 CPU 地址相关联。比较器 B 根据所选的触发模式比较 CPU 地址或 8 位 CPU 数据总线。由于 CPU 数据总线分为只读数据和写数据总线, RWAEN 和 RWA 控制位有一个额外的目的, 在完整地址加上数据比较中, 它们被用来确定其中哪些总线用在比较器 B 数据总线比较中。如 RWAEN=1(激活), RWA0(写), 则使用 CPU 的写数据总线, 否则用 CPU 的只读数据。

当比较器检测到合格的匹配条件时, 当前选择触发模式确定调试器逻辑做什

么。匹配可以导致以下情况：

- 生成 CPU 断点
- 将数据总线值存储到 FIFO 中
- 开始将流变化地址存储到 FIFO 中(开始类型跟踪)
- 停止将流变化地址存储到 FIFO 中(结束类型跟踪)

15.3.2 总线捕获信息和 FIFO 操作

使用 FIFO 的通常方式是建立触发模式和其它控制选项，然后打开调试器。当 FIFO 填满后，或调试器停止将数据存储到 FIFO 后，你可以按信息存储的顺序从中读取信息。状态位指示数据所在的 FIFO 中的有效信息的字数。如果在满 (CNT=1:0:0:0) 之前将 ARM 写为 0，以人工停止跟踪，信息移动一个位置，主机必须执行 $((8 - \text{CNT}) - 1)$ FIFO 虚读操作，使信息进入到 FIFO 中的第一个重要入口。

在大多数触发模式中，存储在 FIFO 中的信息包含 16 位流变化地址。在这些情况中，先读 DBGFH 然后读 DBGFL，从 FIFO 中获得一个一致的信息字节。读 DBGFL (FIFO 数据端口的低阶字节) 会使 FIFO 移动，这样下一个信息字可以在 FIFO 数据端口提供。在仅事件触发模式 (参见 17.3.5，“触发模式”) 中，8 位数据信息存储在 FIFO 中。在这些情况中，FIFO (DBGFH) 的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。

15.3.3 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 间接地址寄存器的内容，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。

15.3.4 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，除非指令被 CPU 真正执行才操作。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后台调试模式，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 进入激活后台调试模式，而不是执行被标记的指令。当 DBGTR 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止一个比较事件被跟踪。

15.3.5 触发模式

触发模式控制在调试运行中的整体行为。DBGT 寄存器中的 4 位 TRG 字段选择九个触发模式中的一个。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)。

写 1 到寄存器中的 ARM 位可启动调试运行，它设置 DBGS 中的 ARMF 标记，并清除 AF 和 BF 标记及 CNT 位。当 FIFO 满时，开始跟踪调试运行结束。在所选触发事件发生时，结束跟踪运行结束。任何调试运行均可通过写 0 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 都会存储流变化地址。在纯事件触发模式中，FIFO 将数据存储存储在 FIFO 的低八位。

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，无需在比较中使用 R/W 位，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。

仅 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)没有使用，但是如果使用了，就会忽略比较器 B 的数据匹配，导致向 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 的低位部分，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)没有使用，但是如果被使用了，就会忽略比较器 B 数据匹配，导致向 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。

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

范围外(地址 < A 或地址 > 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。

15.3.6 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 15.3.5，“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，进入激活后台调试模式，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，然后进入激活后台调试模式。

如果后台调试模式没有被通过 BKGD 引脚的串行 WRITE_CONTROL 命令激活(ENBDM=1)，CPU 将执行 SWI 指令，而不是进入激活后台调试模式。

15.4 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，了解所有 DBG 寄存器的

绝对地址分配。本小节只按名字涉及了寄存器和控制位。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。

15.4.1 BDC 寄存器和控制位

BDC 包括两个寄存器：

- BDC 状态和控制寄存器(BDCSCR)是一个 8 位寄存器包括后台调试控制器的控制和状态位。

- BDC 断点匹配寄存器(BDCBKPT)保存 16 位断点匹配地址。

这些寄存器通过专用的串行 BDC 命令访问，没有位于目标 MCU 的存储器空间(所以它们没有地址，不能被用户程序访问)。

BDCSCR 的一些位有写限制；另外可以在任何时候读写。例如当 MCU 处于激活后台模式时，ENBDM 控制位可能不允许写。这避免了当 MCU 处于激活后台模式时，控制位禁止激活后台模式的模糊条件。同时状态位(BDMACT、WS、WSF)是只读状态标识位，不能通过 WRITE_CONTROL 串行 BDC 命令写。时钟开关(CLKSW)控制位随时都可读或写。

15.4.1.1 BDC 状态和控制寄存器(BDCSCR)

该寄存器可以通过串行 BDC 命令(READ_STATUS 和 WRITE_CONTROL)读或写。但是不能被用户程序访问，因为它不位于 MCU 的普通内存映射中。

	7	6	5	4	3	2	1	0
读	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
写								
正常复位	0	0	0	0	0	0	0	0
在激活 DBM 复位	1	1	0	0	1	0	0	0

 =未生效的或者预留的

图 15-5. BDC 寄存器和控制位(BDCSCR)

表 15-2. BDCSCR 寄存器域描述

域	描述
---	----

7 ENBDM	<p>使能 BDM(允许激活后台模式) — 通常, 在调试会话开始后或只要调试主机复位目标并且直到正常复位使它清零时仍为 1, 该位通过调试主机快速写 1。如果应用可进入停止模式, 如果调试需要这位需要置位。</p> <p>0 BDM 不能激活(允许非介入性命令)。 1 BDM 可以激活允许激活后台模式命令。</p>
6 BDMACT	<p>后台激活模式状态 — 该位为只读位。</p> <p>0 BDM 无效(用户程序正在运行)。 1 BDM 有效并且等待串行命令。</p>
5 BKPTEN	<p>BDC 断点使能 — 如果该位被清零, 则 BDC 断点不可用, 同时 FTS 控制位和 BDCBKPT 匹配寄存器将被忽略。</p> <p>0 禁止 BDC 断点。 1 允许 BDC 断点。</p>
4 FTS	<p>强制/标签选择位 — 当 FTS=1 时, 无论 CPU 地址总线是否与 BDCBKPT 匹配寄存器相匹配, 都会产生断点。当 FTS=0 时, CPU 地址总线和 BDCBKPT 寄存器之间的匹配将会给已获得的操作码作标记。若该被标记的操作码没有到达指令队列的末尾, 则 CPU 进入激活后台模式而不是执行已标记的操作码。</p> <p>0 在断点地址处给操作码做标记, 如果 CPU 尝试执行这条指令就进入激活后台模式。 1 在下一个指令边界, 断点匹配强制执行激活后台模式(地址无需是一个操作码)。</p>
3 CLKSW	<p>选择 BCD 通信时钟源 — CLKSW 默认 0, 选择其它 BDC 时钟源。</p> <p>0 其它 BDC 时钟源 1 MCU 总线时钟</p>
2 WS	<p>等待/停止状态位 — 当目标 MCU 处于等待/停止模式时, 大部分 BDC 指令将不被执行。然而, BACKGROUND 指令可用于使目标 CPU 跳出等待/停止模式并进入激活后台模式, 在此所有的 BDC 指令都能运行。无论何时主机迫使目标 MCU 进入激活后台模式, 主机在尝试其他 BDC 指令前都应当发出一个 READ_STATUS 指令来核实 BDMACT=1。</p> <p>0 目标 CPU 正执行用户程序代码或正处于激活后台模式(当后台模式被激活时, 不是等待/停止模式)。 1 目标 CPU 处于等待/停止模式, 或者使用 BACKGROUND 指令使它从等待/停止模式变为激活后台调试模式。</p>
1 WSF	<p>等待/停止失败状态位 — 如果内存访问指令时由于目标 CPU 在差不多同一时刻执行了一条等待或停止指令, 而导致失败时则置位该位。通常的恢复策略是发出一个 BACKGROUND 指令来跳出等待或停止模式而进入激活后台模式, 如果重复指令仍然失败, 则回到用户程序。(如果需要的话, 主机可以恢复 CPU 寄存器并用堆栈存储数值, 同时重新执行等待或停止指令。)</p> <p>0 内存访问没有与等待或停止指令发生冲突。 1 由于 CPU 进入等待或停止模式, 内存访问指令失败。</p>

0	数据有效失败状态 — 这个状态没有在 MC9S08AW60 系列中使用，因为它没有缓存存储器。
DVF	0 存储器存取与缓存存储器不冲突
	1 存储器存取命令失败，因为 CPU 没有完成缓存存储器的存取

15.4.1.2 BDC 断点匹配寄存器(BDCBKPT)

这个 16 位的寄存器用于存储 BDC14 位的地址的硬件断点。BDCSCR 中的 BKPTEN 和 FTS 控制位用于允许和配置断点逻辑。专门的串行 BDC 指令 (READ-BKPT 和 WRITE-BKPT) 用于读写 BDCBKPT 寄存器，但是用户程序不能存取它，因为它不在 MCU 的普通存储器映射空间中。当目标 MCU 在运行用户应用程序之前处于激活后台模式时通常设置断点。关于建立和使用 BDC 中的硬件断点逻辑的更多信息，请参见 15.2.4，“BDC 硬件断点”。

15.4.2 系统后台调试强制复位寄存器(SBDFR)

该寄存器仅包含一个只写控制位。必须要用一个串行后台模式命令，例如 WRITE_BYTE，来写 SBDFR。从用户程序试图写该寄存器将被忽略。读该寄存器总是返回 0x00。

	7	6	5	4	3	2	1	0
读	0	0	0	0	0	0	0	0
写								BDFR ¹
复位	0	0	0	0	0	0	0	0

 =未生效的或者预留的

图 15-6. 系统后台调试强制复位寄存器(SBDFR)

¹BDFR 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。

表 15-3. 寄存器域描述

域	描述
0 BDFR	后台调试强制复位 — 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。向该位写 1，强制 MCU 复位。该位不能从用户程序写。

15.4.3 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，分别是三个 16 位寄存器和三个 8 位控制和一个状态寄存器。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 修补机制。

15.4.3.1 后台比较器 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。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。

15.4.3.2 后台比较器 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。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。

15.4.3.3 后台比较器 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。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。

15.4.3.4 后台比较器 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。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。

15.4.3.5 调试 FIFO 高寄存器(DBGFHF)

这个寄存器提供对 FIFO 的高 8 位的只读接入。写该寄存器没有意义或无效。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 DBGFHF 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 DBGFHF，因为读 DBGFL 会导致 FIFO 优先于下个字的信息。

15.4.3.6 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。写该寄存器没有意义或无效。

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 DBGFL。

当 FIFO 仍然打开时(打开后，但 FIFO 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 DBGFL，会使最近获取的操作码的地址存储到 FIFO 中的最后位置。定期读取 DBGFL 之后读 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。

15.4.3.7 调试控制寄存器(DBGC)

该寄存器可在任何时候被读或写。

	7	6	5	4	3	2	1	0
读	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
写								
复位	0	0	0	0	0	0	0	0

图 15-7. 调试控制寄存器(DBGC)

表 15-4. DBGC 寄存器域描述

域	描述
7 DBGEN	调试模块启用——用来启用调试模块。如果 MCU 是安全的，DBGEN 不能设置为 1。 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 位)，完成调试运行就自动清除它。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 CPU 生成中断请求。对于结束跟踪, 如果比较器和 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——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

15.4.3.8 调试触发寄存器(DBGT)

这个寄存器在任何时候都可以读, 但是只有当 ARM=0 时才可以写, 除非位 4 和位 5 硬件连线至 0。

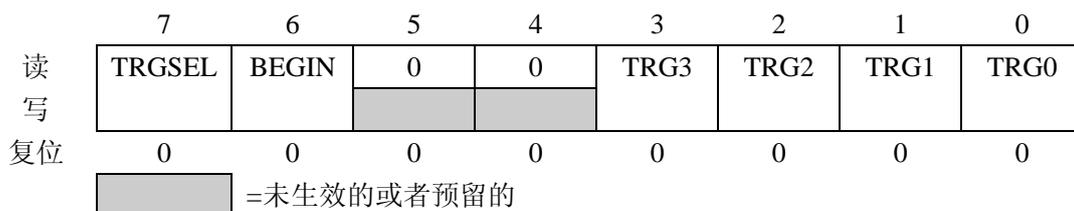


图 15-5. 调试触发寄存器(DBGT)

表 15-4. DBGT 寄存器域描述

域	描述
7 TRGSEL	<p>触发类型——控制比较器 A 和 B 的匹配输入是否与调试模块中的操作码跟踪逻辑匹配。如果 TRGSEL 已设置，比较器 A 或 B 的匹配信号必须通过操作码跟踪逻辑传播，如果匹配地址的操作码实际已执行，则只有触发事件发送到 FIFO 逻辑。</p> <p>0 存取比较地址时触发(强制)</p> <p>1 如果比较地址的操作码已执行(标签)，则触发</p>
6 BEGIN	<p>开始/结束触发选择——控制 FIFO 在触发时开始填充还是以循环形式填充直到触发结束信息的捕获。在纯事件触发模式中，忽略这个位，所有调试运用都假定为起始跟踪。</p> <p>0 数据存储存储在 FIFO，直到触发(结束跟踪)</p> <p>1 触发启动数据存储(起始跟踪)</p>
3:0 TRG[3:0]	<p>选择触发模式——选择下面 9 个触发模式中的一个。</p> <p>0000 只有 A</p> <p>0001 A 或 B</p> <p>0010 A 然后 B</p> <p>0011 只有事件 B(存储数据)</p> <p>0100 A 然后只有事件 B(存储数据)</p> <p>0101 A 和 B 数据(满模式)</p> <p>0110 A 和非 B 数据(满模式)</p> <p>0111 范围内: $A \leq \text{地址} \leq B$</p> <p>1000 范围外: 地址 < A 或 地址 > B</p> <p>1001-1111 (无触发)</p>

15.4.3.9 调试状态寄存器(DBGS)

该寄存器为只读寄存器。

	7	6	5	4	3	2	1	0
读	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
写								
复位	0	0	0	0	0	0	0	0

 =未生效的或者预留的

图 15-9. 调试状态寄存器(DBGS)

表 15-6. DBGS 寄存器域描述

域	描述
7 AF	<p>触发匹配 A 标记——在调试运行开始时清除 AF，指示打开控制后是否满足触发匹配 A 条件。</p> <p>0 比较器 A 未匹配</p>

	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，指示打开控制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 DBGFC 中 ARM 的只读镜像。将 DBGFC 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 DBGFC 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

附录A 电器特性

A.1 简介

这一章包含 SC9RS08MZ8 微控制器目前的电气和时间特性。

A.2 参数分类

这个附录中显示的电气参数通过各种方式验证。为了让客户更好的理解下列分类被使用的地方，在下表中参数做了适当的标注。

表 A-1 参数分类

P	参数在每一个设备上的生产测试期间得到保证。
C	参数是通过测量跨过程变化统计有关样品大小的设计特性取得。
T	除非另有说明，参数是在典型条件下的典型设备的小型样例的设计特性取得。所有的在典型列中显示的数值都在这个范畴以内。
D	参数主要从仿真中派生而来。

注意：在参数表‘C’栏中显示的分类是它适用的地方。

A.3 最大绝对额定值

最大绝对额定值是强制的额定值，且在最大绝对值下功能工作是没有保证的。在 Table A-2 中，强制在指定限定值之外工作会影响芯片的可靠性或对芯片产生永久性地伤害。关于功能工作条件参见该节的其他表格。

该芯片包含了对高静态电压或电场的保护电路，可以避免因高静态电压或电子磁场产生的损坏。然而，建议采取正常的避免任何大于最大额定电压进入该高阻抗电路。如果不再使用的输入被限制于一个适合的逻辑电压(例如， V_{SS} 或 V_{DD})或者允许和引脚有关的可编程序的上拉电阻，操作的可靠性将会加强。

表 A-2 最大绝对额定值

参数	符号	值	单位
电源电压	V_{DD}	-0.3 到 +5.8	V
输入电压	V_{In}	-0.3 到 $V_{DD} + 0.3$	V
单个引脚限制的瞬间最大电流(应用到所有引脚) ^{1, 2, 3}	I_D	± 25	mA
V_{DD} 的最大电流	I_{DD}	120	mA
存储温度范围	T_{stg}	-55 到 +150	°C

¹ 输入当前限制于描述值。为了决定需要的限流电阻器的值，计算正(V_{DD})和负(V_{SS})箝位电压的电阻系数，然后使用两个电阻系数中较大的一个。

² 除了 \overline{RESET}/VPP 引脚仅在内部箝位到 V_{SS} ，所有功能性非电源引脚在内部箝位至 V_{SS} 和 V_{DD} 。

³ 电源必须维持调节 V_{DD} 的瞬间范围和最大电流情况。如果正的注入电流($V_{in} > V_{DD}$)大于 I_{DD} ，注入电流可能对 V_{DD} 溢出导致外部电源停止调整。确保外荷载 V_{DD} 分流大于最大注入电流。当 MCU 不耗电时，这将是更大的风险。例如：如果系统时钟停止，或时钟速率很低减小总功率功耗。

A.4 热特性

本节提供了工作温度范围、功率消耗和封装热阻的信息。在 I/O 引脚的功率消耗与片内逻辑和电压调整器电路和由用户决定而不被 MCU 设计控制的电路的功率消耗相比通常很小。为了获得 $P_{I/O}$ 功耗的计算，明确实际引脚电压和 V_{SS} 或 V_{DD} 以及和每个 I/O 引脚的电流乘积的不同之处。除了特别的大引脚电流(大负载)，引脚电压和 V_{SS} 或 V_{DD} 的不同之处非常小。

表 A-3 热特性

参数	符号	值	单位
温度范围 ^{1,2,3,4}			
64 引脚 QFP			
	1s	57	
	2s2p	43	

64 引脚 LQFP	1s	θ_{JA}	69	°C/W
	2s2p		54	
48 引脚 QFN	1s		84	
	2s2p		27	
44 引脚 LQFP	1s		73	
	2s2p		56	

¹ 连接处温度是晶元尺寸、片上功耗、封装热阻、安装点(主板)温度、周围温度、气流、主板上的其他组件功耗及主板热阻的函数。

² 连接处与环境的自然对流。

³ 1s-单层板，一层信号层

⁴ 2s2p-四层板，2层信号层和2层电源层

平均片上结合处温度(T_J)(°C)从等式 A-1 中得到:

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

其中:

T_A = 周边温度, °C

θ_{JA} = 封装热阻, 与周边结合处, °C/W

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

$P_{int} = I_{DD} \times V_{DD}$, 瓦特——片内功率

$P_{I/O}$ = 输入和输出引脚上的功耗——有用户决定

在大多数应用中, $P_{I/O} \ll P_{int}$, 可以忽略。 P_D 和 T_J (如果忽略 $P_{I/O}$) 的近似关系是:

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

解等式 1 和 2, 得 K:

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

其中 K 是与特殊部分相关的常量。已知 T_A , K 可以在等式 3 中通过测量 P_D (在平衡时) 确定。对于任何 T_A , 使用该 K 的值, 通过解等式 1 和 2 可以得到 P_D 和 T_J 。

A.5 ESD保护和闭锁抗扰度

虽然静电放电对这些芯片造成的伤害远小于对早期的 CMOS 电路, 但是正常处理规范需要避免暴露在静电放电之前。鉴定试验用来确保这些芯片能抵挡住暴露在合理的静电电平之前不会受到任何永久性伤害。

所有 ESD 测试都符合用于汽车电子等级综合电路的 AEC-Q100 强制测试资格。芯片的强制测试资格，通过人体模型(HBM)，机器模型(MM)和电荷设备模型(CDM)完成。

如果通过 ESD 脉冲测试后，芯片被认定有问题，则该芯片不再符合规范要求。通过高温的全部的 DC 参数和功能测试确保每片应用芯片符合常温规格说明，除非在芯片说明中有其它特别说明。

表 A-4 ESD 和闭锁测试条件

模型	描述	标志	值	单位
人体	串联电阻	R1	1500	Ω
	存储电容	C	100	pF
	每个引脚的脉冲数	-	3	-
机器	串联电阻	R1	0	Ω
	存储电容	C	200	pF
	每个引脚的脉冲数	-	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 A-5 ESD 和闭锁保护特性

编号	额定值	标志	最小	最大	单位
1	人体模型(HBM)	V_{HBM}	± 2000	-	V
2	机器模型(MM)	V_{MM}	± 200	-	V
3	电荷设备模型(CDM)	V_{CDM}	± 500	-	V
4	在 $T_A=125^\circ\text{C}$ 时，闭锁电流	I_{LAT}	± 100	-	mA

A.6 DC特性

这部分包含关于电源要求，I/O 引脚的特性，和各种工作模式下电源电流的信息。

特性	最小值	类型	最大值	单位	
供应电压	2.7	-	5.5	V	
温度				$^\circ\text{C}$	
	M	-40	-		125
	V	-40	-		105
	C	-40	-		85

表 A-7. DC 特性

编号	C	参数	符号	最小值	典型值 ¹	最大值	单位
----	---	----	----	-----	------------------	-----	----

1	P	输出高电压-低驱动($PTxDSn=0$) 5V, $I_{LOAD}=-2mA$ 3V, $I_{LOAD}=-0.6mA$ 5V, $I_{LOAD}=-0.4mA$ 3V, $I_{LOAD}=-0.24mA$	V_{OH}	$V_{DD} - 1.5$	-	-	V
		$V_{DD} - 1.5$ $V_{DD} - 0.8$ $V_{DD} - 0.8$		-	-	-	
2	P	输出高电压-高驱动($PTxDSn=1$) 5V, $I_{LOAD}=-10mA$ 3V, $I_{LOAD}=-3mA$ 5V, $I_{LOAD}=-2mA$ 3V, $I_{LOAD}=-0.4mA$	V_{OH}	$V_{DD} - 1.5$	-	-	V
		$V_{DD} - 1.5$ $V_{DD} - 0.8$ $V_{DD} - 0.8$		-	-	-	
3	D	输出低电压-低驱动($PTxDSn=0$) 5V, $I_{LOAD}=2mA$ 3V, $I_{LOAD}=0.6mA$ 5V, $I_{LOAD}=0.4mA$ 3V, $I_{LOAD}=0.24mA$	V_{OL}	-	-	1.5	V
		-		-	1.5	-	
4	D	输出低电压-高驱动($PTxDSn=1$) 5V, $I_{LOAD}=10mA$ 3V, $I_{LOAD}=3mA$ 5V, $I_{LOAD}=2mA$ 3V, $I_{LOAD}=0.4mA$	V_{OL}	-	-	1.8	V
		-		-	1.8	-	
5	D	输出高电流-所有端口最大总 I_{OH} 5V 3V	I_{OUT}	-	-	100 60	mA
6	D	输出低电流-所有端口最大总 I_{OL} 5V 3V	I_{OLT}	-	-	100 60	mA
7	P	输入高电压; 所有数字输入	V_{IH}	$0.65 \times V_{DD}$	-	-	V
8	P	输入低电压; 所有数字输入	V_{IL}	-	-	$0.35 \times V_{DD}$	V
9	T	输入滞后; 所有数字输入	V_{hys}	$0.06 \times V_{DD}$			mV
10	P	输入泄漏电流; 仅输入引脚 ²	$ I_{In} $	-	0.01	1	μA
11	P	高阻抗(掉电状态)泄漏电流 ²	$ I_{OZ} $	-	0.01	1	μA
12	P	内部上拉电阻 ³	R_{PU}	20	45	65	k Ω
13	P	内部下拉电阻 ⁴	R_{PD}	20	45	65	k Ω
14	C	输出电容; 所有非电源引脚	C_{In}	-	-	8	pF
15	P	POR 重置电压	V_{POR}	0.9	1.4	2.0	V
16	D	POR 重置时间	t_{POR}	10	-	-	μs
15	P	低压探测阈值—高量程 V_{DD} 下降 V_{DD} 上升	V_{LVDH}	4.2	4.3	4.4	V
		4.3		4.4	4.5		
16	P	低压探测阈值—低量程 V_{DD} 下降	V_{LVDL}	2.48	2.56	2.64	V

		V_{DD} 上升		2.54	2.62	2.7	
17	P	低压警告阈值—高量程					
		V_{DD} 下降	V_{LVWH}	4.2	4.3	4.4	V
		V_{DD} 上升		4.3	4.4	4.5	
18	P	低压警告阈值—低量程					
		V_{DD} 下降	V_{LVWH}	2.48	2.56	2.64	V
		V_{DD} 上升		2.54	2.62	2.7	
19	P	低压禁止复位/恢复滞后					
		5V	V_{hys}	-	100	-	mV
		3V		-	60	-	
20	P	带隙电压参考 出厂平衡为 $V_{DD}=5.0V$ Temp= $25^{\circ}C$	V_{BG}	1.185	1.20	1.215	V
21	D	dc 注入电流 ^{5,6,7,8} DC 注入电流 单引脚限制 $V_{IN}>V_{DD}$ $V_{IN}<V_{SS}$ 总 MCU 限制, 包含压力引脚和 $V_{IN}>V_{DD}$ $V_{IN}<V_{SS}$	$ I_{IC} $	0	-	2	mA
				0	-	-0.2	mA
				0	-	25	mA
				0	-	-5	mA

¹ 典型值是温度在 $25^{\circ}C$ 时测量的, 除非另有说明。

² $V_{In} = V_{DD}$ 或 V_{SS} 时测量。

³ $V_{In} = V_{SS}$ 时测量。

⁴ $V_{In} = V_{DD}$ 时测量。

⁵ 在瞬时和操作最大电流条件下, 电源必须维持在操作 V_{DD} 范围内。如果绝对注入电流($V_{In}>V_{DD}$)大于 I_{DD} , 注入电流就可能超出 V_{DD} , 并导致外部电源不可调控。确保外部 V_{DD} 载荷分流大于最大注入电流的电流。当 MCU 不消耗功率时, 就会有最大的风险。这样的例子包括: 如果当前无系统时钟, 或者如果时钟速率非常低, 这都会降低总体功耗。

⁶ 所有功能性非电源引脚均内部限制到 V_{SS} 和 V_{DD} 。

⁷ 输入电流必须限制在指定的值内。为了确定所需的限流电阻的值, 对应正(V_{DD})和负(V_{SS})箝位电压计算阻抗值, 取两个阻抗值中的更大的。

⁸ IRQ 没有 V_{DD} 箝位二极管, 不要把 IPQ 驱动为 V_{DD} 。

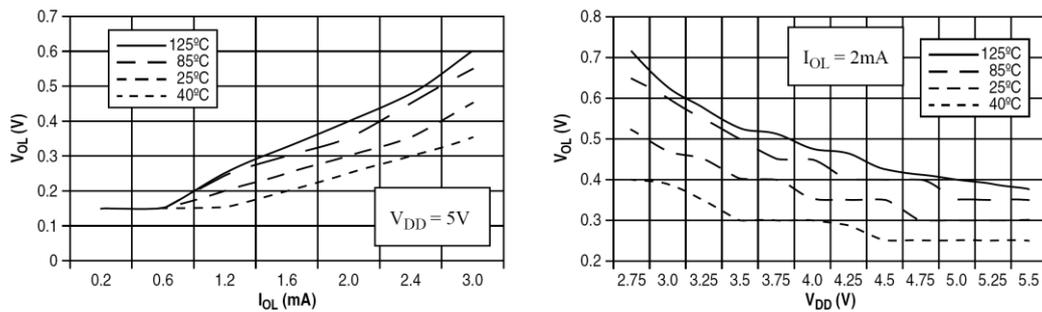


图 A-1. 典型低端驱动(接受)特性 — 低驱动($PTxDSn=0$)

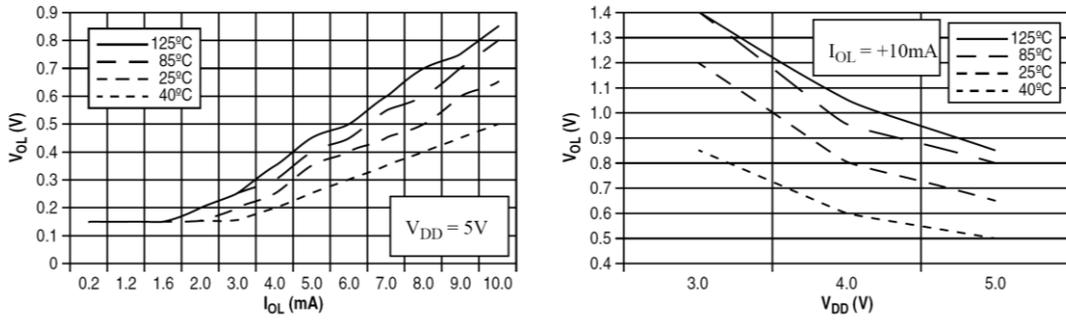


图 A-2. 典型低端驱动(接受)特性 — 高驱动($PTxDSn=1$)

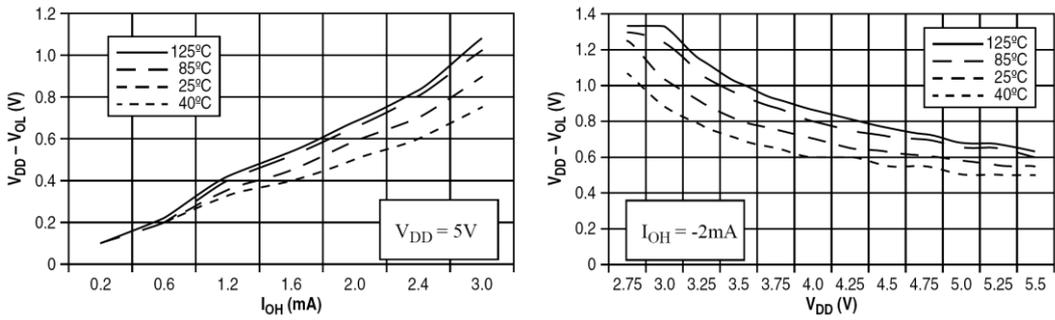


图 A-3. 典型高端驱动(发送)特性 — 低驱动($PTxDSn=0$)

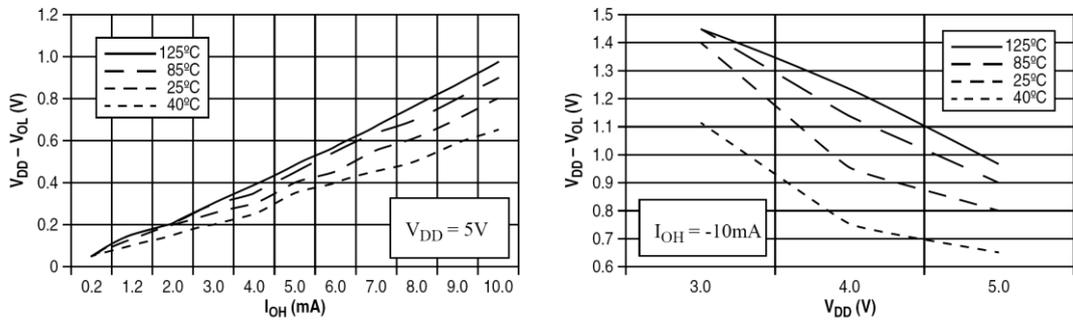


图 A-3. 典型高端驱动(发送)特性 — 高驱动($PTxDSn=1$)

A.7 电源电流特性

编号	C	参数	符号	V_{DD} (V)	类型 ¹	最大值	单位	温 度 (°C)
1	P	当 CPU 时钟=2MHz, fBus=1MHz 时测量的电源电流 ²	RI_{DD}	5	0.750	0.950	mA	-40 到 125°C
				3	0.570	0.770		
2	P	当 CPU 时钟=16MHz, fBus=8MHz 时测量的电源电流 ²	RI_{DD}	5	4.90	5.10	mA	-40 到 125°C
				3	3.50	3.70		

3	P	当 CPU 时钟=40MHz,fBus=20MHz 时测量的电源电流 ²	RI _{DD}	5	16.8	18.5	mA	-40 到 125℃
				3	11.5	12.5		
4	P	Stop2 模式电源电流	S2I _{DD}	5	0.900	18.0	μA	-40 到 85℃
						60		-40 到 125℃
				3	0.720	17.0	μA	
						50		-40 到 125℃
5	P	Stop3 模式电源电流	S3I _{DD}	5	0.975	20.0	μA	
						90		-40 到 125℃
				3	0.825	19.0	μA	
						85		-40 到 125℃
6	C	带有 RTI 的 Stop2 或 Stop3 ⁴ 模式	S23I _D DRTI	5	300	500	nA	
						500		-40 到 125℃
				3	300	500	nA	
						500		-40 到 125℃
7	C	带有 LVD 的 Stop3 模式 (LVDE=LVDSE=1)	S3I _{DD} LVD	5	110	180	μA	
				3	90	180		-40 到 125℃
8	C	带有振荡器启用时的 stop3 模式 (OSCSTEN=1)	S3I _{DD} OSC	5,3	5	8	μA	

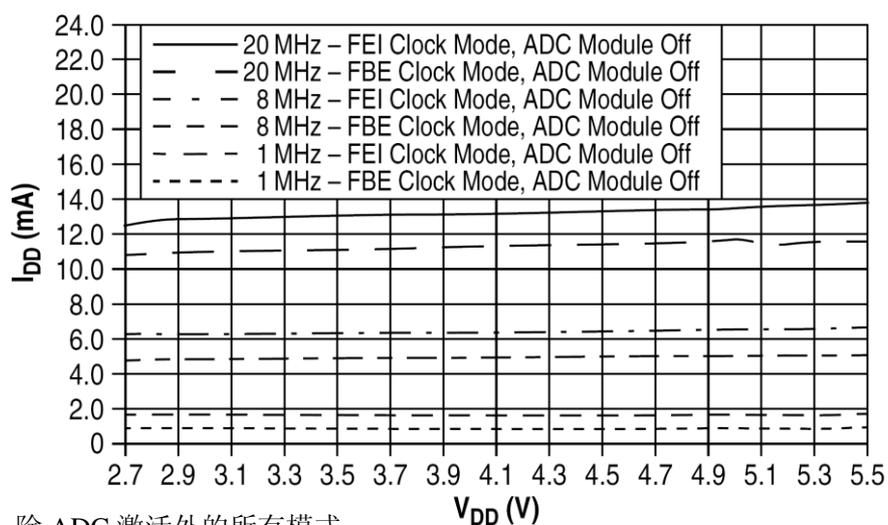
¹ 典型值是温度在 25℃时测量的，除非另有说明。见图 A-5 到 A-7 典型曲线交叉电压/温度。

² 除 ADC 使能外，所有模块都未激活。ICG 为 FBE 配置。在端口引脚上不包括任何 DC 负载。

³ 除 ADC 使能外，ICG 为 FBE 配置并且在端口引脚不包括任何 DC 负载。

⁴ 大多数用户都希望可以使用时 stop2 或 stop3 的自动唤醒来待其更高电流的风带模式。典型的等待模式为 f_{Bus}=1MHz, 5V, 500μA。

⁵ 以下条件下的给定值：低量程操作(RANGE=0)32.786 的晶振，低功率模式(HGO=0)，始终监控丧失(LOCD=1)。



注：除 ADC 激活外的所有模式

图 A-5. 典型 FBE 和 FEI 运行模式下 I_{DD}, I_{DD} vs. V_{DD}

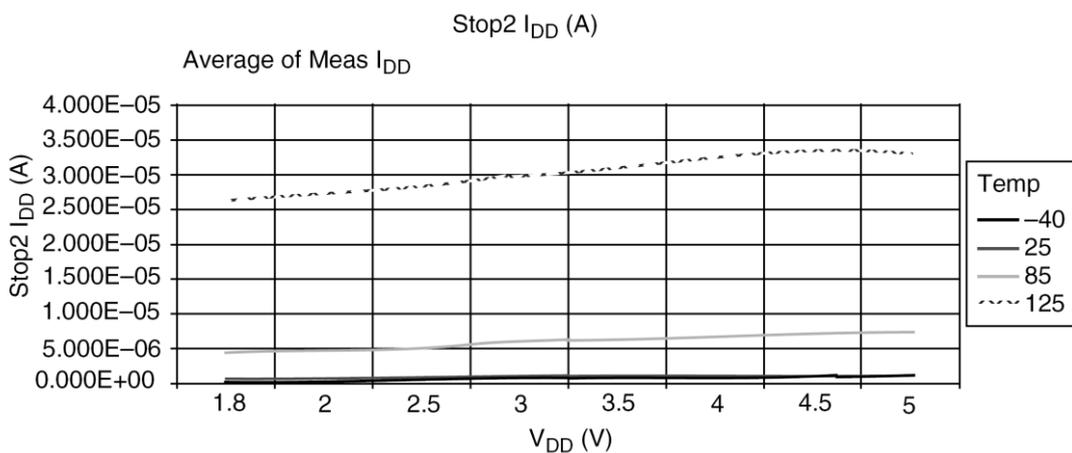


图 A-6. 典型 Stop2 下 I_{DD}

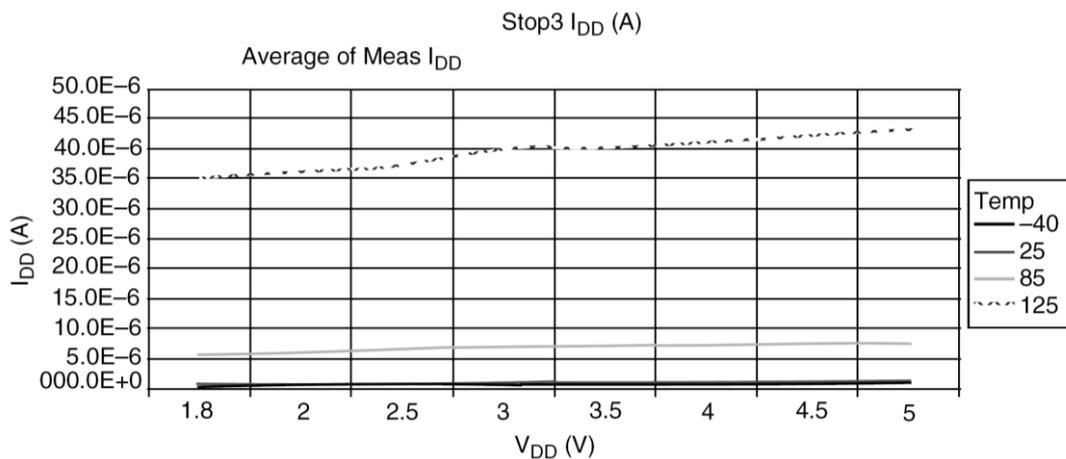


图 A-7. 典型 Stop3 下 I_{DD}

特性	条件	C	符号	最 小 值	类型 ¹	最 大 值	单位
电源电流 ADLPC=1 ADLSMP=1 ADCO=1		T	I _{DDAD}	-	133	-	μA
电源电流 ADLPC=1 ADLSMP=0 ADCO=1		T	I _{DDAD}	-	218	-	μA
电源电流 ADLPC=0 ADLSMP=1 ADCO=1		T	I _{DDAD}	-	327	-	μA
电源电流 ADLPC=0 ADLSMP=0 ADCO=1	V _{DDAD} ≤5.5V	P	I _{DDAD}	-	582	990	μA
电源电流	停止, 复位, 模块关		I _{DDAD}	-	0.011	1	μA
ADC 同步时钟源 t _{ADACK} =1/f _{ADACK}	高速(ADLPC=0)	P	f _{ADACK}	2	3.3	5	MHzS
	低速(ADLPC=1)			1.25	2	3.3	
转换时间 (包括采样时间)	短采样(ADLSMP=0)	P	t _{ADC}	-	20	-	ADCK 周期
	长采样(ADLSMP=1)			-	40	-	
采样时间	短采样(ADLSMP=0)	P	t _{ADS}	-	3.5	-	ADCK 周期
	长采样(ADLSMP=1)			-	23.5	-	
未调整误差总数	10 位模式	P	E _{TUE}	-	±1	±2.5	LSB ²
	8 位模式			-	±0.5	±1.0	
差分非线性误差	10 位模式	P	DNL	-	±0.5	±1.0	LSB ²
	8 位模式			-	±0.3	±0.5	
单调性和无丢码保证							
积分非线性误差	10 位模式	C	INL	-	±0.5	±1.0	LSB ²
	8 位模式			-	±0.3	±0.5	
零刻度误差	10 位模式	P	E _{ZS}	-	±0.5	±1.5	LSB ²
	8 位模式			-	±0.5	±0.5	
满刻度误差 V _{ADIN} =V _{DDA}	10 位模式	P	E _{FS}	-	±0.5	±1.5	LSB ²
	8 位模式			-	±0.5	±0.5	
量化误差	10 位模式	P	E _Q	-	-	±0.5	LSB ²
	8 位模式			-	-	±0.5	
输入漏电误差 Pad leakage ³ *R _{AS}	10 位模式	D	E _{IL}	-	±0.2	±2.5	LSB ²
	8 位模式			-	±0.1	±1	
温度传感器范围	-40°C 到 25°C	T	m	-	3.266	-	mV/°C

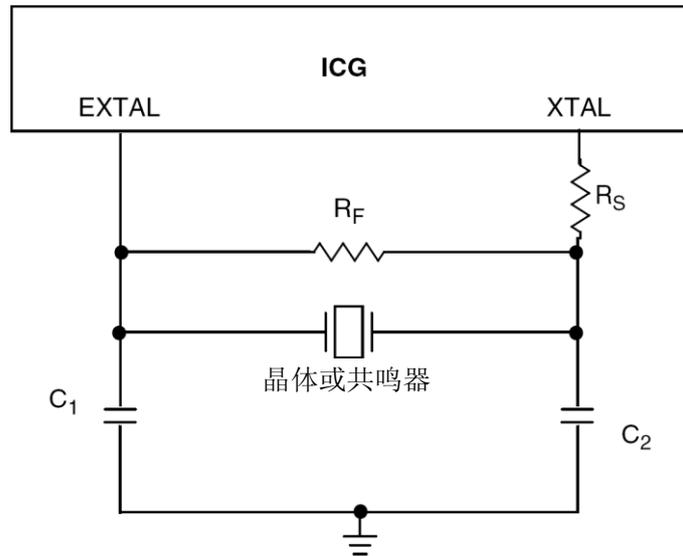
	25°C 到 125°C			-	3.638	-	
温度传感器电压	25°C	P	V _{TEMP25}	-	1.396	-	V

¹ 典型值设定 V_{DDAD}=5V,Temp=25°C,f_{ADCK}=1.0MHz 除非另有说明。典型值仅用于参考,并未在生产中测试。

² 1LSB=(V_{REFH}-V_{REFL})/2^N

³ 基于输入板泄漏电源。参考板的电器特性。

A.9 内部时钟发生器模块特性



特性	符号	最小值	类型 ¹	最大值	单位
负载电容	C ₁	见注 ²			
	C ₂				
反馈寄存器					
低范围(32k 到 100kHz)	R _F		10		MΩ
高范围(1M 到 16MHz)			1		MΩ
串联电阻					
低范围					
低增加(HGO=0)		-	0	-	
高增加(HGO=1)		-	100	-	
高范围					
低增加(HGO=0)	R _S	-	0	-	kΩ
高增加(HGO=1)					
≥8MHz		-	0	-	
4MHz		-	10	-	
1MHz		-	20	-	

¹ 典型值设定 V_{DDAD}=5V,Temp=25°C,f_{ADCK}=1.0MHz 除非另有说明。典型值仅用于参考,并未在生产中测试。

² 参见晶体或共鸣器厂商建议。

A.9.1 ICG 频率规范

表 A-12. ICG 频率规格
($V_{DDA}=V_{DDA}$ (最小值)到(最大值), 周围温度范围-40 到 125°C)

编号	C	特性	符号	最小值	类型 ¹	最大值	单位	
1	T	晶体振荡器或共鸣器 (REFS=1)(基本模式晶体或陶瓷共鸣器)						
		低范围	f _{lo}	32	-	100	kMz	
		高范围						
		高增加, FBE(HGO=1,CLKS=10)	f _{hi_byp}	1	-	16	MHz	
		高增加, FEE(HGO=1,CLKS=11)	f _{hi_eng}	2	-	10	MHz	
低功率, FBE(HGO=0,CLKS=10)	f _{hp_byp}	1		8	MHz			
低功率, FEE(HGO=0,CLKS=11)	f _{hp_eng}	2		8	MHz			
2	T	输入时钟频率 (CLKS=11,REFS=0)						
		低范围	f _{lo}	32	-	200	kHz	
		高范围	f _{hi_eng}	2	-	10	MHz	
3	T	输入时钟频率 (CLKS=10,REFS=0)	f _{Extal}	0	-	40	MHz	
4	P	内部参考频率(未修饰)	f _{ICG RCLK}	182.25	243	303.75	kHz	
5	P	输入时钟的职责周期 (REFS=0)	t _{dc}	40	-	60	%	
6	P	输出时钟 ICGOUT 频率 CLKS=10, REFS=0	f _{ICGOUT}	f _{Extal} (最小)	-	f _{Extal} (最大)	MHz	
		所有其他例子		f _{lo} (最小)	-	f _{ICGDCLKmax} (最大)		
7		最小 DCO 时钟(ICGDCLK)频率	f _{ICGDCLKmin}	8	-		MHz	
8		最大 DCO 时钟(ICGDCLK)频率	f _{ICGDCLKmax}		-	40	MHz	
9	P	自时钟模式(ICGOUT)频率 ²	f _{Self}	f _{ICGDCLKmin}		f _{ICGDCLKmax}	MHz	
10	P	自时钟模式重置(ICGOUT)频率	f _{Self_reset}	5.5	8	10.5	MHz	
		失去参考频率 ³						

11	T	低范围 高范围	f_{LOR}	5 50		25 500	kHz
12	T	失去 DCO 频率 ⁴	f_{LOD}	0.5		1.5	MHz
13	T	晶体开始时间 ^{5,6} 低范围 高范围	t_{CSTL}	-	430	-	ms
			t_{CSTH}	-	4	-	
14	P	FLL 锁定时间 ⁷ 低范围 高范围	t_{Lockl}	-		5	ms
			t_{Lockh}	-		5	
15	T	FLL 频率解锁范围	n_{Unlock}	- 4*N		4*N	counts
16	T	FLL 频率锁定范围	n_{Lock}	- 2*N		2*N	counts
17	T	ICGOUT 周期 Jitter, ⁸ 在 f_{ICGOUT} 最大长周期 Jitter 测得 (平均超过 2ms 间隔)	C_{Jitter}	-		0.2	% f_{ICG}
18	T	振荡器振幅(最高点间) HGO=0 HGO=1	V_{oscamp}	-	1	-	V
				-	V_{DD}	-	
19	MC9S08AWxx: 内部振荡器平衡频率的误差 ⁹						
	C	$V_{DD}=2.7-5.5V$, (常温)	ACC_{int}	-	± 0.5	± 2	%
	P	$V_{DD}=5.0V \pm 10\%$, -40 到 125°C		-	± 0.5	± 2	%
	S9S08AWxx: 内部振荡器平衡频率的偏差 ⁹						
	C	$V_{DD}=2.7-5.5V$, (常温)	ACC_{int}	-	± 0.5	± 1.5	%
	P	$V_{DD}=5.0V \pm 10\%$, -40 到 85°C		-	± 0.5	± 1.5	%
P	$V_{DD}=5.0V \pm 10\%$, -40 到 125°C		-	± 0.5	± 2	%	

¹ 典型值是 $V_{DD}=5.0V$, 温度在 25°C 时测量的, 除非另有说明。

² 当 FLL 回路打开时, 自时钟频率由 DOC 产生。

³ 丧失参考频率由内部探测的参考频率, 如果该频率不在所需范围内, 将 ICG 转换到自时钟模式。

⁴ 丧失 DCO 频率由内部探测的 DCO 频率, 如果该频率不在所需范围内, 将 ICG 转换到 FLL 旁路外部模式(如果外部参考存在)。

⁵ 该参数在限定特性前胜于 100% 测试。

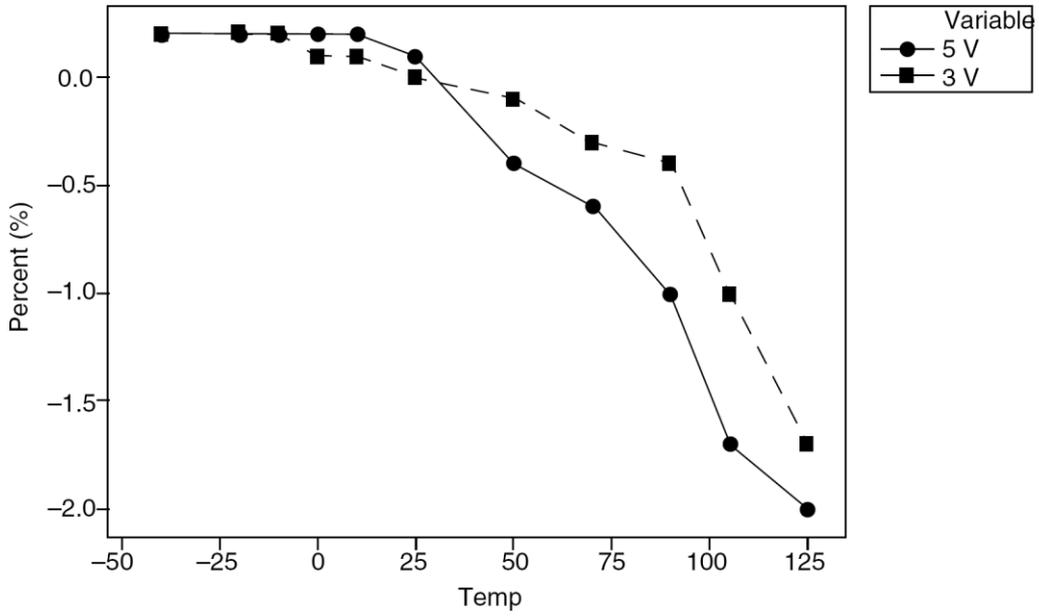
⁶ 正确的 PC 板层生产必须遵守完成规格。

⁷ 该规范适用 FLL 进入使用内部或外部模式后, 需求锁定时钟周期内。如果晶体/共鸣器正被用作参考, 该规范假设它已经运行。

⁸ Jitter 是在最大 f_{ICGOUT} 指定间隔内测得的编程频率的平均偏离。测量的设备电源是由滤波器提供, 时钟是稳定的外部始终信号。噪声通过 V_{DDA} 和 V_{SSA} 注入 FLL 电路, 并且在晶体振荡器中的变化在给定时间间隔中加大 C_{Jitter} 的百分比。

⁹ 见图 A-9。

内部振荡器的平衡频率偏离



25°C, 3.0V 设备平衡

图 A-9. 内部振荡器的平衡频率偏离

A.10 AC特性

本节描述了每个外围系统得 AC 时序特性。更多关于总线时钟如何产生的资料，详见第 8 章，“内部时钟发生器(S08ICGV4)”。

A.10.1 控制时序

表 A-13. 控制时序

编号	C	参数	符号	最小值	类型 ¹	最大值	单位
1		总线频率($t_{cyc}=1/f_{Bus}$)	f_{Bus}	dc	-	20	MHz
2	P	实时中断内部振荡器周期	t_{RTI}	700		1300	μs
3		外部复位脉冲宽幅 ² ($t_{cyc}=1/f_{Self_reset}$)	t_{extrst}	$1.5 \times t_{Self_reset}$		-	ns
4		复位低驱动 ³	t_{rstdrv}	$34 \times t_{cyc}$		-	ns
5		激活后台调试模式锁定建立时间	t_{MSSU}	25		-	ns
6		激活后台调试模式锁定保持时间	t_{MSH}	25		-	ns
7		IRQ 脉冲宽幅					
		异步通道 ²	t_{LIH}	100	-	-	ns

		同步通道 ³	t_{LIL}	$1.5 \times t_{cyc}$				
8		KBIPx 脉冲宽幅			-	-	ns	
		异步通道 ²	t_{LILH}	100				
		同步通道 ³	t_{LIL}	$1.5 \times t_{cyc}$				
9	T	端口升降时间 – 低输出驱动($PTxDS=0$, 负载 $=50pF$) ⁵						
		斜率控制禁用($PTxSE=0$)	t_{Rise}	-	40	-	ns	
		斜率控制启用($PTxSE=1$)	t_{Fall}	-	75	-		
		端口升降时间 – 高输出驱动($PTxDS=1$)(负载 $=50pF$)						ns
		斜率控制禁用($PTxSE=0$)	t_{Rise}	-	11	-		
		斜率控制启用($PTxSE=1$)	t_{Fall}	-	35	-		

¹ 典型值是 $V_{DD}=5.0V$, 温度在 $25^{\circ}C$ 时测量的, 除非另有说明。

² 这是被识别为复位引脚请求所保证的最小脉冲宽度。更窄的脉冲不保证覆盖来自内部时钟源的复位请求。

³ 当初始任意复位时, 内部电路驱动复位引脚为低, 实现大约 34 个总线周期, 之后大约 38 个总线周期后采样复位引脚来辨别内部请求中的外部复位请求。

⁴ 这是确保能够通过引脚同步电路的最小脉冲宽度。更窄的脉冲比一定能被识别。在停止模式下, 同步设备是设置为旁路的, 所以更窄的脉冲在该情况下可以被识别。

⁵ 时序以 $V_{DD}20\%$ 到 80% 电平显示。工作温度范围 $-40^{\circ}C$ 到 $125^{\circ}C$ 之间。

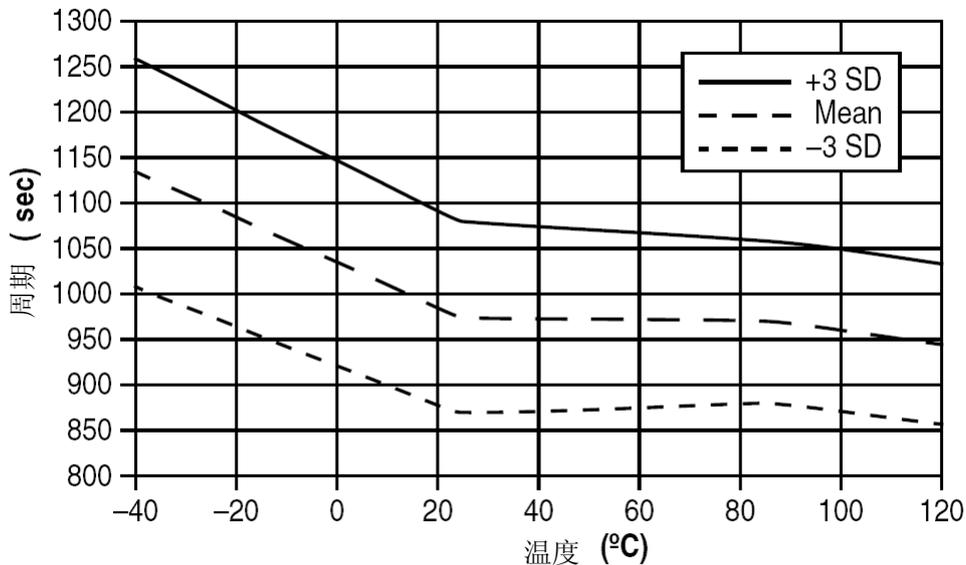


图 A-10. 典型 RTI 时钟周期 vs. 温度

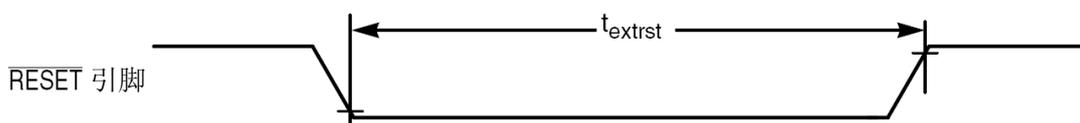


图 A-11. 复位时序

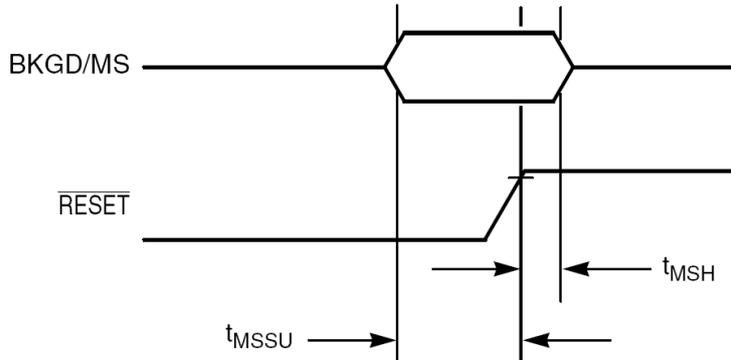


图 A-12. 激活后台调试模式锁定时序

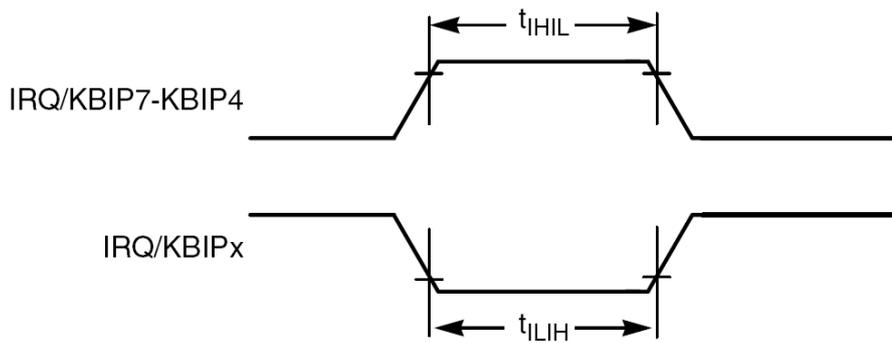


图 A-13. IRQ/KBIPx 时序

A.10.2 定时器/PWM(TPM)模式时序

同步设备电路决定可识别的最短输入脉冲或可以用定时器计数器的可选外部源的最快时钟。这些同步设备由当前的总线速率时钟进行操作。

表 A-14. TPM 输入时序

功能	符号	最小值	最大值	单位
外部时钟频率	f_{TPMext}	dc	$f_{Bus}/4$	Hz
外部时钟周期	t_{TPMext}	4	-	t_{cyc}
外部时钟高时期	t_{clkh}	1.5	-	t_{cyc}
外部时钟低时期	t_{clkl}	1.5	-	t_{cyc}
输入捕捉脉冲宽度	t_{ICPW}	1.5	-	t_{cyc}

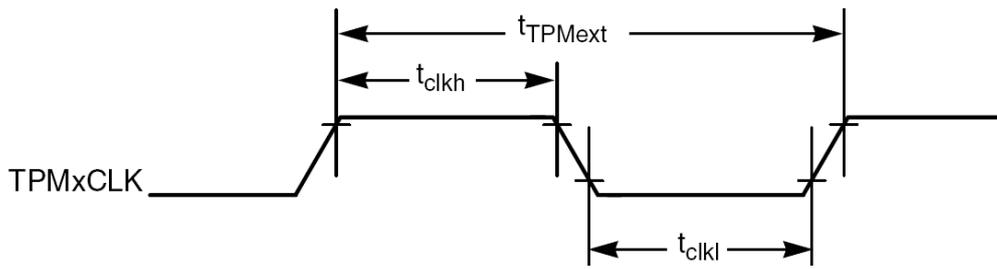


图 A-14. 计时器外部时钟

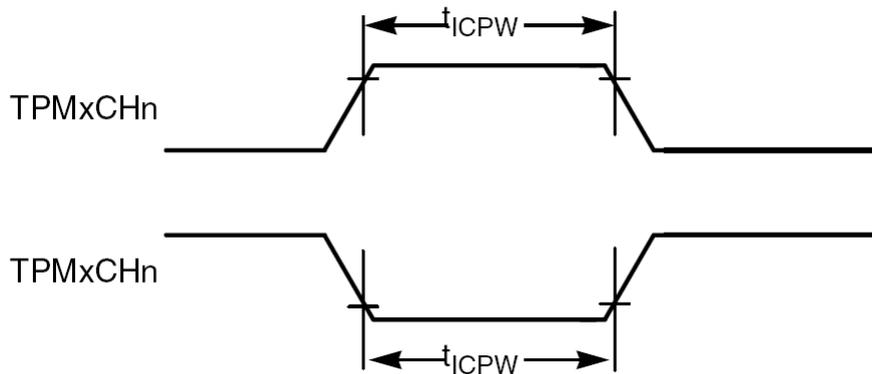


图 A-15. 计时器输入捕捉脉冲

A.11 SPI特性

表 A-15 和图 A-16 通过图 A-19 描述了 SPI 系统得时序要求。

表 A-15. SPI 电器特性

编号 ¹	特性 ²	符号	最小值	最大值	单位
	运行频率 ³				
	主机	f_{op}	$f_{Bus}/2048$	$f_{Bus}/2$	Hz
	从机	f_{op}	dc	$f_{Bus}/4$	
1	周期时间				
	主机	t_{SCK}	2	2048	t_{cyc}
	从机	t_{SCK}	4	-	t_{cyc}
2	使能引导时间				
	主机	t_{Lead}	-	1/2	t_{SCK}
	从机	t_{Lead}	1/2	-	t_{SCK}
3	使能滞后时间				
	主机	t_{Lag}	-	1/2	t_{SCK}
	从机	t_{Lag}	1/2	-	t_{SCK}
4	时钟(SPCK)高时期主机和从机	t_{SCKH}	$1/2t_{SCK}-25$	-	ns
5	时钟(SPCK)低时期主机和从机	t_{SCKL}	$1/2t_{SCK}-25$	-	ns
	数据上线时间(输入)				

6		主机	$t_{SI(M)}$	30	-	ns
		从机	$t_{SI(S)}$	30	-	ns
7	数据保持时间(输入)	主机	$t_{HI(M)}$	30	-	ns
		从机	$t_{HI(S)}$	30	-	ns
8	存取时间, 从机 ⁴		t_A	0	40	ns
9	禁止时间, 从机 ⁵		t_{dis}	-	40	ns
10	数据上线时间(输出)	主机	t_{SO}	25	-	ns
		从机	t_{SO}	25	-	ns
11	数据保持时间(输出)	主机	t_{HO}	- 10	-	ns
		从机	t_{HO}	- 10	-	ns

¹ 参见图 A-16 到图 A-19。

² 所有时序以 $V_{DD}20\%$ 到 80% 电平显示, 除非另有说明; 所有 SPI 引脚负载为 100pF 。所有时序初设回速率控制禁止并且 SPI 输出引脚为高驱动使能。

³ 根据垫板输入特性, 最大波特率必须限制在 5MHz 内。

⁴ 从高阻抗状态激活数据的时间。

⁵ 高阻抗状态的保持时间。

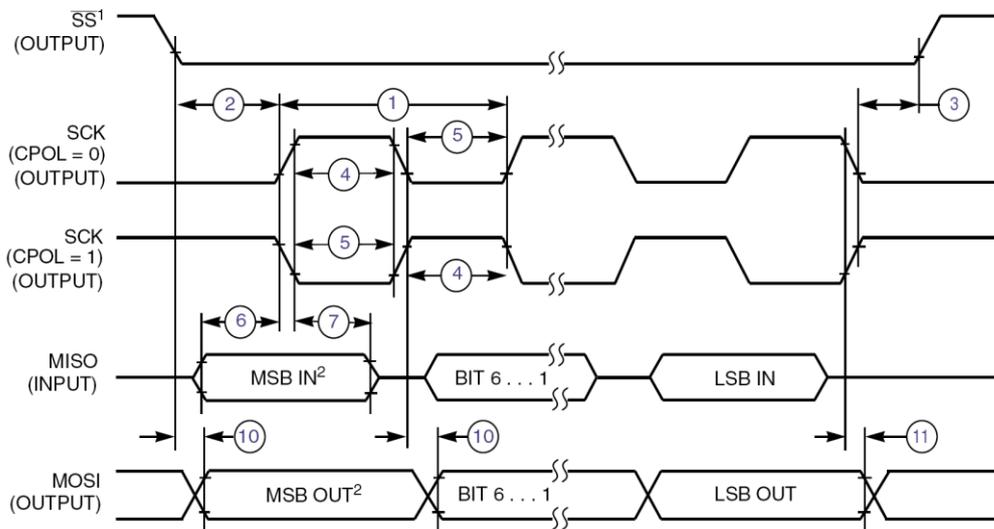


表 A-16. SPI 主机时序(CPHA=0)

注:

1. \overline{SS} 输出模式(MODFEN=1,SSOE=1)。

2. LSBF=0。当 LSBF=1 时, 位次序为 LSB, 位 1, ..., 位 6, MSB。

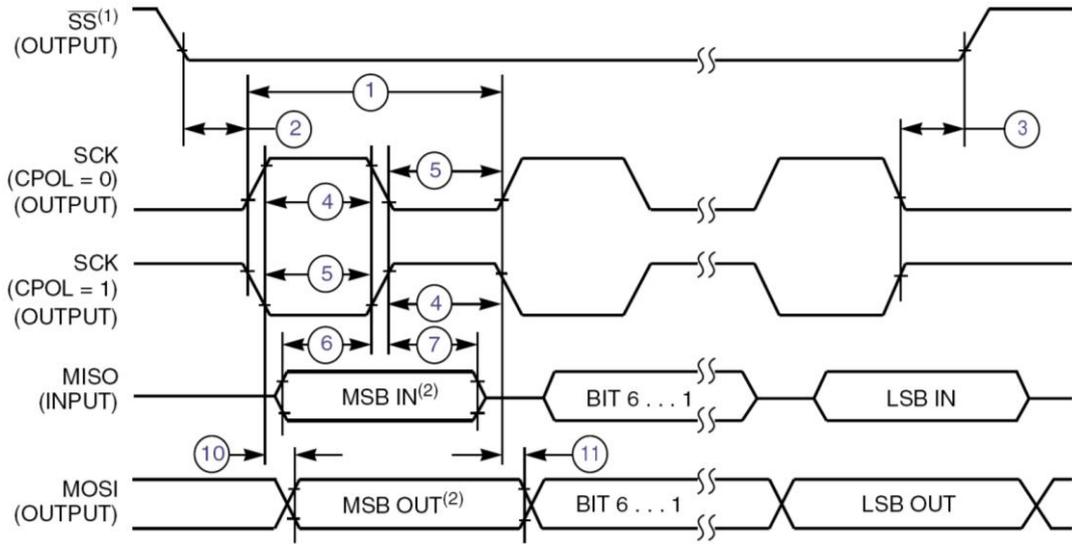


图 A-17. SPI 主机时序(CPHA=1)

注:

1. \overline{SS} 输出模式(MODFEN=1,SSOE-1)。
2. LSBF=0。当 LSBF=1 时, 位次序为 LSB, 位 1, ..., 位 6, MSB。

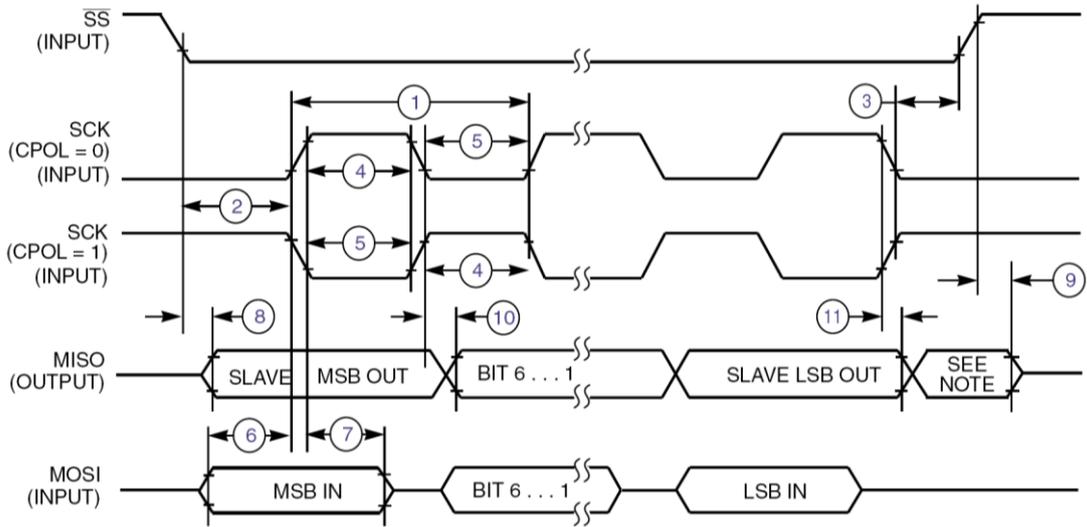


图 A-18. SPI 从机时序(CPHA=0)

注:

1. 未被定义时, 通常 MSB 特性为标准。

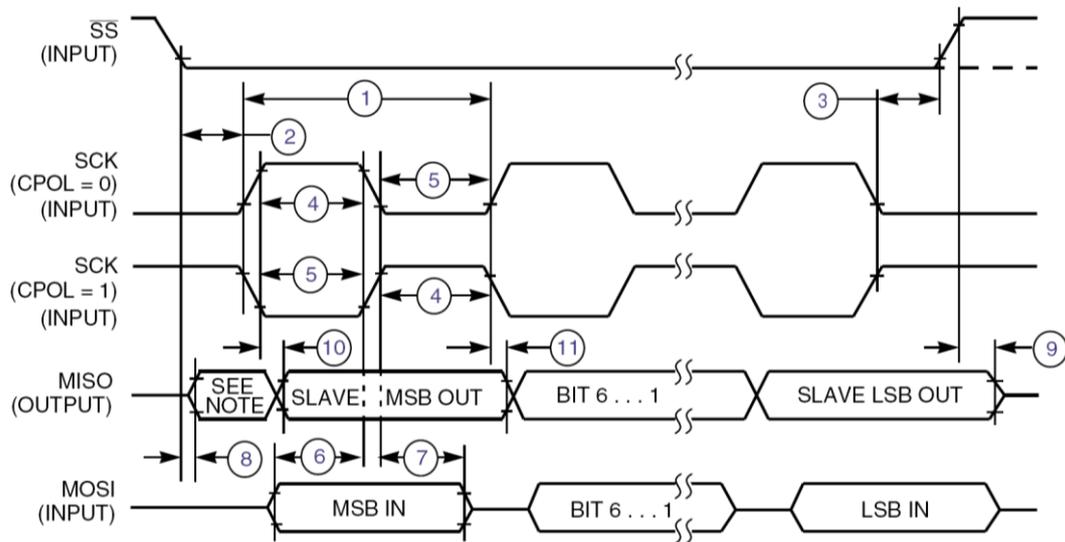


图 A-19. SPI 从机时序(CPHA=1)

注:

1. 未被定义时, 通常 MSB 特性为标准。

A.12 FLASH特性

该节详细描述了 Flash 存储器编程/擦除时间及编程-擦除强度的详细描述。

编程和擦除操作除正常 V_{DD} 电源外不需要任何特殊电源。更多关于编程/擦除操作的信息, 参见第 4 章, “存储器”。

表 A-16. FLASH 特性

编号	C	特性	符号	最小值	类型 ¹	最大值	单位
1	P	编程/擦除的电源电压	$V_{prog/erase}$	2.7		5.5	V
2	P	读操作的电源电压	V_{Rdad}	2.7		5.5	V
3	P	内部 FCLK 频率 ²	f_{FCLK}	150		200	kHz
4	P	内部 FCLK 周期 (1/FCLK)	t_{Fcyc}	5		6.67	μs
5	P	字节编程时间(任意位置) ³	t_{prog}	9			t_{Fcyc}
6	C	字节编程时间(突发模式) ³	t_{Burst}	4			t_{Fcyc}
7	P	页擦除时间 ³	t_{page}	4000			t_{Fcyc}
8	P	块擦除时间 ³	t_{Mass}	20000			t_{Fcyc}
9	C	编程/擦除强度 ⁴					cycles
		T_L 到 $T_H = -40^\circ C$ 到 $125^\circ C$		10000	-	-	
		$T=25^\circ C$		-	100000	-	
10	C	数据保持 ⁵	t_{D_ret}	15	100	-	years

- ¹ 典型值是 $V_{DD}=5.0V$ ，温度在 $25^{\circ}C$ 时测量的，除非另有说明。
- ² 该时钟频率由软件设置控制。
- ³ 这些值由硬件状态机械控制。用户代码不需要计数周期。提供该信息的目的是为了计算编程和擦除的大约时间。
- ⁴ FLASH 的典型强度的评估用于该系列产品 9S12Dx64。关于更多飞思卡尔半导体如何定义典型强度的信息，请参见工程报告 EB619/D，非易失性存储器典型强度。
- ⁵ 典型数据保持值基于在高温时测量的技术的内在性能，并使用 Arrhenius 公式降到 $25^{\circ}C$ 。关于更多飞思卡尔半导体如何定义典型数据保持的信息，请参见工程报告 EB618/D，非易失性存储器典型数据保持。

A.13 EMC性能

电磁兼容性的 (EMC) 性能高度依赖于 MCU 存在的环境。板的设计和布局，电路拓扑选择，外部组件的位置和特性，以及 MCU 的软件运行对 EMC 性能发挥有重要作用。系统设计者应该参考飞思卡尔的应用手册，像 AN2321, AN1050, AN1263, AN2764 以及 AN1259 的专门针对最优 EMC 性能的意见和指导。

A.13.1 辐射排放量

微控制器辐射 RF 的排放量根据 IEC61967-2 和 SAEJ1752/3 标准使用 TEM/GTME 单元方式在 150kHz 到 1GHz 测量。测量是在安装了自定义的 EMC 评估电路板微控制器上执行的，同时运行专门的 EMC 测试软件。来自微控制器的辐射排放量在 TEM 单元两个封装方向（北部和东部）测量。有关评估结果、条件和设置等的更多信息，请参考本设备的 EMC 评估报告。

所有方位中已测配置的最大 RF 辐射排放性应小于或等于已报告排放水平。

表 A-17. 辐射排放量

参数	符号	条件	f_{OSC}/f_{BUS}	频率	水平 ¹ (最大值)	单位	
辐射排放量， 导电域	V_{RE_TEM}	$V_{DD}=5.5V$ $T_A=25^{\circ}C$ 64QFP 封装类型	4MHz 晶体 20MHz 总线	0.15-50MHz	16	dB μ V	
				50-150MHz	3		
				150-500MHz	1		
				500-1000MHz	0		
				IEC 水平	L		-
				SAE 水平	2		-
			32kHz 晶体 8MHz 总线	0.15-50MHz	-1	dB μ V	
				50-150MHz	-6		
				150-500MHz	-10		
				500-1000MHz	-10		
				IEC 水平	N		-
				SAE 水平	1		-

¹ 数据基于条件测试结果。

A.13.2 传导暂时磁化

微控制器传导暂时磁化的测量与内部飞思卡尔测试方法一致。测量是在安装了自定义的 EMC 评估电路板微控制器上执行的，同时运行与测试方法相配合的专门 EMC 测试软件。

磁化传导是由每个微控制器的引脚流入的短暂磁化系数所决定的。短暂波形和流入方法基于 ICE61000-4-4(EFT/B)。在测试配置中，任何引脚引起执行降级所需的短暂电压是大于或等于报告水平的，除下面表格中的脚注显示以外。

表 A-18. 传导磁化

参数	符号	条件	f _{OSC} /f _{BUS}	结果	振幅 ¹ (最小值)	单位
传导电磁性，高速电瞬时/突发(EFT/B)	V _{CS_EFT}	V _{DD} =5.5V T _A =25°C 64QFP 封装类型	32768Hz 晶体 2MHz 总线	A	±0, ±2.0 ²	kV
				B	±2.5	
				C	±3.0	
				D	>±3.0	

¹ 数据基于条件测试结果。在生产中未测试。

² *RESET* 引脚受 220V 的最小供应瞬时影响。所有其它引脚导致 A 一直到 2000V 的最小值。

表 A-19 描述了磁化性能的分类。

表 A-18. 磁化性能分类

结果	执行标准	
A	无故障	在暴露时和暴露后 MCU 按照原计划执行。
B	自我复位故障	在暴露时，MCU 不按原计划执行。在暴露移除后，MCU 自动返回常态操作。
C	软件故障	在暴露时，MCU 不按原计划执行。直道暴露移除后，MCU 才返回常态操作并且 <i>RESET</i> 引脚被设置。
D	硬件故障	在暴露时，MCU 不按原计划执行。直道暴露移除后，MCU 才返回常态操作并且 MCU 功率循环。
E	损坏	在暴露时，MCU 不按原计划执行。由于物理损坏或者其他永久性能退化，MCU 不能返回到正确的操作。

附录B 订购信息和机械图

B.1 订购信息

本章包含 MC9S08AW60 系列设备的订购单号。设备单号系统例子如下。

表 B-1. 客户和工业设备编号系统

设备号码 ¹	存储器	可用封装 ²
-------------------	-----	-------------------

	FLASH	RAM	类型
MC9S08AW60	63280	2048	64 引脚 LQFP
MC9S08AW48	49152		64 引脚 QFP
MC9S08AW32	32768		48 引脚 QFN
MC9S08AW16	16384	1024	44 引脚 LQFP

¹ 每个设备的各个模块的完整描述见表 1-1

² 封装信息见表 B-3。

表 B-2. 汽车设备编号系统

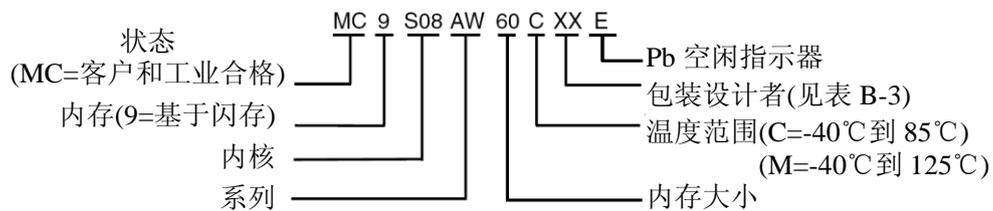
设备号码 ¹	存储器		可用封装 ²
	FLASH	RAM	类型
MC9S08AW60	63280	2048	64 引脚 LQFP
MC9S08AW48	49152		48 引脚 QFN
MC9S08AW32	32768		44 引脚 LQFP
MC9S08AW16	16384	1024	48 引脚 QFN 44 引脚 LQFP

¹ 每个设备的各个模块的完整描述见表 1-1

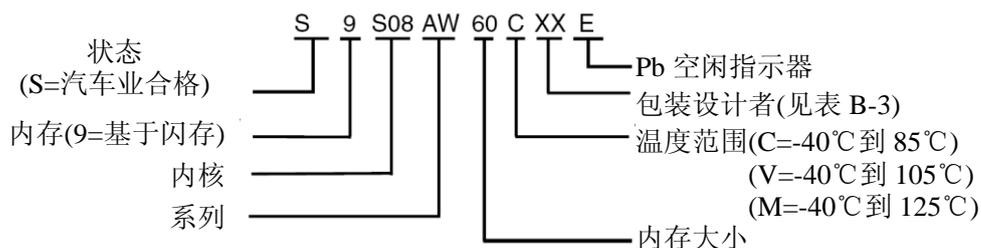
² 封装信息见表 B-3。

B.2 订购部分编号系统

B.2.1 客户和工业订购部分编号系统



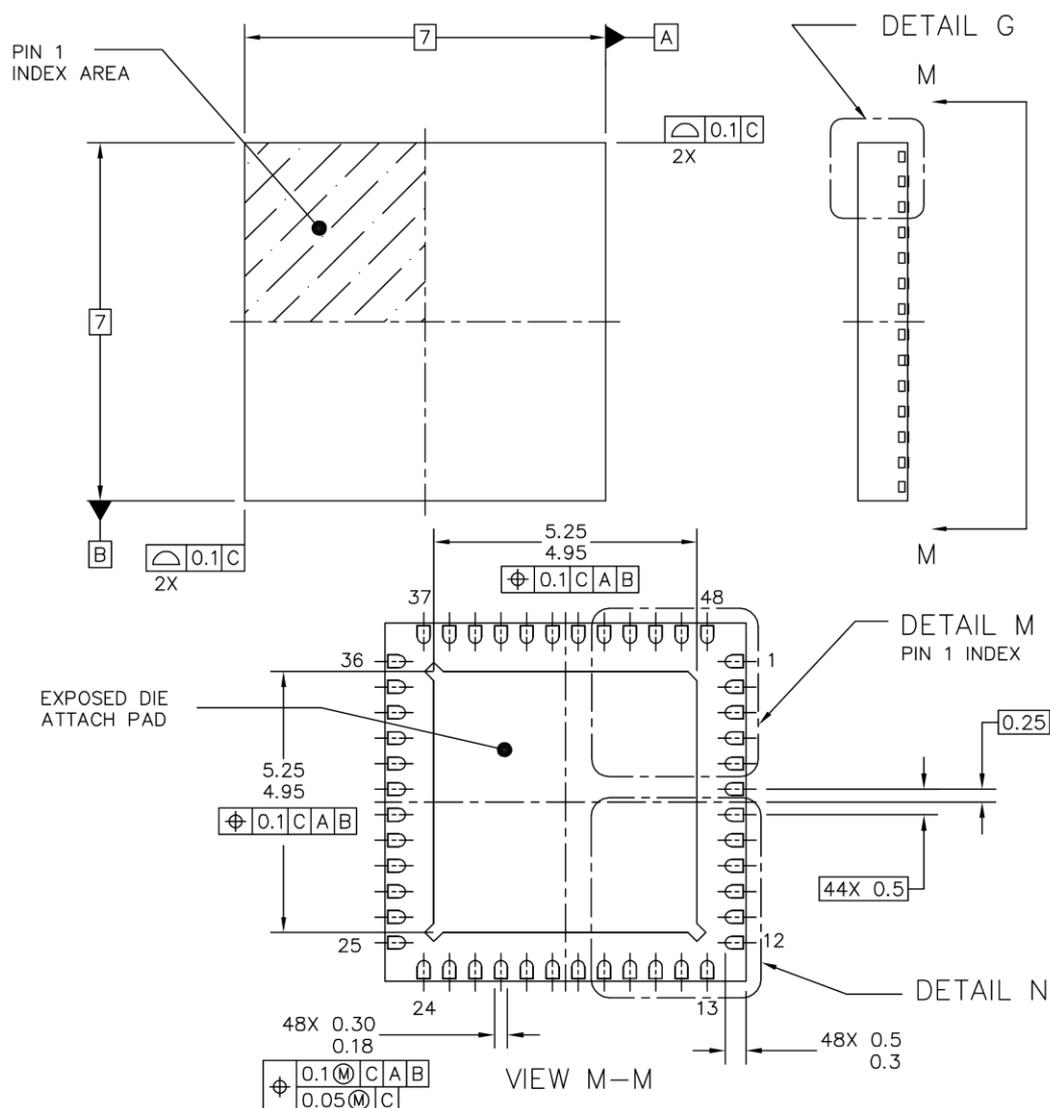
B.2.2 汽车订购部分编号系统



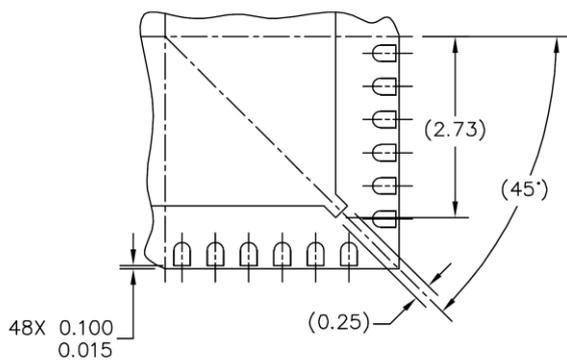
B.3 机械图

以下页包含 MC9S08AW60 系列封装选项的机械规格。见表 B-3 与每个封装类型相符合的文档编号。

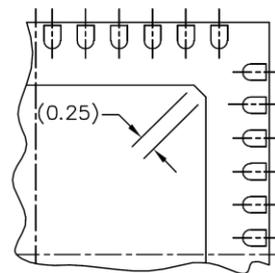
引脚数	类型	指定	文档编号
44	LQFN	FG	98ASS23225W
48	QFN	FD	98ARH99048A
64	LQFP	PU	98ASB23234W
64	QFP	FU	98ASB42844B



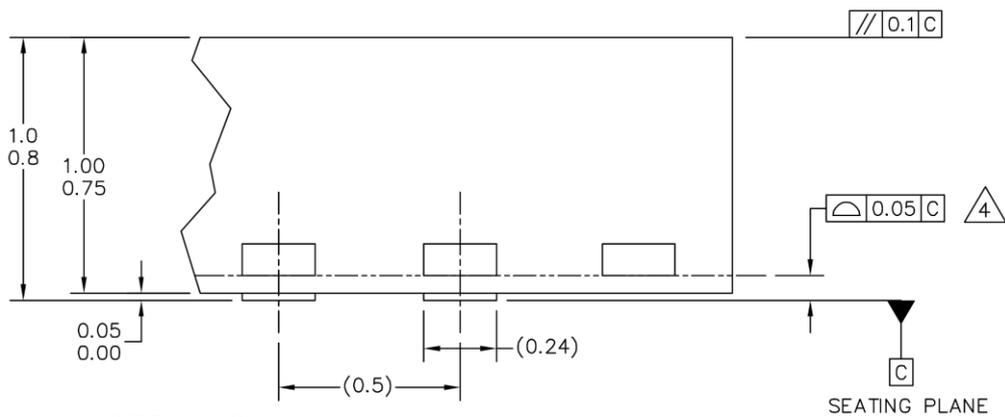
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED QUAD FLAT NON-LEADED PACKAGE (QFN) 48 TERMINAL, 0.5 PITCH (7 X 7 X 1)	DOCUMENT NO: 98ARH99048A	REV: F
	CASE NUMBER: 1314-05	05 DEC 2005
	STANDARD: JEDEC-MO-220 VKKD-2	



DETAIL N
PREFERRED CORNER CONFIGURATION

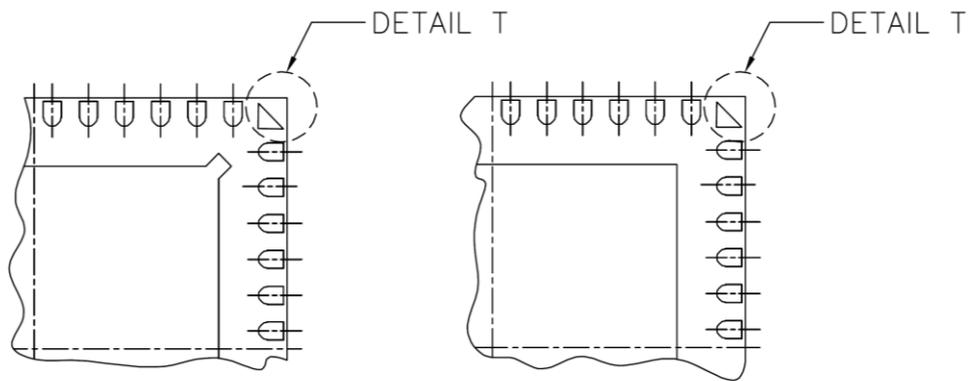


DETAIL M
PREFERRED PIN 1 BACKSIDE IDENTIFIER



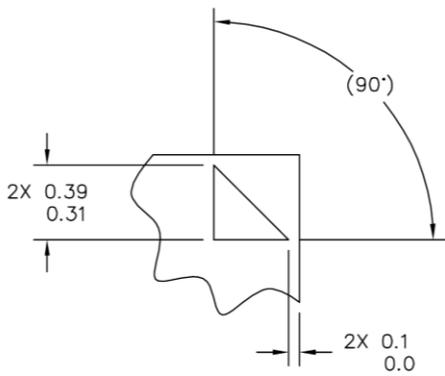
DETAIL G
VIEW ROTATED 90° CW

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED QUAD FLAT NON-LEADED PACKAGE (QFN) 48 TERMINAL, 0.5 PITCH (7 X 7 X 1)	DOCUMENT NO: 98ARH99048A	REV: F	
	CASE NUMBER: 1314-05	05 DEC 2005	
	STANDARD: JEDEC-MO-220 VKKD-2		



DETAIL M
PIN 1 BACKSIDE IDENTIFIER OPTION

DETAIL M
PIN 1 BACKSIDE IDENTIFIER OPTION



DETAIL T

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED QUAD FLAT NON-LEADED PACKAGE (QFN) 48 TERMINAL, 0.5 PITCH (7 X 7 X 1)	DOCUMENT NO: 98ARH99048A	REV: F	
	CASE NUMBER: 1314-05	05 DEC 2005	
	STANDARD: JEDEC-MO-220 VKKD-2		

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. THE COMPLETE JEDEC DESIGNATOR FOR THIS PACKAGE IS: HF-PQFN.
4.  COPLANARITY APPLIES TO LEADS, CORNER LEADS, AND DIE ATTACH PAD.
5. MIN METAL GAP SHOULD BE 0.2MM.