

MC9S08QD4

参考手册

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

MC9S08QD4 系列特点

8 位 HCS08 中央处理单元（CPU）

- 16-MHz HCS08CPU(中央处理单元)
- HC08 指令集，增加了 BGND 指令
- 背景调试系统
- 在线调试（in-circuit debugging）期间可以设置单个断点（在片上调试模式时可以在增加两个断点）
- 支持高达 32 个中断/复位源

存储器选项

- 闪存读/写/擦除全部操作电压和温度
- 闪存容量
 - MC9S08QD4/S9S08QD4: 4096 字节
 - MC9S08QD2/S9S08QD2: 2048 字节

RAM 容量

- MC9S08QD4/S9S08QD4: 256 字节
- MC9S08QD2/S9S08QD2: 128 字节

省电模式

- 增加了三种停止模式

时钟源选项

- ICS—带锁频环（FLL）的内部时钟源模块（ICS）由内部控制。精密微调的内部参考允许温度和电压 0.2% 决议和 2% 偏差。

系统保护

- 可选的系统正常操作（COP）复位，使用独立的 32KHz 内部时钟源或总线时钟运行。
- 低压检测复位或中断
- 非法操作符检测复位
- 非法地址检测复位
- FLASH 块保护

外部设备

- ADC—4 通道，10 位 AD 转换器，包含自动比较功能，异步时钟源，温度传感器和内部能带隙参考通道。ADC 是使用 RTI 计数触发的硬件。
- TIM1—2 通道定时器/脉宽调制，每个通道可被用作输入捕捉，输出比较，带缓冲的边对齐 PWM 或带缓冲的中心对齐 PWM
- TIM2—1 通道定时器/脉宽调制，每个通道可被用作输入捕捉，输出比较，带缓冲的边对齐 PWM 或带缓冲的中心对齐 PWM
- KBI—4 引脚键盘中断模块，带软件选择边沿或边沿/电平模式上的极性

输入/输出

- 4 个通用输入/输出 (I/O) 引脚，一个只输入引脚和一个只输出引脚。每个引脚输出电流 10mA，封装允许最大电流值 60mA。
- 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速率控制
- RESET, IRQ, BKGD/MS 引脚在内部上拉以减小系统消耗

开发支持

- 单线背景调试接口

封装选择

- 8 引脚 SOIC
- 8 引脚 PDIP (仅用于 MC9S08QD4 和 MC9S08QD2)
- 所有封装选项适应 RoHS

MC9S08QD4系列数据手册

——涵盖MC9S08QD4、
MC9S08QD2、
S9S08QD4和
S9S08QD2等型号

翻译自MC9S08QD4 Series MCU Data Sheet, Rev. 3（2007年11月）

苏州大学Freescale MCU研发中心

2008年5月18日

目 录

第一章 芯片概述	10
1.1 简介	10
1.2 MC9S08QD4 系列芯片	10
1.2.1 MCU 框图	11
1.3 系统时钟分配	12
第二章 外部信号描述	13
1.1 芯片引脚分配	13
2.2 推荐的系统连接	13
2.2.1 电源	14
2.2.2 振荡器	14
2.2.3 复位（仅输入）	15
2.2.4 背景/模式选择（BKGD/MS）	15
2.2.5 通用 I/O 及外设端口	15
第三章 操作模式	17
3.1 简介	17
3.2 特征	17
3.3 运行模式	17
3.4 背景模式	17
3.5 等待模式	18
3.6 停止模式	18
3.6.1 停止模式 2	18
3.6.2 停止模式 3	19
3.6.3 停止模式下激活 BDM 使能	19
3.6.4 停止模式下 LVD 使能	20
3.6.5 停止模式下的片上外设模块	20
第四章 内存映像和寄存器定义	21
4.1 MC9S08QD4 系列内存映像	21
4.2 复位和中断向量分配	22
4.3 寄存器地址和位分配	22
4.4 RAM	26
4.5 Flash	27
4.5.1 特征	27
4.5.2 写入和擦除时间	27
4.5.3 擦除和写入命令执行	28
4.5.4 突发写入时间	29
4.5.5 访问错误	31
4.5.6 FLASH 块保护	31
4.5.7 向量重定向	32
4.6 安全性	32
4.7 FLASH 寄存器和控制位	33
4.7.1 FLASH 时钟分频寄存器(FCDIV)	33

4.7.2 FLASH 选项寄存器(FOPT 和 NVOPT)	34
4.7.3 FLASH 配置寄存器(FCNFG)	34
4.7.4 FLASH 保护寄存器(FPROT 和 NVPROT)	35
4.7.5 FLASH 状态寄存器(FSTAT)	35
4.7.6 FLASH 命令寄存器(FCMD)	37
第五章 复位, 中断和系统控制	39
5.1 简介	39
5.2 特征	39
5.3 MCU 复位	39
5.4 计算机正常操作监控模块(COP)看门狗	39
5.5 中断	40
5.5.1 中断堆栈结构	41
5.5.2 外部中断请求引脚 (IRQ)	41
5.5.3 中断向量、中断源和局部屏蔽	42
5.6 低电压检测系统 (LVD)	43
5.6.1 上电复位操作	43
5.6.2 LVD 复位操作	43
5.6.3 LVD 中断操作	43
5.6.4 低电压警告(LVW)	43
5.7 实时中断(RTI)	43
5.8 复位、中断、系统控制寄存器和控制位	44
5.8.1 中断引脚请求状态和控制寄存器 (IRQSC)	44
5.8.2 系统复位状态寄存器 (SRS)	45
5.8.3 系统背景调试强制复位寄存器 (SBDFR)	46
5.8.4 系统选项寄存器 1 (SOPT1)	46
5.8.5 系统选项寄存器 2 (SOPT2)	47
5.8.6 系统设备识别寄存器(SDIDH、SDIDL)	47
5.8.7 系统实时中断状态和控制寄存器 (SRTISC)	48
5.8.8 系统电源管理状态和控制寄存器 1 (SPPMSC1)	48
5.8.9 系统电源管理状态和控制寄存器 2 (SPPMSC2)	49
第六章 并行输入/输出控制	51
6.1 端口数据和数据方向	51
6.2 引脚控制 - 上拉, 回转速率和驱动强度	52
6.3 停止模式下的引脚行为	52
6.4 并行 I/O 寄存器	52
6.4.1A 口寄存器	52
6.4.2A 口控制寄存器	53
第七章 中央处理单元 (S08CPUV2)	55
7.1 简介	55
7.1.1 特征	55
7.2 编程模型和 CPU 寄存器	55
7.2.1 累加器 A (accumulator)	56
7.2.2 变址寄存器 HX (index pointer)	56
7.2.3 堆栈指针 SP (Stack Pointer)	56

7.2.4 程序计数器 PC (program counter)	57
7.2.5 条件码寄存器 CCR (condition code register)	57
7.3 寻址方式	58
7.3.1 内在寻址方式 (INH)	58
7.3.2 相对寻址方式 (REL)	58
7.3.3 立即寻址方式 (IMM)	58
7.3.4 直接寻址方式 (DIR)	58
7.3.5 扩展寻址方式 (EXT)	58
7.3.6 变址寻址方式	58
7.4 特殊操作	59
7.4.1 复位序列	59
7.4.2 中断序列	59
7.4.3 等待模式操作	60
7.4.4 停止模式操作	60
7.4.5 BGND 指令	60
7.5 HCS08 指令集	61
第八章 AD 转换器(ADC0V1)	71
8.1 概述	71
8.1.1 模块配置	71
8.1.2 特征	73
8.1.3 框图	73
8.2 外部信号描述	74
8.2.1 模拟电源 (V_{DDAD})	74
8.2.2 模拟地 (V_{SSAD})	74
8.2.3 参考高电压 (V_{REFH})	75
8.2.4 参考低电压 (V_{REFL})	75
8.2.5 模拟通道输入 (ADx)	75
8.3 寄存器定义	75
8.3.1 状态和控制寄存器 1 (ADCSC1)	75
8.3.2 状态和控制寄存器 2 (ADCSC2)	76
8.3.3 数据高结果寄存器 (ADCRH)	77
8.3.4 数据低结果寄存器 (ADCRL)	77
8.3.5 比较值高寄存器 (ADCCVH)	77
8.3.6 比较值低寄存器 (ADCCVL)	78
8.3.7 配置寄存器 (ADCCFG)	78
8.3.8 引脚控制 1 寄存器 (APCTL1)	79
8.3.9 引脚控制 2 寄存器 (APCTL2)	80
8.3.10 引脚控制 3 寄存器 (APCTL3)	80
8.4 功能描述	81
8.4.1 时钟选择和分频控制	81
8.4.2 输入选择和引脚控制	82
8.4.3 硬件触发	82
8.4.4 转换控制	82
8.4.5 自动比较功能	84

8.4.6 MCU 等待模式操作	84
8.4.7 MCU stop3 模式操作	84
8.4.8 MCU stop1 和 stop2 模式操作	84
8.5 初始化信息	85
8.5.1 ADC 模块初始化举例	85
8.6 应用信息	86
8.6.1 外部引脚和安排	86
8.6.2 错误源	87
第九章 内部时钟源 (S08ICSV1)	91
9.1 简介	91
9.1.1 ICS 配置信息	91
9.1.2 特征	92
9.1.3 操作模式	93
9.1.4 框图	93
9.2 外部信号描述	94
9.3 寄存器定义	94
9.3.1 ICS 控制寄存器 1 (ICSC1)	94
9.3.2 ICS 控制寄存器 2 (ICSC2)	95
9.3.3 ICS 调整寄存器 (ICSTRM)	96
9.3.4 ICS 状态和控制寄存器 (ICSSC)	96
9.4 功能描述	97
9.4.1 操作模式	97
9.4.2 模式转换	98
9.4.3 总线频率分频因子	99
9.4.4 模式转换	99
9.4.5 内部参考时钟	99
9.4.6 可选的外部参考时钟	99
9.4.7 固定频率时钟	99
9.5 模块初始化	100
9.5.1 ICS 模块初始化顺序	100
第十章 键盘中断 (S08KBIV2)	103
10.1 简介	103
10.1.1 特征	104
10.1.2 模式选择	104
10.1.3 KBI 框图	104
10.2 外部信号描述	104
10.3 寄存器定义	105
10.3.1 KBI 状态和控制寄存器 (KBISC)	105
10.3.2 KBI 引脚使能寄存器 (KBIPE)	105
10.3.3 KBI 边沿选择寄存器 (KBIES)	106
10.4 功能描述	106
10.4.1 仅边沿触发	106
10.4.2 边沿和电平触发	106
10.4.3 KBI 上拉/下拉电阻	107

10.4.4 KBI 初始化.....	107
第十一章 定时器/脉宽调制(S08TPMV2).....	109
11.1 简介.....	109
11.1.1 TPM2 配置信息.....	109
11.1.2 TCLK1 和 TCLK2 配置信息.....	109
11.1.3 特征.....	110
11.1.4 框图.....	111
11.2 外部信号描述.....	112
11.2.1 外部 TPM 时钟源.....	112
11.2.2 TPMxCHn——TPMx 通道 n I/O 引脚.....	112
11.3 寄存器定义.....	112
11.3.1 定时器状态和控制寄存器 (TPMxSC)	112
11.3.2 定时器计数寄存器 (TPMxCNTH:TPMxCNTL)	113
11.3.3 定时器计数预置寄存器 (TPMxMODH:TPMxMODL)	114
11.3.4 定时器通道 n 状态和控制寄存器 (TPMxCnSC)	114
11.3.5 定时器通道值寄存器 (TPMxCnVH:TPMxCnVL)	115
11.4 功能描述.....	116
11.4.1 计数器.....	116
11.4.2 通道模式选择.....	117
11.4.3 中心对齐 PWM 模式	118
11.5 TPM 中断	119
11.5.1 清零定时器中断标志.....	119
11.5.2 定时器翻转中断描述.....	119
11.5.3 通道事件中断描述.....	120
11.5.4 PWM 占空比结束时间	120
第十二章 开发支持	121
12.1 介绍	121
12.1.1 强制背景调试.....	121
12.1.2 模块配置.....	121
12.1.3 特点	121
12.2 背景调试控制器(BDC).....	121
12.2.1 BKGD 引脚描述	122
12.2.2 通信细节.....	123
12.2.3 BDC 命令	124
12.2.4 BDC 硬件断点	126
12.3 寄存器定义	126
12.3.1 BDC 寄存器和控制位	126
12.3.2 系统背景调试强制复位寄存器 (SBDFR)	128
附录 A 电气特性.....	129
A.1 简介.....	129
A.2 最大绝对额定值.....	129
A.3 热特性.....	129
A.4 ESD 保护和闭锁抗扰度	130
A.5 DC 特性	131

A.6 电源电流特性.....	136
A.7 内部时钟源特性.....	138
A.8 AC 特性	139
A.8.1 控制时许.....	139
A.9ADC 特性	141
A.10 FLASH 规格说明	142
附录 B 订购须知和机械制图	144
B.1 订购须知.....	144
B.1.1 设备编号机制.....	144
B.2 机械制图.....	144

第一章 芯片概述

1.1简介

MC9S08QD4 系列 MCU 是 8 位低耗，高效的微控制器。该家族所有的 MCU 使用增强的 HCS08 内核，包括多种模块，存储器容量，存储器类型，和封装类型。

1.2MC9S08QD4系列芯片

该系列含有：

- MC9S08QD4
- MC9S08QD2
- S9S08QD4
- S9S08QD2

注：

MC9S08QD4 和 MC9S08QD2 芯片主要应用消费品和工业控制。

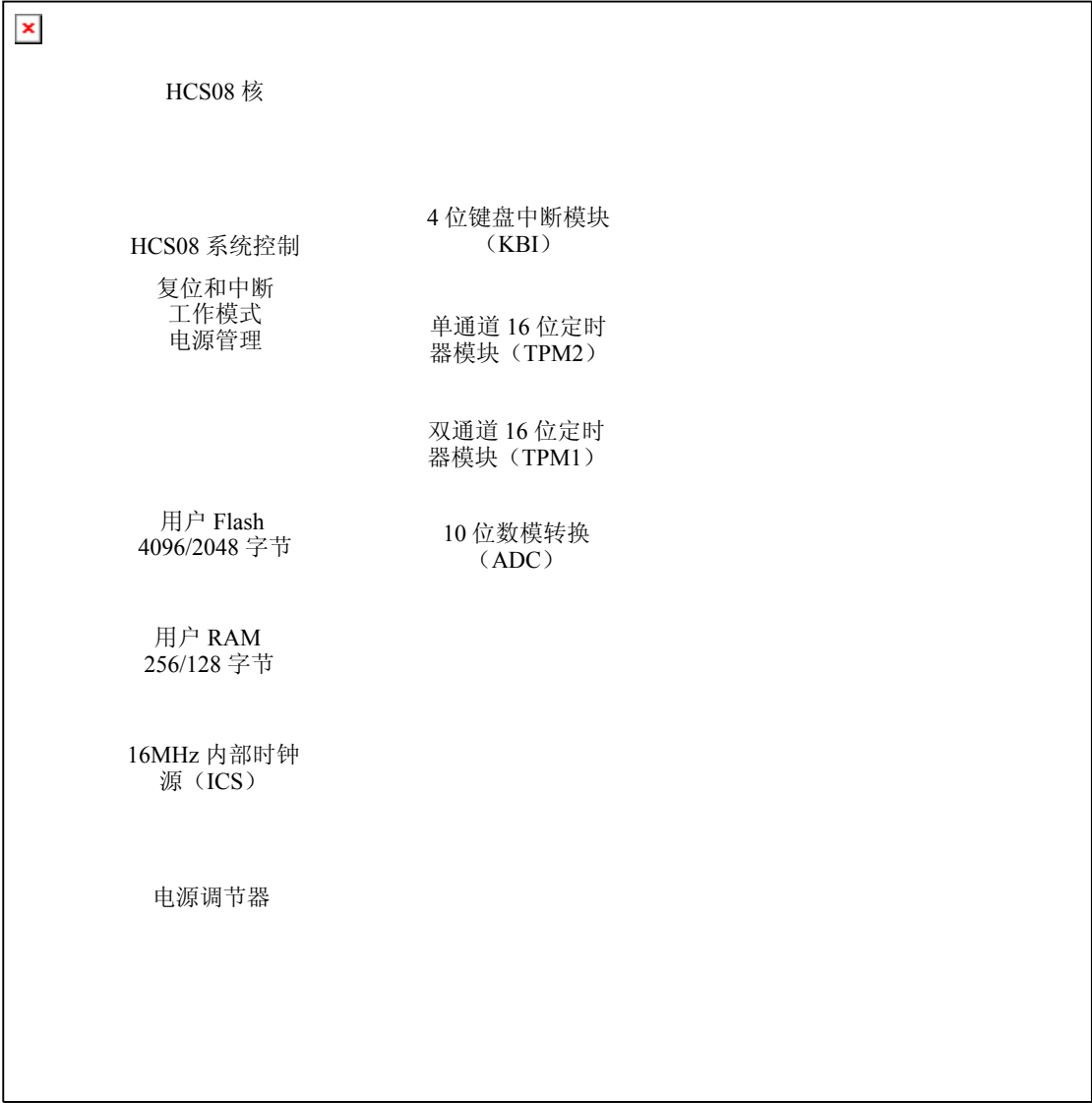
S9S08QD4 和 S9S08QD2 主要应用于汽车产品。

表 1-1 罗列了这些 MCU 的特征

表 1-1MCU 特征和封装

消费和工业控制芯片		
特征	MC9S08QD4	MC9S08QD2
Flash	4KB	2KB
RAM	256B	128B
ADC	4通道，10位	
总线速度	5V时8MHz	
工作电压	2.7V—5.5V	
16位定时器	单通道，双通道	
GPIO	4个I/O脚；一个输入，一个输出	
LVI	是	
封装类型	8脚PDIP；8脚NB SOIC	
是否满足消费和工业控制需求	是	是
是否满足汽车产品需求	否	否
汽车产品芯片		
特征	S9S08QD4	S9S08QD2
Flash	2KB	2KB
RAM	256B	128B
ADC	4通道，10位	
总线速度	5V时8MHz	
工作电压	2.7V—5.5V	
16位定时器	单通道，双通道	
GPIO	4个I/O脚；一个输入，一个输出	
LVI	是	
封装类型	8脚NB SOIC	
是否满足消费和工业控制需求	否	否
是否满足汽车产品需求	是	是

1.2.1MCU框图



- 注:
1. 端口引脚作为输入时可以通过软件设置选择内部上拉设备。
 2. 端口引脚可通过软件设置为输出驱动增强。
 3. 端口引脚可通过软件设置为输出回速率率控制。
 4. 如果PTA5作为IRQ引脚使用（IRQPE=1）IRQ包含一个软件配置的上拉下拉设备。
 5. 如果PTA5作为复位引脚使用（RSTPE=1）RESET包含内部上拉设备。
 6. PTA5没有通过钳位二极管连向V_{DD}。IRQ不能加载高于V_{DD}的电平。当内部上拉有效时该引脚电压可测约在V_{DD}-0.7V。内部连接到该引脚上拉接V_{DD}。
 7. 如果BKGD有效（BKGDPE=1）PTA4包含内部上拉设备。
 8. 当引脚作为KBI引脚（KBIPEn=1）且相关引脚允许上拉时，KBEDGn可以把上拉重新配置为下拉。

图1-1 MC9S08QD4系列框图

表 1-2 提供了片上模块的功能版本。

表 1-2 片上模块的版本

模块	版本
模数转换器（ADC）	1
中央处理单元（CPU）	2
内部时钟源（ICS）	1
键盘中断（KBI）	2
定时器脉宽调制模块（TPM）	2

1.3系统时钟分配

图 1-2 显示了一个简单的时钟连接图。MCU 的某些模块包含选择时钟输入。该模块的时钟输入表明这些时钟是被用来驱动模块功能的。所有与模块相关的存储器映像寄存器使用 BUSCLK 时钟。

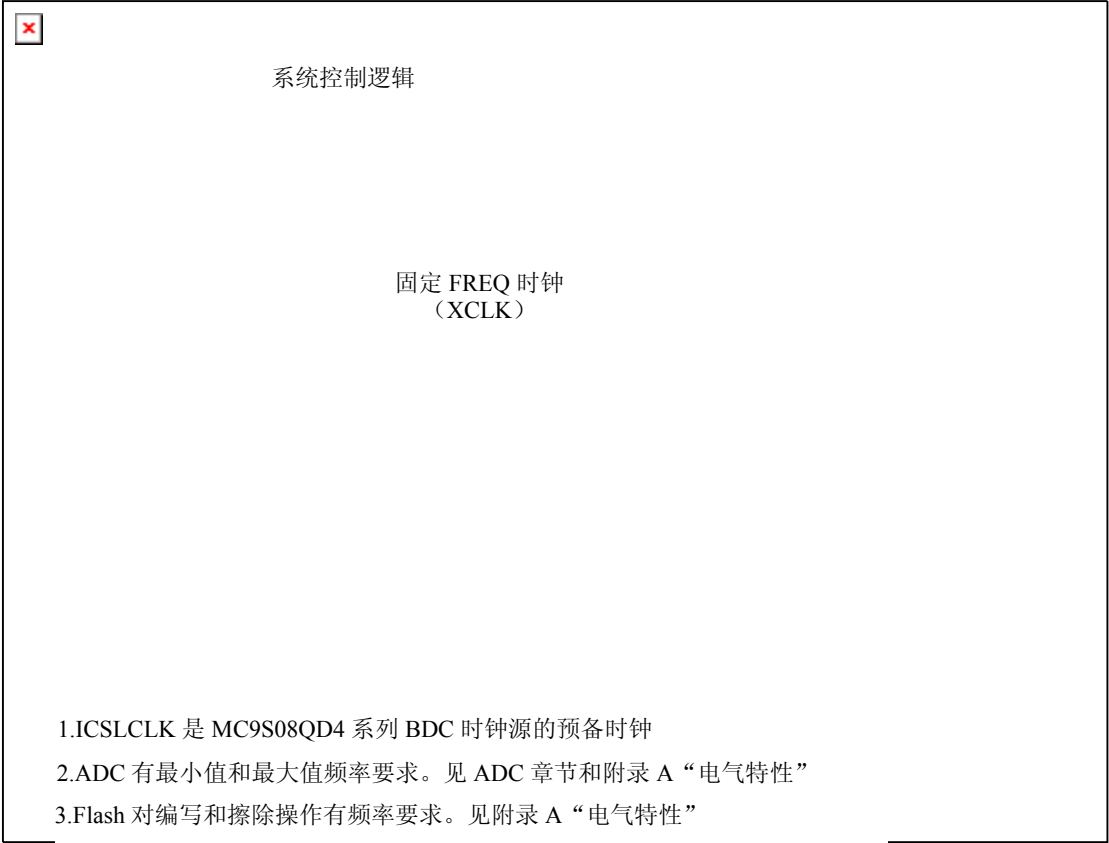


图1-2系统时钟分配框图

第二章 外部信号描述

这章节主要描述连接到封装引脚上的信号。包括引脚输出图，信号特性表和详细的信号功能讨论。

1.1 芯片引脚分配

图 2-1 显示了 8 脚封装的引脚分配。

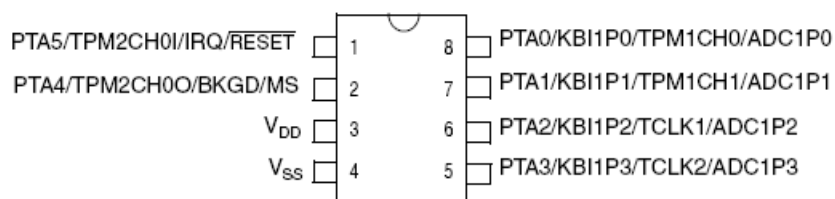


图2-1 8脚封装

2.2 推荐的系统连接

图 2-2 显示了所有 MC9S08QD4 系列应用系统的一般引脚连接。

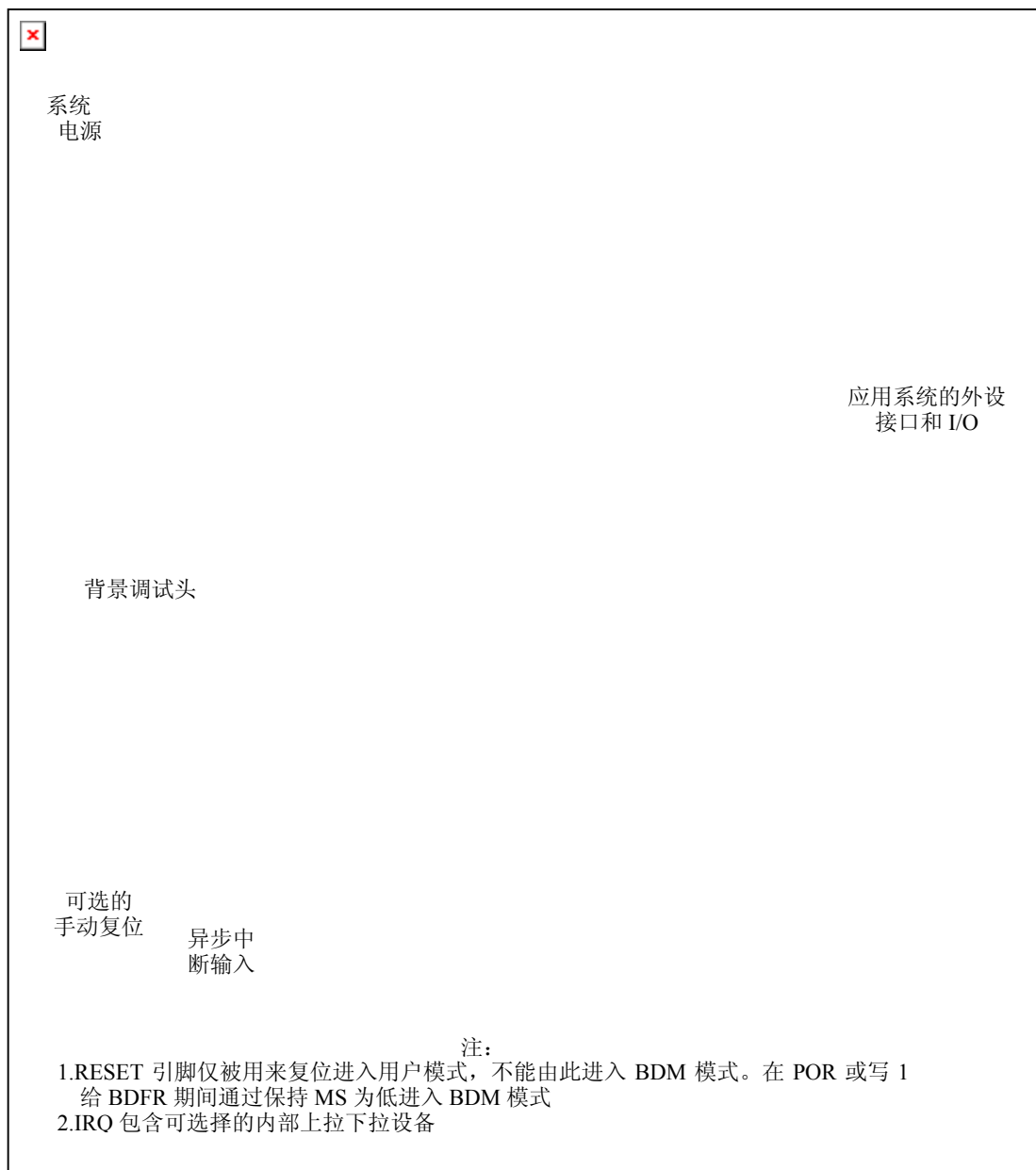


图2-2基本系统连接

2.2.1 电源

V_{DD} 和 V_{SS} 是 MCU 主要电源供应引脚。该电源给所有 I/O 缓存电路，ADC 模块，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。

典型的，应用系统含有两个独立的电容和电源引脚相连：一个大容量电解电容器，如 10 μ F 的钽电容，用来为全体系统提供大电量存储，和一个 0.1 μ F 的陶瓷旁路电容，尽可能的靠近 MCU 电源引脚，用以抑制高频噪声。

2.2.2 振荡器

出复位的 MCU 使用一个内部发生时钟由内部时钟源模块提供 (ICS)。内部频率通常是 16MHz 缺省的 ICS 设置将提供一个 4MHz 总线出复位频率。更多 ICS 信息见内部时钟源章节。

2.2.3 复位（仅输入）

在一个电源复位（POR）进入用户模式后，PTA5/TPM2CH0I/IRQ/ $\overline{\text{RESET}}$ 引脚缺省为一个通用输入端口引脚，PTA5。设置停止模式 1 中的 RSTPE 配置该引脚为 $\overline{\text{RESET}}$ 输入引脚。一旦配置为 $\overline{\text{RESET}}$ 引脚，该引脚将保持直到下个电源复位。当该引脚被拉底时可通过它从一个外部时钟源复位 MCU。当 $\overline{\text{RESET}}$ 有效时（RSTPE=1），一个内部上拉设备自动有效。

在一个电源复位后进入背景调试模式，PTA5/TPM2CH0I/IRQ/ $\overline{\text{RESET}}$ 引脚缺省为 $\overline{\text{RESET}}$ 引脚。

当 TPM2 被配置为输入捕捉模式时，该引脚为输入捕捉引脚 TPM2CH0I。

注：

该引脚不包含一个钳位电阻连接到 V_{DD} 且不能加高于 V_{DD} 的电压。

内部上拉 $\overline{\text{RESET}}$ 引脚可测电压范围为 $V_{DD} - 0.7V$ 。内部连接到该引脚的通过上拉下拉连接到 V_{DD}

2.2.4 背景/模式选择（BKGD/MS）

在电源复位（POR）或背景调试强制复位期间（更详细信息见 5.8.3，“系统背景调试强制复位寄存器（SBDPR）”），PTA4/TPM2CH0O/BKGD/MS 引脚将作为模式选择引脚。在复位后该引脚立即变为北京引脚被用于背景调试通信。当 BKGD/MS 引脚有效时（BKGDPE=1），一个内部上拉设备自动有效。

当停止模式 1 中的 BKGDPE 被置位时该引脚背景调试通信功能有效。BKGDPE 在任意 MCU 复位后被置位时必须清零以使用 PTA4/TPM2CH0O/BKGD/MS 引脚的选择引脚功能。

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 6 引脚的标准背景调试接口，它可以在复位上升沿时保持 BKGD/MS 为低，强制 MCU 进入背景模式。

BKGD 引脚主要被背景调试控制器(BDC)使用，它使用约定好的协议进行通信，该协议使用目标 MCU 的 16 个 BDC 时钟周期来传送每一位。目标 MCU 的 BDC 时钟频率可以达到最大的总线时钟频率，因此不要将任何大的电容和 BKGD/MS 引脚相连，那会干扰背景串行通信。

虽然 BKGD 引脚是一个伪开漏引脚，但是背景调试通信协议提供了简短的，主动驱动，高加速脉冲以确保快速上升时间。电缆上的小电容和内部上拉设备的绝对值对判断 BKGD 引脚上上升和下降时间几乎不起任何作用。

2.2.5 通用 I/O 及外设端口

MC9S08QD4 系列上支持 4 个通用引脚一个只输入引脚和一个只输出引脚和片上外设功能模块所共用（比如定时器和串行 I/O 系统，键盘中断等）。MC9S08QD4 系列的每一个芯片上都含有一个只输入引脚和一个只输出引脚。

当一个端口引脚被配置成一个通用输出引脚或一个外设使用这个引脚作为输出时，软件可选择两个驱动增强中的一个和允许或禁止回转速率控制。当一个端口引脚被配置成一个通用输入或一个外设使用这个引脚作为输入时软件能使能一个上拉设备。

关于控制这些引脚作为通用 I/O 引脚的更多信息，见第六章“并行输入/输出控制”。关于如何及何时片上外设系统使用这些引脚见表 2-1 中的相关内容。

紧随在复位后，所有不是只输出的引脚被配置为高阻态，带内部上拉设备禁止的通用输入引脚。在复位后这些只输出端口功能被禁止但被配置为带回转速率控制允许的低输出驱动增强。在任何复位时 PTA4 引脚缺省为 BKGD/MS。

注：

为了避免来自浮动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允许片上上拉设备或改变不常用引脚的方向置为输出因此使该引脚将不再浮动。

2.2.5.1 引脚控制寄存器

为选择驱动增强或使能回转速率控制或上拉设备，用户需写相关引脚控制寄存器，该寄存器在存储器映像模块的高地址页寄存器中。该引脚控制寄存器操作独立于并行 I/O 寄存器并允许控制一个单独引脚端口。

2.2.5.1.1 内部上拉使能

通过设置一个上拉使能寄存器 (PTxPEN) 中的相应位，一个内部上拉设备可来使能每个端口引脚。如果该引脚通过并行 I/O 控制逻辑或任何复用的外设功能配置为输出引脚，不管相应上拉使能寄存器位的状态如何上拉设备禁止。如果该引脚被一个模拟功能控制上拉设备也同样禁止。

当上升沿检测有效时 KBI 模块和 IRQ 功能将导致一个内部上拉设备变为下拉。

2.2.5.2 输出回转速率控制

通过设置一个回转速率控制寄存器 (PTxSEn) 中的相应位来，回转速率控制可使能每个端口引脚。当使能时，回转控制限制速率以使一个输出按一定顺序传送降低 EMC 影响。回转速率控制对配置为输入的引脚无影响。

2.2.5.3 输出驱动增强选择

通过设置驱动增强选择寄存器 (PTxDSn) 中的相应位一个输出引脚能被选择带有高输出驱动增强。当高驱动被选择时，一个引脚能够采集承受更大电流。即使每个 I/O 引脚能被选择为高驱动，用户必须确保用于芯片的所有电源和下降限制不会超出范围。驱动增强选择更倾向于影响 I/O 引脚上直流操作。但交流受影响。高驱动允许一个引脚驱动一个更大的负载，该负载与低驱动使能引脚具有相同的转换速度，使该负载变为一个功耗小的负载。

由此而来，通过使能引脚为高驱动 EMC 排放可被影响。

表 2-1 引脚复用优先权

端口引脚	最低级〈—引脚功能优先级—〉最高级			参考 ¹
	可选用功能	可选用功能	可选用功能	
PTA0	KBI1P0	TPM2CH0O	ADC1P0 ³	KBI1、ADC1和TPM1章节
PTA1	KBI1P1	TPM2CH0I	ADC1P1 ³	KBI1、ADC1和TPM1章节
PTA2	KBI1P2	TCLK1	ADC1P2 ³	KBI1、ADC1和TPM1章节
PTA3	KBI1P3	TCLK2	ADC1P3 ³	KBI1、ADC1和TPM1章节
PTA4	TPM2CH0O	BKGD/MS		TPM2章节
PTA5 ²	TPM2CH0I	IRQ	RESET	IRQ ⁴ 和TPM2章节

¹ 见模块部分列出的基于模块复用这些引脚的信息。

² 引脚不需通过一个钳位电阻连接到V_{DD}不能被加高于V_{DD}电压。当内部上拉有效时该引脚电压可测约在V_{DD}-0.7V。内部连接到该引脚上拉接V_{DD}。

³ 如果这些模拟模块都有效那么都能访问该引脚。

⁴ 见 5.8 节，“复位，中断和系统控制寄存器和控制位”关于配置 IRQ 模块信息。

第三章 操作模式

3.1 简介

本章将描述 MC9S08QD4 系列的操作模式。进入每种模式，退出每种模式和在每种模式中的功能性描述。

3.2 特征

- 用于代码调试的背景调试模式
- 等待模式：
 - CPU 关闭以节省电源
 - 系统时钟运行
 - 满电压调节保持
- 停止模式：
 - CPU 和总线时钟停止
 - 停止模式 2 — 内部部分电路断电，内存内容不丢失
 - 停止模式 3 — 为快速恢复，所有靠内部电路供电

3.3 运行模式

这是 MC9S08QD4 系列一般情况下的操作模式。当 BKGD/MS 在复位上升沿时为高时此模式被选择。在这种模式下，CPU 在复位后从内存 0XFFFE:0XFFFF 地址处开始执行代码。

3.4 背景模式

背景模式功能由 HCS08 核中的背景调试控制器（BDC）管理。在软件开发期间 BDC 提供各种方式来分析 MCU 操作。

进入背景模式的五种方式：

- 在复位上升沿 BKGD/MS 引脚为低时
- 当通过 BKGD 引脚接收到一个 BACKGROUND 命令时
- 当一个 BGND 指令被执行时
- 当进入一个 BDC 断点时

在进入背景调试模式后，CPU 保持暂停状态等待背景命令而不是执行来自用户应用程序的指令。

背景命令的两种类型：

- 非插入命令，定义为可在用户程序运行时发出的指令。当 MCU 在运行模式下时，非插入命令可经由 BKGD 引脚发出。当 MCU 在背景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
 - 存储器访问状态命令
 - BDC 寄存器访问命令

- BACKGROUND 命令
- 背景命令，只有当 MCU 在背景调试模式下才可以执行。背景命令包括以下命令：
 - 读写 CPU 寄存器
 - 单步调试指令
 - 离开背景调试模式，返回用户应用程序(GO)

背景调试模式通常用于在 MCU 第一次在运行模式下运行之前向 FLASH 存储器写入一个引导程序或者用户应用程序。当 MC9S08QD4 从 Freescale 出厂时，FLASH 存储器默认为擦除的除非特别指明，在 FLASH 存储器初次编程之前，由于 FLASH 存储器中没有程序，因此在运行模式下没有程序可以运行。背景调试模式也可用于在 FLASH 存储器已经写入程序后擦除和重新写入程序。

更多有关背景调试模式的信息可见第十二章“开发支持部分”。

3.5等待模式

执行到 WAIT 指令时，即可进入等待模式。在执行 WAIT 指令后，CPU 进入没有时钟的低功耗状态。当 CPU 进入等待模式时，CCR 寄存器里面的 I 位将被清零并允许中断。当有中断请求发生，CPU 就会脱离等待模式，继续正常处理，从相应的中断服务程序开始运行。

当 MCU 在等待模式下时，使用背景调试命令有一些限制，只有 BACKGROUND 命令和存储器访问状态命令可用。虽然存储器访问状态命令不允许访问存储器，但是它会报出错误来说明 MCU 在停止模式或等待模式。BACKGROUND 命令可用于使 MCU 从等待模式唤醒进入背景调试模式。

3.6停止模式

在系统选择寄存器中的 STOPE 位置位时，当执行到一个 STOP 指令就进入了两种停止模式中的一种。在所有的停止模式中，所有的内部时钟被停止。如果 STOPE 位在 CPU 执行 STOP 指令时没有置位，MCU 将不会进入任何停止模式，并强制产生一个非法指令复位。可以通过设置 SPMSC2 里的相应标志位来选择进入哪种停止模式。

HSC08 芯片被设计用于低电压操作（1.8V-3.6V）同时包括停止模式 1。MC9S08QD4 系列不包括停止模式 1。

表 3-1 总结了 MCU 在每种停止模式下的行为。

表 3-1 停止模式下的行为

模式	PPDC	CPU, 数字外设, FLASH	RAM	ICS	ADC1	电压调节器	I/O引脚	RTI
STOP2	1	关	待命	关	禁止	待命	维持状态	可选开
STOP3	0	待命	待命	关 ¹	可选开	待命	维持状态	可选开

¹ICS能在停止模式3下运行，参见ICS寄存器

3.6.1停止模式2

STOP2 模式提供非常低的能保持 RAM 内容和所有 I/O 端口状态的待命功耗。为了进入 STOP2 模式，在执行 STOP 指令前必须将 SPMSC2 的 PDC 位和 PPDC 位置 1。另外 LVD 在停止模式下必须被禁止操作（LVDSE=0 或 LVDE=0）。如果 LVD 在停止模式下使能，那么在执行停止指令后不论 PPDC 的状态如何 MCU 将进入停止模式 3。

进入 STOP2 模式之前，用户可以在 RAM 中保存 I/O 口寄存器的内容，和其它想在退出

STOP2 模式后恢复的寄存器的内容。在离开 STOP2 模式后，这些内容可在闭锁的端口开放之前被用户软件恢复。

当 MCU 处于 STOP2 模式下时，所有由电压调节器供电的内部电路都是关闭的，除了 RAM。电压调节器处于低功耗待命状态，就像 ADC 模块一样。进入 STOP2 模式后，I/O 端口的状态被锁定。这些状态将在 STOP2 模式中被保持，直到在退出 STOP2 模式后 SPMSC2 寄存器中的 PPDACK 标志位被写 1。

通过 MCU 的唤醒端口(RESET 或者 IRQ)或者 RTI 中断可离开 STOP2 模式。不管在进入 STOP2 之前 IRQ 是如何设置的，它在 STOP2 模式下总是低电平输入有效。

注：

虽然 IRQ 引脚是自动配置为低输入有效，但与 IRQ 相关的上拉不是自动使能的。因此如果没有使用一个外部上拉，那么必须通过设置 IRQSC 中的 IRQPE 位使能内部上拉。

当从 STOP2 模式中被唤醒后，MCU 将以类似上电复位的方式启动，但是端口状态仍然保持锁定。CPU 将取复位向量。系统和所有外设将处于它们的默认复位状态，且必须被初始化。

从 STOP2 模式中唤醒之后 SPMSC2 寄存器中的 PPDF 标志位将置位。该标志位也许用于指示用户代码执行 STOP2 恢复程序。PPDF 保持置位并且 I/O 端口状态保持锁定，直到 SPMSC2 寄存器中的 PPDACK 标志位写 1。

为保持设置为通用 I/O 引脚的状态，在向 PPDACK 标志位写入之前，用户必须用存储在 RAM 中的 I/O 端口寄存器中的内容来恢复端口寄存器。如果在写入 PPDACK 之前，没有从 RAM 恢复端口寄存器，当 I/O 端口闭锁打开后，I/O 端口转换成复位状态时，寄存器中的内容将是它们的复位默认状态的值。

对于设置为外设 I/O 口的引脚，在向 PPDACK 标志位写入之前，用户必须重新配置外设模块的接口引脚。如果在写入 PPDACK 之前，外设模块不可用，当 I/O 端口闭锁打开时，端口将被它们的相关端口控制寄存器控制。

3.6.2 停止模式3

在表 3-1 所示的条件下执行一个 STOP 指令就可以进入停止模式 3。所有内部寄存器和逻辑的状态，RAM 内容，I/O 引脚状态被保持。

退出 STOP3 模式的方法有：复位 **RESET**，或者通过实时中断、LVD、ADC、IRQ 或 KBI 引脚。

如果通过 **RESET** 退出 STOP3 模式，MCU 将复位，并从复位向量处执行。如果通过内部中断源退出，MCU 将从相应的中断向量处执行。

3.6.3 停止模式下激活BDM使能

如果 BDCSCR 寄存器里的 ENBDM 标志位被置位，从运行模式进入背景调试模式是可能的。BDCSCR 寄存器将在第十二章开发支持部分介绍。如果当 CPU 执行 STOP 指令时，ENBDM 被置位，这样当 MCU 进入到停止模式后背景调试逻辑的系统时钟仍然在工作，所以背景调试的通信仍然是可以的。另外电压调节器没有进入低功耗待命状态，而是在满负荷工作中。如果用户尝试通过置 ENBDM 位进入 STOP2 模式，MCU 会进入到 STOP3 模式而不是 STOP2 模式。

大部分背景命令在停止模式下不可用。存储器存取状态指令不允许存储器存取，但它们会报告一个错误标志，标志 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背景调试模式。一旦设备进入背景调试模式，所有背

景命令都可用。表 3-2 总结了当 MCU 在进入 STOP 时背景调试模块使能的情况下 MCU 的行为。

表 3-2 STOP 模式下 BDM 使能时 MCU 行为

模式	PPDC	CPU, 数字外设, FLASH	RAM	ICS	ADC1	电压调节器	I/O引脚	RTI
STOP3	0	待命	待命	活跃	可选开	活跃	维持状态	可选开

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。当 CPU 执行到 STOP 指令时, 如果此时已经通过置 LVDE 和 LVDSE 位使得 LVD 有效, 则在停止模式下电压调节器仍然保持工作。当 LVD 可用时(LVDSE=1), 用户尝试进入 STOP2 模式, MCU 会进入到 STOP3 模式而不是 STOP2 模式。表 3-3 总结了 MCU 在启用 LVD 的 STOP 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	ICS	ADC1	电压调节器	I/O引脚	RTI
STOP3	0	待命	待命	关 ¹	可选开	待命	维持状态	可选开

¹ICS能在停止模式3下运行, 参见ICS寄存器

3.6.5 停止模式下的片上外设模块

当 MCU 进入任何停止模式, 内部外设模块的系统时钟会停止。即使在例外情况下 (ENBDM=1), 背景调试逻辑的时钟继续工作, 外设的时钟也被停止以减少功耗。见 3.6.1 STOP2 模式、3.6.2 STOP3 模式以获得停止模式下系统行为的更多信息。

表 3-4 停止模式下的 MCU 行为

外设	模式	
	STOP2	STOP3
CPU	关	待命
RAM	待命	待命
Flash	关	待命
并行端口寄存器	关	待命
ADC1	关	可选择开 ¹
ICS	关	待命
TPM1&TPM2	关	待命
电压调节器	待命	待命
I/O引脚	状态保持	状态保持

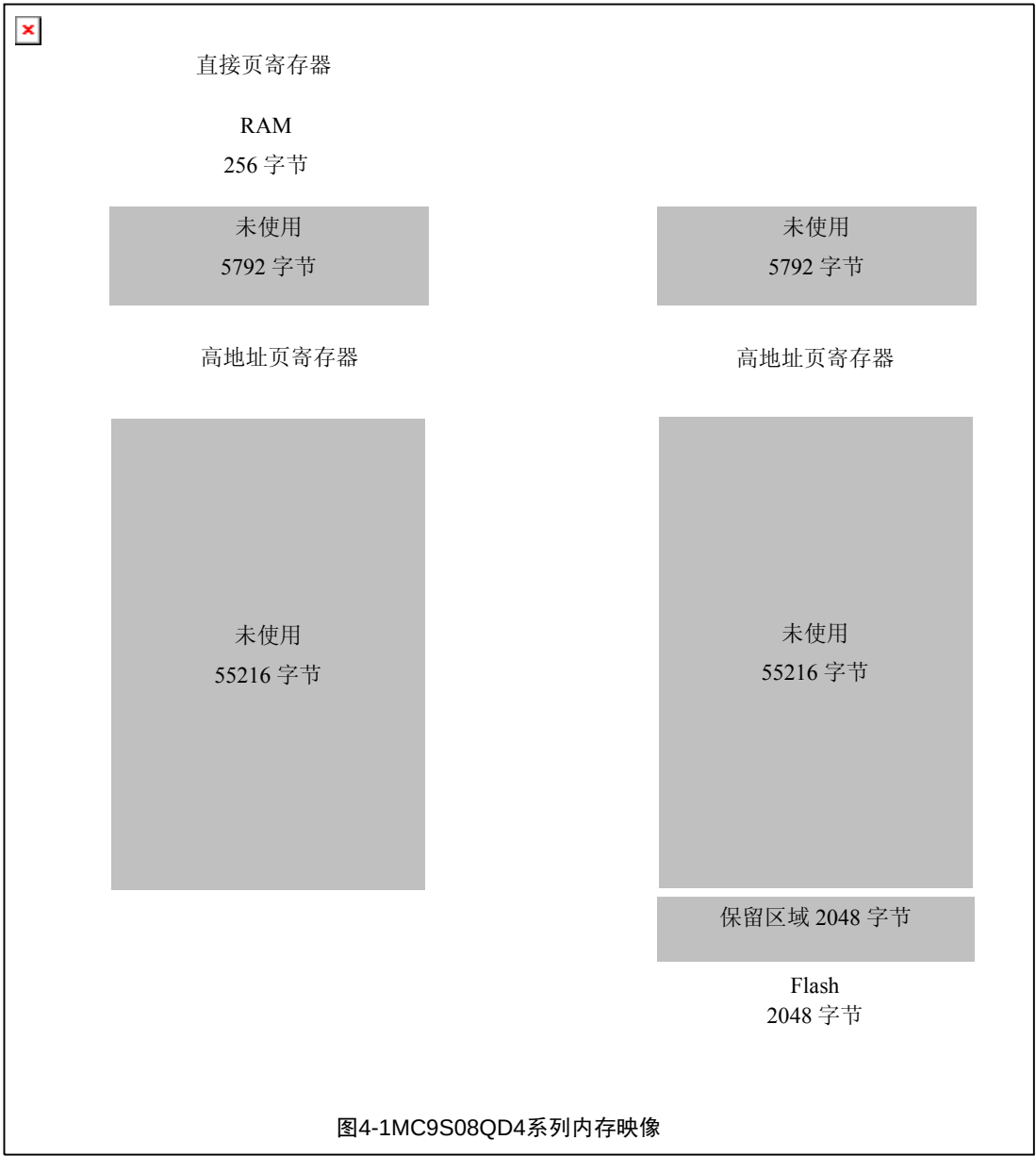
¹要求异步ADC时钟和LVD使能, 其它的待命

第四章 内存映像和寄存器定义

4.1 MC9S08QD4系列内存映像

如图 4-1 所示 MC9S08QD4 系列 MCU 片上存储器由 RAM, 用于非易失性数据存储的 flash 存储器, I/O 和控制/状态寄存器组成。这些寄存器分成一下几组:

- 直接页寄存器 (0X0000-0X005F)
- 高地址页寄存器 (0X1800-0X184F)
- 非易失性寄存器 (0XFFB0-0XFFBF)



4.2 复位和中断向量分配

表 4-1 给出了复位与中断向量的地址分配。表中所示的向量名在由飞思卡尔提供关于 MC9S08QD4 系列 MCU 的头文件中做标签用。

表 4-1 复位和中断向量

地址(高/低)	向量	向量名称
\$FFC0:FFC1	不用的向量空间 (可用于用户程序)	
↑ ↓		
\$FFCA:FFCB		
\$FFD0:FFD1	RTI	Vrti
\$FFD2:FFD3	保留的	
\$FFD4:FFD5	保留的	
\$FFD2:FFD3	保留的	
\$FFD8:FFD9	ADC1转换	Vadc1
\$FFDA:FFDB	键盘中断	Vkeyboard1
\$FFDC:FFDD	保留的	
\$FFDE:FFDF	保留的	
\$FFE0:FFE1	保留的	
\$FFE2:FFE3	保留的	
\$FFE4:FFE5	保留的	
\$FFE6:FFE7	保留的	
\$FFE8:FFE9	保留的	
\$FFEA:FFEB	TPM2溢出	Vtpm2ovf
\$FFEC:FFED	保留的	
\$FFEE:FFEF	TPM2通道0	Vtpm2ch0
\$FFF0:FFF1	TPM1溢出	Vtpm1ovf
\$FFF2:FFF3	TPM1通道1	Vtpm1ch1
\$FFF4:FFF5	TPM1通道0	Vtpm1ch0
\$FFF6:FFF7	保留的	
\$FFF8:FFF9	IRQ	IRQ
\$FFFA:FFFB	低电压检测	Vlvd
\$FFFC:FFFD	SWI	Vswi
\$FFFE:FFFF	RESET	Vreset

4.3 寄存器地址和位分配

MC9S08QD4 系列 MCU 中的寄存器分为以下几组：

- 直接页寄存器位于存储空间前 96 字节处，所以可由高效的直接寻址指令访问。
- 高地址页寄存器较少被使用，所以它们位于存储器的\$1800 地址之后，这样就可以在直接页中留出更多的空间给经常使用的寄存器和变量。
- 非易失性寄存器(\$FFB0-\$FFBF)由 FLASH 存储器中位于\$FFB0 - \$FFBF 地址处的 16 个位置组成。非易失性寄存器的位置包括：
 - NVPROT 和 NVOPT 在复位时定位到工作寄存器

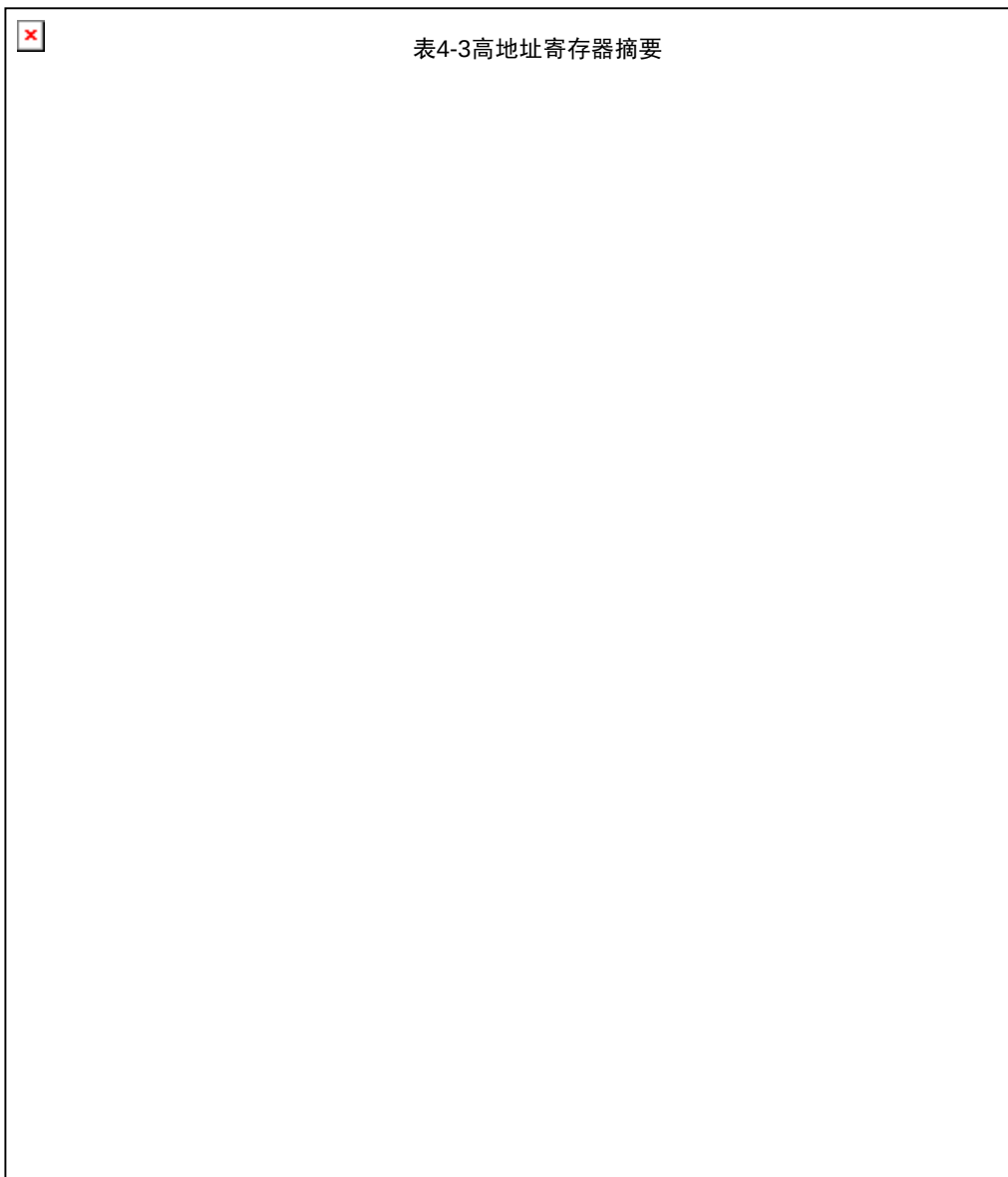
直接页寄存器可以被高效的直接寻址方式指令访问。位操作指令可用于访问直接页寄存器中的任何一位。表 4-2 总结了所有用户可访问的直接页寄存器及其控制位。

✖	表 4-2 直接页寄存器摘要



表4-2直接页寄存器摘要（续）

高地址寄存器，如表 4-3 所示被访问的次数少于其它 I/O 和控制寄存器。因此他们被定位在直接地址内存空间之外在 0C1800 处开始。



¹ 这个保留位必须一直写为 0

非易失性 FLASH 寄存器如表 4-4 中所示被定位在 FLASH 存储器。这些寄存器包括一个 8 字节密钥被用来获取存储器内容。在复位期间，FLASH 存储器的非易失性寄存器区域中的 NVPROT 和 NVOPT 的内容被转移到高地址寄存器中相应的 FPROT 和 FOPT 运作的寄存器来控制安全和保护选项。



表 4-4 非易失性寄存器摘要

--

假设密钥使能位 (KEYEN) 为 1，该 8 字节比较密钥被用来暂时脱离存储器安全保护。该密钥仅通过运行在保护存储器中的用户代码来访问。(一个安全的密钥不能直接通过背景调试命令进入) 该安全密钥可通过置 KEYEN 位为 0 完全禁止。如果该安全密钥被禁止，唯一脱离安全保护的方式是通过整体擦除 FLASH(如果需要的话) (一般情况下通过背景调试接口) 和验证 FLASH 是空白的。为避免在下个复位后返回到安全模式，写该安全位 (SEC01:SEC00) 到非安全状态 (1:0)。

该内部时钟源工厂调整值将被存储在 0XFFAE 和 0XFFAF。开发工具如编辑器能调整该 ICS 和内部温度传感器 (通过 ADC) 和存储该值到 0XFFAD-0XFFAF。

4.4 RAM

MC9S08QD4 包含静态 RAM 位于\$0100 下面可用更有效的直接寻址方式访问，这一区域中的任何单独一位都可以通过位操作指令(BCLR、BSET、BRCLR 和 BRSET)访问。一般将最常用的程序变量放在 RAM 的这一区域。

当 MCU 处于低压等待，STOP2 或 STOP3 模式时，这时 RAM 中的内容是可以保持的。在上电或从 STOP1 唤醒后，RAM 的内容不是未初始化的。若提供的电压不会下降到 RAM 保持的临界电压值以下 (V_{RAM}) 则任何复位都不会影响 RAM 中的数据。

为了和旧的 M68HC05 MCU 兼容，HCS08 初始化时将栈指针定在\$00FF。在 MC9S08QD4 中，一般来说最好将栈指针重新初始化指向 RAM 的顶部，这样 RAM 的直接页可以用于存储经常使用的 RAM 变量以及可以位寻址的程序变量。将以下两个指令加入到初始化程序中。(在 Freescale 提供的文档中，RamLast 等同于 RAM 的最高地址)。

LDHX	#RamLast+1	;指示上一个RAM
TXS		;SP<-(H:X-1)

当安全位使能，RAM 被认为是一个安全可靠的存储器资源是不能通过 BDM 或来自非可靠存储器代码执行访问的。详细信息见 4.6 节“安全性”。

4.5 Flash

FLASH 主要用于程序存储。在线编程允许应用程序在最后一次编译以后下载到 FLASH 存储器中。可以通过单一连线的背景调试接口对整个 FLASH 进行写入。因为对 FLASH 的擦除和写入操作不需要特殊的电压，也可以通过一些其他软件可控制的通信方式对应用程序进行下载。在线编程的详细信息请参见飞思卡尔文件序号为 HCS08RMv1，HCS08 家族参考手册卷 1。

4.5.1 特征

FLASH 有以下特征：

- FLASH 大小
 - MC9S08QD4/S9S08QD4: 4096 字节（8 页，每页 512 字节）
 - MC9S08QD2/S9S08QD2: 2048 字节（4 页，每页 512 字节）
- 单一电压供电的擦除/写入
- 用于快速写入和擦除操作的命令接口
- 在正常的电压和温度下，高达 100000 的写入/擦除次数
- 灵活易用的块保护
- FLASH 和 RAM 的安全机制
- 自动低功耗和低频率读访问

4.5.2 写入和擦除时间

在执行任何一条擦除或者写入命令之前，必须首先设置 FLASH 时钟分频寄存器 (FCDIV)，将 FLASH 模块的内部时钟设到 150kHz 到 200kHz 之间 (f_{FCLK})，(见 4.7.1 节 FLASH 时钟分频寄存器 FCDIV)。这个寄存器只能被写一次，所以一般在复位初始化中设置该寄存器。如果 FSTAT 中的访问错误标志，FACCERR 被置位那么 FCDIV 不能被写。在写 FCDIV 寄存器之前，用户必须确保 FACCERR 没有被置位。设置好的时钟周期被命令处理器用来对擦除和写入脉冲进行计时。这些计时脉冲的整数倍作为时钟被命令处理器用来完成一个擦除或者写入的命令。

表 4-5 显示了写入和擦除时间。总线时钟频率和 FLASH 时钟分频寄存器 (FCDIV) 决定 FLASH 时钟的频率 (f_{FCLK})。FCLK 的一个周期是 $t_{FCLK}=1/f_{FCLK}$ 。时间显示为 FCLK 周期的整数倍且作为绝对时间时 $t_{FCLK}=5\mu s$ 。擦除和写入时间显示包括用于命令状态机制的时间和擦除写入电压的使能。

表4-5擦除和写入时间

参数	FCLK周期	在FCLK=200KHz情况下的时间
字节写入	9	45 μs
字节写入(快速)	4	20 μs (1)
页擦除	4000	20ms
整体擦除	20000	100ms

(1)不包括开始和结束时间

4.5.3 擦除和写入命令执行

任何指令的执行步骤将在下面列出来。在开始指令执行前 FCDIV 寄存器必须完成初始化且任一错误标志被清空。命令执行步骤是：

1. 写一个数据到 FLASH 数组中的一个地址。这个写入的地址和数值的信息将被放在 FLASH 缓冲区中。这个写是任何命令序列必须的第一步。对擦除和空白检测命令来说，这个数据的值是不重要的。但是对于页擦除命令，这个地址必须是要擦除的 512 字节的 FLASH 页中的任意一个地址。对于整体擦除命令和空白检测命令，这个地址可以是 FLASH 存储器中的任意一个地址。512 字节的整页是可擦除的最小的 FLASH 块。

注：

FLASH 中最后一页的整体或页擦除将会擦除工厂已编写好的内部参考时钟调整值。在一次成功擦除操作后不写任何字节到 FLASH 中。一个字节中已经编写好的程序重调位在没首先擦除该页部分或整体字节不允许进入 FLASH 存储器。在没有先擦除的情况下重编写程序可能会干扰 FLASH 存储器中已存储的数据。

2. 写预期的指令代码到 FCMD。五个已经设计好的命令是：空白检查命令(\$05)，写入一个字节命令(\$20)，突发模式写入命令(\$25)，页擦除命令(\$40)，整体擦除命令(\$41)。这些命令将会写入到命令缓冲区。

3. 向 FSTAT 寄存器的 FCBEF 位写 1，可以清 FCBEF 位和登记命令(包括它的地址和数据信息)。

在写存储器之后，写 1 清零 FCBEF 和已完成的命令之前的任何时候可以向 FCBEF 手动写 0，使部分命令序列失效。用这种方式使命令失败可使 FACCERR 存取错误标志位置位，而这位在开始执行新命令之前必须被清零。

必须严格地紧跟着一个监控程序，否则命令不会被接受。这样可以将无意识的改变 FLASH 存储器的可能性降到最低。命令缓冲区空标志(FCBEF)表明命令缓冲区何时空间来接收新的命令。命令完成标志(FCCF)表明何时所有的指令都完成了并且在 FLASH 命令缓冲区中没有新的命令在等待执行。图 4-2 所有指令执行的一个流程图除了突发性写入。在使用任何 FLASH 命令前任何复位后 FCDIV 寄存器必须被初始化。

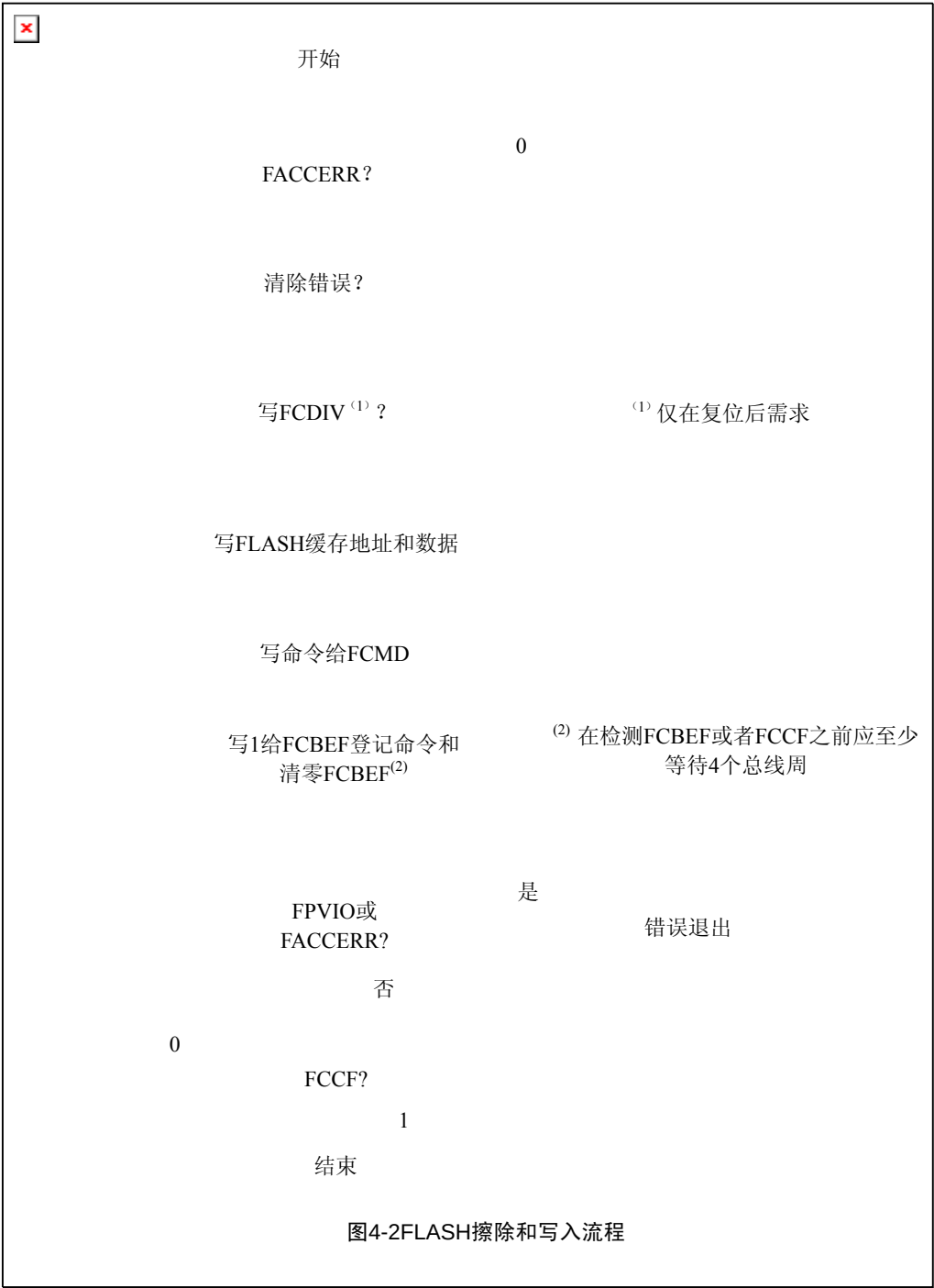


图4-2FLASH擦除和写入流程

4.5.4突发写入时间

突发写入命令用来写入连续数据字节，与标准写入命令相比花费的时间更短。这是可能的因为在程序执行期间当加高电压时 FLASH 阵列不需要被禁止。一般情况下，当一个擦除和写入指令被执行后，与 FLASH 存储器相关的一个内部电荷泵被提供高电压加到 FLASH 阵列上。执行完指令后，电荷泵关闭。当一个突发写入指令被执行后，如果下面两个条件被满足电荷泵使能并保持到该突发写入命令完成后：

- 在当前指令完成之前下个突发写入指令已排列好。
- 下个连续地址选择一个字节，该字节与当前正在写入的字节在同一个物理行。FLASH 存储器的一行由 64 字节组成。可以通过地址 A5-A0 选择一行中的一个字节。当从地

址 A5-A0 都是 0 则开始一个新行。

在突发写入模式下正在写入的一系列字节的首字节将花费与正常模式写入时间相同的时间。如果上面的两个条件被满足则后续的字节的写入时间将按突发写入模式情况写入时间计算。如果下个要写的连续地址是一个新行的开始,那么该字节的写入时间和正常模式下写入时间相同。这是因为加到阵列的高电压被禁止稍后才允许。如果在当前指令完成之前一个新的突发写入命令还未排列好,那么电荷泵将被禁止使用并从阵列上移除高电压。

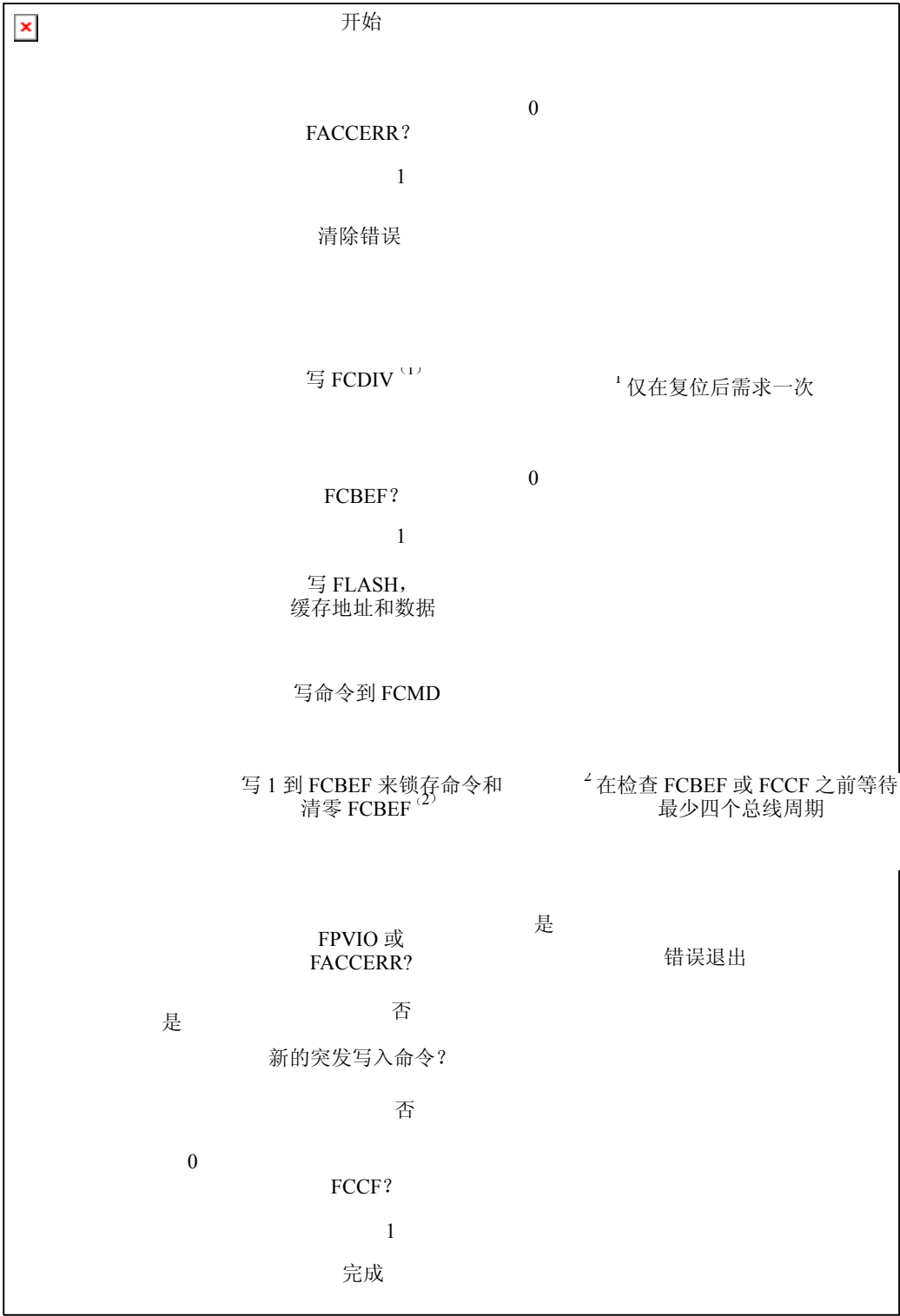


图4-3FLASH突发写入流程图

4.5.5 访问错误

任何时候违反命令执行协议都会产生一个访问错误。下面的任何一个操作都会造成 FSTAT 中的访问错误标志(FACCERR)置位。在执行新的命令之前必须向 FSTAT 中的 FACCERR 写 1 来清 FACCERR 标志。

- 在写一个 FLASH 地址之前，内部 FLASH 时钟频率没有通过 FCDIV 寄存器设置。
- 当 FCBEF 没有置位时，写 FLASH 地址(在命令缓冲区为空之前不可以开始一个新的命令)。
- 在登记先前的命令之前，再一次写入 FLASH 地址(每一个命令只写一次 FLASH)。
- 在登记先前的命令之前，再一次写 FCMD(对每一个命令只写一次 FCMD)。
- 向写入 FLASH 地址后，设置除 FCMD 外的任意其他控制寄存器。
- 向 FCM 写除了五个正确的命令(\$05,\$20,\$25,\$40,或\$41)之外的任何命令。
- 向 FCMD 中写入命令字之后，设置除 FSTAT(清 FCBEF 和命令)之外的任意 FLASH 控制寄存器。
- 当执行写入或者擦除命令时，MCU 进入停止模式。(这个命令是被中止的)
- 当 MCU 处于安全状态时，用背景调试命令写字节写入、突发模式写入和页擦除这些命令(当 MCU 处于安全状态时，背景调试控制器只能进行空白检测和整体擦除命令)。
- 向 FCBEF 位写 0 取消一个未完成的命令。

4.5.6 FLASH块保护

块保护防止对指定地址范围的 FLASH 存储空间进行写入或擦除操作。块保护由 FLASH 保护寄存器(FPROT)控制。当块保护允许时，它开始于 FLASH 末尾地址 0xFFFF 下面的任一个 512 字节。(见 4.7.4 节 FLASH 保护寄存器 FPROT 和 NVPROT)。

复位时，高地址页寄存器(FPROT)装入 NVPROT(FLASH 存储器的非易失寄存器)的内容。FPROT 的值不能被应用软件直接改变，所以一个跑飞的程序不能改变块保护设置。因为 NVPROT 拥有 FLASH 最后的 512 字节，如果任意大的存储器被保护，NVPROT 将自行保护且不能被应用程序软件修改(有意或者无意)。用背景调试命令来写 FPROT 控制位可以擦除一个被保护的 FLASH 块。

块保护机制可见图 4-4 图示。FPS 位作为未保护存储器末尾地址的最高位。该地址通过连接 FPS7:FPS1 用逻辑 1 作显示形成。例如，为了保护存储器的后 8192 存储器字节(0XE0000-0XFFFF)，该 FPS 位必须被设置为 1101111，它可产生一个值为 0XDFFF 作为未保护区域的末位地址。另外为了写 FPS 位以设置适当值，FPDIS(NVPROT 的 0 位)必须被写逻辑 0 以使能块保护。因此值 0XDE 必须被写入到 NVPROT 以保护地址 0XE0000-0XFFFF。



图4-4.块保护机制

块保护的一个用途是为引导程序提供一个保护了的 FLASH 存储区域。这样引导程序就可以用于擦除并且重新写剩余的 FLASH 存储器。因为引导程序被保护，所以即使 MCU 在写入或者擦除操作过程中掉电，该程序也是完整的。

4.5.7 向量重定向

无论是哪种块保护，复位和中断向量都将被保护。向量重定向允许用户修改中断向量信息而不需要解保护引导程序和复位向量空间。向地址为\$FFBF 的 NVOPT 寄存器的 FNORED 位写入 0，可以使向量重定向有效。为了让重定向发生，必须设置位于\$FFBD 地址处的 NVPROT 寄存器让 FLASH 存储器中的一部分而不是全部块保护。所有的中断向量(\$FFC0 - \$FFFD)都会重定向，除了复位向量(\$FFFE:\$FFFF)。

举个例子，如果 FLASH 的 512 个字节被保护，保护地址区域是从\$FE00 到\$FFFF。则中断向量\$FFC0 - \$FFFD 被重定向到\$FDC0 - \$FDFD。以一个 SPI 中断来举例，地址\$FDE0: FDE1 将取代地址\$FFE0:FFE1 的值被中断向量使用。这就允许用户在离开保护区域时，可以使用新的程序代码和新的向量值去编程 FLASH 未被保护的部分，而留下包含默认向量地址的被保护区域不被改写。

4.6 安全性

MC9S08QD4 使用硬件电路来防止对 FLASH 存储器和 RAM 存储器的数据的非法访问。当安全机制使用时，FLASH 和 RAM 都被看作被保护资源。直接页寄存器，高地址页寄存器，背景调试控制器被看作是非保护资源。保护区的程序执行时，可以正常访问任何位置的内存数据和资源。通过背景调试接口或来自非保护存储空间执行程序来访问保护资源是不允许的(写操作被忽略，读时总为 0)。

是否使用安全机制是由 FOPT 中的非易失性的寄存器位 SEC01:SEC00 来设置的。在芯片复位时，非易失性的 NVOPT 的内容从 FLASH 中复制到工作着的高地址页寄存器 FOPT 中。用户通过写 NVOPT 来使用 MCU 的安全机制，当该动作与写 FLASH 存储器同时进行时也可被执行。如果是 1: 0 状态，则取消 MCU 保密性，其他的组合都启用 MCU 保密性。注意擦除的状态(1:1)使得 MCU 安全。在开发过程中，无论 FLASH 何时被擦除，立即将 NVOPT 中的 SEC00 位清 0 是明智的操作。这样就可以在复位以后让 MCU 仍然是非保护的。

当 MCU 处于安全模式时，芯片的调试模式是不能被使能的。这个独立的背景调试控制器仍然用于非插入的背景存储器访问命令，但是，MCU 不会进入背景调试模式，除非在复位的上升沿，保持 BKGD/MS 引脚低电平。

用户可以通过 8 字节的后门密钥来允许或者禁止安全机制。如果 NVOPT 或者 FOPT 中的非易失性位 KEYEN 是 0，后门密钥是无效的，这时如果没有擦除整个的 FLASH 就没有方法离开安全模式。如果 KEYEN 是 1，一个在安全模式下的用户程序可以通过以下的方法暂时离开安全模式：

- 1.将 FCNFG 的 KEYACC 位置 1。这使 FLASH 模块将后门密钥地址(NVBACKKEY 到 NVBACKKEY+7)写入的数据解释为比较密钥，而不是 FLASH 写入或擦除的第一步。
- 2.向 NVBACKKEY 到 NVBACKKEY+7 的位置依次写入 8 字节的用户进入密钥值。写入顺序为 NVBACKKEY 开始，NVBACKKEY+7 结束。通常，用户程序可以从 MCU 系统外部通过一个通讯接口(如 SCI)获得该密钥。
- 3.将 FCNFG 的 KEYACC 位置 0，如果用户写入的 8 字节和其在 FLASH 之中的 8 字节位置是匹配的，MCU 将 SEC01:SEC00 改写成 1: 0，从而取消 MCU 的保密性。

这种机制只能由处于保护模式下的代码执行，而不能由背景调试命令和非保护区域的代码执行。

后门密钥(NVBACKKEY 到 NVBACKKEY+7)分布在 FLASH 存储器的非易失性寄存器空间中，因此用户可以象写其他 FLASH 寄存器一样写这些寄存器。这个非易失性的寄存器与复位

和中断向量一起都位于同样的一块 512 字节的 FLASH 存储区域中，所以块保护这块区域也就保护了后门密钥。块保护不能通过用户的应用程序改变，因此，如果向量空间是块保护空间，这个后门安全密钥机制永远不能改变块保护、安全设置和后门密钥。

安全模式可以通过背景调试接口按照下面的步骤来消除：

- 1.通过写 FPROT 来使块保护失效，FPROT 只能通过背景调试命令来写，而不是通过应用软件。
- 2.如果需要可以整体擦除 FLASH 存储器。
- 3.FLASH 的空白检测，如果 FLASH 被完全擦除，安全模式被禁止直到下一次复位。为了避免下一次复位时返回安全模式，置 NVFOPT 的 SEC01:SEC00=1:0。

4.7 FLASH寄存器和控制位

FLASH 模块有 9 个 8 位的寄存器位于高地址页寄存器空间，两个位于 FLASH 存储器的非易失性寄存器空间，当复位时，内容被复制到相应的高页区域的控制寄存器。同样也有一个 8 字节的对照密钥在 FLASH 存储器中。参考表 4-3 和 4-4 所有 FLASH 寄存器绝对地址分配。这里通过这些寄存器和控制位的名称来引用它们。通常，Freescale 提供一个头文件把它们名称翻译为绝对地址。

4.7.1 FLASH时钟分频寄存器(FCDIV)

在任何擦除或写入操作之前，必须向该寄存器写入内容用以在有效的范围内设置时钟的频率。

	7	6	5	4	3	2	1	0
读	DIVLD	PRDI	DIV					
写	无效	V8						
复位	0	0	0	0	0	0	0	0

图4-5 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	时钟分频加载标志位——这个只读标志位被置位时，表示FCDIV寄存器在复位后已经被写过了。复位清零该位且第一次向该寄存器写入导致该位置位不管数据是否写入。 1 = FCDIV已经被写过，可以对FLASH进行擦除和写入操作。 0 = FCDIV没有被写过，不可以对FLASH进行擦除和写入操作。
6 PRDIV8	PRDIV8——FLASH预分频设置位 1 = FLASH分频器的时钟输入是总线时钟频率的八分之一。 0 = FLASH分频器的时钟输入是总线时钟频率。
5: 0 DIV	时钟分频位——FLASH时钟分频器通过这六位的值来对总线时钟频率(或者当PRDIV8=1时是总线时钟频率的八分之一)进行分频。在对FLASH进行操作时，内部时钟必须降到150KHz~200KHz。擦除/写入操作的时序脉冲是内部FLASH工作时钟的一个时钟周期，所以擦/写的时间相应的在6.7 μs~5 μs。自动的写入逻辑使用这个时钟周期的整数倍时间完成擦除或写入操作。内部时钟fCLK的计算方法见：等式4-1和等式4-2

如果 PRDIV8=0，则 $f_{CLK} = f_{Bus} \div ([DIV5:DIV0] + 1)$ (等式 4-1)

如果 PRDIV8=1，则 $f_{CLK} = f_{Bus} \div (8 \times ([DIV5:DIV0] + 1))$ (等式 4-2)

表 4-7 列出了对 PRDIV8 和 DIV5:DIV0 设置不同数值时的 FLASH 时钟频率。

表 4-7 FLASH 时钟分频器设置

f_{Bus}	PRDIV8 (二进制)	DIV5:DIV0 (十进制)	f_{CLK}	擦除/写入脉冲时间 (最少5μs,最多6.7μs)
-----------	-----------------	--------------------	-----------	------------------------------

8 MHz	0	39	200kHz	5μs
4 MHz	0	19	200kHz	5μs
2 MHz	0	9	200kHz	5μs
1 MHz	0	4	200kHz	5μs
200 MHz	0	0	200kHz	5μs
150 MHz	0	0	150kHz	6.7μs

4.7.2 FLASH选项寄存器(FOPT和NVOPT)

复位时，非易失性的 NVOPT 的内容从 FLASH 拷贝到 FOPT。为了改变这个寄存器的值，像平常一样擦除或者写入 FLASH 存储器中的 NVOPT 位置，然后执行 MCU 复位。

	7	6	5	4	3	2	1	0
读	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
写	无效	无效	无效	无效	无效	无效	无效	无效
复位	在复位时，这个寄存器从非易失性区域NVOPT中装载							

图4-6 FLASH选项寄存器(FOPT和NVOPT)

表 4-8 FLASH 选项寄存器(FOPT 和 NVOPT)

字段	描述
7 KEYEN	后门密钥机制使能——当这位为0时，后门密钥机制不可用于离开安全模式。后门密钥机制只能被安全的用户固件使用。BDM命令不能用于改变密钥的值。更多的后门密钥机制的描述见4.6节安全性。 KEYEN=0，禁止FLASH安全密钥。 KEYEN=1，如果用户固件写的一个8字节值与非易失性的后门密钥即 NVBACKKEY~NVBACKKEY+7相匹配，安全模式暂时失效直到下一次MCU复位。
6 FNORED	向量重定向禁止位——该位为1时，向量重定向不可用。 FNORED = 1，向量重定向禁止。 FNORED = 0，向量重定向使能。
1: 0 SEC0[1:0]	安全状态码——这两位决定MCU的安全状态，详见表4-9。当MCU在安全模式时，FLASH和RAM的内容不能被任何来自非法源(包括背景调试接口)的指令访问。安全模式更详细的信息参见4.6节安全性。

表 4-8 安全状态码¹

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

¹ 在一次成功的后门密钥进入或 FLASH 空白检测后 SEC01:SEC00 变为 1: 0。

4.7.3 FLASH配置寄存器(FCNFG)

	7	6	5	4	3	2	1	0
读	0	0	KEYACC	0	0	0	0	0
写	无效	无效		无效	无效	无效	无效	无效
复位	0	0	0	0	0	0	0	0

图4-7 FLASH配置寄存器(FCNFG)

表 4-10 FCNFG 寄存器字段描述

字段	描述
5 KEYACC	写访问密钥允许位——该位使能写访问密钥。更详细的描述见4.6节安全性。 KEYACC = 1, 写NVBACKKEY(\$FFB0 - \$FFB7)被认为是进行密码比较。 KEYACC = 0, 写\$FFB0 - \$FFB7被认为是FLASH写入或擦写命令的开始。

4.7.4 FLASH保护寄存器(FPROT和NVPROT)

复位期间, 非易失 NVPROT 的内容从 FLASH 拷贝到 FPROT。该寄存器在任何时刻可读, 但是用户程序向它写入是无效且没有意义的。

	7	6	5	4	3	2	1	0
读	FPS ⁽¹⁾							0
写								无效
复位	在复位时, 这个寄存器从非易失性区域NVOPT中装载							

(1) 背景命令可以改变FPROT寄存器中的这些位

图4-8 FLASH保护寄存器(FPROT)

表 4-11FPROT 寄存器字段描述

字段	描述
7:1 FPS	FLASH保护区域选择——当FPDIS = 0, 这七个位决定在FLASH末端的高地址位的未保护块的大小。保护块不能被擦除或写入
0 FPDIS	FLASH保护禁止位 FPDIS = 1, FLASH不进行保护。 FPDIS = 0, 由FPS2:FPS1:FPS0决定的FLASH区域被块保护。

4.7.5 FLASH状态寄存器(FSTAT)

	7	6	5	4	3	2	1	0
读	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
写		无效			无效	无效	无效	无效
复位	1	1	0	0	0	0	0	0

图4-9 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命令缓冲区空标志位——FCBEF位用于登记命令。它同时也标志命令缓冲区为空因此当一个突发写入发生时一个新的命令序列可以被执行。该位可以通过向它写1来清零或当一个突发写入命令被发送到FLASH阵列用于写入时也可清零该位。只有突发写入命令可以被缓存。 0 命令缓冲区满(还没准备接收其它的命令) 1 一个新的突发写入命令可以被写到命令缓冲区
6 FCCF	FLASH命令完成标志位——当命令缓冲区为空且没有命令在执行时该位自动置位。当一个新的命令开始时(通过向FCBEF写1来存入命令)，该位自动清零。向该位写入无效且没有效果。 FCCF = 1，所有命令都完成。 FCCF = 0，命令还在执行。
5 FPVIOL	保护标志位——当FCBEF位被清除来存入一个擦写命令试图访问保护区域时，FPVIOL位自动置1(这个错误命令被忽略)。向FPVIOL写1可以自动清FPVIOL。 FPVIOL = 1，表示试图对保护区域进行擦写操作。 FPVIOL = 0，表示不进行侵害保护设置。
4 FACCERR	访问出错标志位——当正常的命令序列没有正确的跟随时，在FCDIV寄存器初始化之前尝试擦除或者写入操作；或者当命令正在处理时MCU进入停止模式，FACCERR被置位。会引起访问出错的行为的详细讨论见4.5.4节访问错误。向FACCERR位写1将清FACCERR，向FACCERR写入0是没有意义和无效的。 FACCERR = 1，表示一个访问错误已经发生。 FACCERR = 0，没有访问错误发生。
2 FBLANK	FLASH空白标志位——如果执行空白检测命令检测到整个FLASH都是被擦除的，FBLANK位将自动置位。通过清除FCBEF来写入一个新的命令可以清FBLANK位。向该位写入是没有意义和无效的。 FBLANK = 1，一个空白检测命令完成之后且FCCF=1，FBLANK=1表示FLASH是空的，即所有的字节为\$FF。 FBLANK = 0，一个空白检测命令完成之后且FCCF=1，FBLANK=0表示FLASH是非空。

4.7.6 FLASH命令寄存器(FCMD)

在一般用户模式下只有 5 个命令代码可以被识别，见表 4-13。FLASH 的写入，擦除详细讨论见 4.5.2 节。

	7	6	5	4	3	2	1	0
读								
写	FCMD							
复位	0	0	0	0	0	0	0	0

图4-10 FLASH命令寄存器(FCMD)

表 4-13 FLASH 命令

命令	FCMD	等价的文件标志
空白检测	\$05	mBlank
字节写入	\$20	mByteProg
字节写入—快速模式	\$25	mBurstProg
页擦除	\$40	mPageErase
整体擦除(整个FLASH)	\$41	mMassErase

所有其他的命令都是非法的并且产生一个访问错误。整体擦除操作之后没有必要去执行空白检测命令。只有当作为安全解锁机制的一部分时空白检测命令才是需要的。

第五章 复位，中断和系统控制

5.1简介

本章介绍了 MC9S08QD4 系列 MCU 基本的复位和中断机制以及 MC9S08QD4 系列 MCU 的各种复位和中断源，一些来自外设模块的中断源在本手册的其他章节会做详细介绍，本章汇总了所有复位和中断源信息以方便参考。有些复位和中断源，如系统正常操作监视模块(COP)看门狗和定期中断定时器，不属于片上外设系统的部分，它们有单独的章节仅是系统控制逻辑的一部分。

5.2特征

复位和中断特征包括：

- 多种复位源用于灵活的系统配置和可靠的操作
- 复位状态寄存器（SRS）表明最近发生的复位源
- 用于每个模块的独立的中断向量（减少轮训溢出）（见表 5-2）

5.3MCU复位

复位 MCU 提供一种从已知的初始条件设置来启动处理的方法。在复位期间，多数控制和装提案寄存器被强制初始化值且程序计数器从复位向量处装载（0XFFFE:0XFFF）。片上外设模块被禁止同时 I/O 引脚被初始化配置为上拉设备禁止的通用高阻抗输入。条件码寄存器(CCR)的 I 位被置位来屏蔽中断，因此用户程序有机会初始化堆栈指针(SP)和系统控制设置。复位时，SP 被设为\$00FF。

MC9S08QD4 系列 MCU 包含以下复位源：

- 外部引脚复位—允许使用 SOPT1 寄存器中的 RSTPE 位
- 上电复位(POR)
- 低电压检测（LVD）
- 正常操作模式定时模块（COP）
- 非法代码检测（ILOP）
- 非法地址检测（ILAD）
- 背景调试强制复位

除了背景调试强制复位以外，每种复位源在系统复位状态寄存器中都有相应位。

5.4 计算机正常操作监控模块(COP)看门狗

当应用程序没有按照预期那样执行时，COP 看门狗会强制系统复位。为防止来自 COP 定时器（当它使能时）的系统复位，应用程序必须周期性的复位 COP 计数器。如果应用软件在 COP 定时器溢出前忘记或者没有复位 COP，就会产生一个系统复位信号使系统回到可行的开始点。在任何复位后，SOPT1 寄存器中的 COPE 位置位使能 COP 看门狗（详情见 5.85 节，系统选项寄存器 2（SOPT2））。如果 COP 看门狗在应用程序中未被使用，那么可通过清零 COPE 位来禁止它。通过写任意值到 SRS 的地址可复位 COP 计数器。这次的写入不影响只读寄存器

SRS 中的数据。相反写该地址是解码和发送一个复位信号给 COP 计数器。

SOPT2 寄存器中的 COPCLKS 位（详情见 5.8.5 节，“系统选项寄存器 2（SOPT2）”）选择用于 COP 定时器的时钟源。时钟源可以是总线时钟或一个内部 32KHz 时钟源。对于每个时钟源都可通过 SOPT1 寄存器中的 COPT 位进行相应的定是溢出控制，表 5-1 总结了 COPCLKS 和 COPT 位的控制功能。COP 看门狗缺省为 32KHz 时钟操作并为长溢出时间（ 2^8 周期）。

表 5-1COP 配置选项

控制位		时钟源	COP溢出计数
COPCLKS	COPT		
0	0	~32KHz	2^{10} 周期（32ms） ¹
0	1	~32KHz	2^{13} 周期（256ms） ¹
1	0	总线	2^{13} 周期
1	1	总线	2^{18} 周期

¹栏中所显示的值是基于 $t_{RTI}=1ms$ 。见A.8.1节“控制定时”的 t_{RTI} 。

即使应用程序将使用 COPE, COPCLKS 和 COPT 复位的缺省设置，用户也必须在复位初始化期间写仅写一次寄存器 SOPT1 和 SOPT2 来锁存设置。那样如果应用程序跑飞了这些设置也不会被意外地改变。初始化写 SOPT1 和 SOPT2 将复位 COP 计数器。

写 SRS 清零 COP 计数器一定不能放在中断服务例程中(ISR)因为即使主应用程序失败 ISR 能继续周期性执行下去。

在背景调试模式下，COP 计数器计数将不会增加。

当选择总线时钟源被时，COP 计数器将不会增加虽然系统处于停止模式。一旦 MCU 退出停止模式 COP 计数器将继续工作。

当选择 32KHz 时钟源时，COP 计数器被重新初始化为零以进入停止模式。一旦 MCU 退出停止模式 COP 计数器从零开始。

5.5中断

中断提供了一种方法来保存当前 CPU 状态和各寄存器内容，执行中断服务例程(ISR)，然后恢复 CPU 状态使处理器从中断发生的地方继续执行。这不同于软件中断(SWI)，SWI 是一条程序指令，而这种中断都是由 IRQ 引脚边缘触发或者连续的 I/O 接收等硬件事件引起的。该调试模块在某个条件下也能产生一个 SWI。

如果一个可用中断源中一个事件发生，一个相关的只读状态标志位将被置位。当本地中断屏蔽位被置为 1(使得中断可用)且 CCR 的 I 位为 0(允许中断)时 CPU 才会对这个事件做出反应。CCR 的中断屏蔽位(IMASK)在复位后被置位可以屏蔽所有的中断源。这样在清这一位以允许 CPU 响应中断之前就可以让用户程序初始化堆栈指针并执行其他系统设置。

当 CPU 接收到一个有效的中断请求时，它在完成当前的指令后响应此中断。中断的执行过程就像 SWI 指令的一个周期一个周期地执行顺序一样，包括以下步骤：

- 在堆栈中保存 CPU 寄存器；
- 将 CCR 中的 I 位置位来屏蔽中断；
- 取出当前等待的最高优先级的中断向量；
- 用从中断向量地址处取得的前三个字节的程序信息来填充指令队列；

当 CPU 响应中断时，I 位自动置位以防止其他的中断进入而打断 ISR(称为嵌套中断)。通常状态下，当 CCR 恢复到进入 ISR 前堆栈保留的值，I 位回复到 0。在很特殊的情况下，即若中断过程也允许响应新的中断，可在中断服务程序中用 CLI 指令开放中断。I 位可能会被中断清零让其他中断程序可以在不等待第一个中断服务程序结束后就先执行。这种做法不被推荐，除非是非常有经验的编程人员，否则会导致一些难以调试的错误，可用其他编程技巧处理相关问题。

中断服务程序由 RTI 指令结束，用堆栈中的值来恢复 CCR、A、X 及 PC 寄存器，用以获得先前被保存的信息。

注：

为了能够与 HC08 兼容，H 寄存器是不会自动的保存和恢复的。因此，好的编程习惯是：在中断服务例程(ISR)开始时将 H 压栈，在 ISR 返回指令 RTI 前将其恢复。

当 I 位被清零时有两个或更多的中断为决策时，具有最高优先级的中断源被最先执行（见表 5-2）。

5.5.1 中断堆栈结构

图 5-1 显示了一个堆栈结构的内容和组织。中断之前，堆栈指针(SP)指向堆栈的下一个可用的位置。CPU 寄存器的当前值被储存在堆栈中。依次保存的是从程序计数器中低地址位(PCL)开始到 CCR 结束的部分。在这些寄存器入栈以后，SP 指向堆栈中的下一个可用位置，这个位置的地址是 CCR 被保存的地址处的下一个地址。入栈的 PC 值是主程序中的若中断没有发生所要执行的下一个指令的地址。

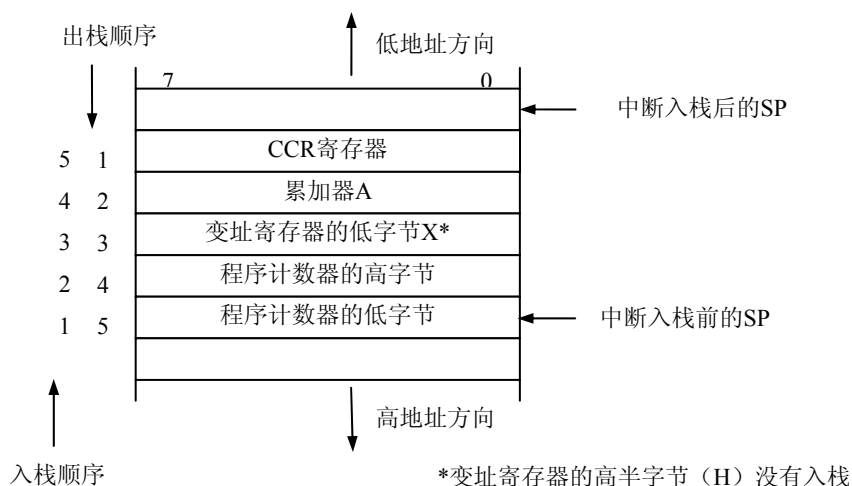


图5-1 中断堆栈框架

当一个 RTI 指令被执行时，这些值都会按照出栈的顺序从堆栈中恢复过来。作为 RTI 序列的一部分，CPU 通过读取三个字节的程序信息来装满流水线，从由堆栈中恢复的 PC 地址开始。

由中断触发所修改的状态标志必须在 ISR 返回前清除。一般地，这个标志应该在 ISR 开始时清零，这样另一个同源中断发生时，该状态标志可以记录这个事件，使得在完成当前 ISR 后，该事件可被响应。

5.5.2 外部中断请求引脚 (IRQ)

外部中断由 IRQ 状态和控制寄存器 IRQSC 管理。当 IRQ 功能可用，同步逻辑监控引脚是边沿触发还是边沿/电平触发。当 MCU 处于停止状态并且系统时钟被关闭时，一个异步通道被使用以至于 IRQ(如果被使能)可以将 MCU 从停止模式唤醒。

5.5.2.1 引脚配置选项

为了将引脚中断请求(IRQ)输入，寄存器 IRQSC 的 IRQ 引脚使能控制位(IRQPE)必须设为 1。作为 IRQ 输入时，用户可以选择检测的边沿和电平的极性(IRQEDG)，不论仅是边沿检测还是

边沿和电平同时检测，事件发生时是直接触发中断还是只置 IRQF 标志，待软件处理。

当 IRQ 引脚使能时被设定使用内部电阻（IRQPDD=0），选择下拉电阻或上拉电阻取决于检测的极性。如果用户期望使用一个外部上拉或下拉，那么 IRQPDD 被写 1 来禁用内部电阻。

当该引脚被配置作为 IRQ 引脚输入时，BIH 和 BIL 指令用来检测 IRQ 引脚上的电平。

注：

该引脚不需通过一个钳位二极管连接到 V_{DD} 且不能给它加高于 V_{DD} 的电压。

内部上拉 IRQ 引脚标准电压在 $V_{DD} - 0.7V$ 之间。所有的其他带有上拉电阻的引脚将有 V_{DD} 的无负荷度量。

5.5.2.2 边沿/电平触发

IRQMOD 控制位重新配置检测逻辑因此它检测边沿事件和引脚电平。在这个边沿检测模式下，当一个边沿被检测到时（当 IRQ 引脚从一个已知电平发生改变时）IRQF 状态标志置位，该标志一直保持置位只要 IRQ 引脚维持改变后的电平。

5.5.3 中断向量、中断源和局部屏蔽

表 5-2 提供了一个所以有中断源的摘要。高优先级的源被定位在表的底部。中断服务程序的高地址字节被定位到向量地址阵列的首地址，中断服务程序的低地址字节被定位在向量地址阵列的下一个高地址。

当中断条件发生，相关的标志位被置位。如果该相关的中断屏蔽位为 1，则中断请求发送到 CPU。在 CPU 的内部，如果全局中断屏蔽位为 0 (CCR 的 I 位)，CPU 将完成当前的指令，然后将 CPU 寄存器 PCL、PCH、X、A 和 CCR 入栈，I 位置位，然后取最高中断优先级的中断向量。然后到中断服务程序中处理。

表 5-2 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
从低到高	31-24	从\$FFC0:FFC1 到\$FFCE:FFCF	位使用的向量空间 (用户程序可用)				
	23	\$FFD0:FFD1	Vrti	系统 控制	RTIF	RTIE	实时中断
	22	\$FFD2:FFD3					
	21	\$FFD4:FFD5					
	20	\$FFD6:FFD7					
	19	\$FFD8:FFD9	Vadc1	ADC1	COCO	AIEN	ADC1
	18	\$FFDA:FFDB	Vkeyboard1	KBI1	KBF	KBIE	键盘引脚
	17	\$FFDC:FFDD					
	16	\$FFDE:FFDF					
	15	\$FFE0:FFE1					
	14	\$FFE2:FFE3					
	13	\$FFE4:FFE5					
	12	\$FFE6:FFE7					
	11	\$FFE8:FFE9					
	10	\$FFEA:FFEB	Vtpm2ovf	TPM2	TOF	TOIE	TPM2溢出
	9	\$FFEC:FFED					
	8	\$FFEE:FFEF	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2通道 0
	7	\$FFF0:FFF1	Vtpm1ovf	TPM1	TOF	TOIE	TPM1溢出
	6	\$FFF2:FFF3	Vtpmch1	TPM1	CH1F	CH1IE	TPM1通道 1
	5	\$FFF4:FFF5	Vtpmch0	TPM1	CH0F	CH0IE	TPM1通道 0

	4	\$FFF6:FFF7					
	3	\$FFF8:FFF9	Virq	IRQ	IRRQF	IRQIE	IRQ引脚
	2	\$FFFA:FFFB	Vlvd	系统控制	LVDF	LVDIE	低电压检测
	1	\$FFFC:FFFD	Vswi	CPU	SWI指令		软中断
	0	\$FFFE:FFFF	Vreset	系统控制	COP ——LVD—— RESET 引脚 非法操作码 非法地址 POR	COPE LVDRE RSTPE	看门狗定时器 低电压检测 外部引脚 非法操作码 非法地址 上电复位

5.6 低电压检测系统 (LVD)

MC9S08QD4 系列包含一个预防低电压的保护系统，在电压变更时用来保护存储器的内容和控制 MCU 系统状态。这个系统是由一个上电复位(POR)电路和一个 LVD 电路(用户选择触发电压，高电压 V_{LVDH} 或者低电压 V_{LVDL})。当 SPMSC1 的 LVDE 为高并且触发电压通过 SPMSC3 的 LVDV 位被选择时，LVD 电路被激活。停止模式时 LVD 在所有入口上不可用除非 SPMSC1 的 LVDSE 位被置位。如果 LVDSE 和 LVDE 都被置位，则 MCU 不能进入停止 1 模式和停止 2 模式，并且处于 LVD 被激活的停止 3 模式的功耗将会变大。

5.6.1 上电复位操作

当 MCU 刚通电时，或者当供应电压降落到 V_{POR} 以下时，POR 电路将会产生一个复位条件。随着供应电压的升高，LVD 电路将保持芯片在复位状态直到供电电压升高到大于 V_{LVDL} 。伴随着上电复位，SRS 的 POR 位和 LVD 位都将被置位。

5.6.2 LVD复位操作

此时 LVD 能被设置为在侦测到一个低电压条件时(即 LVDRE 为 1 时)产生一个复位。LVD 复位发生后，LVD 系统将会保持 MCU 在复位状态直到供应电压升高到高于由 LVDV 设定的电压。当 LVD 复位产生时，LVDV 是不会改变的。伴随着 LVD 复位或者上电复位，SRS 寄存器的 LVD 位将被置位。

5.6.3 LVD中断操作

当一个低电压条件被侦测到，并且 LVD 电路被设置为中断操作(LVDE 置位，LVDIE 置位，LVDRE 清零)，然后 SPMSC1 的 LVDF 将会被置位且一个 LVD 中断将发生。

5.6.4 低电压警告(LVW)

LVD 系统有一个低电压警告标志位用以告诉用户供应电压已经接近 LVD 电压，但是目前还在 LVD 电压以上。LVW 没有与此相关的中断。LVW 有两个触发电压可供选择，高的 V_{LVWH} 和低的 V_{LVWL} 。触发电压通过 SPMSC2 寄存器的 LVWV 位选择。

5.7 实时中断(RTI)

实时中断功能用来产生定期中断。RTI 认可两类时钟源，1KHz 的内部时钟或一个 32KHz

ICS 时钟如果可用的话。SRTISC 中的 RTICKS 位通常被用来选择 RTI 时钟源。

当 MCU 运行在等待或停止模式 3 下时两个时钟源都能被使用。当在停止模式 3 下使用 32KHz ICS 时钟时，它必须在停止模式下使能（EREFSTEN =1）并配置成低频率操作（RANGE=0）。仅当选择内部 1KHz 时钟源时能够把 MCU 从停止模式 1 或停止模式 2 中唤醒。

SRTISC 寄存器包括一个只读状态标志，一个只写识别标志，和一个 3 位控制值（RTIS）用来选择七个唤醒周期中的一个。该 RTI 含有一个本地中断使能，RTIE，允许屏蔽实时中断。通过置 RTIS 每一位为零使 RTI 禁止同时没有中断产生。详情见 5.8.7 节，“系统实时中断状态和控制寄存器（SRTISC）”。

5.8 复位、中断、系统控制寄存器和控制位

涉及到复位和中断系统的寄存器有一个在直接页寄存器区的 8 位寄存器，8 个在高地址页寄存器区的 8 位寄存器。

寄存器绝对地址的分配参见本手册的第 3 章有关“操作模式”直接页寄存器的介绍。这部分只通过他们的名字来引用相关的寄存器和控制位。Freescall 提供的通用文件或头文件习惯将这些名称转化为相应的绝对地址。

SOPT1, SOPT2 和 SPMSC2 寄存器的一些控制位涉及到操作的模式。这里只提供那些位的简要描述，而相关函数在第 3 章“操作模式”里有更加详细的论述。

5.8.1 中断引脚请求状态和控制寄存器（IRQSC）

直接页寄存器包括 2 个总是读为 0 的未用位，4 个可读写位，一个只读状态位，和一个只写位。这些位都是被用于设置 IRQ 功能，报告状态和确认 IRQ 事件。

	7	6	5	4	3	2	1	0
读	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IQRMOD
写	无效				无效	IRQACK		
复位	0	0	0	0	0	0	0	0

图5-2中断引脚请求状态和控制寄存器（IRQSC）

表 5-3 IRQSC 寄存器字段描述

字段	描述
6 IRQPDD	中断请求（IRQ）上拉设备禁止——当IRQ引脚使能（IRQPE=1）允许一个外部设备使用时这个读/写控制位被用来禁止内部上拉/下拉设备。 0 如果IRQPE=1IRQ上拉允许。 1 如果IRQPE=1IRQ上拉禁止。
5 IRQEDG	中断请求(IRQ)边沿选择——这个可读写控制位用来选择导致IRQF被置位的IRQ引脚的边沿极性或者电平。IQRMOD控制位决定IRQ引脚是边沿/电平触发还是仅仅为边沿触发，当IRQ引脚作为IRQ输入引脚使能，并且设置为侦测引脚上升沿时，可选择上拉电阻被重新配置为可选择下拉电阻。 IRQEDG=1时，上升沿或者上升沿/高电压触发。 IRQEDG=0时，下降沿或者下降沿/低电压触发。
4 IRQPE	IRQ引脚使能——这个可读写控制位使能IRQ引脚功能。当这位被置位，IRQ引脚能被用做一个中断请求。 IRQPE=1时，IRQ引脚功能可用。 IRQPE=0时，IRQ引脚功能不可用。
3 IRQF	IRQ标志——该只读状态控制位表明一个中断请求时间已发生。 0 没有IRQ请求。 1 IRQ事件被检测。
2 IRQACK	IRQ确认——该只写位被用于确认中断请求事件(写1清IRQF位)。写0表示没有意义或无影响。读其通常为0。如果选择边沿/电平侦测(IQRMOD=1)，当IRQ引脚还保持它的电平时IRQF不能被清零。

1 IRQIE	IRQ中断使能——这个可读写控制位决定IRQ事件是否产生硬件中断请求。 IRQIE=1时, IRQF=1时产生硬件中断请求。 IRQIE=0时, 从IRQF发来的硬件中断请求不可用。
0 IRQMOD	IRQ检测模式——该可读写控制位选择是边沿/电平触发还是边沿触发。IRQED控制位决定触发中断请求事件的边沿和电平的极性。5.5.2.2节有更加详细的介绍。 IRQMOD=1时, 在下降沿和低电压, 或者上升沿和高电压时IRQ事件触发。 IRQMOD=0时, 仅在下降沿或者上升沿时IRQ事件触发。

5.8.2 系统复位状态寄存器 (SRS)

这个寄存器包括用于指示最近复位源的 7 个只读标志位。当一个调试器通过向 SBD FR 寄存器的 BDFR 位写入 1 强制复位时, SRS 的各标志位都不会被置位。向该寄存器地址写入任意值可清零 COP 看门狗计数器但不影响寄存器的内容。这些位的复位状态取决于 MCU 复位的原因。

	7	6	5	4	3	2	1	0
读	POR	PIN	COP	ILOP	0	ICG	LVD	0
写	向该寄存器地址写入任意值可清零COP看门狗计数器							
上电复位	1	0	0	0	0	0	1	0
低电压复位	0	0	0	0	0	0	1	0
其他复位	0	(1)	(1)	(1)	(1)	0	0	0

(1)若上述任一复位源在进入复位时活跃将导致相应位被置位; 与复位源相应的位在进入复位时不活跃的将被清零

图5-3系统复位状态

表 5-4 SRS 寄存器字段描述

字段	描述
7 POR	上电复位——复位是由上电检测逻辑引起。因为内部供应电压在这个时候发生跳跃, 低电压复位(LVD)状态位也被置位, 用以表明当内部供应电压下降到LVD限度时复位发生。 POR=1时, 复位由POR引起。 POR=0时, 复位不是由POR引起。
6 PIN	外部复位引脚——复位由在外部复位引脚上产生的有效低电压而引发。 0 复位不是由外部复位引脚引发。 1 复位由外部复位引脚引发。
5 COP	计算机正常操作监控(COP)看门狗——复位因看门狗定时器溢出而引发。复位源可通过COPE=0锁住。 0 复位不是由COP时间溢出而引起。 1 复位原因是COP时间溢出。
4 ILOP	非法操作码复位——由非法操作码而引起的复位。如果stop被通过设置SOPT寄存器的STOPE=0而设为不可用的, 则STOP指令被认为是非法操作。如果主动背景调试模式通过设置BDCSC寄存器中的ENBDM=0设为不可用, 则BGND指令被认为是非法操作。 1复位由非法操作码引起。 0 复位不是由非法操作码引起。
3 ILAD	非法地址复位——复位由试图存取一个无效地址的数据或指令引起 0 复位不是由非法地址引起 1 复位由非法地址引起
1 LVD	低电压检测复位——如果LVD复位可用(LVDE=LVDRE=1)且供应电压下降到低于LVD临界电压, 就会产生LVD复位。该位由POR置位。 LVD=1, 复位由LVD或者POR引起。 LVD=0, 复位不是由LVD或POR而引起的。

5.8.3 系统背景调试强制复位寄存器（SBDFR）

这个高地址寄存器包括一个只写控制位。一个连续背景命令例如 WRITE_BYTE 必须被写入 SBDFR。系统忽略应用程序对该寄存器的写入。读始终返回\$00。

	7	6	5	4	3	2	1	0
读	0	0	0	0	0	0	0	0
写	无效	无效	无效	无效	无效	无效	无效	BDFR ⁽¹⁾
复位	0	0	0	1	0	0	0	0

(1) BDFR只能通过背景调试命令而不能通过用户程序写入

图5-4系统背景调试强制复位寄存器（SBDFR）

表 5-5 SBDFR 寄存器字段描述

字段	描述
0 BDFR	背景调试强制复位——一个连续的背景调试模式命令例如WRITE_BYTE允许一个外部调试器强制目标系统复位。向该位写1可强制MCU复位。但用户程序不可写该位。若要进入用户模式，PTA4/TPM2CH00/BKGD/MS必须在执行WRITE_BYTE命令后立即置高。若要进入BDM模式，PTA4/TPM2CH00/BKGD/MS必须在执行WRITE_BYTE命令后立即置低。详情见A8.1，“控制定时”。

5.8.4 系统选项寄存器1（SOPT1）

这个高地址寄存器是只写寄存器因此在复位后仅第一次写被接收。它可在任何时间被读取。任何并发试图写 SOPT1（有意或无意）将别忽略以避免对这些敏感设置的意外改变。在用户复位初始化程序来设置期望的控制（即使这些设置和复位设置相同）期间 SOPT1 必须被写。

	7	6	5	4 ¹	3	2	1	0
读					0	0		
写	COPE	COPT	STOPE	无效	无效	无效	BKGDPE	RSTPE
复位	1	1	0	1	0	0	1	U
上电复位	1	1	0	1	0	0	1	0

¹位4被保留，写该位将改变该值但对MCU无影响。U=无影响

图5-5系统选项寄存器1（SOPT1）

表 5-6 SOPT1 寄存器字段描述

字段	描述
7 COPE	COP看门狗使能——这个只写一次位选择COP看门狗是否使能 0 COP看门狗定时禁止 1 COP看门狗定时使能（溢出时间强制复位）
6 COPT	COP看门狗溢出时间——这个只写一次位选择COP.COPT的溢出时间周期 该周期在SOPT2中被定义。 0 短溢出时间周期被选择 1 长溢出时间周期被选择
5 STOPE	停止模式使能——这个一次写入位在复位后默认值为0，使得停止模式不可用。如果停止模式不可用且用户程序试图去执行STOP指令，将产生一个非法操作码复位。 STOPE=1，Stop停止模式可用。 STOPE=0，Stop停止模式不可用。
1 BKGDPE	背景调试模式引脚使能——此位使能PTG0/BKGD/MS引脚功能为BKGD/MS。当这个位被清零，该引脚将作为PTD0使用，只是一个输出通用I/O口。在任何复位后，该引脚默认为BKGD/MS。 BKGDPE=1，BKGD引脚可用。 BKGDPE=0，BKGD引脚不可用。

0 RSTPE	复位引脚使能——当置位时这个只写一次位使能PTA5/TPM2CH0I/IRQ/RESET引脚作为复位脚。当清零时，该引脚作为它的输入引脚使用。在MCU上电复位后该引脚缺省为仅输入端口功能。当RSTPE置位时，复位引脚上的内部上拉有效。 0 PTA5/TPM2CH0I/IRQ/RESET引脚作为PTA5，IRQ,TPM2CH0I引脚 1 PTA5/TPM2CH0I/IRQ/RESET引脚作为复位脚
------------	---

5.8.5 系统选项寄存器2（SOPT2）

该高位寄存器所包含的位用来配置基于 MC9S08QD4 系列设备 MCU 具体特性。

	7	6	5	4	3	2	1	0
读	COPCLKS ¹	0	0	0	0	0	0	0
写		无效	无效	无效	无效	无效	无效	无效
复位	0	0	0	0	0	0	0	0

¹该位在复位后仅能写一次。其它写将被忽略

图5-7系统选项寄存器2（SOPT2）

表 5-7SOPT2 寄存器字段描述

字段	描述
7 COPCLKS	COP看门狗时钟选择——这个只写一次位选择COP看门狗的时钟源。 0 选择内部32KHz时钟 1 选择总线时钟

5.8.6 系统设备识别寄存器(SDIDH、SDIDL)

这个高地址只读寄存器使得开发系统可以识别 HCS08 的 MCU 类型和修正版本号。允许开发软件识别目标 MCU 的特殊存储块，寄存器和目标 MCU 中的控制位。

	7	6	5	4	3	2	1	0
读	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
写	无效	无效	无效	无效	无效	无效	无效	无效
复位	0 ⁽¹⁾	0 ⁽¹⁾	0 ⁽¹⁾	0 ⁽¹⁾	0	0	0	0

(1) 这个在固件代码中版本号反映该版本

图5-7系统设备识别寄存器—高位（SDIDH）

表 5-8 SDIDH 寄存器字段描述

字段	描述
7:4 REV[3:0]	修正版本号——SDIDH地址的高四位是硬件代码来反映当前屏蔽设置修正版本号（0-F）。
3:0 ID[11:8]	识别编号——HCS08系列的所有的派生版本都有一个唯一识别编号。MC9S08QD4被硬编码为值\$011。见表5-9 ID位

	7	6	5	4	3	2	1	0
读	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
写	无效	无效	无效	无效	无效	无效	无效	无效
复位	0	0	0	0	1	0	0	1

图5-8系统设备识别寄存器—低位（SDIDL）

表 5-9 SDIDL 寄存器字段描述

字段	描述
7:0 ID[7:0]	识别编号——HCS08系列的所有的派生版本都有一个唯一识别编号。MC9S08QD4被硬编码为值\$011。见表5-8 ID位

5.8.7 系统实时中断状态和控制寄存器 (SRTISC)

这个高地址页寄存器包含 RTI 状态和控制位。

	7	6	5	4	3	2	1	0
读	RTIF	0	RTICLKs	RTIE	0	RTIS		
写	无效	RTIACK			无效			
复位	0	0	0	0	0	0	0	0

图5-9系统实时中断状态和控制寄存器 (SRTISC)

表 5-10 SRTISC 寄存器字段描述

字段	描述
7 RTIF	实时中断标志——这个只读状态位表明周期性唤醒定时器溢出 0 周期性唤醒定时器未溢出 1 周期性唤醒定时器溢出
6 RTIACK	实时中断识别——这个只写位用来确认实时中断请求(写1可清RTIF)。向该位写0表示没有意义且无效。读它总为0。
5 RTICLKs	实时中断时钟选择——该读写位选择实时中断的时钟源。 1 实时中断请求时钟源为外部时钟。 0 实时中断请求时钟源为内部1KHz振荡器。
4 RTIE	实时中断使能——这个只读位使能实时中断。 0 实时中断禁止 1 实时中断允许
2:0 RTIS	实时中断延时选择——这些可读写位选择RTI的唤醒延时。见表5-11。

表 5-11 实时中断周期

RTIS2:RTIS1:RTIS0	1KHz的时钟源延时 ^(1,2)	使用32KHz ICS时钟 周期= t_{ext}^3
0:0:0	禁止RTI	禁止RTI
0:0:1	8ms	256分频
0:1:0	32ms	1024分频
0:1:1	64ms	2048分频
1:0:0	128ms	4096分频
1:0:1	256ms	8192分频
1:1:0	512ms	16384分频
1:1:1	1024ms	32768分频

¹显示在该表中的值基于 $t_{RTI}=1$ ms. 详情见A8.1 “控制定时”。

²最初的实时中断定时溢出周期将增加到1KHz小于指定的时间

³ t_{ext} 是32KHz ICS时钟频率的周期

5.8.8 系统电源管理状态和控制寄存器1 (SPPMSC1)

这个高地址寄存器包含状态和控制位用来支持低电压检测功能，并使能 ADC 模块使用的能带隙参考电压。为配置低电压检测触发，见表 5-13SPMSC2 中的 LVDV 位。

	7	6	5	4	3	2	1 ¹	0
读	LVDf	0	LVDIE	LVDRE ²	LVDSE	LDVE ²	0	BGBE
写	无效	LVDACK					无效	
复位	0	0	0	1	1	1	0	0

¹位1是保留位必须一直写为0

²在复位后该位只能被写一次。额外写无效

图5-10系统电源管理状态和控制寄存器1（SPPMSC1）

表 5-12 SPPMSC1 寄存器字段描述

字段	描述
7 LVDF	低电压检测标志——若LVDE=1，该只读状态位表明一个低电压检测事件。
6 LVDACK	低电压检测识别——该只写位用来识别低电压检测错误（写1清零LVDF）。读为0。
5 LV DIE	低电压检测中断使能——该读写位使能从LVDF发出的硬件中断请求。 1 当LVDF=1时请求一个硬件中断。 0 硬件中断请求不可用。
4 LVDRE	低电压检测复位使能——该只写一次位使能LVDF事件来产生一个硬件中断复位（假设LVDE=1）。 1 当LVDF=1时强制MCU复位 0 LVDF不会产生一个硬件复位
3 LV DSE	低电压检测使能——若LVDE=1，当MCU处于停止模式时该读写位决定低电压检测功能操作是否有效 0 停止模式期间低电压检测禁止 1 停止模式期间低电压检测允许
2 LV DE	低电压检测使能——这个可读写位使能低电压检测逻辑和限定该寄存器的其他位的操作。 1 LVD逻辑可用。 0 LVD逻辑不可用。
0 BG BE	能带隙缓冲使能——该位使能一个内部缓冲区用于能带隙参考电压被ADC模块使用。 0 能带隙缓冲禁止 1 能带隙缓冲允许

5.8.9系统电源管理状态和控制寄存器2（SPPMSC2）

该高地址页寄存器包含状态和控制位来配置 MCU 停止模式行为。详情见 3.6 节，“停止模式”。

	7	6	5	4	3	2	1 ¹	0
读	LVWF	0	LVDV	LVWV ²	PPDF	0	无效	PPDC ¹
写	无效	LVDACK			无效	PPDACK		
POR	0 ²	0	0	0	0	0	0	0
LVDR	0 ²	0	U	U	0	0	0	0
其它复位	0 ²	0	U	U	0	0	0	0

¹在复位后该位只能被写一次。额外写无效。U=复位无影响

²LVWF下列情况下将被置位，当V_{SUPPLY}跳转电压位于触发点之下时或复位后V_{SUPPLY}电压一直处于V_{LVW}之下。

图5-11系统电源管理状态和控制寄存器2（SPPMSC2）

表 5-13 SPPMSC2 寄存器字段描述

字段	描述
7 LVWF	低电压警告标志——该位显示低电压警告状态。 1 低电压警告已经发生或者正在发生。 0 低电压警告没有发生。
6 LVWACK	低电压警告确认位——该位表示低电压警告状态，如果低电压警告不是正在发生，向该位写入1，使得LVWF清零。
5 LVDV	低电压检测电压选择——该位选择LVD临界电压(VLVD)。 1 选择较高临界电压(VLVD=VLVDH)。 0 选择较低临界电压(VLVD=VLVDL)。
4 LVWV	低电压警告电压选择——该位选择LVW临界电压(V _{LVW})。 1 选择较高临界电压(V _{LVW} =V _{LVWH})。 0 选择较低临界电压(V _{LVW} =V _{LVWL})。
3 PPDF	局部低功耗标志位——该位显示MCU已退出停止模式2。
2 PPDACK	局部低功耗确认位——向该位写入1可清零PPDF位。
0 PPDC	局部低功耗控制位——这个一次性写入PPDC位控制进入哪种低功耗模式，stop1还是stop2。 1 如果PDC置位，选择stop2。 0 如果PDC置位，选择stop1。

第六章 并行输入/输出控制

这部分解释使用软件控制相关的并行输入/输出（I/O）和引脚控制。MC9S08QD4 系列含有一个并行 I/O 口，该 I/O 口包含 4 个 I/O 引脚，一个只输出引脚和一个只输入引脚。关于引脚分配和这些引脚的外部硬件考虑事项的详细信息见第二章，“外部信号描述”。

所有这些 I/O 引脚复用片上外设功能如表 2-1 所示。外设模块优先级高于 I/O 引脚优先级因此当一个外设被允许时，与 I/O 功能相关的复用引脚被禁止。在复位后，复用的外设功能被禁止因此这些引脚将由 I/O 控制。所有的 I/O 引脚被配置为上拉禁止（ $PtxPEN=0$ ）的输入（ $PTxDDn=0$ ）引脚，除了仅输出引脚 PTA4 缺省为 BKGD/MS 引脚。

注：

在所有包裹上的所有通用 I/O 引脚都是可用的。为了避免额外的电流流入输入引脚，应用程序中的用户的初始化例程必须使能片上上拉或改变未使用无连接的引脚的方向置为输出因此引脚电流不会浮动。

6.1 端口数据和数据方向

读和写并行 I/O 口是通过端口数据寄存器来执行的。输入或输出的方向是由端口数据方向寄存器来控制的。一个独立引脚的并行 I/O 端口功能将图 6-1 的框图进行图例说明。

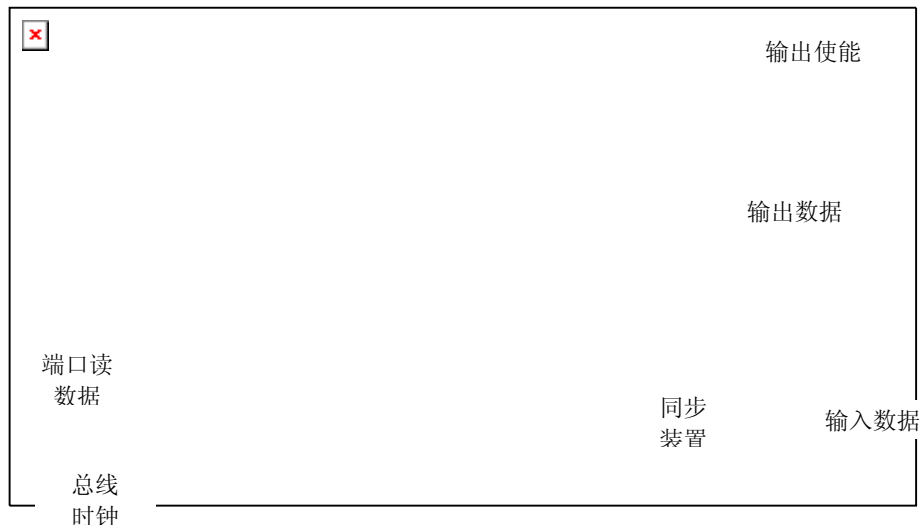


图6-1并行I/O框图

该数据方向控制位（ $PTxDDn$ ）决定相应引脚的输出缓冲区是否允许，同时也控制端口数据寄存器读数据的来源。相应引脚的输入缓冲区一直有效除非该引脚使能作为一个模式功能引脚或为一个只输出引脚。

当一个引脚的数据复用功能允许时，输出缓冲区由复用功能控制。不论怎样，数据方向寄存器位将继续控制端口数据寄存器的读。

当一个引脚的复用模式功能被允许时，输入和输出缓冲区都被禁止。任何数据端口位被置位输入（ $PTxDDn=0$ ）该位读为 0 且输入缓冲区被禁止。总的来说，数字输入功能和模拟输入功能复用同一引脚时，模拟输入功能优先级更高，也就是如果数字和模拟功能同时使能，模拟输入功能控制引脚。

在改变引脚方向为输出时写端口数据寄存器，这是良好的编程习惯。这样确保了旧的数值不会即刻出现在端口数据寄存器。

6.2 引脚控制——上拉、回转速率和驱动强度

相关的并行 I/O 口是一些列寄存器，它们位于高地址页寄存器空间是独立操作的并行 I/O 寄存器。这些寄存器用来控制引脚的上拉，回转速率和驱动强度。

6.3 停止模式下的引脚行为

在执行一个 STOP 指令后引脚行为取决于它所进入的停止模式。下面将介绍各种停止模式下的引脚行为：

- 在 stop1 模式下，所有的内部寄存器包括并行 I/O 控制和数据寄存器被关闭。每个引脚恢复为其默认复位状态（输出缓冲区和内部上拉禁止）。在退出 stop1 模式后，若 MCU 已复位则所有引脚必须被配置为相同状态。
- stop2 模式是一种部分断电模式，在执行 STOP 指令之前即锁存保持该引脚状态。在进入 stop2 模式之前 CPU 寄存器状态和 I/O 寄存器状态必须被保存到 RAM 中。在退出 stop2 模式后，在访问任何 I/O 口之前，用户必须检查 SPMSC2 寄存器中 PPDF 位的状态。如果 PPDF 位为零，I/O 口必须像上电复位一样被初始化。如果 PPDF 位为 1，I/O 口数据先被保存到 RAM 中，在 STOP 指令执行之前，外设先被使能将要求被初始化和恢复到先前停止状态。用户那时必须将 SPMSC2 寄存器中的 PPDACK 位置 1。用户应用程序的引脚访问将再次被准许。
- 在 stop3 模式下，所有引脚状态被保持因为内部逻辑保持上电状态。恢复以后，所有引脚功能和进入 stop3 模式前一样。

6.4 并行 I/O 寄存器

6.4.1 A 口寄存器

这一节介绍了与并行 I/O 口相关的寄存器。涉及到并行 I/O 口绝对地址分配参见第 4 章“内存映像和寄存器定义”。这节涉及到寄存器和控制位的只通过它们的名字引用。飞思卡尔将提供一个头文件将这些名称翻译成对应的绝对地址。

6.4.1.1 A 口数据寄存器

	7	6	5	4	3	2	1	0
读	0	0	PTAD5 ¹	PTAD4 ²	PTAD3	PTAD2	PTAD1	PTAD0
写								
复位	0	0	0	0	0	0	0	0

¹读 PTAD5 将返回 PTA5 的值，不论 PTADD5 位的值是多少

²读 PTAD4 将返回 PTA5 的值，不论 PTADD4 位的值是多少

图 6-2 A 口数据寄存器（PTAD）

表 6-1 PTAD 寄存器字段描述

字段	描述
5:0 PTAD[5:0]	A 口数据寄存器位——若 A 口引脚为输入，读返回该引脚上的逻辑电平。若 A 口引脚为输出，读返回写给该寄存器的最后值。 写被锁存到寄存器的所有位。若 A 口引脚为输出，则逻辑电平被发送到 MCU 相应的引脚。

6.4.1.2A口数据方向寄存器（PTADD）

	7	6	5	4	3	2	1	0
读	0	0	PTAD5 ¹	PTAD4 ²	PTAD3	PTAD2	PTAD1	PTAD0
写								
复位	0	0	0	0	0	0	0	0

¹PTADD5对只输入引脚PTA5无影响。读该位一直为零。

²PTADD4对只输出引脚PTA4无影响。

图6-3 A口数据方向寄存器（PTADD）

表 6-2PTADD 寄存器字段描述

字段	描述
5:0 PTAD[5:0]	A口数据方向寄存器位——这些读写位控制A口引脚的方向和读PTAD的内容 0 输入（输出驱动禁止）和读返回引脚值。 1 A口位n的输出驱动允许且PTAD读返回PTAn的内容。

6.4.2A口控制寄存器

与A口相关的引脚由该寄存器控制。这些寄存器控制A口引脚的引脚上拉，回转速率和驱动强度，独立于并行I/O寄存器。

6.4.2.1A口内部上拉使能（PTAPE）

通过设置上拉使能寄存器（PTAPEn）中的相应位使能每个端口引脚的内部上拉设备。如果引脚通过并行I/O口控制逻辑或任何复用的外设功能（不论相应上拉使能寄存器位的状态如何）配置为输出，则上拉被禁止。如果引脚作为模拟功能则上拉设备也被禁止。

	7	6	5	4	3	2	1	0
读	0	0	PTAPE5 ¹	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
写								
复位	0	0	0	0	0	0	0	0

¹PTAPE4对只输出引脚PTA4无影响。

图6-4 A口寄存器内部上拉使能（PTAPE）

表 6-3PTAPE 寄存器字段描述

字段	描述
5:0 PTAD[5:0]	A口内部上拉使能位——这些控制位的每一位决定相关的PTA引脚内部上拉设备是否允许。若A口引脚配置为输出，这些位将无效同时内部上拉也被禁止。 0 A口位n的内部上拉设备禁止 1 A口位n的内部上拉设备允许

6.4.2.2A口回转速率使能（PTASE）

通过设置回转速率控制寄存器（PTASEn）中的相应位来使能每个端口引脚的回转速率控制。当允许时，回转控制限制输出跳转时的速率以降低 EMC 干扰。回转速率控制对配置为输入的引脚无影响。

	7	6	5	4	3	2	1	0
读	0	0	PTASE5 ¹	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
写								
复位	0	0	1	1	1	1	1	1

¹PTASE5对只输出引脚PTA5无影响。

图6-5 A口回转速率使能寄存器（PTASE）

表 6-4PTASE 寄存器字段描述

字段	描述
5:0 PTAD[5:0]	A口输出回转速率使能位——这些控制位的每一位决定相应PTA引脚输出回转速率控制是否允许。若A口引脚配置为输入，这些位将无效。 0 A口位n的输出回转速率控制禁止 1 A口位n的输出回转速率控制允许

6.4.2.3A口驱动强度选择（PTADS）

一个输出引脚可通过设置驱动强度选择寄存器（PTADSn）中的相应位来获得高输出驱动强度。当高驱动强度被选择时引脚能够承受更大电流。即使每个 I/O 引脚能被选择为高驱动强度，用户必须确保所有电流和不超过芯片的限制范围。驱动强度选择倾向于影响 I/O 引脚的交流和直流行为。高驱动强度允许引脚以相同的转换速度驱动一个更大负载。因为这样使能引脚为高强度驱动将影响 EMC 干扰。

6.4.2.4A口驱动强度选择（PTADS）

	7	6	5	4	3	2	1	0
读	0	0	PTADSS ¹	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
写								
复位	0	0	0	0	0	0	0	0

¹PTADS5对只输出引脚PTA5无影响。

图6-6A口驱动强度选择寄存器（PTADS）

表 6-5PTADS 寄存器字段描述

字段	描述
5:0 PTAD[5:0]	A口输出驱动强度选择位——这些控制位的每一位为相关PTA引脚选择低或高输出驱动。若A口引脚配置为输入，这些位敬爱那个无效。 0 A口位n选择低输出驱动 1 A口位n选择高输出驱动

第七章 中央处理单元（S08CPUV2）

7.1 简介

本章介绍了寄存器，地址模式，和 HCS08 系列的 CPU 的指令集。更多详细的讨论，参见 HCS08 系列参考手册，第 I 卷，Freescale 半导体公司（文件编号 HCS08RMV1/D）。

HCS08CPU 的源和目标代码都与 M68HC08CPU 兼容。新加入了几个指令和增强地址模式来提高 C 编译效率和支持一种新的背景调试系统，取代了早期 M68HC08 微控制器（MCU）的监控模式。

7.1.1 特征

HCS08CPU 特征包括：

- 目标代码完全向上兼容 M68HC05 和 M68HC08 系列
- 所有寄存器和存储器映射到一个单一的 64 KB 的地址空间
- 16 位堆栈指针（在 64 k 字节的任意地址空间内任意大小的栈）
- 16 位变址寄存器（H:X）带强大的变址地址模式
- 8 位累加器（A）
- 许多指令把 X 作为通用 8 位寄存器
- 7 种寻址模式：
 - 内在寻址方式——指令包含了操作数的所在之处
 - 相对变址寻址——8 位有符号偏移量寻址
 - 立即寻址方式——操作数在下一个目标代码字节中
 - 直接寻址方式——操作数在 0x0000-0x00FF 的存储器中
 - 扩展寻址方式——操作数在 64K 字节地址空间的任意地方
 - H:X 相对变址寻址方式——包含自加的五个子模式
 - SP 相对变址寻址方式——明显的提高 C 效率
- 内存-内存数据转移指令的四个寻址模式组合
- 溢出，半进位，负，零，和进位条件码支持有条件的分支上带符号，无符号，BCD 码操作数的结果
- 高效率的位操作指令
- 快速 8 位乘 8 位和 16 位除 8 位指令
- STOP 和 WAIT 指令调用低功耗运行模式

7.2 编程模型和CPU寄存器

图 7-1 描述了 5 个 CPU 寄存器。CPU 寄存器不是内存映射的一部分。

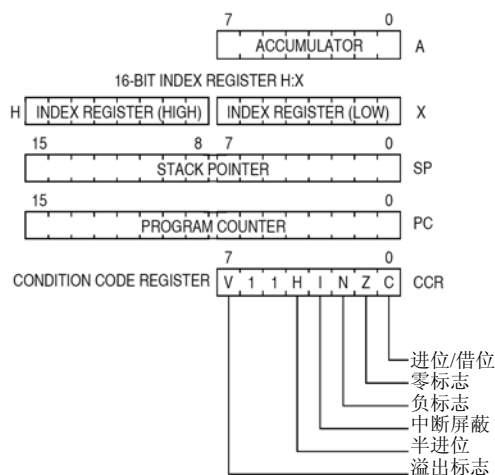


图7-1 CPU寄存器

7.2.1 累加器A (accumulator)

这个累加器(A)是通用的 8 位寄存器。一个操作数输入到算术逻辑单元 (ALU)，而 ALU 和累加器相连，在算术和逻辑运算后，ALU 结果往往存入 A 累加器。可以用不同的寻址模式指定需要载入数据的地址，累加器将值放入存储器，或者根据不同的寻址模式指定 A 要存入的地址，累加器 A 中的数据可以存储存储器中。

复位对累加器 A 中的数据无影响。

7.2.2 变址寄存器HX (index pointer)

这个 16 位寄存器实际上是两个 8 位寄存器 (H,X) 的组合，常作为一个 16 位地址指示器，H 装地址的高字节，X 装地址的低字节。所有变址寻址方式指令用 H:X 中的 16 位全值作为变址参考值。然而，为了和早期 M68HC05 系列兼容，一些指令只能在低 8 位 (X) 上运行。

许多指令把 X 作为通用 8 位寄存器用来存储 8 位数据。X 可以被清零，加，减，取反，移位，循环移位。转移指令允许在算术逻辑操作被执行时数据可以转移到或转移出累加器 A。

为了和早期 M68HC05 系列兼容，H 在复位时被设为 0X00。复位对 X 无影响。

7.2.3 堆栈指针SP (Stack Pointer)

在一个自动后进先出(LIFO)堆栈中这个 16 位地址指针寄存器指向下一个可用地址。堆栈可以映射到有 RAM 和规模最高限额为可用内存的 64K 字节的任意地址空间。堆栈可以自动保存子程序调用返回值，中断时返回地址值和 CPU 寄存器值，和局部变量。AIS (add immediate to stack pointer) 指令加 8 位有符号立即数给 SP。这是在栈中最常用的分配或回收本地堆栈变量的空间。

为了兼容早期的 M68HC05 系列 SP 复位后被设为 0XFF。在复位初始化释放直接页 RAM (从片上寄存器的最后地址到 0X00FF) 时，HCS08 程序通常将 SP 中的值改为在片上 RAM 中最后地址值 (高地址)。

复位堆栈指针 RSP (reset stack pointer) 指令和 M68HC05 系列兼容且很少用在 HCS08 程序中，因为它仅影响堆栈指针的低位地址。

7.2.4 程序计数器PC (program counter)

程序计数器是一个 16 位寄存器，包含下一条指令或待取操作数的地址。

程序在正常执行期间，程序计数器在每次数据或指令被取出时会自动增加到存储器的下一个连续位置。跳转，分支，中断和返回操作装载一个带地址而不是下个连续位置的程序计数器。这就是流程的改变。

复位时，程序计数器从 0XFFFE 和 0XFFFF 处装载复位向量。向量存储了将在退出复位状态后执行第一个指令的位置。

7.2.5 条件码寄存器CCR (condition code register)

8 位条件码寄存器包含中断屏蔽位 I 和五位状态标志位。第 6 位和第 5 位永远为逻辑 1。下图简要描述了 CCR 各位的信息及功能。详细的用指令设置 CCR 各位，参见 HCS08 系列参考手册，卷 I，Freescale 半导体文件编号 HCS08RMV1/D。

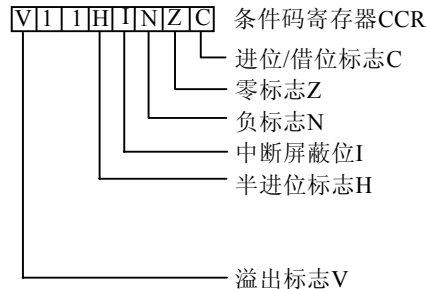


图7-2 条件码寄存器(CCR)

表 7-1 CCR 寄存器域描述符

位置	描述
7 V	溢出标志位—当二进制补码溢出发生，CPU将设置溢出标志位。有符号指令BGT、BGE、BLE、BLT使用溢出标志。 0 无溢出 1 溢出
4 H	半进位标志—执行加法指令（ADD）和带进位加法指令（ADC）时，如果累加器D3向D4有进位，CPU设置半进位标志。半进位标志在BCD码算术运算中很有用。DAA指令用H,C的状态来自动调整BCD值。
3 I	中断屏蔽位—当中断屏蔽位被设置时，禁止CPU中断。当中断屏蔽位被清除，开放CPU中断。当中断发生时，在CPU寄存器值被保存到堆栈后中断屏蔽被自动设置，但在第一个中断服务例程被执行之前。在任何指令正在清I（CLI、TAP）时，中断不被认可。这确保了CLI或TAP后的指令执行时不被干扰。 0 允许中断 1 禁止中断
2 N	负标志—CPU进行运算过程中，如果产生负结果则将负标志置1，设置位7的结果。如果装载或存储值的较多重要位为1的话，简单的装载或存储8位或16位值也会引起N置1。
1 Z	零标志—CPU进行运算过程中，如果数据或运算结果为零，零标志置1，否则置零。 0 无零结果 1 有零结果
0 C	进位/借位标志—当进行加法时，在最高位D7上有进位；或在进行减法运算时需要向更高位借位，则CPU将进位/借位标志C置1。一些指令如位测试，跳转，移位指令等也会影响该标志。 0 无进位/借位 1 有进位/借位

7.3 寻址方式

寻址方式定义了 CPU 存取数据和操作数的方法。在 HCS08 系列芯片中，所有的存储器，状态和控制寄存器，I/O 口共享一个单一的 64K 字节线性地址空间，所以一个 16 位二进制地址可以唯一确定一个存储位置。这样安排意味着存取 RAM 中变量的相同指令同时也能存取 I/O 和控制寄存器或非易失性程序空间。

一些指令有不只一种寻址方式。举例来说，MOVE 指令用一种寻址方式来指定源操作数和另一种寻址方式来确定目标地址。当假设条件为真时，一些指令如 BRCLR、BRSET、CBEQ、DBNZ，用一种寻址方式来确定一个操作数地址然后用相对寻址方式来确定转移目标地址。象 BRCLR、BRSET、CBEQ、DBNZ，列在指令集表上的寻址方式是一种存取操作数来测试，相对寻址方式隐含着转移目标地址的寻址方式。

7.3.1 内在寻址方式 (INH)

内在寻址方式，指令中已经包含了操作数所在之处，所以 CPU 不需要存取存储器中的操作数。

7.3.2 相对寻址方式 (REL)

相对寻址方式用于转移指令确定目的地址。8 位有符号的偏移值在操作码之后立即在存储器中被定位。在执行期间，如果转移条件为真，有符号偏移值扩展为 16 位有符号值且加到当前程序计数器内容中，导致程序继续在转移目标地址处执行。

7.3.3 立即寻址方式 (IMM)

立即寻址方式中，在指令操作码之后紧跟着需要完成目标代码中的指令的操作数。在 16 位立即操作数情况下，高位字节在操作码之后定位在写个存储位置，低位字节定位在其之后的下个存储位置。

7.3.4 直接寻址方式 (DIR)

直接寻址方式中，在直接页 (0X0000-0X00FF) 中指令包含地址的低 8 位。在运行期间，通过连接隐含的高位地址 0X00 和指令的直接地址得到一个 16 位地址以获取目标操作数。这比给操作数分配一个全 16 位地址要更快，存取效率更高。

7.3.5 扩展寻址方式 (EXT)

在扩展寻址方式中，操作数的 16 位地址被包含在操作码后的两字节目标码内 (高位在先)。

7.3.6 变址寻址方式

编制寻址方式包含七种，其中五种用到 16 位 H:X 变址寄存器，两种用到堆栈作基值参考。

7.3.6.1 无偏移量变址方式 (IX)

在这种变址方式中，变址寄存器 H:X 的内容被用于访问操作数的地址。

7.3.6.2 无偏移量变址、变址加1寻址方式(IX+)

在这种变址方式中，H:X 变址寄存器的内容用来访问目的操作数，然后 H:X 寄存器自增 1。CBEQ 和 MOV 指令是唯一使用这种变址方式的指令。

7.3.6.3 8位偏移量变址方式(IX1)

在这种变址方式中，一个无符号的 8 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址。

7.3.6.4 8位偏移量变址、变址加1寻址方式(IX1+)

在这种变址方式中，一个无符号的 8 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址。在操作数被取出后编制寄存器增 1。CBEQ 是唯一使用这种寻址方式的指令。

7.3.6.5 16位偏移量变址方式(IX2)

在这种变址方式中，一个无符号的 16 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址。

7.3.6.6 8位偏移量堆栈寻址方式(SP1)

在这种变址方式中，指令集提供一个无符号的 8 位偏移量与堆栈指针(SP)中的 16 位值相加，得到要访问的操作数地址。

7.3.6.7 16位偏移量堆栈寻址方式(SP2)

在这种变址方式中，无符号的 16 位偏移量与堆栈指针(SP)中的 16 位值相加，得到要访问的操作数地址。

7.4 特殊操作

CPU 中描述了一些和指令相似的特殊操作，但是不像其他 CPU 指令一样有操作码。另外一些指令如 STOP，WAIT 直接影响其他 MCU 的电路。这部分提供这些操作的额外信息。

7.4.1 复位序列

复位可以由上电复位、COP 看门狗超时、外部复位引脚的低电平引起。当复位事件发生，复位事件强制 CPU 立即停止正在执行的任何事件(在响应复位事件前，MCU 不等待指令边界)。想要了解更详细的关于 MCU 如何确认复位及其来源的信息，可参阅复位，中断和系统配置章节。

当序列确定复位源是否来自内部和复位引脚不再低电平时复位事件被认为结束。由一个复位事件可知，CPU 需执行一个 6 周期的序列来从 0XFFFE，0XFFFF 处取出复位向量和在准备执行第一个程序指令时填满指令队列。

7.4.2 中断序列

当中断请求发生时，在响应中断之前 CPU 先完成当前指令。这时程序计数器指向下一条指令的开始位置，即 CPU 中断返回地址。CPU 对中断的响应是执行相同的操作序列的，软中断(SWI)指令也是一样，除非当中断序列开始时，由高优先级中断决定的用于向量获取的址是未定的。

CPU 中断顺序如下：

- 1.按序存储 PCL、PCH、X、A、CCR 的内容到堆栈中。
- 2.置位 CCR 中的 I 位。

3. 获取中断向量高位。
4. 获取中断向量低位。
5. 延迟一个空闲总线周期。
6. 获取程序信息在中断向量所标识的地址的开始的 3 个字节。

CCR 中的内容被压入堆栈后，在进行中断服务例程时置位 CCR 的 I 位防止其它中断的干扰。虽然也可以通过指令在中断服务例程中清 I，但是可能致嵌套中断发生（这是不推荐的，因为它导致程序难以调试和维护）。

为了和早期的 M68HC05MCU 兼容，H:X 编制寄存器中的 H 不作为中断序列的一部分保存到堆栈中。用户必须在终端服务例程开始时用 PSHH 指令保存 H 内容，然后在 RTI 返回之前用 PULH 指令结束中断服务例程。如果你确定中断服务例程不用任何指令或自增寻址方式来改变 H 的值，H 值也可以不被保存。

除了软中断不能被屏蔽外，其它和硬中断一样，它是和程序的指令操作码相联系的，所以它不是异步程序执行。

7.4.3 等待模式操作

WAIT 指令通过清零 CCR 的 I 位是使能中断。然后关闭 CPU 时钟以节省功耗，CPU 维持在低功耗状态下直到中断或复位唤醒它。当中断或复位事件发生时，CPU 从等待模式被唤醒，中断或复位事件将被正常处理。

当 CPU 在等待模式中如果通过背景调试接口发出连续的 BACKGROUND 命令到 MCU，CPU 时钟将被开启并进入有效背景模式，在这种模式下其它的背景指令能被处理。这样确保即使在等待模式中主机开发系统仍能访问目标 MCU。

7.4.4 停止模式操作

通常在停止模式下，系统的所有时钟包括晶振（当使用时）都被关闭以使功耗最小化。在这样的系统中，外部电路需要控制时间，当继续处理进程时，确保有信号唤醒目标 MCU。不像早期的 M68HC05 和 M68HC08，HCS08 在停止模式中可以设定运行最小的一部分时钟。这就允许一个内部周期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）且通过背景调试接口（或因为 MCU 复位进入有效背景模式）的连续命令置位 ENBDM 位时，在 MCU 进入停止模式时振荡器被迫保持有效。在这种情况下，当 CPU 在停止模式中如果后台命令通过背景调试接口流入到 MCU 中，CPU 时钟将被开启并 CPU 进入有效背景模式，在这种模式下其它背景指令能被处理。这样确保即使在等待模式中主机开发系统仍能访问目标 MCU。

跳出停止模式依靠特殊的 HCS08 和停止模式中振荡器是否被停止，详细参阅操作模式章节。

7.4.5 BGND 指令

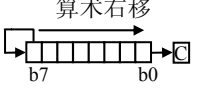
与 M68HC08 相比，对 HCS08 来说背景调试指令是一个新指令。BGND 不会使用在普通用户程序中，因为它强迫 CPU 停止处理用户指令并进入有效背景模式。重新继续运行用户程序的唯一方式是通过复位或主机调试系统的背景调试接口发出 GO、TRACE1、TAGGO 串行命令。

软断点可以通过用 BGND 操作码替换目标断点地址操作码来设置。当程序到达断点地址时，CPU 更可能进入有效背景模式而不是继续用户程序。

7.5 HCS08指令集

表 7-2 提供了 HCS08 指令集各种可能寻址模式的概要。表中描述了操作数的构成，执行时间（内部总线周期），和每个指令的各种寻址方式的每个周期的细节。

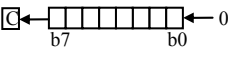
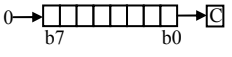
表 7-2 指令集概述

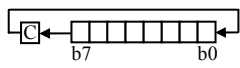
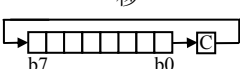
指令助记符	操作	寻址方式	条件						操作码	操作数	周期细节	周期
			V	H	I	N	Z	C				
ADC #opr8i	带进位加 $A \leftarrow (A) + (M) + (C)$	IMM							A9	ii	pp	2
ADC opr8a		DIR							B9	dd	rpp	3
ADC opr16a		EXT							C9	hh ll	prpp	4
ADC oprx16,X		IX2	√	√	-	√	√	√	D9	ee ff	prpp	4
ADC oprx8,X		IX1							E9	ff	rpp	3
ADC ,X		IX							F9		rff	3
ADC oprx16,SP		SP2							9ED9	ee ff	prpp	5
ADC oprx8,SP		SP1							9EE9	ff	prpp	4
ADD #opr8i	不带进位加 $A \leftarrow (A) + (M)$	IMM							AB	ii	pp	2
ADD opr8a		DIR							BB	dd	rpp	3
ADD opr16a		EXT							CB	hh ll	prpp	4
ADD oprx16,X		IX2	√	√	-	√	√	√	DB	ee ff	prpp	4
ADD oprx8,X		IX1							EB	ff	rpp	3
ADD ,X		IX							FB		rff	3
ADD oprx16,SP		SP2							9EDB	ee ff	prpp	5
ADD oprx8,SP		SP1							9EEB	ff	prpp	4
AIS #opr8i	有符号立即数加到堆栈指针 $SP \leftarrow (SP) + (M)$	IMM	-	-	-	-	-	-	A7	ii	pp	2
AIX #opr8i	有符号立即数加到变址寄存器 $H:X \leftarrow (H:X) + (M)$	IMM	-	-	-	-	-	-	AF	ii	pp	2
AND #opr8i	逻辑“与” $A \leftarrow (A) \text{AND} (M)$	IMM							A4	ii	pp	2
AND opr8a		DIR							B4	dd	rpp	3
AND opr16a		EXT							C4	hh ll	prpp	4
AND oprx16,X		IX2	0	-	-	√	√	-	D4	ee ff	prpp	4
AND oprx8,X		IX1							E4	ff	rpp	3
AND ,X		IX							F4		rff	3
AND oprx16,SP		SP2							9ED4	ee ff	prpp	5
AND oprx8,SP		SP1							9EE4	ff	prpp	4
ASL opr8a		DIR							38	dd	rffwpp	4
ASLA		INH							48		p	1
ASLX		INH	√	-	-	√	√	√	58		p	1
ASL oprx8,X		IX1							68	ff	rffwpp	4
ASL ,X		IX							78		rffw	3
ASL oprx8,SP		SP1							9E68	ff	prffwpp	5
ASR opr		DIR							37	dd	rffwpp	4
ASRA		INH							47		p	1
ASRX		INH	√	-	-	√	√	√	57		p	1
ASR opr,X		IX1							67	ff	rffwpp	4
ASR opr,X		IX							77		rffw	3
ASR opr,SP		SP1							9E67	ff	prffwpp	5
BCC rel	进位标志为0则转移	REL	-	-	-	-	-	-	24	rr	ppp	3

BCLR n,opr8a	清存储器第n位	DIR(b0)							11	dd	rfwpp	5
		DIR(b1)							13	dd	rfwpp	5
		DIR(b2)							15	dd	rfwpp	5
		DIR(b3)	-	-	-	-	-	-	17	dd	rfwpp	5
		DIR(b4)							19	dd	rfwpp	5
		DIR(b5)							1B	dd	rfwpp	5
		DIR(b6)							1D	dd	rfwpp	5
		DIR(b7)							1F	dd	rfwpp	5
BCS rel	进位标志为1则转移	REL	-	-	-	-	-	-	25	rr	ppp	3
BEQ rel	相等则转移	REL	-	-	-	-	-	-	27	rr	ppp	3
BGE opr	大于或等于则转移(有符号数)	REL	-	-	-	-	-	-	90	rr	ppp	3
BGND	ENBDM=1进入有效背景,等待和处理BDM命令直到GO,TRACE1,或TAGGO	INH	-	-	-	-	-	-	82		fp..ppp	5+
BGT rel	大于则转移(有符号数)	REL	-	-	-	-	-	-	92	rr	ppp	3
BHCC rel	半进位标志为0则转移	REL	-	-	-	-	-	-	28	rr	ppp	3
BHCS rel	半进位标志为1则转移	REL							29	rr	ppp	3
BHI rel	大于则转移	REL							22	rr	ppp	3
BHS rel	大于或等于则转移	REL							24	rr	ppp	3
BIH rel	$\overline{\text{IRQ}}$ 引脚为高则转移	REL							2F	rr	ppp	3
BIL rel	$\overline{\text{IRQ}}$ 引脚为低则转移	REL							2E	rr	ppp	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)与(M) CCR改变但操作数不变	IMM DIR EXT IX2 IX1 IX SP2 SP1	0	-	-	√	√	-	A5 B5 C5 D5 E5 F5 9ED5 9EE5	ii dd hh ll ee ff ff ff ee ff ff	pp rpp prpp prpp rpp rfp prpp prpp	2 3 4 4 3 3 5 4
BLE opr	小于或等于则转移(如果 $Z \oplus V=1$)(有符号数)	REL	-	-	-	-	-	-	93	rr	ppp	3
BLO rel	小于则转移(如果C=1)(和BCS相同)	REL	-	-	-	-	-	-	25	rr	ppp	3
BLS rel	小于等于则转移(如果 $C Z=1$)	REL	-	-	-	-	-	-	23	rr	ppp	3
BLT opr	小于则转移(如果 $N \oplus V=1$)(有符号数)	REL	-	-	-	-	-	-	91	rr	ppp	3
BMC rel	中断屏蔽位为0(如果I=0)则转移	REL	-	-	-	-	-	-	2C	rr	ppp	3
BMI rel	负值则转移(如果N=1)	REL	-	-	-	-	-	-	2B	rr	ppp	3
BMS rel	中断屏蔽位为1(如果I=1)则转移	REL	-	-	-	-	-	-	2D	rr	ppp	3
BNE rel	不等于则转移(如果Z=0)	REL	-	-	-	-	-	-	26	rr	ppp	3
BPL rel	正值则转移(如果N=0)	REL	-	-	-	-	-	-	2A	rr	ppp	3

BRA rel	无条件转移(如果I=1)	REL	-	-	-	-	-	-	20	rr	ppp	3
BRCLR n,opr8a,rel	存储器第n位为0 (如果 (Mn)=0) 则转移	DIR(b0)							01	dd rr	rpppp	5
		DIR(b1)							03	dd rr	rpppp	5
		DIR(b2)							05	dd rr	rpppp	5
		DIR(b3)	-	-	-	-	-	√	07	dd rr	rpppp	5
		DIR(b4)							09	dd rr	rpppp	5
		DIR(b5)							0B	dd rr	rpppp	5
		DIR(b6)							0D	dd rr	rpppp	5
		DIR(b7)							0F	dd rr	rpppp	5
BRN rel	永不转移(如果I=0)	REL	-	-	-	-	-	-	21	rr	ppp	3
BRSET n,opr8a,rel	存储器第n位为1 (如果 (Mn)=1) 则转移	DIR(b0)							00	dd rr	rpppp	5
		DIR(b1)							02	dd rr	rpppp	5
		DIR(b2)							04	dd rr	rpppp	5
		DIR(b3)	-	-	-	-	-	√	06	dd rr	rpppp	5
		DIR(b4)							08	dd rr	rpppp	5
		DIR(b5)							0A	dd rr	rpppp	5
		DIR(b6)							0C	dd rr	rpppp	5
		DIR(b7)							0E	dd rr	rpppp	5
BSET n,opr8a	置存储器第n位为1 (Mn ←1)	DIR(b0)							10	dd	rffwppp	5
		DIR(b1)							12	dd	rffwppp	5
		DIR(b2)							14	dd	rffwppp	5
		DIR(b3)	-	-	-	-	-	-	16	dd	rffwppp	5
		DIR(b4)							18	dd	rffwppp	5
		DIR(b5)							1A	dd	rffwppp	5
		DIR(b6)							1C	dd	rffwppp	5
		DIR(b7)							1E	dd	rffwppp	5
BSR rel	跳至子程序 PC←(PC)+\$0002 Push(PSL):SP← (SP)-\$0001 push(PCH):SP← (SP)-\$0001 PC←(PC)+rel	REL	-	-	-	-	-	-	AD	rr	ssppp	5
CBEQ opr8a,rel	比较若(A)=(M)	DIR							31	dd rr	rpppp	5
CBEQA #opr8i,rel	(A)=(M)	IMM							41	ii rr	pppp	4
CBEQX #opr8i,rel	(X)=(M)	IMM	-	-	-	-	-	-	51	ii rr	pppp	4
CBEQ oprx8,X+,rel	(A)=(M)	IX1+							61	ff rr	rpppp	5
CBEQ X+,rel	(A)=(M)	IX+							71	rr	rfppp	5
CBEQ oprx8,SP,rel	(A)=(M)则转移	SP1							9E61	ff rr	prpppp	6
CLC	清进位标志(C←0)	INH	-	-	-	-	-	0	98		p	1
CLI	清中断屏蔽 标志(I←0)	INH	-	-	0	-	-	-	9A		p	1
CLR opr8a	清零指令M←\$00	DIR							3F	dd	rfwpp	5
CLRA	A←\$00	INH							4F		p	1
CLR X	X←\$00	INH							5F		p	1
CLR H	H←\$00	INH	0	-	-	0	1	-	8C		p	1
CLR oprx8,X	M←\$00	IX1							6F	ff	rfwpp	5
CLR ,X	M←\$00	IX							7F		rfwp	4
CLR oprx8,SP	M←\$00	SP1							9E6F	ff	prfwpp	6

CMP #opr8i		IMM						A1	ii	pp	2
CMP opr8a		DIR						B1	dd	rpp	3
CMP opr16a		EXT						C1	hh ll	prpp	4
CMP oprx16,X	比较累加器与存储器 A-M(CCR改变但操作数 不变)	IX2	√	-	-	√	√	D1	ee ff	prpp	4
CMP oprx8,X		IX1						E1	ff	rpp	3
CMP ,X		IX						F1		rfp	3
CMP oprx16,SP		SP2						9ED1	ee ff	pprpp	5
CMP oprx8,SP		SP1						9EE1	ff	prpp	4
COM opr8a	求反 $M \leftarrow \overline{M} = \$FF - (M)$	DIR						33	dd	rhwpp	5
COMA	$A \leftarrow \overline{A} = \$FF - (M)$	INH						43		p	1
COMX	$X \leftarrow \overline{X} = \$FF - (M)$	INH	0	-	-	√	√	53		p	1
COM oprx8,SP	$M \leftarrow \overline{M} = \$FF - (M)$	IX1						63	ff	rhwpp	5
COM ,X	$M \leftarrow \overline{M} = \$FF - (M)$	IX						73		rwp	4
COM oprx8,SP	$M \leftarrow \overline{M} = \$FF - (M)$	SP1						9E63	ff	prhwpp	6
CPHX opr16a	比较变址寄存器与存储器 (H:X)-(M:M+1)(CCR 改变但操作数不变)	EXT						3E	hh ll	prfpp	6
CPHX #opr16i		IMM	√	-	-	√	√	65	ii kk	ppp	3
CPHX opr8a		DIR						75	dd	rrfpp	5
CPHX oprx8,SP		SP1						9EF3	ff	prfpp	6
CPX #opr8i	比较X寄存器与存储器 (X)-(M) (CCR改变但操 作数不变)	IMM						A3	ii	pp	2
CPX opr8a		DIR						B3	dd	rpp	3
CPX opr16a		EXT						C3	hh ll	prpp	4
CPX oprx16,X		IX2	√	-	-	√	√	D3	ee ff	prpp	4
CPX oprx8,X		IX1						E3	ff	rpp	3
CPX ,X		IX						F3		rfp	3
CPX opr16,SP		SP2						9ED3	ee ff	pprpp	5
CPX opr8,SP		SP1						9EE3	ff	prpp	4
DAA	在BCD值的ADD或ADC 后十进位转换	INH	U	-	-	√	√	72		p	1
DBNZ opr8a,rel	减去1后结果不为0则转 移 DBNZX影响X不影响H	DIR						3B	dd rr	rhwpppp	7
DBNZA rel		INH						4B	rr	fppp	4
DBNZX rel		INH	-	-	-	-	-	5B	rr	fppp	4
DBNZ oprx8,X,rel		IX1						6B	ff rr	rhwpppp	7
DBNZ X,rel		IX						7B	rr	rhwppp	6
DBNZ oprx8,SP,rel		SP1						9E6B	ff rr	prhwppp	8
DEC opr8a	减1指令 $M \leftarrow (M) - 1$	DIR						3A	dd	rhwpp	5
DECA	$A \leftarrow (A) - 1$	INH						4A		p	1
DECX	$X \leftarrow (X) - 1$	INH	√	-	-	√	√	5A		p	1
DEC oprx8,X	$M \leftarrow (M) - 1$	IX1						6A	ff	rhwpp	5
DEC ,X	$M \leftarrow (M) - 1$	IX						7A		rwp	4
DEC oprx8,SP	$M \leftarrow (M) - 1$	SP1						9E6A	ff	prhwpp	6
DIV	除法指令 $A \leftarrow (H:A)/(X)$ $H \leftarrow$ 余数	INH	-	-	-	-	√	52		ffffp	6
EOR #opr8i	存储器与累加器进行“异 或” $A \leftarrow (A \oplus M)$	IMM						A8	ii	pp	2
EOR opr8a		DIR						B8	dd	rpp	3
EOR opr16a		EXT						C8	hh ll	prpp	4
EOR oprx16,X		IX2	0	-	-	√	√	D8	ee ff	prpp	4
EOR oprx8,X		IX1						E8	ff	rpp	3
EOR ,X		IX						F8		rfp	3
EOR oprx16,SP		SP1						9ED8	ee ff	pprpp	5
EOR oprx8,SP		SP2						9EE8	ff	prpp	4

INC opr8a	加1指令 $M \leftarrow (M)+1$	DIR							3C	dd	rffwpp	5
INCA	$A \leftarrow (A)+1$	INH							4C		p	1
INCX	$X \leftarrow (X)+1$	INH	√	-	-	√	√	-	5C		p	1
INC oprx8,X	$M \leftarrow (M)+1$	IX1							6C	ff	rffwpp	5
INC ,X	$M \leftarrow (M)+1$	IX							7C		rffw	4
INC oprx8,SP	$M \leftarrow (M)+1$	SP1							9E6C	ff	prffwpp	6
JMP opr8a	跳转指令 $PC \leftarrow$ 跳转地址	DIR							BC	dd	ppp	3
JMP opr16a		EXT							CC	hh ll	pppp	4
JMP oprx8,X		IX2	-	-	-	-	-	-	DC	ee ff	pppp	3
JMP oprx16,X		IX1							EC	ff	ppp	3
JMP ,X		IX							FC		ppp	3
JSR opr8a	跳至子程序 $PC \leftarrow (PC)+n(N=1,2, \text{or } 3)$ Push(PCL); $SP \leftarrow (SP)-1$ Push(PCH); $SP \leftarrow (SP)-1$ $PC \leftarrow$ 子程序入口	DIR							BD	dd	ssppp	5
JSR opr16a		EXT							CD	hh ll	pssppp	6
JSR opr16,X		IX2	-	-	-	-	-	-	DD	ee ff	pssppp	5
JSR oprx8,X		IX1							ED	ff	ssppp	5
JSR ,X		IX							FD		ssppp	5
LDA #opr	将存储器数值送到累加器 $A \leftarrow (M)$	IMM							A6	ii	pp	2
LDA opr8i		DIR							B6	dd	rpp	3
LDA opr8a		EXT							C6	hh ll	prpp	4
LDA oprx16,X		IX2	0	-	-	√	√	-	D6	ee ff	prpp	4
LDA oprx8,X		IX1							E6	ff	rpp	3
LDA ,X		IX							F6		rff	3
LDA oprx16,SP		SP2							9ED6	ee ff	pprpp	5
LDA oprx8,SP		SP1							9EE6	ff	prpp	4
LDHX #opr16i	将存储器数值送到变址寄存器 $H:X \leftarrow (M:M+1)$	IMM							45	ii jj	ppp	3
LDHX opr8a		DIR							55	dd	rrpp	4
LDHX opr16a		EXT							32	hh ll	prpp	5
LDHX ,X		IX	0	-	-	√	√	-	9EAE		prff	5
LDHX oprx16,X		IX2							9EBE	ee ff	pprrpp	6
LDHX oprx8,X		IX1							9ECE	ff	prpp	5
LDHX oprx8,SP		SP1							9EFE	ff	prpp	5
LDX #opr8i	将存储器数值送到X寄存器 $X \leftarrow (M)$	IMM							AE	ii	pp	2
LDX opr8a		DIR							BE	dd	rpp	3
LDX opr16a		EXT							CE	hh ll	prpp	4
LDX oprx16,X		IX2	0	-	-	√	√	-	DE	ee ff	prpp	4
LDX oprx8,X		IX1							EE	ff	rpp	3
LDX ,X		IX							FE		rff	3
LDX oprx16,SP		SP2							9EDE	ee ff	pprpp	5
LDX oprx8,SP		SP1							9EEE	ff	prpp	4
LSL opr8a	逻辑左移 	DIR							38	dd	rffwpp	5
LSLA		INH							48		p	1
LSLX		INH	√	-	-	√	√	√	58		p	1
LSL oprx8,A		IX1							68	ff	rffwpp	5
LSL ,X		IX							78		rffw	4
LSL oprx8,SP		SP1							9E68	ff	prffwpp	6
LSR opr	逻辑右移 	DIR							34	dd	rffwpp	5
LSRA		INH							44		p	1
LSRX		INH	√	-	-	0	√	√	54		p	1
LSR opr,X		IX1							64	ff	rffwpp	5
LSR ,X		IX							74		rffw	4
LSR opr,SP		SP1							9E64	ff	prffwpp	6
MOV opr8a,opr8a	移动指令 (M)目的 $\leftarrow$ (M)源	DIR/DIR							4E	dd dd	rpwpp	5
MOV opr8a,X+		DIR/IX+	0	-	-	√	√	-	5E	dd	rffwpp	4
MOV #opr8i,opr8a		IMD/DIR							6E	ii dd	pwpp	4
MOV X+,opr8a		IX+/DIR							7E	dd	rffwpp	4

MUL	无符号数乘法指令 $X:A \leftarrow (X) \times (A)$	INH	-	0	-	-	-	0	42		ffffp	5
NEG opr8a	求补 $M \leftarrow -(M) = \$00-(M)$	DIR							30	dd	rffwpp	5
NEGA	$A \leftarrow -(A) = \$00-(A)$	INH							40		p	1
NEGX	$X \leftarrow -(X) = \$00-(X)$	INH							50		p	1
NEG oprx8,X	$M \leftarrow -(M) = \$00-(M)$	IX1	√	-	-	√	√	√	60	ff	rffwpp	5
NEG ,X	$M \leftarrow -(M) = \$00-(M)$	IX							70		rffw	4
NEG oprx8,SP	$M \leftarrow -(M) = \$00-(M)$	SP1							9E60	ff	prffwpp	6
NOP	空操作	INH	-	-	-	-	-	-	9D		p	1
NSA	累加器半字节交换 $A \leftarrow (A[3:0]:A[7:4])$	INH	-	-	-	-	-	-	62		p	1
ORA #opr8i	累加器与存储器“或” $A \leftarrow (A)$ 和(M)或运算结果	IMM							AA	ii	pp	2
ORA opr8a		DIR							BA	dd	rpp	3
ORA opr16a		EXT							CA	hh ll	prpp	4
ORA oprx16,X		IX2	0	-	-	√	√	-	DA	ee ff	prpp	4
ORA oprx8,X		IX1							EA	ff	rpp	3
ORA ,X		IX							FA		rff	3
ORA oprx16,SP		SP2							9EDA	ee ff	pprpp	5
ORA oprx8,SP		SP1							9EEA	ff	prpp	4
PSHA	累加器内容进栈 $Push(A); SP \leftarrow (SP)-1$	INH	-	-	-	-	-	-	87		sp	2
PSHH	H寄存器内容进栈 $push(H); SP \leftarrow (SP)-1$	INH	-	-	-	-	-	-	8B		sp	2
PSHX	X寄存器内容进栈 $push(X); SP \leftarrow (SP)-1$	INH	-	-	-	-	-	-	89		sp	2
PULA	累加器内容出栈 $SP \leftarrow (SP)+1; pull(A)$	INH	-	-	-	-	-	-	86		ufp	3
PULH	H寄存器内容出栈 $SP \leftarrow (SP)+1; pull(H)$	INH	-	-	-	-	-	-	8A		ufp	3
PULX	X寄存器内容出栈 $SP \leftarrow (SP)+1; pull(X)$	INH	-	-	-	-	-	-	88		ufp	3
ROL opr8a	包括进位标志的循环左移 	DIR							39	dd	rffwpp	5
ROLA		INH							49		p	1
ROLX		INH							59		p	1
ROL oprx8,X		IX1	√	-	-	√	√	√	69	ff	rffwpp	5
ROL ,X		IX							79		rffw	4
ROL oprx8,SP		SP1							9E69	ff	prffwpp	6
ROR opr8a	包括进位标志的循环右移 	DIR							36	dd	rffwpp	5
RORA		INH							46		p	1
RORX		INH							56		p	1
ROR oprx8,X		IX1	√	-	-	√	√	√	66	ff	rffwpp	5
ROR ,X		IX							76		rffw	4
ROR oprx8,SP		SP1							9E66	ff	prffwpp	6
RSP	堆栈指针复位)(低字节) $SPL \leftarrow \$FF$ (高字节不受影响)	INH	-	-	-	-	-	-	9C		P	1
RTI	中断返回 $SP \leftarrow (SP)+1; pull(CCR)$ $SP \leftarrow (SP)+1; pull(A)$ $SP \leftarrow (SP)+1; pull(X)$ $SP \leftarrow (SP)+1; pull(PCH)$ $SP \leftarrow (SP)+1; pull(PCL)$	INH	√	√	√	√	√	√	80		uuuuufpp	9
RTS	子程序返回 $SP \leftarrow (SP)+1; pull(PCH)$ $SP \leftarrow (SP)+1; pull(PCL)$	INH	-	-	-	-	-	-	81		ufppp	5

SBC #opr	包括进位标志的减法指令 $A \leftarrow (A) - (M) - (C)$	IMM							A2	ii	pp	2
SBC opr		DIR							B2	dd	rpp	3
SBC opr		EXT							C2	hh ll	prpp	4
SBC opr,X		IX2	√	-	-	√	√	√	D2	ee ff	prpp	4
SBC opr,X		IX1							E2	ff	rpp	3
SBC ,X		IX							F2		rff	3
SBC opr,SP		SP1							9EE2	ff	pprpp	4
SBC opr,SP		SP2							9ED2	ee ff	prpp	5
SEC	进位标志置1 $C \leftarrow 1$	INH	-	-	-	-	-	1	99		p	1
SEI	中断屏蔽位置1 $I \leftarrow 1$	INH	-	-	1	-	-	-	9B		p	1
STA opr8a	累加器内容送到存储器 $M \leftarrow (A)$	DIR							B7	dd	wpp	3
STA opr16a		EXT							C7	hh ll	pwpp	4
STA oprx16,X		IX2							D7	ee ff	pwpp	4
STA oprx8,X		IX1	0	-	-	√	√	-	E7	ff	wpp	3
STA ,X		IX							F7		wp	2
STA oprx16,SP		SP2							9ED7	ee ff	ppwpp	5
STA oprx8,SP		SP1							9EE7	ff	pwpp	4
STHX opr8a	变址寄存器内容送到目标地址 $(M:M+1) \leftarrow (H:X)$	DIR	0	-	-	√	√	-	35	dd	wwpp	4
STHX opr16a									96	hh ll	pwpp	5
STHX oprx8,SP									9EFF	ff	pwpp	5
STOP	允许 $\overline{IRQ}$ 引脚中断, $I \leftarrow 1$; 停止晶振	INH	-	-	0	-	-	-	8E		rp...	2+
STX opr8a	X寄存器内容送到存储器 $M \leftarrow (X)$	DIR							BF	dd	wpp	3
STX opr16a		EXT							CF	hh ll	pwpp	4
STX oprx16,X		IX2							DF	ee ff	pwpp	4
STX oprx8,X		IX1	0	-	-	√	√	-	EF	ff	wpp	3
STX ,X		IX							FF		wp	2
STX oprx16,SP		SP2							9EDF	ee ff	ppwpp	5
STX oprx8,SP		SP1							9EEF	ff	pwpp	4
SUB #opr8i	减法指令 $A \leftarrow (A) - (M)$	IMM							A0	ii	pp	2
SUB opr8a		DIR							B0	dd	rpp	3
SUB opr16a		EXT							C0	hh ll	prpp	4
SUB oprx16,X		IX2	√	-	-	√	√	√	D0	ee ff	prpp	4
SUB oprx8,X		IX1							E0	ff	rpp	3
SUB ,X		IX							F0		rff	3
SUB oprx16,SP		SP2							9ED0	ee ff	pprpp	5
SUB oprx8,SP		SP1							9EE0	ff	prpp	4
SWI	软中断指令 $PC \leftarrow (PC) + 1$; push(PCL) $SP \leftarrow (SP) - 1$; push(PCH) $SP \leftarrow (SP) - 1$; push(X) $SP \leftarrow (SP) - 1$; push(A) $SP \leftarrow (SP) - 1$; push(CCR) $SP \leftarrow (SP) - 1$; $I \leftarrow 1$ $PCH \leftarrow$ 中断矢量高字节 $PCL \leftarrow$ 中断矢量低字节	INH	-	-	1	-	-	-	83		sssssvvf ppp	11
TAP	累加器内容送到条件码寄存器 $CCR \leftarrow (A)$	INH	√	√	√	√	√	√	84		p	1
TAX	累加器内容送到X寄存器 $X \leftarrow (A)$	INH	-	-	-	-	-	-	97		p	1
TPA	条件码寄存器内容送到累加器 $A \leftarrow (CCR)$	INH	-	-	-	-	-	-	85		p	1

TST opr8a	符号及零标志的测试 (M)-\$00	DIR							3D	dd	rfpp	4
TSTA	(A)-\$00	INH							4D		p	1
TSTX	(X)-\$00	INH	0	-	-	√	√	-	5D		p	1
TST oprx8,X	(M)-\$00	IX1							6D	ff	rfpp	4
TST ,X	(M)-\$00	IX							7D		rfp	3
TST oprx8,SP	(M)-\$00	SP1							9E6D	ff	prfpp	5
TSX	堆栈指针送到变址寄存器 $H:X \leftarrow (SP)+1$	INH	-	-	-	-	-	-	95		fp	2
TXA	X寄存器内容送到累加器 $A \leftarrow (X)$	INH	-	-	-	-	-	-	9F		p	1
TXS	变址寄存器内容送到堆栈指针 $(SP) \leftarrow (H:X)-1$	INH	-	-	-	-	-	-	94		fp	2
WAIT	允许中断, I bit $\leftarrow 0$ 停止处理器		-	-	0	-	-	-	INH	8F	fp...	2+

助记符：在助记符一列的信息必须出现在汇编源文件中，词首的 3-5 个字符和 (#, () 和 +) 有助于记忆。

n	标志或表达式用 0-7 的整型数来表示
opr8i	标志或表达式用 8 位的立即数来表示
opr16i	标志或表达式用 16 位的立即数来表示
opr8a	标志或表达式用 8 位的直接页地址(\$00xx)来表示
opr16a	标志或表达式用 16 位地址(\$00xx)来表示
opr8	标志或表达式用无标志 8 位值来表示，通常用做变址寻址
opr16	标志或表达式用 16 位值来表示，通常用做变址寻址
rel	标志或表达式用做下一条指令开始的-128 到+127 的地址

操作标志：

A	累加器
CCR	条件码寄存器
H	编制寄存器高字节
M	内存地址
n	任意位
opr	操作数(1 或 2 个字节)
PC	程序计数器
PCH	程序计数器高字节
PCL	程序计数器低字节
rel	程序相对偏移量
SP	堆栈指针
SPL	堆栈指针低字节
X	变址寄存器低字节
&	逻辑与

地址模式：

DIR	直接寻址方式
EXT	扩展寻址方式
IMM	立即寻址方式
INH	内在寻址方式
IX	无偏移量变址方式
IX1	8 位偏移量变址方式
IX2	16 位偏移量变址方式
IX+	无偏移量、变址加一寻址方式
IX1+	8 位无偏移量、变址加一寻址
REL	相对地址寻址
SP1	8 位偏移量堆栈寻址
SP2	16 位偏移量堆栈寻址

各周期编码：

f	空闲周期。标志着 CPU 在该周期不需要使用系统总线。f 周期总是一个系统总线周期，总是读周期。
p	程序取值。从可编程内存的下一个连续的地址读。
r	读 8 位的操作数

()	的内容	s	压栈(写)一个字节到栈中
+	加	u	从栈中出栈(读)一个字节
-	减, 负(反码)	v	从\$FFxx(先高字节)读向量
×	乘	w	写 8 位操作数
÷	除以		CCR 影响:
#	立即数	✓	置位或清位
←	载入	-	无影响
:	连接	U	未定义
	CCR 位:		
V	溢出标志		
H	半进位标志		
I	中断屏蔽位		
N	负标志		
Z	零标志		
C	进位/借位标志		

表 7-3 操作码映像 (1/2)

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

INH	内在	REL	相对	SP1	8位偏移量堆栈
IMM	立即	IX	In无偏移量变址	SP2	16偏移量堆栈
DIR	直接	IX1	8位偏移量变址	IX+	无偏移量变址加一
EXT	扩展	IX2	16位偏移量变址	IX1+	8位偏移量变址加一

DD DIR - DIR IMD IMM - DIR
IX+D IX+-DIR DIX+ DIR - IX+

操作码十六进制

F0	3
SUB	
1	IX

 HCS08 周期
字节数 指令助记符
寻址模式

表 7-3 操作码映像 (2/2)

Bit-Manipulation	Branch	Read-Modify-Write				Control				Register/Memory					
					9E60 NEG 3 SP1	6						9ED0 SUB 4 SP2	5	9EE0 SUB 3 SP1	4
					9E61 CBEQ 4 SP1	6						9ED1 CMP 4 SP2	5	9EE1 CMP 3 SP1	4
												9ED2 SBC 4 SP2	5	9EE2 SBC 3 SP1	4
					9E63 COM 3 SP1	6						9ED3 CPX 4 SP2	5	9EE3 CPX 3 SP1	4
					9E64 LSR 3 SP1	6						9ED4 AND 4 SP2	5	9EE4 AND 3 SP1	4
												9ED5 BIT 4 SP2	5	9EE5 BIT 3 SP1	4
					9E66 ROR 3 SP1	6						9ED6 LDA 4 SP2	5	9EE6 LDA 3 SP1	4
					9E67 ASR 3 SP1	6						9ED7 STA 4 SP2	5	9EE7 STA 3 SP1	4
					9E68 LSL 3 SP1	6						9ED8 EOR 4 SP2	5	9EE8 EOR 3 SP1	4
					9E69 ROL 3 SP1	6						9ED9 ADC 4 SP2	5	9EE9 ADC 3 SP1	4
					9E6A DEC 3 SP1	6						9EDA ORA 4 SP2	5	9EEA ORA 3 SP1	4
					9E6B DBNZ 4 SP1	8						9EDB ADD 4 SP2	5	9EEB ADD 3 SP1	4
					9E6C INC 3 SP1	6									
					9E6D TST 3 SP1	5									
												9EAE LDHX 2 IX	5	9EBE LDHX 4 IX2	6
												9ECE LDHX 3 IX1	5	9EDE LDX 4 SP2	5
					9E6F CLR 3 SP1	6						9EDF STX 4 SP2	5	9EEF STX 3 SP1	4
														9EFF STHX 3 SP1	5

INH 内在 REL 相对 SP1 8位偏移量堆栈
IMM 立即 IX In无偏移量变址 SP2 16偏移量堆栈
DIR 直接 IX1 8位偏移量变址 IX+ 无偏移量变址加一
EXT 扩展 IX2 16位偏移量变址 IX1+ 8位偏移量变址加一
DD DIR - DIR IMD IMM - DIR
IX+D IX+-DIR DIX+ DIR - IX+

注：所有第二页的操作码都前缀（9E）

前缀字节(9E)操作码十六进制

9E60	6
NEG	
3	SP1

 HCS08 周期
字节数 指令助记符
寻址模式

第八章 AD转换器(ADC0V1)

8.1 概述

10 位的 AD 转换器是一个逐次逼近 ADC，可以设计在一个集成微控制器片上系统里面。

ADC 支持高达 28 个独立模拟输入（AD0-AD27）。在 MC9S08QD4 系列的 MCU 中只实现了四个输入（ADCP0-ADCP3）。这些输入可以通过 ADCH 位选择。

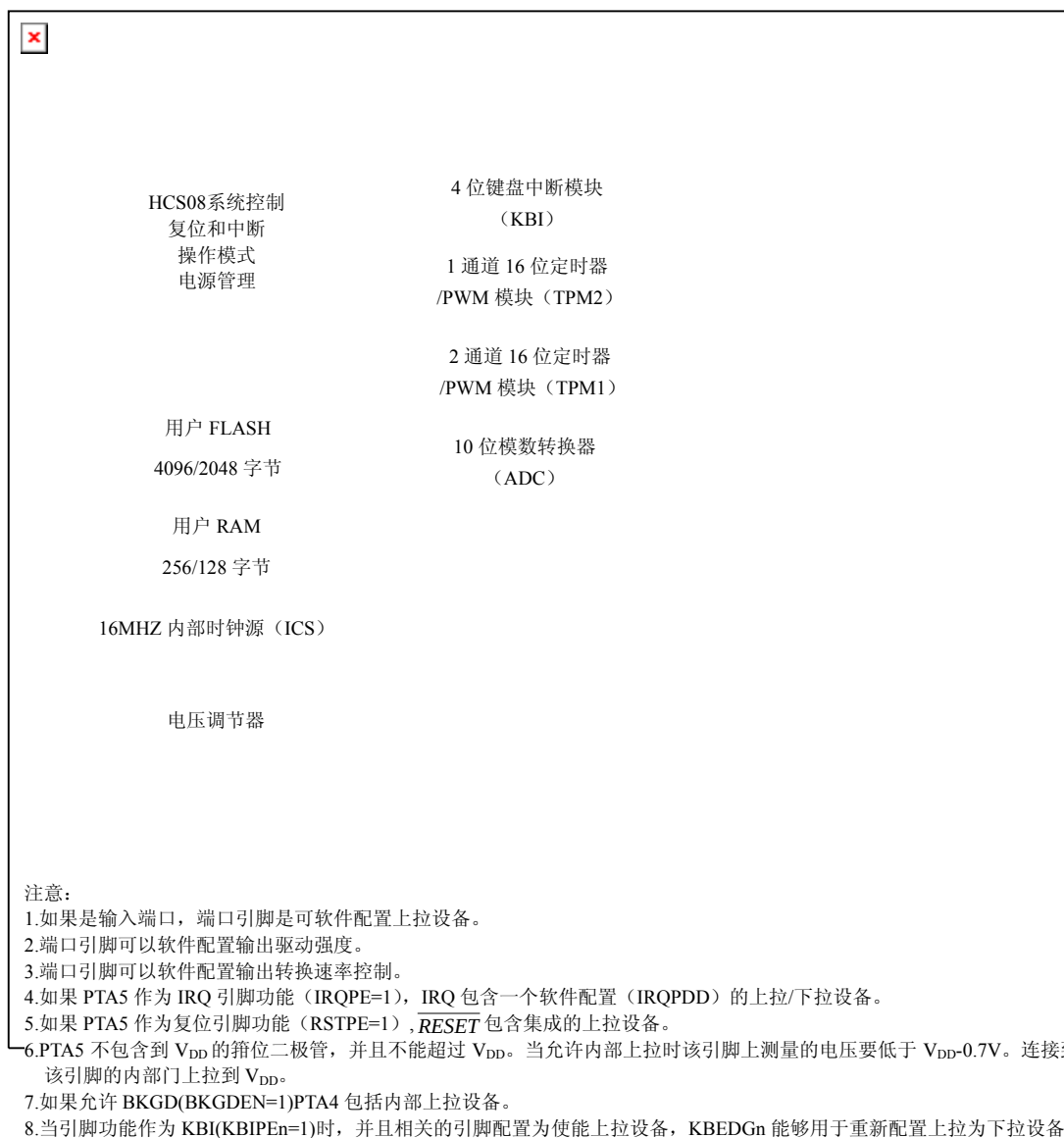


图8-1 MC9S08QD4系列框图高亮显示ADC框和引脚

8.1.1 模块配置

该节提供了配置 MC9S08QD4 系列的 ADC 的设备描述信息。

8.1.1.1 通道分配

MC9S08QD4 系列设备的 ADC 通道分配如表 8-1 所示。未实现的通道在内部连接到 V_{REFL} 。保留的通道是未知值。

表 8-1 ADC 通道分配

ADCH	通道	输入	引脚控制	ADCH	通道	输入	引脚控制
00000	AD0	PTA0/AD1P0	ADPC0	10000	AD16	V_{SS}	N/A
00001	AD1	PTA1/AD1P1	ADPC1	10001	AD17	V_{SS}	N/A
00010	AD2	PTA2/AD1P2	ADPC2	10010	AD18	V_{SS}	N/A
00011	AD3	PTA3/AD1P3	ADPC3	10011	AD19	V_{SS}	N/A
00100	AD4	V_{SS}	N/A	10100	AD20	V_{SS}	N/A
00101	AD5	V_{SS}	N/A	10101	AD21	V_{SS}	N/A
00110	AD6	V_{SS}	N/A	10110	AD22	Reserved	N/A
00111	AD7	V_{SS}	N/A	10111	AD23	Reserved	N/A
01000	AD8	V_{SS}	ADPC8	11000	AD24	Reserved	N/A
01001	AD9	V_{SS}	ADPC9	11001	AD25	Reserved	N/A
01010	AD10	V_{SS}	ADPC10	11010	AD26	Temp Sensor ¹	N/A
01011	AD11	V_{SS}	ADPC11	11011	AD27	Internal Bandgap ²	N/A
01100	AD12	V_{SS}	N/A	11100	V_{REFH}	V_{DD}	N/A
01101	AD13	V_{SS}	N/A	11101	V_{REFH}	V_{DD}	N/A
01110	AD14	V_{SS}	N/A	11110	V_{REFL}	V_{SS}	N/A
01111	AD15	V_{SS}	N/A	11111	module disable	None	N/A

1.更多信息，参见8.1.1.5节“温度传感器”。
 2.要求SPMSC1中的BGBE=1，参见5.8.8节系统电源管理状态和控制1寄存器（SPMSC1）。更多关于能带隙电压参见A.6节“DC特性”。

8.1.1.2 交替时钟

ADC 模块能够变化地使用 MCU 总线时钟，二分频的总线时钟，模块内的异步时钟（ADACK）。交替时钟，ALTCLK 的输入在 MC9S08QD4 系列 MCU 设备中没有实现。

8.1.1.3 硬件触发器

ADC 硬件触发器，ADHWT，是实时中断计数器（RTI）的输出。RTI 计数器可以使用 ICGERCLK 或 RTI 时钟内的名义上的 32kHz 时钟源作为时钟。

RTI 的周期可以由输入时钟和 RTIS 位决定。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。当允许 ADC 硬件触发器时，RTI 计数器溢出时，初始化一个转换。

MCU 处于运行，等待，stop3 模式时，RTI 可以配置为导致一个硬件触发。

8.1.1.4 模拟引脚使能

MC9S08QD4 系列的 ADC 只包含一个模拟引脚使能寄存器，APCTL1。

8.1.1.5 温度传感器

为了使用片上的温度传感器，用户必须做如下操作：

- 配置 ADC 为最大 1Mhz 时钟的长采样。
- 转换带隙参考电压通道（AD27）。

- 使用 VBG 的值通过转换带隙参考电压通道的数字值，用户可以决定 VDD。更多关于带隙电压的值，参见附录 A.5 “DC 特性”。

- 转换温度传感器通道 (AD26)。

- 通过使用计算的 VDD 的值，转换 AD26 的数据值为电压，VTEMP。

等式8-1提供了温度传感器的近似转换函数。其中VDD=3.0V，Temp=25°C，使用ADC的fADCK=1.0MHz并且配置为长采样。

ADC 模块包含一个温度传感器，它的输出连接到一个 ADC 模拟通道输入。

$$\text{Temp}=25-((\text{VTEMP}-1.3894)/(0.0033)) \quad (\text{等式 8-1})$$

0.0017 是未校准电压温度斜率（伏/摄氏度）。温度传感器的未校准大约是±12°C，使用等式 8-1。

为了提供准确性用户必须校准带隙参考电压和温度传感器。

在 25°C 时的校准将提高准确性到±4.5°C。

在-40°C，25°C，105°C 3 个点校准将提高准确性到±2.5°C。一旦校准完成，用户将计算冷和热的斜率。在应用代码中，用户可以使用上面描述的等式 8-1 并且确定温度是大于或小于 25°C 来计算温度。一旦确定了温度是大于还是小于 25°C，用户可以使用校验得到的冷或热斜率来重新计算温度。

8.1.1.6 低功耗模式操作

ADC 能够在 stop3 模式下运行，但是要求 SPMSC1 中的 LVDSE 置位。

8.1.2 特征

ADC 模块特点包括：

- 线性逐次逼近算法，10 位精度。
- 多达 28 个模拟输入。
- 3.8 位或 10 位右对齐格式输出。
- 单个或连续的转换（单个转换后自动返回到空闲）。
- 设置采样时间和转换速度（功率）。
- 转换完成标志和中断。
- 输入时钟可以选择高达四个时钟源。
- 在等待或 stop3 模式中操作为低噪音操作。
- 异步时钟源的低噪音操作。
- 可选的异步硬件转换触发。
- 自动比较小于，大于或等于编程值。

8.1.3 框图

图 8-2 提供了 ADC 模块的框图。

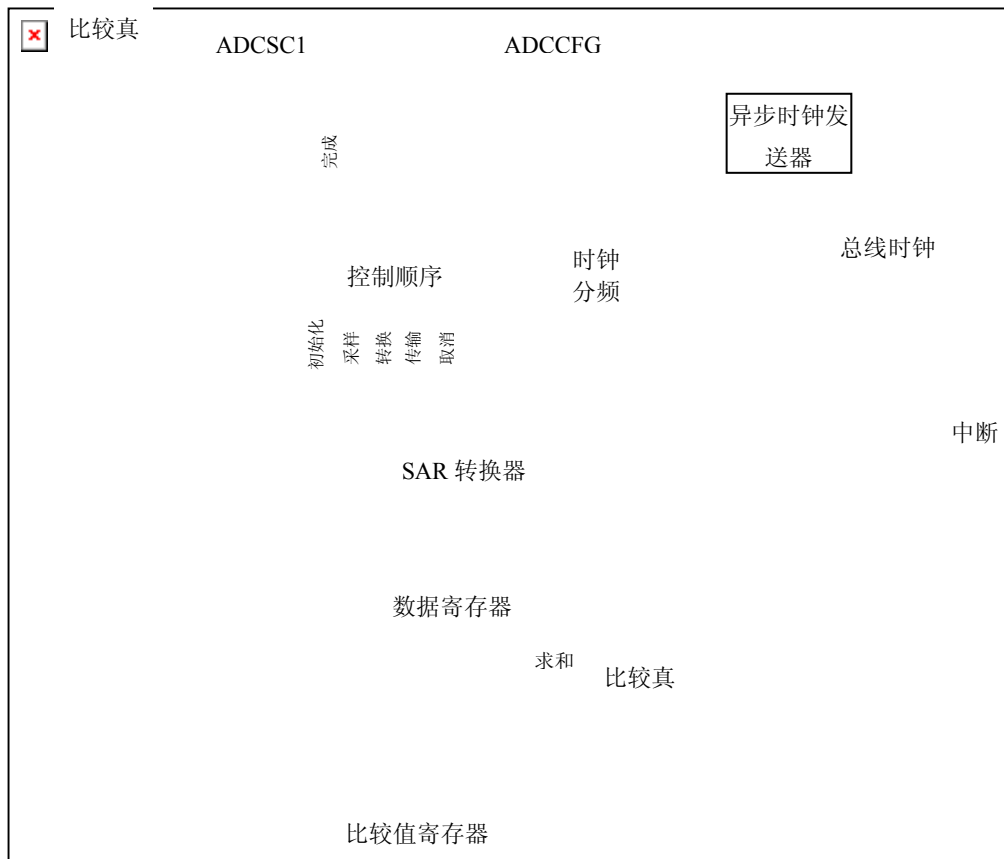


图8-2 ADC框图

8.2 外部信号描述

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

表 8-2 信号属性

名称	功能
AD27-AD0	模拟通道输入
VREFH	高参考电压
VREFL	低参考电压
VDDAD	模拟电压供电
VSSAD	模拟地

8.2.1 模拟电源 (V_{DDAD})

ADC 模拟部分使用作为它的电源连接。在相同的封装中， V_{DDAD} 在内部连接到 V_{DD} 。如果外部可能，连接到 V_{DDAD} 的引脚和 V_{DD} 到相同电压。外部滤波对 V_{DDAD} 可能是必要的。

8.2.2 模拟地 (V_{SSAD})

ADC 模拟部分使用作为它的地连接。在相同的封装中， V_{SSAD} 在内部连接到 V_{SS} 。如果外部可能，连接到 V_{SSAD} 引脚和 V_{SS} 到相同电压。

8.2.3 参考高电压 (V_{REFH})

V_{REFH} 是转换器的参考高电压，在相同的封装中， V_{REFH} 在内部连接到 V_{DDAD} ，如果外部可能，连接到 V_{REFH} 的引脚和 V_{DDAD} 到相同电压。或者被外部源（在最小的 V_{DDAD} 规格和 V_{DDAD} 电压之间）驱动。

8.2.4 参考低电压 (V_{REFL})

V_{REFL} 是转换器的参考低电压，在相同的封装中， V_{REFL} 在内部连接到 V_{SSAD} ，如果外部可能，连接到 V_{REFL} 的引脚和 V_{SSAD} 到相同电压。

8.2.5 模拟通道输入 (AD_X)

ADC 模块支持高达 28 个独立的模拟输入。通过 $ADCH$ 通道选择位，一个输入被选择用于转换。

8.3 寄存器定义

这些内存映像寄存器控制和管理 ADC 的操作：

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

8.3.1 状态和控制寄存器1 ($ADCSC1$)

本节描述 ADC 状态和控制寄存器 ($ADCSC1$) 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）。

	第7位	6	5	4	3	2	1	第0位
读	COCO	AIEN	ADCO	ADCH				
写								
复位	0	0	0	1	1	1	1	1

图8-3 状态和控制寄存器($ADCSC1$)

表 8-3 $ADCSC1$ 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。当比较功能禁止 ($ACFE=0$) 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 $ADCSC1$ 或读 $ADCRL$ ，该位清零。 0 转换未完成。 1 转换完成。
6 AIEN	中断允许——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 禁止转换完成中断。 1 允许转换完成中断。

5 ADCO	连续转换使能——ADCO用于使能连续转换。 0 当选择软件触发中断时，写ADCSC1后开始一个转换。当选择硬件触发中断时，确认了ADHWT后开始一个中断。 1 当选择软件触发中断时，写ADCSC1后初始化连续转换。当选择硬件触发中断时，连续转换被ADHWT事件初始化。
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8-4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和从所有的源中孤立输入通道。

ADCH	输入选择
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7
01000	AD8
01001	AD9
01010	AD10
01011	AD11
01100	AD12
01101	AD13
01110	AD14
01111	AD15
10000	AD16
10001	AD17
10010	AD18
10011	AD19
10100	AD20
10101	AD21
10110	AD22
10111	AD23
11000	AD24
11001	AD25
11010	AD26
11011	AD27
11100	保留
11101	V _{REFH}
11110	V _{REFL}
11111	模块被禁止
图8-4 输入通道选择	

8.3.2 状态和控制寄存器2（ADCSC2）

ADCSC2 寄存器用于控制比较功能，转换触发和 ADC 模块的转换行为。

	第7位	6	5	4	3	2	1	第0位
读	ADACT	ADTRG	ACFE	ACFGT	0	0	R ¹	R ¹
写								
复位	0	0	0	0	0	0	0	0

1.位1和0保留，写总为0。

图8-5 状态和控制寄存器2(ADCSC2)

表 8-4 ADCSC2 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 转换未进行。 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。选择软件触发，写ADCSC1后初始化一个转换。选择硬件触发，确认了ADHWT输入后初始化转换。 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 禁止比较功能。 1 允许比较功能。
4 ACFGT	比较功能更大使能——当监控的输入的转换结果大于或等于比较结果时，ACFGT用于设置比较功能的触发。当监控的输入的转换结果小于比较结果时，比较功能默认触发。 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。

8.3.3 数据高结果寄存器（ADCRH）

ADCRH 包含 10 为转换结果的高 2 位。当设置位 8 位转换时，ADR8 和 ADR9 都等于 0。每次转换完成，ADCRH 被更新，除非自动比较被允许而且不满足比较结果。在 10 位数据模式中，直到 ADCRL 被读，再读 ADCRH 可以避免传输转换结果的一部分到结果寄存器。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRL 的互锁。在这种情况下，MODE 位被改变，ADCRH 中的任何数据都无效。

	第7位	6	5	4	3	2	1	第0位
读	0	0	0	0	0	0	ADR9	ADR8
写								
复位	0	0	0	0	0	0	0	0

图8-6 数据高结果寄存器(ADCRH)

8.3.4 数据低结果寄存器（ADCRL）

ADCRL 包含 10 为转换结果的低 8 位，一个 8 位转换的所有 8 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 位数据模式中，直到 ADCRL 被读，再读 ADCRH 可以避免传输转换结果的一部分到结果寄存器。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRH 的互锁。在这种情况下，MODE 位被改变，ADCRL 中的任何数据都无效。

	第7位	6	5	4	3	2	1	第0位
读	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
写								
复位	0	0	0	0	0	0	0	0

图8-7 数据低结果寄存器(ADCRL)

8.3.5 比较值高寄存器（ADCCVH）

该寄存器包含了 10 位比较值的高 2 位。当允许比较功能时，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。

	第7位	6	5	4	3	2	1	第0位
读	0	0	0	0			ADCV9	ADCV8
写								
复位	0	0	0	0	0	0	0	0

图8-8 比较值高寄存器(ADCCVH)

8.3.6 比较值低寄存器 (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 8 位。在 10 位或 8 位模式中，ADCV7:ADCV0 转换结果的低 8 位比较。

	第7位	6	5	4	3	2	1	第0位
读	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
写								
复位	0	0	0	0	0	0	0	0

图8-9 比较值低寄存器(ADCCVL)

8.3.7 配置寄存器 (ADCCFG)

ADCCFG 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。

	第7位	6	5	4	3	2	1	第0位
读	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
写								
复位	0	0	0	0	0	0	0	0

图8-10 配置寄存器(ADCCFG)

表 8-5 ADCCFG 寄存器域描述

域	描述
7 ADLPC	低功耗配置——ADLPC控制逐次渐进转换器的速度和功耗配置。当不需要更高采样速率时，可以优化功耗。 0 高速配置。 1 低功耗配置：以最大化时钟速率的代价降低功耗。
6:5 ADIV	时钟分频选择——ADIV选择ADC使用的分频因子，产生内部时钟ADCK。表8-6描述了时钟配置。
4 ADLSMP	长采样时间配置——ADLSMP选择长和短采样时间。这可以调整采样周期，使在更高阻抗的输入下也能得到精确的采样，也可以在低阻抗时最大化转换速度。如果连续采样允许而且不需要高转换速率，更长的采样时间可以用于更低的总功耗。 0 短采样时间。 1 长采样时间。
3:2 MODE	转换模式选择——MODE位用于选择8位或10位操作。参见表8-7。
1:0 ADICLK	输入时钟选择——ADICLK选择产生内部时钟ADCK的输入时钟源。参见表8-8。

表 8-6 时钟分频选择

ADIV	分频因子	时钟
00	1	输入时钟
01	2	输入时钟/2
10	4	输入时钟/4
11	8	输入时钟/8

表 8-7 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 8-8 输入时钟选择

ADICLK	时钟源选择
00	总线时钟
01	总线时钟/2
10	交替时钟(ALTCLK)
11	异步时钟(ADACK)

8.3.8 引脚控制1寄存器 (APCTL1)

引脚控制寄存器用于禁止 MCU 引脚的 I/O 口用作模拟输入。APCTL1 用于控制和 ADC 模块的通道 0-7 相关的位。

	第7位	6	5	4	3	2	1	第0位
读	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
写								
复位	0	0	0	0	0	0	0	0

图8-11 引脚控制1寄存器(APCTL1)

表 8-9 APCTL1 寄存器域描述

域	描述
7 ADPC7	ADC引脚控制7——ADPC7用于控制和通道AD7相关的位。 0 允许AD7引脚I/O控制。 1 禁止AD7引脚I/O控制。
6 ADPC6	ADC引脚控制6——ADPC6用于控制和通道AD6相关的位。 0 允许AD6引脚I/O控制。 1 禁止AD6引脚I/O控制。
5 ADPC5	ADC引脚控制5——ADPC5用于控制和通道AD5相关的位。 0 允许AD5引脚I/O控制。 1 禁止AD5引脚I/O控制。
4 ADPC4	ADC引脚控制4——ADPC4用于控制和通道AD4相关的位。 0 允许AD4引脚I/O控制。 1 禁止AD4引脚I/O控制。
3 ADPC3	ADC引脚控制3——ADPC3用于控制和通道AD3相关的位。 0 允许AD3引脚I/O控制。 1 禁止AD3引脚I/O控制。
2 ADPC2	ADC引脚控制2——ADPC2用于控制和通道AD2相关的位。 0 允许AD2引脚I/O控制。 1 禁止AD2引脚I/O控制。
1 ADPC1	ADC引脚控制1——ADPC1用于控制和通道AD1相关的位。 0 允许AD1引脚I/O控制。 1 禁止AD1引脚I/O控制。
0 ADPC0	ADC引脚控制0——ADPC0用于控制和通道AD0相关的位。 0 允许AD0引脚I/O控制。 1 禁止AD0引脚I/O控制。

8.3.9 引脚控制2寄存器 (APCTL2)

APCTL2 用于控制和 ADC 模块的通道 8-15 相关的位。

	第7位	6	5	4	3	2	1	第0位
读	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
写								
复位	0	0	0	0	0	0	0	0

图8-12 引脚控制2寄存器(APCTL2)

表 8-10 APCTL2 寄存器域描述

域	描述
7 ADPC15	ADC引脚控制15——ADPC15用于控制和通道AD15相关的位。 0 允许AD15引脚I/O控制。 1 禁止AD15引脚I/O控制。
6 ADPC14	ADC引脚控制14——ADPC14用于控制和通道AD14相关的位。 0 允许AD14引脚I/O控制。 1 禁止AD14引脚I/O控制。
5 ADPC13	ADC引脚控制13——ADPC13用于控制和通道AD13相关的位。 0 允许AD13引脚I/O控制。 1 禁止AD13引脚I/O控制。
4 ADPC12	ADC引脚控制12——ADPC12用于控制和通道AD12相关的位。 0 允许AD12引脚I/O控制。 1 禁止AD12引脚I/O控制。
3 ADPC11	ADC引脚控制11——ADPC11用于控制和通道AD11相关的位。 0 允许AD11引脚I/O控制。 1 禁止AD11引脚I/O控制。
2 ADPC10	ADC引脚控制10——ADPC10用于控制和通道AD10相关的位。 0 允许AD10引脚I/O控制。 1 禁止AD10引脚I/O控制。
1 ADPC9	ADC引脚控制9——ADPC9用于控制和通道AD9相关的位。 0 允许AD9引脚I/O控制。 1 禁止AD9引脚I/O控制。
0 ADPC8	ADC引脚控制8——ADPC8用于控制和通道AD8相关的位。 0 允许AD8引脚I/O控制。 1 禁止AD8引脚I/O控制。

8.3.10 引脚控制3寄存器 (APCTL3)

APCTL3 用于控制和 ADC 模块的通道 16-23 相关的位。

	第7位	6	5	4	3	2	1	第0位
读	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
写								
复位	0	0	0	0	0	0	0	0

图8-13 引脚控制3寄存器(APCTL3)

表 8-11 APCTL3 寄存器域描述

域	描述
7 ADPC23	ADC引脚控制23——ADPC23用于控制和通道AD23相关的位。 0 允许AD23引脚I/O控制。 1 禁止AD23引脚I/O控制。

6 ADPC22	ADC引脚控制22——ADPC22用于控制和通道AD22相关的位。 0 允许AD22引脚I/O控制。 1 禁止AD22引脚I/O控制。
5 ADPC21	ADC引脚控制21——ADPC21用于控制和通道AD21相关的位。 0 允许AD21引脚I/O控制。 1 禁止AD21引脚I/O控制。
4 ADPC20	ADC引脚控制20——ADPC20用于控制和通道AD20相关的位。 0 允许AD20引脚I/O控制。 1 禁止AD20引脚I/O控制。
3 ADPC19	ADC引脚控制19——ADPC19用于控制和通道AD19相关的位。 0 允许AD19引脚I/O控制。 1 禁止AD19引脚I/O控制。
2 ADPC18	ADC引脚控制18——ADPC18用于控制和通道AD18相关的位。 0 允许AD18引脚I/O控制。 1 禁止AD18引脚I/O控制。
1 ADPC17	ADC引脚控制17——ADPC17用于控制和通道AD17相关的位。 0 允许AD17引脚I/O控制。 1 禁止AD17引脚I/O控制。
0 ADPC16	ADC引脚控制16——ADPC16用于控制和通道AD16相关的位。 0 允许AD16引脚I/O控制。 1 禁止AD16引脚I/O控制。

8.4 功能描述

当复位或 ADCH 位全高时，禁止 ADC 模块。当转换完成而且另一个转换还未初始化时，该模块空闲。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被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（ADCRH 和 ADCRL）。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 并且如果允许转换完成中断（AIEN=1），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。通过置位 ACFE 位，允许比较功能。该功能和任何转换模式和设置协力完成操作。

8.4.1 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。这个时钟源除以一个设置值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 ADICK 位决定。

- 总线时钟，等于软件执行时的频率。这是复位后的默认值。
- 总线时钟/2。对于更高的时钟，可以允许最大总线时钟除以 16。
- ALTCLK，由 MCU 定义（参见模块一节的介绍）
- 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 stop3 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，它的频率都必须低于规定的 ADCK 频率范围。如果可用的时钟太慢，根据规定 ADC 将不会工作。如果可用的时钟太快，时钟必须分频到适当的频率。分频因子由 ADIV 位决定，可以除以 1、2、4、8。

8.4.2 输入选择和引脚控制

引脚控制寄存器（APCTL3、APCTL2、APCTL1）可以禁止引脚的 I/O 控制用于模拟输入。当引脚控制寄存器位置位时，相应的 MCU 位将会服从接下来的条件：

- 输出缓冲区强制为高阻抗状态。
- 禁止输入缓冲区。读这些禁止的缓冲区的任何位返回 0。
- 禁止上拉电阻。

8.4.3 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 位置位时，它被允许。这个源并不是在所有的 MCU 都有效。关于该 MCU 的 ADHWT 源的具体细节参考模块介绍。

当 ADHWT 源可用并且硬件触发被允许（ADTRG=1），在 ADHWT 的上升沿初始化转换。如果当一个上升沿产生，一个转换正在处理中，上升沿被忽略。在连续转换设置中，只有引起连续转换的首次上升沿可以被发现。硬件触发功能和任何转换模式和设置协力完成操作。

8.4.4 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。

8.4.4.1 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 ADCSC1 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。

如果允许连续转换，当前转换完成后，一个新的转换可以自动初始化。在软件触发中，连续转换在写 ADCSC1 后开始，并继续直到终止。在硬件触发操作中，连续转换在硬件触发事件后开始，并继续直到终止。

8.4.4.2 完成转换

当转换的结果传到数据结果寄存器，ADCRH 和 ADCRL 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（ADCRH 已经被读但是 ADCRL 还未被读），闭锁机制保护新的数据不会重写在 ADCRH 和 ADCRL 中以前的数据。当锁有效，数据传送被锁，COCO 不能置位，新的数据丢失。在允许比较功能的单个转换的情况下，并且比较条件为假，锁没有作用，ADC 操作被终止。在其他情况下，当数据传送被锁，除非 ADCO 的状态（允许单个或连续转换），另一个转换被初始化。

如果允许单个转换，闭锁机制可能导致丢弃几个转换并且额外的功耗。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。

8.4.4.3 终止转换

任何正在出来的转换都将终止，当

- 写 ADCSC1 发生（如果 ADCH 不是全 1，当前的转换被取消并开始了新的转换，）。
- 写 ADCSC2、ADCCFG、ADCCVH 或 ADCCVL 发生。这表明转换模式发生改变，因

此当前的转换无效。

- MCU 复位。
- MCU 进入停止模式并且禁止 ADACK。

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。

8.4.4.4 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 时钟产生器也被允许。

当有效时功耗可以通过设置 ADLPC 减小。这导致更小的 f_{ADCK} 最大值（参考电气描述）。

8.4.4.5 总转换时间

总转换时间依赖于抽样时间（由 ADLSMP 决定），MCU 总线频率，转换模式（8 位或 10 位），转换时钟的频率（ f_{ADCK} ）。模块有效后，输入的采样开始。ADLSMP 用于选择长或短采样时间。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值。转换算法完成后，转换结果传送到 ADCRH 和 ADCRL。

如果总线频率小于 f_{ADCK} 频率，当允许短采样（ADLSMP=0）时，无法保证连续转换的精确采样时间。如果总线频率小于 f_{ADCK} 频率的 1/11，当允许长采样（ADLSMP=1）时，无法保证连续转换的精确采样时间。

表 8-12 中总结了不同条件下的最大的总转换时间。

表 8-12 不同控制条件的总转换时间

转换类型	ADICLK	ADLSMP	最大总转换时间
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	40ADCK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

最大的总转换时间由转换时钟和分频因子决定。时钟源由 ADICLK 位决定，分频因子由 ADIV 描述。例如，在 10 位模式中，选择总线时钟作为输入时钟源，输入时钟除以 1 分频，8MHz 总线时钟，则单个转换的转换时间是：

$$\text{转换时间} = \frac{23 \text{ ADCK 周期}}{8 \text{ MHz} / 1} + \frac{5 \text{ 总线周期}}{8 \text{ MHz}} = 3.5 \mu\text{s}$$

总线时钟的个数=3.5 μ s $\times$ 8MHz=28（周期）

注：ADCK 频率必须在 ADC 说明书中的 f_{ADCK} 最小值和 f_{ADCK} 最大值之间。

8.4.5 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（ADCCVH 和 ADCCVL）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 ADCRH 和 ADCRL。

允许比较功能时，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。如果允许 ADC 中断（AIEN=1）当 COCO 置位时，产生一次 ADC 中断。

注：当 MCU 在等待或 stop3 模式时，比较功能用于监控通道上的电压。满足比较条件时，ADC 中断将唤醒 MCU。

8.4.6 MCU 等待模式操作

WAIT 指令使 MCU 进入低功耗待命模式。因为时钟源仍然活动，这种模式可以很快恢复。如果 MCU 进入等待模式时，有一个转换正在处理，它将继续直到完成。当 MCU 处于等待模式时，通过硬件触发的方式或者如果允许连续转换，可以初始化转换。

处于等待模式时，总线时钟，总线时钟的一半和 ADACK 可以作为转换时钟源。在等待模式时，ALTCLK 作为转换时钟源使用是由该 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。

一次转换完成事件置位 COCO 并且如果允许 ADC 中断（AIEN=1）将会产生一个 ADC 中断将 MCU 从等待模式唤醒。

8.4.7 MCU stop3 模式操作

在 MCU 中禁止了大多数或所有的时钟源期间，STOP 指令可以使 MCU 进入低功耗待命模式。

8.4.7.1 禁止 ADACK 的 stop3 模式

如果不选择异步时钟 ADACK 作为转换时钟，执行 STOP 指令终止当前转换并且使 ADC 进入空闲状态。stop3 模式不影响 ADCRH 和 ADCRL 的值。从 stop3 模式退出后，需要一个软件或硬件触发重新开始转换。

8.4.7.2 允许 ADACK 的 stop3 模式

如果选择 ADACK 作为转换时钟，在 stop3 模式时，ADC 继续工作。为了保证 ADC 操作，MCU 的电源调整器在 stop3 模式时必须仍然有效。参考该 MCU 模块介绍中的配置信息。

如果 MCU 进入 stop3 模式时，有一个转换正在处理，它将继续直到完成。当 MCU 处于 stop3 模式时，通过硬件触发的方式或者如果允许连续转换，可以初始化转换。

转换完成事件置位 COCO，并且如果中断允许（AIEN=1）还会产生一个 ADC 中断将 MCU 从等待模式唤醒。

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电平中断。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闭锁机制（在第 11.4.5.2 节“完成转换”）清零。

8.4.8 MCU stop1 和 stop2 模式操作

当 MCU 进入 stop1 或 stop2 模式时，自动禁止 ADC 模块。从 stop1 或 stop2 退出时，所有

的模块寄存器存放的是复位值。因此从 stop1 或 stop2 退出时，模块必须被重新使能和重新配置。

8.5 初始化信息

该节举例说明了用户如何初始化和配置 ADC 模块的一些基本方用法。用户可以灵活地选择配置模块，8 位或 10 位精度，单个或连续转换，循环或中断方式，还有其他选项。在该例中的信息可以参见表 8-6，表 8-7，表 8-8。

注：十六进制的前缀是 0x，二进制的前缀是%，十进制没有前缀。

8.5.1 ADC模块初始化举例

8.5.1.1 初始化顺序

在 ADC 模块可以用于完成转换操作前，必须初始化。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（如果允许）。
- 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，并且允许或禁止转换完成中断。选择哪路输入通道完成转换也是在这里操作。

8.5.1.2 伪代码举例

在该例中，ADC 模块将提供允许中断产生一个单个 10 位转换，低功耗，输入通道 1 上的长采样时间，内部的 ADCK 时钟将由总线时钟除以 1 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	设置位长采样时间
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	禁止比较功能
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Freescall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（禁止连续转换）
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

禁止 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。

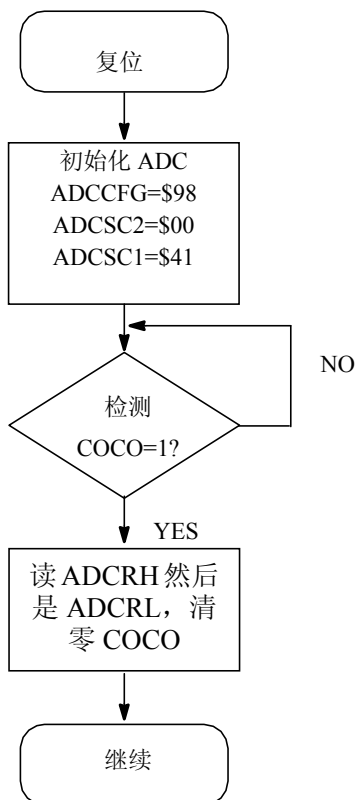


图8-14 该例的初始化流程

8.6 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。

8.6.1 外部引脚和安排

下面讨论了和 ADC 模块相关的外部引脚和如何最好的使用它们。

8.6.1.1 模拟电源引脚

ADC 模块有电源和地引脚 (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 $MCUV_{SS}$ 复用相同的引脚。在一些设备中， V_{SSAD} 和 V_{DDAD} 共享数字供电引脚。在这些情况中，有独立的焊盘来模拟供电，和相应的数字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} 接单个的地位置。

8.6.1.2 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。高参考电源是 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_{REFL} 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 $0.1 \mu F$ 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)。

8.6.1.3 模拟输入引脚

外部模拟输入通常和 MCU 设备的 I/O 引脚复用。通过置位引脚控制寄存器的相应位可以禁止引脚 I/O 控制。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禁止上拉电阻。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。置位引脚控制位，所有的引脚作为模拟输入，可以达到最低的操作电路。

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 $0.01 \mu F$ 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} 转换电路把信号转换成 \$3FF (共 10 位表示) 或 \$FF (共 8 位表示)，如果输入等于或小于 V_{REFH} ，转换电路把信号转换成 \$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。

8.6.2 错误源

A/D 转换中存在几种错误源。它们在该节的后面讨论。

8.6.2.1 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 $7k \Omega$ 和输入电容 $5.5pF$ ，外部模拟源 (R_{AS}) 的电阻小于 $5k \Omega$ ，则采样 1/4LSB (10 位精度) 可以在最小的采样窗口完成 (3.5 周期@8MHz 最大 ADCK 频率)。

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。

8.6.2.2 引脚泄露误差

如果外部模拟源 (R_{AS}) 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，保持 R_{AS} 小于 $V_{DDAD} / (2^N * I_{LEAK})$ 将会后更少泄露错误 (8 位模式 $N=8$, 10 位模式 $N=10$)

8.6.2.3 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。只要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 $0.1 \mu F$ 的低内阻电容。

- VDDAD 和 VSSAD 之间有一个 $0.1\ \mu\text{F}$ 的低内阻电容。
- 如果电源使用感应隔离，一个 $1\ \mu\text{F}$ 的电容放在 VDDAD 和 VSSAD 之间。
- VSSAD (和 VREFL, 如果连接) 连接到 VSS (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, MCU 处于等待或 stop3 状态。
 - 对于软件触发转换, 用 WAIT 或 STOP 指令写 ADCSC1 后。
 - 对于 stop3 模式操作, 选择 ADACK 作为时钟源。在 stop3 模式的操作减少 VDD 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 选择, 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:

- 在选择的输入通道和 VREFH 或 VREFL 之间放置一个 $0.01\ \mu\text{F}$ 的电容 (CAS) (这将增加噪音问题但是影响基于外部的模拟源电阻的采样率)。
- 求多次转换模拟值的平均。需要四次采样减少 1LSB 的影响, 以前的错误。
- 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。

8.6.2.4 编码宽度和量化误差

ADC 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_{delta} , 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:

$$1\text{LSB} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N \quad (\text{等式 8-2})$$

数字化结果存在固有的量化误差。对于 8 位或 10 位转换。用两点的中点表示电压, 代码被转化。因此在 8 位或 10 位模式中, 量化误差 $\pm 1/2\text{LSB}$ 。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 。

8.6.2.5 线性误差

ADC 可能存在几种非线性的情况。各种方法减少这些错误, 但是系统仍然会存在, 因为它们影响全局的精度。这些错误是:

- 归零错误 (EVS) (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, 然后实际的 \$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, 然后实际的 \$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。

8.6.2.6 编码抖动, 非单调性和遗编码

ADC 易受三种特殊形式的错误影响。它们是代码抖动, 非单调性, 遗编码。

代码抖动, 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 8.6.2.3 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。遗编码是那些对于任何输入都不会转换的值。

在 8 位或 10 位模式中，ADC 将确保单调并且不遗失编码。

第九章 内部时钟源（S08ICSV1）

9.1 简介

内部时钟源（ICS）模块提供 MCU 的时钟选择。模块包括一个锁频环（FLL），FLL 作为时钟源可以被内部参考时钟控制。该模块可以提供 FLL 或这内部参考时钟作为 MCU 系统时钟（ICSOUT）源。

无论选择哪个作为时钟源，ICSOUT 通过一个总线时钟分频器，将会得到一个更低的最终的输出时钟频率。ICSOUT 是总线频率的两倍。

图 9-1 高亮显示了 MC9S08QD4 系列的 ICS 模块。

9.1.1 ICS配置信息

ICS 控制寄存器 1（ICSC1）的位 1 和位 2 在读时总为 1。



注意:

- 1.如果是输入端口，端口引脚是可软件配置上拉设备。
- 2.端口引脚可以软件配置输出驱动强度。
- 3.端口引脚可以软件配置输出转换速率控制。
- 4.如果 PTA5 作为 IRQ 引脚功能 (IRQPE=1)，IRQ 包含一个软件配置 (IRQPDD) 的上拉/下拉设备。
- 5.如果 PTA5 作为复位引脚功能 (RSTPE=1)，RESET 包含集成的上拉设备。
- 6.PTA5 不包含到 V_{DD} 的箝位二极管，并且不能超过 V_{DD} 。当允许内部上拉时该引脚上测量的电压要低于 $V_{DD}-0.7V$ 。连接到该引脚的内部门上拉到 V_{DD} 。
- 7.如果允许 BKGD(BKGDEN=1)PTA4 包括内部上拉设备。
- 8.当引脚功能作为 KBI(KBIPEn=1)时，并且相关的引脚配置为使能上拉设备，KBEDGn 能够用于重新配置上拉为下拉设备。

图9-1 MC9S08QD4系列框图高亮显示ICS框

9.1.2 特征

ICS 模块的主要特征是:

- 锁频环 (FLL) 可以调整精度
 - 0.2%的精度，使用内部 32kHz 参考
 - 2%的电压和温度误差，使用内部 32kHz 参考
- 使用高达 5MHz 的内部或外部时钟来控制 FLL
 - 提供了 3 位来选择参考分频因子
- 内部参考时钟可以有 9 个调整位
- 可以选择内部或外部参考时钟作为 MCU 的时钟源
- 无论选择哪个作为源都可以被分频
 - 提供了 2 位来选择时钟分频因子
 - 允许的分频为：1、2、4、8

- 提供 BDC 时钟作为常数，DCO 的输出除以 2
- 提供了作为外部参考时钟的低功耗振荡器的控制信号
 - HGO、RANGE、EREFS、ERCLKEN、EREFSTEN
- 复位后自动进入 FEI（FLL 内部使用）模式

9.1.3 操作模式

ICS 有 7 中操作模式：FEI、FEE、FBI、FBILP、FBE、FBELP 和 stop。

9.1.3.1 FLL 内部使用(FEI)

FLL 内部使用模式时默认模式，在 FLL 内部使用模式中，ICS 提供从 FLL 得到的时钟，FLL 由内部参考时钟控制。BDC 时钟由 FLL 提供。

9.1.3.2 FLL 外部使用(FEE)

在 FLL 外部使用模式中，ICS 提供从 FLL 得到的时钟，FLL 由外部参考时钟控制。BDC 时钟由 FLL 提供。

9.1.3.3 FLL 旁路内部(FBI)

在 FLL 旁路内部模式中，允许 FLL 且由内部参考时钟控制，但是被旁路。ICS 提供从内部参考时钟得到的时钟。BDC 时钟由 FLL 提供。

9.1.3.4 FLL 旁路内部低功耗(FBILP)

在 FLL 旁路内部低功耗模式中，禁止 FLL 且被旁路。ICS 提供从内部参考时钟得到的时钟，BDC 时钟不可用。

9.1.3.5 FLL 旁路外部(FBE)

在 FLL 旁路外部模式中，允许 FLL 且由内部参考时钟控制，但是被旁路。ICS 提供从外部参考时钟得到的时钟。外部参考时钟可以由 ICS 控制的 OSC 提供的外部的晶体或共鸣器，或可以是另一个外部时钟源。BDC 时钟由 FLL 提供。

9.1.3.6 FLL 旁路外部低功耗(FBELP)

在 FLL 旁路外部低功耗模式中，禁止 FLL 且被旁路。ICS 提供从外部参考时钟得到的时钟。外部参考时钟可以由 ICS 控制的 OSC 提供的外部的晶体或共鸣器，或可以是另一个外部时钟源。BDC 时钟不可用。

9.1.3.7 Stop(STOP)

在 stop 模式中，禁止 FLL 且被旁路。可以选择允许或禁止内部或外部参考时钟。BDC 时钟不可用并且 ICS 不提供 MCU 时钟源。

9.1.4 框图

图 9-2 是 ICS 框图。

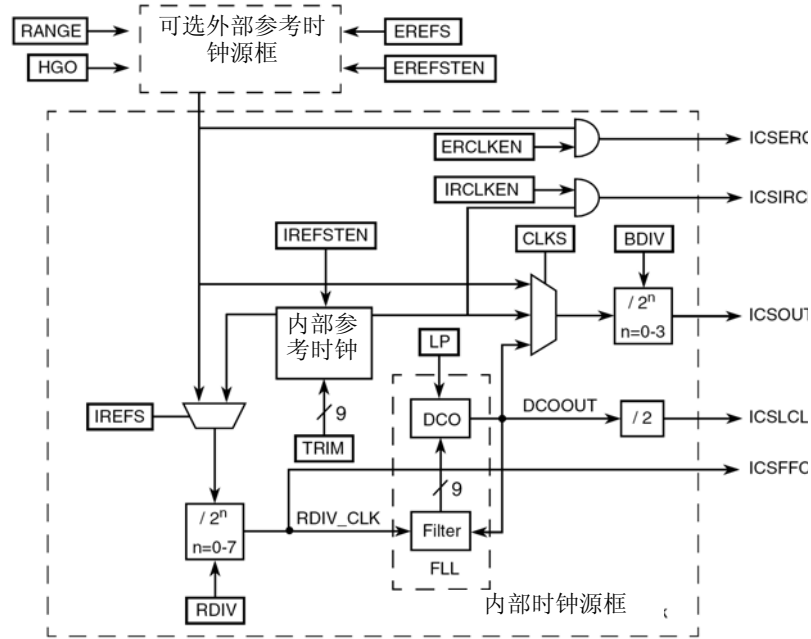


图9-2 内部时钟源（ICS）框图

9.2 外部信号描述

没有 ICS 信号外连芯片。

9.3 寄存器定义

9.3.1 ICS控制寄存器1（ICSC1）

	第7位	6	5	4	3	2	1	第0位
读	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
写								
复位	0	0	0	0	0	1	0	0

图9-3 ICS控制寄存器1（ICSC1）

表 9-1 ICS 控制寄存器 1 域描述

域	描述
7:6 CLKS	时钟源选择——选择时钟源控制总线频率。实际的总线频率依赖于BDIV位的值。 00 选择FLL输出。 01 选择内部参考时钟。 10 选择外部参考时钟。 11 保留，默认是00。

5:3 RDIV	参考分频因子——选择FLL参考时钟（FLL参考时钟由IREFS位来选择）分频的数量。 000 编码0-参考时钟除以1（复位默认） 001 编码1-参考时钟除以2 010 编码2-参考时钟除以4 011 编码3-参考时钟除以8 100 编码4-参考时钟除以16 101 编码5-参考时钟除以32 110 编码6-参考时钟除以64 111 编码7-参考时钟除以128
2 IREFS	内部参考选择——IREFS位选择FLL参考时钟。 1 选择内部参考时钟。 0 选择外部参考时钟。
1 IRCLKEN	内部参考时钟使能——IRCLKEN位允许内部参考时钟作为ICSIRCLK使用。 1 ICSIRCLK有效。 0 ICSIRCLK无效。
0 IREFSTEN	内部参考停止使能——当ICS进入stop模式时，IREFSTEN位控制是否仍然允许内部参考时钟。 1 在stop下，如果IRCLKEN置位，或进入stop前，ICS处于FEI、FBI、或FBILP模式。仍然允许内部参考时钟。 0 在stop下，禁止内部参考时钟。

9.3.2 ICS控制寄存器2（ICSC2）

	第7位	6	5	4	3	2	1	第0位
读	BDIV	RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTEN	
写								
复位	0	1	0	0	0	0	0	0

图9-4 ICS控制寄存器2（ICSC2）

表 9-2 ICS 控制寄存器 2 域描述

域	描述
7:6 BDIV	总线频率分频因子——选择时钟源（时钟源由CLKS位来选择）分频的数量。 00 编码0-选择的时钟除以1。 01 编码1-选择的时钟除以2（复位默认）。 10 编码2-选择的时钟除以4。 11 编码3-选择的时钟除以8。
5 RANGE	频率范围选择——选择外部振荡器的频率范围。 1 为外部振荡器选择高频率范围。 0 为外部振荡器选择低频率范围。
4 HGO	高增益振荡器选择——HGO位控制外部振荡器的操作模式。 1 配置外部振荡器为高增益操作。 0 配置外部振荡器为低功耗操作。
3 LP	低功耗选择——LP位控制是否在FLL旁路模式中禁止FLL。 1 除非BDM有效，在旁路模式中禁止FLL。 0 在旁路模式中不禁止FLL。
2 EREFS	外部参考选择——EREFS位选择外部参考时钟源。 1 振荡器。 0 外部时钟源。
1 ERCLKEN	外部参考选择——ERCLKEN位允许外部参考时钟作为ICSERCLK使用。 1 ICSERCLK有效。 0 ICSERCLK无效。

0	外部参考停止使能——当ICS进入stop模式时，EREFSTEN位控制是否仍然允许外部参考时钟。
EREFSTEN	1 在stop下，如果ERCLKEN置位，或进入stop前，ICS处于FEI、FBI或FBILP模式。仍然允许外部参考时钟。 0 在stop下，禁止外部参考时钟。

9.3.3 ICS调整寄存器（ICSTRM）

	第7位	6	5	4	3	2	1	第0位
读	TRIM							
写								
POR	1	0	0	0	0	0	0	0
复位	U	U	U	U	U	U	U	U

图9-5 ICS调整寄存器（ICSTRM）

表 9-3 ICS 调整寄存器域描述

域	描述
7:0 TRIM	ICS调整设置——TRIM位通过控制内部参考时钟周期来控制内部参考时钟频率。这些位的影响是二进制权重（例如，位1将调整位0的两倍）。增加TRIM中的二进制值将增加周期，减少该值则减少周期。在ICSSC中还有一个作为微调位的FTRIM。

9.3.4 ICS状态和控制寄存器（ICSSC）

	第7位	6	5	4	3	2	1	第0位
读	0	0	0	0	CLKST		OSCINIT	FTRIM
写								
POR	1	0	0	0	0	0	0	0
复位	0	0	0	0	0	0	0	U

图9-6 ICS状态和控制寄存器（ICSSC）

表 9-4 ICS 状态和控制寄存器域描述

域	描述
7:4	保留，必须清零。
3:2 CLKST	时钟模式状态——CLKST位指示当前的时钟模式。写CLKS位后，CLKST位不立即更新，因为时钟域的内同步。 00 选择FLL输出。 01 FLL被旁路，选择内部参考时钟。 10 FLL被旁路，选择外部参考时钟。 11 保留。
1	OSC初始化——如果通过ERCLKEN或者当ICS处于FEE、FBE、FBELP模式而且EREFS置位，则在外部振荡器的初始化周期完成后，该位置位。只在ERCLKEN或EREFS清零时，该位清零
0	ICS微调——FTRIM位控制内部参考时钟频率的最小调整。置位FTRIM将以最小的可能来增加周期，清零该位将以最小的可能来减小周期。

9.4 功能描述

9.4.1 操作模式

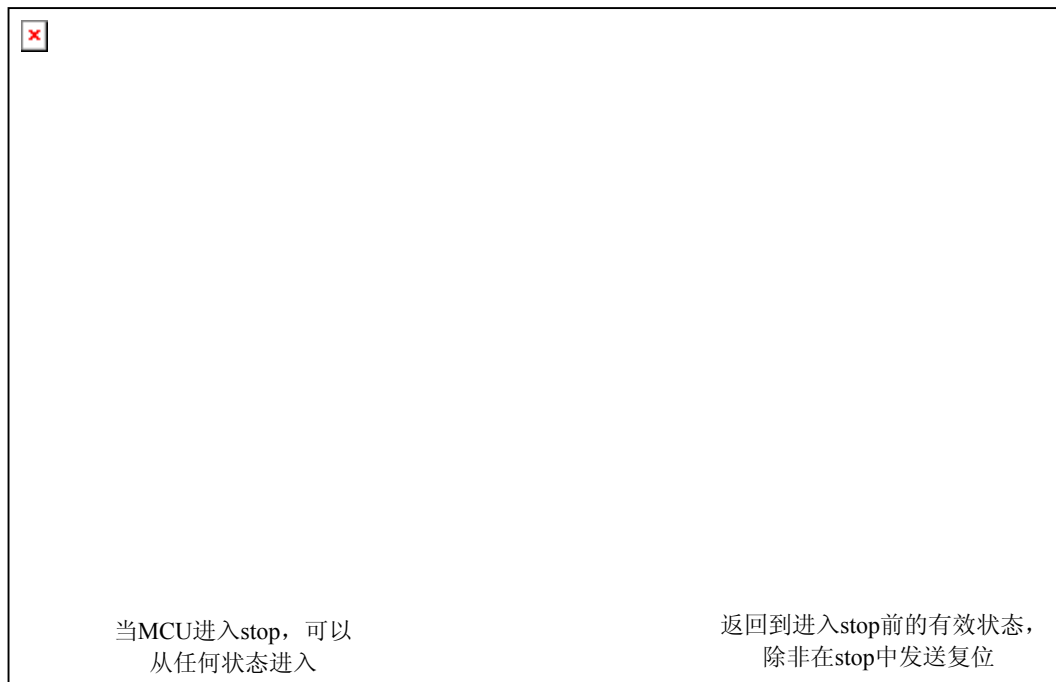


图9-7 时钟转换模式

ICS 的 7 种状态显示在状态图中，被描述如下。箭头指示状态所允许的转换。

9.4.1.1 FLL内部使用（FEI）

FLL 内部使用（FEI）时操作的默认模式，当下列所有条件发生时才进入该模式。

- 向 CLKS 位写 00。
- 向 IREFS 位写 1。
- 向 RDIV 位写 31.25kHz 到 39.0625kHz 范围的调整参考时钟。

在 FLL 内部使用模式中，ICSOUT 时钟从 FLL 时钟得到，FLL 由内部参考时钟控制。当通过 RDIV 位选择时，FLL 环将锁住频率 512 次滤波频率。ICSLCLK 在 BDC 通信中可用，允许内部参考时钟。

9.4.1.2 FLL Engaed External(FEE)

当下列所有条件发生时才进入 FLL 外部使用（FEE）模式。

- 向 CLKS 位写 00。
- 向 IREFS 位写 0。
- 向 RDIV 位写 31.25kHz 到 39.0625kHz 范围的调整参考时钟。

在 FLL 外部使用模式中，ICSOUT 时钟从 FLL 时钟得到，FLL 由外部参考时钟控制。当通过 RDIV 位选择时，FLL 环将锁住频率 512 次滤波频率。ICSLCLK 在 BDC 通信中可用，允许外部参考时钟。

9.4.1.3 FLL旁路内部(FBI)

当下列所有条件发生时才进入 FLL 旁路内部(FBI)模式。

- 向 CLKS 位写 01。

- 向 IREFS 位写 1。
- BDM 有效且向 LP 位写 0。

在 FLL 旁路内部模式中, ICSOUT 时钟从内部参考时钟得到, FLL 由内部参考时钟控制。当通过 RDIV 位选择时, FLL 环将锁住频率 512 次滤波频率。ICSLCLK 在 BDC 通信中可用, 允许内部参考时钟。

9.4.1.4 FLL旁路内部低功耗(FBILP)

当下列所有条件发生时才进入 FLL 旁路内部低功耗(FBILP)模式。

- 向 CLKS 位写 01。
- 向 IREFS 位写 1。
- BDM 无效且向 LP 位写 1。

在 FLL 旁路内部低功耗模式中, ICSOUT 时钟从内部参考时钟得到, 禁止 FLL。ICSLCLK 在 BDC 通信中可用, 允许内部参考时钟。

9.4.1.5 FLL旁路外部(FBE)

当下列所有条件发生时才进入 FLL 旁路外部(FBE)模式。

- 向 CLKS 位写 10。
- 向 IREFS 位写 0。
- BDM 有效且向 LP 位写 0。

在 FLL 旁路外部模式中, ICSOUT 时钟从外部参考时钟得到, 由外部参考时钟控制。当通过 RDIV 位选择时, FLL 环将锁住频率 512 次滤波频率。ICSLCLK 在 BDC 通信中可用, 允许外部参考时钟。

9.4.1.6 FLL旁路外部低功耗(FBELP)

当下列所有条件发生时才进入 FLL 旁路外部低功耗(FBELP)模式

- 向 CLKS 位写 10。
- 向 IREFS 位写 0。
- BDM 模式无效且向 LP 位写 1。

在 FLL 旁路外部低功耗模式中, ICSOUT 时钟从外部参考时钟得到并且禁止 FLL。ICSLCLK 在 BDC 通信中不可用。允许外部参考时钟。

9.4.1.7 Stop(STOP)

只要当 MCU 进入 STOP 状态就进入 stop 模式, 在该模式中, 所有的 ICS 时钟信号都是静态的, 处理下面的情况:

当下列所有的条件都发生时, ICSIRCLK 在 stop 模式下仍然有效

- 向 IRCLKEN 位写 1。
- 向 IREFSTEN 位写 1。

当下列所有的条件都发生时, ICSECLK 在 stop 模式下仍然有效

- 向 ERCLKEN 位写 1。
- 向 EREFSTEN 位写 1。

9.4.2 模式转换

当在 FLL 内部使用(FEI)和 FLL 外部使用(FEE)模式之间转换时, IREFS 位可以随时改变, 但是 RDIV 位被同步改变, 这样产生的频率维持在 31.25kHz 到 39.0625kHz 的范围。IREFS 的值改变后, 在一些产生的分频参考频率周期后, FLL 将开始再次锁定。

CLKS 位可以随时改变，但是 RDIV 位被同步改变，这样产生的频率维持在 31.25kHz 到 39.0625kHz 的范围。直到一些新时钟周期后，才会发生转换到新的选择的时钟的动作。如果新选择的时钟不可用，仍然选择一起的时钟。

9.4.3 总线频率分频因子

BDIV 可以随时改变，转换到新频率的动作将会立即发生。

9.4.4 模式转换

低功耗位 (LP) 可以禁止 FLL，这样就可以在 FLL 不使用时减少消耗。然而在一些应用中，需要使能 FLL，在转换到一个 FLL 使用模式时允许 FLL 锁定最大精度。通过向 LP 写 0 来完成。

9.4.5 内部参考时钟

当 IRCLKEN 置位时，内部参考时钟信号作为 ICSIRCLK，可以作为其他的时钟源来使用。ICSIRCLK 频率可以通过调整内部参考时钟的周期重新对齐。这可以通过向 ICSTRM 寄存器的 TRIM 位写新值来完成。写更大的值将减小 ICSIRCLK 频率，写更小的值将增大 ICSIRCLK 频率。如果 ICS 处于 FLL 内部使用(FEI)、FLL 旁路内部(FBI) or FLL 旁路内部低功耗(FBILP)模式中，TRIM 位将影响 ICSOUT 频率。复位不影响 TRIM 和 FTRIM 的值。

直到 ICSIRCLK 被调整，可编程的低参考分频 (RDIV) 因子可能导致 ICSOUT 频率超过最大的片级 (Chip-level) 频率和违反片级时钟调整描述 (参见“设备概述”一章)

如果 IREFSTEN 置位并且向 IRCLKEN 位写 1，内部参考时钟在 stop 模式中将保持运行，这样的话，在退出 stop 时，可以快速恢复。

所有的 MCU 设备在工厂都在保留的内部地址中写入了一个调整值。当复位初始化时，该值可以复制到 ICSTRM 寄存器。该调整值不包括 FTRIM 位。为了更高的精度，在应用中，用户可以调整内部振荡器，设置合适的 FTRIM 位。

9.4.6 可选的外部参考时钟

ICS 模块的所有模式都支持 31.25kHz 到 5MHz 的外部参考时钟。当 ERCLKEN 置位时，外部参考时钟信号作为 ICSECLK，可以作为其他的时钟源来使用。当 IREFS=1 时，外部参考时钟不能被 FLL 使用，仅作为 ICSECLK 时钟。在这些模式中，频率可以等于片级调整说明所支持的最大频率 (参见“设备概述”一章)。

如果 EREFSTEN 置位并且向 ERCLKEN 位写 1，外部参考时钟在 stop 模式中将保持运行，这样的话，在退出 stop 时，可以快速恢复。

9.4.7 固定频率时钟

ICS 使用分频的 FLL 参考时钟作为 ICSEFCLK，ICSEFCLK 可以作为外围模块的时钟源使用。ICS 提供一个输出信号 (ICSFFE)，标识 ICS 正在提供的 ICSOUT 频率四倍或者大于四倍分频 FLL 参考时钟 (ICSFFLCLK)。在 FLL 使用(FEI and FEE)模式中，这总是真且 ICSFFE 总是高。在 ICS bypass 模式中，ICSFFE 将通过 BDIV 和 RDIV 的值的组合来确定。

- BDIV=00(除以 1)，RDIV $\geq$ 010
- BDIV=01(除以 2)，RDIV $\geq$ 011
- BDIV=10(除以 4)，RDIV $\geq$ 100

- BDIV=11(除以 8), RDIV $\geq$ 101

9.5 模块初始化

该节描述如何初始化和配置 ICS 模块。接下来的章节包含了两个初始化的例子。

9.5.1 ICS模块初始化顺序

ICS 有 POR 产生, 被配置为 FEI 模式, BDIV 设置位 2 分频。在 FLL 活动锁前, 内部参考将在 t_{IRST} 微秒内稳定。只要内部参考稳定了, FLL 将在 $t_{Acquire}$ 微秒内获得锁。

在 POR 之上, 内部参考将需要调整来确保准确的时钟。Freescale 推荐使用 FLASH 地址 0xFFAE 来存放 ICSSC 寄存器中的微调位 (FTRIM), 0xFFAF 存放 ICSTRM 寄存器的 8 位调整值。MCU 不会自动的复制这些 FLASH 地址中的值到各自寄存器。因此用户代码必须从 FLASH 复制这些值到寄存器中。

注:

如果不首先对齐内部参考, BDIV 的值不能改变成 1 分频。如果这样做可能导致 MCU 偏离规定。

9.5.1.1 初始化序列, 内部时钟模式-外部时钟模式

为了从 FEI 或 FBI 时钟模式变到 FEE 或 FBE 时钟模式, 按照下面的过程:

1. 通过设置 ICSC2 中的相应位使能外部时钟源。
 - 如果 FBE 是选择的模式, 在该步骤中也要置位 LP 以减少功耗。
2. 如果有必要, 等待外部时钟源稳定。典型的晶振启动时间在电气特性附录中给出。如果 EREFS 在步骤 1 中置位, 只要振荡器完成初始化周期, OSCINIT 位就将置位。
3. 写 ICSC1 选择时钟源。
 - 如果进入 FEE, 设置参考分频因子并且清零 IREFS 位转换到外部参考。
 - 通过设置 IRCLKEN 位内部参考可以选择保持运行。如果应用来回地转换内部时钟和外部时钟模式, 这很有用。为了最小化功耗, 当处于外部时钟模式时, 禁止内部参考。
4. 可以监控 CLKST 位来决定何时模式转换完成。如果选择 FEE, 总线时钟将在 $t_{Acquire}$ 微秒内稳定。当从 FEI 转换到 FEE, CLKST 位不改变。

9.5.1.2 初始化序列, 外部时钟模式-内部时钟模式

为了从 FEE 或 FBE 时钟模式变到 FEI 或 FBI 时钟模式, 按照下面的过程:

1. 如果保存了 TRIM 和 FTRIM, 将 FLASH 中 TRIM 和 FTRIM 复制到 ICSTRM 和 ICSSC 寄存器。这仅需要在 POR 后做一次。
2. 通过选择 ICSC1 中的 FBI (CLKS=0:1) 或选择 FEI (CLKS=0:0, RDIV=0:0:0 并且 IREFS=1), 允许内部时钟参考。
3. 等待内部时钟源稳定。典型的晶振启动时间在电气特性附录中给出。
4. 写 ICSC2 禁止外部时钟。
 - 通过置位 ERCLKEN 位外部参考可选择继续运行。如果应用来回地转换内部时钟和外部时钟模式, 这很有用。为了最小化功耗, 当处于内部时钟模式时, 禁止外部参考。
 - 如果 FBI 是所选择的模式, 在该步骤中也要置位 LP 位以减少功耗。

注:

在禁止外部时钟前, 内部参考必须被允许且运行。因此在执行步骤 4 前执行步骤 2 和 3 时很有必要的。

5. 可以监控 CLKST 位来决定何时模式转换完成。当从 FEE 转换到 FEI, CLKST 位不改

变。如果选择 FEI，总线时钟将在 tAcquire 微秒内稳定。

第十章 键盘中断（S08KBIV2）

10.1 简介

键盘中断 KBI 模块提供了高达 8 个独立使能的外部中断源。在 MC9S08QD4 系列 MCU 中，只有可能中断的 4 个（KBI1P4-KBI1P0）被实现。这些输入通过 KBIPE 位选择。

图 10-1 高亮显示了 MC9S08QD4 的 KBI 模块和引脚。

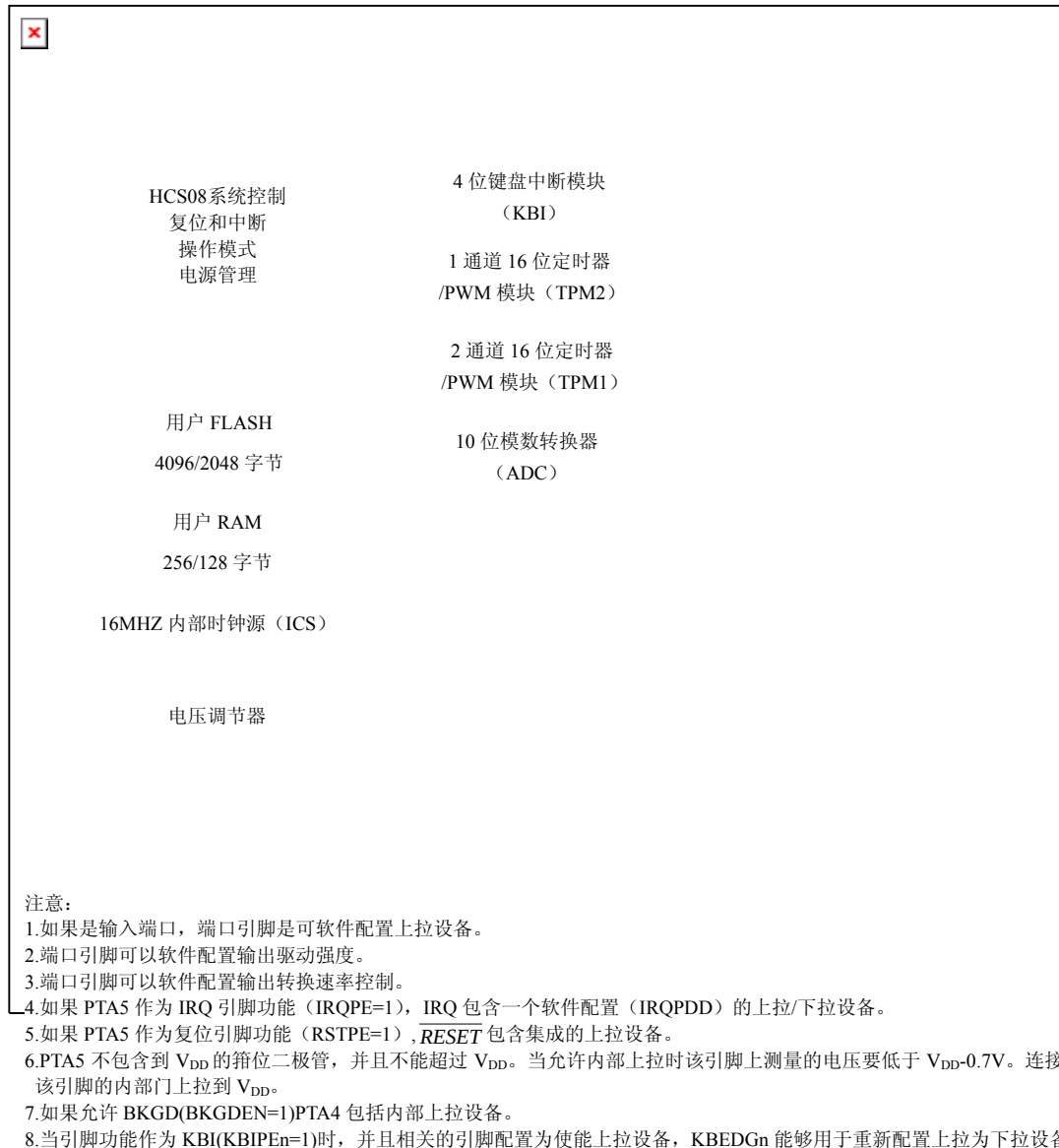


图10-1 MC9S08QD4系列框图高亮显示KBI框和引脚

表 10-1 信号属性

信号	功能	I/O
KBIPn	键盘中断引脚	I

10.3 寄存器定义

KBI 包括 3 个寄存器：

- 8 位引脚状态和控制寄存器。
- 8 位引脚使能寄存器。
- 8 位边沿选择寄存器。

KBI 寄存器的绝对地址参见本数据手册内存一章中的直接页寄存器的概述。这里通过这些寄存器和控制位的名称来引用它们。

某些 MCU 可能不只一个 KBI，所以寄存器的名字包括了前面的字符来表示所引用的是那个 KBI。

10.3.1 KBI状态和控制寄存器（KBISC）

KBISC 包括状态标志和控制位，用于配置 KBI。

	第7位	6	5	4	3	2	1	第0位
读					KBF	0	KBIE	KBIMOD
写						KBACK		
复位	0	0	0	0	0	0	0	0

图10-3 KBI状态和控制寄存器（KBISC）

表 10-2 KBISC 寄存器域描述

域	描述
7:4	未使用的寄存器位，读总为0。
3 KBF	键盘中断标志——KBF标识检测到了键盘中断。写不影响KBF。 0 没有检测到键盘中断。 1 检测到键盘中断。
2 KBACK	键盘中断确认——通过置1KBACK可以清KBF状态标志。读KBACK总为0。
1 KBIE	键盘中断使能——KBI确定是否有中断请求。 0 禁止中断请求。 1 允许中断请求。
KBIMOD	键盘检测模式——KBMOD（总是和KBEDG位一起）控制键盘中断引脚的检测模式。 0 键盘仅检测边沿。 1 键盘检测边沿和电平。

10.3.2 KBI引脚使能寄存器（KBIPE）

KBISC 包括引脚控制使能位。

	第7位	6	5	4	3	2	1	第0位
读	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
写								
复位	0	0	0	0	0	0	0	0

图10-4 KBI引脚使能寄存器（KBIPE）

表 10-3 KBIPE 寄存器域描述

域	描述
7:0 KBIPE[7:0]	键盘引脚使能位——每个KBIPEn位允许相应的键盘中断引脚。 0 引脚不能作为键盘中断输入。 1 引脚可以作为键盘中断输入。

10.3.3 KBI边沿选择寄存器（KBIES）

KBIES 包括边沿选择控制位。

	第7位	6	5	4	3	2	1	第0位
读	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBEDG3	KBEDG2	KBEDG1	KBEDG0
写								
复位	0	0	0	0	0	0	0	0

图10-5 KBI边沿选择寄存器（KBIES）

表 10-4 KBIES 寄存器域描述

域	描述
7:0 KBEDGn	键盘边沿选择位——每个KBEDGn位为相应的引脚选择下降沿/低电平后上升沿/高电平功能。 0 下降沿/低电平。 1 上升沿/高电平。

10.4 功能描述

该片上外围模块称为键盘中断（KBI）模块，因为起初它被设计用来简化键盘开关的行列举证的连接和使用。然而这些输入也可以作为额外的外部中断输入和从 stop 或等待低功耗模式中唤醒 MCU 的外部方法。

KBI 模块允许高达 8 个引脚作为额外的中断源。写键盘中断引脚使能寄存器（KBIPE）的 KBIPEn 位可以独立允许或禁止每个 KBI 引脚。根据键盘状态和控制寄存器（KBISC）的 KBMOD 位，每个 KBI 引脚都可以配置为边沿触发或边沿和电平触发。边沿触发可以通过软件编程为下降或上升；电平可以时高或低。边沿或边沿和电平的极性使用键盘中断边沿选择寄存器（KBIES）的 KBEDGn 位来选择。

10.4.1 仅边沿触发

同步逻辑用于检测边沿。在一个总线周期内，使能的键盘输入信号是逻辑 1（确认信号），下一个总线周期为逻辑 0（非确认信号），则下降沿被检测到。在一个总线周期内，使能的键盘输入信号是逻辑 0（非确认信号），下一个总线周期为逻辑 1（确认信号），则上升沿被检测到。检测都任何边沿后，在检测到新的边沿前，所有允许的键盘中断信号必须返回到非确认信号。

使能的 KBI 引脚上的有效边沿将置位 KBISC 的 KBF。如果 KBISC 的 KBIE 置位，将会向 CPU 产生一个中断请求。向 KBISC 的 KBACK 写 1 可以完成清零 KBF。

10.4.2 边沿和电平触发

使能的 KBI 引脚上的有效边沿将置位 KBISC 的 KBF。如果 KBISC 的 KBIE 置位，将会向 CPU 产生一个中断请求。如果所有的使能的键盘输入都在非确认电平上，向 KBISC 的 KBACK 写 1 可以完成清零 KBF。。如果任何使能的键盘输入在确认电平上，试图向 KBISC 的 KBACK

写1来清零KBF时，KBF仍然置位。

10.4.3 KBI上拉/下拉电阻

通过相应的I/O端口上拉使能寄存器，KBI引脚可以配置为使用内部上拉/下拉电阻。如果允许内部点最，KBIES寄存器用来选择电阻是上拉（KBEDGn=0）还是下拉(KBEDGn=1)。

10.4.4 KBI初始化

当最初允许键盘中断引脚时，可能会得到一个假的键盘中断标志。在初始化期间，为了避免假的中断请求，用户必须按照下面的做：

- 1.通过清零KBISC的KBIE不允许键盘中断。
- 2.通过设置KBIES的适当的KBEDGn位允许KBI的极性。
- 3.如果使用内部上拉/下拉设备，配置PtxPE中相应的使能位。
- 4.通过设置KBIPE的适当KBIPEn位，使能KBI引脚。
- 5.写KBISC的KBACK清零所有的假标志。
- 6.置位KBISC的KBIE允许中断。

第十一章 定时器/脉宽调制(S08TPMV2)

11.1 简介

图 11-1 高亮显示了 MC9S08QD4 系列的 TPM 模块和引脚。

11.1.1 TPM2配置信息

TPM2 模块包括一个单通道，TPM2CH0(和输入引脚 PTA4 与输出引脚 PTA5 配合使用)。当 TPM2 配置为输入捕捉，TPM2CH0 将连到 PTA5 (TPM2CH0I)。当 TPM2 配置为输出比较，TPM2CH0 将连接到 PTA4 (TPM2CH0O)。当禁止 TPM2，PTA4 和 PTA5 作为标准端口。

11.1.2 TCLK1和TCLK2配置信息

TCLK1 和 TCLK2 分别作为 TPM1 和 TPM2 的外部时钟源输入。



注意:

- 1.如果是输入端口，端口引脚是可软件配置上拉设备。
- 2.端口引脚可以软件配置输出驱动强度。
- 3.端口引脚可以软件配置输出转换速率控制。
- 4.如果 PTA5 作为 IRQ 引脚功能 (IRQPE=1)，IRQ 包含一个软件配置 (IRQPDD) 的上拉/下拉设备。
- 5.如果 PTA5 作为复位引脚功能 (RSTPE=1)， $\overline{RESET}$ 包含集成的上拉设备。
- 6.PTA5 不包含到 V_{DD} 的箝位二极管，并且不能超过 V_{DD} 。当允许内部上拉时该引脚上测量的电压要低于 $V_{DD}-0.7V$ 。连接到该引脚的内部门上拉到 V_{DD} 。
- 7.如果允许 BKGD(BKGDEN=1)PTA4 包括内部上拉设备。
- 8.当引脚功能作为 KBI(KBIPEn=1)时，并且相关的引脚配置为使能上拉设备，KBEDGn 能够用于重新配置上拉为下拉设备。

图11-1 MC9S08QD4系列框图高亮显示TPM框和引脚

11.1.3 特征

TPM 模块包括下列特征:

- 每个 TPM 的所有通道都配置为带缓存，中心对齐脉宽调制 CPWM (center-aligned pluse-width modulation)。
- 每个 TPM (若干 TPM 设备) 独立选择时钟源。
- 可选的时钟源 (依赖于设备): 总线，固定系统时钟，外部引脚。
- 时钟预分频除以 1、2、4、8、16、32、64、128。
- 16 位自由运行或向上/向下 (CPWM) 计数操作。
- 16 位预置寄存器控制技术范围。
- 定时器使能。
- 每个 TPM 模块 (若干 TPM 设备) 的每个通道都有一个中断，加上一个定期的计数中断。

- 通道特征：
 - 每个通道都有输入捕捉，输出比较，带缓冲中心对齐 PWM。
 - 上升沿，下降沿，任意边沿输入捕捉触发。
 - 输出比较动作包括置位，清零，翻转。
 - PWM 输出可选择极性。

11.1.4 框图

图 11-2 显示了 TPM 的结构。一些 MCU 包含超过一个的 TPM，不同数量的通道。

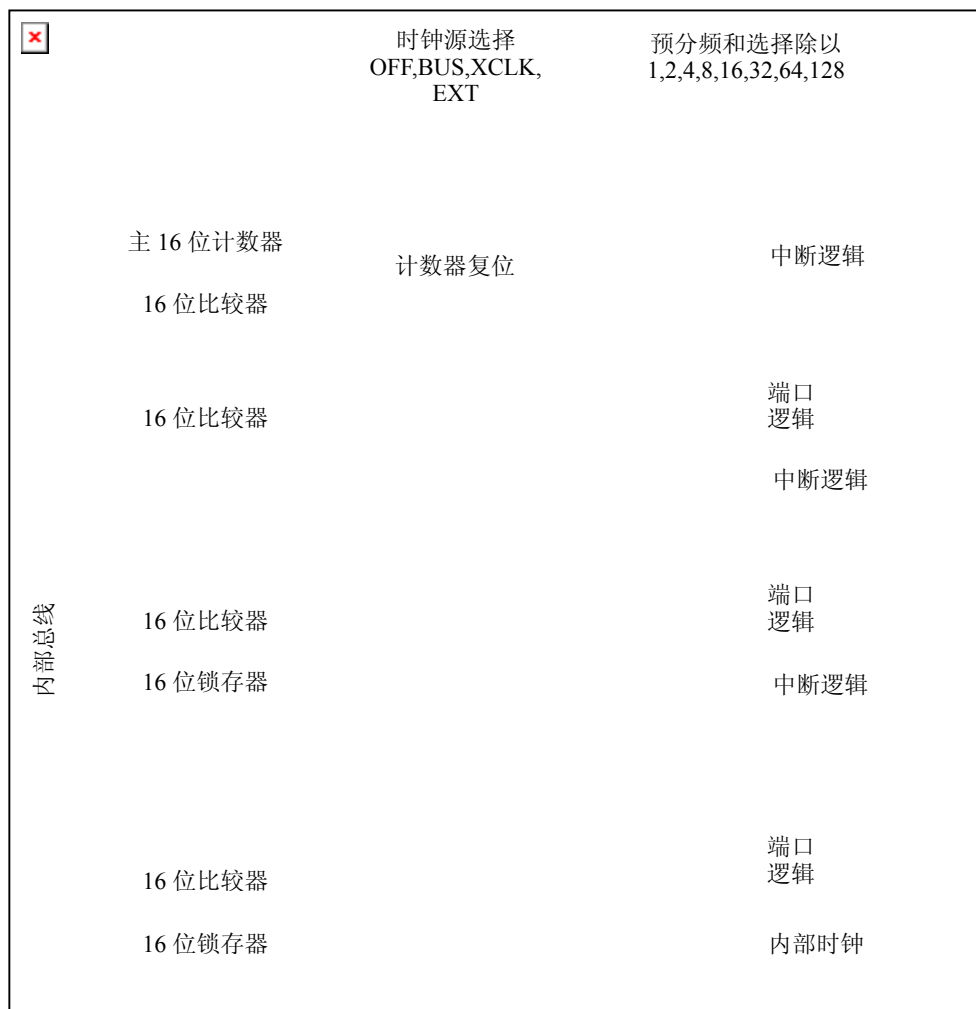


图11-2 TPM框图

TPM 的中心部件是 16 位计数器，当配置为中心边沿 PWM 时，可以作为一个自由运行计数器，预置计数器，向上/向下计数器。TPM 计数器（当处于正常加 1 操作时）提供输入捕捉，输出比较，和中心边沿 PWM 功能的时序参考。定时器预置计数器，TPMxMODH:TPMxMODL，控制计数器的预置值。（0x0000 或 0xFFFF 使计数器自由运行。）软件可以随时读取计数值而不影响计数时序。向 TPMxCNT 的任意字节写入，计数器都会复位，除非数据值已经写完了。

所有的 TPM 通道都可编程，可以独立作为输入捕捉，输出比较，或带缓冲中心对齐 PWM 通道。

11.2 外部信号描述

当和定时器相关的引脚配置为定时器输入时。被动的上拉使能。复位后，禁止 TPM 模块，所有的引脚默认禁止上拉的通用输入。

11.2.1 外部TPM时钟源

当定时器状态和控制寄存器中的控制位 CLKSB:CLKSA 设置成 1:1 时，16 位计数器 TPMx 的预分频器和结果被连到 I/O 引脚上的外部时钟源 TPMxCLK 驱动。外部时钟和 TPM 之间需要一个同步器。这个同步器的时钟使用总线时钟，所以外部时钟的频率必须小于总线时钟的一半。这个外部时钟的最高频率限制规定为总线时钟的四分之一，如此谨慎可以容忍占空比和锁相环或锁频环的频率抖动影响。

在某些设备中，外部时钟输入和一个 TPM 通道复用。当 TPM 通道复用为外部时钟输入时，相应的 TPM 通道不能使用引脚。（通道仍可在输出比较模式中作为软件定时器使用）。也就是说，如果一个 TPM 通道作为外部时钟输入使用，相应的 ELSnB:ELSnBA 控制位必须设置位 0:0，所以通道不能使用相同的引脚。

11.2.2 TPMxCHn——TPMx通道n I/O引脚

每个 TPM 通道都和 MCU 的 I/O 引脚相关。引脚的功能依赖于通道的配置。在某些情况，没有引脚使用，所以引脚用于通用 I/O 控制。当定时器控制端口引脚时，端口数据和方向寄存器都对相应的引脚没有作用。更多关于引脚功能复用的信息参见“引脚和连接”一章。

11.3 寄存器定义

TPM 包括：

- 一个 8 位状态和控制寄存器（TPMxSC）。
- 一个 16 位计数寄存器（TPMxCNTH:TPMxCNTL）。
- 一个 16 位预置寄存器（TPMxMODH:TPMxMODL）。

每个定时器通道包括：

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

关于所有 TPM 寄存器的绝对地址分配参见该数据手册中的“存储器”一章中的直接页寄存器概述。这里通过这些寄存器和控制位的名称来引用它们。通常，Freescale 提供一个头文件把它们的名称翻译为绝对地址。

11.3.1 定时器状态和控制寄存器（TPMxSC）

TPMxSC 包括用于配置中断使能的溢出状态和控制位，TPM 配置，时钟源，预分频因子。这些控制和该定时器模块的所有通道都有关系。

	第7位	6	5	4	3	2	1	第0位
读	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
写								
复位	0	0	0	0	0	0	0	0

图11-3 定时器状态和控制寄存器（TPMxSC）

表 11-1 TPMxSC 寄存器域描述

域	描述
7 TOF	定时器翻转标志——当TPM计数器达到TPM计数器预置寄存器所编程的预置值，TPM计数器变成0x0000时，该位置位。当TPM配置为CPWM，计数器达到预置寄存器的值时，TOF置位，并转换到下一个更低的计数值。当TOF置位时通过读TPM状态和控制寄存器然后向TOF写0，可以清零TOF。如果在清零序列完成前另一个翻转发生，则序列复位，所以在更早的TOF完成清零序列后，TOF仍然置位。复位会清零TOF。向TOF写1没有任何影响。 0 TPM计数器未达到预置数或翻转。 1 TPM已经翻转。
6 TOIE	定时器溢出中断使能——该位可读写。允许TPM溢出中断。如果置位TOIE，当TOF等于1时产生中断。复位清零TOIE。 0 禁止TOF中断（使用软件轮询）。 1 允许TOF中断。
5 CPWMS	中心对齐PWM选择——该位可读写。选择CPWM操作模式。复位清零该位，所以TPM以向上计数模式操作输入捕捉，输出比较，边沿对齐PWM功能。置位CPWMS重新配置TPM以向上/向下计数模式操作CPWM功能。复位清零CPWMS。 0 所有的TPMx通道通过各自的通道状态和控制寄存器的MSnB:MSnA控制位来选择输入捕捉，输出比较，边沿对齐PWM模式。 0 所有的TPMx通道处于中心对齐PWM模式操作。
4:3 CLKS[B:A]	时钟源选择——如图11-2所示，这2位用于禁止TPM系统或选择3个时钟源中的一个来驱动计数器预分频。外部源和XCLK通过片上同步电路和总线时钟同步。
2:0 PS[2:0]	分频因子选择——这3位为TPM时钟输入选择图11-3所示的8个因子中的一个。这个分频在时钟源同步或时钟源选择后确定。无论TPM选择什么时钟源驱动TPM系统，它都会影响。

表 11-2 TPM 时钟源选择

CLKSB:CLKSA	分频输入选择的时钟源
0:0	没有选择时钟(禁止TPMx)
0:1	总线时钟(BUSCLK)
1:0	固定系统时钟(XCLK)
1:1	外部源(TPMxCLK) ^{1,2}

1.外部时钟所允许的最大频率时总线频率的四分之一。

2.如果外部时钟输入和通道n，并且选择为TPM时钟源，相应的ELSnB:ELSnA控制位必须设置成0: 0，这样的话通道n就不会使用相同的引脚。

表 11-3 分频因子选择

PS2:PS1:PS0	时钟源分频因子
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

11.3.2 定时器计数寄存器（TPMxCNTH:TPMxCNTL）

两个只读 TPM 计数寄存器包括 TPM 计数器的高字节和低字节。读其中一个字节（TPMxCNTH 或 TPMxCNTL）都将两个字节的內容锁存一个缓冲区，一直锁存到读另一个字节。这可以使没有同时读的 16 位一致。这种一致性机制可以在 MCU 复位，写 TPMxCNTH 或 TPMxCNTL，写定时器状态/控制寄存器后自动重启。

复位清零 TPM 计数寄存器。

	第7位	6	5	4	3	2	1	第0位
读	15位	14	13	12	11	10	9	8
写	任何写TPMxCNTH都清零16位计数器							
复位	0	0	0	0	0	0	0	0

图11-4 定时器计数寄存器高 (TPMxCNTH)

	第7位	6	5	4	3	2	1	第0位
读	7位	6	5	4	3	2	1	0
写	任何写TPMxCNTL都清零16位计数器							
复位	0	0	0	0	0	0	0	0

图11-5 定时器计数寄存器低 (TPMxCNTL)

当背景调试有效时，定时器计数和一致性机制被冻结，这样的话，当背景模式有效，即使读一个或两个计数器，锁存仍然保持它们所处的状态。

11.3.3 定时器计数预置寄存器 (TPMxMODH:TPMxMODL)

两个只读 TPM 预置寄存器包括 TPM 计数器的预置值。TPM 计数器达到预置值时，TPM 计数器在下一个周期 (CPWMS=0) 或开始继续向下计数 (CPWMS=1) 时从 0x0000 重新计数，并且翻转标志 (TOF) 置位。写 TPMxMODH 或 TPMxMODL 时禁止 TOF 和翻转中断直到两个都写完。复位设置 TPM 计数器预置寄存器为 0x0000，将会导致自由运行计数器 (禁止预置)。

	第7位	6	5	4	3	2	1	第0位
读	15位	14	13	12	11	10	9	8
写	任何写TPMxCNTH都清零16位计数器							
复位	0	0	0	0	0	0	0	0

图11-6 定时器预置寄存器高 (TPMxMODH)

	第7位	6	5	4	3	2	1	第0位
读	7位	6	5	4	3	2	1	0
写	任何写TPMxCNTL都清零16位计数器							
复位	0	0	0	0	0	0	0	0

图11-7 定时器预置寄存器低 (TPMxMODL)

通常等待一个翻转中断这样预置寄存器的两部分可以在新的翻转中断前完成。替代方法是写 TPM 预置寄存器前复位 TPM 计数器，以避免第一个计数器翻转发生时发生冲突。

11.3.4 定时器通道n状态和控制寄存器 (TPMxCnSC)

TPMxCnSC 包含通道中断状态标志和用于配置中断使能的控制位，通道配置，引脚功能。

	第7位	6	5	4	3	2	1	第0位
读	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
写								
复位	0	0	0	0	0	0	0	0

图11-8 定时器通道n状态和控制寄存器 (TPMxCnSC)

表 11-4 TPMxCnSC 寄存器域描述

域	描述
---	----

7 CHnF	通道n标志——当通道n配置为输入捕捉时，若在通道n引脚上产生了有效边沿，则该标志置位。当通道n作为输出比较或边沿对齐PWM通道时，若TPM计数器寄存器的值和TPM通道n的值寄存器相匹配时，则CHnF置位。该标志在中心对齐PWM中很少使用，因为每次计数值和通道值寄存器相匹配就会置位，这的有效占空比的两个边沿相同。 当CHnF置位且允许中断时（CHnIE=1），产生相应的中断请求。当CHnF置位时读TPMxCnSC然后写0到CHnF，可以清零CHnF。如果在清零序列完成前另一个中断请求发生，则序列复位，所以在更早的CHnF完成清零序列后，CHnF仍然置位。这样做的话，清零以前的CHnF不会丢失CHnF上的中断请求。复位会清零CHnF。向CHnF写1没有任何影响。 0 在通道n上没有输入捕捉或输出比较事件发生。 1 在通道n上发生输入捕捉或输出比较事件。
6 CHnIE	通道n中断使能——该位可读写。允许通道n上的中断。复位清零CHnIE。 0 禁止通道n的中断请求（使用软件轮询）。 1 允许通道n的中断请求。
5 MSnB	TPM通道n的模式选择B——当CPWMS=0，MSnB=1，配置TPM通道n为边沿对齐PWM模式，关于通道模式和控制参见表11-5。 0 所有的TPMx通道通过各自的通道状态和控制寄存器的MSnB:MSnA控制位来选择输入捕捉，输出比较，边沿对齐PWM模式。 0 所有的TPMx通道处于中心对齐PWM模式操作。
4 MSnA	TPM通道n的模式选择A——当CPWMS=0，MSnB=0，配置TPM通道n为输入捕捉或输出比较模式，关于通道模式和控制参见表11-5。
3:2 ELSn[B:A]	边沿/电平选择位——依赖于定时器通道的操作模式，如表中CPWMS:MSnB:MSnA的设置，这些位选择了触发一个输入捕捉事件所需要的输入边沿的极性，选择输出比较匹配时的驱动电平，或选择PWM输出的极性。 设置ELSnB:ELSnA为0:0则会配置相应的定时器引脚作为通用I/O，和任何定时器通道功能都没有关系。该功能用于临时禁止输入捕捉通道，或当相应的定时器通道设置为软件定时器而不需要使用引脚时，使定时器引脚作为通用I/O引脚，。

表11-5 模式，边沿和电平选择

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00		引脚不作为TPM通道使用，作为TPM的外部时钟使用或作为通用I/O使用。
0	00	01	输入步骤	仅捕捉上升沿
		10		仅捕捉下降沿
		11		仅捕捉上升沿或下降沿
	01	00	输出比较	仅软件比较
		01		翻转输出比较
		10		清零输出比较
1	1X	11	边沿对齐 PWM	设置输出比较
		10		高脉冲(清零输出比较)
		X1		低脉冲(设置输出比较)
		X1		高脉冲(清零输出比较)
1	XX	10	中心对齐 PWM	低脉冲(设置输出比较)
		X1		高脉冲(设置输出比较)

改到输入捕捉模式前，如果相应的端口引脚之上有两个总线周期不稳定，可能会得到一个所不希望的边沿触发。通常该变了通道配置位后而且允许通道中断前或使用状态标志前，编程清零状态标志可以避免意外动作。

11.3.5 定时器通道值寄存器（TPMxCnVH:TPMxCnVL）

这些可读写寄存器包括输入捕捉功能的TPM计数器捕捉值或输出比较或PWM功能的输出比较值。通道值寄存器通过复位清零。

	第7位	6	5	4	3	2	1	第0位
读 写	15位	14	13	12	11	10	9	8

复位	0	0	0	0	0	0	0	0
----	---	---	---	---	---	---	---	---

图11-9 定时器值寄存器高 (TPMxCnVH)

	第7位	6	5	4	3	2	1	第0位
读	7位	6	5	4	3	2	1	0
写	7位	6	5	4	3	2	1	0
复位	0	0	0	0	0	0	0	0

图11-10 定时器值寄存器低 (TPMxCnVL)

在输入捕捉模式中，读任何一个 (TPMxCnVH 或 TPMxCnVL) 锁存这两个字节的内容到一个缓冲区，将会锁存到有其他字节被读。当写 TPMxCnSC 寄存器时，该锁存机制也会复位 (不锁存)。

在输出比较模式中，写任何一个 (TPMxCnVH 或 TPMxCnVL) 锁存这两个字节的内容到一个缓冲区，当两个值都写完了，把它们作为一个整体的 16 位值传到定时器通道寄存器。当写 TPMxCnSC 寄存器时，该锁存机制也可以被人为复位。

这种锁存机制使没有同时写 16 位值一致，可以在不同的编译器中实现。

11.4 功能描述

所有的 TPM 功能都和主 16 位计数器相关，它允许时钟源和分频因子的灵活选择。在 TPM 中 16 位预置寄存器也和主 16 位计数器相关。每个 TPM 通道都可以选择和 MCU 引脚相关和中断功能。

TPM 有中心对齐功能，由 TPMxSC 的 CPWMS 控制位控制。当 CPWMS 设置为 1 时，定时器计数器 TPMxCNT 改变成一个向上/向下计数并且所有和 TPM 有关的通道都作为中心对齐 PWM 通道。当 CPWMS=0 时，每个通道都独立配置为输入捕捉操作，输出比较或带缓冲边沿对齐 PWM 模式。

接下来的几节将描述主 16 位计数器和各种定时器操作模式 (输入捕捉，输出比较，边沿对齐 PWM，中心对齐 PWM)。因为引脚操作和中断行为的细节依赖于操作模式，这些主题将包含在相关的模式一节。

11.4.1 计数器

所有的定时器功能都是基于主 16 位计数器 (TPMxCNTH:TPMxCNTL)。该节讨论时钟源，向上计数和向上/向下计数，计数后翻转，手动计数器复位。

任何 MCU 后，CLKSB:CLKSA=0:0，所以没有选择时钟源并且 TPM 无效。通常 CLKSB:CLKSA 设置成 0:1 所以总线时钟驱动定时器计数器。TPM 的时钟源可以选择关，总线时钟 (BUSCLK)，固定系统时钟 (XCLK)，或外部输入。外部时钟选项所允许的最大频率时总线的四分之一。更多关于时钟源选择的信息参见 11.3.1 节“定时器状态和控制寄存器 (TPMxSC)”和表 11-2。

当微控制器处于有效地背景模式时，TPM 临时挂起所有的计数直到微控制器返回到正常用户操作模式。在 stop 模式时，所有的 TPM 时钟停止；因此 TPM 被禁止直到时钟恢复。在等待模式，TPM 继续正常操作。

主 16 位计数器有两种计数模式。当选择中心对齐 PWM (CPWMS=1)，计数器操作处于向上/向下计数模式。否则计数器作为一个简单的向上计数操作。作为一个向上计数器，主 16 位计数器从 0x0000 开始计数到结束计数然后又从 0x0000 继续。结束计数可以是 0xFFFF 或 TPMxMODH:TPMxMODL 中的预置值。当规定为中心对齐 PWM 操作时，定时器从 0x0000 向上计数到结束计数，然后再向下计数到 0x0000，在向上计数。0x0000 和结束计数值 (在

TPMxMODH:TPMxMODL 中的值) 都是正常长度计数 (一个定时器周期长度)。

中断标志和使能也和主 16 位计数器相关。定时器翻转标志 (TOF) 是一个软件可读取标志, 标识定时器计数溢出。使能信号选择软件轮询 (TOIE=0) (没有硬件中断产生), 或者中断驱动操作 (TOIE=1) (只要 TOF 标志为 1 则自动产生一个静态硬件中断)。

导致 TOF 置位的条件依赖于操作模式 (向上或向上/向下)。在向上计数模式, 主 16 位计数器从 0x0000 计数到 0xFFFF 并且溢出为下一个计数周期的 0x0000。从 0xFFFF 转换到 0x0000 时 TOF 置位。当置位预置限制时, 从预置寄存器设置的值转换到 0x0000 时 TOF 置位。当主 16 位计数器处于向上/向下操作模式时, 当计数器改变方向从预置寄存器设置的值到下一个更低计数值时, TOF 标志置位。这符号 PWM 周期的结束。(0x0000 计数值对应周期的中心)

因为 HCS08MCU 是 8 位体系结构, 在定时器计数器的读操作中存在一致性机制。只要计数器中有一个字节需要读 (TPMxCNH 或 TPMxCNTL), 两个字节都被捕捉到缓冲区, 直到另一个字节也被读完。这将确保在读第一个字节时的第二个字节。计数器继续正常计数, 但是没有新的值从其中一个字节读, 直到以前计数器的两个字节都读完。

在任何时候, 向 TPMxCNH 或 TPMxCNTL 的任何一个字节写任意值都会复位主定时器计数器。万一在复位计数前只读了计数器的一个字节, 这种方式复位计数器也要复位一致性机制。

11.4.2 通道模式选择

如果 CPWMS=0 (不规定边沿对齐 PWM 操作), 通道 n 状态和控制寄存器的 MSnB 和 MSnA 控制位决定相应通道的基本操作模式。选择包括输入捕捉, 输出比较, 和带缓冲边缘对齐 PWM。

11.4.2.1 输入捕捉模式

输入捕捉功能, TPM 可以捕捉外部事件的发生。当一个有效边沿在输入捕捉通道的引脚上产生时, TPM 锁存 TPM 计数器的内容到通道值寄存器 (TPMxCnVH:TPMxCnVL)。上升沿, 下降沿, 或任何边沿都可以作为触发输入捕捉的有效边沿。

当读 16 位捕捉寄存器中的一个字节时, 两个字节都被锁存到缓冲区, 支持 16 位访问的一致而不关心顺序。通过写通道状态和控制寄存器 (TPMxCnSC), 可以人工复位一致行序列。

输入捕捉时间置位可选产生 CPU 中断请求的标志位 (CHnF)。

11.4.2.2 输出比较模式

对于输出比较功能, TPM 可以产生定时的脉冲, 位置, 极性, 持续时间和频率都是可编程的。当计数器达到输出比较通道的中通道值寄存器的值时, TPM 置位, 清零, 或翻转通道引脚。

在输出比较模式中, 只有在 16 位寄存器的两个 8 位寄存器都写完后才能将值传到相应的定时器通道值寄存器。通过写通道状态和控制寄存器 (TPMxCnSC) 可以人工复位这种一致性顺序。

输出比较事件置位可选产生 CPU 中断请求的标志位 (CHnF)。

11.4.2.3 边沿对齐 PWM 模式

这种类型的 PWM 输出使用定时器计数器的正常的向上计数模式 (CPWMS=0) 并且当相同的 TPM 的其他通道配置为输入捕捉或输出比较功能时仍然能够使用。该 PWM 信号的周期通过设置预置寄存器 (TPMxMODH:TPMxMODL) 来确定。占空比通过设置定时器通道值寄存器 (TPMxCnVH:TPMxCnVL) 来确定。PWM 信号的极性通过设置 ELSnA 控制位来确定。占空比可以选择 0% 到 100%。

如图 11-11 所示, TPM 通道寄存器的输出比较值决定了 PWM 信号的脉冲宽度 (占空比)。预置溢出和输出比较的时间就是脉冲宽度。如果 ELSnA=0, 计数器溢出强制 PWM 信号高并且输出比较强制 PWM 信号低。如果 ELSnA=1, 计数器溢出强制 PWM 信号低并且输出比较强制

PWM 信号高。

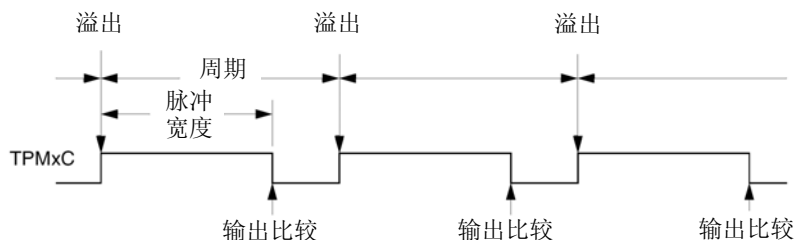


图11-11 PWM周期和脉宽 (ELSnA=0)

当通道值寄存器设置为 0x0000 时，占空比为 0%，通过设置定时器通道值寄存器 (TPMxCnVH:TPMxCnVL) 一个大于预置设置的值，可以得到 100%占空比。这隐含这预置设置必须小于 0xFFFF 才能得到 100%占空比。

因为 HCS08 是 8 位 MCU 系列，定时器通道寄存器的设置被缓存确保 16 位更新，避免不希望 PWM 脉宽。写 TPMxCnVH 或 TPMxCnVL 中的一个寄存器，写缓冲进存取。在边沿 PWM 模式中，只有在写了 16 位寄存器的两个字节 8 位并且 TPMxCNTH:TPMxCNTL 计数器的值时 0x0000，值才被传输到相应的定时器通道寄存器。(新的占空比直到下一个周期才产生作用。)

11.4.3 中心对齐PWM模式

这种类型的 PWM 输出使用定时器计数器的向上/向下计数模式 (CPWMS=1)。通过设置 TPMxCnVH:TPMxCnVL 的输出比较值确定 PWM 信号的占空比。设置 TPMxMODH:TPMxMODL 来确定周期。TPMxMODH:TPMxMODL 必须保持在 0x0001 到 0x7FFFF 的范围，因为超过这个范围将会产生模糊的结果。ELSnA 将确定 CPWM 输出的极性。

$$\text{脉冲宽度} = 2 \times (\text{TPMxCnVH:TPMxCnVL}) \quad (\text{等式 11-1})$$

$$\text{周期} = 2 \times (\text{TPMxMODH:TPMxMODL}) \quad (\text{等式 11-2})$$

其中 TPMxMODH:TPMxMODL=0x0001—0x7FFFF

如果通道值寄存器 TPMxCnVH:TPMxCnVL 是 0 或负数 (15 位置位)，占空比为 0%。如果 TPMxCnVH:TPMxCnVL 是一个整数 (15 位清零) 并且大于 (非零) 预置设置，占空比为 100%，因为不会产生占空比较。这隐含了通过预置寄存器设置的可用周期范围时 0x0001 到 0x7FFE (如果产生 100%不需要 0x7FFF)。因为产生的周期不比需要的正常应用更长，这不是重要的限制。

TPMxMODH:TPMxMODL=0x0000 是一个特殊值，不能用于中心对齐 PWM 模式。当 CPWMS=0 时，这种情况对应于计数器从 0x0000 自由运行到 0xFFFF，但是当 CPWMS=1，计数器需要有效的匹配预置寄存器而不是在 0x0000 来改变从向上计数到向下计数。

图 11-12 表明了 TPM 通道寄存器的输出比较值 (乘以 2)，TPM 通道寄存器决定 CPWM 信号的脉宽 (占空比)。如果 ELSnA=0，当向上计数强制 CPWM 信号低时比较匹配并且当向下计数强制输出高时比较匹配。计数器向上计数直到达到 TPMxMODH:TPMxMODL 设置的预置，然后向下计数直到 0。这样设置周期等于 TPMxMODH:TPMxMODL 的两倍。

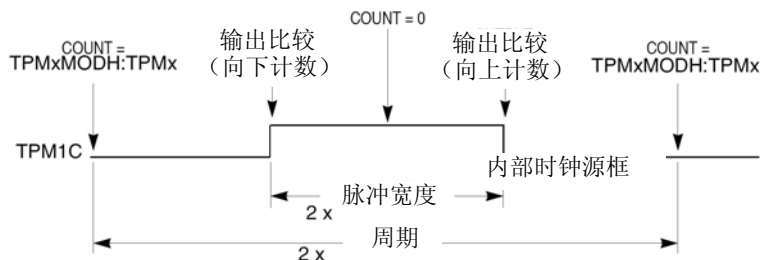


图11-12 CPWM周期和脉宽 (ELSnA=0)

因为在相同的系统时钟边沿会排列更少的 I/O 引脚转换，中心对齐 PWM 输出通常比边沿对齐 PWM 产生更小的噪音。这种 PWM 也需要一些类型的驱动。

应为HCS08是8位MCU系列，定时器通道寄存器的设置被缓存，确保16位更新一致，并且避免不希望的PWM脉宽。写TPMxMODH、TPMxMODL、TPMxCnVH和TPMxCnVL的任何一个寄存器，实际上都是写缓存寄存器。只有在16位寄存器的两个8位字节都写完并且定时器计数器翻转（在预置寄存器中的结束计数的最后，翻转方向从向上计数到向下计数）之后，值才被传到相应的定时器通道寄存器。

可选的，当TPMxMODH:TPMxMODL=TPMxCnVH:TPMxCnVL，在计数的最后TPM产生TOF中断。用户可以选择重新载入PWM缓存的任何数，并且它们在新周期的开始同步更新。

写 TPMxSC 将会取消写到 TPMxMODH 或/和 TPMxMODL 的值，并且会复位预置寄存器的一致性机制。写 TPMxCnSC 将会取消写到通道值寄存器的值，并且会复位 TPMxCnVH:TPMxCnVL 的一致性机制。

11.5 TPM中断

TPM 产生主寄存器溢出的可选中断和每个通道上的中断。通道中断的方式依赖于每个通道的操作模式。如果通道配置为输入捕捉，每次识别到选择的输入捕捉边沿，中断标志置位。如果通道配置为输出比较或PWM模式，每次主定时器计数器和16位通道值寄存器的值相匹配时，中断标志置位。关于绝对中断向量地址，极性和局部中断掩码控制位参见“复位，中断和系统配置”。

TPM 的每个中断源，当识别了中断条件例如定时器溢出，通道输入捕捉，或输出比较事件，则标志位置位。该标志可以通过软件读（轮询）来确保动作已将发生或相应的使能位（TOIE 或 ChnIE）被设置成允许硬件中断产生。当中断使能位置位，只要相应的中断标志等于1就会产生静态中断。用户软件有责任在从中断服务例程返回前，执行一系列步骤清零中断标志。

11.5.1 清零定时器中断标志

通过两个步骤的程序可以 TPM 中断标志清零，包括当它被置1时，写0到该位然后读标志位。如果在这两个步骤中检测到新的事件，序列被复位并且在第二个步骤后中断标志仍然置位，以避免丢失新事件的可能性。

11.5.2 定时器翻转中断描述

导致 TOF 置位的条件依赖于计数模式（向上或向上/向下）。在向上计数模式中，16位定时器计数器从0x0000到0xFFFF计数，并且在下一个计数周期溢出到0x0000。TOF在从0xFFFF转换到0x0000时置位。当定时器操作处于向上/向下计数模式时，当计数器方向从预置寄存器

设置的值转换到下一个更低计数值时，TOF 标志置位。这对应于 PWM 周期的结束。（0x0000 计数值对应周期的中心。）

11.5.3 通道事件中断描述

通道中断的方式依赖于通道的当前模式（输入捕捉，输出比较，边沿对齐 PWM 或中心对齐 PWM）。

当通道配置为输入捕捉通道时，ELSnB:ELSnA 控制位选择上升沿，下降沿，任何边沿，或无边沿（关）作为触发输入捕捉事件的边沿。当检测到选择的边沿，中断标志置位。通过在 11.5.1 节“清零定时器中断标志”所描述的两个步骤可以清零该标志。

当通道配置为输出比较通道时，每次主定时器计数器和 16 位的通道值寄存器相匹配时，中断标志置位。通过在 11.5.1 节“清零定时器中断标志”所描述的两个步骤可以清零该标志。

11.5.4 PWM 占空比结束时间

通道被配置为 PWM 操作，有两个可能性：

- 当通道配置为边沿对齐 PWM 时，每个 PWM 周期定时器计数器和标记有效占空比周期结束的通道值寄存器匹配两次，在 CPWM 情况下，在有效占空比的开始和结束通道标志置位，这正是定时器计数器和通道值寄存器相匹配的次数。
 - 当通道配置为中心对齐 PWM 时，当定时器计数器和相匹配时，通道标志置位。
- 通过在 11.5.1 节“清零定时器中断标志”所描述的两个步骤可以清零该标志。

第十二章 开发支持

12.1 介绍

HCS08 系列的开发支持系统包括背景调试控制器(BDC)。BDC 向目标 MCU 提供了一个单线调试接口, 该接口为片上 FLASH 和其他非易失性存储器的编程提供了便捷途径。同时, 对芯片开发以及对非介入性访问内存数据和传统调试特性(如 CPU 寄存器修改, 断点调试以及单步指令调试等)的支持来说, BDC 是主流调试接口。

在 HCS08 系列中, 地址和数据总线信号在外部引脚上不可被访问(即使在测试模式下)。调试是通过单线背景调试接口向目标 MCU 发送命令来完成的。调试模块提供了选择性触发和捕捉总线信息的手段, 因此外部开发系统能够重建 MCU 内部每个周期的活动, 而无需从外部对地址和数据信号进行访问。

12.1.1 强制背景调试

强制背景模式有效的方法依赖于具体的 HCS08 设备。对于 MC9S08QD4 系列, 通过保持 BKGD 引脚低, 当 MCU 退出复位时, 不依赖于导致复位的原因, 可以强制背景模式有效。如果没有调试点连接到 BKGD 引脚, MCU 将总是从正常操作模式复位。

12.1.2 模块配置

MC9S08QD4 的后备 BDC 时钟源时 ICGCLK, 更多关于 ICGCLK 和如何选择时钟源的信息参见第九章“内部时钟源(S08ICSV1)”。

12.1.3 特点

背景调试控制器的特点包括:

- 单引脚模式选择和背景通信。
- BDC 寄存器不与内存统一编址。
- 用 SYNC 命令决定目标通信速率。
- 用于内存访问的非介入式命令。
- 用于 CPU 寄存器访问的背景调试模式命令。
- GO 和 TRACE1 命令。
- BACKGROUND 命令能唤醒停止或等待模式下的 CPU。
- BDC 中建立硬件地址断点。
- 如果 BDC 使能, 晶振能运行在停止模式。
- 当处于背景调试模式时禁止 COP 看门狗。

12.2 背景调试控制器(BDC)

在 HCS08 系列中所有的 MCU 都包含有一个单线的背景调试接口, 它支持片上非易失性存储器的在线编程和复杂的非介入式的调试能力。与早期 8 位 MCU 的调试接口不同的是, 本系

统不会干涉正常的应用资源。它不使用任何用户存储空间或内存映像地址，而且不共享任何片上外围设备。

BDC 命令分成两组：

- 背景模式命令需要目标 MCU 处于背景调试模式(用户程序不在运行)。BACKGROUND 命令能让目标 MCU 进入背景调试模式。背景调试模式命令允许 CPU 寄存器能被读写，同时能允许用户一次跟踪一条用户指令，或发送 GO 命令从背景调试模式转到用户程序执行状态。
- 非介入式命令在任何时候都能执行，即使是在用户程序运行的时候。非介入式命令允许用户使用背景调试控制器读写 MCU 内存区域或访问状态和控制寄存器。

典型的是，一个简单的相关接口用于把来自主计算机的命令到用户串行接口命令转化成单线背景调试系统。根据开发工具厂商的不同，这个接口可使用标准的 RS232 串口、并行打印口或其他通信类型接口，如通用串行总线(USB)等通信接口来实现其与 PC 主机和之间的通信。这个接口一般通过地、BKGD 引脚、 $\overline{\text{RESET}}$ 有时还有 VDD 信号引脚与目标系统相连。一个开漏连接到复位允许主机能强制目标系统复位，对于恢复失去控制的目标系统或控制目标系统在片上的非易失性存储器编程之前复位是很有用的。有时候 VDD 用于允许接口使用目标系统电源而不需要附加的电源供电。然而如果它单独供电，可以连接到运行的目标系统而不用强制目标系统复位或影响正在运行的应用程序。

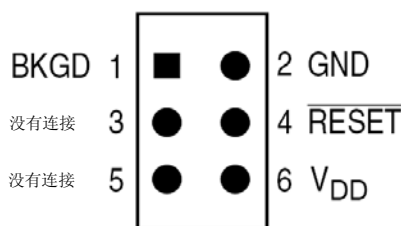


图12-1 标准BDM工具连接器

12.2.1 BKGD引脚描述

BKGD 是单线的背景调试接口引脚。这个引脚最主要的功能是能用于背景调试的指令和数据的双向串行通信。在复位期间，这个引脚用于选择背景模式还是用户应用程序。这个引脚同样用于发出定时同步脉冲来允许主机开发工具为背景调试串口通信校正时钟频率。

背景调试控制器(BDC)串行通信使用一种典型的串行协议，此协议首先被引进到 M68HC12 系列微控制器上。这个协议假定主机知道由目标 BDC 时钟率决定的通信的时钟频率。所有的通信被初始化并由主机控制，该主机还驱动一个下降沿来表示每一位的开始。命令和数据都从最高位开始发送。通信协议的具体描述参见 12.2.2 的“通信细节”。

如果主机试图与一个未知 BDC 时钟频率的目标 MCU 通信，可以向目标 MCU 发一个 SYNC 命令来请求一个时钟同步信号，以此主机能检测到正确的通信速率。

BKGD 虚开漏引脚，有一个片上上拉，所以需要外部上拉电阻。不像通常的开漏引脚，在该引脚上的外部 RC 时序固定，可以被外部的电容影响，几乎不影响信号的上升时序。典型的协议提供简短，实际的驱动速率脉冲来牵制该引脚上快速上升时序而不危害有害的驱动电平冲突。更多细节参见 12.2.2 的“通信细节”。

当没有调试头连接到 6 脚 BDM 接口连接器时，BKGD 的内部上拉选择正常的操作模式。当调试连接到 BKGD 时，复位后可能强迫 MCU 进入背景调试模式。轻拍背景调试的具体条件

依赖于 HCS08 设备（参见开发支持一节介绍）。为了和 MCU 使用背景调试接口通信没有必要复位 MCU。

12.2.2 通信细节

BDC 串行接口需要一个外部控制器在 BKGD 引脚上来产生一个下降沿，来表明每一位时间的开始。不管数据是否是发送还是接收，外部控制器始终产生下降沿。

BKGD 是一个虚开漏引脚，它通过外部控制器或 MCU 驱动。数据从最高位开始传送，每位需 16 个 BDC 时钟周期(正常速率)。如果 512 个 BDC 时钟周期发生在来自主机的下降沿之间则表示接口超时。当超时发生时任何 BDC 指令流被中止、以免影响内存或目标系统 MCU 的操作模式。

传统的串行协议需要调试器清楚目标 BDC 通信时钟的速率。

BDC 状态和控制寄存器的时钟选择 (CLKSW) 控制位允许用户选择 BDC 时钟源。BDC 时钟源要么时总线或后备 BDC 时钟源。

BKGD 引脚能收和发高低逻辑电平。下面图中表示每个状态的时序。接口时序与目标 BDC 同步，但与外部主机异步。内部 BDC 时钟信号按计数周期给出，谨供参考。

图 12-2 表示外部主机向目标 HCS08MCU 的 BKGD 引脚发送逻辑 1 或 0 信号。对于目标机来说主机是异步的，所以从主机产生下降沿到目标机感知到位开始有 0 ~ 1 个周期的延迟。10 个目标机 BDC 时钟周期后，目标机判断 BKGD 引脚上的位电平。典型的，在主到目标的发送期间，主机主动驱动虚开漏 BKGD 引脚来加速上升沿。因为在这个期间目标机不驱动 BKGD 引脚。所以在主机到目标机的发送过程中就可以忽略作为开漏信号的这条线。

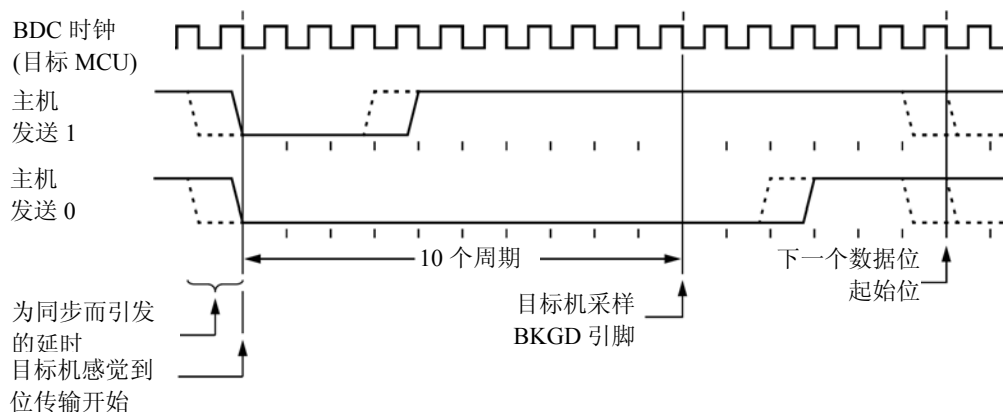


图12-2 BDC主机到目标机串行位时序

图 12-3 表示主机接收来自目标 HCS08MCU 逻辑 1 的图例。对于目标机来说主机是异步的，所以从主机在 BKGD 引脚上产生下降沿到目标机感知到位开始有 0 ~ 1 个周期的延迟。主机在 BKGD 引脚上长时间保持低电平以便目标机来识别它(至少两个目标 BDC 周期)。从目标 MCU 感知到位时间开始后的第七个周期，目标 MCU 将主动驱动一个简短的高电平加速脉冲，在此之前主机必须释放低电平驱动。主机在位时间开始后的第 10 个周期对位电平进行采样。

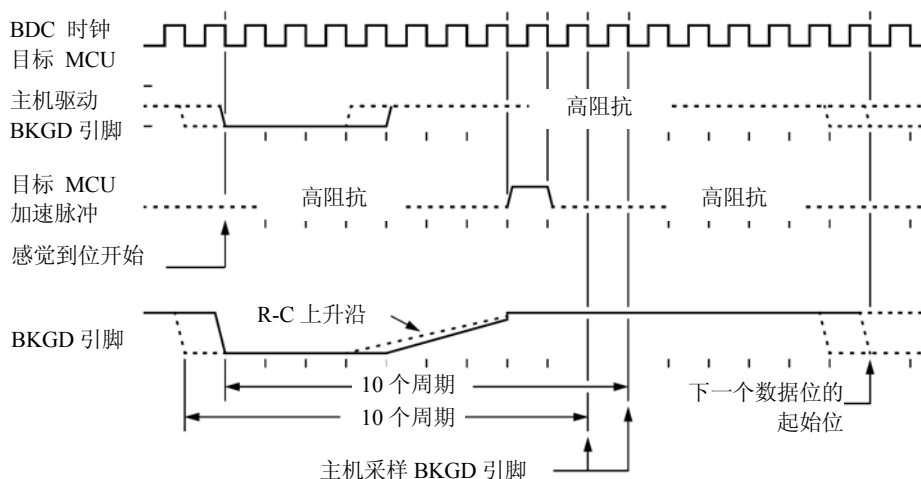


图12-3 BDC目标机到主机串行位时序(逻辑1)

图 12-4 表示主机接收来自目标 MCU 的逻辑 0。对于目标机来说主机是异步的，所以从主机产生下降沿到目标机感知到位开始有 0 ~ 1 个周期的延迟。主机开始位时序，但由目标机 HCS08 完成它。因为目标机需要主机接收逻辑 0，它驱动 BKGD 引脚低电平保持 13 个 BDC 时钟周期，然后在很短的时间内迅速驱动它为高电平来加速上升沿。主机在位时间开始后的第 10 个周期对位电平进行采样。

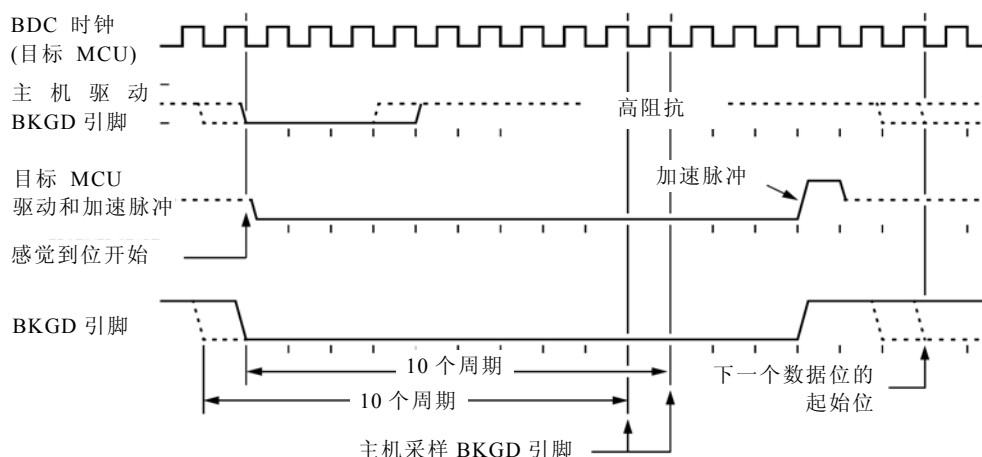


图12-4 BDC目标机到主机串行位时序(逻辑0)

12.2.3 BDC命令

BDC 命令由主计算机串行发向目标 HCS08MCU 的 BKGD 引脚。所有的命令和数据都先发送最高位并使用传统的 BDC 通信协议。背景模式命令需要目标 MCU 当前处在背景模式，而非介入式命令可在任何时候发出，无论目标 MCU 是处在背景模式还是在运行用户应用程序。

表 12-1 列出了所有 HCS08 的 BDC 命令，有简单的代码结构描述和每个命令的解释。

代码别名术语

这些术语在表 12-1 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 16 进制指令代码

- | | | |
|------|---|--|
| / | — | 命令分隔符 |
| d | — | 延时 16 个目标机 BDC 时钟周期(BDCSCR 中的 CLKSW 位控制 BDC 时钟源) |
| 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 位读取的数据，目标机向主机发送
WD16	—	16 位写入的数据，主机向目标机发送
SS	—	BDCSCR 的内容，目标机向主机传送时(状态)
CC	—	对 BDCSCR 写入的 8 位数据，主机向目标机发送(控制)
RBKP	—	读取的 16 位数据，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 12-1 BDC 命令汇总

命令命名	背景模式/非介入式	指令组成	描述
SYNC	非介入式	n/a ⁽¹⁾	请求同步参考脉冲以取得目标BDC通信速率
ACK_ENABLE	非介入式	D5/d	允许握手，执行指令后发出一个ACK脉冲
ACK_DISABLE	非介入式	D6/d	不允许握手，该指令不允许发出ACK脉冲
BACKGROUND	非介入式	90/d	如果允许进入背景模式(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	从刚才读过的地址处重读字节并报告状态
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	背景模式	08/d	执行开始于当前PC中地址处的用户应用程序
TRACE1	背景模式	10/d	跟踪PC中地址处的1条用户指令，然后回到背景模式
TAGGO	背景模式	18/d	同GO类似，但允许外部标记(HCS08设备没有外部标记引脚，因此在HCS08中TAGGO就像GO一样)
READ_A	背景模式	68/d/RD	读累加器(A)
READ_CCR	背景模式	69/d/RD	读条件码寄存器(CCR)
READ_PC	背景模式	6B/d/RD16	读程序计数器(PC)
READ_HX	背景模式	6C/d/RD16	读H和X寄存器对(H:X)
READ_SP	背景模式	6F/d/RD16	读堆栈指针(SP)
READ_NEXT	背景模式	70/d/RD	H:X增加1位，然后读H:X所指的内存字节
READ_NEXT_WS	背景模式	710/d/SS/RD	HX增加1位，然后读HX所指的内存字节。报告状态和数据
WRITE_A	背景模式	48/WD/d	写累加器(A)
WRITE_CCR	背景模式	49/WD/d	写条件码寄存器(CCR)
WRITE_PC	背景模式	4B/WD16/d	写程序计数器(PC)
WRITE_HX	背景模式	4C/WD16/d	写H和X寄存器对(H:X)
WRITE_SP	背景模式	4F/WD16/d	写堆栈指针(SP)
WRITE_NEXT	背景模式	50/WD/d	H:X增加1位，然后写H:X所指的内存字节
WRITE_NEXT_WS	背景模式	51/WD/d/SS	H:X增加1位，然后写H:X所指的内存字节。报告状态

1. SYNC指令是一种特殊的操作，它没有指令代码。

SYNC 命令不像其他 BDC 命令，因为在使用该命令之前主机需要知道通信频率，而频率是在分析 SYNC 命令应答之后得到的。

SYNC 命令讨论，主机：

- 驱动 BKGD 引脚低电平持续至少 BDC 最慢时钟下的 128 个周期(最慢的时钟通常时参考振荡器/64 或自时钟/64)

- 驱动 BKGD 引脚高电平形成一个加速脉冲来获得一个快速的上升时间(该加速脉冲一般是主机时钟的一个周期, 这个时钟周期和最快的目标 BDC 时钟一样快)
- 卸载 BKGD 引脚上的所有电平驱动使它回复到高阻状态
- 监听 BKGD 引脚上的同步响应脉冲

目标机检测来自主机的 SYNC 请求(它比曾出现在正常 BDC 通信中低速时间要长得多):

- 等待 BKGD 返回的逻辑高电平
- 延迟 16 个周期来允许主机停止驱动高电平的加速脉冲
- 驱动 BKGD 低电平 128 个 BDC 时钟周期
- 向 BKGD 驱动 1 个周期的高电平加速脉冲迫使 BKGD 上电平快速上升
- 卸载所有在 BKGD 引脚上的驱动使它回复高阻状态

主机检测这 128 个周期的同步响应脉冲的低电平时间, 并为以后的 BDC 通信确定正确的速率。一般来说, 主机能根据一部分目标机的实际速度来确定正确的通信速度, 并且通信协议可以允许小部分差异。

12.2.4 BDC硬件断点

BDC 包括一个相对简单的硬件断点, 它使得 CPU 地址总线地址与 BDCBKPT 寄存器中的 16 位值对应起来。该断点可以是强制断点也可以是标识断点。一个强制的断点使 CPU 在访问断点地址后的第一个指令边界进入活动背景模式。标识断点使指令操作码在断点地址处贴上标签, 以使 CPU 能进入背景模式而不是到指令队列末尾才执行。这意味着断点标签只可以放置在指令的操作码上而强制断点可以被设置在任意地址。

在BDC状态控制寄存器(BDCSCR)中, 断点使能(BKPTEN)控制位是用来允许断点逻辑的(BKPTEN=1)。当BKPTEN=0(复位后的默认值)时, 该断点逻辑将被禁止, 即没有BDC断点可以要求其他BDC断点寄存器和控制位的值改变。通常BDCSCR的强制/标志的选择(FTS)是通过选择强制(FTS=1)和标志(FTS=0)两种类型的断点来实现。

8 位的 BDCSCR 和 16 位的 BDCBKPT 地址匹配寄存器直接设置于 BDC 内而与一般的 MCU 地址映射没有瓜葛。也就是说用户应用程序无法访问这些寄存器。只有专用的 BDC 系列的命令才可以访问这些寄存器。READ_STATES 和 WRITE_CONTROL 被用于读写 BDCSCR。READ_BKPT 和 WRITE_BKPT 用于读写 16 位 BDCBKPT 地址匹配寄存器。一个主机调试接口可以随时读写这些寄存器, 即使是在用户程序执行过程中。然而, 更常见的是当 MCU 处于活动背景模式的情况下设置断点。

BDC提供了控制和状态信号的访问方法, 它允许在BDC逻辑外但仍在MCU芯片上设置更复杂的断点。有些HCS08的衍生部件可能还有额外的, 更为复杂的硬件断点。这些额外的断点需要通过读写MCU映像存储器来访问与其相关联的寄存器以及控制位。

12.3 寄存器定义

该节包括 BDC 寄存器和控制位的描述。

这里通过这些寄存器和控制位的名称来引用它们。通常, Freescale 提供一个头文件把它们的名称翻译为绝对地址。

12.3.1 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 位寄存器包括背景调试控制器的控制和

状态位。

- BDC 断点匹配寄存器（BDCBKPT）保持 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，不位于目标 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 处于背景调试模式时，ENBDM 控制位可能不允许写。（这避免了当 MCU 准备处于背景调试模式时，控制位禁止背景调试模式的模糊条件。）同时四个状态位（BDMACT、WS、WSF、DVF）是只读状态标识位，不能通过 WRITE_CONTROL 串行命令写。时钟选择（CLKSW）控制位可以在任意时间读或写。

12.3.1.1 BDC寄存器和控制位(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。

	第7位	6	5	4	3	2	1	第0位
读		BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
写	ENBDM							
正常复位	0	0	0	0	0	0	0	0
激活BDM复位	1	1	0	0	1	0	0	0

图12-5 BDC寄存器和控制位(BDCSCR)

表 12-2 BDCSCR 寄存器域描述

域	描述
7 ENBDM	使能BDM(允许背景模式)——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。 0 BDM不能有效(允许非介入性命令)。 1 BDM可以有效允许背景调试模式命令。
6 BDMACT	背景模式状态——该位为只读位。 0 BDM无效（用户程序正在运行）。 1 BDM有效并且等待串行命令。
5 BKPTEN	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。 0 禁止BDC断点。 1 允许BDC断点。
4 FTS	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进入背景模式而不是执行已标记的操作码。 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背景模式。 1 在下一个指令边界，断点匹配强制执行背景调试模式(地址无需是一个操作码)。
3 CLKSW	BDC通信时钟源选择位——当MCU以正常的用户模式复位时，CLKSW被置为0以选择固定的后备频率源作为BDC时钟。在MC9S08GB60中，该后备频率源是BDC模块中允许于8MHz的内部晶振。当MCU在背景调试模式下复位时，CLKSW被置为1以选择总线时钟作为BDC时钟。当允许可能改变总线频率的用户程序时，应该避免使用CLKSW=1选项，因为这可能导致BDC通信的损失。 0 明确的后备频率源 1 CPU总线时钟。

2 WS	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，可以用BACKGROUND指令可被用于使目标CPU跳出等待/停止模式并进入背景模式，在此所有的BDC指令都能运行。无论何时主机迫使目标MCU进入背景模式，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。 0 目标CPU正执行用户程序代码或正处于背景模式(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/停止模式变为激活的背景调试模式。
1 WSF	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背景模式，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，同时重执行等待或停止指令。) 0 内存存取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，内存存取指令失败。
0 DVF	数据有效失败状态位——如果由于目标CPU在差不多同一时刻执行了一条低速的内存存取指令，而导致内存访问指令失败时则置该状态位。通常的恢复策略是发出READ_LAST指令，直到返回的状态信息表明原始存取过程已经彻底完成。因为目前的HCS08系列没有存储模块支持低速存取，该位总是被读作0。如果不确定某款特定的HCS08系列是否包含低速存储模块，可以参考数据手册。 0 内存进入没有和低速内存存取发生冲突。 1 CPU在带有低速内存存取时内存进入指令失败。

12.3.1.2 BDC断点匹配寄存器

这个 16 位的寄存器拥有的地址用于 BDC 中的硬件断点。BDCSCR 中的 BKPTEN 和 FTS 控制位用于允许和配置断点逻辑。专门的 BDC 指令(READ-BKPT 和 WRITE-BKPT)用于读写 BDCBKPT 寄存器。当目标 MCU 在运行用户应用程序之前处于背景模式时正常设置断点。然而，因为 READ-BKPT 和 WRITE-BKPT 是非介入性指令，所以即使用户程序正在执行，它们仍能被执行。要获得安装和使用 BDC 中硬件断点逻辑的更多信息，参考 12.2.4 BDC 硬件断点。

12.3.2 系统背景调试强制复位寄存器 (SBDFR)

该寄存器包括单个只写控制位。必须使用串行背景模式命令来写 SBDFR。试图写该寄存器的用户程序被忽略。读总是返回 0x00。

	第7位	6	5	4	3	2	1	第0位
读	0	0	0	0	0	0	0	0
写								BDFR ¹
复位	0	0	0	0	0	0	0	0

1.BDFR只能通过串行背景模式调试命令写，用户程序不能操作。

图12-6系统背景调试强制复位寄存器 (SBDFR)

表 12-3 SBDFR 寄存器域描述

域	描述
0 BDFR	背景调试强制复位——串行背景模式命令例如WRITE_BYTE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。该位不能被用户程序写。

附录A电气特性

A.1简介

这章包含电气和定时说明

A.2最大绝对额定值

最大绝对额定值只是压力值。最大值时的功能操作不能保证。压力值超过了表 A-1 所描述的限制可能影响设备的可靠性或导致设备的永久性损坏。关于功能操作条件参见该节的其他表格。

该设备中包含了保护电路，可以避免因高静态电压或电子磁场产生的损坏。然而，建议采取正常的预防避免任何大于最大额定电压进入该高阻抗电路。如果不将输入连到一个低电压电平（例如， V_{SS} 或 V_{DD} ）可以加强操作的可靠性。

表 A-1 最大绝对额定值

额定值	标志	值	单位
电源电压	V_{DD}	-0.3 to +5.8	V
V_{DD} 的最大电流	I_{DD}	120	mA
输入电压	V_{in}	- 0.3 to $V_{DD} + 0.3$	V
单个引脚限制(应用到所有引脚) 1. 2. 3的瞬时最大电流	I_D	± 25	mA
存储温度	T_{stg}	-55 to +150	°C

1.输入必须是限制描述值中的电流。为了决定限流电阻器的值，计算正（ V_{DD} ）和负（ V_{SS} ）的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.所以功能的非电源引脚在内部箝位至 V_{SS} 和 V_{DD} 。
 3.电源必须维持调节操作 V_{DD} 的瞬时范围和操作最大电流情况。如果正的注入电流（ $V_{in}>V_{DD}$ ）大于 I_{DD} ，注入电流可能溢出可能导致外部电压失去调节。确保外荷载 V_{DD} 将分路电流比最大注入电流还大。当MCU不耗电时，这是最大的危险。例如：如果没有系统时钟存在，或如果时钟率很小，这可能减小总的耗电。

A.3热特性

该节提供了关于操作温度范围，功率消耗和封装的热敏电阻的信息。和芯片逻辑的功耗相比，在 I/O 引脚上的功率消耗通常很小，并且它是由用户决定而不是有 MCU 设计来控制。为了在功率计算中考虑 P_{IO} ，确定实际的引脚电压和 V_{SS} 或 V_{DD} 的不同，加倍每个 I/O 引脚的电流。除了异常的高引脚电流（重载荷），引脚电压和 V_{SS} 或 V_{DD} 的不同很小。

表 A-2 热特性

额定值	标志	值	单位
操作温度范围（已封装）	T_A	TL 到 TH -40 to 125	°C
热敏电阻（单层板） 8引脚PDIP 8引脚NB SOIC	θ_{JA}	113 150	°C/W

热敏电阻（四层板） 8引脚PDIP 8引脚NB SOIC	θ_{JA}	72 87	$^{\circ}\text{C}/\text{W}$
------------------------------------	---------------	----------	-----------------------------

芯片连接的平均温度（T_J）（ $^{\circ}\text{C}$ ）从等式 A-1 中得到：

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

其中：

T_A = 周围环境温度， $^{\circ}\text{C}$

T_{JA} = 封装的热敏电阻，连接到周围环境， $^{\circ}\text{C}/\text{W}$

P_D = P_{int} + P_{I/O}

P_{int} = I_{DD} × V_{DD}，瓦特——芯片内部电源

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} << P_{int}，P_{I/O} 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 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.4 ESD保护和闭锁抗扰度

虽然在设备中静电放电的破坏比早期的 CMOS 电路小得多，也应该使用正规的预防处理避免暴露在静态放电中。通过品质测试确保这些设备能够抵挡暴露在合理的静态电平中，而不产生永久的损坏。

所有的ESD测试和AEC-Q100汽车级集成电路的压力品质测试一致。在设备品质时，ESD压力通过人体模型（HBM），机器模型（MM）和电荷设备模型（CDM）完成，

如果暴露到 ESD 脉冲后，设备不再满足设备规格，一个设备定义为失败。完善 DC 参数，和在高温后在室温按照每个应用设备规格完成品质测试，在除非在设备规格中说明

表 A-3 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-4 ESD 和闭锁保护特性

编号	C	额定值 ¹	标志	最小	最大	单位
1	C	人体模型（HBM）	V _{HBM}	±2000	-	V
2	C	机器模型（MM）	V _{MM}	±200	-	V
3	C	电荷设备模型（CDM）	V _{CDM}	±500	-	V
4	C	在TA=125 $^{\circ}\text{C}$ 时，闭锁电流	I _{LAT}	±100	-	mA

¹参数可以达到通过在一个小的样例上从典型设备在典型条件下设计特性，除非另有说明

A.5 DC特性

该节包含各种电源条件，I/O 引脚特性。

表 A-6 DC 特性（温度范围：-40 到 125℃）

参数	标志	最小	典型	最大	单位
电源电压（运行，等待，停止模式）	V_{DD}	2.7		5.5	V
连接到VDD的RAM最小保持电压	V_{RAM}	$V_{POR}^{1, 2}$			V
低电压检测阀——大范围 (消费和工业芯片MC9S08QDx) V_{DD} 上升 V_{DD} 下降	V_{LVDH}	4.2 4.3	4.3 4.4	4.4 4.5	V V
低电压检测阀——大范围 (汽车芯片 S9S08QDx) (V_{DD} 上升) -40℃到0℃ 0到125℃ (V_{DD} 下降) -40℃到0℃ 0到125℃		4.175 4.2 4.275 4.3	4.3 4.4 4.3 4.4	4.4 4.5 4.4 4.5	V
低电压检测阀——小范围 V_{DD} 上升 V_{DD} 下降		2.48 2.54	2.56 2.62	2.64 2.7	V V
低电压检测阀——小范围 V_{DD} 下降 V_{DD} 上升					
低压警告阀——大范围 (消费和工业芯片MC9S08QDx) V_{DD} 下降 V_{DD} 上升	V_{LVWH}	4.2 4.3	4.3 4.4	4.4 4.5	V
低电压检测阀——大范围 (汽车芯片 S9S08QDx) (V_{DD} 上升) -40℃到0℃ 0到125℃ (V_{DD} 下降) -40℃到0℃ 0到125℃	V_{LVWH}	4.175 4.2 4.275 4.3	4.3 4.4 4.3 4.4	4.4 4.5 4.4 4.5	V
低压检测阀——小范围 V_{DD} 下降 V_{DD} 上升	V_{LVWL}	2.48 2.54	2.56 2.62	2.64 2.7	V V
低压禁止复位/还原滞后 5V 3V	V_{hys}	- -	100 35	- -	mV
能带隙参考电压 (消费和工业芯片MC9S08QDx) $V_{DD}=3.0V$, TEMP=25℃	V_{BG}	1.19	1.20	1.21	V
能带隙参考电压 (S9S08QDx) $V_{DD}=3.0V$, TEMP=25℃ -40到125℃	V_{BG}	1.18	1.20	1.215	V
输入高电压；所有的数字输入 $2.7V \leq V_{DD} \leq 5.5V$	V_{IH}	$0.7 \times V_{DD}$			V
输入低电压；所有的数字输入 $2.7V \leq V_{DD} \leq 5.5V$	V_{IL}	-	-	$0.3 \times V_{DD}$	V

输入滞后；所有的数字输入	V_{hys}	$0.06 \times V_{\text{DD}}$			V
输入泄露电流；仅输入引脚 ²	$ V_{\text{in}} $	-	0.025	1	μA
高阻抗（掉电状态）泄露电流 ²	$ V_{\text{OZ}} $	-	0.025	1	μA
内部上拉电阻 ³	R_{PU}	17.5		52.5	k Ω
内部下拉电阻 ⁴	R_{PD}	17.5		52.5	k Ω
输出高电压——低驱动($\text{PTxDSn}=0$)	V_{OH}	$V_{\text{DD}}-1.5$	-	-	V
5V, $I_{\text{load}}=-2\text{mA}$		$V_{\text{DD}}-1.5$	-	-	
3V, $I_{\text{load}}=-0.6\text{mA}$		$V_{\text{DD}}-0.8$	-	-	
5V, $I_{\text{load}}=-0.4\text{mA}$		$V_{\text{DD}}-0.8$	-	-	
3V, $I_{\text{load}}=-0.24\text{mA}$					
输出高电压——高驱动($\text{PTxDSn}=1$)	V_{OH}	$V_{\text{DD}}-1.5$	-	-	V
5V, $I_{\text{load}}=-10\text{mA}$		$V_{\text{DD}}-1.5$	-	-	
3V, $I_{\text{load}}=-3\text{mA}$		$V_{\text{DD}}-0.8$	-	-	
5V, $I_{\text{load}}=-2\text{mA}$		$V_{\text{DD}}-0.8$	-	-	
3V, $I_{\text{load}}=-0.4\text{mA}$					
所有端口引脚的最大 I_{OH}	$ I_{\text{OHT}} $			100	mA
5V				60	
3V					
输出低电压——低驱动($\text{PTxDSn}=0$)	V_{OL}			1.5	V
5V, $I_{\text{load}}=-2\text{mA}$				1.5	
3V, $I_{\text{load}}=-0.6\text{mA}$				0.8	
5V, $I_{\text{load}}=-0.4\text{mA}$				0.8	
3V, $I_{\text{load}}=-0.24\text{mA}$					
输出低电压——高驱动($\text{PTxDSn}=1$)	V_{OL}			1.5	V
5V, $I_{\text{load}}=-10\text{mA}$				1.5	
3V, $I_{\text{load}}=-3\text{mA}$				0.8	
5V, $I_{\text{load}}=-2\text{mA}$				0.8	
3V, $I_{\text{load}}=-0.4\text{mA}$					
所有端口引脚的最大 I_{OL}	I_{OLT}	-		100	mA
5V		-		60	
3V					
直流吸纳电流 ^{2, 4, 5, 6} $V_{\text{IN}} < V_{\text{SS}}$, $V_{\text{IN}} > V_{\text{DD}}$	$ I_{\text{IC}} $			0.2 5	mA
输入电容；所有的非电源引脚	C_{in}	-	-	7	pF

¹RAM 将保持数据直至降到 POR 电压。RAM 中的数据在上电复位后不保证有效。

²这个参数是表现特征的并不是在每个设备上测试。

³上拉电阻的度量条件 $V_{\text{IN}}=V_{\text{SS}}$, 下拉条件 $V_{\text{IN}}=V_{\text{DD}}$

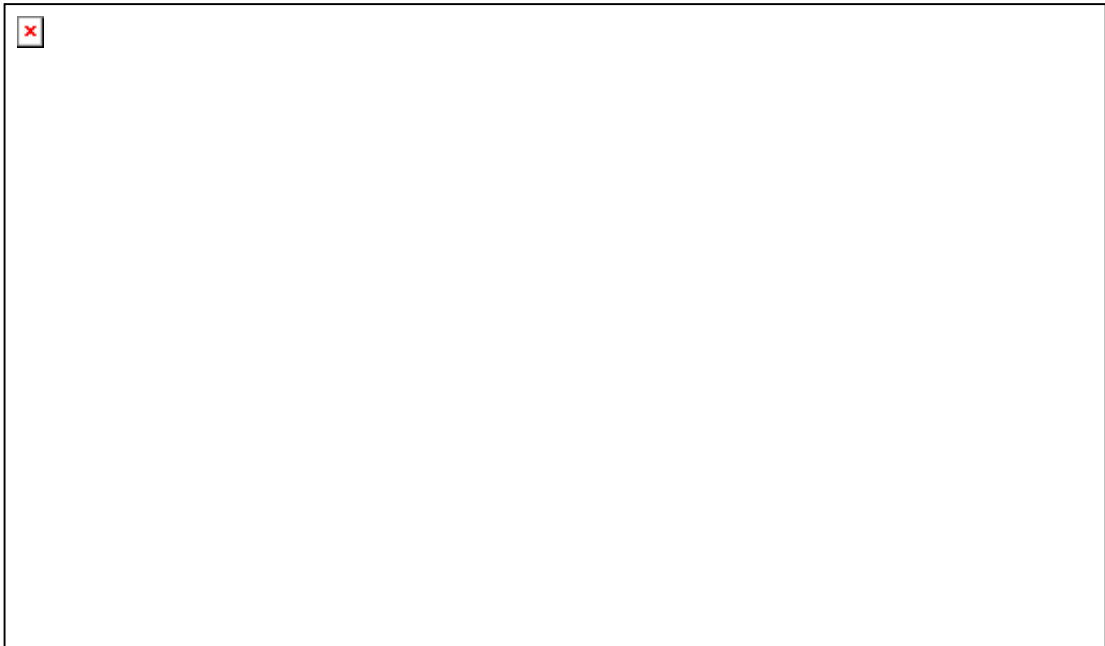
⁴所有的功能性非供电引脚通过一个内部钳位电阻连接到 V_{SS} 和 V_{DD}

⁵输入电流值必须受限与指定值。为决定必须的限流电阻值，必须计算正极和负极钳位电源电阻值，然后采用这两个值中的最大值。

⁶在瞬间和运行时的最大电流值时电源必须保持在 V_{DD} 范围内规律的供应。如果正极吸纳电流 ($V_{\text{IN}} > V_{\text{DD}}$) 大于 I_{DD} ，该吸纳电流将溢出 V_{DD} 允许范围和导致外部供电电压失去规律。为确保外部 V_{DD} 要求电流不大于最大吸纳电流。当 MCU 不耗电时，这是最大的危险。例如：如果没有系统时钟存在，或如果时钟率很小，这可能减小总的耗电。



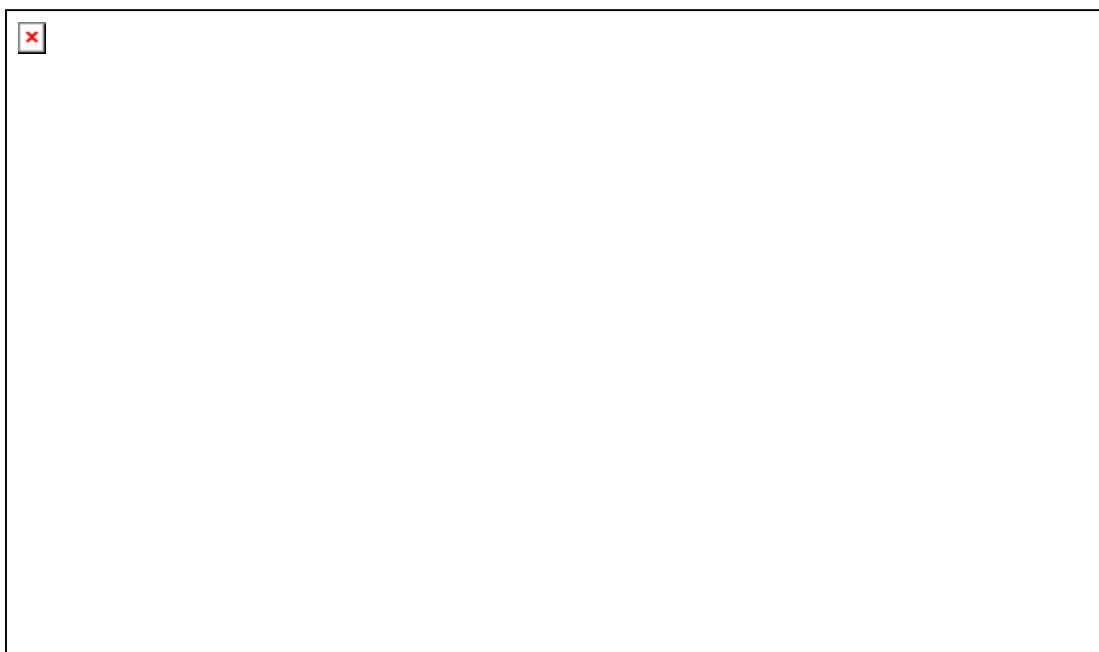
图A-1 $V_{DD}=5V$ 时，典型低边驱动特性，低驱动($PTxDSn=0$) V_{OL} vs I_{OL}



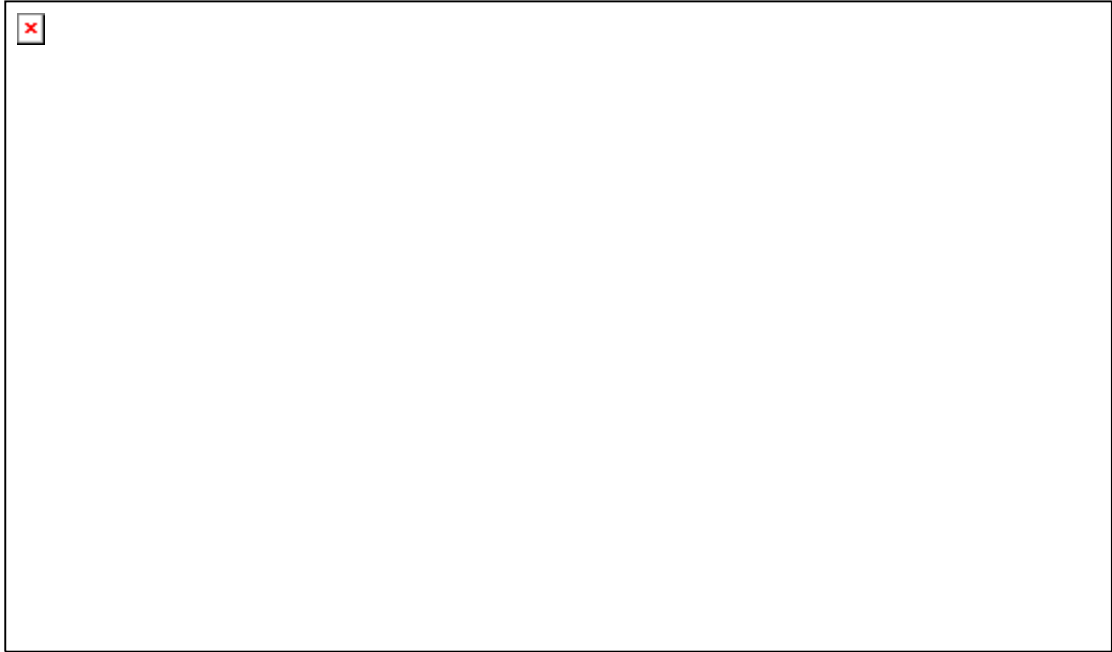
图A-2 $V_{DD}=3V$ 时，典型低边驱动特性，低驱动($PTxDSn=0$) V_{OL} vs I_{OL}



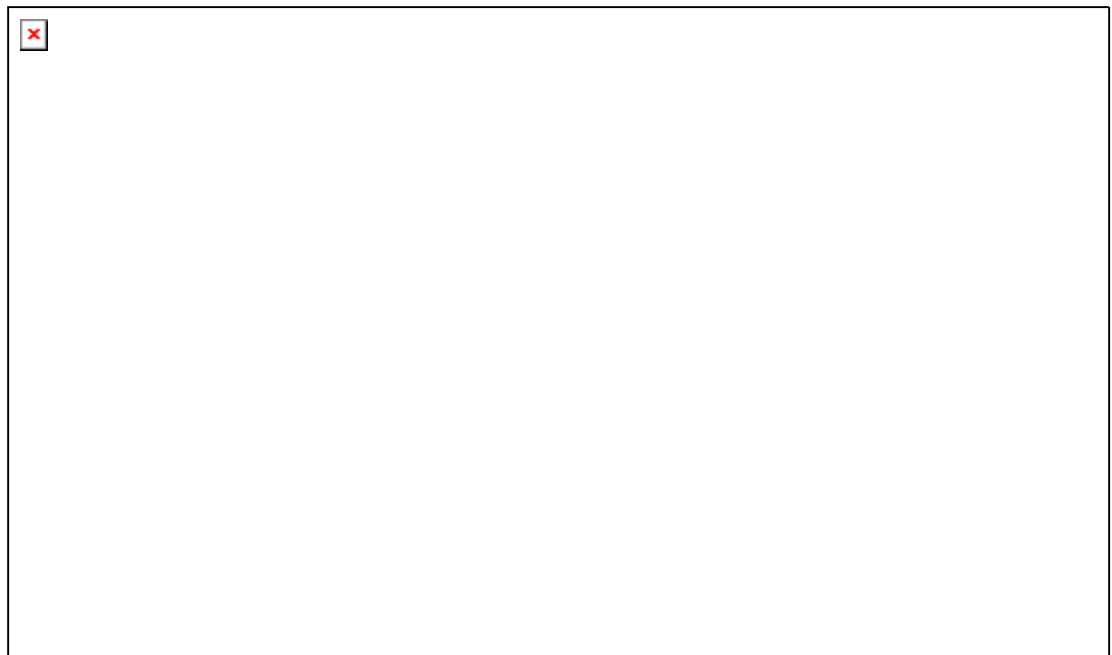
图A-3 $V_{DD}=5V$ 时，典型低边驱动特性，高驱动($PTxDSn=0$) V_{OL} vs I_{OL}



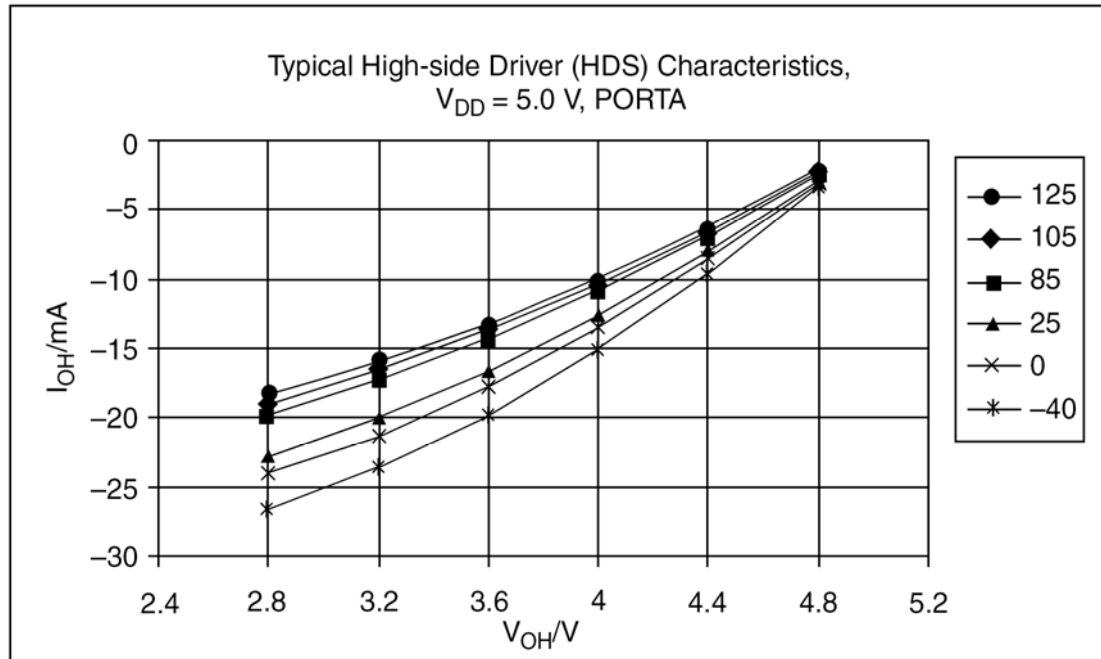
图A-4 $V_{DD}=3V$ 时，典型低边驱动特性，高驱动($PTxDSn=0$) V_{OL} vs I_{OL}



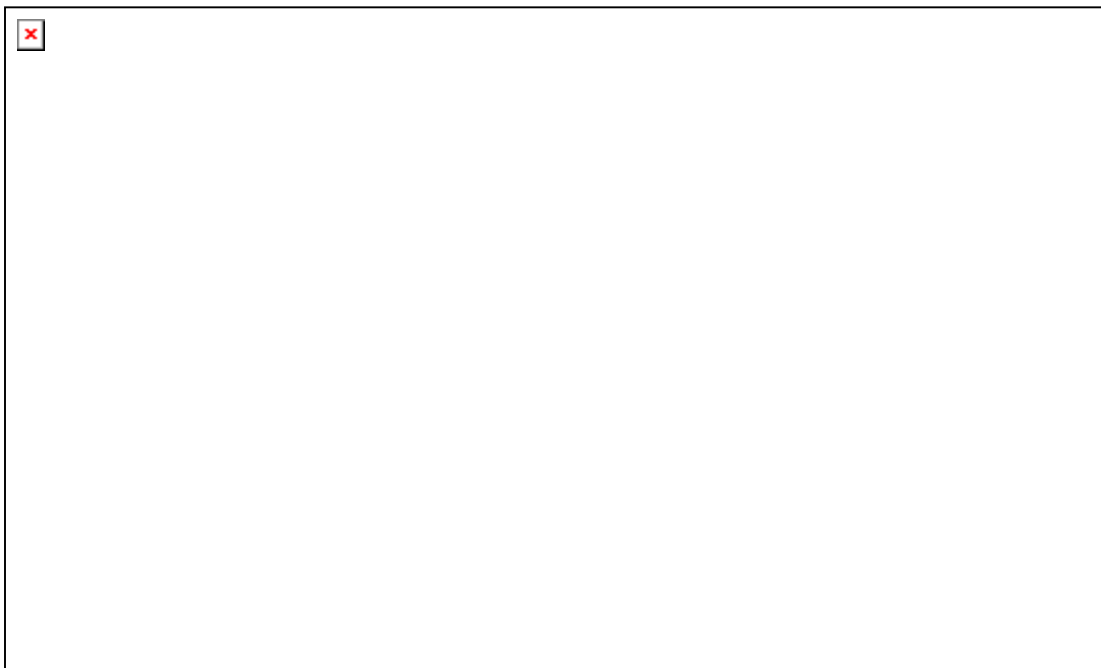
图A-5 $V_{DD}=5V$ 时，典型高边驱动特性，低驱动($PTxDSn=0$) V_{OL} vs I_{OL}



图A-6 $V_{DD}=3V$ 时，典型高边驱动特性，低驱动($PTxDSn=0$) V_{OL} vs I_{OL}



图A-7 $V_{DD}=5V$ 时，典型高边驱动特性，高驱动($PTxDSn=0$) V_{OL} vs I_{OL}



图A-8 $V_{DD}=3V$ 时，典型高边驱动特性，高驱动($PTxDSn=0$) V_{OL} vs I_{OL}

A.6 电源电流特性

这部分包括各种操作模式下的电源电流

参数	标志	V _{DD} (V)	典型 ¹	最大 ²	单位	温度
运行时电源电流 ³ (CPU 时钟=2MHz, f _{Bus} =1MHz)	RI _{DD}	5	0.95	1.5 ⁴	mA	
		3	0.90	1.5		
运行时电源电流 ⁵ (CPU时钟=16MHz, f _{Bus} =8MHz)	RI _{DD}	5	3.5	5 ⁶	mA	
		3	3.4	5		
等待模式时电源电流 ⁷ (CPU时钟=16MHz, f _{Bus} =8MHz)	WI _{DD}	5	1.55	2.2	mA	
		3	1.50	2.2		
Stop2模式电源电流 (消费和工业芯片 MC9S08QDx)	S2I _{DD}	5	0.80	7.5 ⁸ 20 ⁴	μA	-40 到85℃ -40 到125℃
		3	0.80	7.0 ⁸ 15	μA	
Stop2模式电源电流 (汽车芯片S9S08QDx)		5	0.80	7.5 ⁹ 25 ⁴	μA	-40 到85℃ -40 到125℃
		3	0.80	7.0 ⁸ 20	μA	
Stop3模式电源电流 (消费和工业芯片 MC9S08QDx)	S3I _{DD} S	5	0.90	8.0 ⁸ 25 ⁴	μA	-40 到85℃ -40 到125℃
		3	0.90	7.1 ⁸ 20	μA	-40 到85℃ -40 到125℃
Stop3模式电源电流 (汽车芯片S9S08QDx)		5	0.90	8.0 ⁸ 30 ⁴	μA	-40 到85℃ -40 到125℃
		3	0.90	7.1 ⁸ 25	μA	-40 到85℃ -40 到125℃
带有RTI的Stop2或stop3 ⁷		5	400		nA	25℃
		3	350		nA	
带有LVD的stop3 (LVDE=LVDS=1)		5	110		μA	
		3	90		μA	
带有振荡器的stop3 ⁸ (OSCSETN=1)		5	75		μA	
		3	65		μA	

¹典型的是在25℃测量

²这里给出的值时初步估计而不是完整描述。

³除了ADC,配置为FBE的ICG, 所有的模块在引脚上不包含任何直流电载入。

⁴这个参数在每个个体进行了测试, 所有其他在最大列的值则是通过描述保证。

⁵除了ADC,配置为FBE的ICG, 所有的模块在引脚上不包含任何直流电载入。

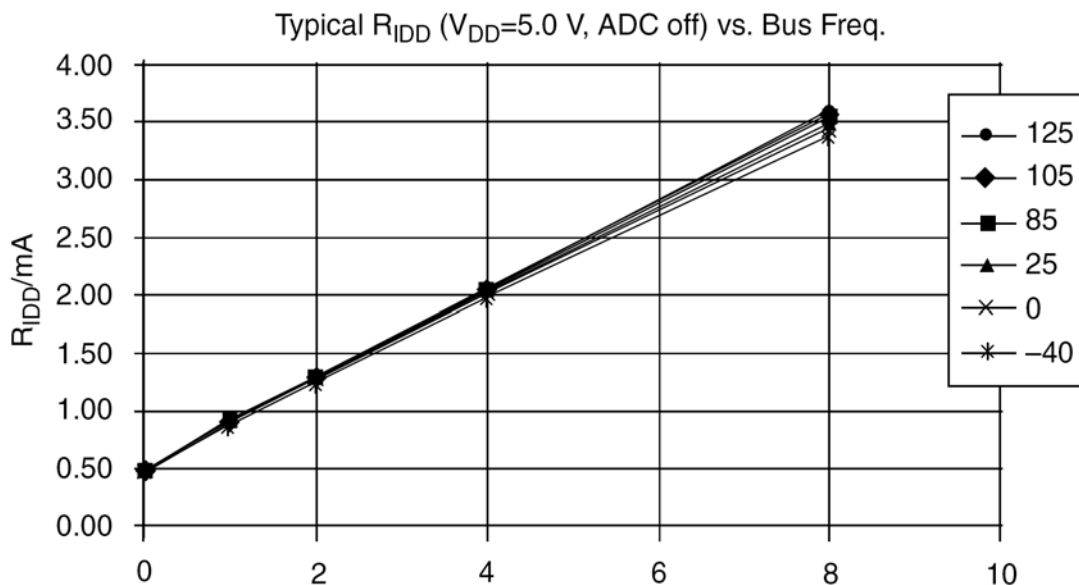
⁶这个参数在每个个体进行了测试, 所有其他在最大列的值通过描述保证。

⁷大多数消费者希望发现从stop2或stop3自动唤醒可以用来代替更高电路等待模式。

⁸这个参数是表现特征的并不是在每个设备上都测试。

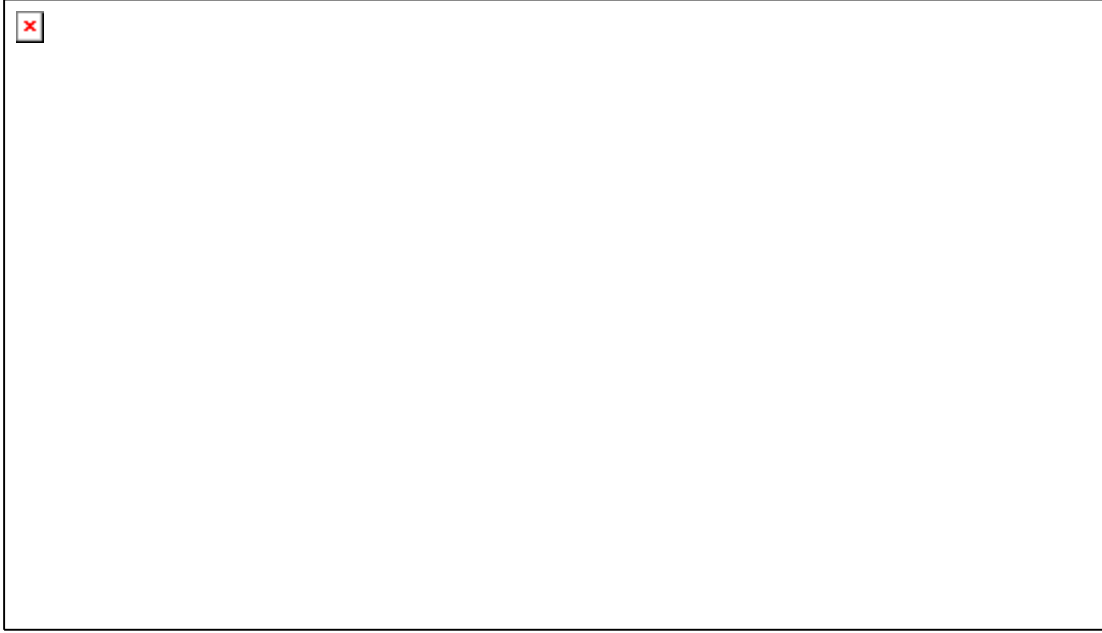
⁹这个参数是表现特征的并不是在每个设备上都测试。

¹⁰大多数消费者希望发现从stop2或stop3自动唤醒可以用来代替更高电路等待模式。等待模式通常560μA, 3V, f_{Bus}=1MHz。

图A-9运行模式下典型的 I_{DD} ，总线频率。(FEI) (ADC关)

A.7内部时钟源特性

表 A-7 内部时钟源说明					
特性	标志	最小值	类型 ¹	最大值	单位
内部平均参考频率—未平衡的	f_{int_ut}	25	31.25	41.66	KHz
内部平均参考频率—平衡的	f_{int_t}	-	31.25	-	KHz
DCO输出频率范围—未平衡的	f_{dco_ut}	12.8	16	21.33	MHz
DCO输出频率范围—平衡的	f_{dco_t}	-	16	-	MHz
固定电压下平衡DCO输出频率的解决和温度（消费和工业芯片MC9S08QDx） ²	$\Delta f_{dco_res_t}$	-	-	-0.2到0.2	% f_{dco}
固定电压下平衡DCO输出频率的解决和温度（汽车芯片S9S08QDx） ² -40到0°C 0到125°C		-	-	-0.3到0.3 -0.2到0.2	
溢出电压下的平衡的DCO输出频率的总偏差和温度 ² 消费和工业芯片MC9S08QDx，汽车芯片S9S08QDx	Δf_{dco_t}	-	-	-2到2 -3到3	% f_{dco}
FLL获取时间 ^{2,3}	$t_{acquire}$			1	ms
DCO输出时钟的长期抖动（平均超过2ms间隔） ⁴	C_{jitter}	-	-	0.6	% f_{dco}
¹ 上表中数据的典型特性测量在3.0V和3.0V.25°C，或典型的推荐值					
² 标志但不测试					
³ 该说明使用于任何时候被改变的FLL参考源或参考分频器，调整值从FLL禁止改变为FLL允许。如果一个晶振/共鸣器被用来作参考，该说明假定它已运行。					
⁴ 抖动是平均偏离了程控频率测量所指定间隔的最大值 f_{BUS} 。测量是由滤波器提供电源外部提供时钟信号的设备完成的。噪音通过 V_{DD} 和 V_{SS} 注入到FLL电路，晶振频率的变化增加了给定时间间隔内 C_{jitter} 的百分比。					



图A-10典型的DCO输出比温度偏离



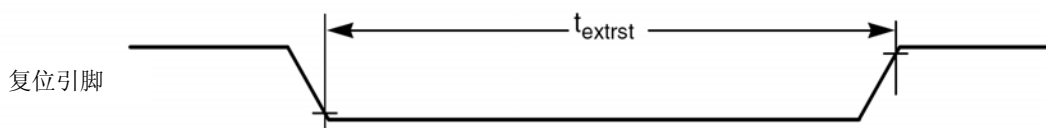
图A-11典型的DCO输出比操作电压偏离

A.8 AC特性

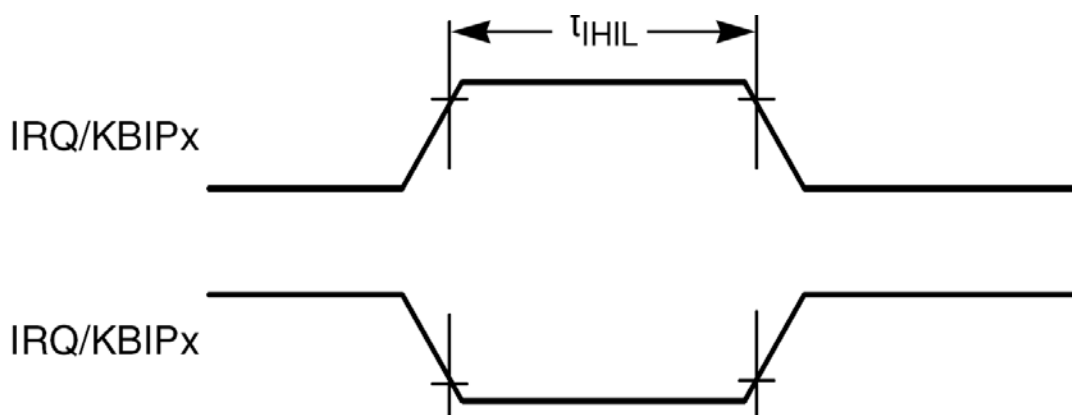
该节描述了每个外设系统的交流时序特性。

A.8.1 控制时许

表 A-8 控制时序					
参数	标志	最小	典型 ¹	最大	单位
总线频率($t_{cyc} = 1/f_{Bus}$)	f_{Bus}	1	-	8	MHz
实时中断内部振荡器周期	t_{RTI}	700	-	1300	μs
外部复位脉宽 ²	t_{extrst}	100	-	-	ns
IRQ脉宽 异步路径 ² 同步路径 ³	t_{ILIH}, t_{IHIL}	100 1.5 t_{cyc}	-	-	ns
KBIPx脉宽 异步路径 ² 同步路径 ³	t_{ILIH}, t_{IHIL}	100 1.5 t_{cyc}	-	-	ns
端口上升和下降时间 (load = 50 pF) ⁵ 禁止回转速控制 (PTxSE=0) 允许回转速控制 (PtxSE=1)	t_{Rise}, t_{Fall}	- -	3 30		ns
执行背景调试强制复位 进入用户或BDM模式 BKGD/MS设定定时时间	t_{MSSU}	500	-	-	ns
执行背景调试强制复位 进入用户或BDM模式 BKGD/MS保持定时时间	t_{MSH}	100	-	-	μs
¹ 上表中数据的典型特性测量在3.0V,25℃ ² 这是确保可承认的最小脉宽 ³ 这是确认通过引脚同步电路的最小值脉宽。更小的脉宽可能不被识别。在停止模式下，同步器是旁路的因此更小一点的脉宽可被识别。 ⁴ 定时以 V_{DD} 20 %到 80 %电平显示。工作温度范围-40 ℃至 125 ℃之间。 ⁵ 在上电期间在POR，BKGD/MS后进入BDM模式必须保持低在 V_{DD} 上升超过 V_{LVD} 后保持 t_{MSH} 一段时间。					



图A-12复位定时



图A-15定时输入捕捉脉冲

A.9 ADC特性

表 A-10 ADC 特性

表 A-10ADC 特性							
特性	条件	标志	最小值	类型 ¹	最大值	单元	注释
电源电流 ADLPC=1 ADLSMP=1 ADCO=1	$V_{DDAD} \leq 3.6V$ (3.0V类型)	I_{DDAD}	-	110	-	μA	超过温度(类型 25℃)
	$V_{DDAD} \leq 5.5V$ (5.0V类型)		-	130	-		
电源电流 ADLPC=1 ADLSMP=0 ADCO=1	$V_{DDAD} \leq 3.6V$ (3.0V类型)	I_{DDAD}	-	200	-	μA	
	$V_{DDAD} \leq 5.5V$ (5.0V类型)		-	220	-		
电源电流 ADLPC=0 ADLSMP=1 ADCO=1	$V_{DDAD} \leq 3.6V$ (3.0V类型)	I_{DDAD}	-	320	-	μA	
	$V_{DDAD} \leq 5.5V$ (5.0V类型)		-	360	-		
电源电流 ADLPC=0 ADLSMP=0 ADCO=1	$V_{DDAD} \leq 3.6V$ (3.0V类型)	I_{DDAD}	-	580	-	μA	
	$V_{DDAD} \leq 5.5V$ (5.0V类型)		-	660	-		
电源电流	停止, 复位, 模块关	I_{DDAD}	-	<1	100	nA	
高参考电压		V_{REFH}	2.7	V_{DDAD}	V_{DDAD}	V	
低参考电压		V_{REFL}	V_{SSAD}	V_{SSAD}	V_{SSAD}	V	
ADC转换时钟	高速度 (ADLPC=0)	f_{ADCK}	0.4	-	8.0	MHz	$t_{ADCK} = 1/f_{ADCK}$
	低功耗 (ADLPC=1)		0.4	-	4.0		
ADC异步时钟源	高速度 (ADLPC=0)	f_{ADACK}	2.5	4	6.6	MHz	$t_{ADACK} = 1/f_{ADACK}$
	低功耗 (ADLPC=1)		1.25	2	3.3		
转换时间	短样本 (ADLSMP=0)	t_{ADC}	20	20	23	t_{ADCK} 周期	2加5 $t_{BUS} = 1/f_{BUS}$ 周期
	长样本 (ADLSMP=1)		40	40	43		

采样时间	短样本 (ADLSMP=0)	t _{ADS}	4	4	4	t _{ADCK} 周期	
	长样本 (ADLSMP=1)		24	24	24		
输入电压		V _{ADIN}	V _{REFL}	-	V _{REFH}	V	
输入电容		C _{ADIN}	-	7	10	PF	不测试
输入电阻		R _{ADIN}	-	5	15	k Ω	不测试
模拟源电阻		R _{AS}	-	-	10 ²	k Ω	外接到 MCU
理想方案 (1 _{LSB})	10位模式	RES	2.637	4.883	5.371	MV	V _{REFH} /2 ^N
	8位模式		10.547	19.53	21.48		
全部未调整的 错误	10位模式	E _{TUE}	0	±1.5	±3.5	LSB	包括量 化
	8位模式		0	±0.7	±1.0		
微分非线性	10位模式	DNL	0	±0.5	±1.0	LSB	
	8位模式		0	±0.3	±0.5		
	单调性和无失码保证						
积分非线性	10位模式	INL	0	±0.5	±1.0	LSB	
	8位模式		0	±0.3	±0.5		
零尺度误差	10位模式	E _{ZS}	0	±1.5	±3.1	LSB	V _{ADIN} = V _{SSA}
	8位模式		0	±0.5	±0.7		
全尺度误差	10位模式	E _{FS}	0	±1.0	±1.5	LSB	V _{ADIN} = V _{DDA}
	8位模式		0	±0.5	±0.5		
量化错误	10位模式	E _Q	-	-	±0.5	LSB	8位模式 未被截 短

¹典型值假定V_{DDAD}=5.0V, Temp=25℃, f_{ADCK}=1.0 Mhz除非在其它状态。典型值仅作参考并不测试

²在4兆赫, 为最大频率, 使用比例较低的源阻抗

A.10 FLASH规格说明

该节详细描述了写入和擦除时序及 FLASH 存储器写入擦除的寿命。

写入和擦除不需要任何其他特殊的电源, 只需普通的 V_{DD} 电源。更多关于擦除和写入操作的信息, 参见内存章节。

表 A-11 FLASH 特性

特性	标志	最小	典型 ¹	最大	单位
写入/擦除电压	$V_{prog/erase}$	2.7		5.5	V
读操作电压	V_{read}	2.7		5.5	V
内部 FCLK 频率 ²	f_{FCLK}	150		200	KHz
内部 FCLK 周期 (1/FCLK)	t_{Fcyc}	5		6.67	μs
字节写入时间 (随机地址) ⁽²⁾	t_{prog}	9			t_{Fcyc}
字节写入时间 (突发模式) ⁽²⁾	t_{Bust}	4			t_{Fcyc}
页擦除时间 ²	t_{Page}	4000			t_{Fcyc}
整体擦除时间 ⁽²⁾	t_{Mass}	20,000			t_{Fcyc}
写入/擦除强度 ³ TL 到 TH=-40°C到+125°C T=25°C		10,000 -	- 10,000	- -	周期
数据保持 ⁵	t_{D_ret}	15	100	-	年

¹ 时钟的频率通过软件设置来控制。

² 这些值由硬件状态机控制，用户代码不需要来计数周期。这个信息提供给计算写入和擦除的大约时间。

³ FLASH 典型强度评估了在 9S12Dx64 中的这个产品家族，对于其他的 Freescale 如何定义典型强度，可以参见工程公告 EB619，“非易失性存储器的典型强度”。

⁴ 典型的数据保持值基于在高温下测量所具有的固有的技术能力。使用 Arrhenius 等式。对于其他的 Freescale 如何定义典型数据保持，可以参见工程公告 EB618，“非易失性存储器的典型数据保持”。

附录B 订购须知和机械制图

B.1 订购须知

本节包含了 MC9S08AC16 系列设备的订购须知。

表 B-1 编号系统

设备编号 ¹	存储器		可用封装 ²	用途
	FLASH	RAM	类型	类型
MC9S08QD4	4KB	256B	8 引脚 PDIP	消费和工业
MC9S08QD2	2KB	128B	8 引脚 NB SOIC	
S9S08QD4	4KB	256B	8 引脚 NB SOIC	汽车
S9S08QD2	2KB	128B		

¹ 参见表 1-2 有每个设备所包含模块的详细描述。

B.1.1 设备编号机制

消费和工业产品编号机制

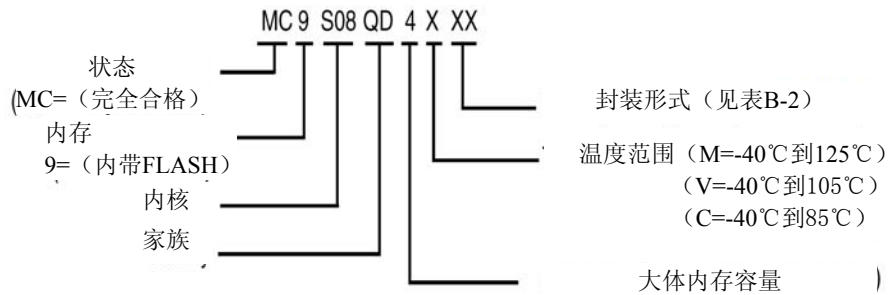
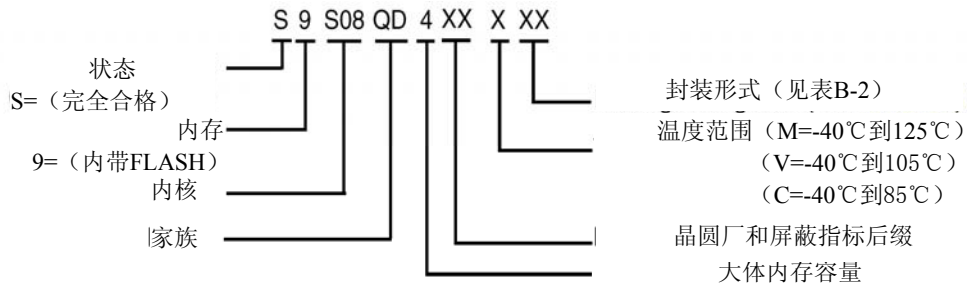


图12-7消费和工业产品的编号机制

汽车产品的编号机制



图B-1汽车产品的编号机制

B.2 机械制图

接下来的内容是封装选项的机械说明。参见表 B-2 中文档号与每个封装类型相对应。

表 B-2 封装信息

引脚数	类型	指定者	文件号
8	PDIP	PC	98ASB42420B
8	NB SOIC	SC	98ASB42564B



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

8 LD PDIP

文档号: 98ARH99048A

修订版本: N

案例号: 1314-05

2005-5-19

标准: 非 JEDEC



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

8 LD PDIP

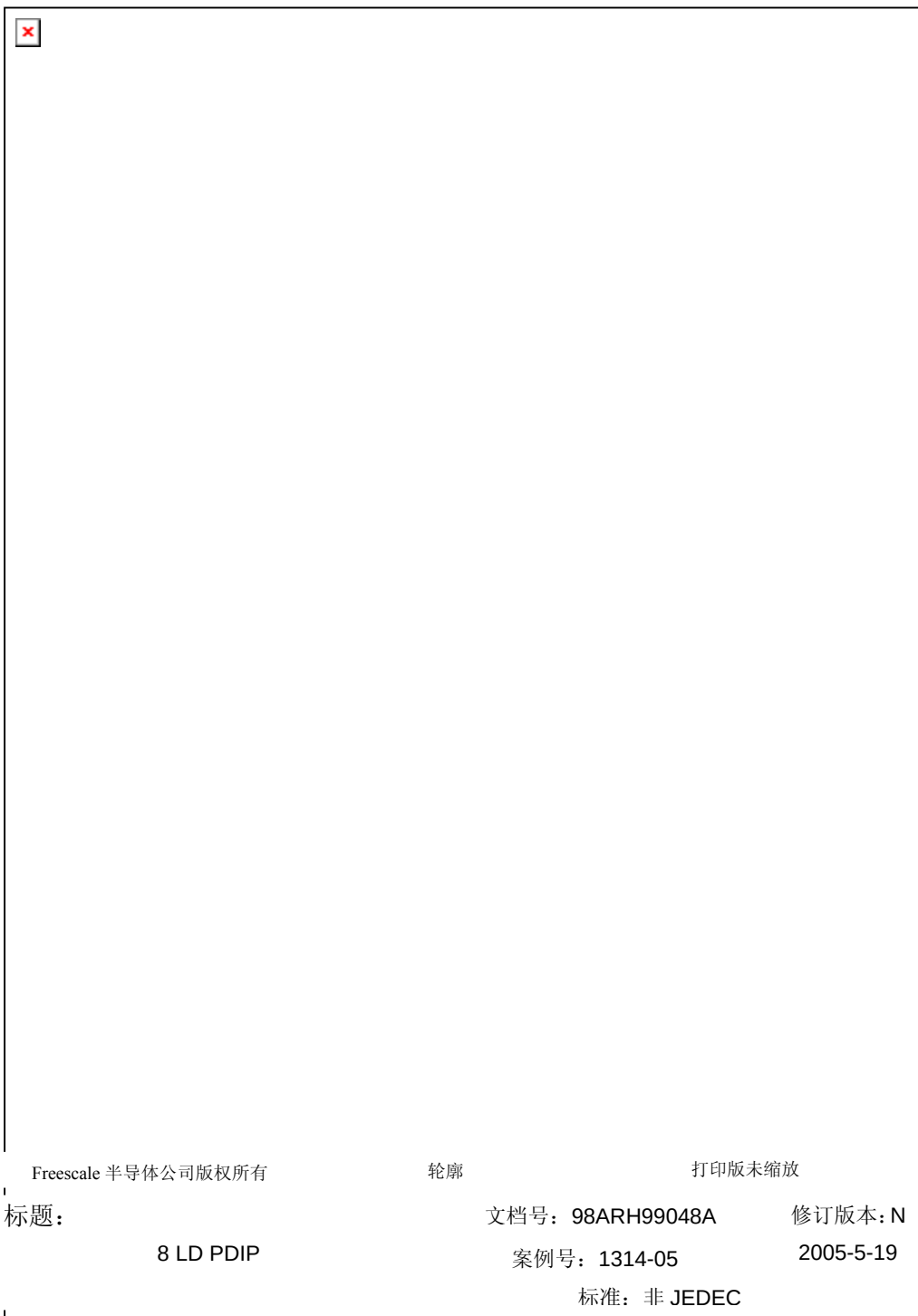
文档号: 98ARH99048A

修订版本:N

案例号: 1314-05

2005-5-19

标准: 非 JEDEC





Freescale 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ARH99048A

修订版本:U

8 LD PDIP

案例号: 1314-05

2005-4-07

标准: 非 JEDEC



Freescale 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ARH99048A

修订版本:U

8 LD PDIP

案例号: 1314-05

2005-4-07

标准: 非 JEDEC