

MC9S08AC16

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

MC9S08AC16 系列特点

MC9S08AC16 系列特点

- 消费&工业
 - MC9S08AC16
 - MC9S08AC8
- 汽车
 - MC9S08AW16A
 - MC9S08AW8A

8 位 HCS08 中央处理单元 (CPU)

- 40-MHz HCS08CPU(中央处理单元)
- 20-MHz 内部总线频率
- 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)
- 调试模式包括 2 个比较仪和 9 个触发器模式。8 个 FIFO 来存储流控制地址和事件数据。调试模式支持标签和强制断点
- 支持高达 32 个中断/复位源

存储器选项

- 高达 16KB 的片上在线可编程 FLASH 存储器, 有块保护和安全选项
- 高达 1KB 的片上 RAM

时钟源选项

- 时钟源选项包括晶体, 蜂鸣器, 外部时钟, 或能够 NVM 调整的精确的内部集成时钟

系统保护

- 可选的系统正常操作 (COP) 复位, 使用独立的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

省电模式

- 增加了两种停止模式

外部设备

- ADC-8 通道, 10 位 AD 转换器, 包含自动比较功能
- SCI-两个串行通信接口模块, 可选的 13 位停止符
- SPI-串行外设接口模块
- IIC-内部集成电路总线模块, 可以工作在 100kps 的最大总线负载上, 减少负载将会允许更高

的波特率

- 定时器-3 个 16 位定时器/脉宽调制 (TPM) 模块-2 个 2 通道和 1 个 4 通道；每个定时器在每个通道上都有可选的输入捕捉，输出比较，PWM。每个定时器在每个通道上都可以配置为带缓冲，中心 PWM (CPWM)
- KBI-7 引脚键盘中断模块

输入/输出

- 高达 38 个通用输入/输出 (I/O) 引脚
- 输入时，每个端口都有软件选择的上拉电阻
- 输出时，每个端口都有软件选择的转换速率控制
- 输出时，每个端口都有软件选择的驱动强度
- 主机复位引脚和上电复位 (POR)
- RESET, IRQ, BKGD/MS 引脚在内部上拉以减小系统消耗

封装选择

- 48 引脚 QFN
- 44 引脚 LQFP
- 32 引脚 LQFP

目 录

第一章 绪论.....	12
1.1 概述.....	12
1.2 MCU 结构框图.....	12
1.3 系统时钟分配.....	14
第二章 引脚及其连接.....	15
2.1 简介.....	15
2.2 设备的管脚分配.....	15
2.3 推荐的系统连接.....	17
2.3.1 电源 (V_{DD} 、 $2 \times V_{SS}$ 、 V_{DDAD} 和 V_{SSAD})	18
2.3.2 振荡器 (XTAL 与 EXTAL)	19
2.3.3 复位.....	19
2.3.4 背景/模式选择(BKGD/MS).....	19
2.3.5 ADC 参考引脚 (V_{REFH} 与 V_{REFL})	20
2.3.6 外部的中断引脚 (IRQ)	20
2.3.7 通用 I/O 及外设端口.....	20
第三章 工作模式.....	21
3.1 简介.....	21
3.2 特征.....	21
3.3 运行模式.....	21
3.4 背景调试模式.....	21
3.5 等待模式.....	22
3.6 停止模式.....	22
3.6.1 Stop2 模式.....	23
3.6.2 Stop3 模式.....	23
3.6.3 停止模式下激活 BDM 使能.....	24
3.6.4 停止模式下 LVD 使能.....	24
3.6.5 停止模式下的片上外设模块.....	24
第四章 片上存储器.....	26
4.1 简介 MC9SAC16 系列存储映像.....	26
4.1.1 复位和中断向量分配.....	26
4.2 寄存器地址和位分配.....	27
4.3 RAM.....	32
4.4 FLASH.....	32
4.4.1 特征.....	32
4.4.2 写入和擦除时间.....	33

4.4.3 写入和擦除命令的执行	33
4.4.4 突发模式写入的执行	35
4.4.5 访问错误	36
4.4.6 FLASH 块保护	36
4.4.7 向量重定向	37
4.5 安全性	37
4.6 FLASH 寄存器和控制位	38
4.6.1 FLASH 时钟分频寄存器(FCDIV)	38
4.6.2 FLASH 选项寄存器(FOPT 和 NVOPT)	39
4.6.3 FLASH 配置寄存器(FCNFG)	40
4.6.4 FLASH 保护寄存器(FPROT 和 NVPROT)	40
4.6.5 FLASH 状态寄存器(FSTAT)	40
4.6.6 FLASH 命令寄存器(FCMD)	41
第五章 复位、中断和系统配置	43
5.1 简介	43
5.2 特征	43
5.3 MCU 复位	43
5.4 计算机正常操作监控模块(COP)看门狗	44
5.5 中断	44
5.5.1 中断堆栈结构	45
5.5.2 外部中断请求(IRQ)引脚	46
5.5.2.1 引脚配置选项	46
5.5.2.2 边沿/电平触发	46
5.5.3 中断向量、中断源和局部屏蔽	46
5.6 低电压检测系统(LVD)	47
5.6.1 上电复位操作	48
5.6.2 LVD 复位操作	48
5.6.3 LVD 中断操作	48
5.6.4 低电压警告(LVW)	48
5.7 实时中断(RTI)	48
5.8 MCLK 输出	48
5.9 复位、中断以及系统控制寄存器和控制位	49
5.9.1 中断请求状态和控制寄存器(IRQSC)	49
5.9.2 系统复位状态寄存器(SRS)	50
5.9.3 系统背景调试强制复位寄存器(SBDFR)	51
5.9.4 系统选项寄存器(SOPT)	51
5.9.5 系统 MCLK 控制寄存器(SMCLK)	51
5.9.6 系统设备识别寄存器(SDIDH、SDIDL)	52
5.9.7 系统实时中断状态和控制寄存器(SRTISC)	52
5.9.8 系统电源管理状态控制寄存器 1(SPMSC1)	53
5.9.9 系统电源管理状态和控制寄存器 2(SPMSC2)	54
5.9.10 系统系统选项寄存器 2(STOP2)	55
第六章 并行输入和输出	56

6.1 简介	56
6.2 特征	57
6.3 引脚描述	58
6.3.1 A 口	58
6.3.2 B 口	58
6.3.3 C 口	58
6.3.4 D 口	59
6.3.5 E 口	59
6.3.6 F 口	60
6.3.7 G 口	60
6.4 并行 I/O 控制	60
6.5 引脚控制	61
6.5.1 内部上拉电阻使能	61
6.5.2 输出转换速率控制使能	61
6.5.3 输出驱动强度选择	62
6.6 停止模式下的引脚	62
6.7 并行 I/O 与引脚控制寄存器	62
6.7.1 A 口 I/O 寄存器 (PTAD 和 PTADD)	62
6.7.2 A 口引脚控制寄存器 (PTAPE、PTASE、PTADS)	63
6.7.3 B 口 I/O 寄存器 (PTBD 和 PTBDD)	64
6.7.4 B 口引脚控制寄存器 (PTBPE、PTBSE、PTBDS)	64
6.7.5 C 口 I/O 寄存器 (PTCD 和 PTCDD)	65
6.7.6 C 口引脚控制寄存器 (PTCPE、PTCSE、PTCDS)	66
6.7.7 D 口 I/O 寄存器 (PTDD 和 PTDDD)	67
6.7.8 D 口引脚控制寄存器 (PTDPE、PTDSE、PTDDS)	68
6.7.9 E 口 I/O 寄存器 (PTED 和 PTEDD)	69
6.7.10 E 口引脚控制寄存器 (PTEPE、PTESE、PTEDS)	69
6.7.11 F 口 I/O 寄存器 (PTFD 和 PTFDD)	70
6.7.12 F 口引脚控制寄存器 (PTFPE、PTFSE、PTFDS)	71
6.7.13 G 口 I/O 寄存器 (PTGD 和 PTGDD)	72
6.7.14 G 口引脚控制寄存器 (PTGPE、PTGSE、PTGDS)	72
第七章 中央处理单元 (S08CPUV2)	74
7.1 介绍	74
7.1.1 特征	74
7.2 编程结构和 CPU 寄存器	74
7.2.1 累加器(A)	75
7.2.2 索引寄存器 (H:X)	75
7.2.3 堆栈指针 (SP)	75
7.2.4 程序计数器 (PC)	75
7.2.5 条件码寄存器(CCR)	76
7.3 寻址模式	77
7.3.1 内在寻址方式 (INH)	77
7.3.2 相对寻址方式 (REL)	77
7.3.3 立即寻址方式(IMM)	77

7.3.4 直接寻址方式(DIR)	77
7.3.5 扩展寻址方式(EXT).....	77
7.3.6 变址寻址方式.....	77
7.3.6.1 无偏移量变址方式(IX).....	77
7.3.6.2 无偏移量变址、变址加 1 寻址方式(IX+).....	78
7.3.6.3 8 位偏移量变址方式(IX1)	78
7.3.6.4 8 位偏移量变址、变址加 1 寻址方式(IX1+).....	78
7.3.6.5 16 位偏移量变址方式(IX2).....	78
7.3.6.6 8 位偏移量堆栈寻址方式(SP1).....	78
7.3.6.7 16 位偏移量堆栈寻址方式(SP2).....	78
7.4 特殊操作.....	78
7.4.1 复位序列.....	78
7.4.2 中断序列.....	78
7.4.3 等待模式.....	79
7.4.4 停止模式.....	79
7.4.5 背景模式.....	79
7.5 HCS08 指令设置摘要	79
第八章 内部时钟发生器(S08ICGV4).....	91
8.1 介绍.....	92
8.1.1 特征.....	93
8.1.2 操作方式.....	93
8.1.3 框图.....	94
8.2 外部信号描述.....	95
8.2.1 EXTAL—外部参考时钟/振荡器输入	95
8.2.2 XTAL—振荡器输出.....	95
8.2.3 外部时钟连接.....	95
8.2.4 外部晶振/共鸣器连接.....	95
8.3 寄存器定义.....	96
8.3.1 ICG 控制寄存器 1 (ICGC1)	96
8.3.2 ICG 控制寄存器 2 (ICGC2)	97
8.3.3 ICG 状态寄存器 1 (ICGS1)	98
8.3.4 ICG 状态寄存器 2 (ICGS2)	98
8.3.5 ICG 滤波器寄存器 (ICGFLTU、ICGFLTL)	99
8.3.6 ICG 调整寄存器 (ICGTRM)	99
8.4 功能性描述.....	100
8.4.1 关模式.....	100
8.4.1.1 活跃 BDM.....	100
8.4.1.2 OSCSTEN 位设置	100
8.4.1.3 停止/关模式恢复.....	100
8.4.2 自时钟模式 (SCM)	100
8.4.3 FLL 内部时钟模式 (FEL)	102
8.4.4 FLL 内部失锁.....	102
8.4.5 FLL 内部锁定.....	102
8.4.6 FLL 旁路外部时钟模式 (FBE)	102

8.4.7 FLL 外部时钟模式 (FEE)	102
8.4.7.1 FLL 外部失锁	103
8.4.7.2 FLL 外部锁定	103
8.4.8 FLL 锁和锁丢失检测	103
8.4.9 FLL 时钟丢失检测	103
8.4.10 时钟模式必要条件	104
8.4.11 固定频率时钟	105
8.4.12 高增益振荡器	105
8.5 初始化/应用信息	105
8.5.1 简介	105
8.5.2 例 1: 外部晶振=32kHz, 总线频率=4.19MHz	106
8.5.3 例 2: 外部晶振=4MHz, 总线频率=20MHz	108
8.5.4 例 3: 无外部晶振, 总线频率=5.4MHz	109
8.5.5 例 4: 内部时钟产生器调整	110
第九章 键盘中断 (S08KBIV1)	112
9.1 简介	112
9.2 键盘引脚复用	112
9.3 特征	112
9.3.1 KBI 框图	113
9.4 寄存器定义	114
9.4.1 KBI 状态控制寄存器 (KBISC)	114
9.4.2 KBI 引脚使能寄存器 (KBIPE)	115
9.5 功能描述	115
9.5.1 引脚使能	115
9.5.2 边沿和电平触发	115
9.5.3 KBI 中断控制	115
第十章 定时器/脉宽调制 (S08TPMV3)	116
10.1 简介	116
10.2 特征	116
10.3 TPMV3 与以前版本的区别	117
10.3.1 TPMV1 的变换	119
10.3.2 特征	119
10.3.3 操作模式	120
10.3.4 框图	120
10.4 信号描述	122
10.4.1 详细信号描述	122
10.4.1.1 EXTCLK——外部时钟源	122
10.4.1.2 TPMxCHn——TPM 通道 n 输入输出引脚	122
10.5 寄存器定义	124
10.5.1 TPM 状态和控制寄存器 (TPMxSC)	124
10.5.2 TPM 计数器寄存器 (TPMxCNTH:TPMxCNHTL)	125
10.5.3 TPM 计数器模寄存器 (TPMxMODH:TPMxMODL)	126
10.5.4 TPM 通道 n 状态和控制寄存器 (TPMxCnSC)	127

10.5.5 TPM 通道值寄存器 (TPMxCnVH : TPMxCnVL)	128
10.6 功能描述	129
10.6.1 计数器	129
10.6.1.1 计数器时钟源	129
10.6.1.2 计数器溢出和模块复位	129
10.6.1.3 计数模式	130
10.6.1.4 手动计数器复位	130
10.6.2 通道模式选择	130
10.6.2.1 输入捕捉模式	130
10.6.2.2 输出比较模式	130
10.6.2.3 边对齐 PWM 模式	131
10.6.2.4 中心对齐 PWM 模式	131
10.7 复位概述	133
10.7.1 概要	133
10.7.2 复位操作描述	133
10.8 中断	133
10.8.1 概述	133
10.8.2 中断操作描述	133
10.8.2.1 定时器溢出中断 (TOF) 描述	133
10.8.2.1.1 通常情形	133
10.8.2.1.2 中心对齐 PWM 情形	134
10.8.2.2 通道事件中断描述	134
10.8.2.2.1 输入捕捉事件	134
10.8.2.2.2 输出比较事件	134
10.8.2.2.3 PWM 结束占空比事件	134
第十一章 串行通信接口 (S08SCIV4)	135
11.1 简介	135
11.1.1 特征	136
11.1.2 操作模式	137
11.1.3 框图	137
11.2 寄存器定义	139
11.2.1 SCI 波特率寄存器 (SCIxBDH、SCIxBDL)	139
11.2.2 SCI 控制寄存器 1 (SCIxC1)	140
11.2.3 SCI 控制寄存器 2 (SCIxC2)	141
11.2.4 SCI 状态寄存器 1 (SCIxS1)	142
11.2.5 SCI 状态寄存器 2 (SCIxS2)	143
11.2.6 SCI 控制寄存器 3 (SCIxC3)	144
11.2.7 SCI 数据寄存器 (SCIxD)	145
11.3 功能描述	145
11.3.1 波特率产生	145
11.3.2 发送功能描述	146
11.3.2.1 发送终止和等待空闲	146
11.3.3 接收功能描述	146
11.3.3.1 数据采样技术	147

11.3.3.2 接收唤醒操作	147
11.3.4 中断和状态标志	148
11.3.5 其他 SCI 功能	148
11.3.5.1 8 位和 9 位数据模式	148
11.3.5.2 停止模式操作	149
11.3.5.3 循环模式	149
11.3.5.4 单线操作	149
第十二章 串行外设接口 (S08SPIV3)	150
12.1 简介	150
12.1.1 特征	152
12.1.2 框图	152
12.1.2.1 SPI 系统框图	152
12.1.2.2 SPI 模块框图	153
12.1.3 SPI 波特率发生器	154
12.2 外部信号描述	154
12.2.1 SPSCK — SPI 串行时钟	154
12.2.2 MOSI — 主入从出引脚	154
12.2.3 MISO — 主入从出引脚	154
12.2.4 $\overline{SS}$ — 从机选择引脚	155
12.3 操作模式	155
12.3.1 停止模式中的 SPI	155
12.4 寄存器定义	155
12.4.1 SPI 控制寄存器 1 (SPI1C1)	155
12.4.2 SPI 控制寄存器 2 (SPI1C2)	156
12.4.3 SPI 比特率寄存器 (SPI1BR)	157
12.4.4 SPI 状态寄存器 (SPI1S)	158
12.4.5 SPI 数据寄存器 (SPI1D)	159
12.5 功能性描述	159
12.5.1 SPI 时钟格式	159
12.5.2 SPI 中断	161
12.5.3 模式故障检测	162
第十三章 IIC 总线 (S08IICV2)	163
13.1 简介	163
13.1.1 特征	165
13.1.2 操作模式	165
13.1.3 框图	165
13.2 外部信号描述	166
13.2.1 SCL—串行时钟线	166
13.2.2 SDA—串行数据线	166
13.3 寄存器定义	167
13.3.1 IIC 地址寄存器 (IIC1A)	167
13.3.2 IIC 频率分频寄存器 (IIC1F)	167

13.3.3 IIC 控制寄存器 (IIC1C1)	168
13.3.4 IIC 状态寄存器 (IIC1S)	169
13.3.5 IIC 数据输入输出寄存器 (IIC1D)	170
13.3.6 IIC 控制寄存器 2 (IIC1C2)	170
13.4 功能性描述	171
13.4.1 IIC 协议	171
13.4.1.1 开始信号	172
13.4.1.2 从机地址传送	172
13.4.1.3 数据发送	172
13.4.1.4 停止信号	173
13.4.1.5 重新开始信号	173
13.4.1.6 仲裁程序	173
13.4.1.7 时钟同步	173
13.4.1.8 握手	174
13.4.1.9 时钟扩展	174
13.4.2 10-位地址	174
13.4.2.1 主机发送从机接收	174
13.4.2.2 主机接收从机发送	175
13.4.3 一般寻址	175
13.5 复位	175
13.6 中断	175
13.6.1 字节传送中断	176
13.6.2 地址检测中断	176
13.6.3 仲裁丢失中断	176
13.7 初始化/应用信息	176
第十四章 AD 转换器(S08ADC10V1)	179
14.1 概述	179
14.2 通道分配	179
14.2.1 交替时钟	179
14.2.2 硬件触发器	180
14.2.2.1 模拟引脚使能	180
14.2.2.2 低功耗模式操作	180
14.2.3 温度传感器	181
14.2.4 特点	182
14.2.5 框图	182
14.3 外部信号描述	183
14.3.1 模拟电源 (V_{DDAD})	183
14.3.2 模拟地 (V_{SSAD})	183
14.3.3 参考高电压 (V_{REFH})	184
14.3.4 参考低电压 (V_{REFL})	184
14.3.5 模拟通道输入 (ADx)	184
14.4 寄存器定义	184
14.4.1 状态和控制寄存器 1 (ADC1SC1)	184
14.4.2 状态和控制寄存器 2 (ADC1SC2)	185

14.4.3 数据高结果寄存器 (ADC1RH)	186
14.4.4 数据低结果寄存器 (ADC1RL)	186
14.4.5 比较值高寄存器 (ADC1CVH)	186
14.4.6 比较值低寄存器 (ADC1CVL)	187
14.4.7 配置寄存器 (ADC1CFG)	187
14.4.8 引脚控制 1 寄存器 (APCTL1)	188
14.4.9 引脚控制 2 寄存器 (APCTL2)	189
14.4.10 引脚控制 3 寄存器 (APCTL3)	189
14.5 功能描述	190
14.5.1 时钟选择和分频控制	190
14.5.2 输入选择和引脚控制	191
14.5.3 硬件触发	191
14.5.4 转换控制	191
14.5.4.1 初始化转换	191
14.5.4.2 完成转换	191
14.5.4.3 终止转换	191
14.5.4.4 电源控制	192
14.5.4.5 总转换时间	192
14.5.5 自动比较功能	193
14.5.6 MCU 等待模式操作	193
14.5.7 MCU stop3 模式操作	193
14.5.7.1 禁止 ADACK 的 stop3 模式	193
14.5.7.2 允许 ADACK 的 stop3 模式	193
14.5.8 MCU stop1 和 stop2 模式操作	193
14.6 初始化信息	194
14.6.1 ADC 模块初始化举例	194
14.6.1.1 初始化顺序	194
14.6.1.2 伪代码举例	194
14.7 应用信息	195
14.7.1 外部引脚和安排	195
14.7.1.1 模拟电源引脚	195
14.7.1.2 模拟参考引脚	195
14.7.1.3 模拟输入引脚	196
14.7.2 错误源	196
14.7.2.1 采样错误	196
14.7.2.2 引脚泄露误差	196
14.7.2.3 噪音错误	196
14.7.2.4 编码宽度和量化误差	197
14.7.2.5 线性误差	197
14.7.2.6 编码抖动、非单调性和遗编码	198
第十五章 开发支持	199
15.1 简介	199
15.1.1 特征	199
15.2 背景调是控制器	200

15.2.1 BKGD 引脚描述.....	200
15.2.2 通信细节.....	201
15.2.3 BDC 命令.....	202
15.2.4 BDC 硬件断点.....	204
15.3 片上调试系统 (DBG)	204
15.3.1 比较器 A 和 B	204
15.3.2 总线信息捕捉和 FIFO 操作	205
15.3.3 流程改变信息.....	205
15.3.4 标记与强制断点和触发器.....	206
15.3.5 触发模式.....	206
15.3.6 硬件断点.....	207
15.4 寄存器定义.....	208
15.4.1 BDC 寄存器和控制位.....	208
15.4.1.1 BDC 状态和控制寄存器 (BDCSCR)	208
15.4.1.2 BDC 断点匹配寄存器 (BDCBKPT)	209
15.4.2 系统背景调试强制复位寄存器 (SBD FR)	210
15.4.3 DBG 寄存器和控制位.....	210
15.4.3.1 调试比较器 A 的高地址页寄存 (DBGCAH).....	210
15.4.3.2 调试比较器 A 的低位寄存器(DBGCAL)	210
15.4.3.3 调试比较器 B 的高地址页寄存(DBGCAH).....	210
15.4.3.4 调试比较器 B 的低位寄存器(DBGCAL)	210
15.4.3.5 调试 FIFO 高地址页寄存器(DBGFH)	210
15.4.3.6 调试 FIFO 低位寄存器(DBGFL).....	211
15.4.3.7 调试控制寄存器 (DBG C)	211
15.4.3.8 调试触发寄存器 (DBG T)	212
15.4.3.9 调试状态寄存器 (DBG S)	212
附录 A 电气特性和时序规范	214
A.1 概述.....	214
A.2 参数分类.....	214
A.3 最大绝对额定值.....	214
A.4 热特性.....	215
A.5 ESD 保护和闭锁抗扰度.....	216
A.6 DC 特性	216
A.7 电源电流特性.....	219
A.8 ADC 特性.....	222
A.9 内部时钟产生模块特性	224
A.9.1 ICG 频率规格	225
A.10 AC 特性.....	227
A.10.1 控制时序	227
A.10.2 定时器/PWM(TPM)模块时序.....	229
A.11 SPI 特性	229
A.12 FLASH 规格	232
A.13 EMC 性能	233

附录 B 订购须知和机械制图	234
B.1 订购须知	234
B.2 机械制图	235

第一章 绪论

1.1 概述

MC9S08AC16 系列属于 8 位微控制单元（MCU），是高性能低功耗的 HCS08 家族中的成员。这个家族中的所有的 MCU 都使用 HCS08 核，并且可以使用多种模块、内存容量、内存类型和封装类型。

注：

1. MC9S08AC16 和 MC9S08AC8 被规定使用在消费和工业的应用中。
2. S9S08AW16A 和 S9S08AW8A 被规定使用在汽车的应用中。

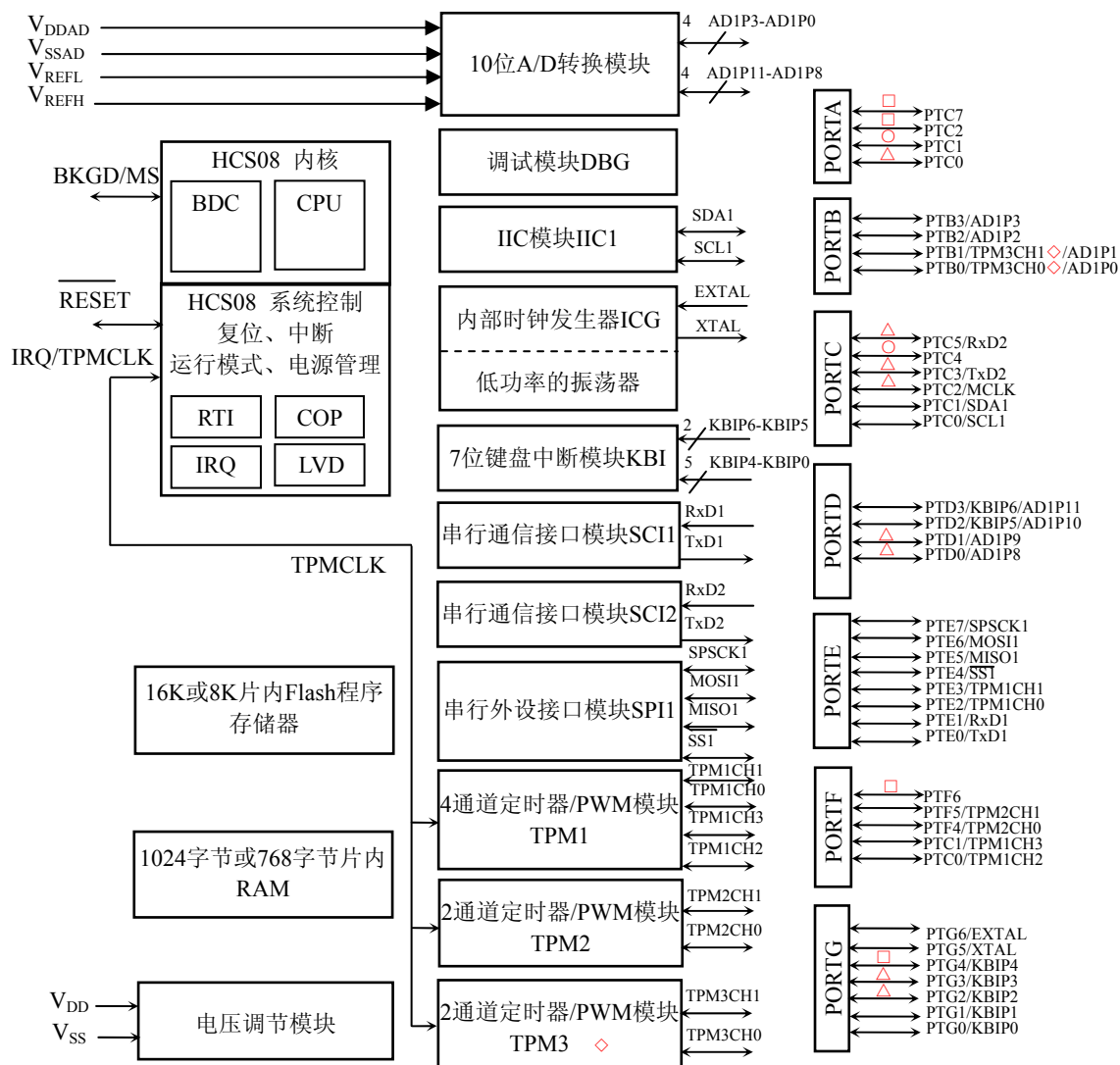
表 1-1 概括了在 MCU 中可用的特征集合。

表 1-1 MCU 的特征和封装

消费工业类的“AC”设备						
特征	MC9S08AC16			MC9S08AC8		
FLASH容量（字节）	16K			8K		
内存容量（字节）	1024			768		
引脚数量	48	44	42	48	44	42
ADC通道	8	8	6	8	8	6
TPM1通道	4	4	2	4	4	2
TPM2通道	2	2	2	2	2	2
TPM3通道	2	2	2	2	2	2
KBI引脚	7	6	4	7	6	4
GPIO引脚	38	34	22	38	34	22
适用于消费工业领域	yes			yes		
适用于汽车领域	no			no		
汽车类的“AW”设备						
特征	S9S08AW16A			MC9S08AW8A		
FLASH容量（字节）	16K			8K		
内存容量（字节）	1024			768		
引脚数量	48	44	42	48	44	42
ADC通道	8	8	6	8	8	6
TPM1通道	4	4	2	4	4	2
TPM2通道	2	2	2	2	2	2
TPM3通道	-	-	-	-	-	-
KBI引脚	7	6	4	7	6	4
GPIO引脚	38	34	22	38	34	22
适用于消费工业领域	no			no		
适用于汽车领域	yes			yes		

1.2 MCU结构框图

MC9S08AC16 系列 MCU 的结构框图如图 1-1 所示。



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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测(IRQEDG=1)，下拉使能。
3. IRQ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。
4. 引脚包含集成的上拉设备。
5. PTD3、PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能(KBIPEn=1)而且上升沿被选择(KBEDGn=1)，下拉使能。

图1-1 MC9S08AC16结构框图

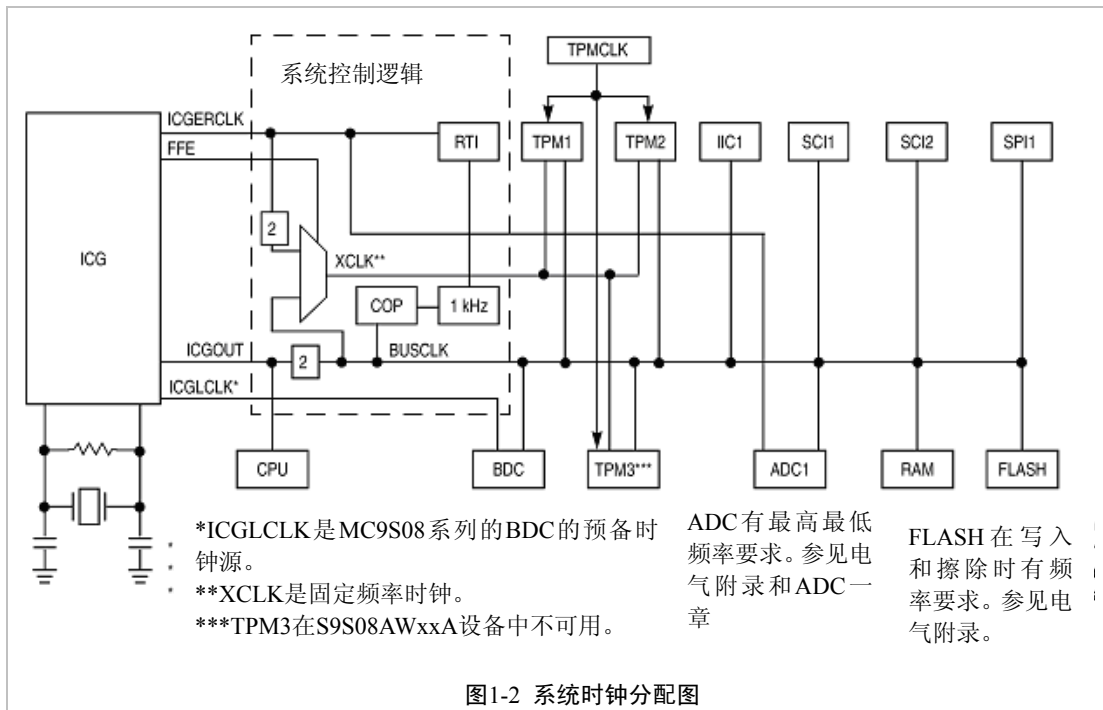
表 1-2 列举了片上模块的版本。

表 1-2 片上模块版本

模块	版本
模数转换(ADC)	1
内部时钟发生器(ICG)	4
内部IC总线(IIC)	2
键盘中断(KBI)	1
串行通信接口(SCI)	4

串行外设接口（SPI）	3
定时器脉宽调制（TPM）	3
中央处理单元（CPU）	2

1.3 系统时钟分配



MCU 内部的一些模块可以选择时钟脉冲源。图 1-2 表示了一个简化的时钟接线图。ICG 提供了时钟脉冲源：

- ICGOUT 是 ICG 模块的输出，它是下列各项中的一个：
 - 外部的晶体振荡器
 - 外部的时钟脉冲源
 - 数控振荡器（DCO）在频率锁定环子模块中的输出
 - ICG 内的控制位决定它连在哪个源上
- FFE 是在 ICG 里面产生的一个控制信号。如果 ICGOUT 的频率 $>4 \times$ （ICGERCLK 的频率，该信号为逻辑 1，固定频率的时钟信号是 ICGERCLK/2。否则固定频率的时钟信号是 BUSCLK
- ICGLCLK——在总线时钟比较慢的系统中，开发工具可以选择内部自发生时钟源（~8MHz）来加速系统的 BDC 通信
- ICGERCLK——外部引用时钟可以作为实时中断的时钟源，也可以作为 ADC 模块的 AKTCKL 的输入

第二章 引脚及其连接

2.1 简介

本章描述了连接到封装引脚上的信号，包括一张管脚图，一张信号表和信号的具体描述。

2.2 设备的管脚分配

图 2-1 表示了 MC9S08AC16 系列设备的 48 引脚 QFN 引脚排列。



图 2-2 中显示了 MC9S08AC16 系列设备的 44 引脚 LQFP 引脚排列。



图2-2 MC9S08AC16系列44引脚LQFP封装¹

1. TPM3 在 S9S08AwxxA 中不可用。

图 2-3 中显示了 MC9S08AC16 系列设备的 32 引脚 LQFP 引脚排列。

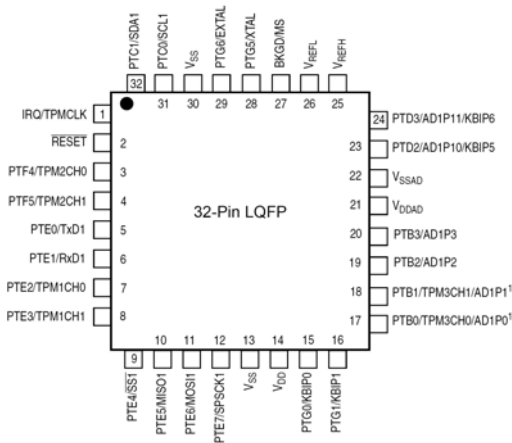


图2-3 MC9S08AC16系列32引脚LQFP封装¹

1. TPM3 在 S9S08AwxxA 中不可用。

表 2-1 各封装的可用引脚

引脚号			最低 优先级 最高			引脚号			最低 优先级 最高		
48	44	32	端口引脚	高1	高2	48	44	32	端口引脚	高1	高2
1	1	—	PTC4			25	—	—	PTA7		
2	2	1		IRQ	TPMCLK	26	23	17	PTB0	TPM3CH0 ¹	AD1P0
3	3	2			RESET	27	24	18	PTB1	TPM3CH1 ¹	AD1P1
4	4	—	PTF0	TPM1CH2		28	25	19	PTB2	AD1P2	
5	5	—	PTF1	TPM1CH3		29	26	20	PTB3	AD1P3	
6	6	3	PTF4	TPM2CH0		30	27	—	PTD0	AD1P8	

7	7	4	PTF5	TPM2CH1		31	28	—	PTD1	AD1P9	
8	—	—	PTF6			32	29	21			V _{DDAD}
9	8	5	PTE0		TxD1	33	30	22			V _{SSAD}
10	9	6	PTE1		RxD1	34	31	23	PTD2	AD1P10	KBIP5
11	10	7	PTE2	TPM1CH0		35	32	24	PTD3	AD1P11	KBIP6
12	11	8	PTE3	TPM1CH1		36	33	—	PTG3		KBIP3
13	12	9	PTE4		SS1	37	—	—	PTG4	KBIP4	
14	13	10	PTE5		MISO1	38	34	25			V _{REFH}
15	14	11	PTE6		MOSI1	39	35	26			V _{REFL}
16	15	12	PTE7		SPSCK1	40	36	27	BKGD	MS	
17	16	13			VSS	41	37	28	PTG5	XTAL	
18	17	14			VDD	42	38	29	PTG6	EXTAL	
19	18	15	PTG0		KBIP0	43	39	30			V _{SS}
20	19	16	PTG1		KBIP1	44	40	31	PTC0		SCL1
21	20	—	PTG2		KBIP2	45	41	32	PTC1		SDA1
22	21	—	PTA0			46	42	—	PTC2		MCLK
23	22	—	PTA1			47	43	—	PTC3		TxD2
24	—	—	PTA2			48	44	—	PTC5		RxD2

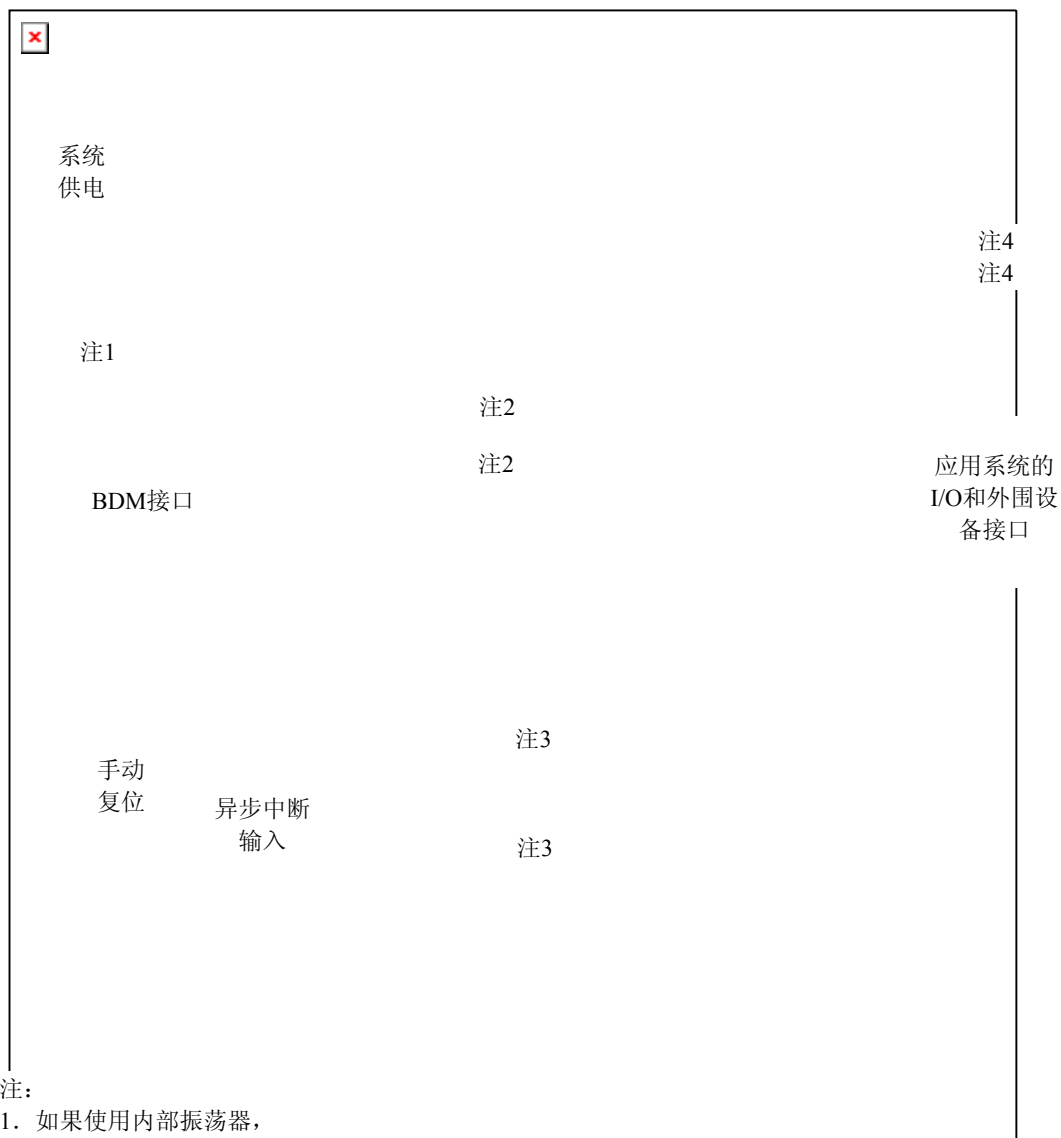
1. TPM3在S9S08AwxxA中不可用。

表 2-2 引脚功能参考

信号功能	举例	参考
端口引脚	PTAx、PTBx	第6章“并行输入输出”
SPI	SS、MISO、MOSI、SPSCK	第12章“SPI”
键盘中断	KBIPx	第9章“键盘中断”
定时器/PWM	TCLK、TPMCHx	第10章“定时器/PWM”
IIC	SCL、SDA	第13章“内部集成电路”
内部集成电路	TxD、RxD	第11章“串行通信接口”
串行通信接口	EXTAL、XTAL	第8章“内部时钟发生器”
晶振/时钟	ADPx	第14章“AD转换转换器”
电源/核	BKGD/MS、V _{DD} 、V _{SS}	第2章“引脚和连接”
复位和中断	RESET、IRQ	第5章“复位，中断和系统配置”

2.3 推荐的系统连接

图 2-1 显示了几乎所有 MC9S08AC16 系列应用系统最典型的引脚连接。



注：

1. 如果使用内部振荡器，这是不需要的。
2. XTAL和EXTAL分别是PTG5、PTG6。
3. 在电磁兼容性应用中，推荐将RESET和ISR接上RC过滤器。
4. TPM3在S9S08AWxxA中不可用。

图2-4 基本系统连接

下面将详细介绍系统连接。

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

V_{DD} 和 V_{SS} 是 MCU 的主要供电引脚。该电压源对所有的输入/输出(I/O)缓冲电路和内部电压调节器供电。内部电压调节器向 CPU 和其他 MCU 内部电路提供稳定的低压源。

通常应用系统有两个单独的电容器和电源引脚相连。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 V_{DD} 和 V_{SS} ，用以抑制高频噪声。MC9S08AC16 还有一个次要 V_{SS} 引脚。这个引脚连接到系统地或经过一个低阻抗连接到主要的 V_{SS} 引脚。

V_{DDAD} 和 V_{SSAD} 也是 MCU 的电源引脚。该电压源给 ADC 模块提供电源。一个 $0.1\ \mu\text{F}$ 的陶瓷旁路电容应尽可能靠近 MCU 的这两个电源引脚，用来抑制高频噪声。

2.3.2 振荡器（XTAL与EXTAL）

复位 MCU 时使用内部产生的频率（自发生时钟模式）相当于 8MHz 晶体频率。这个频率源在复位启动时使用，也可以作为时钟源以避免长时间的晶体启动延时。MCU 也包含了一个内部时钟产生器（ICG）模块，这个模块用来启动 MCU。关于 ICG 的更多信息参考第八章“内部时钟产生器（S08ICGV4）”。

MCU 中的振荡器为传统的皮尔斯振荡器，它可容纳一个晶体或陶瓷谐振器，其频率范围可通过 ICGC1 寄存器中的 RANGE 位在两个频率范围里进行选择。除了晶体或陶瓷谐振器，外部振荡器也能和 EXTAL 输入引脚相连，但 XTAL 输出引脚必须悬空。

下面的讨论将参考图 2-4。R_S(如果使用)和 R_F 应该是低电感的电阻，如碳合成物电阻。绕线电阻和一些金属薄膜电阻有很大的电感值。C1 和 C2 应该是为高频应用专门设计的高品质陶瓷电容。

R_F 在晶振启动时被用来提供一个偏压电路以保持 EXTAL 输入在其线性范围内，它的值通常不是关键的。典型的系统使用 $1\ \text{M}\Omega$ 到 $10\ \text{M}\Omega$ 。高值对湿度敏感，低值减小增益(在特殊情况下)可以阻止启动。

C1 和 C2 通常在 5pF 到 25pF 的范围内，并按照要求和特定的晶体和谐振器相匹配。在给 C1 和 C2 选定大小时一定要考虑到 PCB 板电容和 MCU 引脚电容。晶体制造商通常指定一个负载电容，该电容值应等于 C1 和 C2 的串联值，C1 和 C2 一般选用同样大小的电容。作为首选的近似值，为每个振荡器引脚(EXTAL 和 XTAL)使用约 10pF 的电容作为引脚和 PCB 的连接电容。

2.3.3 复位

复位引脚是一个有上拉电阻的专用引脚。它有输入滞后，一个大的电流输出驱动，没有输出摆率控制。因为有内部上电复位和低压复位电路，所以通常不需要外部的复位电路。这个引脚通常连接到标准 6 引脚的背景调试连接器上，这样开发系统能够直接复位 MCU 系统。如果需要，可以通过附加一个简单的接地开关手动复位(下拉复位引脚强制复位)。

每当有复位产生(无论来自外部信号还是来自内部系统)，复位引脚被拉低约 34 总线周期。复位电路解析复位的原因并且在复位状态寄存器(SRS)中设置相应的位。

在对 EMC 敏感的应用中，在复位引脚上推荐一个外部的 RC 过滤器。参见图 2-4 的例子。

2.3.4 背景/模式选择(BKGD/MS)

在复位时，背景/模式选择(BKGD/MS) 引脚的功能是模式选择。在复位升起后，这个引脚立即变为背景脚，可用作背景调试。作为背景/模式选择引脚时，这个引脚包括一个内部上拉设备，输入滞后，一个标准的输出驱动和无输出摆率控制。

若这个引脚没有连接，则 MCU 在复位的上升沿进入正常的操作模式。如果调试系统被连接到 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

和总线时钟频率一样快，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。

2.3.5 ADC参考引脚（ V_{REFH} 与 V_{REFL} ）

V_{REFH} 和 V_{REFL} 分别对应输入到 ADC 模块的参考高电压和参考低电压。

2.3.6 外部的中断引脚（IRQ）

IRQ 引脚既是 IRQ 中断的输入源也是 BIH 和 BIL 指令的输入。如果中断请求函数未允许，这个引脚不会产生任何作用。

在对 EMC 敏感的应用中，在复位引脚上推荐一个外部的 RC 过滤器。参见图 2-4 的例子。

2.3.7 通用I/O及外设端口

其他的引脚分配给通用 I/O 和片上的外围功能，例如定时器和串行 I/O 系统。一旦复位，所有这些引脚都设定为高阻通用输入口，内部上拉设备无效。

注：

为了避免来自浮动输入引脚上的额外的电流消耗，应用程序中的复位初始化程序应使能片上上拉设备或将闲置引脚的方向改变为输出，使引脚不浮动。

有关使用这些引脚作为通用 I/O 引脚的信息，参考第六章“并行输入输出”。或者有关片上外设系统如何以及何时用这些引脚的信息，请参考表 2-2 中适当的章节。

当一个片上外设系统使用一个引脚时，即使外设模块通过控制引脚输出缓冲使能来控制引脚方向，但是数据方向控制位仍决定从端口数据寄存器读出的是什么。参考第六章“并行输入输出”有更多信息。

只要引脚作为输入，即使这些引脚被片上外设模块控制，但是它们的上拉使能位仍然可以控制这些引脚是否有片上的上拉设备。当 PTD3, PTD3 和 PTG4 引脚被 KBI 模块控制并且被设定为上升沿/高电平敏感时，上拉使能控制位使能下拉设备而不是上拉设备。相似的，当 IRQ 作为 IRQ 引脚而且被规定为检测上升沿，上拉使能控制位使能下拉设备而不是使能上拉设备。

注意：

当一个可选的功能第一次被使能时，模块可能会得到一个伪边。用户软件应该在中断使能前清除任何相关的标志位。当多个模块被使能，表 2-1 说明了优先级关系。最高优先级的模块将会控制引脚。在引脚上有一个较低优先级的功能下，选择这个引脚更高优先级的功能会产生伪边给较低优先级模块。建议在使能另一个模块前所有的模块共享一个无效的引脚。

第三章 工作模式

3.1 简介

本章讲述了 MC9S08AC16 系列的工作模式，并描述了如何进入和退出各个模式及各个模式的功能。

3.2 特征

- 普通用户在运行模式下操作
- 等待模式：
 - CPU 关闭以节能
 - 系统时钟仍然运行
 - 维持完全的电压稳压
- 停止模式：
 - 系统时钟停止；电压调节器待命
 - 停止 2：内部电路的部分断电，RAM 内容保留
 - 停止 3：所有内部电路供电，以快速恢复

3.3 运行模式

这是 MC9S08AC16 的正常操作模式。当 BKGD/MS 在复位的上升沿时是高电平，此模式被选择。在此模式下复位之后，CPU 从存储器中的\$FFFE:\$FFFF 处取出程序执行的起始地址，执行内存中的代码。

3.4 背景调试模式

背景调试模式的功能由 HCS08 核中的背景调试控制器(BDC，Background Debug Controller)管理。BDC 和片上调试模块(DBG)一起，提供了在软件运行时分析 MCU 的操作的方法。

可通过以下五种方式进入背景调试模式：

- BKGD/MS 引脚在复位的上升沿时是低电平
- 从 BKGD 引脚接收到 BACKGROUND 命令
- 当执行 BGND 指令时
- 当遇到 BDC 断点时
- 当遇到 DBG 断点时

进入背景调试模式后，CPU 被置于挂起状态，等待串行背景命令而不是执行用户应用程序的指令。

背景命令有两种类型：

- 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 MC9S08AC16 从 Freescale 出厂时，除非特别指明，FLASH 存储器默认为擦除的，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十五章开发支持部分。

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栈操作开始运行。

在等待模式下，有些背景调试命令的使用限制，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。在这两种停止模式中，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。

HCS08 设备被设计成低压操作（1.8V 到 3.6V），也包括停止 1 模式。MC9S08AC16 系列设备不包括停止 1 模式。

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	时钟模块	ADC	电压调节器	I/O	RTI
STOP2	1	关	待命	关	不可用	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

(1) STOP3 模式下晶振可以设置为运行，参见 ICG 寄存器。

3.6.1 Stop2模式

STOP2 模式提供非常低的待命电压能保持 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 指令前必须选择停止 2 模式（PPDC=1）和使能停止模式（STOPE=1）。另外，LVD 在停止模式（LVDSE=LVDE=1）下不能操作。如果 LVD 在停止模式下使能，不管 PPDC 的状态是什么，MCU 在执行 STOP 指令后都会进入停止 3 模式。

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($\overline{\text{RESET}}$ 或者 IRQ/TPMCLK)或者 RTI 中断可离开 STOP2 模式。不管在进入 STOP2 之前 IRQ/TPMCLK 是如何设置的，它在 STOP2 模式下总是低电平输入有效。

当从 STOP2 模式中被唤醒后，MCU 将以类似上电复位（POR）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 端口状态保持闭锁，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 端口闭锁打开时，寄存器中的内容将是它们的复位默认状态的值，I/O 端口转换成复位状态。

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，引脚将被它们的相关端口控制寄存器控制。

3.6.2 Stop3模式

为了进入 STOP3 模式，在执行 STOP 指令前必须选择停止 3 模式（PPDC=1）和使能停止模式（STOPE=1）。进入 STOP3 模式后，所有 MCU 的时钟、包括振荡器本身都停止了。时钟模块(ICG)，电压调节器和 ADC 都在待命状态。所有内部寄存器和逻辑，也包括 RAM 的内容都被保持。I/O 端口的状态不像 STOP2 模式中一样被锁定。取而代之的是，使用内部逻辑的状态控制需要保持的引脚的特点来保持它们的状态。

退出 STOP3 模式的方法有：复位 $\overline{\text{RESET}}$ ，或者一个由实时中断（RTI），LVD 系统，ADC，IRQ，KBI 或 SCI 产生的中断。

如果通过 $\overline{\text{RESET}}$ 退出 STOP3 模式，MCU 将复位，并从复位向量处执行。如果通过异步中断或实时中断退出，MCU 将从相应的中断向量处执行。

实时中断的单独时钟脉冲源(约等于 1kHz)允许在没有额外部件的情况下将 MCU 从 STOP2 或 STOP3 模式唤醒。当 RTIS2:RTIS1:RTIS0=0:0:0，实时中断功能和 1kHz 的时钟源不可用。当 1kHz 的时钟源不可用时，功耗更低，但此种情况下实时中断不可以将 MCU 从

STOP 模式中唤醒。

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 寄存器将在第十五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 STOP1 或 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 总结了当背景调试模块使能时 MCU 在 STOP 模式下的行为。

表 3-2 MCU 在启用背景调试模块的 STOP 模式下的行为

模式	PPDC	CPU, 数字外设, FLASH	RAM	ICG	ADC	电压调节器	I/O引脚	RTI
STOP3	任意	待命	待命	活跃	不可用	活跃	状态保持	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 SPMSC1 寄存器中的 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 STOP1 或 STOP2 模式。表格 3-3 总结了在 LVD 使能时 MCU 在 STOP 模式下的行为。

表 3-3 MCU 在启用 LVD 的 STOP 模式下的行为

模式	PPDC	CPU、数字外设、FLASH	RAM	ICG	ADC	电压调节器	I/O	RTI
STOP3	任意	待命	待命	关闭	可选开	活跃	状态保持	可选开

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 3.6.2 STOP3 模式以获得停止模式下系统行为的更多信息。

I/O 引脚:

- 当 MCU 在 STOP3 模式下，所有 I/O 端口状态保持不变。
- 如果 MCU 配置为进入 STOP2 模式，所有 I/O 端口状态在进入停止模式前锁定。

存储器:

- 所有 RAM 和寄存器内容在 STOP3 模式下都是被保存的。
- 从 STOP2 模式下唤醒后，所有寄存器都将复位，但 RAM 内容保存，端口状态保持锁定直到 PPDACK 位被写入。用户可以在进入 STOP2 模式之前，将寄存器的内容保存在 RAM 中，并在离开 STOP2 模式后恢复这些数据。
- FLASH 的内容是非易失的，在任何停止模式下都会保存。

ICG——在 STOP3 模式下，ICG 进入低功耗待命状态。当 ICG 在待命状态时，通过设置适当的控制位(OSCSTEN)，振荡器可以保持工作。在 STOP2 模式下，ICG 是关闭的。即使 OSCSTEN 使能，振荡器也不能在 STOP2 下工作。MCU 从 STOP2 模式退出时从停止模式中唤醒，TPM 模块被复位并且必须被重新初始化。

TPM——当 MCU 进入停止模式时，TPM1 和 TPM2 模块的时钟停止。模块停止工作。如果 MCU 配置成进入 STOP2 模式，当从停止模式唤醒以后 TPM 模块被复位并且必须被重新初始化。

ADC——当 MCU 进入停止模式时，除非异步时钟源，ADACK 无效，否则 ADC 将进入低功耗待命状态。如果 MCU 配置为进入 STOP2，当从停止模式唤醒以后 ADC 模块被复位并且必须被重新初始化。

KBI——在 STOP3 模式中，KBI 引脚仍然可用，并且作为一个中断源可以将 MCU 从 STOP3 模式唤醒。KBI 在 STOP2 中不可用，当从 STOP 模式中唤醒时，KBI 不能使用必须重新初始化。

SCI——当 MCU 进入停止模式时，SCI1 和 SCI2 的时钟停止。模块停止操作。如果 MCU 配置为进入 STOP2 或者 STOP1 模式，当从停止模式唤醒以后 SCI 模块被复位并且必须被重新初始化。

SPI——当 MCU 进入停止模式时，SPI 的时钟停止。模块停止操作。如果 MCU 配置为进入 STOP2 模式，当从停止模式唤醒以后 SPI 模块被复位并且必须被重新初始化。

IIC——当 MCU 进入停止模式时，IIC 的时钟停止。模块停止操作。如果 MCU 配置为进入 STOP2 模式，当从停止模式唤醒以后 IIC 模块被复位并且必须被重新初始化。

电压调节器——当 MCU 进入任何停止模式时，电压调节器将进入低功耗待命状态，除非 LVD 在停止模式下使能或者 BDM 使能。

第四章 片上存储器

4.1 简介MC9SAC16系列存储映像

图 4-1 描述了 MC9S08AC16 系列 MCU 的存储映像，MC9S08AC16 的片上存储器由 RAM、非易失的 FLASH 存储器、输入/输出和控制/状态寄存器组成。

这些寄存器分成三组：

- 直接页寄存器(0x0000-0x006F)
- 高地址页寄存器(0x1800-0x185F)
- 非易失性寄存器(0xFFB0-0xFFBF)



图 4-1 MC9S08AC16 系列内存映像

4.1.1 复位和中断向量分配

表 4-1 描述了复位与中断向量的地址分配，至于其他 HCS08 系列 MCU 的向量名字和地址分配，请参考相关的数据手册。表中所示向量名是 Freescalse 公司提供的 MC9S08AC16 文档中给出的名字。有关复位、中断、中断优先权和中断屏蔽控制的详细信息，请参阅第 5 章复位和中断。

表 4-1 复位和中断向量表

地址(高/低)	向量	向量名称
0xFFC0:FFC1 ↑ ↓ 0xFFC4:FFC5	不用的向量空间 (可用于用户程序)	—
0xFFC6:FFC7	TPM3 ¹ 溢出	Vtpm3ovf
0xFFC8:FFC9	TPM3 ¹ 通道1	Vtpm3ch1
0xFFCA:FFCB	TPM3 ¹ 通道0	Vtpm3ch0
0xFFCC:FFCD	RTI1	Vrti
0xFFCE:FFCF	IIC1	Viic1
0xFFD0:FFD1	ADC1模数转换	Vadc1
0xFFD2:FFD3	KBI	Vkeyboard1
0xFFD4:FFD5	SCI2发送	Vsci2tx
0xFFD6:FFD7	SCI2接收	Vsci2rx
0xFFD8:FFD9	SCI2错误	Vsci2err
0xFFDA:FFDB	SCI1发送	Vsci1tx
0xFFDC:FFDD	SCI1接收	Vsci1rx
0xFFDE:FFDF	SCI1错误	Vsci1err
0xFFE0:FFE1	SPI1	Vspi1
0xFFE2:FFE3	TPM2溢出	Vtpm2ovf
0xFFE4:FFE5	TPM2通道1	Vtpm2ch1
0xFFE6:FFE7	TPM2通道0	Vtpm2ch0
0xFFE8:FFE9	TPM1溢出	Vtpm1ovf
0xFFEA:FFEB	未使用	—
0xFFEC:FFED	未使用	—
0xFFEE:FFEF	TPM1通道3	Vtpm1ch3
0xFFFF0:FFF1	TPM1通道2	Vtpm1ch2
0xFFFF2:FFF3	TPM1通道1	Vtpm1ch1
0xFFFF4:FFF5	TPM1通道0	Vtpm1ch0
0xFFFF6:FFF7	ICG	Vicg
0xFFFF8:FFF9	低电压检测	Vlvd
0xFFFFA:FFFB	IRQ	Virq
0xFFFFC:FFFD	SWI	Vswi
0xFFFFE:FFFF	RESET	Vreset

(1) TPM3在S9S08AWxxA中不可以

4.2 寄存器地址和位分配

MC9S08AC16 系列中的寄存器分为以下三组：

- 直接页寄存器位于存储空间前 112 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：

- 在复位时加载到工作寄存器的两个值。
- 用户可以使用一个 8 字节的密码取得加密存储器的使用访问权。

因为非易失性寄存器是存储在 FLASH 存储器中的，所以它必须像其他的 FLASH 存储器那样擦除和写入。

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

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 中的寄存器可以使用比较高效的直接寻址方式访问，这种方式只需要地址的低位字节。因此，第一纵列中地址的低位字节用粗体字表示。表 4-3 和表 4-4 中，第一纵列中完整的地址用粗体字表示。表 4-2，表 4-3，表 4-4 中，第二纵列的寄存器的名称亦用粗体字表示，以与右边的字节名称区分。没有定义的字节单元格以阴影表示。有 0 的阴影单元格表示不使用的字节总是读作 0。有横线的阴影单元格表示不使用或保留的字节，可以读作 1 或 0。

表 4-2 直接页寄存器概述

地址	寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$0000	PTAD	PTAD7	R	R	R	R	PTAD2	PTAD1	PTAD0
\$0001	PTADD	PTADD7	R	R	R	R	PTADD2	PTADD1	PTADD0
\$0002	PTBD	R	R	R	R	PTBD3	PTBD2	PTBD1	PTBD0
\$0003	PTBDD	R	R	R	R	PTBDD3	PTBDD2	PTBDD1	PTBDD0
\$0004	PTCD	0	R	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
\$0005	PTCDD	0	R	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
\$0006	PTDD	R	R	R	R	PTDD3	PTDD2	PTDD1	PTDD0
\$0007	PTDDD	R	R	R	R	PTDDD3	PTDDD2	PTDDD1	PTDDD0
\$0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
\$0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
\$000A	PTFD	R	PTFD6	PTFD5	PTFD4	R	R	PTFD1	PTFD0
\$000B	PTCDD	R	PTFDD6	PTFDD5	PTFDD4	R	R	PTFDD1	PTFDD0
\$000C	PTGD	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
\$000D	PTGDD	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
\$000E	保留	—	—	—	—	—	—	—	—
\$000F	保留	—	—	—	—	—	—	—	—
\$0010	ADC1SC1	COCO	AIEN	ADCO	ADCH				
\$0011	ADC1SC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
\$0012	ADC1RH	0	0	0	0	0	0	ADR9	ADR8
\$0013	ADC1RL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
\$0014	ADC1CVH	0	0	0	0	0	0	ADCV9	ADCV8
\$0015	ADC1CVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
\$0016	ADC1CFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
\$0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
\$0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
\$0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
\$001A	保留	—	—	—	—	—	—	—	—
\$001B	保留	—	—	—	—	—	—	—	—
\$001C	IRQSC	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
\$001D	保留	—	—	—	—	—	—	—	—
\$001E	KBISC	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
\$001F	KBIPE	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
\$0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
\$0022	TPM1CNTH	Bit 7	6	5	4	3	2	1	Bit 0
\$0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
\$0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
\$0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
\$0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0
\$0028	TPM1C1SC	CH1F	CH1IE	MA1B	MS1A	ELS1B	ELS1A	0	0
\$0029	TPM1C1VH	Bit 15	14	13	12	11	10	9	Bit 8
\$002A	TPM1C1VL	Bit 7	6	5	4	3	2	1	Bit 0
\$002B	TPM1C2SC	CH2F	CH2IE	MS2B	ELS2B	MS2A	ELS2A	0	0

\$002C	TPM1C2VH	Bit 15	14	13	12	11	10	9	Bit 8
\$002D	TPM1C2VL	Bit 7	6	5	4	3	2	1	Bit 0
\$002E	TPM1C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
\$002F	TPM1C3VH	Bit 15	14	13	12	11	10	9	Bit 8
\$0030	TPM1C3VL	Bit 7	6	5	4	3	2	1	Bit 0
\$0031	保留	—	—	—	—	—	—	—	—
\$0032	保留	—	—	—	—	—	—	—	—
\$0033	保留	—	—	—	—	—	—	—	—
\$0034	保留	—	—	—	—	—	—	—	—
\$0035	保留	—	—	—	—	—	—	—	—
\$0036	保留	—	—	—	—	—	—	—	—
\$0037	保留	—	—	—	—	—	—	—	—
\$0038	SCI1BDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
\$0039	SCI1BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
\$003A	SCI1C1	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x003B	SCI1C2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x003C	SCI1S1	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x003D	SCI1S2	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
0x003E	SCI1C3	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
0x003F	SCI1D	Bit 7	6	5	4	3	2	1	Bit 0
0x0040	SCI2BDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0041	SCI2BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x0042	SCI2C1	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x0043	SCI2C2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x0044	SCI2S1	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x0045	SCI2S2	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
0x0046	SCI2C3	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
0x0047	SCI2D	Bit 7	6	5	4	3	2	1	Bit 0
0x0048	ICGC1	HGO	RANGE	REFS	CLKS		OSCSTEN	LOCD	0
0x0049	ICGC2	LOLRE	MFD			LOCRE	RFD		
0x004A	ICGS1	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
0x004B	ICGS2	0	0	0	0	0	0	0	DCOS
0x004C	ICGFLTU	0	0	0	0	FLT			
0x004D	ICGFLTL	FLT							
0x004E	ICGTRM	TRIM							
0x004F	保留	—	—	—	—	—	—	—	—
0x0050	SPI1C1	SPIE	SPE	SPTIE	MSTR	CPOL			
0x0051	SPI1C2	SPMIE	0	0	MODFEN	BIDIROE	SPIMODE	SPISWAI	SPC0
0x0052	SPI1BR	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
0x0053	SPI1S	SPRF	SPIMF	SPTEF	MODF	0	0	0	0
0x0054	SPI1DH	Bit 15	14	13	12	11	10	8	Bit 8
\$0055	SPI1DL	Bit 7	6	5	4	3	2	1	Bit 0
\$0056	SPI1MH	Bit 15	14	13	12	11	10	8	Bit 8
\$0057	SPI1ML	Bit 7	6	5	4	3	2	1	Bit 0
\$0058	IIC1A	AD7	AD6	AD5	AD4	AD3	AD2	AD1	0
0x0059	IIC1F	MULT		ICR					
0x005A	IIC1C	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
0x005B	IIC1S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
0x005C	IIC1D	DATA							
0x005D	IIC1C2	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8

0x005E	保留	—	—	—	—	—	—	—	—
0x005F	保留	—	—	—	—	—	—	—	—
0x0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0061	TPM2CNTH	Bit 15	14	13	12	11	10	8	Bit 8
\$0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
\$0063	TPM2MODH	Bit 15	14	13	12	11	10	8	Bit 8
\$0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0066	TPM2C0VH	Bit 15	14	13	12	11	10	8	Bit 8
\$0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
\$0069	TPM2C1VH	Bit 15	14	13	12	11	10	8	Bit 8
\$006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
\$006B	保留	—	—	—	—	—	—	—	—
\$006C	保留	—	—	—	—	—	—	—	—
\$006D	保留	—	—	—	—	—	—	—	—
\$006E	保留	—	—	—	—	—	—	—	—
\$006F	保留	—	—	—	—	—	—	—	—

表 4-3 中所示的高地址页寄存器比其他的 I/O 和控制寄存器要少访问，所以它们位于可直接寻址的存储器空间之外，开始于\$1800。

表 4-3 高地址页寄存器概要

地址	寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$1800	SRS	POR	PIN	COP	ILCP	0	ICG	LVD	0
\$1801	SBD FR	0	0	0	0	0	0	0	BDFR
\$1802	SOPT	COPE	COPT	STOPE	-	0	0	-	-
\$1803	SBD FR	0	0	0	MPE	0	MCSEL		
\$1804- \$1805	保留	- -	- -	- -	- -	- -	- -	- -	- -
\$1806	SDIDH	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
\$1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
\$1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
\$1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0 ¹	BGBE
\$180A	SPMSC2	LVWF	LVWACK	LVDV	LVWV	PPDF	PPDACK	-	PPDC
\$180B	保留	-	-	-	-	-	-	-	-
\$180C	SOPT2	-	-	-	-	-	-	-	-
\$180D- \$180F	保留	- -	- -	- -	- -	- -	- -	- -	- -
\$1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
\$1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
\$1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
\$1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
\$1814	DBGFHH	Bit 15	14	13	12	11	10	9	Bit 8
\$1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
\$1816	DBGC	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
\$1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
\$1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
\$1819- \$181F	保留	- -	- -	- -	- -	- -	- -	- -	- -
\$1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
\$1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
\$1822	保留	-	-	-	-	-	-	-	-
\$1823	FCNFG	0	0	KEYACC	0	0	0	0	0
\$1824	FPROT	FPOPEN	FPDIS	FPS2	FPS1	FPS0	0	0	0

\$1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
\$1826	FCMD	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
\$1827-\$182F	保留	-	-	-	-	-	-	-	-
0x1830	TPM3SC ²	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x1831	TPM3CNTH ²	Bit 15	14	13	12	11	10	9	Bit 8
0x1832	TPM3CNTL ²	Bit 7	6	5	4	3	2	1	Bit 0
0x1833	TPM3MODH ²	Bit 15	14	13	12	11	10	9	Bit 8
0x1834	TPM3MODL ²	Bit 7	6	5	4	3	2	1	Bit 0
0x1835	TPM3C0SC ²	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x1836	TPM3C0VH ²	Bit 15	14	13	12	11	10	9	Bit 8
0x1837	TPM3C0VL ²	Bit 7	6	5	4	3	2	1	Bit 0
0x1838	TPM3C1SC ²	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x1839	TPM3C1VH ²	Bit 15	14	13	12	11	10	9	Bit 8
0x183A	TPM3C1VL ²	Bit 7	6	5	4	3	2	1	Bit 0
\$183B-\$183F	保留	-	-	-	-	-	-	-	-
0x1840	PTAPE	PTAPE7	R	R	R	R	PTAPE2	PTAPE1	PTAPE0
0x1841	PTASE	PTASE7	R	R	R	R	PTASE2	PTASE1	PTASE0
0x1842	PTADS	PTADS7	R	R	R	R	PTADS2	PTADS1	PTADS0
0x1843	保留	—	—	—	—	—	—	—	—
0x1844	PTBPE	R	R	R	R	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0x1845	PTBSE	R	R	R	R	PTBSE3	PTBSE2	PTBSE1	PTBSE0
0x1846	PTBDS	R	R	R	R	PTBDS3	PTBDS2	PTBDS1	PTBDS0
0x1847	保留	—	—	—	—	—	—	—	—
0x1848	PTCPE	0	R	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0x1849	PTCSE	0	R	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
0x184A	PTCDS	0	R	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
0x184B	保留	—	—	—	—	—	—	—	—
0x184C	PTDPE	R	R	R	R	PTDPE3	PTDPE2	PTDPE1	PTDPE0
0x184D	PTDSE	R	R	R	R	PTDSE3	PTDSE2	PTDSE1	PTDSE0
0x184E	PTDDS	R	R	R	R	PTDDS3	PTDDS2	PTDDS1	PTDDS0
0x184F	保留	—	—	—	—	—	—	—	—
0x1850	PTEPE	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
0x1851	PTESE	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
0x1852	PTEDS	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
0x1853	保留	—	—	—	—	—	—	—	—
0x1854	PTFPE	R	PTFPE6	PTFPE5	PTFPE4	R	R	PTFPE1	PTFPE0
0x1855	PTFSE	R	PTFSE6	PTFSE5	PTFSE4	R	R	PTFSE1	PTFSE0
0x1856	PTFDS	R	PTFDS6	PTFDS5	PTFDS4	R	R	PTFDS1	PTFDS0
0x1857	保留	—	—	—	—	—	—	—	—
0x1858	PTGPE	0	PTGPE6	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x1859	PTGSE	0	PTGSE6	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x185A	PTGDS	0	PTGDS6	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
0x185B	保留	—	—	—	—	—	—	—	—
0x185F		-							

表 4-5 中所示的非易失性 FLASH 寄存器位于 FLASH 存储器中。包括两个 FLASH 存储器模块初始化寄存器和一个 8 字节的密钥，这个密钥用于获得对加密存储器的访问权。在复位后，FLASH 存储器的非易失性寄存器中 NVPROT 和 NVOPT 的内容被转移到相应的位于高页面寄存器中的 FPROT 和 FOPT 工作寄存器中，从而控制加密和阻止保护选项。

表 4-4 非易失性寄存器概要

地址	寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$FFB0-\$FFB7	NVBACKKEY	8字节密钥							
\$FFB8-\$FFBC	保留	-	-	-	-	-	-	-	-
\$FFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
\$FFBE	保留(1)	-	-	-	-	-	-	-	-
\$FFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

注：1、这个位置可用于存储ICG的工厂预设值。

如果密钥使能位(KEYEN)为“1”，8字节的密钥将被用于暂时性的跳出存储器的安全保护模式。密钥机制只能通过安全模式下运行的用户代码访问(密钥不可直接通过背景调试命令键入)。当KEYEN标志位为0时，密钥功能失效。如果密钥不可用，离开安全保护模式的唯一途径就是整体擦除FLASH存储器(通常通过背景调试接口)，并且校验FLASH存储器为空。为避免在下次复位后又返回安全保护模式，向安全位(SEC01:SEC00)写入(1:0)进入非安全状态即可。

4.3 RAM

MC9S08AC16系列包含静态RAM。可以使用更高效的直接地址寻址方式访问RAM中低于0x0100的地址。这一区域的每一位都可以用位操作指令(BCLR, BSET, BRCLR, BRSET)来访问。将访问最频繁的程序变量放在这个区域。

当MCU处于低压等待，STOP2，STOP3模式时，RAM保持数据。上电时，RAM的内容被重新初始化。如果所提供的电压不小于RAM所需的最小电压，RAM数据不会被任何复位影响。

为了和以前的M68HC05 MCU兼容，HCS08初始化时栈指针定为0x00FF。在MC9S08AC16系列中，通常重新初始化栈指针指向RAM的顶端，这样直接页RAM可以用于频繁访问的RAM变量和可位寻址的程序变量。在重启初始化程序中包括接下来的两条指令（在Freescale提供的文件中RamLast等于RAM的最高地址）。

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

4.4 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。允许通过单一连线的背景调试接口对整个数组写入。因为对FLASH的擦除和写入操作不需要特殊的电压，在线应用程序编程(IAP)可以通过一些其他软件控制的通信方式进行下载。在线编程和IAP的详细信息请参见“HCS08 Family Reference Manual, Volume I”，它的飞思卡尔文档顺序号是HCS08RMv1/D。

4.4.1 特征

FLASH有以下特征：

- FLASH容量
 - MC9S08AC16和MC9S08AW16：16384字节（每页512字节，共32页）
 - MC9S08AC8和MC9S08AW8：8192字节（每页512字节，共16页）

- 一个电源提供写入和擦除。
- 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，可高达 100000 次写入/擦除
- 灵活易用的块保护
- FLASH 和 RAM 的安全机制
- 自动低功耗低频率读访问

FLASH 存储模块包括完整的擦除/写入电压发生器和独立的指令处理状态机，可以实现自动字节写入，页擦除(512 字节的 FLASH)、整体擦除或空白检测命令的功能。这些命令是写在命令接口中的，状态标志可以报告错误并且可以指示命令是什么时候完成的。

在 FLASH 存储区的末端的 512、1K、2K、4K、8K、16K 或者 32K 字节的块可以进行块保护。另外的控制位允许对整个 FLASH 区的 60K 字节进行块保护(见 4.7.4 节 FLASH 保护寄存器(FPROT 和 NVFPROT))。可以通过设置非易失性的初始化寄存器(NVFPROT)来设置块保护。可以使用安全机制来预防对 FLASH 和 RAM 存储内容的非法访问。用户可以使用密钥来取得安全模式下的存储器内容的访问控制。

4.4.2 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 4.6.1 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。如果访问错误标志位 (FSTAT 的 FACCERR) 被置位 FCDIV 不能被写。用户在写 FCDIV 寄存器前确保 FACCERR 没有置位。设置好的时钟周期 ($1/f_{FCLK}$) 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 描述了写入和擦除的时间。总线时钟频率和 FCDIV 决定了 FCLK 的频率(f_{FCLK}) FCLK 一个周期的时间是 $t_{FCLK}=1/f_{FCLK}$ 。写入和擦除时间由许多 FCLK 时钟周期组成，绝对时间是 $t_{FCLK}=5\mu s$ 。这个时间也包括了命令状态机和允许与禁止写入和擦除电压使用的时间。

表 4-5 写入和擦除时间

参数	FCLK周期	时间 (FCLK=200KHz)
字节写入	9	45 μs
字节写入 (突发)	4	20 μs ¹
页擦除	4000	20ms
整体擦除	20000	100ms

(1)不包括开始/结束的时间花费

4.4.3 写入和擦除命令的执行

执行任何命令的步骤如下所列。FCDIV 寄存器必须被初始化，并且在开始执行命令前要将所有的错误标志清零。命令的执行步骤是：

- 向要擦除的 FLASH 区域的一个地址写入一个数值。这个写入的地址和数值的信息将被放在命令缓冲区中。命令序列的第一步就是进行这个写入。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 字节的一整页

是 FLASH 中可以擦除的最小块。在 60K 版本中，用户可存取的块小于 512 字节的情况有两种，分别是 RAM 后面的第一页和高地址寄存器。

注：

完成一次写入操作后，FLASH 内的每个字节只能被写入一次。如果一个位已经被写入，再次向这个字节写入一些位需要先擦除这个字节或整体擦除 FLASH。未经擦除而直接写入会影响存储在 FLASH 内的数据。

- 向 FCMD 写入已经设计好的命令。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。
- 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。

必须遵守严格的监控程序，否则命令不会被接受。这减小了无意中改变 FLASH 内容的可能性。命令完成标志 (FCCF) 指示一个命令完成。通过清零 FCBEF 来开始执行命令，命令序列被完成。图 4-2 所有命令执行的流程图除了突发模式写入。使用任何 FLASH 命令前，FCDIV 寄存器必须被初始化。这些必须紧接着复位完成。

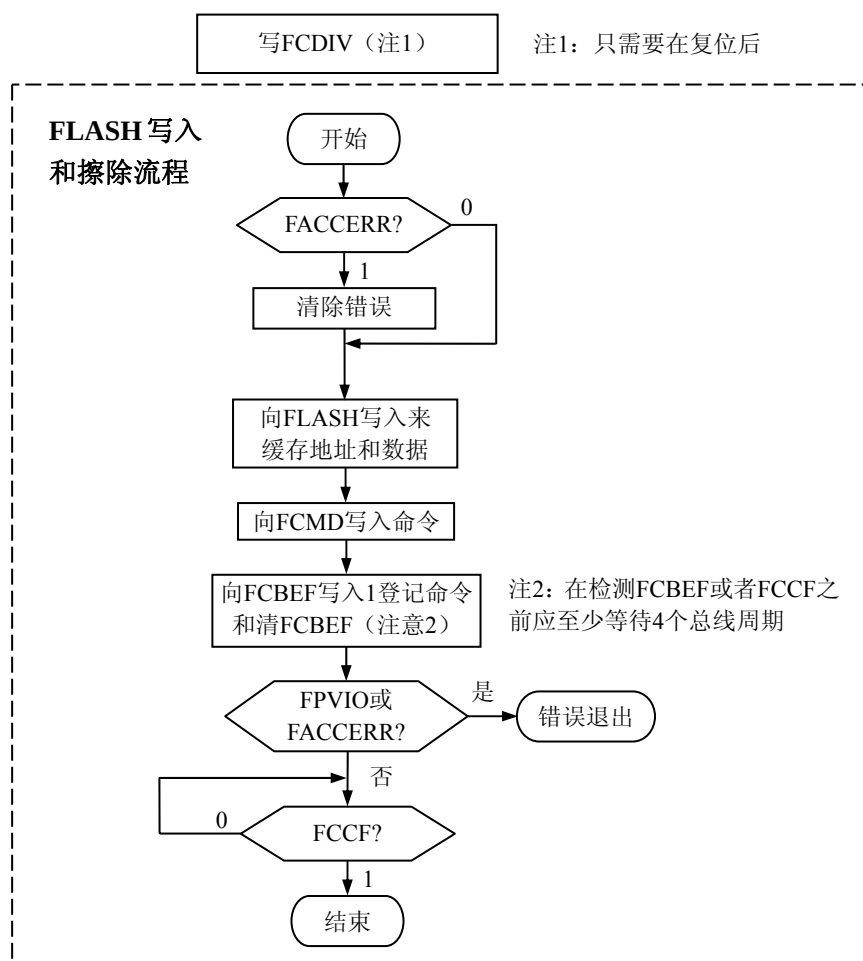


图4-2 FLASH命令流程图

4.4.4 突发模式写入的执行

突发模式写入通常写入连续字节的数据，与标准写入命令相比需要更少的时间。这可能是因为写入时 FLASH 阵列中的高电压不需要失效。通常，当发布写入或擦除命令时，和 FLASH 存储器相关的内部电荷泵必须被允许来为阵列提供高电压。命令完成后电荷泵被关闭。当发布突发模式写入命令时，电荷泵被允许。如果满足下面两个条件，当突发模式写入操作完成，电荷泵仍然保持打开。

- 当前的写入操作完成以前，下一个突发模式写入命令正在等待。
- 下一个连续地址选择了和当前正在写入的字节具有相同的物理行的字节。FLASH 内存行包括 64 个字节。每行的字节通过地址的 A5 到 A0 选择。一个新行的起始地址中的 A5 到 A0 全零。

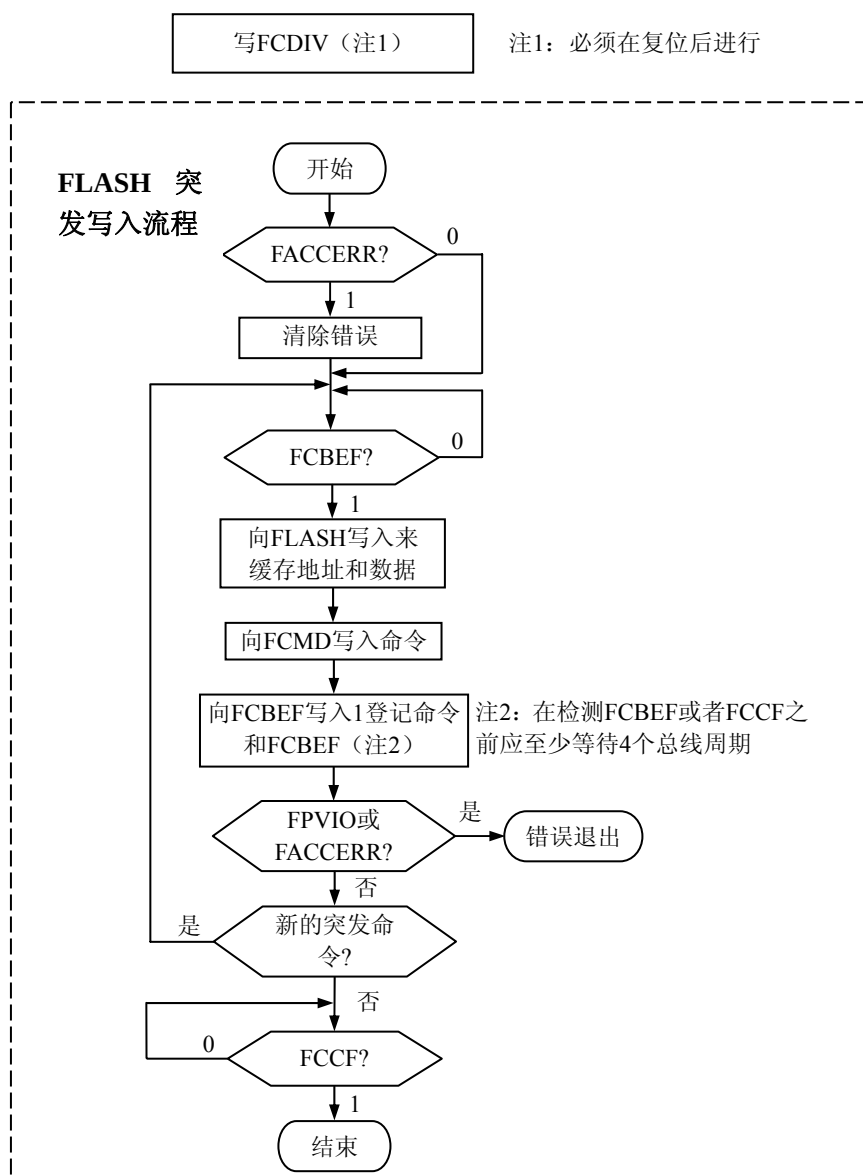


图4-3 FLASH突发写入流程图

在突发模式下，写入序列的第一个字节和在标准模式下写入一个字节花费相同的时间。如果满足上面的条件，剩下的字节也会在突发模式下写入。在下一个序列的地址是新行的

情况下，那个字节的写入时间是标准时间而不是突发模式时间。这是因为阵列的高电压必须先失效然后再使能。如果在当前的命令完成前，没有一个突发模式命令在等待，那么电荷泵将会失效，阵列上的高电压被移除。

4.4.5 访问错误

只要违反命令执行协议，错误就会产生。

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。一旦发生访问错误，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)。
- 写入 FLASH 地址后，设置除 FCMD 外的任意其他控制寄存器。
- 向 FCMD 写除了五个正确的命令(0x05、0x20、0x25、0x40,或 0x41)之外的任何命令代码。
- 向 FCMD 中写入命令字之后，设置除 FSTAT(清 FCBEF 和登记命令)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。

4.4.6 FLASH块保护

块保护防止对被保护范围的 FLASH 存储空间进行写入或擦除操作。块保护通过块保护寄存器控制 (FPORT) 控制。当使能时，块保护可以在任意 512 字节的边界开始，这个边界低于 FLASH 最后的地址 (0xFFFF)。

复位时，FPROT 装入 NVPROT 地址的内容。NVPROT 在 FLASH 存储器中的非易失性寄存器块中。FPROT 不能直接通过应用软件改变，所以一个跑飞的程序不能改变块保护设置。因为 NVPROT 在 FLASH 的最后 512 字节，只要有 FLASH 被保护，NVPROT 本身就会被保护而不能被应用软件改变 (有意或无意)。通过背景调试命令可以写 FPROT，这是擦除和重新写入一个被保护的 FLASH 存储器的方法。

图 4-4 描述了块保护机制。FPS 使用了未被保护的最后一个地址的高字节。就像图中表示的，地址由 FPS7:FPS1 连接上逻辑 1 组成。例如，保护 1536 个字节 (从 0xFA00 到 0xFFFF)，FPS 位设置为 1111100，也就是未被保护的最后一个地址 0xF9FF 的值。除了将 FPS 写为适当的值，FPDIS (NVPROT 的 0 位) 必须置 0 以允许块保护。因此 0xF8 必须写到 NVPROT 中保护从 0xFA00 到 0xFFFF 的地址。

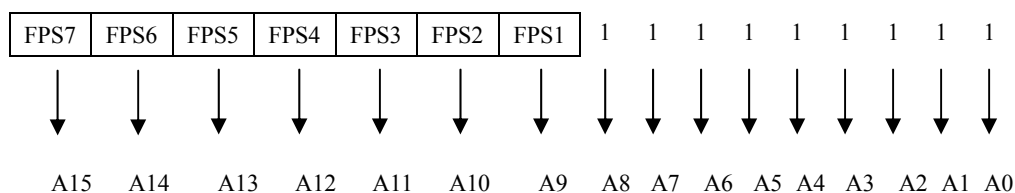


图4-4 块保护机制

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。

4.4.7 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(\$FFE0:\$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。这就允许用户将新的程序代码和新的向量值写入 FLASH 中未被保护的部分，而包含默认向量地址的被保护区域不被改写。

4.5 安全性

MC9S08AC16 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，但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。在写 FLASH 时，用户通过写 NVOPT 来使用 MCU 的安全机制。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 是提倡的操作，SEC01:SEC00=1:0。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：

- 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。

- 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。STHX 不在这些写入中使用，因为这些写入不能在一个邻近的总线周期完成。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。
- 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，在下次复位前都会禁止保密。

安全密钥只能从安全存储器（RAM 或 FLASH）写入。如果没有安全用户程序的协调不能由背景调试命令进入。

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：

- 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。
- 如果需要可以整体擦除 FLASH 存储器。
- 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。

4.6 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，3 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 3 个相应的高页区域的控制寄存器。同样也有一个 8 字节的密钥在 FLASH 存储器中。表 4-3 和表 4-4 列出了所有 FLASH 寄存器的绝对地址。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。

4.6.1 FLASH时钟分频寄存器(FCDIV)

该寄存器的第 7 位是一个只读状态标志位。第 0 位到第 6 位在任何时候都可以被读，但只能写一次。在可以任何擦除或写入操作之前，向该寄存器写入内容用以在有效的范围内为非易失性存储系统设置时钟的频率。

	第7位	6	5	4	3	2	1	第0位
读	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
写								
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 寄存器域描述

域	描述
---	----

7 DIVLD	时钟分频加载位——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。 0 FCDIV没有被写过，不可以对FLASH进行擦除和写入操作。 1 FCDIV已经被写过，可以对FLASH进行擦除和写入操作。
6 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[5:0]	时钟分频位——FLASH时钟分频器通过DIV5: DIV0这六位加上1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。参考等式4-1，等式4-2和表4-7。

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

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

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)
20 MHz	1	12	192.3kHz	5.2μs
10 MHz	0	49	200kHz	5μs
8 MHz	0	39	200kHz	5μs
4 MHz	0	19	200kHz	5μs
2 MHz	0	9	200kHz	5μs
1 MHz	0	4	200kHz	5μs
200KHz	0	0	200kHz	5μs
150KHz	0	0	150kHz	6.7μs

4.6.2 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。第 2 到 5 位为无效位，读时一直为 0。该寄存器在任何时刻可读，但是写入没有任何影响。为了改变这个寄存器的值，象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。

	第7位	6	5	4	3	2	1	第0位
读	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
写								
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）

表 4-8 FOPT 寄存器域描述

域	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4.5节安全性。 0 禁止FLASH安全密钥。 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。 0 向量重定向使能。 1 向量重定向禁止。
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4.5节安全性。

表 4-9 安全状态码

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

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。

4.6.3 FLASH配置寄存器(FCNFG)

第 5 位到第 7 位可以在任何时间被读写。第 0 位到第 4 位通常读为 0 且不能被写。

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

图4-7 FLASH配置寄存器（FCNFG）

表 4-10 FOPT 寄存器域描述

域	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。 0 写\$FFB0 - \$FFB7被认为是FLASH写入或擦写命令的开始。 1 写NVBACKKEY(\$FFB0 - \$FFB7)被认为是进行密码比较。

4.6.4 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，如果 FPDIS=0，保护可以加强，例如更小的 FPS 值可以被写入。如果 FPDIS=1，写入不能改变保护。

	第7位	6	5	4	3	2	1	第0位
读	FPS ⁽¹⁾							FPDIS ⁽¹⁾
写								
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 4-11 FPROT 寄存器域描述

域	描述
7:1 FPS[7:1]	FLASH保护区域选择——当FPDIS = 0，这7位决定在高地址位的FLASH末端的未被保护FLASH区域的末地址，受保护的FLASH区域不可被擦写。
0 FPDIS	FLASH保护禁止位 0 禁止FLASH块保护。 1 由FPS0[7:1]决定的FLASH区域被块保护(不允许写入和擦除)。

4.6.5 FLASH状态寄存器(FSTAT)

第 0，1，3 位一直读为 0 且对它的写入都是无效且没有意义的。剩下的五位都是状态

位，它们任何时刻都是可读的。向这些位写入是有特殊的意义的，这些意义在下面描述。

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

图4-9 FLASH状态寄存器（FSTAT）

表 4-12 FSTAT 寄存器域描述

域	描述
7 FCBEF	FLASH命令缓冲区空标志位——FCBEF位通常用来开始命令。在执行突发写入时，FCBEF位标记命令缓冲区为空，一个新的命令序列可以被执行。向FCBEF位写1或一个突发写入命令进入写入数组可以清FCBEF位。只有突发写入命令可以被缓冲。 0 命令缓冲区满（不能被另外的命令使用）。 1 新的突发写入命令被写到命令缓冲区。
6 FCCF	FLASH命令完成标志位——当命令缓冲区为空而且没有命令正被处理，FCCF自动置位。当一个新的命令开始（通过置1FCBEF来存入命令），FCCF自动清零。 0 命令处理中。 1 所有的命令都已经完成。
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。 0 表示不进行侵害保护设置。 1 表示试图对保护区域进行擦写操作。
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（错误的命令被忽略），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4.4.5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。 0 没有访问错误发生。 1 表示一个访问错误已经发生。
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。 0 一个空白检测命令完成之后且FCCF=1，FBLANK=0表示FLASH是非空。 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。

4.6.6 FLASH命令寄存器(FCMD)

在正常用户模式中，只有 5 个命令代码可以被识别，见表 4-14。FLASH 的写入，擦除和空白检测操作见 4.4.3 节写入和擦除命令执行。

	第7位	6	5	4	3	2	1	第0位
读	FCMP7	FCMP6	FCMP5	FCMP4	FCMP3	FCMP2	FCMP1	FCMP0
写								
复位	0	0	0	0	0	0	0	0

图4-10 FLASH命令寄存器（FCMD）

表 4-13 FCMD 寄存器域描述

域	描述
7:0 FCMD[7:0]	FLASH命令位——参见表4-14。

表 4-14 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（每页512字节）	\$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、中断和系统配置

5.1 简介

本章介绍了基本的复位和中断机制以及大部分 MC9S08AC16 系列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，它们有单独的章节但是系统控制逻辑的一部分。

5.2 特征

复位和中断特征包括：

- 多种复位源使系统配置灵活和操作可靠：
 - 上电复位(POR)
 - 使能低压检测 (LVD)
 - 外部 $\overline{\text{RESET}}$ 引脚
 - COP 看门狗使能和两个溢出时间选择
 - 非法操作码
 - 非法地址
 - 背景调试主机发的连续命令
- 复位状态寄存器(SRS)指出最近一次发生复位的原因
- 每个模块独立的中断向量(减少查询的开销)（参见表 5-11）

5.3 MCU复位

复位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，大部分控制和状态寄存器都设为复位默认值，从 0xFFFFE~0xFFFF 取出两字节的复位向量送到程序计数器 PC。片上外设模块不可用，I/O 引脚初始化设为通用的高阻抗输入引脚无上拉。条件码寄存器(CCR)的 I 位被置位来屏蔽中断，所以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 被强制设为\$00FF。

MC9S08AC16 系列包括下列复位源：

- 上电复位(POR)
- 低压检测 (LVD)
- 计算机正常操作监控模块 (COP) 定时器
- 非法操作码检测
- 非法地址检测
- 背景调试强制复位
- 复位引脚 ($\overline{\text{RESET}}$)
- 时钟发生器失锁和丢失时钟复位

除了背景调试强制复位，每个源对应系统复位状态寄存器相应位。

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执行时，COP 看门狗会强制系统复位。为了防止 COP 复位(当 COP 被使能)，应用软件必须周期性地复位 COP 定时器。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。

在任何一个复位之后，置位 SOPT 寄存器(更多信息见 5.9.4 系统选项寄存器(SOPT))的 COPE 位来使能 COP 看门狗。如果在应用中不使用看门狗，清 COPE 位来禁止看门狗。写任意值到 SRS 可以复位 COP 计数器。写入操作不影响只读 SRS 的数据。相反地，写入操作被解码并向 COP 计数器发送一个复位信号。

SOPT2 寄存器的 COPCLKS 位（更多信息见 5.9.10 系统选项寄存器 2(STOP2)）选择用于 COP 定时器的时钟源。时钟源选项是总线周期或内部的 1-kHz 时钟源。每个时钟源有相应的短和长溢出，由 SOPT 的 COPT 控制。表 5-1 总结了 COPCLKS 和 COPT 位的控制功能。COP 看门狗默认操作是总线时钟源和相应的长溢出（ 2^{18} 周期）。

表 5-1 COP 配置选项

控制位		时钟源	COP 溢出计时
COPCLKS	COPT		
0	0	~1kHz	2^5 周期 (32ms) ¹
0	1	~1kHz	2^8 周期 (256ms) ¹
1	0	总线	2^{13} 周期
1	1	总线	2^{18} 周期

即使应用使用 COPE、COPCLKS 和 COPT 的默认复位设置，在复位初始化时，用户仍应该向一次性写入的 SOPT 寄存器写入，来锁存这些设置，这样即使应用程序跑飞也不会被意外的改变。对 SOPT 和 SOPT2 将会设置 COP 计数器。

对 SRS 的写入(清零 COP 定时器)不能放到中断服务程序(ISR)里，因为即使主程序失败，ISR 可能仍会周期性地执行。

在背景调试模式下，COP 计数器不能增加。

选择总线时钟源后，当系统处于停止模式时 COP 计数器不能增加。MCU 一退出停止模式 COP 计数器就重新开始。

选择 1-kHz 源后，当系统进入停止模式时 COP 计数器重新初始化为 0。MCU 一退出停止模式 COP 计数器就开始计数。

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 引脚边缘触发或者定时器溢出等硬件事件引起的。在特定环境下调试模块也能产生 SWI 软中断。

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)时 CPU 才会对这个事件作出反应。CCR 的 I 位为 0 允许中断。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以后响应此中断。中断的执行过程就像 SWI 指令的一个周期一个周期的执行顺序一样，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，处于 ISR 时 I 位可能会被中断清零（清零后可以产生中断）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小程序错误。

中断服务程序由 RTI 指令结束，用先前保存在堆栈中的值来恢复 CCR、A、X 及 PC 寄存器为中断前的值。

注：

为了能够与 HCS08 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 位清零导致两个或更多中断挂起时，高优先级的中断首先响应（参见表 5-2）。

5.5.1 中断堆栈结构

图 5-1 显示了堆栈结构的内容和组织形式。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，这个地址是 CCR 被保存的地址的低一个地址。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。

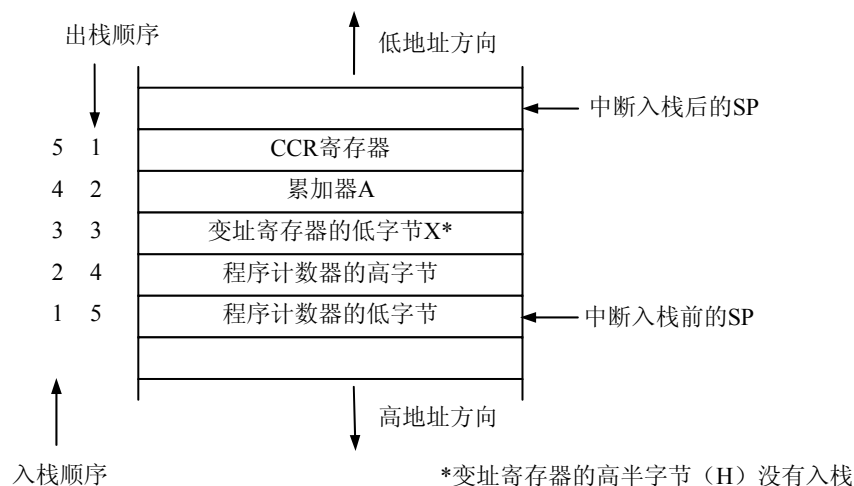


图5-1 中断堆栈框架

当一条 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(IRQ)引脚

外部中断被 IRQSC 状态和控制寄存器管理。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 引脚使能时默认使用内部的拉电设备（IRQPDD=0），是上拉还是下拉由选择的极性决定。如果用户需要使用内部上拉或下拉，向 IRQPDD 写 1 关闭内部设备。

BIH 和 BIL 指令可能在引脚设为 IRQ 输入时被用做侦测 IRQ 引脚的电平。

注：

引脚到 V_{DD} 不包含箝位二极管而且低于 V_{DD} 不能被驱动。上拉 IRQ 引脚标准电压不低于 $V_{DD} - 0.7V$ 。所有的其他带有上拉电阻的引脚将有 V_{DD} 的无负荷度量。

注：

当使能 IRQ 使用时，IRQF 被置位，而且应该在使能中断之前清零。在一个 5V 的系统中，当配置引脚为下降沿而且电平触发，有必要在清标志位和使能中断之间等待 6 个周期。

5.5.2.2 边沿/电平触发

IRQMOD 控制位重新配置检测逻辑，所以他检测时间和引脚电平。在边沿和电平检测模式下，当检测到一个边沿（当 IRQ 引脚从非指定的变成指定电平时）IRQF 状态标志位被置位，但只要 IRQ 引脚保持在指定电平，这个标志位持续置位。

5.5.3 中断向量、中断源和局部屏蔽

表 5-2 给予出了所有中断源的摘要。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更高地址。

当中断条件发生，相关的标志位被置位。如果该相关的中断使能位为 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 中断概述

中断优先级	中断号	地址(高/低)	向量名称	模块	源	使能位	描述
从	29-31	0xFFC0/FFC1-0xFFC4/FFC5	不可用的向量空间(对用户程序不可用)				
	28	0xFFC6/FFC7	Vtpm3ovf	TPM3	TOF	TOIE	TPM3溢出
	27	0xFFC9/FFC9	Vtpm3ch1	TPM3	CH1F	CH1IE	TPM3通道1
	26	0xFFCA/FFCB	Vtpm3ch0	TPM3	CH0F	CH0IE	TPM3通道0
	25	0xFFCC/FFCD	Vrti	系统控制	RTIF	RTIE	实时中断中断
	24	0xFFCE/FFCF	Viic	IIC	IICIS	IICIE	IIC1

低 到 高	23	0xFFD0/FFD1	Vatd	ATD	COCO	AIEN	ADC1
	22	0xFFD2/FFD3	Vkeyboard	KBI	KBF	KBIE	键盘引脚
	21	0xFFD4/FFD5	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2发送
	20	0xFFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2接收
	19	0xFFD8/FFD9	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2错误
	18	0xFFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1发送
	17	0xFFDC/FFDD	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1接收
	16	0xFFDE/FFDF	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1错误
	15	0xFFE0/FFE1	Vspi	SPI	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI
	14	0xFFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2溢出
	13	0xFFE4/FFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2通道1
	12	0xFFE6/FFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2通道0
	11	0xFFE8/FFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1溢出
	10	0xFFEA/FFEB	不可用的向量空间				
	9	0xFFEC/FFED	不可用的向量空间				
	8	0xFFEE/FFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1通道3
	7	0xFFFF/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1通道2
	6	0xFFFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1通道1
	5	0xFFFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1通道0
	4	0xFFFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG
	3	0xFFFF8/FFF9	Vlvd	系统控制	LVDF	LVDIE	低电压检测
	2	0xFFFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ引脚
	1	0xFFFFC/FFFD	Vswi	核	SWI Instruction	—	软件中断
	0	0xFFFFE/FFFF	Vreset	系统控制	COP LVD RESET pin Illegal opcode	COPE LVDRE — —	看门狗定时器 低电压检测 外部引脚 非法操作码

5.6 低电压检测系统(LVD)

MC9S08AC16 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 SPMSC2 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 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 VPOR 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 V_{LVDV} 设定的电压。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 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。不推荐设置 LVW 触发电压等于 LVD 触发电压。LVW 的典型使用是选择 V_{LVWH} 和 V_{LVWL} 。

5.7 实时中断(RTI)

实时中断功能经常用来产生定期中断。RTI 接受两个时钟源，1-kHz 内部时钟源或外部时钟。1-kHz 内部时钟源完全独立于任何总线时钟源，只被用于 RTI 模块，在某些 MCU 中也可用于 COP 看门狗。使用外部时钟源，它必须可用而且可行。SRTISC 的 RTICLKS 位是用来选择 RTI 时钟源。

MCU 在运行、等待或 stop3 模式下，其中一个 RTI 时钟源可以使用。当在 stop3 下使用外部晶振，必须在 stop 下使能 (OSCSTEN=1) 而且配置为低带宽操作 (RANGE=0)。从 stop2 模式下唤醒 MCU 只能选择 1-kHz 内部时钟源。

寄存器 SRTISC 包括一个只读状态标志位，一个只写确认位，和一个 3 位的控制值(RTIS2:RTIS1:RTIS0)，该 3 位用于禁止实时中断的时钟源或者选择 7 个唤醒延迟时间中的一个。RTI 的本地中断使能 RTIE 可用于屏蔽实时中断。写 RTIS 的每位为 0 禁止 RTI，将没有中断产生。5.9.7 节系统实时中断状态和控制寄存器(SRTISC)，给出了各寄存器的细节信息。

5.8 MCLK输出

PTC2 和 MCLK 时钟输出共享引脚。置位引脚使能位 MPE，PTC2 引脚输出内部 MCU 总线时钟的除数。除数由 MCSEL 位决定。当 MPE 置位，PTC2 引脚强制为输出引脚除非这个引脚用于端口数据控制位的状态。如果 MCSEL 位全 0，引脚为低。引脚的转换速度和驱动强度分别由 PTCSE2 和 PTCDS2 控制。如果转换速度控制使能，最大时钟输出频率被限制，转换速度使能的引脚上升和下降的时间参见电气章节。

5.9 复位、中断以及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 4 章有关片上存储器直接页寄存器的介绍。本节通过名称来引用各寄存器和控制位，Freescale 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。

SOPT 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，而相关函数在第三章“操作模式”里有更加详细的论述。

5.9.1 中断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。

	第7位	6	5	4	3	2	1	第0位
读	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
写						IRQACK		
复位	0	0	0	0	0	0	0	0

图5-2 中断请求状态和控制寄存器（IRQSC）

表 5-3 IRQSC 寄存器域描述

域	描述
6 IRQPDD	中断请求拉电设备禁止位——当IRQ引脚被使能允许使用外部设备，这个可读写控制位用来禁止内部上拉或下拉设备。 0 如果IRQPE=1，IRQ拉电设备允许。 1 如果IRQPE=0，IRQ拉电设备禁止。
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。IRQMOD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，引脚选择下拉电阻。当IRQ引脚作为IRQ输入引脚使能，并且设置为侦测引脚下降沿时，引脚选择上拉电阻。 0 下降沿或者下降沿/低电压触发。 1 上升沿或者上升沿/高电压触发。
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。 0 IRQ引脚功能不可用。 1 IRQ引脚功能可用。
3 IRQF	IRQ标志位——当一个中断请求事件发生时该只读位被置位。 0 没有IRQ请求。 1 侦测到IRQ事件。
2 IRQACK	IRQ确认——这个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(IRQMOD=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事件是否产生中断请求。 0 从IRQF发来（使用轮询）的中断请求不可用。 1 IRQF=1时产生中断请求。
0 IRQMOD	IRQ监控模式——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。 0 仅在下降沿或者上升沿时IRQ事件触发。 1 在下降沿和低电压，或者上升沿和高电压时IRQ事件触发。

5.9.2 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 SBDFR 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。

	第7位	6	5	4	3	2	1	第0位
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	U	0	0	0	0	0	1	0
其他复位	0	(1)	(1)	(1)	0	(1)	0	0

U=不被复位影响

(1) 在复位时任何有效的复位源都将导致相应的位被置位。同时相应的无效的位被清零。

图5-3 系统复位状态（SRS）

表 5-4 SRS 寄存器域描述

域	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。 0 复位不是由POR引起。 1 复位由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=1设为不可用，则BGND指令被认为是非法操作。 0 复位不是由非法操作码引起。 1 复位由非法操作码引起。
3 ILAD	非法操作码复位——由访问未指定的非法地址而引起的复位。 0 复位不是有访问非法地址引起。 1 复位是有访问非法地址引起。 MC9S08AC16的非法地址区域： 0x0470-0x17FF——从RAM的最后到高页寄存器的开始的区域。 0x1860-0xBFFF——从高页寄存器的开始到Flash的开始的区域。 寄存器区域中未使用和保留的地址不被认为是非法地址也不会触发非法地址中断。
2 ICG	内部时钟发生模块复位——复位由ICG模块引起。 0 复位不是由ICG引起。 1 复位由ICG引起。
1 LVD	低电压触发复位——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。 0 复位不是由LVD或POR而引起的。 1 复位由LVD或者POR引起。

5.9.3 系统背景调试强制复位寄存器(SBDFR)

这个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 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 寄存器域描述

域	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。

5.9.4 系统选项寄存器(SOPT)

这个寄存器可能在任何时间被读。第 3, 2 位不被用且始终为 0。这是个一次可写入寄存器所以只在复位后的第一次写入有效。任何后来的尝试向 SOPT 写入都是被忽视的, 用以避免敏感设置的意外变化。SOPT 应该在用户复位初始化程序期间被写入用以实现期望的控制, 即使期望的设置与复位后的设置一样。

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

图5-5 系统选项寄存器(SOPT)

表 5-6 SOPT 寄存器域描述

域	描述
7 COPE	COP看门狗使能——这个一次写入位在复位后的默认值为1 0 COP看门狗定时器不可用。 1 COP看门狗定时器可用(时间溢出后复位)。
6 COPT	COP看门狗溢出时间——这个一次写入位在复位后的默认值为1 0 选择较短溢出时间。 1 选择较长溢出时间。
5 STOPE	Stop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。 0 Stop停止模式不可用。 1 Stop停止模式可用。

5.9.5 系统MCLK控制寄存器(SMCLK)

这个寄存器用于控制 MCLK 时钟输出。

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

图5-6 系统MCLK控制寄存器(SMCLK)

表 5-7 SMCLK 寄存器域描述

域	描述
4 MPE	MCLK引脚使能——这个位用于使能MCLK功能。 0 MCLK输出禁止。 1 PTC2引脚上MCLK输出允许。
2:0 MCSEL	MCLK分频选择——这些位用来选择

$$\text{MCLK 频率} = \text{总线频率} \div (2 * \text{MCSEL}) \quad (\text{等式 5-1})$$

5.9.6 系统设备识别寄存器(SDIDH、SDIDL)

这个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 的具体存储块，寄存器和目标 MCU 中的控制位。

	第7位	6	5	4	3	2	1	第0位
读					ID11	ID10	ID9	ID8
写								
复位	-	-	-	-	0	0	0	0

图5-7 系统标识寄存器-高(SDIDH)

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

图5-8系统标识寄存器-低(SDIDL)

表 5-8 SDIDH 寄存器域描述

域	描述
7:4 保留	位7：4被保留。读这些位将会得到不确定值；写没有影响。
3:0 ID[11:8]	识别编号——每一个HCS08家族的派生版本都有一个唯一识别编号。MC9S08AC16系列编码位0x012。参见表5-9的ID位。

表 5-9 SDIDL 寄存器域描述

域	描述
7:0 ID[7:0]	识别编号——每一个HCS08家族的派生版本都有一个唯一识别编号。MC9S08AC16系列编码位0x012。参见表5-9的ID位。

5.9.7 系统实时中断状态和控制寄存器(SRTISC)

该寄存器包含一个只读状态标志位，一个只写确认位，三个可读写延时选择，和三个总是读为0的无效位。

	第7位	6	5	4	3	2	1	第0位
读	RTIF	0	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
写		RTIACK						
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 寄存器域描述

域	描述
7 RTIF	实时中断标志位——这个只读状态位显示周期性唤醒定时器已经溢出。 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	实时中断确认位——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 RTICLK5	实时中断时钟选择——该读写位选择实时中断的时钟源。 0 实时中断请求时钟源为内部1KHz振荡器。 1 实时中断请求时钟源为外部时钟。
4 RTIE	实时中断使能——这个可读写位使能实时中断 0 实时中断不可用。 1 实时中断可用。
2:0 RTIS[2:0]	实时中断延时选择——这些可读写位选择RTI的唤醒延时。实时中断的时钟源是一个频率为1KHz的自时钟源，它独立于MCU的其他时钟源。 使用外部时钟源，延时为通过RTIS2:RTIS1:RTIS0分频后的晶振频率。

表 5-11 实时中断频率

RTIS2:RTIS1:RTIS0	1=kHz 的时钟源延时 ¹	使用外部时钟延时(晶振频率)
0:0:0	不可用的周期性唤醒定时器	不可用的周期性唤醒定时器
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. 列中的值以 $f_{RTI}=1\text{kHz}$ 为基准，关于这些值的规定参附录 A

5.9.8 系统电源管理状态控制寄存器1(SPMSC1)

	第7位	6	5	4	3	2	1(1)	第0位
读	LVDF	0	LVDIE	LVDRE(2)	LVDSE(2)	LVDE(2)		
写		LVDACK						
复位	0	0	0	1	1	1	0	0

1 位1是保留，必须总是写0。

2 这个位复位后只能写一次。其他的写入被忽略。

图5-10 系统电源管理状态控制寄存器1(SPMSC1)

表 5-12 SRTISC 寄存器域描述

域	描述
7 LVDF	低电压检测标志位——如果LVDE=1，这个只读状态位表示一个低电压触发事件。
6 LVDACK	低电压检测确认位——这个只写位用于确认低电压监控错误(写1可清LVDF位)。读它总是返回逻辑0。

5 LVDIE	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。 0 硬件中断请求不可用。 1 当LVDF=1时请求一个硬件中断。
4 LVDRE	低电压检测复位使能——该读写位使能LVDF错误来产生一个硬件复位(当LVDE=1时)。 0 LVDF不会产生一个硬件复位 1 当LVDF=1时强制MCU复位
3 LVDSE	低电压检测停止使能——当LVDE=1时，这个可读写位决定在MCU停机模式时低电压监控功能是否可用。0 停止模式下禁止低电压监控。 1 停止模式下允许低电压监控。
2 LVDE	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。 0 LVD逻辑不可用。 1 LVD逻辑可用。
0 BGBE	能带隙缓存使能位——BGBE位用来使能用于能带隙电压的内部缓存，参考ADC模块一个内部通道的使用。 0 能带隙缓存禁止。 1 能带隙缓存使能。

5.9.9 系统电源管理状态和控制寄存器2(SPMSC2)

该寄存器被用于报告低压警告功能的状态，和设置 MCU 停机模式行为。

	第7位	6	5	4	3	2	1	第0位
读	LVWF	0	LVDV	LVWV	PPDF	0		PPDC ⁽¹⁾
写		LVWACK				PPDACK		
上电复位	0 ⁽²⁾	0	0	0	0	0	0	0
LVD复位	0 ⁽²⁾	0	U	U	0	0	0	0
其他复位	0 ⁽²⁾	0	U	U	0	0	0	0

U=复位时不受影响。

1 这个位复位后只能写一次。其他的写入被忽略。

2 在复位和 V_{Supply} 已经低于 V_{LVW} 或者 V_{Supply} 在临界以下时，LVWF会被置位

图5-11 系统电源管理状态和控制寄存器2(SPMSC2)

表 5-13 SPMSC2 寄存器域描述

域	描述
7 LVWF	低电压警告标志位——该位显示低电压警告状态。 0 低电压警告没有发生。 1 低电压警告已经发生或者正在发生。
6 LVWACK	低电压警告确认位——该位表示低电压警告确认，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)。 0 选择较低临界电压($V_{LVW}=V_{LVWL}$)。 1 选择较高临界电压($V_{LVW}=V_{LVWH}$)。
4 LVWV	低电压警告电压选择——该位选择LVW临界电压(VLVW)。 0 选择较高临界电压($V_{LVW}=V_{LVWL}$)。 1 选择较高临界电压($V_{LVW}=V_{LVWH}$)。
3 PPDF	局部低功耗标志位——该位显示MCU已退出stop2模式。 0 没有从stop2模式恢复。 1 从stop2模式恢复。
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，stop2还是stop3。 0 stop3模式使能。 1 stop2模式局部低功耗使能。
-----------	--

5.9.10 系统系统选项寄存器2(STOP2)

高页寄存器包括设置 MC9S08AC16 系列设备的具体特征的位。

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

(1) 这个位复位后只能写一次。其他的写入被忽略。

图5-12 系统系统选项寄存器2(STOP2)

表 5-14 SOPT2 寄存器域描述

域	描述
7 COPCLKS	COP看门狗时钟选择位——这个一次性写入位选择COP看门狗时钟源。 0 内部的1-kHz。 1 总线时钟源。

第六章 并行输入和输出

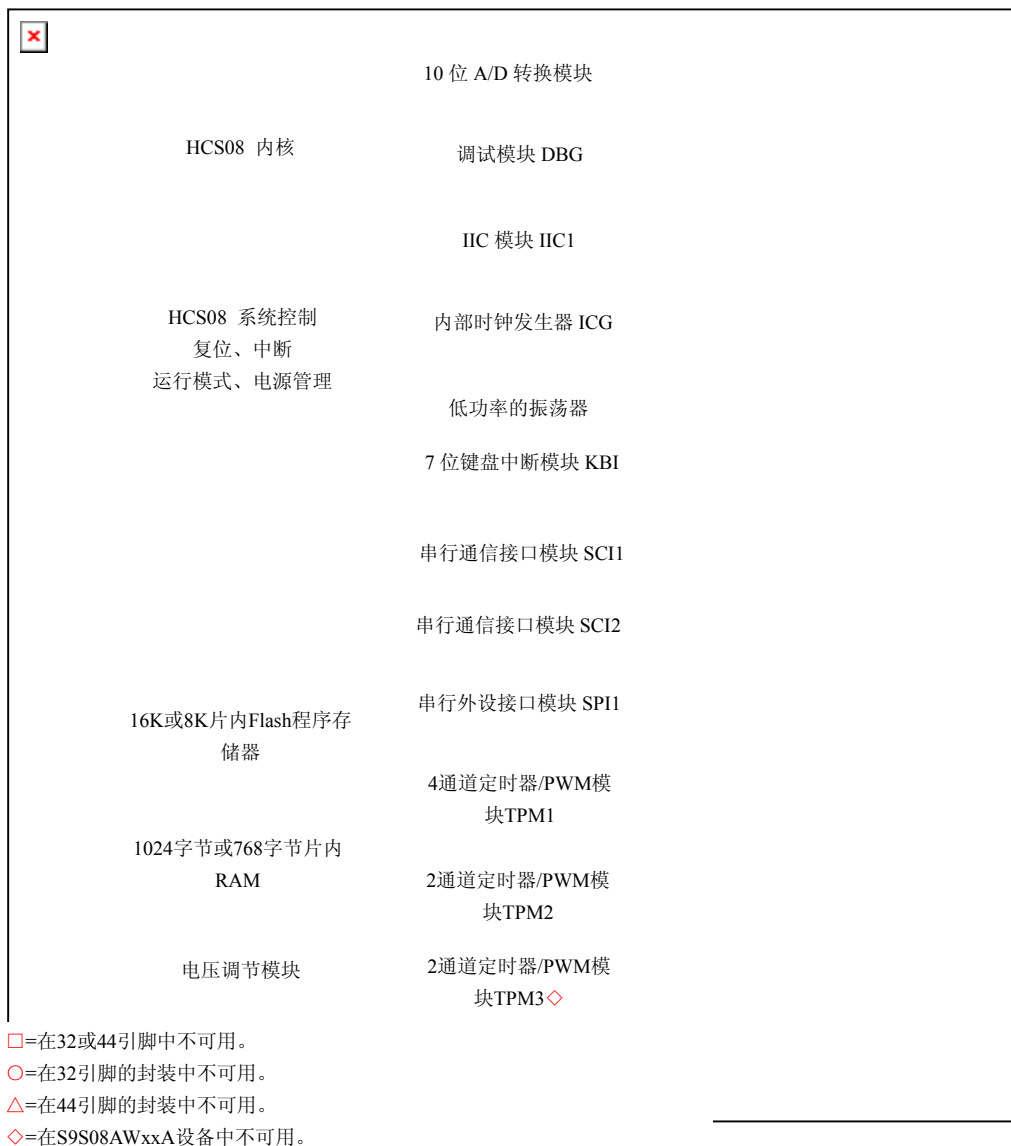
6.1 简介

本章解释了软件控制相关的并行输入输出（I/O）。MC9S08AC16 有 7 个并行 I/O 端口，总共 54 个通用 I/O 引脚。关于这些引脚的逻辑和硬件方面的更多信息参见第二章引脚和连接。

这些引脚中有许多于片上设备复用，比如定时器系统，通信系统，或键盘中断。当其他模块不控制这些引脚时。他们用于通用 I/O 控制。

注：

并不是所有的通用 I/O 引脚在所有的封装上都可用。为了避免输入引脚上额外的电流消耗，用户的应用程序的复位初始化程序应使能片上上拉电阻或改变为连接引脚的方向位输出以使引脚不浮动。



注意：

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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测（IRQEDG=1），下拉使能。
3. IRQ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
4. 引脚包含集成的上拉设备。
5. PTD3,PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能（KBIPEn=1）而且上升沿被选择（KBEDGn=1），下拉使能。

图6-1 框图高亮显示并行I/O

6.2 特征

并行 I/O 和引脚控制特征取决于封装的选择，包括：

- 7 个端口共 38 个通用 I/O 引脚。
- 输入缓存区滞后
- 每个引脚上的可软件控制的上拉电阻
- 4 个 A 口引脚
- 4 个 B 口引脚与 ADC1 和 TPM3（1）复用

- 6 个 C 口引脚与 SCI2, IIC1 和 MCLK 复用
- 4 个 D 口引脚与 ADC1, KBI, TPM1 和 TPM2 的外部时钟复用
- 8 个 E 口引脚与 SCI1, TPM1 和 SPI1 复用。
- 5 个 F 口引脚与 TPM1 和 TPM2 复用。
- 7 个 G 口引脚与 XTAL, EXTAL 和 KBI 复用。

6.3 引脚描述

MC9S08AC16 系列总共有 7 个端口（PTA-PTG）共 38 个并行 I/O 引脚。并不是所有的引脚都在封装上引出来。当其他模块不控制这些引脚时。他们用于通用 I/O 控制。

复位后，复用设备功能被禁止，引脚用于并行 I/O。所有的并行 I/O 被设置为输入（PTxDDn=0）。每个引脚的控制功能被设置如下：转换速率控制使能（PTxSEn=1），选择低压驱动强度（PTxDSn=0）和内部上拉电阻使能（PTxPEN=0）。

接下来讨论每个端口和软件控制每个引脚的使用。

6.3.1 A 口

	第7位	6	5	4	3	2	1	第0位
MCU 引脚	PTA7	R	R	R	R	PTA2	PTA1	PTA0

图6-2 A口引脚名称

A 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器（PTAD）和 A 口数据方向寄存器（PTADD）控制。引脚控制寄存器上拉电阻使能（PTAPE），转换速率控制（PTASE）和驱动强度选择（PTADS）位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节并行 I/O 控制，更多关于引脚控制的信息参见 6.5 节引脚控制。

6.3.2 B 口

	第7位	6	5	4	3	2	1	第0位
MCU 引脚	R	R	R	R	PTB3/ PM3CH0/ AD1P3	PTB2/ PM3CH1/ AD1P2	PTB1/ AD1P1	PTB0/ AD1P0

图6-3 B口引脚名称

B 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器（PTBD）和 B 口数据方向寄存器（PTBDD）控制。引脚控制寄存器上拉电阻使能（PTBPE），转换速率控制（PTBSE）和驱动强度选择（PTBDS）位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节“并行 I/O 控制”，更多关于引脚控制的信息参见 6.5 节“引脚控制”。

B 口通用 I/O 与 ADC 和 TPM3 定时器通道复用。任何引脚使能作为 ADC 输入时将禁止通用 I/O 功能。当任何 TPM3 功能使能时，TPM3 控制引脚的方向（输入或输出）而不是并行 I/O 口的数据方向寄存器。更多关于把 B 口引脚用作 TPM 通道的信息参见第十章“定时器/PWM(S08TPMV3)”，更多关于 B 口用作模拟输入的信息参见第十四章“AD 转换器(S08ADC10V1)”。

6.3.3 C 口

	第7位	6	5	4	3	2	1	第0位
--	-----	---	---	---	---	---	---	-----

MCU 引脚	0	R	PTC5/ RxD2	PTC4	PTC3/ TxD2	PTC2/ MCLK	PTC1/ SDA1	PTC0/ SCL1
-----------	---	---	---------------	------	---------------	---------------	---------------	---------------

图6-4 C口引脚名称

C 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器（PTCD）和 C 口数据方向寄存器（PTCDD）控制。引脚控制寄存器上拉电阻使能（PTCPE），转换速率控制（PTCSE）和驱动强度选择（PTCDS）位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节并行 I/O 控制，更多关于引脚控制的信息参见 6.5 节引脚控制。

C 口通用 I/O 与 SCI2、IIC 和 MCLK 复用。当任何复用功能使能时，复用功能控制引脚方向（输入或输出）而不是通用 I/O 口的数据方向寄存器。同样，对于复用功能控制为输出的引脚，它的输出数据由复用功能控制而不是端口的数据寄存器。

关于把 C 口引脚用作 SCI 引脚的更多信息参见第十一章“串行通信接口（S08SCIV4）”。

关于把 C 口引脚用作 IIC 引脚的更多信息参见第十三章“内部集成电路（S08IICV2）”。

关于把 PTC2 引脚用作 MCLK 引脚的更多信息参见第五章复位，中断和系统配置。

6.3.4 D 口

	第7位	6	5	4	3	2	1	第0位
MCU 引脚	R	R	R	R	PTD3/ AD1P10/ KBIP6	PTD2/ AD1P10/ KBIP5	PTD1/ AD1P9	PTD0/ AD1P8

图6-5 D口引脚名称

D 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器（PTDD）和 D 口数据方向寄存器（PTDDD）控制。引脚控制寄存器上拉电阻使能（PTDPE），转换速率控制（PTDSE）和驱动强度选择（PTDDS）位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节“并行 I/O 控制”，更多关于引脚控制的信息参见 6.5 节“引脚控制”。

D 口通用 I/O 与 ADC 和 KBI 复用。当任何复用功能使能时，复用功能控制引脚方向（输入或输出）而不是通用 I/O 口的数据方向寄存器。当 ADC 和数字设备功能同时共享时，ADC 有较高优先级。例如 ADC 和 KBI 同时使用 PTD7 的情况，引脚由 ADC 模块控制。

关于把 D 口引脚用作 TPM 引脚的更多信息参见第十章“定时器/PWM(S08TMPV3)”。

关于把 D 口引脚用作模拟输入的更多信息参见第十四章“AD 转换器(S08ADC10V1)”。

关于把 D 口引脚用作键盘引脚的更多信息参见第九章“键盘中断（S08KBIV1）”。

6.3.5 E 口

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

图6-6 E口引脚名称

E 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器（PTED）和 E 口数据方向寄存器（PTEDD）控制。引脚控制寄存器上拉电阻使能（PTEPE），转换速率控制（PTESE）和驱动强度选择（PTEDS）位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节并行 I/O 控制，更多关于引脚控制的信息参见 6.5 节引脚控制。

E 口通用 I/O 与 SCI1、SPI 和 TPM1 定时器通道复用。当任何复用功能使能时，复用功能控制引脚方向（输入或输出）而不是通用 I/O 口的数据方向寄存器。同样，对于复用功能控制为输出的引脚，它的输出数据由复用功能控制而不是端口的数据寄存器。

关于把 E 口引脚用作 SCI 引脚的更多信息参见第十一章“串行通信接口 (S08SCIV4)”。

关于把 E 口引脚用作 SPI 引脚的更多信息参见第十二章“串行外设接口 (S08SPIV3)”。

关于把 E 口引脚用作 TPM 通道引脚的更多信息参见第十章“定时器/PWM(S08TMPV3)”。

6.3.6 F 口

	第7位	6	5	4	3	2	1	第0位
MCU 引脚	R	PTF6	PTF5/ TPM2CH1	PTF4/ TPM2CH0	R	R	PTF1/ TPM1CH3	PTF0/ TPM1CH2

图6-7 F口引脚名称

F 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器 (PTFD) 和 F 口数据方向寄存器 (PTFDD) 控制。引脚控制寄存器上拉电阻使能 (PTFPE)，转换速率控制 (PTFSE) 和驱动强度选择 (PTFDS) 位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节“并行 I/O 控制”，更多关于引脚控制的信息参见 6.5 节“引脚控制”。

F 口通用 I/O 与 TPM1 和 TPM2 定时器通道复用。当任何复用功能使能时，复用功能控制引脚方向（输入或输出）而不是通用 I/O 口的数据方向寄存器。同样，对于复用功能控制为输出的引脚，它的输出数据由复用功能控制而不是端口的数据寄存器。

关于把 F 口引脚用作 TPM 通道引脚的更多信息参见第十章“定时器/PWM(S08TMPV3)”。

6.3.7 G 口

	第7位	6	5	4	3	2	1	第0位
MCU 引脚	0	PTG6/ EXTAL	PTG5/ XTAL	PTG4/ KBIP4	PTG3/ KBIP3	PTG2/ KBIP2	PTG1/ KBIP1	PTG0/ KBIP0

图6-8 G口引脚名称

G 口引脚是通用 I/O 引脚。并行 I/O 功能由位于零页寄存器空间的 A 口数据寄存器 (PTGD) 和 G 口数据方向寄存器 (PTGDD) 控制。引脚控制寄存器上拉电阻使能 (PTGPE)，转换速率控制 (PTGSE) 和驱动强度选择 (PTGDS) 位于高页寄存器。更多关于通用 I/O 控制的信息参见 6.4 节“并行 I/O 控制”，更多关于引脚控制的信息参见 6.5 节“引脚控制”。

G 口通用 I/O 与 KBI、XTAL 和 EXTAL 复用。当引脚为 KBI 输入时，引脚作为输入而不是相应 PTG 数据方向寄存器位的状态。当外部晶振使能时，PTG5 和 PTG6 作为晶振引脚。在这种情况下相应的并行 I/O 和引脚控制寄存器不控制这些引脚。

关于 G 口用脚用作 XTAL 和 EXTAL 引脚的更多信息参见第八章“内部时钟发生器 (S08ICGV4)”。

关于 G 口用脚用作键盘输入的更多信息参见第九章“键盘中断(S08KBIV1)”。

6.4 并行I/O控制

读写并行 I/O 通过端口数据寄存器完成。方向（输入或输出）通过端口数据方向寄存器控制。单个引脚的并行 I/O 功能在图 6-9 并行 I/O 框图中说明。

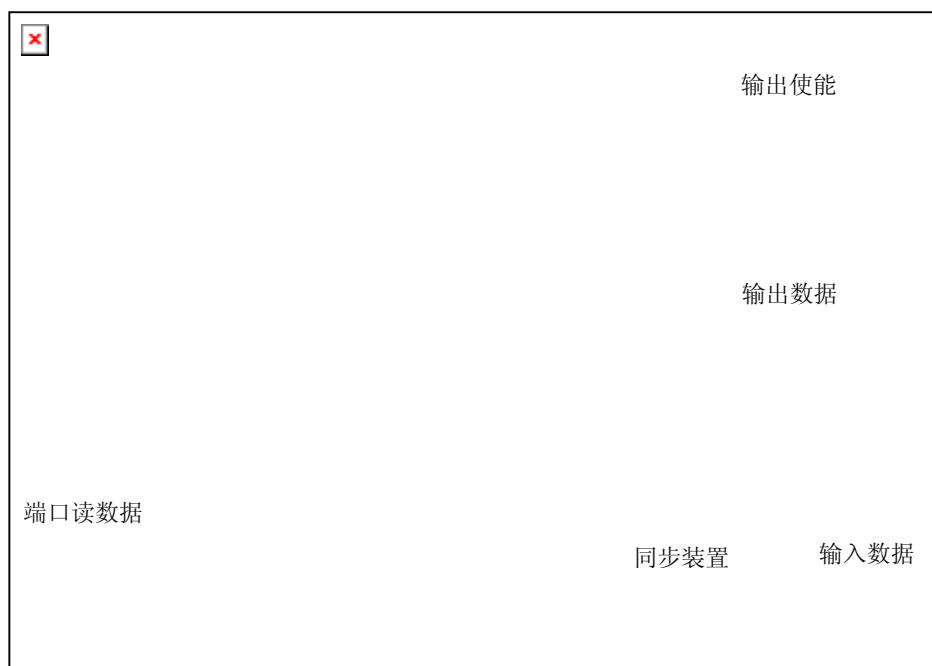


图6-9 并行I/O框图

数据方向控制位决定引脚输出驱动是否使能，而且他们控制从端口数据寄存器所读的内容。每一个端口引脚有一个数据方向寄存器位。当 $PTxDDn=1$ 时，相应的引脚是输出，读 $PTxD$ 返回最后写道端口数据寄存器的值。当设备模块或系统功能控制端口引脚时，即使设备系统重写了实际位方向的控制，数据方向寄存器位读端口数据寄存器返回的内容。

当模拟输入功能与引脚复用时，所有数字引脚功能被禁止。对于复用模拟输入功能的任何位，读端口数据寄存器返回值为 0。总的来说，数字输入功能和模拟输入功能共享同一引脚时，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，这是好的编程方法。这样确保了旧的数值不会即刻出现在端口数据寄存器。

6.5 引脚控制

引脚控制寄存器位于存储器的高页寄存器。这些寄存器控制 I/O 引脚的上拉电阻，转换速率和驱动强度。引脚控制寄存器独立于并行 I/O 寄存器进行操作。

6.5.1 内部上拉电阻使能

通过设置上拉电阻使能寄存器（ $PTxPEN$ ）的相应位可以使能引脚内部的上拉电阻。除非用于相应上拉电阻使能寄存器位的状态，当并行 I/O 控制逻辑或其他共享设备功能设置引脚位输出时，上拉电阻被禁止。

6.5.2 输出转换速率控制使能

通过设置转换速率控制寄存器（ $PTxSEN$ ）的相应位可以使能引脚的转换速率控制。当允许时，转换控制限制输出传输的速率以减少 EMC 发射。转换速率控制对设置为输入的引脚没有作用。

6.5.3 输出驱动强度选择

通过设置驱动强度寄存器（PTxDSn）的相应位，输出引脚可以选择高输出设备强度。当选择高驱动时，引脚能够产生和接收更大的电流。即使引脚被设置为高驱动，用户也必须确保不要超过芯片的总产生和接收限制电流。驱动强度选择会影响 I/O 引脚的直流电。然而交流电也被影响。高驱动允许引脚用同驱动小负载的低驱动引脚一样的转换速度来驱动更高的负载。因此引脚高驱动使能时，EMC 释放会受影响。

6.6 停止模式下的引脚

不同的停止模式下，执行 STOP 指令的结果不同后，I/O 功能不同。不同停止模式下的 I/O:

1 stop2 模式是局部掉电模式，在 STOP 指令执行以前 I/O 保持它的状态。在 STOP 指令执行 MCU 进入 stop2 模式前，CPU 寄存器状态和 I/O 寄存器状态保存在 RAM 中。从 stop2 模式返回时，访问任何 I/O 前，用户检测 SPMSC2 寄存器的 PPDF 位的状态。如果 PPDF 为 0，I/O 必须像上电复位一样重新初始化。如果 PPDF 为 1，I/O 数据先前存入 RAM，在执行 STOP 指令以前，外围设备必须重新初始化，重新载入先前的数据。用户必须置 1SPMSC2 寄存器的 PPDACK 位。然后用户应用程序就可以访问 I/O 了。

2 在 stop3 模式下，因为内部逻辑电路维持上电，所有 I/O 被保持。返回后，允许用户正常的 I/O 功能。

6.7 并行I/O与引脚控制寄存器

这一节介绍了与并行 I/O 口和引脚控制功能相关的寄存器。并行 I/O 寄存器位于存储器的零页，引脚控制寄存器位于高页寄存器部分。

6.7.1 A口I/O寄存器（PTAD和PTADD）

A 口并行 I/O 功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	PTAD7	R	R	R	R	PTAD2	PTAD1	PTAD0
写								
复位	0	0	0	0	0	0	0	0

图6-10 A口数据寄存器（PTAD）(1)

(1)3-6位是保留位，必须被写为0。

表 6-1 PTAD 寄存器域描述

域	描述
7, 2: 0 PTADn	A口数据寄存器位——A口引脚位输入时，读操作引脚的逻辑电平。A口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。A口引脚为输出时，逻辑电平有相应的MCU引脚驱动。复位强制PTAD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	PTADD7	R	R	R	R	PTADD2	PTADD1	PTADD0

写								
复位	0	0	0	0	0	0	0	0

图6-11 A口数据方向寄存器（PTADD）(1)

(1)3-6位是保留位，必须被写为0。

表 6-2 PTADD 寄存器域描述

域	描述
7, 2: 0 PTADDn	A口数据方向寄存器位——这些可读写位控制A口引脚的方向和PTAD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 A口n位输出使能而且PTAD读返回PTADn的内容。

6.7.2 A口引脚控制寄存器（PTAPE、PTASE、PTADS）

除了 I/O 控制，A 口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	PTAPE7	R	R	R	R	PTAPE2	PTAPE1	PTAPE0
写								
复位	0	0	0	0	0	0	0	0

图6-12 A口内部上拉电阻使能（PTAPE）(1)

(1)3-6位是保留位，必须被写为0。

表 6-3 PTAPE 寄存器域描述

域	描述
7, 2: 0 PTAPEn	A口内部上拉电阻使能位——每个控制位决定相应的PTA引脚是否允许内部上拉电阻。当A口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 A口n位禁止内部上拉电阻。 1 A口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	PTASE7	R	R	R	R	PTASE2	PTASE1	PTASE0
写								
复位	0	0	0	0	0	0	0	0

图6-13 A口转换速率控制使能寄存器（PTASE）(1)

(1)3-6位是保留位，必须被写为0。

表 6-4 PTASE 寄存器域描述

域	描述
7, 2: 0 PTASEn	A口输出转换速率使能位——每个控制位决定相应的PTA引脚是否允许输出转换速率控制。当A口设置为输入时，这些位不起作用。 0 A口n位禁止输出转换速率控制。 1 A口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	PTADS7	R	R	R	R	PTADS2	PTADS1	PTADS0
写								
复位	0	0	0	0	0	0	0	0

图6-14 A口输出驱动强度寄存器（PTADS）(1)

(1)3-6位是保留位，必须被写为0。

表 6-5 PTADS 寄存器域描述

域	描述
7, 2: 0 PTADS _n	A口输出驱动强度选择位——每个控制位决定相应的PTA引脚选择低还是高输出驱动。 0 A口 _n 位低输出驱动。 1 A口 _n 位高输出驱动。

6.7.3 B口I/O寄存器（PTBD和PTBDD）

B口并行 I/O 功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTBD3	PTBD2	PTBD1	PTBD0
写	R	R	R	R	PTBD3	PTBD2	PTBD1	PTBD0
复位	0	0	0	0	0	0	0	0

图6-15 B口数据寄存器（PTBD）(1)

(1)4-7位是保留位，必须被写为0。

表 6-6 PTBD 寄存器域描述

域	描述
3: 0 PTBD _n	B口数据寄存器位——B口引脚位输入时，读操作引脚的逻辑电平。B口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。B口引脚为输出时，逻辑电平有相应的MCU引脚驱动。复位强制PTBD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTBDD3	PTBDD2	PTBDD1	PTBDD0
写	R	R	R	R	PTBDD3	PTBDD2	PTBDD1	PTBDD0
复位	0	0	0	0	0	0	0	0

图6-16 B口数据方向寄存器（PTBDD）(1)

(1)4-7位是保留位，必须被写为0。

表 6-7 PTBDD 寄存器域描述

域	描述
3: 0 PTBDD[3:0]	B口数据方向寄存器位——这些可读写位控制B口引脚的方向和PTBD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 B口 _n 位输出使能而且PTBD读返回PTBD _n 的内容。

6.7.4 B口引脚控制寄存器（PTBPE、PTBSE、PTBDS）

除了 I/O 控制，B口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTBPE3	PTBPE2	PTBPE1	PTBPE0
写	R	R	R	R	PTBPE3	PTBPE2	PTBPE1	PTBPE0

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

图6-17 B口内部上拉电阻使能（PTBPE）(1)

(1)4-7位是保留位，必须被写为0。

表 6-8 PTBPE 寄存器域描述

域	描述
3: 0 PTBPE[3:0]	B口内部上拉电阻使能位——每个控制位决定相应的PTB引脚是否允许内部上拉电阻。当B口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 B口n位禁止内部上拉电阻。 1 B口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTBSE3	PTBSE2	PTBSE1	PTBSE0
写	R	R	R	R	PTBSE3	PTBSE2	PTBSE1	PTBSE0
复位	0	0	0	0	0	0	0	0

图6-18 B口转换速率控制使能寄存器（PTBSE）(1)

(1)4-7位是保留位，必须被写为0。

表 6-9 PTBSE 寄存器域描述

域	描述
3: 0 PTBSE[3:0]	B口输出转换速率使能位——每个控制位决定相应的PTB引脚是否允许输出转换速率控制。当B口设置为输入时，这些位不起作用。 0 B口n位禁止输出转换速率控制。 1 B口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTBDS3	PTBDS2	PTBDS1	PTBDS0
写	R	R	R	R	PTBDS3	PTBDS2	PTBDS1	PTBDS0
复位	0	0	0	0	0	0	0	0

图6-19 B口输出驱动强度寄存器（PTBDS）(1)

(1)4-7位是保留位，必须被写为0。

表 6-10 PTBDS 寄存器域描述

域	描述
3: 0 PTBDS[3:0]	B口输出驱动强度选择位——每个控制位决定相应的PTB引脚选择低还是高输出驱动。 0 B口n位低输出驱动。 1 B口n位高输出驱动。

6.7.5 C口I/O寄存器（PTCD和PTCDD）

C口并行I/O功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	0	R	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
写	0	R	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
复位	0	0	0	0	0	0	0	0

图6-20 C口数据寄存器（PTCD）(1)

(1)6位是保留位，必须被写为0。

表 6-11 PTCDD 寄存器域描述

域	描述
5: 0 PTCDD[5:0]	C口数据寄存器位——C口引脚位输入时，读操作引脚的逻辑电平。C口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。C口引脚为输出时，逻辑电平有相应的MCU引脚驱动。 复位强制PTCDD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	0	R	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
写								
复位	0	0	0	0	0	0	0	0

图6-21 C口数据方向寄存器（PTCDD）(1)

(1)6位是保留位，必须被写为0。

表 6-12 PTCDD 寄存器域描述

域	描述
5: 0 PTCDD[5:0]	C口数据方向寄存器位——这些可读写位控制C口引脚的方向和PTCDD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 C口n位输出使能而且PTCDD读返回PTCDDn的内容。

6.7.6 C口引脚控制寄存器（PTCPE、PTCSE、PTCDS）

除了 I/O 控制，C 口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	0	R	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
写								
复位	0	0	0	0	0	0	0	0

图6-22 C口内部上拉电阻使能（PTCPE）(1)

(1)6位是保留位，必须被写为0。

表 6-13 PTCPE 寄存器域描述

域	描述
5: 0 PTCPE[5:0]	C口内部上拉电阻使能位——每个控制位决定相应的PTC引脚是否允许内部上拉电阻。当C口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 C口n位禁止内部上拉电阻。 1 C口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	0	R	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
写								
复位	0	0	0	0	0	0	0	0

图6-23 C口转换速率控制使能寄存器（PTCSE）(1)

(1)6位是保留位，必须被写为0。

表 6-14 PTCSE 寄存器域描述

域	描述
5: 0 PTCSE[5:0]	C口输出转换速率使能位——每个控制位决定相应的PTC引脚是否允许输出转换速率控制。当C口设置为输入时，这些位不起作用。 0 C口n位禁止输出转换速率控制。 1 C口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	PTCDS7	R	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
写								
复位	0	0	0	0	0	0	0	0

图6-24 C口输出驱动强度寄存器（PTCDS）(1)

(1)6位是保留位，必须被写为0。

表 6-15 PTCDS 寄存器域描述

域	描述
5: 0 PTCDS[5:0]	C口输出驱动强度选择位——每个控制位决定相应的PTC引脚选择低还是高输出驱动。 0 C口n位低输出驱动。 1 C口n位高输出驱动。

6.7.7 D口I/O寄存器（PTDD和PTDDD）

D口并行I/O功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTDD3	PTDD2	PTDD1	PTDD0
写								
复位	0	0	0	0	0	0	0	0

图6-25 D口数据寄存器（PTDD）(1)

(1)4-7位是保留位，必须被写为0。

表 6-16 PTDD 寄存器域描述

域	描述
3: 0 PTDD[3:0]	D口数据寄存器位——D口引脚位输入时，读操作引脚的逻辑电平。D口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。D口引脚为输出时，逻辑电平有相应的MCU引脚驱动。复位强制PTDD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTDDD3	PTDDD2	PTDDD1	PTDDD0
写								
复位	0	0	0	0	0	0	0	0

图6-26 D口数据方向寄存器（PTDDD）(1)

(1)4-7位是保留位，必须被写为0。

表 6-17 PTDDD 寄存器域描述

域	描述
---	----

3: 0 PTDDD[3:0]	D口数据方向寄存器位——这些可读写位控制D口引脚的方向和PTDD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 D口n位输出使能而且PTDD读返回PTDDn的内容。
--------------------	---

6.7.8 D口引脚控制寄存器（PTDPE、PTDSE、PTDDS）

除了 I/O 控制，D 口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTDPE3	PTDPE2	PTDPE1	PTDPE0
写	R	R	R	R	PTDPE3	PTDPE2	PTDPE1	PTDPE0
复位	0	0	0	0	0	0	0	0

图6-27 D口内部上拉电阻使能（PTDPE）(1)

(1)4-7位是保留位，必须被写为0。

表 6-18 PTDPE 寄存器域描述

域	描述
3: 0 PTDPE[3:0]	D口内部上拉电阻使能位——每个控制位决定相应的PTD引脚是否允许内部上拉电阻。当D口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 D口n位禁止内部上拉电阻。 1 D口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTDSE3	PTDSE2	PTDSE1	PTDSE0
写	R	R	R	R	PTDSE3	PTDSE2	PTDSE1	PTDSE0
复位	0	0	0	0	0	0	0	0

图6-28 D口转换速率控制使能寄存器（PTDSE）(1)

(1)4-7位是保留位，必须被写为0。

表 6-19 PTDSE 寄存器域描述

域	描述
3: 0 PTDSE[3:0]	D口输出转换速率使能位——每个控制位决定相应的PTD引脚是否允许输出转换速率控制。当D口设置为输入时，这些位不起作用。 0 D口n位禁止输出转换速率控制。 1 D口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	R	R	R	R	PTDDS3	PTDDS2	PTDDS1	PTDDS0
写	R	R	R	R	PTDDS3	PTDDS2	PTDDS1	PTDDS0
复位	0	0	0	0	0	0	0	0

图6-29 D口输出驱动强度寄存器（PTDDS）(1)

(1)4-7位是保留位，必须被写为0。

表 6-20 PTDDS 寄存器域描述

域	描述
3: 0 PTDDS[3:0]	D口输出驱动强度选择位——每个控制位决定相应的PTD引脚选择低还是高输出驱动。 0 D口n位低输出驱动。 1 D口n位高输出驱动。

6.7.9 E口I/O寄存器（PTED和PTEDD）

E口并行I/O功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
写								
复位	0	0	0	0	0	0	0	0

图6-30 E口数据寄存器（PTED）

表 6-21 PTED 寄存器域描述

域	描述
7: 0 PTED [7:0]	E口数据寄存器位——E口引脚位输入时，读操作引脚的逻辑电平。E口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。E口引脚为输出时，逻辑电平有相应的MCU引脚驱动。 复位强制PTED为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
写								
复位	0	0	0	0	0	0	0	0

图6-31 E口数据方向寄存器（PTEDD）

表 6-22 PTEDD 寄存器域描述

域	描述
7: 0 PTEDD[7:0]	E口数据方向寄存器位——这些可读写位控制E口引脚的方向和PTED读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 E口n位输出使能而且PTED读返回PTEDn的内容。

6.7.10 E口引脚控制寄存器（PTEPE、PTESE、PTEDS）

除了I/O控制，E口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
写								
复位	0	0	0	0	0	0	0	0

图6-32 E口内部上拉电阻使能（PTEPE）

表 6-23 PTEPE 寄存器域描述

域	描述
7: 0 PTEPE[7:0]	E口内部上拉电阻使能位——每个控制位决定相应的PTE引脚是否允许内部上拉电阻。当E口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 E口n位禁止内部上拉电阻。 1 E口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
写								

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

图6-33 E口转换速率控制使能寄存器（PTESE）

表 6-24 PTESE 寄存器域描述

域	描述
7: 0 PTESE[7:0]	E口输出转换速率使能位——每个控制位决定相应的PTE引脚是否允许输出转换速率控制。当E口设置为输入时，这些位不起作用。 0 E口n位禁止输出转换速率控制。 1 E口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
写								
复位	0	0	0	0	0	0	0	0

图6-34 E口输出驱动强度寄存器（PTEDS）

表 6-25 PTEDS 寄存器域描述

域	描述
7: 0 PTEDS[7:0]	E口输出驱动强度选择位——每个控制位决定相应的PTE引脚选择低还是高输出驱动。 0 E口n位低输出驱动。 1 E口n位高输出驱动。

6.7.11 F口I/O寄存器（PTFD和PTFDD）

F口并行I/O功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	PTFD6	PTFD5	PTFD4	R	R	PTFD1	PTFD0
写								
复位	0	0	0	0	0	0	0	0

图6-35 F口数据寄存器（PTFD）(1)

(1)7,3-2位是保留位，必须被写为0。

表 6-26 PTFD 寄存器域描述

域	描述
6:4, 1:0 PTFDn	F口数据寄存器位——F口引脚位输入时，读操作引脚的逻辑电平。F口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。F口引脚为输出时，逻辑电平有相应的MCU引脚驱动。复位强制PTFD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	R	PTFDD6	PTFDD5	PTFDD4	R	R	PTFDD1	PTFDD0
写								
复位	0	0	0	0	0	0	0	0

图6-36 F口数据方向寄存器（PTFDD）(1)

(1)3-6位是保留位，必须被写为0。

表 6-27 PTFDD 寄存器域描述

域	描述
6:4, 1:0 PTFDDn	F口数据方向寄存器位——这些可读写位控制F口引脚的方向和PTFD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 F口n位输出使能而且PTFD读返回PTFDn的内容。

6.7.12 F口引脚控制寄存器（PTFPE、PTFSE、PTFDS）

除了 I/O 控制，F 口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	R	PTFPE6	PTFPE5	PTFPE4	R	R	PTFPE1	PTFPE0
写								
复位	0	0	0	0	0	0	0	0

图6-37 F口内部上拉电阻使能（PTFPE）(1)

(1)3-6位是保留位，必须被写为0。

表 6-28 PTFPE 寄存器域描述

域	描述
6:4, 1:0 PTFPEn	F口内部上拉电阻使能位——每个控制位决定相应的PTF引脚是否允许内部上拉电阻。当F口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 F口n位禁止内部上拉电阻。 1 F口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	R	PTFSE6	PTFSE5	PTFSE4	R	R	PTFSE1	PTFSE0
写								
复位	0	0	0	0	0	0	0	0

图6-38 F口转换速率控制使能寄存器（PTFSE）(1)

(1)3-6位是保留位，必须被写为0。

表 6-29 PTFSE 寄存器域描述

域	描述
6:4, 1:0 PTFSEn	F口输出转换速率使能位——每个控制位决定相应的PTF引脚是否允许输出转换速率控制。当F口设置为输入时，这些位不起作用。 0 F口n位禁止输出转换速率控制。 1 F口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	R	PTFDS6	PTFDS5	PTFDS4	R	R	PTFDS1	PTFDS0
写								
复位	0	0	0	0	0	0	0	0

图6-39 F口输出驱动强度寄存器（PTFDS）

表 6-30 PTFDS 寄存器域描述

域	描述
---	----

6:4, 1:0 PTFDSn	F口输出驱动强度选择位——每个控制位决定相应的PTF引脚选择低还是高输出驱动。 0 F口n位低输出驱动。 1 F口n位高输出驱动。
--------------------	---

6.7.13 G口I/O寄存器（PTGD和PTGDD）

G口并行I/O功能被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
写								
复位	0	0	0	0	0	0	0	0

图6-40 G口数据寄存器（PTGD）

表 6-31 PTGD 寄存器域描述

域	描述
6: 0 PTGD[6:0]	G口数据寄存器位——G口引脚位输入时，读操作引脚的逻辑电平。G口为输出时，读操作返回最后写入这个寄存器的值。 写被锁存在寄存器的所有位。G口引脚为输出时，逻辑电平有相应的MCU引脚驱动。复位强制PTGD为全0，但这些0不是相应引脚驱动的，因为复位设置所有的端口引脚为禁止上拉电阻的高阻抗输入。

	第7位	6	5	4	3	2	1	第0位
读	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
写								
复位	0	0	0	0	0	0	0	0

图6-41 G口数据方向寄存器（PTGDD）

表 6-32 PTGDD 寄存器域描述

域	描述
6: 0 PTGDD[6:0]	G口数据方向寄存器位——这些可读写位控制G口引脚的方向和PTGD读操作时所读的内容。 0 输入（输出禁止）而且读返回引脚值。 1 G口n位输出使能而且PTGD读返回PTGDn的内容。

6.7.14 G口引脚控制寄存器（PTGPE、PTGSE、PTGDS）

除了I/O控制，G口引脚被下列寄存器控制：

	第7位	6	5	4	3	2	1	第0位
读	0	PTGPE6	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
写								
复位	0	0	0	0	0	0	0	0

图6-42 G口内部上拉电阻使能（PTGPE）

表 6-33 PTGPE 寄存器域描述

域	描述
6: 0 PTGPE[6:0]	G口内部上拉电阻使能位——每个控制位决定相应的PTG引脚是否允许内部上拉电阻。当G口设置为输出时，这些位不起作用而且内部上拉电阻被禁止。 0 G口n位禁止内部上拉电阻。 1 G口n位允许内部上拉电阻。

	第7位	6	5	4	3	2	1	第0位
读	0	PTGSE6	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
写								
复位	0	0	0	0	0	0	0	0

图6-43 G口转换速率控制使能寄存器（PTGSE）

表 6-34 PTGSE 寄存器域描述

域	描述
6: 0 PTGSE[6:0]	G口输出转换速率使能位——每个控制位决定相应的PTG引脚是否允许输出转换速率控制。当G口设置为输入时，这些位不起作用。 0 G口n位禁止输出转换速率控制。 1 G口n位允许输出转换速率控制。

	第7位	6	5	4	3	2	1	第0位
读	0	PTGDS6	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
写								
复位	0	0	0	0	0	0	0	0

图6-44 G口输出驱动强度寄存器（PTGDS）(1)

(1)3-6位是保留位，必须被写为0。

表 6-35 PTGDS 寄存器域描述

域	描述
6: 0 PTGDS[6:0]	G口输出驱动强度选择位——每个控制位决定相应的PTG引脚选择低还是高输出驱动。 0 G口n位低输出驱动。 1 G口n位高输出驱动。

第七章 中央处理单元（S08CPUV2）

7.1 介绍

这部分提供有关 HCS08 些列 CPU 寄存器，寻址方式，指令设置摘要信息。更详细的信息参考 HCS08 些列参考手册 1，飞思卡尔半导体文件序号 HCS08RMV1/D。HCS08 CPU 与 M68HC08 的 CPU 是完全源和目标代码兼容。一些指令和增强寻址模式的加入以提高 C 编译器效率和支持一个新的背景调试系统以代替早期 M68HC08 的监控模式。

7.1.1 特征

HCS08 CPU 特征

- 目标代码完全向上兼容 M68HC05 和 M68HC08 家族
- 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（随时随地在 64k 字节的地址空间任何规模的栈）
- 16 位索引寄存器（H:X）带强大的索引地址模式
- 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
 - 相对 8 位有符号偏移量寻址
 - 立即寻址方式
 - 直接寻址方式
 - 扩展寻址方式
 - H:X 相对变址寻址方式
 - SP 相对变址寻址方式
- 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 编程结构和CPU寄存器

如图 7-1 所示 5 个 CPU 寄存器，这些 CPU 寄存器不在微处理器的存储器中

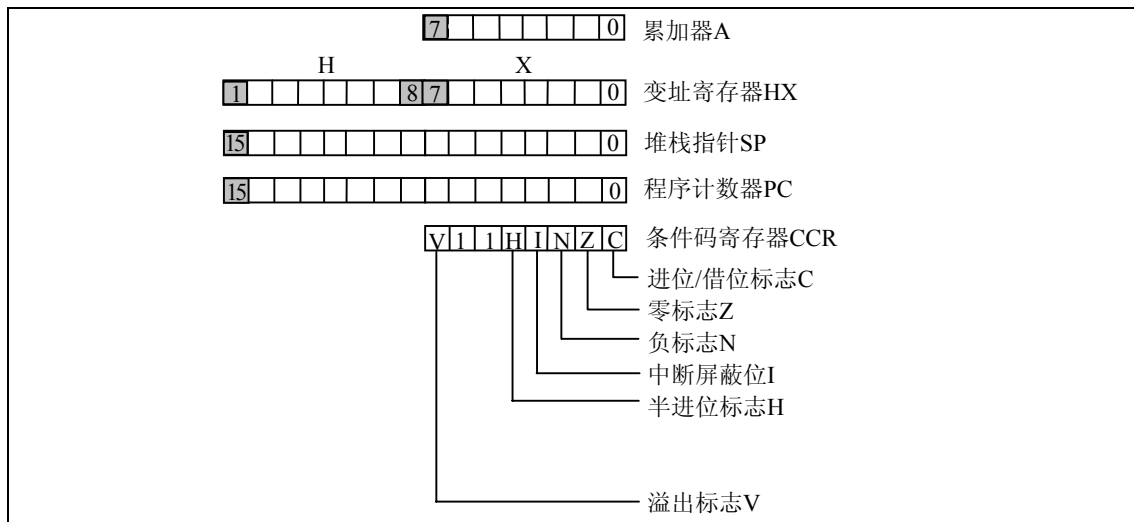


图7-1CPU寄存器

7.2.1 累加器(A)

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)是与累加器和 ALU 结果相关的,在算术和逻辑运算后往往存放入 A 累加器。累加器可以用不同的寻址模式指定地址从存储器中转入数据,或者累加器 A 中的数据可以根据不同的寻址模式制定的地址装入到存储器中。复位键对累加器中的数据无影响。

7.2.2 索引寄存器 (H:X)

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。所有索引寻址方式指令用 H:X 中的 16 位全值作为索引参考值。无论怎样,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)

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 64 字节任意地址空间。堆栈可以自动保存子程序调用返回值,中断地址返回值和 CPU 寄存器值,和本地变量。AIS 指令加 8 位有符号立即数给 SP。这是最常用的为本地堆栈变量分配或回收空间。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 RAM 最低地址到 0X00FF) 时, HCS08 程序在片上 RAM 中通常将 SP 中的值改为最后地址值 (高地址)。RSP (复位对战指针) 指令和 M68HC05 系列兼容且很少用在 HCS08 程序中,因为它仅影响对战指针的低地址。

7.2.4 程序计数器 (PC)

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。通常在程序执行

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0XFFF处装载复位向量。向量存储位置的第一个指令将在退出复位状态后被执行。

7.2.5 条件码寄存器(CCR)

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，请参阅 HCS08 系列参考手册 1，飞思卡尔半导体文件序号 HCS08RMv1。

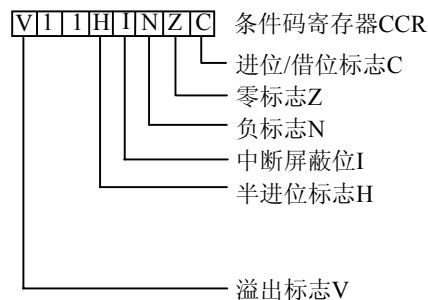


图 7-2条件码寄存器(CCR)

表 7-1.CCR 寄存器位功能描述

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。 无溢出 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。 允许中断 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。 无零结果 有零结果 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。 无进位/借位 有进位/借位

7.3 寻址模式

寻址模式确定了 CPU 存取数据和操作数的路径。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 64 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。一些指令有不止一种寻址方式。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，列在指令设置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）

相对寻址方式用于为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，导致程序在转移目标地址处完成。

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 的大部分行为在指令集中被详尽描述，但仍需考虑特殊指令的操作。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。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，中断和系统结构章节。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. 设置 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. 为中断服务例程将程序信息的首 3 字节填入指令队列为第一条指令执行作准备。

CCR 中的内容被压入堆栈后,在进行中断服务例程时设置 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（这是不被认可的,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

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,事件将被正常执行。当 CPU 在等待模式中如果串行的背景调试指令通过背景调试界面流入到 MCU, CPU 时钟将被开启并进入其它串行后台指令能被处理的活跃后台模式。这样确保即使在等待模式中主机开发系统仍能进入目标 MCU。

7.4.4 停止模式

通常在停止模式下,系统的所有时钟包括晶振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一批最小值时钟。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串行命令设置 ENBDM 位时,在 MCU 进入停止模式时振荡器被迫保持活跃。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 时钟将被开启并进入其它串行后台指令能被处理的活跃后台模式。这样确保即使在等待模式中主机开发系统仍能进入目标 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 背景模式

与 M68HC08 相比,对 HCS08 来说背景调试指令是一个新指令。背景不会用在普通用户程序因为它强迫 CPU 停止处理用户指令和进入活跃背景模式。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输入的 GO、TRACE1、TAGGO 串行命令。软断点可以通过用背景调试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 更可能进入活跃背景模式而不是继续用户程序。

7.5 HCS08指令设置摘要

指令设置摘要术语表 7-2 中指令术语

操作符:

- () =寄存器或存储器内容显示在圆括号里
- ← =被装入（读：获得）

& =布尔与
 | =布尔或
 ⊕ =布尔异或
 : =连接
 x =乘
 ÷ =除
 + =加
 - =减

CPU 寄存器:

A=累加器
 CCR=条件码寄存器
 H=变址寄存器高 8 位
 X=变址寄存器低 8 位
 PC=程序计数器
 PCH=程序计数器高 8 位
 PCL=程序计数器低 8 位
 SP=堆栈指针

存储和寻址:

M=依赖寻址方式的存储位置或绝对数据
 M:M+0X0001=在两个连续存储位置的 16 位值，高 8 位定位到 M 的地址，低 8 位定位到下个高连续地址。
 条件寄存器（CCR）位
 V=溢出标志，位 7
 H=半进位标志，位 4
 I=中断屏蔽标志，位 3
 N=负标志位，位 2
 Z=零标志，位 1
 C=进位/借位标志，位 0

CCR 常用符号:

- =位不影响
 0 =位改为 0
 1 =位改为 1
 =根据操作结果位设置或清零
 U=操作后未定义

机器码符号:

dd=直接地址 0X0000-0X00FF 低 8 位（假设高字节为 0X00）
 ee=16 位偏移量高 8 位
 ff=16 位的低 8 位或 8 位
 ii= 一字节立即数
 jj=16 位立即数值高位字节
 kk=16 位立即数值低位字节
 hh=16 位扩展地址高位字节

ll = 16 位扩展地址低位字节

rr = 相对偏移量

源格式:

源格式中除去用斜体表示的表达式其它文字信息必须用汇编源文件正确表示。原始的 3 至 5 个字母一直是一个文字信息。所有的逗号，井号 (#)，圆括号，加号 (+) 都是文字字符。

- n — 任何用来估值 0-7 中信号数的标签或表达式
- opr8i — 任何用来估值 8 位立即数值的标签或表达式
- opr16i — 任何用来估值 16 位立即数值的标签或表达式
- opr8a — 任何用来估值 8 位立即数值的标签或表达式。指令把 8 位值当作 64 字节直接页地址空间 (0X00XX) 地址的低 8 位。
- opr16a — 任何用来估值 16 位立即数值的标签或表达式。指令把这个值当作 64 字节地址空间的地址
- oprx8 — 任何用来估值 8 位立即数值的标签或表达式用于变址寻址
- oprx16 — 任何用来估值 16 位立即数值的标签或表达式。因为 HCS08 有 16 位地址总线，所以是有符号值或无符号值
- rel — 任何标签或表达式所涉及地址在当前指令末子节目标操作码后的下个地址起 -128 至 +127 之间。汇编程序将会计算 8 位有符号偏移量的值和把它包含在这条指令飞目标码中。

寻址方式:

- INH = 内在寻址方式
- IMM = 立即寻址方式
- DIR = 直接寻址方式
- EXT = 扩展寻址方式
- IX = 无偏移量变址方式
- IX+ = 无偏移量变址、变址加 1 寻址方式
- IX2 = 16 位偏移量变址方式
- REL = 相对寻址方式
- SP1 = 8 位偏移量堆栈寻址方式
- SP2 = 16 位偏移量堆栈寻址方式

表 7-2 HCS08 指令设置摘要(1/7)

源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
ADC#opr8i	进位加	$A \leftarrow (A) + (M) + (C)$							IMM	A9	ii dd	2
ADCopr8a									DIR	B9	hh	3
ADCopr16a									EXT	C9	ll ee	4
ADCoprx16,X			↑	↑	—	↑	↑	↑	IX2	D9	ff ff	4
ADC oprx8,X									IX1	E9		3
ADC ,X									IX	F9	ee	3
ADCoprx16,SP									SP2	9ED9	ff	5
ADC oprx8,SP									SP1	9EE9	ff	4

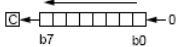
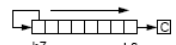
ADD #opr8i ADD opr8a ADD opr16a ADDopr16,X ADD oprx8,X ADD ,X ADDopr16,SP ADD oprx8,SP	无进位加	$A \leftarrow (A) + (M)$							IMM DIR EXT IX2 IX1 IX SP2 SP1	AB BB CB DB EB FB 9EDB 9EEB	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
AIS #opr8i	将立即数（有符号）压入堆栈r	$SP \leftarrow (SP) + (M)$ M是有符号扩展为16位值	-	-	-	-	-	-	IMM	A7	ii	2
AIX #opr8i	将立即数（有符号）压入编制寄存器（H:X）	$H:X \leftarrow (H:X) + (M)$ M是有符号扩展为16位值	-	-	-	-	-	-	IMM	AF	ii	2
AND #opr8i AND opr8a AND opr16a ANDopr16,X AND oprx8,X AND ,X ANDopr16,SP AND oprx8,SP	逻辑与	$A \leftarrow (A) \& (M)$	0	-	-				IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 B4 C4 D4 E4 F4 9ED4 9EE4	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
ASLopr8a ASLA ASLX ASLopr16,X ASL ,X ASL oprx8,SP	算术左移 (Same as LSL)								DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
ASRopr8a ASRA ASRX ASRopr16,X ASR ,X ASR oprx8,SP	算术右移								DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	5 1 1 5 4 6
BCC rel	进位位清零则转移	(C) = 0则转移	-	-	-	-	-	-	REL	24	rr	3

表 7-2 HCS08 指令设置摘要(2/7)

源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
BCLR n,opr8a	存储器中位 n 清零	$M_n \leftarrow 0$	-						DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5

BCS <i>rel</i>	进位位置1则转移 (Same as BLO)	(C) = 1则转移	—	—	—	—	—	—	REL	25	rr	3
BEQ <i>rel</i>	相等则转移	(Z) = 1则转移	—	—	—	—	—	—	REL	27	rr	3
BGE <i>rel</i>	大于等于则转移 (Signed Operands)	(N ⊕ V) = 0则转移	—	—	—	—	—	—	REL	90	rr	3
BGND	ENBDM = 1则 进入活跃背景调 试状态	等待和处理BDM指 令直到 GO, TRACE1, TAGGO	—	—	—	—	—	—	INH	82		5+
BGT <i>rel</i>	大于则转移 (Signed Operands)	(Z) (N ⊕ V) = 0则转 移	—	—	—	—	—	—	REL	92	rr	3
BHCC <i>rel</i>	半进位清零则转 移	(H) = 0则转移	—	—	—	—	—	—	REL	28	rr	3
BHCS <i>rel</i>	半进位置1则转 移	(H) = 1则转移	—	—	—	—	—	—	REL	29	rr	3
BHI <i>rel</i>	为高则转移	(C) (Z) = 0则转移	—	—	—	—	—	—	REL	22	rr	3
BHS <i>rel</i>	高于或相同则转 移 (Same as BCC)	(C) = 0则转移	—	—	—	—	—	—	REL	24	rr	3
BIH <i>rel</i>	IRQ引脚为高则 转移	IRQ pin = 1则转移	—	—	—	—	—	—	REL	2F	rr	3
BIL <i>rel</i>	Branch if IRQ Pin Low	IRQ pin = 0则转移	—	—	—	—	—	—	REL	2E	rr	3
BIT#opr8i BITopr8a BITopr1a BITopr16,X BIT oprx8,X BIT ,X BITopr16,SP BIT oprx8,SP	位测试	(A) & (M) CCR更新但操作数 不变	0	—	—	↑	↑	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 B5 C5 D5 E5 F5 9ED5 9EE5	ii dd hh ll ee ff ff ff	2 3 4 4 3 3 3 5 4
BLE <i>rel</i>	小于等于则 转移 (Signed Operands)	(Z) (N ⊕ V) = 1则 转移	—	—	—	—	—	—	REL	93	rr	3
BLO <i>rel</i>	为低则转移 (Same as BCS)	(C) = 1 则转移	—	—	—	—	—	—	REL	25	rr	3
BLS <i>rel</i>	为低或相同则转 移	(C) (Z) = 1则转移	—	—	—	—	—	—	REL	23	rr	3
BLT <i>rel</i>	小于则转移 (Signed Operands)	(N ⊕ V) = 1则转移	—	—	—	—	—	—	REL	91	rr	3
BMC <i>rel</i>	中断屏蔽位清零 则转移	(I) = 0则转移	—	—	—	—	—	—	REL	2C	rr	3
BMI <i>rel</i>	负则转移	(N) = 1则转移	—	—	—	—	—	—	REL	2B	rr	3
BMS <i>rel</i>	中断屏蔽位置1 则转移	(I) = 1则转移	—	—	—	—	—	—	REL	2D	rr	3
BNE <i>rel</i>	不等则转移	(Z) = 0则转移	—	—	—	—	—	—	REL	26	rr	3
BPL <i>rel</i>	正则转移	(N) = 0则转移	—	—	—	—	—	—	REL	2A	rr	3
BRA <i>rel</i>	一直转移	无测 试	—	—	—	—	—	—	REL	20	rr	3

表 7-2 HCS08 指令设置摘要(3/7)

源格式	操作	描述	CCR	代 方 址 寻	操作	操	总
-----	----	----	-----	---------	----	---	---

			V	H	I	N	Z	C		码	作 数	线 周 期
BRCLR <i>n,opr8a,rel</i>	存储器中位n 清零	(Mn) = 0则转移	—	—	—	—	—	↓	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd	5 5 5 5 5 5 5 5
BRN <i>rel</i>	从不转移	占用3个总线周期	—	—	—	—	—	—	REL	21	rr	3
BRSET <i>n,opr8a,rel</i>	存储器中位n 置1则转移	(Mn) = 1则转移	—	—	—	—	—	↓	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd	5 5 5 5 5 5 5 5
BSET <i>n,opr8a</i>	存储器中位n 置1	Mn←1	—	—	—	—	—	—	DIR(b0) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR <i>rel</i>	转移到子程序	PC ← (PC) + 0x0002 push (PCL); SP ← (SP) – 0x0001push (PCH); SP ← (SP) – 0x0001PC ← (PC) + <i>rel</i>	—	—	—	—	—	—	REL	AD	rr	5
CBEQ <i>opr8a,rel</i> CBEQA# <i>opr8i,rel</i> CBEQX# <i>opr8i,rel</i> CBEQ <i>opr8,X+,rel</i> CBEQ <i>,X+,rel</i> CBEQ <i>opr8,SP,rel</i>	相等则比较转 移	(A) = (M)则转移 (A) = (M) 则转移 (X) = (M) 则转移 (A) = (M) 则转移 (A) = (M) 则转移(A) = (M) 则转移	—	—	—	—	—	—	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff	5 4 4 5 5 6
CLC	清进位位	C ← 0	—	—	—	—	—	0	INH	98		1
CLI	清中断屏蔽位	I ← 0	—	—	0	—	—	—	INH	9A		1
CLR <i>opr8a</i> CLRA CLR X CLR H CLR <i>opr8,X</i> CLR <i>,X</i> CLR <i>opr8,SP</i>	清位	M←0x00 A←0x00 X←0x00 H←0x00 M←0x00 M←0x00 M←0x00	0	—	—	0	1	—	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	5 1 1 1 5 4 6
CMP # <i>opr8i</i> CMP <i>opr8a</i> CMP <i>opr16a</i> CMP <i>opr16,X</i> CMP <i>opr8,X</i> CMP <i>,X</i> CMP <i>opr16,SP</i> CMP <i>opr8,SP</i>	存储器与累加 器相比	(A) & (M) CCR更新但操作数 不变	↓	—	—	↓	↓	↓	IMM DIR EXT IX2 IX1 IX SP2 SP1	A1 B1 C1 D1 E1 F1 9ED1 9EE	ii dd hh ll ee ff ff ff ff ee ff ff	2 3 4 4 3 3 5 4
COM <i>opr8a</i> COMA COMX COM <i>opr8,X</i> COM <i>,X</i> COM <i>opr8,SP</i>	求补 (One's Complement)	M←(M)=0xFF-(M) A←(A)= 0xFF-(A) X←(X)= 0xFF-(X) M←(M)=0xFF-(M) M←(M)=0xFF-(M) M←(M)=0xFF-(M)	0	—	—	↓	↓	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	5 1 1 5 4 6

表 7-2 HCS08 指令设置摘要(4/7)

源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
CPX <i>#opr8i</i> CPX <i>opr8a</i> CPX <i>opr16a</i> CPX <i>opr16,X</i> CPX <i>opr8,X</i> CPX <i>,X</i> CPX <i>opr16,SP</i> CPX <i>opr8,SP</i>	X与存储器相比	(X) - (M) CCR更新但操作数不变	↓	-	-	↓	↓	↓	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 ii B3 dd C3 hh D3 ll ee E3 ff ff F3 9ED3 ee 9EE3 ff ff		2 3 4 4 3 3 5 4
DAA	在BCD码ADD,ADC操作后 转换累加器内容到十进制	(A) ₁₀	U	-	-	↓	↓	↓	INH	72		1
DBNZ <i>opr8a,rel</i> DBNZ <i>rel</i> DBNZX <i>rel</i> DBNZ <i>opr8,X,rel</i> DBNZ <i>,X,rel</i> DBNZ <i>opr8,SP,rel</i>	不为零则自减转移	A, X, or M自减 (result) ≠ 0则转移 DBNZX 影响 X 不影响 H	-	-	-	-	-	-	DIR INH INH IX1 IX SP1	3B dd 4B rr rr 5B rr rr 6B ff ff 7B rr rr 9E6B rr ff		7 4 4 7 6 8
DEC <i>opr8a</i> DECA DECX DEC <i>opr8,X</i> DEC <i>,X</i> DEC <i>opr8,SP</i>	自减	M ← (M) - 0x01 A ← (A) - 0x01 X ← (X) - 0x01 M ← (M) - 0x01 M ← (M) - 0x01 M ← (M) - 0x01	↓	-	-	↓	↓	-	DIR INH INH IX1 IX SP1	3A dd 4A 5A 6A 7A ff 9E6A ff		5 1 1 5 4 6
DIV	除	A ← (H:A) ÷ (X) H ← 余数	-	-	-	-	↓	↓	INH	52		6
EOR <i>#opr8i</i> EOR <i>opr8a</i> EOR <i>opr16a</i> EOR <i>opr16,X</i> EOR <i>opr8,X</i> EOR <i>,X</i> EOR <i>opr16,SP</i> EOR <i>opr8,SP</i>	带累加器的存储器异或	A ← (A ⊕ M)	0	-	-	↓	↓	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 ii B8 dd C8 hh D8 ll ee E8 ff ff F8 9ED8 ee 9EE8 ff ff		2 3 4 4 3 3 5 4
INC <i>opr8a</i> INCA INCX INC <i>opr8,X</i> INC <i>,X</i> INC <i>opr8,SP</i>	自增	M ← (M) + 0x01 A ← (A) + 0x01 X ← (X) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01	↓	-	-	↓	↓	-	DIR INH INH IX1 IX SP1	3C dd 4C 5C 6C 7C ff 9E6C ff		5 1 1 5 4 6
JMP <i>opr8a</i> JMP <i>opr16a</i> JMP <i>opr16,X</i> JMP <i>opr8,X</i> JMP <i>,X</i>	跳转	PC ← 跳转地址	-	-	-	-	-	-	DIR EXT IX2 IX1 IX	BC dd CC hh DC ee EC ff FC	ll ff	3 4 4 3 3
JSR <i>opr8a</i> JSR <i>opr16a</i> JSR <i>opr16,X</i> JSR <i>opr8,X</i> JSR <i>,X</i>	跳转到子程序	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) - 0x0001 Push (PCH); SP ← (SP) - 0x0001 PC ← Unconditional Address	-	-	-	-	-	-	DIR EXT IX2 IX1 IX	BD dd CD hh DD ee ED ff FD	ll ff	5 6 6 5 5
LDA <i>#opr8i</i> LDA <i>opr8a</i> LDA <i>opr16a</i> LDA <i>opr16,X</i> LDA <i>opr8,X</i> LDA <i>,X</i> LDA <i>opr16,SP</i> LDA <i>opr8,SP</i>	从存储器装入累加器	A ← (M)	0	-	-	↓	↓	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 ii B6 dd C6 hh D6 ll ee E6 ff ff F6 9ED6 ee 9EE6 ff ff		2 3 4 4 3 3 5 4
LDHX <i>#opr16i</i> LDHX <i>opr8a</i> LDHX <i>opr16a</i> LDHX <i>,X</i> LDHX <i>opr16,X</i> LDHX <i>opr8,X</i> LDHX <i>opr8,SP</i>	从存储器装入变址寄存器 (H:X)	H:X ← (M:M + 0x0001)	0	-	-	↓	↓	-	IMM DIR EXT IX IX2 IX1 SP1	45 ij 55 dd 32 hh ll 9EAE ee 9EBE ee 9ECE ff ff 9EFE ff	kk ll	3 4 5 5 6 5 5

表 7-2 HCS08 指令设置摘要(5/7)

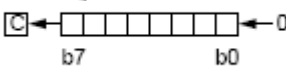
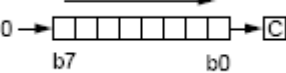
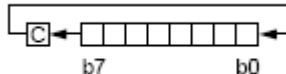
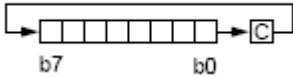
源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
LDX #opr8i LDX opr8a LDX opr16a LDX oprx16,X LDX oprx8,X LDX ,X LDX oprx16,SP LDX oprx8,SP	从存储器中装入X	$X \leftarrow (M)$	0	-	-	↓	↓	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEE	ii dd hh ll ee ff ff ff	2 3 4 4 3 3 5 4
LSL opr8a LSLA LSLX LSL oprx8,X LSL ,X LSL oprx8,SP	逻辑左移 (Same as ASL)		↓	-	-	↓	↓	↓	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
LSR opr8a LSRA LSRX LSR oprx8,X LSR ,X LSR oprx8,SP	逻辑右移		↓	-	-	0	↓	↓	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	5 1 1 5 4 6
MOV opr8a,opr8a MOV opr8a,X+ MOV #opr8i,opr8a MOV ,X+,opr8a	Move	$(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$ $H:X \leftarrow (H:X) + 0x0001$ in IX+/DIR and DIR/IX+ Modes	0	-	-	↓	↓	-	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E 5E 6E 7E	dd dd dd ii dd dd	5 5 4 5
MUL	无符号相乘	$X:A \leftarrow (X) \times (A)$	-	0	-	-	-	0	INH	42		5
NEG opr8a NEGA NEGX NEG oprx8,X NEG ,X NEG oprx8,SP	取负 (Two's Complement)	$M \leftarrow -(M) = 0x00 - (M)$ $A \leftarrow -(A) = 0x00 - (A)$ $X \leftarrow -(X) = 0x00 - (X)$ $M \leftarrow -(M) = 0x00 - (M)$ $M \leftarrow -(M) = 0x00 - (M)$ $M \leftarrow -(M) = 0x00 - (M)$				↓	↓	↓	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	5 1 1 5 4 6
NOP	空操作	Uses 1 Bus Cycle	-	-	-	-	-	-	INH	9D		1
NSA	累加器半字节交换	$A \leftarrow (A[3:0]:A[7:4])$	-	-	-	-	-	-	INH	62		1
ORA #opr8i ORA opr8a ORA opr16a ORA oprx16,X ORA oprx8,X ORA ,X ORA oprx16,SP ORA oprx8,SP	累加器或上存储器	$A \leftarrow (A) (M)$	0	-	-	↓	↓	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA BA CA DA EA FA 9EDA 9EEA	ii dd hh ll ff ff ff ff	2 3 4 4 3 3 5 4
PSHA	把累加器压入堆栈	$\text{Push}(A); SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	87		2
PSHH	把H压入堆栈	$\text{Push}(H); SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	8B		2
PSHX	把X压入堆栈	$\text{Push}(X); SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	89		2
PULA	累加器出栈	$SP \leftarrow (SP + 0x0001); \text{Pull}(A)$	-	-	-	-	-	-	INH	86		3
PULH	H出栈	$SP \leftarrow (SP + 0x0001); \text{Pull}(H)$	-	-	-	-	-	-	INH	8A		3
PULX	X出栈	$SP \leftarrow (SP + 0x0001); \text{Pull}(X)$	-	-	-	-	-	-	INH	88		3
ROL opr8a ROLA ROLX ROL oprx8,X ROL ,X ROL oprx8,SP	进位循环左移		↓	-	-	↓	↓	↓	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	5 1 1 5 4 6

表 7-2 HCS08 指令设置摘要(6/7)

源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	进位循环右移		↓	—	—	↓	↓	↓	DIR INH IX1 IX SP1	36 46 56 66 76 9E66 ff	dd ff	5 1 1 5 4 6
RSP	堆栈复位	SP ← 0xFF (高字节不影响)	—	—	—	—	—	—	INH	9C		1
RTI	中断返回	SP ← (SP) + 0x0001; Pull (CCR) SP ← (SP) + 0x0001; Pull (A) SP ← (SP) + 0x0001; Pull (X) SP ← (SP) + 0x0001; Pull (PCH) SP ← (SP) + 0x0001; Pull (PCL)	↓	↓	↓	↓	↓	↓	INH	80		9
RTS	从子程序返回	SP ← SP + 0x0001; Pull (PCH) SP ← SP + 0x0001; Pull (PCL)	—	—	—	—	—	—	INH	81		6
SBC <i>#opr8i</i> SBC <i>opr8a</i> SBC <i>opr16a</i> SBC <i>opr8,X</i> SBC <i>opr8,X</i> SBC <i>,X</i> SBC <i>opr8,SP</i> SBC <i>opr8,SP</i>	带进位减	$A \leftarrow (A) - (M) - (C)$	↓	—	—	↓	↓	↓	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 B2 C2 D2 E2 F2 9ED2 9EE2 ii dd hh ll ff ff ee ff ff		2 3 4 4 3 3 5 4
SEC	进位位置位	$C \leftarrow 1$	—	—	—	—	—	1	INH	99		1
SEI	中断屏蔽位置位	$I \leftarrow 1$	—	—	1	—	—	—	INH	9B		1
STA <i>opr8a</i> STA <i>opr16a</i> STA <i>opr8,X</i> STA <i>opr8,X</i> STA <i>,X</i> STA <i>opr8,SP</i> STA <i>opr8,SP</i>	将累加器中内容存储到存储器	$M \leftarrow (A)$	0	—	—	↓	↓	—	DIR EXT IX2 IX1 IX SP2 SP1	B7 C7 D7 E7 F7 9ED7 9EE7 dd hh ee ff ff ee ff ff	ll ff	3 4 4 3 2 5 4
STHX <i>opr8a</i> STHX <i>opr16a</i> STHX <i>opr8,SP</i>	存储编制寄存器内容 (Index Reg.)	$(M:M + 0x0001) \leftarrow (H:X)$	0	—	—	↓	↓	—	DIR EXT SP1	35 96 9EFF dd hh ff	ll	4 5 5
STOP	中断使能 停止处理涉及 MCU的文件	$I \text{ bit} \leftarrow 0$; Stop Processing	—	—	0	—	—	—	INH	8E		2+
STX <i>opr8a</i> STX <i>opr16a</i> STX <i>opr8,X</i> STX <i>opr8,X</i> STX <i>,X</i> STX <i>opr8,SP</i> STX <i>opr8,SP</i>	存储X到存储器	$M \leftarrow (X)$	0	—	—	↓	↓	—	DIR EXT IX2 IX1 IX SP2 SP1	BF CF DF EF FF 9EDF 9EEF dd hh ee ff ff ee ff ff	ll ff	3 4 4 3 2 5 4

SUB #opr8i SUB opr8a SUB opr16a SUB oprx16,X SUB oprx8,X SUB ,X SUB oprx16,SP SUB oprx8,SP	减	$A \leftarrow (A) - (M)$	$\uparrow$	—	—	$\uparrow$	$\uparrow$	$\uparrow$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 ii B0 dd C0 hh D0 ll ee E0 ff ff F0 9ED0 ee 9EE0 ff ff	2 3 4 4 3 3 5 4
SWI	软中断	PC $\leftarrow$ (PC) + 0x0001 Push (PCL); SP $\leftarrow$ (SP) - 0x0001 Push (PCH); SP $\leftarrow$ (SP) - 0x0001 Push (X); SP $\leftarrow$ (SP) - 0x0001 Push (A); SP $\leftarrow$ (SP) - 0x0001 Push (CCR); SP $\leftarrow$ (SP) - 0x0001 I $\leftarrow$ 1; PCH $\leftarrow$ Interrupt Vector High Byte PCL $\leftarrow$ Interrupt Vector Low Byte	—	—	1	—	—	—	INH	83	11

表 7-2 HCS08 指令设置摘要(7/7)

源格式	操作	描述	CCR						寻址方式	操作码	操作数	总线周期
			V	H	I	N	Z	C				
TAP	转移累加器到CCR	CCR ← (A)	↓	↓	↓	↓	↓	↓	INH	84		1
TAX	转移累加器到X	X ← (A)	—	—	—	—	—	—	INH	97		1
TPA	转移CCR到累加器	A ← (CCR)	—	—	—	—	—	—	INH	85		1
TST opr8a TSTA TSTX TST oprx8,X TST ,X TST oprx8,SP	测试零或负数	(M) – 0x00 (A) – 0x00 (X) – 0x00 (M) – 0x00 (M) – 0x00 (M) – 0x00	0	—	—	↓	↓	—	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	4 1 1 4 3 5
TSX	转移SP到H:X.	H:X ← (SP) + 0x0001	—	—	—	—	—	—	INH	95		2
TXA	转移X到累加器	A ← (X)	—	—	—	—	—	—	INH	9F		1
TXS	转移H:X到SP	SP ← (H:X) – 0x0001	—	—	—	—	—	—	INH	94		2
WAIT	中断使能，等待中断	I bit ← 0; Halt CPU	—	—	0	—	—	—	INH	8F		2+

总线的时钟频率是 CPU 时钟频率一半。

表 7-3 操作码图（1/2）

Bit-Manipulation	Branc	Read-Modify						Cont	Register/M											
00 5 BRSET0 3 DIR	10 5 BSET0 2 DIR	20 3 BRA 2 REL	30 5 NEG 2 DIR	40 1 NEGA 1 INH	50 1 NEGX 1 INH	60 5 NEG 2 IX1	70 4 NEG 1 IX	80 9 RTI 1 INH	90 3 BGE 2 REL	A0 2 SUB 2 IMM	B0 3 SUB 2 DIR	C0 4 SUB 3 EXT	D0 4 SUB 3 IX2	E0 3 SUB 2 IX1	F0 3 SUB 1 IX					
01 5 BRCLR0 3 DIR	11 5 BCLR0 2 DIR	21 3 BRN 2 REL	31 5 CBEQ 3 DIR	41 4 CBEQA 3 IMM	51 4 CBEQX 3 IMM	61 5 CBEQ 3 IX1+	71 5 CBEQ 2 IX+	81 6 RTS 1 INH	91 3 BLT 2 REL	A1 2 CMP 2 IMM	B1 3 CMP 2 DIR	C1 4 CMP 3 EXT	D1 4 CMP 3 IX2	E1 3 CMP 2 IX1	F1 3 CMP 1 IX					
02 5 BRSET1 3 DIR	12 5 BSET1 2 DIR	22 3 BHI 2 REL	32 5 LDHX 3 EXT	42 5 MUL 1 INH	52 6 DIV 1 INH	62 1 NSA 1 INH	72 1 DAA 1 INH	82 5+ BGND 1 INH	92 3 BGT 2 REL	A2 2 SBC 2 IMM	B2 3 SBC 2 DIR	C2 4 SBC 3 EXT	D2 4 SBC 3 IX2	E2 3 SBC 2 IX1	F2 3 SBC 1 IX					
03 5 BRCLR1 3 DIR	13 5 BCLR1 2 DIR	23 3 BLS 2 REL	33 5 COM 2 DIR	43 1 COMA 1 INH	53 1 COMX 1 INH	63 5 COM 2 IX1	73 4 COM 1 IX	83 11 SWI 2 INH	93 3 BLE 2 REL	A3 2 CPX 2 IMM	B3 3 CPX 2 DIR	C3 4 CPX 3 EXT	D3 4 CPX 3 IX2	E3 3 CPX 2 IX1	F3 3 CPX 1 IX					
04 5 BRSET2 3 DIR	14 5 BSET2 2 DIR	24 3 BCC 2 REL	34 5 LSR 2 DIR	44 1 LSRA 1 INH	54 1 LSRX 1 INH	64 5 LSR 2 IX1	74 4 LSR 1 IX	84 1 TAP 1 INH	94 2 TXS 1 INH	A4 2 AND 2 IMM	B4 3 AND 2 DIR	C4 4 AND 3 EXT	D4 4 AND 3 IX2	E4 3 AND 2 IX1	F4 3 AND 1 IX					
05 5 BRCLR2 3 DIR	15 5 BCLR2 2 DIR	25 3 BCS 2 REL	35 4 STHX 2 DIR	45 3 LDHX 3 IMM	55 4 LDHX 2 DIR	65 3 CPHX 3 IMM	75 5 CPHX 2 DIR	85 1 TPA 1 INH	95 2 TSX 1 INH	A5 2 BIT 2 IMM	B5 3 BIT 2 DIR	C5 4 BIT 3 EXT	D5 4 BIT 3 IX2	E5 3 BIT 2 IX1	F5 3 BIT 1 IX					
06 5 BRSET3 3 DIR	16 5 BSET3 2 DIR	26 3 BNE 2 REL	36 5 ROR 2 DIR	46 1 RORA 1 INH	56 1 RORX 1 INH	66 5 ROR 2 IX1	76 4 ROR 1 IX	86 3 PULA 1 INH	96 5 STHX 3 EXT	A6 2 LDA 2 IMM	B6 3 LDA 2 DIR	C6 4 LDA 3 EXT	D6 4 LDA 3 IX2	E6 3 LDA 2 IX1	F6 3 LDA 1 IX					

07 BRCLR3 3 DIR	17 BCLR3 2 DIR	27 BEQ 2 REL	37 ASR 2 DIR	47 ASRA 1 INH	57 ASRX 1 INH	67 ASR 2 IX1	77 ASR 1 IX	87 PSHA 1 INH	97 TAX 1 INH	A7 AIS 2 IMM	B7 STA 2 DIR	C7 STA 3 EXT	D7 STA 3 IX2	E7 STA 2 IX1	F7 STA 1 IX
08 BRSET4 3 DIR	18 BSET4 2 DIR	28 BHCC 2 REL	38 LSL 2 DIR	48 LSLA 1 INH	58 LSLX 1 INH	68 LSL 2 IX1	78 LSL 1 IX	88 PULX 1 INH	98 CLC 1 INH	A8 EOR 2 IMM	B8 EOR 2 DIR	C8 EOR 3 EXT	D8 EOR 3 IX2	E8 EOR 2 IX1	F8 EOR 1 IX
09 BRCLR4 3 DIR	19 BCLR4 2 DIR	29 BHCS 2 REL	39 ROL 2 DIR	49 ROLA 1 INH	59 ROLX 1 INH	69 ROL 2 IX1	79 ROL 1 IX	89 PSHX 1 INH	99 SEC 1 INH	A9 ADC 2 IMM	B9 ADC 2 DIR	C9 ADC 3 EXT	D9 ADC 3 IX2	E9 ADC 2 IX1	F9 ADC 1 IX
0A BRSET5 3 DIR	1A BSET5 2 DIR	2A BPL 2 REL	3A DEC 2 DIR	4A DECA 1 INH	5A DECX 1 INH	6A DEC 2 IX1	7A DEC 1 IX	8A PULH 1 INH	9A CLI 1 INH	AA ORA 2 IMM	BA ORA 2 DIR	CA ORA 3 EXT	DA ORA 3 IX2	EA ORA 2 IX1	FA ORA 1 IX
0B BRCLR5 3 DIR	1B BCLR5 2 DIR	2B BMI 2 REL	3B DBNZ 3 DIR	4B DBNZ 2 INH	5B DBNZX 2 INH	6B DBNZ 3 IX1	7B DBNZ 2 IX	8B PSHH 1 INH	9B SEI 1 INH	AB ADD 2 IMM	BB ADD 2 DIR	CB ADD 3 EXT	DB ADD 3 IX2	EB ADD 2 IX1	FB ADD 1 IX
0C BRSET6 3 DIR	1C BSET6 2 DIR	2C BMC 2 REL	3C INC 2 DIR	4C INCA 1 INH	5C INCX 1 INH	6C INC 2 IX1	7C INC 1 IX	8C CLRH 1 INH	9C RSP 1 INH		BC JMP 2 DIR	CC JMP 3 EXT	DC JMP 3 IX2	EC JMP 2 IX1	FC JMP 1 IX
0D BRCLR6 3 DIR	1D BCLR6 2 DIR	2D BMS 2 REL	3D TST 2 DIR	4D TSTA 1 INH	5D TSTX 1 INH	6D TST 2 IX1	7D TST 1 IX		9D NOP 1 INH	AD BSR 2 REL	BD JSR 2 DIR	CD JSR 3 EXT	DD JSR 3 IX2	ED JSR 2 IX1	FD JSR 1 IX
0E BRSET7 3 DIR	1E BSET7 2 DIR	2E BI 2 L	3E CPHX 3 EXT	4E MOV 3 DD	5E MOV 2	6E MOV 3 IMD	7E MOV 2	8E STOP 1 INH	9E Page 2	AE LDX 2 IMM	BE LDX 2 DIR	CE LDX 3 EXT	DE LDX 3 IX2	EE LDX 2 IX1	FE LDX 1 IX
0F BRCLR7 3 DIR	1F BCLR7 2 DIR	2F BIH 2 REL	3F CLR 2 DIR	4F CLRA 1 INH	5F CLR 1 INH	6F CLR 2 IX1	7F CLR 1 IX	8F WAIT 1 INH	9F TXA 1 INH	AF AIX 2 IMM	BF STX 2 DIR	CF STX 3 EXT	DF STX 3 IX2	EF STX 2 IX1	FF STX 1 IX

INH 隐含寻址

IMM 立即寻址

DIR 直接寻址

EXT 扩展寻址

DD 直接-直接寻址

IX+D 变址-直接-变址加
1寻址

INH 隐含寻址

IMM 立即寻址

DIR 直接寻址

EXT 扩展寻址

DD 直接-直接寻址

IX+D 变址-直接-变址加
1寻址

INH 隐含寻址

IMM 立即寻址

DIR 直接寻址

EXT 扩展寻址

DD 直接-直接寻址

IX+D 变址-直接-变址加
1寻址

用16进制表示操作码 F0 3 HCS08指令周期

字节数 1 SUB IX 寻址方式

表 7-3 操作码图（1/2）

Bit-Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
						9E60 6 NEG							9ED0 5 SUB	9EE0 4 SUB	
						9E61 6 CBEQ							9ED1 5 CMP	9EE1 4 CMP	
													9ED2 5 SBC	9EE2 4 SBC	
						9E63 6 COM							9ED3 5 CPX	9EE3 4 CPX	9EF3 6 CPHX SP1
						9E64 6 LSR							9ED4 5 AND	9EE4 4 AND	
													9ED5 5 BIT	9EE5 4 BIT	
						9E66 6 ROR							9ED6 5 LDA	9EE6 4 LDA	
						9E67 6 ASR							9ED7 5 STA	9EE7 4 STA	
						9E68 6 LSL							9ED8 5 EOR	9EE8 4 EOR	
						9E69 6 ROL							9ED9 5 ADC	9EE9 4 ADC	
						9E6A 6 DEC							9EDA 5 ORA	9EEA 4 ORA	
						9E6B 8 DBNZ							9EDB 5 ADD	9EEB 4 ADD	

						9E6C 6 INC										
						9E6D 5 TST										
										9EAE 5 LDHX	9EBE 6 LDHX	9ECE 5 LDHX	9EDE 5 LDX	9EEE 4 LDX	9EFE 3 LDHX SP1	5
						9E6F 6 CLR							9EDF 5 STX	9EEF 4 STX	9EFF 3 STHX SP1	5

INH 隐含寻址
IMM 立即寻址
DIR 直接寻址
EXT 扩展寻址
DD 直接-直接寻址
IX+D 变址-直接-变址加
1寻址

INH 隐含寻址
IMM 立即寻址
DIR 直接寻址
EXT 扩展寻址
DD 直接-直接寻址
IX+D 变址-直接-变址加
1寻址

INH 隐含寻址
IMM 立即寻址
DIR 直接寻址
EXT 扩展寻址
DD 直接-直接寻址
IX+D 变址-直接-变址加
1寻址

用16进制表示操作码9E60 6 HCS08指令周期
字节数3 SP1 寻址方式

第八章 内部时钟发生器(S08ICGV4)

内部时钟发生器模块（ICG）可以用来为 MC9S08AC16 系列单片机生成系统时钟。该模拟供电线路 V_{DD} 和 V_{SS} 源自 MCU 的 V_{DD} 和 V_{SS} 的引脚。ICG 的电气参数可以在附录 A 电气特性和定时说明中查询。

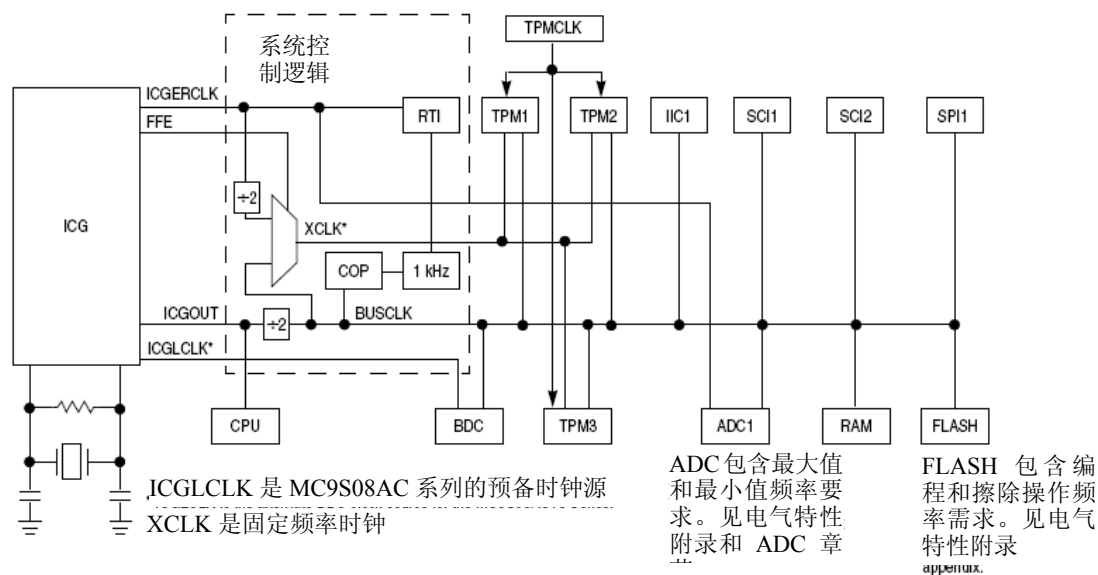
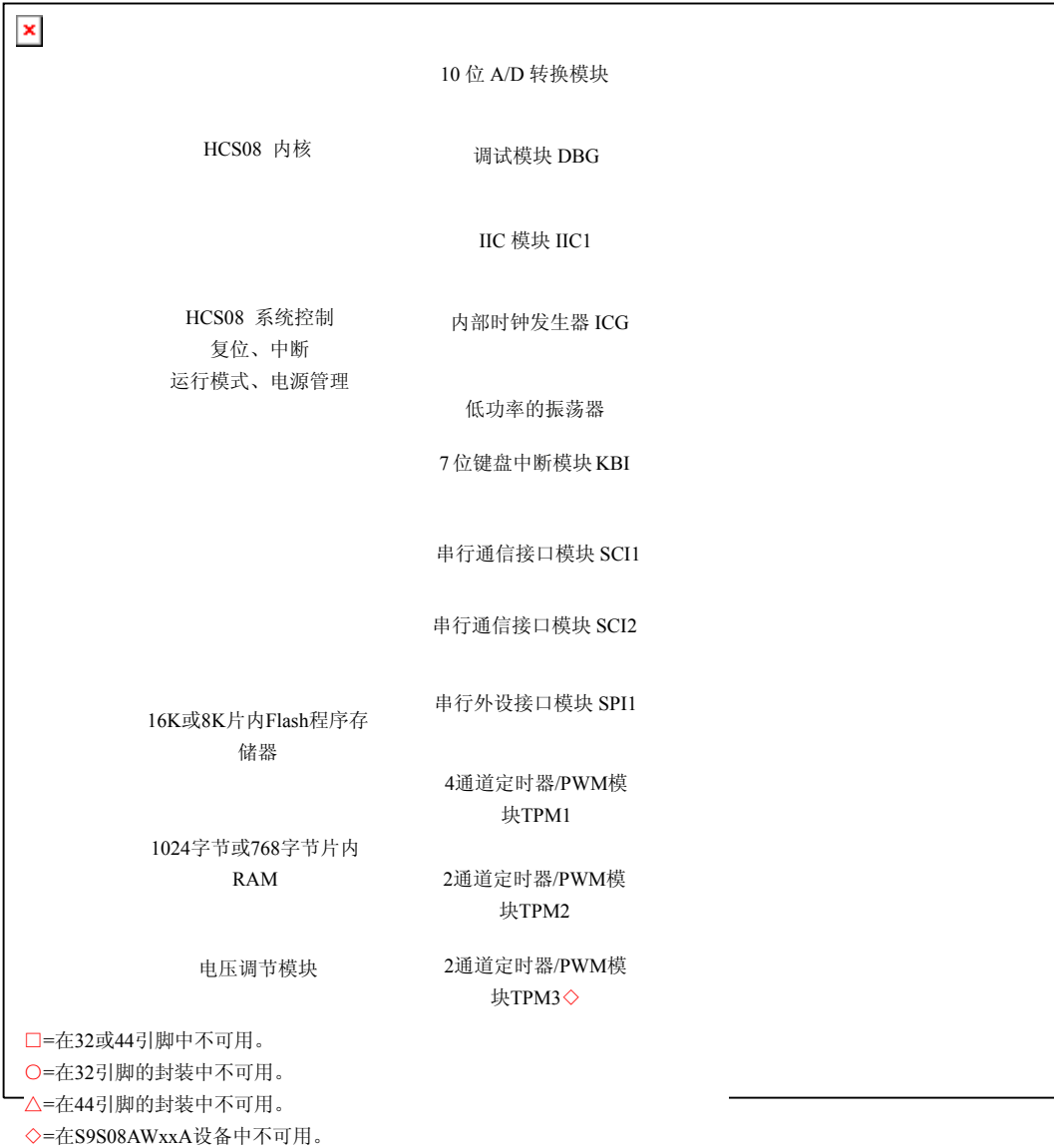


图8-1系统时钟分配图

注：

Freescall 半导体公司建议闪存的位置 0xFFBE 被预留存放非易失性 ICGTRM。这将允许调试器和编程器厂商进行手动调整操作及储存合成的 ICGTRM 值，为用户在稍后时间存取做准备。



注意：

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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测（IRQEDG=1），下拉使能。
3. IRQ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
4. 引脚包含集成的上拉设备。
5. PTD3,PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能（KBIPE_n=1）而且上升沿被选择（KBEDG_n=1），下拉使能。

图8-1 MC9S08AC16框图

8.1 介绍

ICG 为时钟源提供了多种选择。当在成本，精度，电流消耗量，以及绩效中作选择时。这提供给用户极大的灵活性，由于看到在图 8 月 3 日，国际协调小组的组成 4 个功能模块。如图 8-3 所示，ICG 由四个功能模块组成。下面将简要的进行描述，后面章节将再进行详细描述。

- 振荡器模块 — 振荡器提供连接外部晶振或共鸣器的方法。软件选择的两个频率范

围能够获得最佳的启动和稳定性。其中振荡器模块可以用来提供一个外部方波给系统时钟。外部时钟源可以提供一个非常精确的时钟。由 HGO 选择的振荡器能够为低功耗模式或高振模式配置时钟。

- 内部辅助发生器 — 内部辅助发生器由两个受控时钟源组成。一个被设计为大约 8MHz 作为背景调试控制器的本地时钟。其它的内部辅助发生器源是典型的 243kHz，当一个精确的定时事件输入到 MCU 时可通过软件进行平衡获得很好的精确度。这将提供一个非常可靠，低耗的时钟源。
- 频率锁定回路 — 一个频率锁定回路过程获取内部或外部时钟源并将它超到高频。当电路达到阈值或低于阈值，状态位将提供标志信息。另外这个模块可以监控外部参考时钟及时钟是否有效的信号。
- 时钟选择模块 — 时钟选择模块为不同时钟源连接到系统时钟树提供几种开关选择。ICGDCLK 是 FLI 之外的增值时钟频率，ICGDCLK 是源自晶振或外部时钟源的参考时钟频率，FFE 是一个控制信号用来控制系统固定频率时钟（XCLK）。ICGDCLK 是背景调试控制器的时钟源。

8.1.1 特征

该模块准备用无需用户干预的一些自动特征来方便用户使用。快速配置此模块请参考 8.5 “初始化/应用信息” 和选取一个非常适合此应用需要的例子。

ICG 的特征和时钟分配系统：

- 几种主要时钟源的允许在一定范围内对成本，频率和精确度选择。
 - 32KHz-100KHz 的晶振或共鸣器
 - 1MHz-16MHz 的晶振或共鸣器
 - 外部时钟
 - 内部参考发生器
- 默认为自时钟模块来最小化启动延时
- 频率锁定回路（FLI）产生 8MHz-40MHz（总线上升至 20 MHz）— 用外部或内部时钟作参考频率。
 - 无运行时钟源自动停止。
- 丢失时钟所引起的复位或中断或 FLI 时钟的丢失
- 数字控制振荡器（DCO）保留了先前的频率设定，当从 stop3 模式恢复时允许快速频率锁定。
- 在丢失或移除参考时钟时，DCO 将维持操作频率。
- 快速 FLL 分频器选择 8 总线速度约数中之一。
- 实时中断的独立自时钟源。
- 可调整的内部时钟源支持串行通讯，无需额外的外部部件。
- 锁请求后自动 FLL 接触。
- 外部振荡器可选择低功率或高增益

8.1.2 操作方式

这仅是的概要描述。操作方式的详细描述在 8.4 节，“功能描述”

- 模式 1 — OFF

输出时钟，ICGOUT 是静态的，当 STOP 指令被执行时将进入此模式。

- 模式 2 — 自时钟（SCM）

默认模式的操作是在复位后立即进入此模式。内部时钟发生器的 FLL 是开环的，数字控制振荡器是自由运行在由滤波器比特设置的频率上。

● 模式 3 — 内部使用的 FLL

在此模式下，内部时钟的 FLL 用来产生频率且是可调整的，作为多个内部参考时钟。

- 当 FLL 试图上锁时，未上锁的 FEL 是一个跳变状态。FLL 数字控制振荡器频率偏离目标频率同时 FLL 调整数字控制振荡器以匹配目标频率。
- 上锁的 FEL 是一种状态，它发生在 FLL 检测数字控制振荡器被上锁作为一个多重内部参考时。

● 模式 4—FLL 外部旁路

在这种模式下，ICG 被配置绕过 FLL 并使用一个外部时钟作为其时钟源。

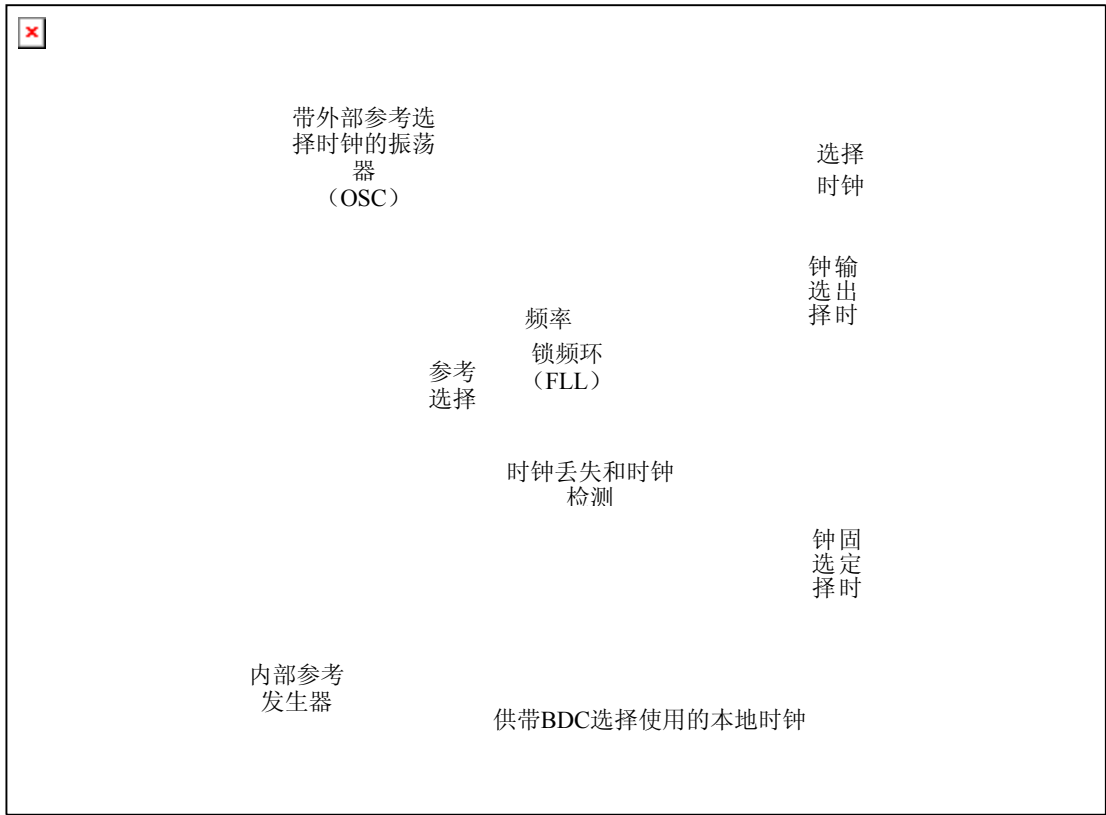
● 模式 5—外部使用的 FLL（FEE）

内部时钟发生器的 FLL 被用来产生频率，该频率可多次调整作为外部时钟参考。

- 未上锁的外部使用的 FLL 是一种跳变状态，它发生在 FLL 试图上锁时。FLL 数字控制振荡器频率偏离目标频率同时 FLL 调整数字控制振荡器以匹配目标频率。
- 上锁的 FEL 是一种状态，它发生在 FLL 检测数字控制振荡器被上锁作为一个多重内部参考时。

8.1.3 框图

图 8-3 是一个简略图用来显示内部时钟发生器（ICG）模块的功能性组织。包括一个摘要和特征清单。



注：

1.ICGOUT、FFE、ICGCLK、ICGERCLK 具体使用方法参阅 8.2 节“外部信号描述”

2.不是所有的 HCS08 芯片都含有唯一的引脚给 ICG 使用。参阅设备引脚分配。

图 8-3 ICG 框图

8.2 外部信号描述

该振荡器引脚用来提供一个外部时钟源给 MCU。振荡器引脚在低功耗模式下（默认）受抑制。在低功耗模式下振荡器振幅限制接近 1V（峰值与峰值之间）。

8.2.1 EXTAL—外部参考时钟/振荡器输入

若是第一次写 ICGC1，无论是 FEE 模式或 FBE 模式被选择，该引脚将作为外部时钟输入或由 REFS 决定的振荡器电路。若是第一次写 ICGC1，无论 FEI 模式或 SCM 模式被选择，该引脚将不被 ICG 使用。

8.2.2 XTAL—振荡器输出

若是第一次写 ICGC1，无论是 FEE 模式或 FBE 模式被选择，该引脚将被作为振荡器电路输出。若是第一次写 ICGC1，无论 FEI 模式或 SCM 模式被选择，该引脚将不被 ICG 使用。振荡器能够被配置提供更高振幅输出，降低噪音影响。这种操作模式通过置 HGO=1 来选择。

8.2.3 外部时钟连接

若一个外部时钟被使用，则该引脚连接如图 8-4 所示。

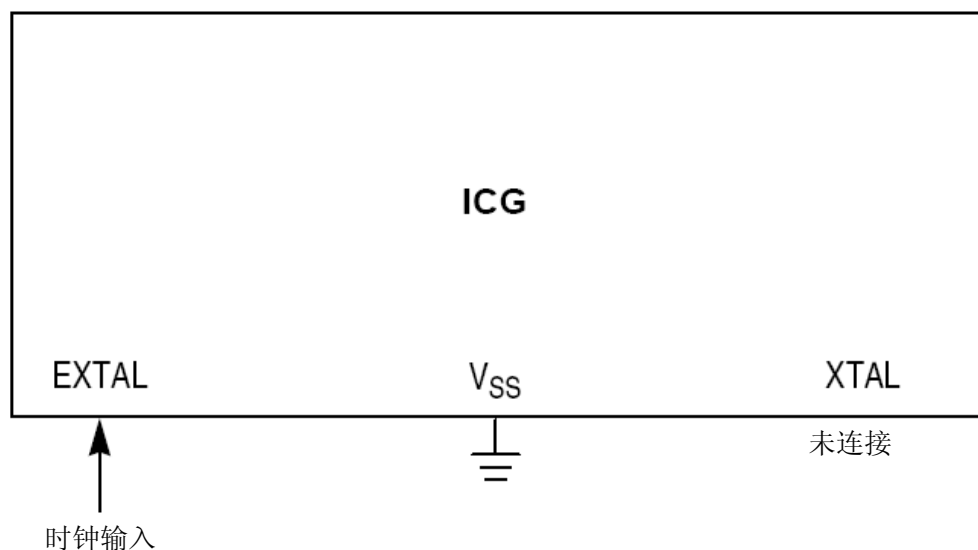


图 8-4 外部时钟连接

8.2.4 外部晶振/共鸣器连接

如果一个外部晶振/共鸣器被使用，那么该引脚连接如图 8-5 所示。所推荐的元件值被列在“电气特性”章节。

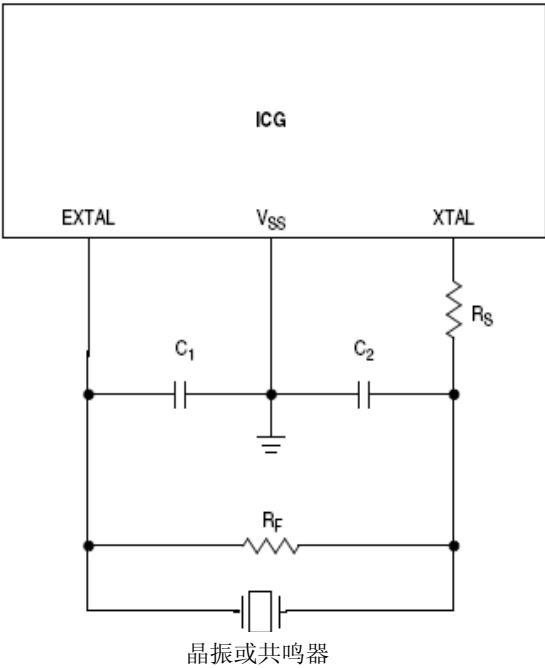


图 8-5 外部参考频率连接

8.3寄存器定义

参阅存储器章节所有 ICG 寄存器绝对地址分配数据表。这部分通过名称检索寄存器和控制位。飞思卡尔提供一个头文件用来把名称转换为相应的绝对地址。

8.3.1 ICG控制寄存器1（ICGC1）

	7	6	5	4	3	2	1	0
读	HGO ¹	RANGE	REFS	CLKS		OSCSTEN	LOCD	0
写								
复位	0	1	0	0	0	1	0	0

 =未定义

图 8-6 ICG 控制寄存器 1（ICGC1）

¹ 该位仅在复位后可写。其它写操作被忽略。

表 8-1 ICG1 寄存器位描述

位	描述
7 HGO	高增益振荡器的选择位—该位用来在低功耗操作和高增益操作之间作选择以降低噪声影响。该位复位后可写。 0 振荡器配置为低功耗操作 1 振荡器配置为高增益操作
6 RANGE	频率范围选择位—该位控制振荡器，参考分频器，FLL环路分频器倍频因子（P）。它为ICG选择两个频率范围中的其中一个。该位复位后可写。该位仅在使用FLL和FLL外部旁路模式有用。 0 振荡器配置为低频率范围。P=64 1 振荡器配置为高频率范围。P=1
5	外部参考选择—该位控制外部参考时钟源用于ICGERCLK。该位复位后可写。

REFS	0 外部时钟请求 1 振荡器需使用晶振或共鸣器请求
4:3 CLKS	时钟模式选择—该位控制如下描述的时钟模式。如果有FLL外部旁路请求，该位将不被选择直到ERCS=1.如果ICG进入关模式，该位将保持不变。如果一个先前写未完成则写CLKS将无效。 00 自时钟 01 内部参考使用的FLL 10 外部参考旁路FLL 11 外部参考使用的FLL CLKS位在任何时间都可写，除非在一个复位后的第一次写为CLKS=0X，CLKS位不能被写为1X直到下个复位操作之后。
2 OSCSTEN	在关模式下振荡器使能位—当ICG进入关模式时该位控制振荡器电路是否保持使能。如果HGO=1、RANGE=1该位无效。 0 当ICG处在关模式下时振荡器禁止，除非使能位为高，CLKS=10、REFST=1。 1 当ICG处于关模式，CLKS=1X、REFST=1时振荡器允许。
1 LOCD	时钟禁用丢失 0 时钟检测丢失允许 1 时钟检测丢失禁止

8.3.2 ICG控制寄存器2（ICGC2）

	7	6	5	4	3	2	1	0
读	L	M	M	M	L	R	R	R
写	L	M	M	M	L	R	R	R
复位	0	0	0	0	0	0	0	0

图 8-7 ICG 控制寄存器 2（ICGC2）

表 8-2 ICG2 寄存器位描述

位	描述
7 LOLRE	锁复位使能丢失—在锁指示丢失后该位决定由ICG发出何种类型请求。当LOLS被置位时该位有效。 0 基于锁丢失产生一个中断请求 1 基于锁丢失产生一个复位请求
6:4	倍频因子—该位控制环路FLL中可变倍频因子。由MFD位指定的值确定应用于参考频率的倍频因子（N）。如果先前的一个写操作未完成，则写MFD位不产生影响。为N选择一个充分低的值这样 $f_{ICGDCLK}$ 就不会超出它的最大值。 000 倍频因子=4 001 倍频因子=6 010 倍频因子=8 011 倍频因子=10 100 倍频因子=12 101 倍频因子=14 110 倍频因子=16 111 倍频因子=18
3 LOCRE	时钟复位使能丢失—该位决定系统如何管理一个时钟条件丢失。 0 基于时钟丢失产生一个中断请求 1 基于时钟丢失产生一个复位请求
2:0 RFD	减少分频器—该位在时钟选择电路后控制分频器值。由RFD位指定的值确定分频因子用于选作为输出的时钟源。如果先前的一个写操作未完成，则写RFD位不产生影响。 000 分频因子=1 001 分频因子=2 010 分频因子=4 011 分频因子=8 100 分频因子=16 101 分频因子=32

	110 分频因子=64
	111 分频因子=128

8.3.3 ICG状态寄存器1（ICGS1）

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

 =未定义

图 8-8 ICG 状态寄存器 1（ICGS1）

表 8-3 ICGS1 寄存器位描述

位	描述
7:6 CLKST	时钟模式状态位—该位表明当前时钟模式。在写CLKS位后由于在时钟域之间的内部同步CLKS位不会立即更新。 00 自时钟 01 内部参考使用的FLL 10 外部参考旁路FLL 11 外部参考使用的FLL
5 REFST	参考时钟状态—该位表明那一个时钟通过参考选择电路在当前被选择。 0 外部时钟别选择 1 晶振/共鸣器被选择
4 LOLS	锁状态丢失FLL—该位是FLL锁状态的一个关联指示。 0 自从LOCS最后被清零FLL并未意外丢失锁。 1 自从LOCS最后被清零FLL并未意外丢失锁，LOLRE决定采取行动。自从LOCS最后被清零FLL意外丢失锁，LOLRE决定采取行动。
3 LOCK	FLL锁状态—该位表明FLL是否已获得锁。在关，自时钟，FLL旁路模式下该位被清零。 0 FLL当前未锁 1 FLL当前被锁
2 LOCS	时钟状态丢失—该位是ICG时钟状态丢失的一个指示。 0自从LOCS最后被清零ICG并未丢失时钟。 1自从LOCS最后被清零ICG丢失时钟，LOCRE决定采取行动。
1 ERCS	外部参考时钟状态—该位表示外部参考时钟（ICGERCLK）是否满足最小频率要求。 0 外部参考时钟不稳定，频率要求未满足 1 外部参考时钟稳定，频率要求满足
0 ICGIF	ICG中断标志—当一个ICG中断请求中断请求未决时ICGIF读/写标志被置位。当ICGIF被置位时可通过复位或读取ICG状态寄存器来清零它，然后给它写逻辑1.如果在清零序列完成之前另一个ICG中断发生，该序列被复位因此在清零较早中断的序列后ICGIF将保持置位。写逻辑0给ICGIF无影响。 0 无ICG中断请求未决 1 一个ICG中断请求未决

8.3.4 ICG状态寄存器2（ICGS2）

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

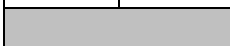
 =未定义

图 8-9 ICG 状态寄存器 2（ICGS2）

表 8-4 ICGS2 寄存器位描述

位	描述
0 DCOS	数字控制振荡器时钟稳定—当DCO时钟稳定时DCOS位被置位，表示计数错误没被超过两次连续采样的 n_{unlock} 所改变，该DCO时钟不是静止的。当退出关状态如果CLKS=X1来决定何时转换到请求时钟模式时该位被使用。它同时也被用在自时钟模式来决定何时开始监视DCO时钟。进入关状态该位被清零。 0 DCO时钟不稳定 1 DCO时钟稳定

8.3.5 ICG滤波器寄存器（ICGFLTU、ICGFLTL）

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

 =未定义

图 8-10 ICG 滤波寄存器高八位（ICGFLTU）

表 8-5 ICGFLTU 寄存器位描述

位	描述
3:0 FLT	滤波器值—该位表明当前滤波器值，该值控制DCO频率。除了在CLKS位被改变到自时钟模式时（CLKS=00）该位可读。在自时钟模式下任何写ICGFLTU更新当前12位滤波器值。如果一个先前的关锁序列未完成，写ICGFLTU寄存器不会影响FLT。

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

图 8-11 ICG 滤波寄存器低八位（ICGFLTL）

表 8-6 ICGFLTL 寄存器位描述

位	描述
7:0 FLT	滤波器值—该位表明当前滤波器值，该值控制DCO频率。除了在CLKS位被改变到自时钟模式时（CLKS=00）该位可读。在自时钟模式下任何写ICGFLTU更新当前12位滤波器值。如果一个先前的关锁序列未完成，写ICGFLTU寄存器不会影响FLT。滤波器寄存器显示滤波器值（FLT）

8.3.6 ICG调整寄存器（ICGTRM）

	7	6	5	4	3	2	1	0
读	TRIM							
写								
POR	1	0	0	0	0	0	0	0

Reset	U	U	U	U	U	U	U	U
-------	---	---	---	---	---	---	---	---

U=不受 MCU 复位影响

图 8-12 ICG 调整寄存器（ICGTRM）

表 8-7 ICGTRM 寄存器位描述

位	描述
7 TRIM	ICG调整设置—该位控制内部参考发生器频率。允许POR周期有25%浮动。该位在该周期上的影响是对二进制加权（位1将调整两次）。在调整时增加着的二进制值将会增长该周期，减少着的值将缩短该周期

8.4功能性描述

这部分将对 ICG 五个工作模式作功能性描述。同时讨论时钟丢失和锁丢失错误及进入每个模式的必要条件。ICG 非常具有灵活性，在某些配置下他可能超越某个时钟规定。当使用 FLL 时，配置该 ICG 以便 ICGDCLK 的频率不会超出它的最大值以确保 MCU 正常工作。

8.4.1关模式

通常情况下当 CPU 进入停止模式时，ICG 将停止所有时钟并进入关状态。不论如何当时钟继续活跃时而 CPU 处于停止模式下时有两个方案可供参考。

8.4.1.1活跃BDM

当 BDM 允许时，ICG 继续像起初设置那样运行。这将允许通过 BDC 控制器存取存储器和控制寄存器。

8.4.1.2 OSCSTEN位设置

当在停止模式下振荡器允许时（OSCSTEN=1），该单独的时钟发生器被允许但给 MCU 复位使用的时钟被关闭。这种选择避免了冗长的振荡器启动时间，或在停止模式 3 下从振荡器运行 RTI 指令。

8.4.1.3停止/关模式恢复

若由于一个中断 CPU 退出停止模式，先前设置的控制位有效且系统时钟重新恢复。如果 FEE 被选择，ICG 将以内部参考时钟为源时钟直到外部时钟稳定。如果 FBE 被选择，在使能 ICGOUT 之前 ICG 将等待外部时钟稳定。

若由于一个复位操作 CPU 退出停止模式，先前设置的 ICG 控制位将被忽略，默认的复位值被使用。因此 ICG 将退出 t 停止模式在 SCM 模式下将被配置为一个大约 8MHz 带调整值的输出。如果使用晶振，在使用 ICGERCLK 之前 4096 个时钟已被检测到。这包含在晶振启动时间里。

8.4.2自时钟模式（SCM）

自时钟模式是默认的操作模式当一下条件中任何一个发生时就将进入 SCM 模式：复位后。

当 CLKKS 不等于 10 时从关模式退出。如果 CLKKS=X1，ICG 将短暂进入该状态直到 DCO 稳定（DCOS=1）。

CLKS 位被写从 X1 到 00

CLKS=1X 且 ICGERCLK 未被检测到（ERCS=0、LOCS=1）。

在这种状态下，FLL 环路是敞开的。DCO 正在运行，其输出时钟信号 ICGOUT 频率由 $f_{ICGDCLK} / R$ 给出。通过写一个新值到滤波器寄存器（ICGFLTH 和 ICGFLTL）该 ICGDCLK 频率在 8MHz 到 40MHz 之间可选。仅在此模式下滤波器寄存器可写。

如果由于复位进入此模式， $f_{ICGDCLK}$ 将默认为 8MHz。如果从内部使用 FLL 进入此模式， $f_{ICGDCLK}$ 将保持先前的频率，但如果 FLL 处于未上锁状态 ICGOUT 将变为其两倍。如果从关模式进入此模式，在进入关模式前 $f_{ICGDCLK}$ 将等于 ICGDCLK 的频率。如果在退出关模式时 CLKS 位被置为 01 或 11，那么直到由 DCOS 位决定的 ICGDCLK 稳定 ICG 才进入自时钟模式。在 ICGDCLK 被认为稳定后，当被 CLKS 位选中时通过切换到 FLL（内部或外部使用），ICG 自动关闭环路。

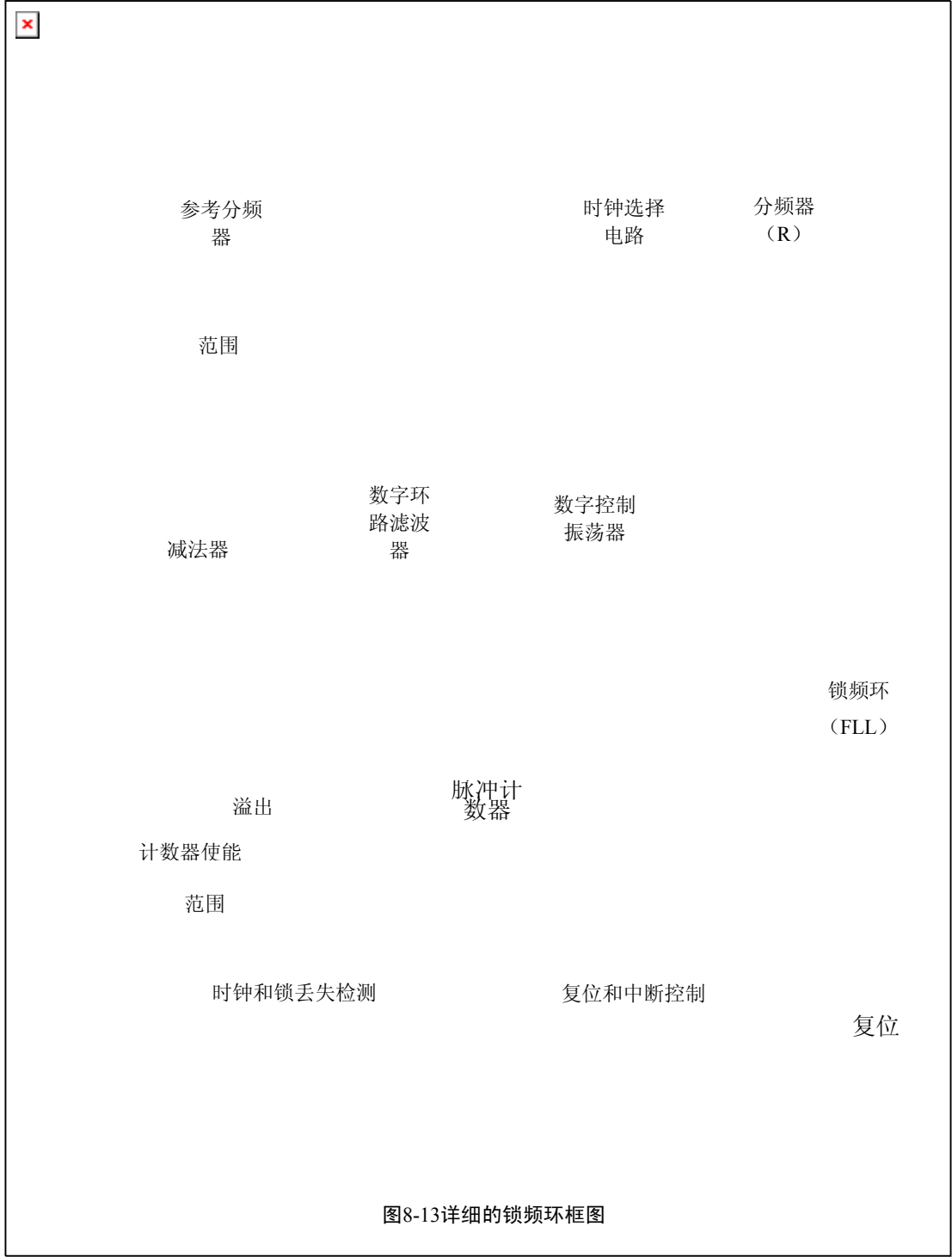


图8-13详细的锁频环框图

8.4.3 FLL内部时钟模式（FEL）

当下列条件发生时进入 FEL 模式：

- CLKS 位被写为 01
- 当退出 CLKS=01 的关状态但处在自时钟模式下该 DCO 时钟稳定（DCOS=1）

在 FEL 模式下，参考时钟源自内部参考时钟 ICGIRCLK，该 FLL 环路试图锁定 ICGDCLK 频率以达到预期值。

8.4.4 FLL内部失锁

当进入 FEL 模式时进入短暂的 FEL 敞开状态，减法器的计数误差大于 n_{unlock} 最大值或小于 n_{unlock} 最小值就像锁检测器检测未上锁条件一样。

ICG 将保持这种状态，而减法器的计数误差大于 n_{unlock} 最大值或小于 n_{unlock} 最小值就像锁检测器检测未上锁条件一样。

在这种状态该输出时钟信号 ICGOUT 频率由 f_{ICGDCLK}/R 给出。

8.4.5 FLL内部锁定

当源自减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值对于给定的样本数目来说，就像锁检测器检测未上锁条件一样。该输出时钟信号 ICGOUT 频率由 f_{ICGDCLK}/R 给出。在上锁的 FEL 模式下，仅当在每四个比较周期时该滤波器值被更新。更新的是前四次比较取样误差值的平均值。

8.4.6 FLL旁路外部时钟模式（FBE）

当下列条件发生时进入 FLL 旁路外部时钟模式：

- 当 CLKS=10 且 ERCS 为高时从 SCM 进入
- 当 CLKS=10 时，ERCS=1 进入关模式，然后退出关模式
- 如果一个 DCO 时钟丢失发生同时外部时钟参考保持稳定（LOCS=1、ERCS=1）从 FLL 嵌入外部时钟模式进入

在这种状态下，DCO 和 ICG 关闭参考时钟使用外部参考时钟 ICGERCLK。该输出时钟信号 ICGOUT 频率由 f_{ICGDCLK}/R 给出。如果一个外部时钟源被使用（REFS），那么 EXTAL 引脚上的输入频率可在 0MHz 到 40MHz 范围内任选。如果一个晶振或共鸣器被使用，那么频率范围或为低 RANGE=0 或为高 RANGE=1。

8.4.7 FLL外部时钟模式（FEE）

当下列条件发生时进入 FEE 模式：

- CLKS=11 同时 ERCS 和 DCOS 都为高
- 当退出 CLKS=11 的关状态但处在自时钟模式下该 DCO 时钟稳定（DCOS=1）

在 FEE 模式下，参考时钟使用外部参考时钟 ICGERCLK，同时 FLL 环路试图锁定 ICGDCLK 频率以达到预期值。为了在 FEE 模式下运行，这里必须存在一个工作的 32KHz-100 KHz 或 2 MHz-10 MHz 外部时钟源。在 FEE 模式下外部时钟频率的最大值被限制在 10 MHz 以防止 DCO 溢出。FLL 的最小增加值从表 8-12 中可知是 4。因为 40 从表 8-12MHz 是 DCO 的运行的极限，参考时钟不能超过 10MHz。

8.4.7.1 FLL外部失锁

当进入 FEE 模式同时源自减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值对于给定的样本数目来说，就像锁检测器检测未上锁条件一样时，进入失锁的 FEE 模式。

在这种状态下，该脉冲计数器，减法器，数字环路滤波器，和 DCO 形成一个闭合环路并依据稍后章节的操作描述锁定它。在进入此状态和直到 FLL 锁定基础上，该输出时钟信号 ICGOUT 频率由 $f_{\text{ICGDCLK}} / (2 \cdot R)$ 给出。在 FLL 被锁定后，如果一个意外的时钟丢失导致它重新进入开锁的状态而 ICG 任然保持在 FEE 模式，该输出时钟信号 ICGOUT 频率由 f_{ICGDCLK} / R 给出。更新的是前四次比较取样误差值的平均值。

8.4.7.2 FLL外部锁定

当减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值对于给定的样本数目来说，就像锁检测器检测未上锁条件一样时，从 FEE 开锁模式进入 FEE 上锁模式。该输出时钟信号 ICGOUT 频率由 f_{ICGDCLK} / R 给出。在 FLL 外部上锁模式下，仅当每过四个比较周期时滤波器值被更新。

8.4.8 FLL锁和锁丢失检测

为决定 FLL 锁定和锁丢失条件，该脉冲计数器计数 DCO 一个比较周期的脉冲（见表 8-9 比较周期的解释）并把所得数传给减法器。减法器把它与 MFD 中的值做比较，产生一个计数误差， Δn 。为达到锁定状态， Δn 必须在 n_{unlock} 最大值与最小值之间。在 FLL 锁定后， Δn 必须处在 n_{unlock} 最大值与最小值之间以保持锁定。如果 Δn 意外溢出此范围，LOLS 状态位被置位并保持置位直到通过软件清零或直到 MCU 复位。通过读 ICGS1 清零 LOLS 位然后写 1 给 ICGIF（LOLRE=0），或通过一个时钟丢失引起复位，或通过任意 MCU 复位。

当 ENBDM=OSCSTEN=0 时如果 ICG 因停止模式进入关状态，则 FLL 丢失锁定状态（LOCK 被清零），但 LOLS 保持不变因为这是一个意外的时钟丢失条件。如果 ENBDM 被清为 0 而 MCU 处在停止模式下，ICG 进入关状态，尽管这不同寻常。因为这是一个意外的时钟停止，当 MCU 从停止模式唤醒时 LOLS 将被置位。

当 MFD 或 CLKS 位被改变或在 FEI 模式下时，当 TRIM 位被改变时，容易发生锁丢失。在这些情况下，LOCK 位将被清零直到 FLL 恢复锁定，但 LOLS 将不被置位。

8.4.9 FLL时钟丢失检测

参考时钟和 DCO 时钟在不同的条件下被监控（见表 8-8）。若参考频率正被监控，ERCS=1 表明参考时钟达到了最小频率值要求。当参考时钟和/或 DCO 时钟正被监控时，如果它们中的任一个下降到某一频率下， f_{LOR} 和 f_{LOD} ，LOLS 状态位将被置位以表明此错误。LOCS 将保持置位直到它被识别或直到 MCU 复位。通过读 ICGS1 可清零 LOCS 然后写 1 给 ICGIF（LOCRES=0），或通过一个时钟丢失引起复位，或通过任意 MCU 复位。

如果 ICG 处于 FEE 模式，一个参考时钟的丢失将导致 ICG 进入 SCM 模式，一个 DCO 时钟丢失将导致 ICG 进入 FBE 模式。如果 ICG 处于 FBE 模式下，一个参考时钟的丢失将导致 ICG 进入 SCM 模式。在每种情况下，CLKST 和 CLKS 位将被自动改变以反映新状态。当一个时钟丢失发生时如果 ICG 处于 FEE 模式同时 ERCS 仍为 1，那么 CLKST 位被置为 1 同时 ICG 恢复到 FBE 模式。

当处于 FEE 或 FEI 模式时一个时钟丢失也将导致一个锁丢失。因为清 LOCS 和 LOLS 位的方法相同，在几乎不可能发生的情况下 LOLRE=1 和 LOCRES=0 这将是一个问题。在这种情形下，通过锁丢失引起的复位中断将被忽略。

表 8-8 时钟监控（当 LOCD=0 时）

模式	CLKS	REFST	ERCS	外部参考时钟监控？	DCO时钟监控？
关模式	0X或11	X	强制低	无	无
	10	0	强制低	无	无
	10	1	实时时间 ¹	有 ¹	无
SCM模式 (CLKST=00)	0X	X	强制低	无	有 ²
	10	0	强制高	无	有 ²
	10	1	实时时间	有	有 ²
	11	X ¹	实时时间	有	有 ²
FEI (CLKST=01)	0X	X	强制低	无	有
	11	X	实时时间	有	有
FBE (CLKST=10)	10	0	强制高	无	无
	10	1	实时时间	有	无
FEE (CLKST=11)	11	X	实时时间	有	有

¹ 如果 ENABLE 为高（在退出停止模式后等待进入外部晶振启动）。

² DCO 时钟将不会被监控直到 DCO=1 从关模式或 FLL 旁路外部模式进入 SCM 模式。

8.4.10 时钟模式必要条件

通过写 CLKST1:CLKST0 一个时钟模式被请求同时通过 CLKST1:CLKST0 该实际时钟模式被表示。如果最小值条件被满足，显示在 CLKST:CLKST0 中的状态应该和 CLKST1:CLKST0 中请求模式一样。表 8-9 显示 CLKS 和 CLKST 之间的关系。它也显示 CLKS=CLKST 的条件或 CLKS!=CLKST 的原因。

注：

如果在下一个复位之前一个晶振被使用，那么确定置 REFS=1 和 CLKS=1X 在第一次写 ICGC1 寄存器时。如果未能这样做将导致 REFS=0 这将防止振荡放大器被启用直至下一次复位发生时。

表 8-9 ICG 状态表

实际模式 CLKST	预期的 模式 CLKS	范围	参考频率 (Preference)	比较周期时间	ICGOUT	CLKS=CLKST 的条件	CLKS!=CLKST 的原因
关模式 (XX)	关(XX)	X	0	—	0	—	—
	FBE(10)	X	0	—	0	—	ERCS=0
SCM (00)	SCM(00)	X	$f_{ICGIRCLK} / 7^2$	$8/f_{ICGIRCLK}$	ICGD CLK/R	不从FBE跳 转到SCM	—
	FEI(01)	0	$f_{ICGIRCLK} / 7^{(1)}$	$8/f_{ICGIRCLK}$	ICGD CLK/R	—	DCOS=0
	FBE(10)	X	$f_{ICGIRCLK} / 7^{(1)}$	$8/f_{ICGIRCLK}$	ICGD CLK/R	—	ERCS=0
	FEE(11)	X	$f_{ICGIRCLK} / 7^{(1)}$	$8/f_{ICGIRCLK}$	ICGD CLK/R	—	DCOS=0或 ERCS=0

FEI(01)	FEI(01)	0	$f_{\text{ICGIRCLK}} / 7$	$8/f_{\text{ICGIRCLK}}$	ICGD CLK/R	DCOS=1	—
	FEE(11)	X	$f_{\text{ICGIRCLK}} / 7$	$8/f_{\text{ICGIRCLK}}$	ICGD CLK/R	—	ERCS=0
FBE (10)	FBE(10)	X	0	—	ICGD CLK/R	ERCS=1	—
	FEE(11)	X	0	—	ICGD CLK/R	—	LOCS=1&ER CS=1
FEE (11)	FEE (11)	0	f_{ICGIRCLK}	$2/f_{\text{ICGIRCLK}}$	ICGD CLK/R 3	ERCS=1且 DCOS=1	—
		1	f_{ICGIRCLK}	$128/f_{\text{ICGIRCLK}}$	ICGD CLK/R (2)	ERCS=1且 DCOS=1	—

¹ 在写 CLKs 后 CLKST 将不立即更新。在 CLKST 更新到新值前需要几个总线周期。

² 在 SCM 模式下参考频率对 ICGOUT 无效，但该参考频率仍然使用在作比较中，该比较由 DCOS 位决定。

³ 在初始的 LOCK 后将会变为 ICGDCLK/2R 在初始锁定过程，在 MFD 位被改变后 FLL 重新锁定。

8.4.11 固定频率时钟

ICG 提供一个固定频率时钟输出，XCLK，被片上外设使用。在所有模式下除了 FEE，这个时钟输出等于内部总线时钟，BUSCLK。在 FEE 模式下。当下列条件被满足时 XCLK 等于 ICGERCLK 除以 2：

$(P*N)/R \geq 4$ ，P 由 RANGE 决定（见表 8-11），N 和 R 由 MFD 和 RFD 各自决定（见表 8-12）。
LOCK=1

如果以上两个条件未满足那么 XCLK 等于 BUSCLK。

当 ICG 处于 FEI 或 SCM 模式时，XCLK 被关闭。当 ICG 处于 FEI 或 SCM 模式时任何使用 XCLK 作为时钟源的外设不必关闭 XCLK。

8.4.12 高增益振荡器

该振荡器可以运行在一个高增益振荡器（HGO）模式，当运行在 FBE 或 FEE 模式下时高增益振荡器（HGO）模式能提高振荡器对 EMC 噪音的抵抗力。通过写 1 到 ICGC1 寄存器中的 HGO 位选择该模式。HGO 可与高频和低频振荡器一起使用但仅当 ICGC1 寄存器中的 REFS=1 时有效。当 HGO=0 时，标准低功耗振荡器被选中。仅在复位后该位可写。

8.5 初始化/应用信息

8.5.1 简介

该节给予了一些在初始化时的基本方向，有关用户希望选择的配置。在一些应用中，串行通信连接可能规定参考时钟的准确性。在其他应用中，最低功耗可能是时钟要考虑的事项。其他时钟可能以最低功耗位最高目标。ICG 允许很大的灵活性来在应用时选择最好的。

该节接下来将介绍一个包含多种配置的例子。

表 8-10 ICG 配置

	时钟参考源=内部	时钟参考源=外部
使用FLL	FEI $4\text{MHz} < f_{\text{Bus}} < 20\text{MHz}$ 中等功率（将会小于FEE，如果震荡范围=高） 高时钟精确度（在IRG调整后） 最低系统功耗（不需要外部器件） IRG打开。DCO打开 ¹	FEE $4\text{MHz} < f_{\text{Bus}} < 20\text{MHz}$ 中等功率（将会小于FEI，如果震荡范围=低） 高时钟精确度 中等或高系统功耗（需要晶振，共鸣器或外部时钟源器件） IRG关闭。DCO打开
FLL旁路	SCM 该模块主要提供给迅速和可靠的系统启动。 $3\text{MHz} < f_{\text{Bus}} < 5\text{MHz}$ （默认） $3\text{MHz} < f_{\text{Bus}} < 20\text{MHz}$ （通过过滤器位） 中等功率 低精确度 IRG关闭。DCO打开并且开循环。	FBE $f_{\text{Bus}} \leq 8\text{MHz}$ ，当使用晶振或蜂鸣器时。 <u>最低功率</u> 最高时钟精确度 中等或高系统功耗（需要晶振，共鸣器或外部时钟源器件） IRG关闭。DCO关闭。

¹ IRG通常消耗100 μA 。根据输出频率，FLL和DCO通常消耗0.5-2.5mA。为了最小功耗和最小抖动，选择尽可能小的N和R。

注：前缀\$表示十六进制，前缀%表示二进制，无前缀表示十进制。

重要的配置信息在这里重复列出，以供参考。

表 8-11 ICGOUT 频率计算选项

时钟配置	f_{ICGOUT}^1	P	注意
SCM	f_{ICGDCLK} / R	NA	复位后典型 $f_{\text{ICGOUT}}=8\text{MHz}$
FBE	f_{ext} / R	NA	
FEI	f_{ICG} / R	64	通常 $f_{\text{IRG}}=243\text{MHz}$
FEE	$f_{\text{ICGOUT}} * P * N / R$	范围=0; P=64 范围=1; P=1	

1. 确保 f_{ICGDCLK} ，等于 $f_{\text{ICGOUT}} * R$ ，不要超过 $f_{\text{ICGDCLKmax}}$

表 8-12 MFD 和 RFD 编码表

MCF值	倍增因子	RFD	分频因子
000	4	000	$\div 1$
001	6	001	$\div 2$
010	8	010	$\div 4$
011	10	011	$\div 8$
100	12	100	$\div 16$
101	14	101	$\div 32$
110	16	110	$\div 64$
111	18	111	$\div 128$

8.5.2 例1：外部晶振=32kHz，总线频率=4.19MHz

在该例中，FLL 将加倍外部 32kHz 振荡器到 8.38MHz，来产生 4.19MHz 的总线频率。

MCU 复位后，ICG 处于自时钟模式（SCM），提供大约 8MHz 的 ICGOUT，对于 4MHz 的总线频率（ f_{Bus} ）。

时钟配置是使用 FLL，外部（FEE）。所以有：

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

N/R:

$N/R=8.38\text{MHz}/(32\text{kHz})\times 64=4$ ；可以选择 $N=4$ ， $R=1$ （等式 8-2）

执行的操作，每个寄存器需要的值：

ICGC1 = \$38 (%00111000)

Bit 7	HGO	0	配置振荡器低功耗
Bit 6	RANGE	0	配置振荡器低频率;FLL 预分频因子 64
Bit 5	REFS	1	振荡器需要使用晶振或蜂鸣器
Bits 4:3	CLKS	11	使用 FLL 外部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	未实现或保留；读为 0

ICGC2 = \$00 (%00000000)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	000	置位 MFD 倍乘因子 4
Bit 3	LOCRE	0	丢失时钟失产生一个中断请求
Bits 2:0	RFD	000	置位分频因子 ÷ 1

ICGS1 = \$xx

该位只读除了清中断标志

ICGS2 = \$xx

该位只读；执行任何临界任务前应该读 DCOS=1

ICGFLTLU/L = \$xx

仅在自时钟模式时需要; FLT 通过回路调整产生 8.38 MHz DCO 时钟

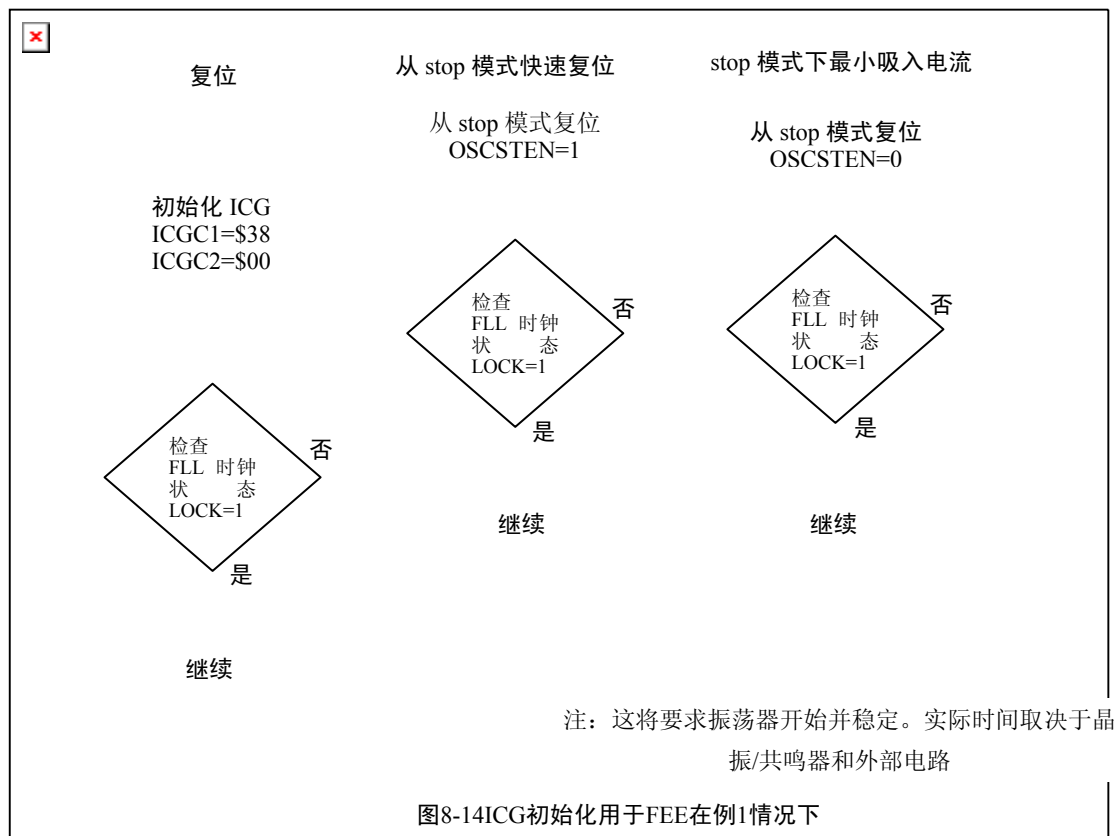
Bits 15:12 未使用 0000

Bits 11:0 FLT 不需要用户初始化

ICGTRM = \$xx

Bits 7:0 TRIM 当调整内部振荡器时，只需要写。当时钟源是外部晶振时不使用。

图 8-14 显示了 ICG 初始化时所需要的三个条件的流程图。



8.5.3 例2：外部晶振=4MHz，总线频率=20MHz

在该例中，FLL(在 FEE 模式)将加倍外部 4MHz 振荡器到 40MHz，来产生 20MHz 的总线频率。

MCU 复位后，ICG 处于自时钟模式（SCM），提供大约 8MHz 的 ICGOUT，对于 4MHz 的总线频率（ f_{Bus} ）。

复位时，初始化软件，时钟配置是使用 FLL，外部（FEE）。所以

$$f_{ICGOUT} = f_{ext} * P * N / R; P=1, f_{ext}=4MHz \quad (\text{等式 8-3})$$

N/R:

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

执行的操作，每个寄存器需要的值：

ICGC1 = \$78 (%01111000)

Bit 7	HGO	0	配置振荡器低功耗
Bit 6	RANGE	1	配置振荡器高频率;FLL 预分频因子 1
Bit 5	REFS	1	振荡器需要使用晶振或蜂鸣器
Bits 4:3	CLKS	11	使用 FLL 外部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	未实现或保留；读为 0

ICGC2 = \$30 (%00110000)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	0110	置位 MFD 倍乘因子 10
Bit 3	LOCRE	0	丢失时钟失产生一个中断请求
Bits 2:0	RFD	000	置位分频因子 ÷ 1

ICGS1 = \$xx

该位只读除了清中断标志
ICGS2 = \$xx
该位只读；执行任何临界任务前应该读 DCOS=1
ICGFLTLU/L = \$xx
在该例中未使用
ICGTRM = \$xx
在该例中未使用

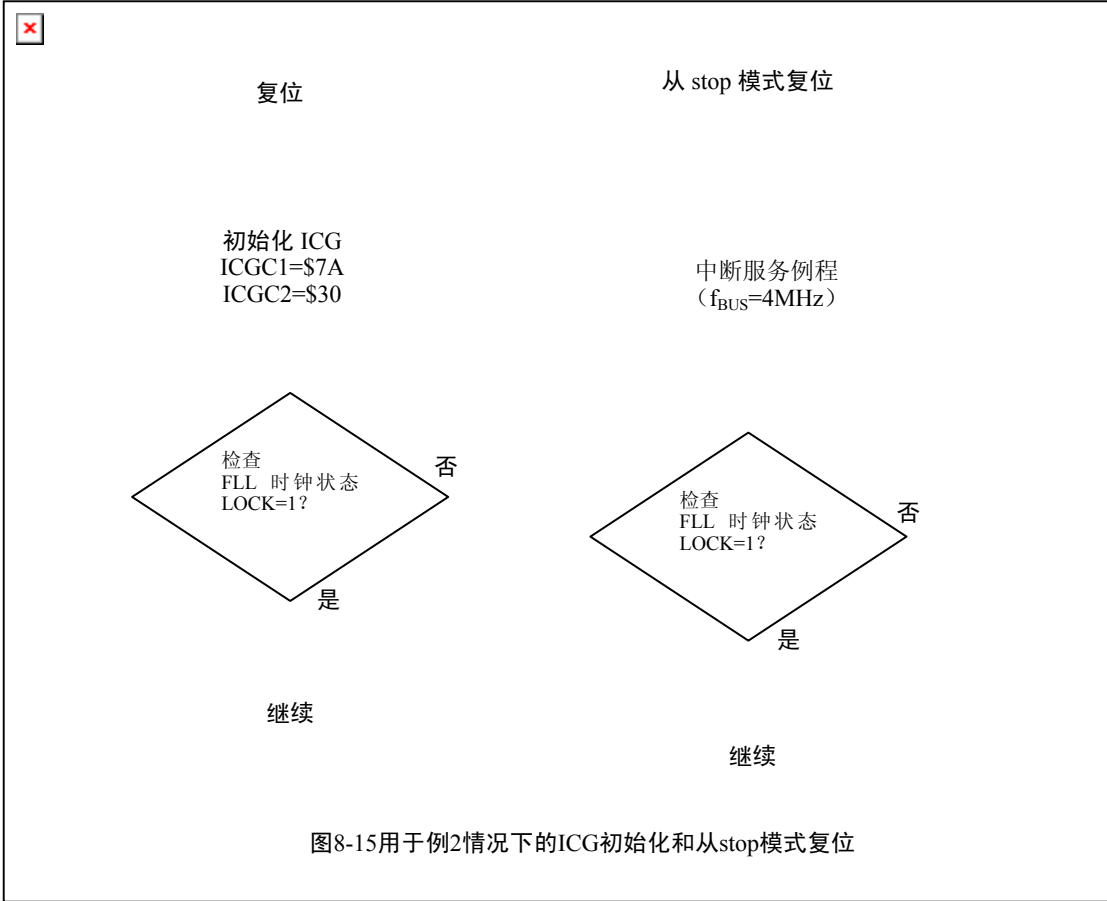


图8-15用于例2情况下的ICG初始化和从stop模式复位

8.5.4 例3：无外部晶振，总线频率=5.4MHz

在该例中，FLL 将加倍内部 243kHz（大约）参考时钟到 10.8MHz，来产生 5.4MHz 的总线频率。该系统也被用于调整，基于外部参考信号调整频率。

MCU 复位后，ICG 处于自时钟模式（SCM），提供大约 8MHz 的 ICGOUT，对于 4MHz 的总线频率（f_{BUS}）。

时钟配置是使用 FLL，内部（FEI）。所以

$$f_{ICGOUT} = (f_{IRG} / 7) * P * N / R; P=64, f_{IRG}=243kHz \quad (\text{等式8-5})$$

N/R:

$$N/R = 10.8MHz / (243/7kHz) * 64 = 4.86; \text{ 可以选择 } N=10, R=2 \quad (\text{等式 8-6})$$

需要一个调整程序调整频率到正好 5.4MHz。例 4 是一个调整程序的例子。

执行的操作，每个寄存器需要的值：

ICGC1 = \$28 (%00101000)

Bit 7 HGO 0 配置振荡器低功耗

Bit 6	RANGE	0	配置振荡器低频率;FLL 预分频因子 64
Bit 5	REFS	1	振荡器需要使用晶振或蜂鸣器（位不用关心）
Bits 4:3	CLKS	01	使用 FLL 内部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	未实现或保留；读为 0

ICGC2 = \$31 (%00110001)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	011	置位 MFD 倍乘因子 10
Bit 3	LOCRES	0	丢失时钟失产生一个中断请求
Bits 2:0	RFD	001	置位分频因子 ÷2

ICGS1 = \$xx

该位只读除了清中断标志

ICGS2 = \$xx

该位只读；执行任何临界任务前应该读该位是一种好的方法

ICGFLTLU/L = \$xx

在该例中未使用

ICGTRM = \$xx

Bits 7:0 TRIM 当调整内部振荡器时，只需要写。独立操作（参见例 4）。

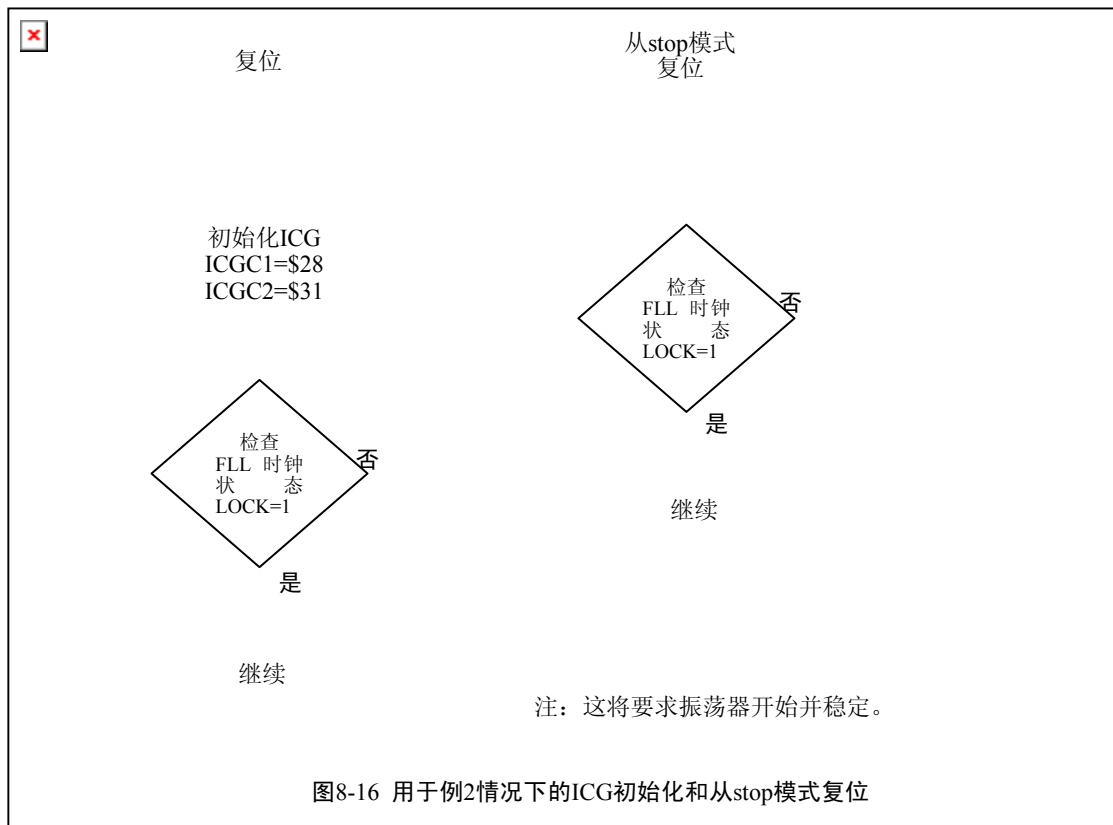


图8-16 用于例2情况下的ICG初始化和从stop模式复位

8.5.5 例4：内部时钟产生器调整

内部产生时钟源确保有一个 25%的浮动值。在一些情况，这可能会有足够的正确性。在其他应用中要求比较紧密的频率，一个调整程序可以帮助产生一个准确源。该节描述了一个调整内部振荡器的例子。

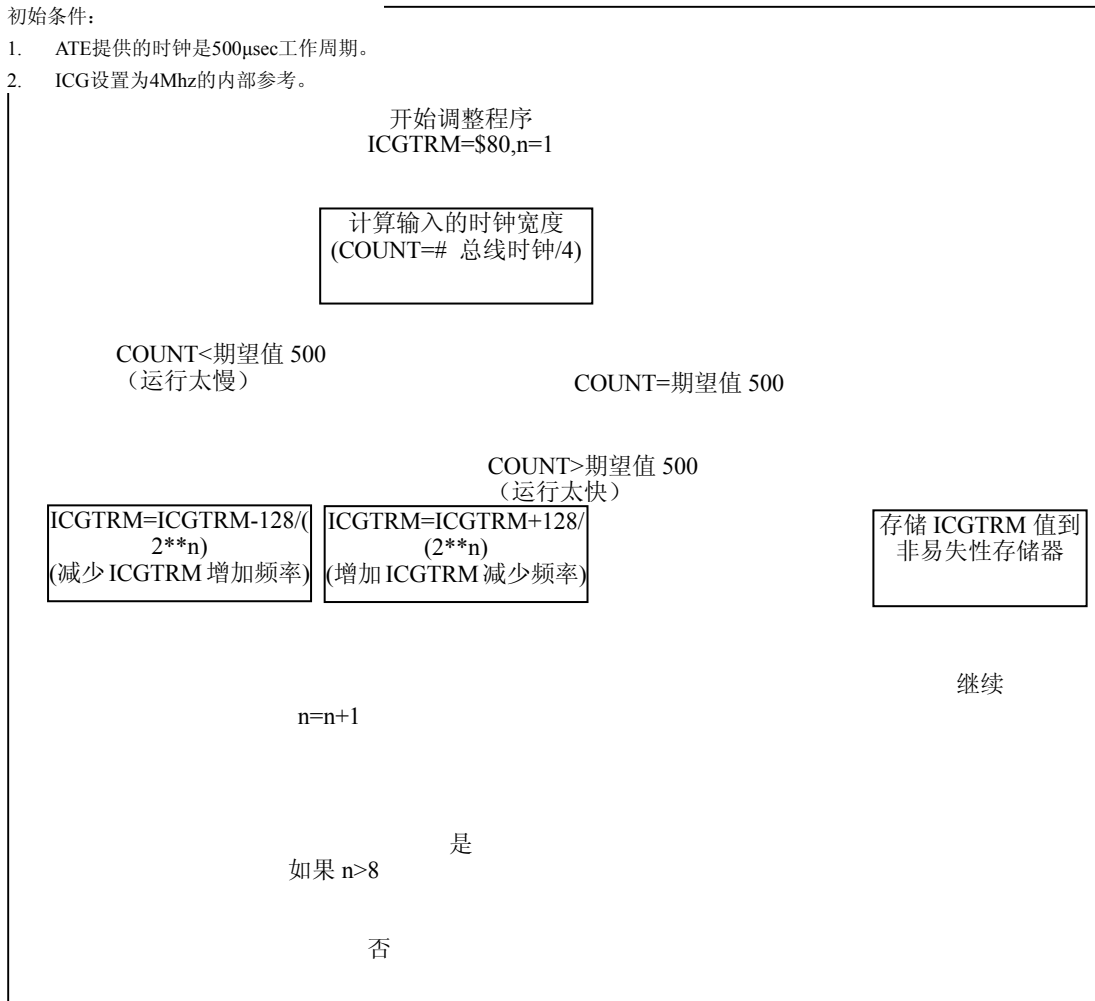


图8-17 调整程序

在这个特殊情况中，MCU 已经放到 PCB 上，整体正在用自动化测试工具进行测试。当用户使用软件控制时，一个单独的信号或信息提供给 MCU 操作。当测试者提供了一个准确的参考信号，MCU 就像图 8-17 所描述的那样初始化调整程序。

如果希望总线频率接近设备所允许的最大值，推荐使用一个减小约数（R）两倍于最终值。调整程序完成后，存储减小约数。这将会避免意外地超过最大时钟频率。

第九章 键盘中断（S08KBIV1）

9.1 简介

本章解释了软件控制相关的并行输入输出（I/O）。MC9S08AC16 有 7 个并行 I/O 端口，总共 54 个通用 I/O 引脚。关于这些引脚的逻辑和硬件方面的更多信息参见第二章“引脚和连接”。

9.2 键盘引脚复用

KBI 的输入 KBIP6 与 PTD3 和 AD11 复用引脚。KBI 的输入 KBIP5 与 PTD2 和 AD10 复用引脚。这些与端口和 ADC 功能操作复用引脚的输入与上面描述 KBIP7 的方式有相同的操作。

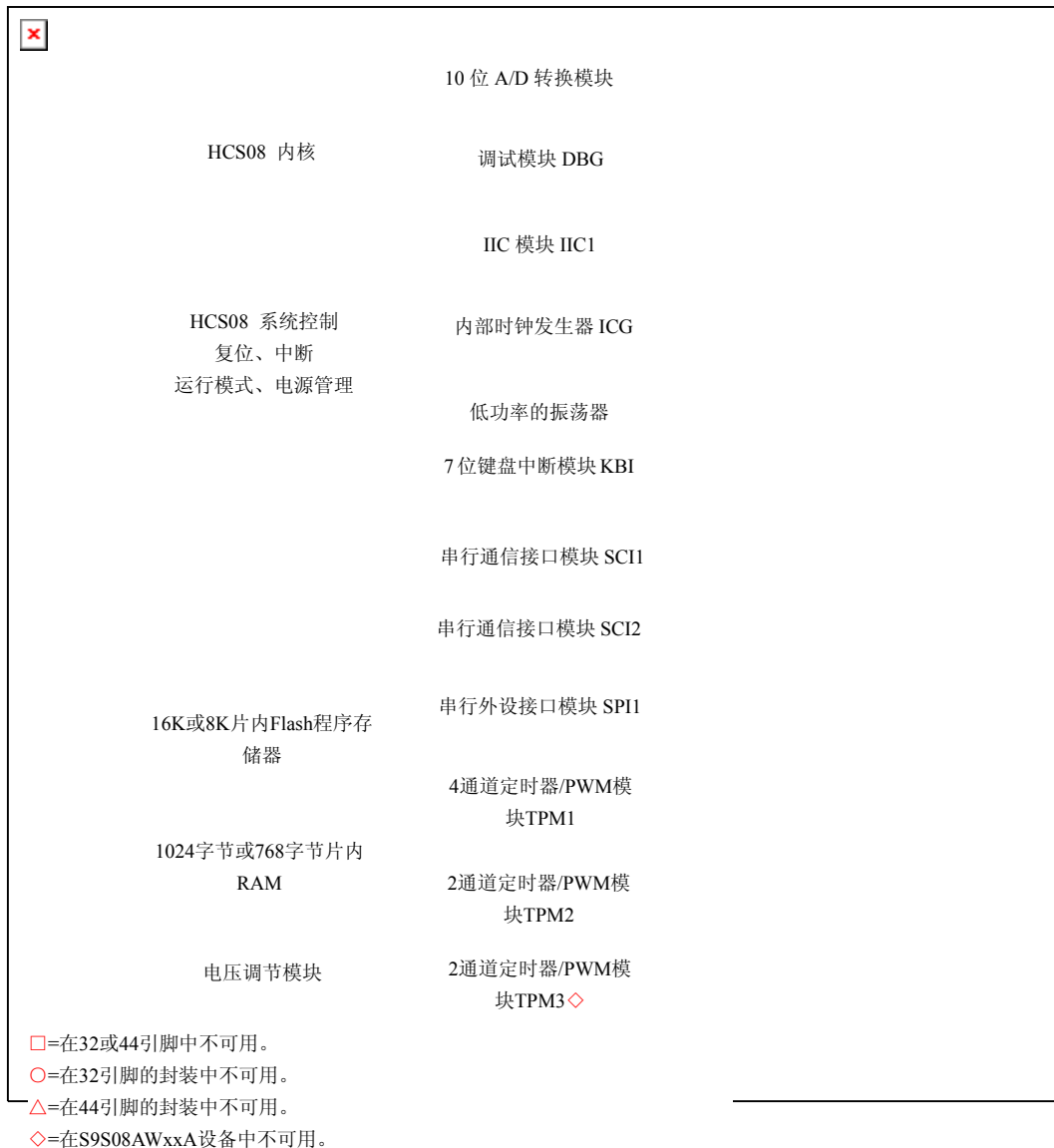
KBI 的输入 KBIP4-KBIP0 和 PTG4-PTG0 复用引脚。这些引脚与上面描述 KBIP7 的方式有相同的操作，除了不复用 ADC 输入。

KBIP3-KBIP0 总是下降沿/低电平触发。KBIP7-KBIP4 可以被设置位上升沿/高电平或下降沿/低电平触发。当所有的 KBIP7-KBIP0 使能并且被设置位检测上升沿/高电平，而且通过引脚的相应端口上拉电阻使能位来使能引脚上拉电阻时，引脚上的下拉电阻寄存器而不是上拉电阻寄存器使能。

9.3 特征

键盘中断模块（KBI）特征包括：

- 4 个下降沿/低电平触发
- 4 个上升沿/高电平或下降沿/低电平触发
- 可选择边沿或边沿电平触发
- 中断标志和中断使能控制
- 从 stop3、stop2、stop1 或等待状态唤醒 MCU



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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测 (IRQEDG=1)，下拉使能。
3. IRQ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
4. 引脚包含集成的上拉设备。
5. PTD3,PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能 (KBIPE_n=1) 而且上升沿被选择 (KBEDG_n=1)，下拉使能。

图9-1 MCS0816AC框图

9.3.1 KBI框图

图 9-2 描述了 KBI 模块的框图。



图9-2 KBI框图

9.4 寄存器定义

本节描述了 KBI 模块所有的寄存器和相关控制位。

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。通常，Freescale 提供一个头文件把它们名称翻译为绝对地址。

9.4.1 KBI状态控制寄存器（KBISC）

	第7位	6	5	4	3	2	1	第0位
读	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
写						KBACK		
复位	0	0	0	0	0	0	0	0

图9-3 KBI状态控制寄存器（KBISC）

表 9-1 KBISC 寄存器域描述

域	描述
7:4 KBEDG[7:4]	键盘边沿选择位——当引脚被设置位键盘中断输入时（KBIPEn=1）时，每个可读写位可以决定相应的KBI端口引脚的触发事件需要的边沿或边沿和电平的极性。参见KBIMOD控制位，它决定引脚是边沿触发还是边沿和电平触发。 0 下降沿/低电平。 1 上升沿/高电平。
3 KBF	键盘中断标志——只要在使能的KBI引脚上检测到选定的边沿事件，这个只读位就被复位。若KBACK控制位置1，标志位清零。如果KBIMOD=1选择了边沿和电平操作而且使能的KBI引脚保持引起事件需要的电平，标志位保持置位1。 KBF可以用来作为软件轮询标志（KBIF=0）或向CPU产生硬件中断请求。 0 无KBI中断挂起。 1 有KBI中断挂起。
2 KBACK	键盘中断确认——通过置1KBACK这个只写位（读为0）可以清KBF状态标志。如果KBIMOD=1选择了边沿和电平操作而且使能的KBI引脚保持引起事件需要的电平，KBF保持置位1，所以置1KBACK不会清KBF标志。
1 KBIE	键盘中断使能——当KBF状态标志位1时，这个可读写控制位决定硬件中断是否产生。当KBIE=0时，没有硬件中断产生，但是KBF仍可作为软件轮询使用。 0 KBF不产生硬件中断（使用轮询）。 1 当KBF=1时，发出KBI硬件中断请求。

KBIMOD	键盘检测模式——这个可读写控制位决定检测仅边沿还是边沿和电平。KBI口的0-3位检测仅下降沿或下降沿和低电平。KBI口的4-7位设置位检测下面中的一种： ● 仅上升沿或上升沿和高电平（KBEDGn=1） ● 仅下降沿或下降沿和低电平（KBEDGn=0） 0 G□n位低输出驱动。 1 G□n位高输出驱动。
--------	--

9.4.2 KBI引脚使能寄存器（KBIPE）

	第7位	6	5	4	3	2	1	第0位
读	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
写						KBACK		
复位	0	0	0	0	0	0	0	0

图9-4 KBI引脚使能寄存器（KBIPE）

表 9-2 KBIPE 寄存器域描述

域	描述
7:0 KBIPE[7:0]	键盘引脚使能位——每个可读写位选择相应的KBI口引脚作为键盘中断输入使能还是通用I/O引脚。 0 KBI口的位n是通用I/O引脚与KBI无关。 1 KBI口的位n作为键盘中断输入。

9.5 功能描述

9.5.1 引脚使能

KBIPE 寄存器的 KBIPE_n 控制位允许用户使能（KBIPE_n=1）所有接到 KBI 模块的与 KBI 相关的引脚的组合。KBIPE 中为 0 的引脚作为通用 I/O 引脚，与 KBI 模块无关。

9.5.2 边沿和电平触发

同步逻辑用于检测边沿。在检测边沿前，KBI 模块中使能的键盘输入不是触发需要的电平（deasserted level）。

在一个总线周期内，使能的键盘输入信号是逻辑 1，下一个总线周期为逻辑 0，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，下一个总线周期为逻辑 1，则上升沿被检测到。

KBIMOD 控制位通过置位来重新配置检测逻辑，如此以至于可以检测边沿和电平。在 KBIMOD=1 时，当检测到边沿时（当一个或多个使能引脚从 1 变成 0，其他使能引脚仍然是 0），KBF 状态标志置位。但是只要有使能的引脚为 1，标志位保持置位（不能清零）。在停止模式下时，KBI 输入作为异步的电平触发输入，所以可以在停止模式下唤醒 MCU。

9.5.3 KBI中断控制

当 KBI 输入引脚检测到边沿事件时，KBF 状态标志置位。如果 KBISC 寄存器的 KBIE=1，当 KBF=1 时，发出硬件中断请求。置 1 键盘确认位（KBACK）KBF 标志清零。

当 KBIMOD=0（选择仅边沿操作）时，置 1 KBACK 则 KBF 标志清零。当 KBIMOD=1（选择边沿和电平操作）时，只要任何键盘输入为 1，KBF 标志不能清零。

第十章 定时器/脉宽调制（S08TPMV3）

10.1 简介

MC9S08AC16 系列包含三个独立定时器/脉宽调制模块，支持传统的输入捕捉，输出捕捉或缓存每条通路上的 PWM。在每个 TPM 中，一个控制位配置所有的通路，这样定时器可用作中心排列的脉宽调制功能。在每一个 TPM 中，定时功能，是基于对单独的带分频器 16 位计数器和模特征用来控制频率和定时参考的范围。这个计时系统非常适合于多种控制应用，TPM 上中心结盟脉宽调制能力扩展应用领域，在小家电上实现电机控制。采用固定的系统时钟，XCLK，作为任何 TPM 模块的时钟源允许 TPM 分频器运行振荡器率的一半频率 (ICGERCLK/2)。如果 ICG 用 FEE 模式配置且满足适当的条件，这种选择是唯一可用的（见 8.4.11 “固定时钟频率”一节）。在其它 ICG 模式中这样的选择是多余的，因为 XCLK、BUSCLK 相同。一个外部时钟源可以被连接到 TPMCLK 引脚上。TPMCLK 最大频率是总线时钟频率的四分之一。所有这三个 TPM 模块可自主选择 TPMCLK 作为时钟源。

注：

S9S08AW16A 和 S9S08AW8A 设备不执行 TPM3。这些装置只具备 44 和 48 脚封装的 2 通道 TPM1 和 2 通道 TPM2。和只有 32 脚封装的 2 通道 TPM1 和 TPM2。

10.2 特征

MC9S08AC16 系列定时器系统包括 4 通道 TPM1（32 位封装的 2 通道 TPM1），一个单独的 2 通道 TPM2 和一个单独的 2 通道 TPM3。定时器系统特征包括：

- 共有多达 8 个通道
 - 每个通道可用作输入捕捉，输出比较或缓存边缘对齐的 PWM
 - 上升沿，下降沿，或任何边沿输入捕捉触发
 - 置位，清零或切换输出比较行动
 - 可选极性的 PWM 输出
- 在所有通道上，每个 TPM 可配置带缓冲，中心对齐脉宽调制（CPWM）
- 每个 TPM 用来分频的时钟源是独立选择作为总线时钟，固定系统时钟或外部引脚
 - 分频因子可以是 1、2、4、8、16、32、64 或 128
 - 外部时钟输入：通过 TPM1、TPM2 和 TPM3 使用 TPMCLK
- 16 位自由运行或上行/下行（CPWM）等计数操作
- 控制计数器范围的 16 位模数寄存器
- 定时系统使能
- 每一个 TPM 模块包括每通道一个中断和一个终端计数中断

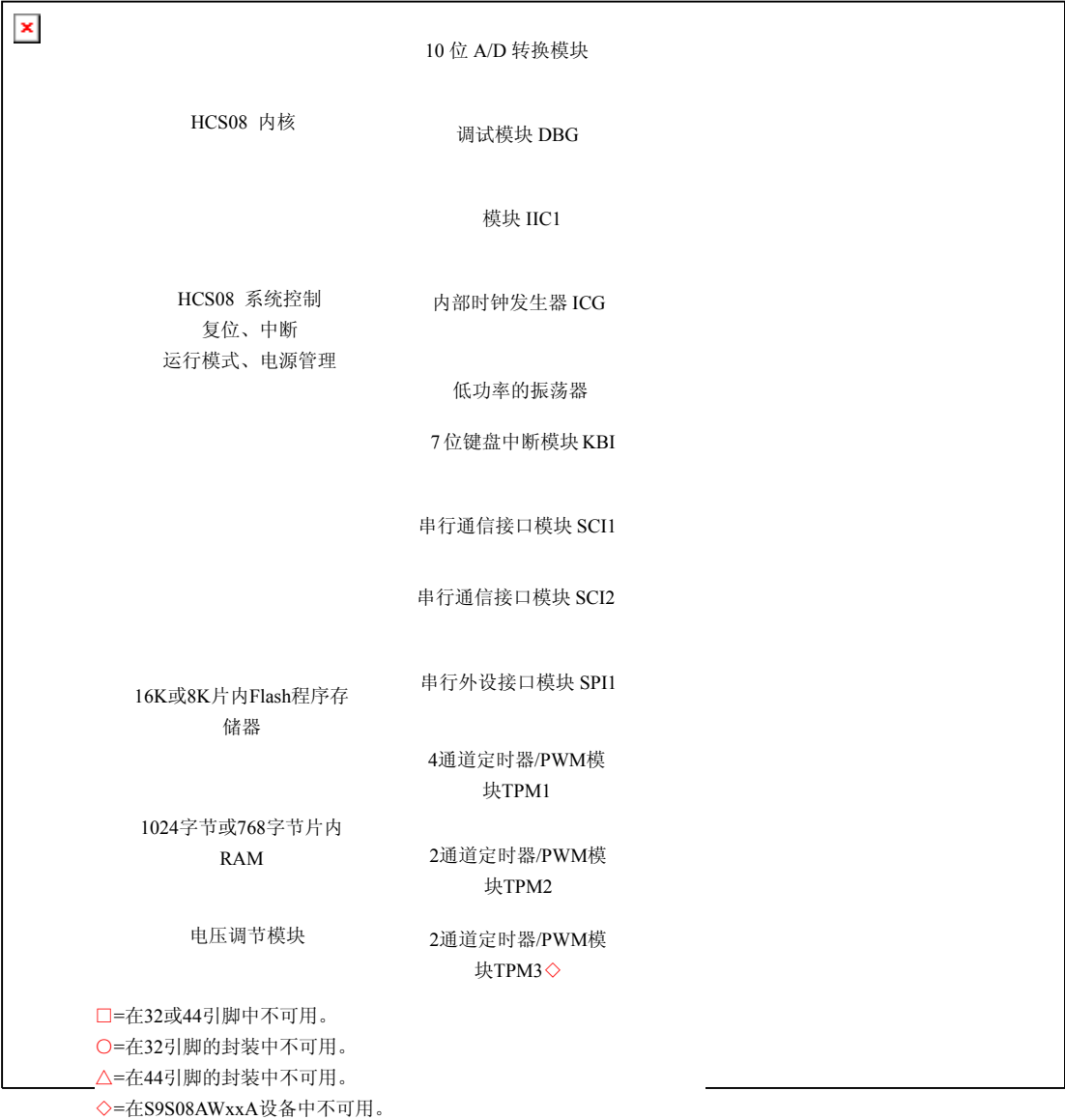


图10-1MC9S08AC16框图

注：

1. 如果是输入端口引脚用上拉设备软件配置
2. 如果 IRQ（IRQPE=1）使能引脚包含上拉/下拉软件配置。如果上升沿检测有效（IRQEDG=1）下拉使能。
3. IRQ 不需要钳位二极管连接到 VDD。IRQ 不需要超过 VDD 电压来驱动。
4. 引脚包含综合上拉设备。
5. PTD3, PTD2, PTD4 包含上拉和下拉设备。当 KBI（KBIPEn=1）使能且上升沿检测有效（KBEDGn=1）则下拉使能。

10.3 TPMV3与以前版本的区别

TPMV3 是定时/脉宽调制模块的最新版本，用来处理先前版本中的勘误。下一节概述了 TPMV3 和 TPMV2 模块的区别，以及移植代码时需要考虑的任何因素。

表 10-1 TPMV2 和 TPMV3 移植条件

作用	TPMV3	TPMV2
----	-------	-------

写入到TPMxCnTH:L寄存器 ¹		
写入到 TPMxCNTH 或 TPMxCNTL 寄存器	清TPM计数器 (TPMxCNTH:L) 和分频计数器	清TPM计数器 (TPMxCNTH:L)
读 TPMxCNTH:L 寄存器 ¹		
在BDM模式下，读 TPMxCNTH:L 寄存器	TPM计数器的返回值，是不可修改的	在BDM模式有效之前，如果只有一字节的TPMxCNTH:L寄存器值可读则从读取缓冲区（而非不可修改的TPM计数器值）返回TPMxCNTH:L锁存值
在BDM模式下，写 TPMxSC，TPMxCNTH，TPMxCNTL	清除读一致机制	不清除读一致机制
读 TPMxCnVH:L 寄存器 ²		
在BDM模式下，读TPMxCnVH:L 寄存器	返回TPMxCnVH:L寄存器的值	在BDM模式有效之前，如果只有一字节的TPMxCnVH:L寄存器值可读则从读取缓冲区（而非TPMxCnVH:L寄存器值）返回TPMxCnVH:L锁存值
在BDM模式下，写TPMxCnSC	清除读一致机制	不清除读一致机制
写 TPMxCnVH:L 寄存器		
在输入捕捉模式下，写TPMxCnVH:L 寄存器 ³	不允许。	允许
在输出比较模式下，当 (CLKSB:CLKSA not = 0:0)，写 TPMxCnVH:L 寄存器 ³	用在第二个字节被写入后，TPM计数器（停止分频计数）下个变化时的写缓冲区的值来更新TPMxCnVH:L寄存器内容	在第二字节内容被写入后，总是更新其寄存器内容

表 10-1 TPMV2 和 TPMV3 移植条件（续）

作用	TPM3	TPM2
在边对其模式下，当 (CLKSB:CLKSA != 00) 写 TPMxCnVH:L 寄存器	当 TPM 计数器值从 (TPMxMODH:L-1) 到 TPMxMODH:L，两个字节被写过后用写缓冲区的值更新TPMxCnH:L值 注：如果TPM计数器是一个自由运行计数器，那么当 TPM 计数器值从 \$FFEE 变化到 \$FFFF，发生更新操作	两字节内容被写入后，TPM计数器值从TPMxMODH:L到\$0000，执行更新操作
在中心对其模式下，当 (CLKSB:CLKSA != 00)，写 TPMxCnVH:L 寄存器 ⁴	当 TPM 计数器值从 (TPMxMODH:L-1) 到 TPMxMODH:L，两个字节被写过后用写缓冲区的值更新TPMxCnH:L值 注：如果TPM计数器是一个自由运行计数器，那么当 TPM 计数器值从 \$FFEE 变化到 \$FFFF，发生更新操作	两字节内容被写入后，TPM计数器值从TPMxMODH:L到(TPMxMODH:L-1)，执行更新操作
中心对齐PWM		
TPMxCnVH:L = TPMxMODH:L ⁵	产生100%占空比	产生0%占空比

TPMxCnVH:L = (TPMxMODH:L - 1) ⁶	产生近 100% 占空比	产生 0% 占空比
TPMxCnVH:L从0x0000变到非零值 ⁷	等待一个新的PWM周期开始来使用新的占空比设置	在当前PWM周期中期(计数为0x0000)改变通道输出
TPMxCnVH:L 从非零值变到0x0000 ⁸	用旧占空比设置完成了当前PWM周期	用新占空比设置完成了当前PWM周期
在BDM模式下写TPMxMODH:L寄存器		
在BDM模式下写TPMxSC寄存器	清除TPMxMODH:L寄存器写一致机制	不清除写一致机制

1 更多信息参考 10.5.2 节, “TPM 计数器寄存器 (TPMxCNTH: TPMxCNTL)”

2 更多信息参考 10.5.5 节, “TPM 通道值寄存器 (TPMxCnVH: TPMxCnVL)”

3 更多信息参考 10.6.2.1 节, “输入捕捉模式”

4 更多信息参考 10.6.2.4 节, “中心对齐 PWM 模式”

5 更多信息参考 10.6.2.4 节, “中心对齐 PWM 模式” 【SE110-TPM 案例 1】

6 更多信息参考 10.6.2.4 节, “中心对齐 PWM 模式” 【SE110-TPM 案例 2】

7 更多信息参考 10.6.2.4 节, “中心对齐 PWM 模式” 【SE110-TPM 案例 3, 5】

8 更多信息参考 10.6.2.4 节, “中心对齐 PWM 模式” 【SE110-TPM 案例 4】

10.3.1 TPMV1的变换

作为 10.3 节“TPMV3 与先前版本的区别”的补充,当从带 TPMV1 的芯片设备变换时,牢记以下需考虑的事项。

- 当定时器是因 TPMV2 而不是因为 TPMV3 不在输入捕捉模式下时,可以写通道值寄存器 (TPMxCnV)。
- 在脉宽调制边缘或中心对齐模式下,当定时器从 TPMMOD-1 变换到 TPMMOD 或者一个自由运行的定时器从 0XFFFE 到 0XFFFF 时,通道值寄存器在更新后导入。
- 同时配置 TPM 芯片模块时,最好在 TPMxCnV 作为一个写动作给 TPMxSC 复位 TPMxCnV 寄存器上的一致机制之前写 TPMxSC。

表10-2 转换到TPMV3的条件表

条件	动作 / 最优练习
写入通道值寄存器 (TPMxCnV) 导入	定时器肯定在输入捕捉模式下
更新记录在中心对齐或边缘对齐模式下的通道值寄存器 (TPMxCnV)	仅发生在定时器从TPMMOD-1变换到TPMMOD时或者一个自由运行的定时器从0XFFFE到0XFFFF时
复位通道值寄存器 (TPMxCnV) 导入一致机制	写TPMxSC
配置TPM模式	先写TPMxSC再写TPMxCnV寄存器

10.3.2特征

TPM 的显著特征:

- 一到 8 通道:
 - 每个通道可以是输入捕捉, 输出比较, 边缘对齐脉宽调制
 - 上升沿, 下降沿, 或任何边缘输入捕捉触发
 - 置位, 清零, 或锁存输出比较
 - 脉宽调制输出的极性选择
- 在所有通道上模块可被设置带缓冲和中心对齐脉宽调制

- 定时器时钟源可选作为分频总线时钟，固定式系统时钟，或外部时钟引脚
 - 分频因子可以是 1、2、4、8、16、32、64 或 128
 - 通过片上同步电路固定式系统时钟和总线时钟同步
 - 外部时钟引脚可以与其它任何定时器通道或独立输入引脚共享
- 16 位自运行可调值计数操作
- 定时器系统使能
- 每通道一个中断加终端计数中断

10.3.3 操作模式

一般来说，TPM 通道可以被独立设置在输入捕捉，输出比较或边缘对齐模式下操作。一个控制位的设置允许整个 TPM 通道转换到中心对齐脉宽调制模式。当中心对齐脉宽调制模式被选中，这个 TPM 模块上的其它通道上，输入捕捉，输出比较和边缘对齐脉宽调制功能都不可用。当微控制器处在活跃 BDM 背景调试模式或 BDM 前景模式下。TPM 暂停所有计数直到微控制器返回到正常的用户操作模式。在停止模式下时，所有的系统时钟，包括主要的振荡器，都被停止，所以说知道时钟恢复 TPM 是有效的低功耗模式。在等待模式下时，TPM 执行通常的操作。假如 TPM 不需产生一个实时参考时间或提供从等待模式唤醒 MCU 的中断源，用户可以通过在进入等待模式前禁止 TPM 功能来节省功耗。

● 输入捕捉模式

当一个选定的边缘事件发生在相关的 MCU 引脚上时，16 位定时器的当前计数器值被采集到通道值寄存器和一个中断标志位被设置。上升沿，下降沿，任意边沿或无边沿（禁用通道）可被选作为触发输入比较事件的活跃边沿。

● 输出比较模式

当定时器计数器寄存器的值匹配通道值寄存器值时，一个中断标志位被设置且一个选择输出行为发生在相应的 MCU 引脚。该输出比较行为可被选来迫使引脚为零，为 1，锁存它，或忽略该引脚（曾用于软件定时功能）。

● 边缘对齐脉宽调制模式

16 位模块寄存器的值加 1 来设置脉宽调制输出信号的周期。通道值寄存器设置脉宽调制输出信号的占空比。用户也可以选择脉宽调制输出信号的极性。在该周期末尾和占空比转换临界点，中断是允许可用的。这种类型的脉宽调制（PWM）信号被称作边缘对齐 PWM 是因为所有 PWM 主要边缘信号与周期开始时对齐，同时所有 TPM 通道也是一样的。

● 中心对齐脉宽调制模式

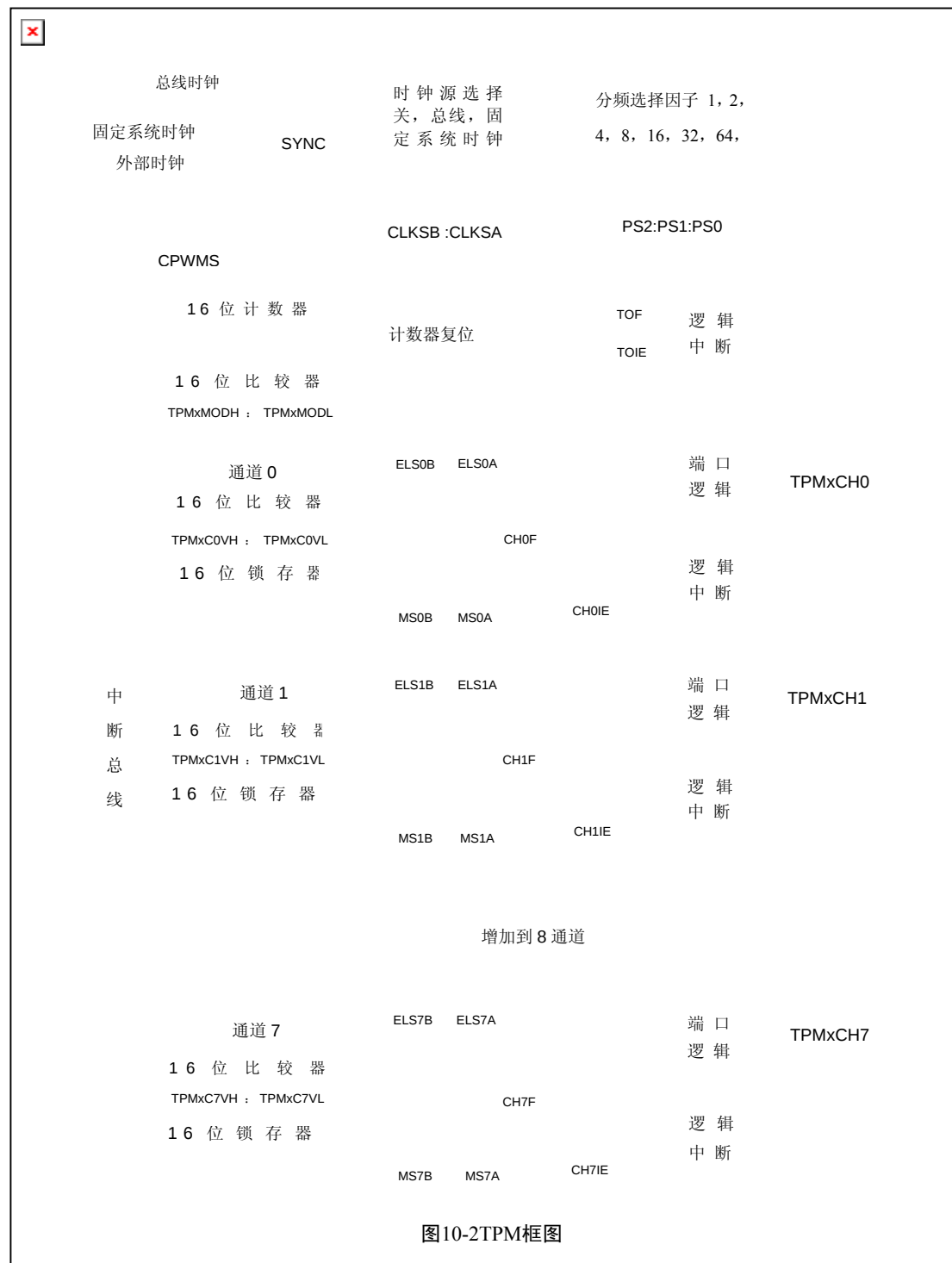
两倍 16 位模块寄存器值设置 PWM 输出周期，同时通道值寄存器设置占空比的期限。定时器计数增加直到它达到模块值然后计数减少直到 0。当计数值与通道值寄存器相匹配但计数值下降时，脉宽调制输出活跃。当计数值与通道值寄存器相匹配但计数值增加时，脉宽调制输出活跃。这种类型的脉宽调制（PWM）信号被称作中心对齐 PWM 是因为所有通道活跃占空比周期的中心与计数值零对齐。这种类型的 PWM 需求小家电中使用的电机。

这仅是一个概述，在稍后章节将进行详细介绍。

10.3.4 框图

TPM 每个通道使用一个输入/输出引脚，TPM_xCH_n（定时器通道 n）n 是通道号（1-8）。TPM 输入输出引脚与通用输入输出引脚复用（涉及输入输出引脚描述用特定芯片全片格式

说明)。表 10-2 下显示了 TPM 结构。TPM 中心组成部分是 16 位计数器，可以作为一个自运行计数器或一个可调值寄存器。TPM 计数器（在正常增加技术模式下操作时）为输入捕捉，输出比较，边缘对齐 PWM 功能提供参考时钟。定时器计数因子寄存器 TPMxMODH:TPMxMODL 控制计数器因子值（值 0X0000 或 0XFFFF 可使计数器自运行）。任何时候在不影响计数序列软件可以读取计数器值。TPMxCNT 计数器的任意一半被写入以复位计数器，不论其中数据是否被转移。



TPM 通道和输入捕捉，输出比较或边对齐 PWM 通道一样是可独立编程的。另一种情况是，TPM 可配置产生 CPWM 输出在所有频道上。当 TPM 被配置成 CPWM 时，计数操作

和计数器调值，输入捕捉，输出比较，EPWM 等功能是无效的。如果通道被配置成输入捕捉模式，则内接一个上拉设备。模块利用引脚相互作用的详情主要取决于芯片的执行情况因为 I/O 口引脚和相关通用 I/O 口控制位是模块的一部分。可参阅全芯片说明中个输入输出端口逻辑的讨论。因为中心边沿 PWM 通常被用于驱动 3 相交流电机和直流无刷电机，应用于三通道或六通道。

10.4 信号描述

表 10-3 显示用户可访问的 TPM 信号。有八个通道。当包含外部时钟，它可以复用任何 TPM 通道相同引脚，可以连接到一个独立输入引脚。具体实现可参阅全芯片说明中的 I/O 口的描述。

表 10-3 信号特性

名称	功能
EXTCLK ¹	外部时钟源可被选来驱动TPM计数器
TPMxCHn ²	TPM通道n相关的I/O引脚

1 若提前预设，该信号可复用任何通道引脚；然而依据全芯片运行，该信号可被连接到独立的外部引脚。

2 $n=(1-8)$

关于复位状态，端口连接，和这些引脚是否有上拉设备的详情请参阅全芯片说明。TPM 通道引脚可与通用 I/O 口引脚相关联且有被动上拉设备，当 TPM 或通用 I/O 控制配置相关引脚为输入时上拉设备可以用一个控制位使能。当无 TPM 功能能够使用相应的引脚时，引脚可被通用 I/O 控制，包括端口数据，数据方向寄存器。紧跟在复位后，无 TPM 功能可用，所以所有相关引脚回复为通用 I/O 控制。

10.4.1 详细信号描述

这部分将详细描述每个用户可访问引脚信号。虽然表 10-3 已经罗列所有引脚，任何 TPM 引脚可复用外部时钟源信号。因为 I/O 引脚逻辑不是 TPM 的一部分，关于 TPM 引脚功能交互和通用 I/O 控制包括端口数据，数据方向和上拉控制的具体实现可参阅全芯片说明。

10.4.1.1 EXTCLK——外部时钟源

位于定时器状态和控制寄存器中的控制位允许用户选择定时器无效，总线速率时钟（默认时钟），晶振时钟，或外部时钟可以作为驱动 TPM 分频器和 16 位计数器的时钟。在 TPM 中外时钟源是同步的。总线时钟设定时钟同步器；外部时钟频率不能超过总线速率时钟频率的四分之一，满足奈奎斯特准则，并允许抖动。外部时钟信号复用相同引脚作为通道 I/O 引脚，所以当被选做外部时钟源时通道引脚通道 I/O 功能不可用。用户应避免产生上述情况。如果这个引脚可用作外部时钟源（CLKSB:CLKSA），该通道能在输出比较模式下作为一个软件定时器（ELSnB:ELSnA=0:0）使用。

10.4.1.2 TPMxCHn——TPM通道n 输入输出引脚

MCU 上的每个 TPM 通道与一个 I/O 引脚相关联。这种引脚功能依赖于通道配置。TPM 引脚与通用 I/O 引脚复用，它的每个引脚有一个端口数据寄存器位，一个数据方向控制位和一个在任何时候若一个端口引脚被作为一个输入引脚时可被使能的带可选被动上拉的端口。当（ELSnB:ELSnA=0:0）或（CLKSB:CLKSA=0:0）时 TPM 通道不能控制 I/O 引脚所以一般转为通用 I/O 控制。当 CPWMS=1（ELSnB:ELSnA!=0:0）时所有带有 TPM 的通道被配置成中心对齐 PWM 且 TPMxCHn 引脚被 TPM 系统控制。当 CPWMS=0 时，MSnB: MSnA 控制位决定该通道是否配置为输入捕捉，输出比较或边对齐 PWM。当一个通道被配置为输入

捕捉模式（CPWMS=0，MSnB:MSnA=0:0，ELSnB:ELSnA!=0:0）时，TPMxCHn 引脚被作为 TPM 的一个边沿有效输入。ELSnB:ELSnA 控制位决定那种极性边沿将触发输入捕捉事件。同步器是基于用来同步总线时钟输入边沿总线时钟上的。这意味着在一个输入捕捉引脚上能被可靠检测到最小脉冲宽度是四倍总线时钟周期（接近两个总线时钟的理想时钟脉宽能被检测到）。TPM 用这个引脚作为一个输入捕捉的输入来顶替端口数据和数据方向控制。当一个通道被配置为输出比较模式（CPWMS=0，MSnB:MSnA=0:1，ELSnB:ELSnA=0:0）时，相关数据方向控制被顶替，TPMxCHn 引脚被认为是由 TPM 控制的一个输出，ELSnB:ELSnA 控制位决定该引脚如何被控制。ELSnB:ELSnA 剩余的三种组合决定 TPMxCHn 引脚是否被锁存，清空或每次 16 位通道值寄存器匹配定时器计数器时置位。当输出比较模式被选中，引脚上的前值被清除直到下个输出比较事件来临然后引脚被锁存。当一个通道被配置为边缘对齐 PWM 模式（CPWMS=0，MSnB=1，ELSnB:ELSnA!=0:0）时，数据方向被或略，TPMxCHn 引脚被强制为由 TPM 控制的输出，ELSnA 控制引脚上 PWM 输出信号的极性。当 ELSnB:ELSnA=1:0 时，TPMxCHn 引脚在每个新周期（TPMxCNT=0X0000）开始被强制拉高，当通道值寄存器等于定时器的计数器时引脚被强制拉低。当 ELSnA=1 时，TPMxCHn 引脚在每个新周期（TPMxCNT=0X0000）开始被强制拉低，当通道寄存器等于定时器的计数器时引脚被强制拉高。

TPMxMODH:TPMxMODL=0X0008

TPMxMODH:TPMxMODL=0X0005

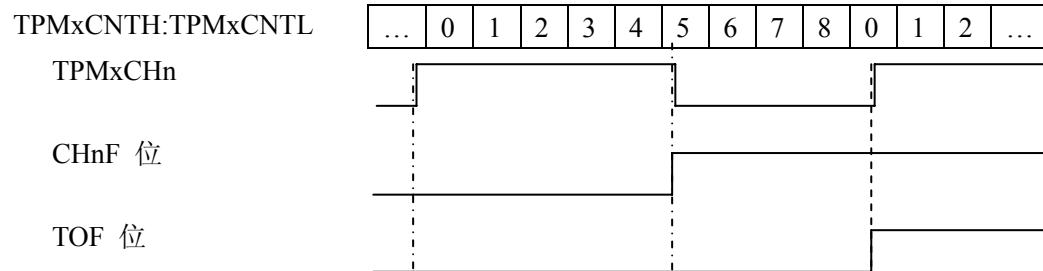


表 10-3 边缘对齐 PWM 模式的高真脉冲

TPMxMODH:TPMxMODL=0X0008

TPMxMODH:TPMxMODL=0X0005

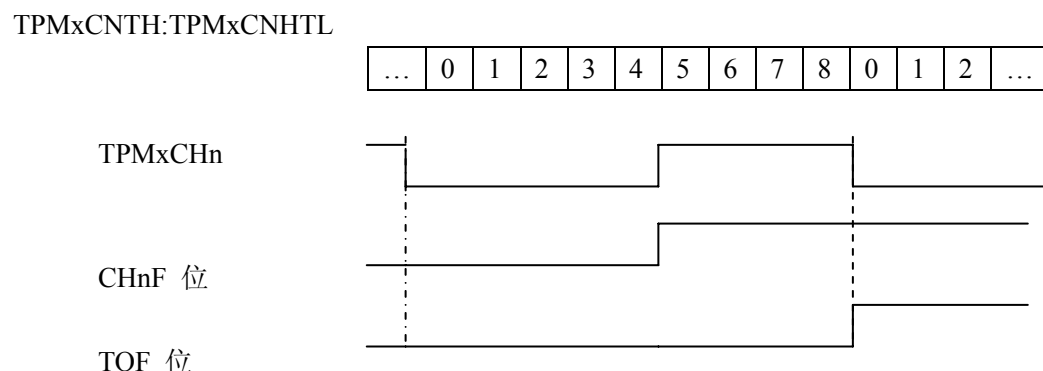


表 10-4 边缘对齐 PWM 模式的低真脉冲

当 TPM 被配置为中心对齐 PWM 模式（ELSnB:ELSnA!=0:0）时，在 TPM 模式下所有通道的数据方向被顶替，TPMxCHn 引脚被强制转换为由 TPM 控制的输出，ELSnA 位控制每个 TPMxCHn 输出的极性。如果 ELSnB:ELSnA=1:0，当定时器计数器值上升，通道值寄存器等于定时器计数器时 TPMxCHn 引脚被清零；当定时器计数器值下降，通道值寄存器等于定时器计数器值时，TPMxCHn 引脚被置位。如果 ELSnA=1，当定时器计数器值上升且通道值寄存器等于定时器计数器时，TPMxCHn 引脚被置位；当定时器计数器值下降且通道值

寄存器等于定时器计数器时 TPMxCHn 引脚被清零。

TPMxMODH:TPMxMODL=0X0008

TPMxMODH:TPMxMODL=0X0005

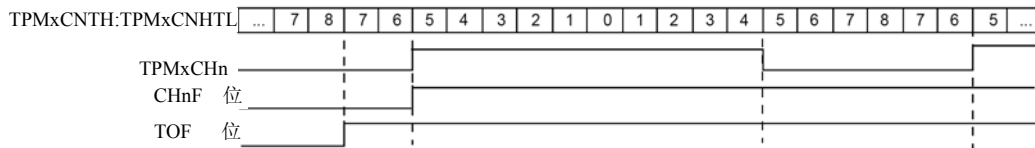


图 10-5 中心对齐 PWM 模式的高真脉冲

TPMxMODH:TPMxMODL=0X0008

TPMxMODH:TPMxMODL=0X0005

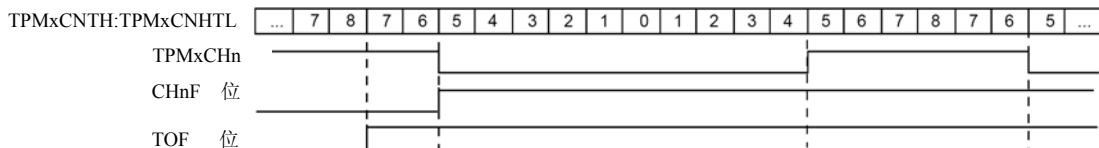


图 10-6 中心对齐 PWM 模式的低真脉冲

10.5 寄存器定义

这部分将按地址顺序描述寄存器。一个典型 MCU 系统可能包含若干 TPM，每个 TPM 有八个通道，所以寄存器的命名包含占位符用来明确那个 TPM 和那个通道被选用了。例如 TPMxCnSC 涉及到(TPM)x，通道 n。TPM1C2SC 可能是定时器 1 通道 2 的状态和控制寄存器。

10.5.1 TPM状态和控制寄存器（TPMxSC）

TPMxSC 包含溢出状态标记和用来配置中断使能的控制位，TPM 结构，时钟源，分频因子。这些控制与所有带这种定时器模块的通道相关。

读	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
写	0							
复位	0	0	0	0	0	0	0	0

图 10-7 TPM 状态和控制寄存器（TPMxSC）

表 10-4 TPMxSC 位描述

位	描述
7 TOF	定时器溢出标志，当TPM在达到预设的TPM计数器模块寄存器值后复位为0X0000时这个读写标志被置位。当TOF被置位可以通过读TPM状态和控制寄存器清TOF然后置TOF为0。如果在彻底清除序列前发生了另一个TPM溢出，该序列复位所以在彻底清空序列后TOF将保持置位。这样在清序列时TOF中断请求不会丢失。复位清

	零TOF。对TOF写1无效。 TPM计数器未达到模块值或溢出 TPM计数器溢出
6 TOIE	定时器溢出中断使能。这个读写位使能TPM溢出中断。如果TOIE被置位，当TOF等于1中断产生。复位清除TOIE。 TOF中断受限（用软件设置） TOF中断使能
5 CPWMS	中心对齐PWM选择位。若在当前，这个读写位选择CPWM操作模式。默认情况下对输入捕捉，输出比较和边对齐PWM功能来说TPM在计数值上升模式中操作。设置CPWMS重新配置TPM在可调值技术模式下完成CPWM功能。复位清零CPWMS。所有通道可作输入捕捉，输出比较，边对齐PWM模式，由MSnB: MsnA控制每个通道的状态和控制寄存器中的相应位来选择 所有通道运行在中心对齐PWM模式下
4-3 CLKS[B:A]	时钟源选择位，如表10-5所示，这2个位用来禁止TPM系统或选择三个时钟源中的一个来驱动计数器分频器。固定系统时钟源在带有一个基于系统PLL时钟的系统中是唯一有效的。若没有PLL时钟，固定系统时钟与总线速率时钟相同。外部时钟源通过TPM模块同步总线时钟，定系统时钟源通过片上同步电路同步总线时钟。当有PLL时钟但无效时，固定系统时钟与总线速率时钟相同。
2-0 PS[2:0]	分频因子选择位。这三位选择8个分割因子中的一个用来分割TPM时钟输入如表10-6所示。在任意时钟源同步或时钟源选择后分频器被定位，所以它影响用来驱动TPM系统的时钟源。在新值被更新到寄存器中的位之后，新的分频因子将对下个系统时钟周期产生影响。

表 10-5 TPM 时钟源选择

CLKSB:CLKSA	TPM时钟源分频输入
00	无时钟选择
01	总线速率时钟
10	固定系统时钟
11	外部时钟源

表 10-6 分频因子选择

PS2:PS1:PS0	分频因子
000	1
001	2
010	4
011	8
110	16
101	32
110	64
111	128

10.5.2 TPM计数器寄存器（TPMxCNTH:TPMxCNHTL）

这两个只读 TPM 计数器寄存器包含其高位和地位字节内容。读（TPMxCNTH 或 TPMxCNHTL）中的一个字节时把两字节内容都锁存到一个缓冲区直到剩下的一个字节内容也被读出。这允许 16 位连读使得更方便于各种编译器编译。这种一致机制是通过一个 MCU 复位或任意写定时器状态和控制寄存器（TPMxSC）自动重新开始的。复位清空 TPM 计数器寄存器。写任意值到 TPMxCNTH 或 TPMxCNHTL 同样清空 TPM 计数器（TPMxCNTH:TPMxCNHTL）和复位一致机制，不管是否涉及在写数据。

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

写	任意写TPMxCNTH将清零该16位计数器							
复位	0	0	0	0	0	0	0	0

图 10-8TPM 计数器寄存器高位（TPMxCNTH）

	7	6	5	4	3	2	1	0
读	位7	6	5	4	3	2	1	位0
写	任意写TPMxCNTL将清零该16位计数器							
复位	0	0	0	0	0	0	0	0

图 10-9 TPM 计数器寄存器低位（TPMxCNTL）

当 BDM 活跃，定时器计数器被暂停（这个值将被用户读取）；当 BDM 转为活跃时一致机制被暂停，缓冲区保持在原来的状态即使计数器的部分或全部内容被读取在 BDM 活跃时。这保证了当 BDM 变活跃时如果用户正在读 16 位寄存器时，在返回到正常执行状态下它将会读 16 位的其它相关值。在 BDM 模式下，写任意值到 TPMxSC，TPMxCNTH，TPMxCNTL 寄存器复位 TPMxCNTH:L 寄存器读一致机制，不管是否涉及在写数据。

10.5.3 TPM计数器模寄存器（TPMxMODH:TPMxMODL）

该读写 TPM 模块寄存器包含其模块值。当 TPM 计数器达到模块值之后，TPM 计数器在下个时钟重新开始从 0X0000 计数，溢出标志（TOF）被置位。写 TPMxMODH 或 TPMxMODL 抑制了 TOF 位和溢出中断直到其他字节被写完。复位设置 TPM 计数器模块寄存器为 0X0000 由一个自运行定时器计数器产生（模块禁止）。写（TPMxMODH:TPMxMODL）其中的一个字节锁存该值到一个缓冲区里，同时寄存器被依据 CLKSB:CLKSA 位值决定的写缓冲区值所更新，所以：

- 如果（CLKSB:CLKSA=0:0），则寄存器在第二个字节被写后更新
- 如果（CLKSB:CLKSA!=0:0）则寄存器在两个字节被写后更新，TPM 计数器从（TPMxMODH:TPMxMODL - 1）改变为（TPMxMODH:TPMxMODL）。如果 TPM 计数器是一个自运行计数器，则更新发生在 TPM 计数器值从 0XFFFE 变为 0XFFFF。

该闭锁机制可通过写 TPMxSC 地址（不论 BDM 是否活跃）手动复位。当 BDM 活跃时，一致机制被暂停（除非通过写 TPMxSC 寄存器来复位），当 BDM 活跃，这些缓冲锁保持在原来的状态，即使一个或两个半模块寄存器被写但 BDM 却是活跃的。任何越过缓冲锁存写模块寄存器和直接写模块寄存器时 BDM 活跃。

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

图 10-10 TPM 计数器模寄存器高位（TPMxMODH）

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

图 10-11 TPM 计数器模寄存器低位（TPMxMODL）

在写 TPM 模寄存器前复位 TPM 计数器以避免混淆计数器第一次溢出发生的时间。

10.5.4 TPM通道n状态和控制寄存器（TPMxCnSC）

TPMxCnSC 包含通道中断状态标志和控制位用来配置中断使能，通道配置和引脚功能。

bits	7	6	5	4	3	2	1	0
读	CHnF	ChnIE	MSnB	MsnA	ELSnB	ELSnA	0	0
写	0							
复位	0	0	0	0	0	0	0	0

 =未定义的

图 10-12TPM 通道 n 状态和控制寄存器（TPMxCnSC）

表 10-7TPMxCnSC 位描述

位	描述
7 CHnF	通道 n 标志。当通道 n 为输入捕捉通道时，且一个活跃边沿发生在通道 n 引脚上是该读写位被置位。当通道 n 为输出比较或边沿/中心对齐 PWM 通道时，当 TPM 计数器的值等于 TPM 通道 n 寄存器的值时 CHnF 被置位。当通道 n 为边对齐/中心对齐 PWM 通道且占空比被设为 0%或 100%时，即使当 TPM 计数器的值等于 TPM 通道 n 寄存器的值时，CHnF 也不会被置位。当 CHnF 被置位且中断允许（ChnIE=1）时一个相应的中断请求产生。当 CHnF 被置位时通过读 TPMxCnSC 清空 CHnF，然后置 CHnF 为逻辑 0。在清空序列完成前如果另一个中断请求发生，该序列复位，所以在清空较早 CHnF 序列完成后 CHnF 保持置位。若这些发生了则一个 CHnF 中断请求不会丢失直到清空较早的 CHnF。复位清空 CHnF 位。CHnF 置逻辑 1 不产生影响。 0 无输入捕捉或输出比较事件发生在通道 n 上 1 输入捕捉或输出比较事件发生在通道 n 上
6 ChnIE	通道 n 中断使能位。该读写位使能通道 n 的中断。复位清空 ChnIE。 0 通道 n 中断请求禁止 1 通道 n 中断请求允许
5 MSnB	TPM 通道 n B 模式。当 CPWMS=0, MSnB=1 时配置 TPM 通道 n 为边对齐 PWM 模式。参阅表 10-8 中通道模式摘要和组织控制。
4 MsnA	TPM 通道 n A 模式。当 CPWMS=0, MSnB=0 时，MsnA 配置 TPM 通道 n 为输入捕捉模式或输出比较模式。参阅表 10-8 中通道模式摘要和组织控制。 注：在转变为输入捕捉模式之前如果 若相关端口引脚在至少两个总线时钟周期内不稳定将会出现边沿触发的意想不到的迹象。
3-2 ELSnB ELSnA	沿/电平选择位。这取决于定时器通道的操作模式通过 CPWMS:MSnB:MsnA 来设置并在表 10-8 中显示，这些位选择输入边沿的极性来触发一个输入捕捉事件，选择电平将驱动一个输出比较匹配或选择 PWM 输出极性。通过设置 ELSnB:ELSnA=0:0 配置相关定时器引脚仅作为通用 I/O 引脚。这种功能用于暂时禁止输入捕捉通道或当相关定时器通道被设为不需使用引脚的软定时器时使定时器引脚作通用 I/O 脚使用。

表 10-8 模式、边沿和电位选择

CPWMS	MSnB:MsnA	ELSnB:ELSnA	模式	配置
X	XX	00	引脚不用于TPM转换为通用I/O或其它外围控制	
CPWMS	MSnB:MsnA	ELSnB:ELSnA	模式	配置
0	00	01	输入捕捉	仅在上升沿捕捉

		10		仅在下降沿捕捉
		11		上升沿或下降沿捕捉
	01	01	输出比较	比较时锁存输出
		10		比较时清空输出
		11		比较时设置输出
	1X	10	边对齐PWM	高保真脉宽（比较时清空输出）
		X1		低保真脉宽（比较时设置输出）
1	XX	10	中心对齐PWM	高保真脉宽（比较时清空输出）
		X1		低保真脉宽（比较时设置输出）

10.5.5 TPM通道值寄存器（TPMxCnVH : TPMxCnVL）

这些读写寄存器包含输入捕捉功能被捕获的 TPM 计数器值或输出比较的值或 PWM 功能。复位清空通道寄存器。

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

图 10-13 TPM 通道值寄存器高位（TPMxCnVH）

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

图 10-14TPM 通道值寄存器低位（TPMxCnVL）

在输入捕捉模式下，在读（TPMxCnVH : TPMxCnVL）其中任一字节时把它所存到一个缓冲里直到另外一半被读出。当 TPMxCnSC 寄存器被写（BDM 模式是否活跃）时，这个锁机制也可被复位（变为非锁态）。任何写通道寄存器在输入不住模式下将被忽略。当 BDM 活跃，该一致机制被暂停（除非通过写 TPMxCnSC 寄存器复位）当 BDM 变为活跃，该缓冲锁存保持在原状态，即使部分或全部通道寄存器被读。这确保了当 BDM 变为活跃时用户正在读 16 位寄存器时，在返回到正常执行后它将从读取 16 位值的另一半。在 BDM 模式下从 TPMxCnVH : TPMxCnVL 寄存器读取的值是寄存器的值而不是缓冲区的值。在输出比较或 PWM 模式下，写（TPMxCnVH : TPMxCnVL）其中之一并锁存到缓冲区。在两个都被写完后，根据 CLKS B:CLKS A 位的值和所选的模式被转换成一个 16 位值放入到定时器通道寄存器，所以：

- 如果（CLKS B:CLKS A=0:0）当第二个字节被写入时寄存器被更新。
- 如果（CLKS B:CLKS A!=0:0 且在输出比较模式下）在第二个字节被写后和在 TPM 计数器下个变化时（结束分频计数）寄存器被更新。
- 如果（CLKS B:CLKS A!=0:0 在 EPWM 或 CPWM 模式下），在两个字节被写后，TPM 计数器从（TPMxMODH : TPMxMODL - 1）变为（TPMxMODH : TPMxMODL），寄存器被更新。

如果 TPM 计数器是自运行计数器则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时发生更新操作。

该锁机制可以通过写 TPMxCnSC 寄存器（不论 BDM 是否活跃）进行手动复位。该锁机制允许写入 16 位值以升序或降序方便各种编译器执行。当 BDM 活跃，该一致机制被暂停（除非通过写 TPMxCnSC 寄存器复位）当 BDM 变为活跃，该缓冲锁存保持在原状态，即使部分或全部通道寄存器被写。任何越过缓冲锁存写通道寄存器和直接写通道寄存器时 BDM 活跃。在 BDM 活跃时且恢复正常执行，写入通道寄存器的值被用于 PWM 和输出比

较操作。当 BDM 活跃，写通道寄存器不需一致序列的干预。在一致机制被完全实行，用用户写入缓冲的值（BDM 不活跃时）来更新通道寄存器。

10.6 功能描述

所有的 TPM 功能与一个核心 16 位计数器有关，它允许灵活的选择时钟源和分频因子。同时也有一个 16 位模寄存器和它相关。CPWMS 控制位为 TPM 的所有通道选择中心对齐 PWM（CPWMS=1）或通用定时功能（CPWMS=0），该功能可以独立配置使之工作在输入捕捉，输出比较或边对齐 PWM 模式下。该 CPWMS 控制位被定位在主 TPM 状态和控制寄存器因为它对所有 TPM 通道有影响和核心计数器操作方式。（在 CPWM 模式下，作为通用计数器功能计数器更可能变为一个增/减模式）随后部分描述了核心计数器和每个定时器操作模式（输入捕捉，输出比较，边对齐 PWM，中心对齐 PWM）。因为引脚操作的细节和中断活跃取决于操作模式，这些议题将会在相关模式说明部分说明。

10.6.1 计数器

所有定时器功能是建立在核心16位计数器基础之上的（TPMxMODH : TPMxMODL）。这部分将讨论时钟源的选择，结束计数溢出，计数增/减，计数器复位手册。

10.6.1.1 计数器时钟源

该两位，CLKSB:CLKSA，在该定时器状态和控制寄存器（TPMxSC）选择三个可能时钟源中的一个或OFF（能有效禁止TPM）。见表10-5。在任何MCU复位后，CLKSB:CLKSA=0:0 没有时钟源被选择，TPM在低功耗状态。这些控制位可在任意时间被读或写和禁止定时器（写00到CLKSB:CLKSA）不影响计数器或其它定时器寄存器中的值。

表 10-9 TPM 时钟源选择

CLKSB:CLKSA	用于分频输入的TPM时钟源
00	无时钟选择（TPM计数器禁止）
01	总线速率时钟
10	固定系统时钟
11	外部时钟源

总线速率时钟是 MCU 的主要系统总线时钟。该时钟源无需同步因为该时钟用于所有内部 MCU 活动包括 CPU 和总线操作。在不带 PLL 电路的 MCU 中或该 PLL 电路不可用时，固定系统时钟源和总线速率时钟源相同，不需要使用同步器。当前使用 PLL 电路时，在二分频时钟源和定时器计数器之间需要一个同步装置，所以计数器转换和总线时钟转换相对齐。一个同步装置将被用在芯片层使与晶振相关的时钟和总线时钟同步。外部时钟可被接到 TPM 通道的任意脚。该时钟源通常过一个同步装置来确保计数器转换和总线时钟转换能很好的对齐。总线速率时钟驱动同步装置；因而为满足奈奎斯特定理（Nyquist）采集信号的要求，外部时钟源的频率不得比总线速率时钟的四分之一更快。理想情况下它们是一样的。当外部时钟源复用 TPM 通道引脚时，该引脚不能被用于其它通道的定时功能。例如，当 TPM 通道 0 引脚被用作定时器外部时钟源时，再把它配置成输入捕捉是不明智的。（用户应尽量避免）该 TPM 通道将被用于输出比较模式中作软件定时功能（引脚控制设置不影响 TPM 通道引脚）。

10.6.1.2 计数器溢出和模块复位

一个中断标志和使能与 16 位核心计数器相关。TOF 是一个软件可访问的标识来说明定时器计数器溢出情况。该使能信号在软件查询（TOIE=0 即无硬件中断产生）或中断驱动操

作（TOIE=1 即一个一个硬件中断产生）做出选择。

该条件导致 TOF 设置取决于 TPM 是否配置为中心对齐 PWM 模式（CPWMS=1）。在简单模式下，没有模数限制，TPM 不是中心对齐 PWM 模式。在这种情况下，该 16 位计数器从 0X0000 计数到 0XFFFF，在下个计数时钟溢出为 0X0000。当转换从 0XFFF 变为 0X000 时 TOF 置位。当 TPM 是在中心对齐 PWM 模式下时（CPWMS=1），当计数器达到预设在模数寄存器中的值改变计数方向时，TOF 置位。这依据于 PWM 周期末尾情况。

10.6.1.3 计数模式

该核心定时器计数器有两种计数模式。当选择中心对齐 PWM 模式时（CPWMS=1），计数器在增/减模式下操作。在其它情况下计数器作一个单增计数器操作。作为一个单增计数器，定时器计数器从 0X0000 计数直到终点值然后再从 0X0000 计数。该终端计数是 0XFFFF 或是 TPMxMODH : TPMxMODL 的一个模数值。

当被指定为中心对齐 PWM 模式操作时，定时器计数器从 0X0000 计数直到终点值然后再下降到 0X0000 循环计数。0X0000 和终点计数值都是正常长度计数值（一个定时器周期长）。在该模式下，定时器溢出标志（TOF）在终点计数周期结尾（计数变为下个更小的计数值时）被置位。

10.6.1.4 手动计数器复位

在任何时候核心定时器计数器可通过写任意值到 TPMxCNTH : TPMxCNTL 其中之一来手动复位。用这种方式复位计数器同时也复位了一致机制以防止在复位计数前仅读取部分计数器值。

10.6.2 通道模式选择

若 CPWMS=0，通道 n 状态和控制寄存器中的 MSnB 和 MSnA 控制位决定相关通道操作的基本模式。包含输入捕捉，输出比较和边对齐 PWM。

10.6.2.1 输入捕捉模式

有输入捕捉功能时，TPM 能够捕捉一个外部事件发生的时间。当一个活跃边沿发生在输入捕捉通道引脚上时，TPM 锁存其计数器内容到通道值寄存器（TPMxCnVH : TPMxCnVL）。上升沿，下降沿或任意边沿可被选作活跃边沿触发一个输入捕捉。在输入捕捉模式下，TPMxCNTH : 和 TPMxCNTL 寄存器只读。

当 16 位捕捉寄存器任一半被读时，另一半被锁存到缓冲中来支持 16 位一致存取。该一致序列可通过写通道状态和控制寄存器（TPMxCnSC）来手动复位。

一个输入捕捉事件设置一个标志位（CHnF）可产生一个 CPU 中断请求。

当在 BDM 模式下时，输入捕捉功能可由用户来配置。当一个外部事件发生时，TPM 锁存其计数器内容（在 BDM 模式下被暂停计数）到通道值寄存器并设置标志位。

10.6.2.2 输出比较模式

有输出比较功能时，TPM 可以根据预设的位置，极性，周期，频率来产生定时脉冲。当计数器达到输出比较通道中通道值寄存器中的值时，该 TPM 可置位，清空或锁存通道引脚。

在输出比较模式下，仅在 16 位寄存器的两个 8 位都被写后，参考 CLKSb: CLKSA 位的值，数值可以传递到相应定时器通道寄存器。所以：

- 如果（CLKSB: CLKSA=0:0），当第二个字节被写时寄存器被更新
- 如果（CLKSB: CLKSA!=0:0），在第二个字节被写后 TPM 计数器下个改变时寄存

器被更新。

该一致序列可通过写通道状态和控制寄存器（TPMxCnSC）来手动复位。

一个输出比较事件设置一个标志位（CHnF）可产生一个 CPU 中断请求。

10.6.2.3 边对齐PWM模式

这种 PWM 输出类型使用定时器计数器单增计数模式（CPWMS=0）且当 T P M 中的其它通道被配置为输入捕捉或输出比较功能时被使用。该 P W M 信号周期由模数寄存器值（TPMxMODH : TPMxMODL）加 1 决定。占空比由设置在定时器通道寄存器（TPMxCnVH : TPMxCnVL）值来决定。该 P W M 信号的极性由 ELSnA 控制位来决定。1% 和 100% 占空比情形是可能的。

在 T P M 通道值寄存器中的输出比较值决定了 P W M 信号的脉宽（占空比）（图 10-15）。模数溢出和输出比较之间的时间为脉冲宽度。如果 ELSnA=0，计数器溢出强制 PWM 信号为高，输出比较强制 PWM 信号为低。如果 ELSnA=1，计数器溢出强制 PWM 信号为低，输出比较强制 PWM 信号为高。

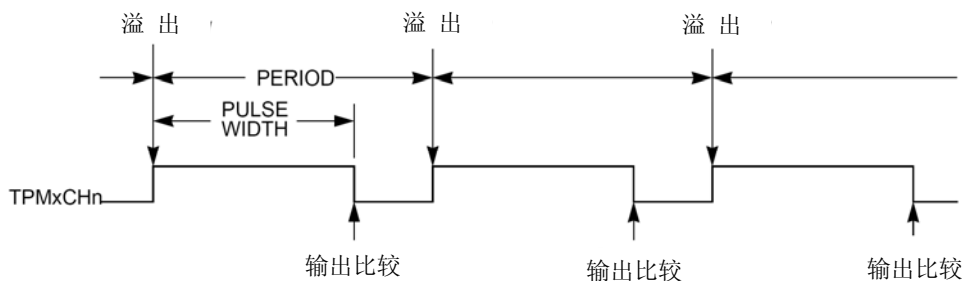


图10-15 PWM周期和脉冲宽度

当通道值寄存器被设为 0X0000 时，占空比为 0%。通过设置定时器寄存器（TPMxCnVH : TPMxCnVL）为一个大于通过模数设置的值可达到 100% 占空比。这表明为了获得 100% 占空比模数设置必须小于 0XFFFF。

因为 TPM 可用于 8 位 MCU，定时器通道寄存器中的设置被放到缓冲以确保 16 位一致更新和避免不可预料的脉宽。写 TPMxCnVH 和 TPMxCnVL 寄存器实际上是写缓冲寄存器。在边对齐 PWM 模式下，参考 CLKSb: CLKSA 位的值，数值可以传递到相应定时器通道寄存器。所以：

- 如果（CLKSB:CLKSA=0:0），当第二个字节被写时寄存器被更新
- 如果（CLKSB:CLKSA!=0:0），在两个字节都被写后 TPM 计数器从（TPMxMODH : TPMxMODL -1）变为（TPMxMODH : TPMxMODL）寄存器被更新。如果 TPM 计数器是一个自运行计数器则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时更新被执行。

10.6.2.4 中心对齐PWM模式

这种 PWM 输出类型使用定时器计数器增/减计数模式（CPWMS=1）。当周期是由 TPMxMODH TPMxMODL 中的值决定时，TPMxCnVH : TPMxCnVL 中的输出比较值决定了 PWM 信号的脉宽（占空比）。TPMxMODH TPMxMODL 应该被保持在 0X0000 到 0X7FFF 范围之间，因为这个范围之外的值易产生不确定结果。ELSnA 将决定 CPWM 输出的极性。

脉宽=2*（TPMxCnVH : TPMxCnVL）

周期=2*（TPMxMODH TPMxMODL）；TPMxMODH TPMxMODL=0X0001- 0X7FFF

如果通道值寄存器 TPMxCnVH : TPMxCnVL 为零或负数（位 15 置位），占空比将为 0%。如果 TPMxCnVH : TPMxCnVL 为正数值（位 15 清空）且大于模数设定值（非负），占空比

为 100% 因为绝不会出现占空比相比。这表明由模数寄存器设定的可用周期范围是从 0X0001 到 0X7FFE。这不是一个重要的限制。产生结果的周期将比通常应用需要的时间要长。TPMxMODH TPMxMODL=0X0000 是一个特定情况将不会用于中心对齐 PWM 模式。当 CPWMS=0 时，这种情况对应于计数器自运行从 0X0000 到 0XFFFF，但当 CPWMS=1 时，为了改变计数方向从单增计数转为单减计数该计数器需要在大于 0X0000 的某处给模数寄存器一个有效的匹配。

TPM 通道寄存器中的输出比较值决定 CPWM 信号（图 10-16）的脉宽（占空比）。如果 ELSnA=0，当计数增大强制 CPWM 输出信号为低时一个比较产生，当计数减小强制输出为低时一个比较产生。计数器计数增加直到它达到 TPMxMODH TPMxMODL 中的模数设定值，然后计数下降直至为零。这将设置该周期等于两倍 TPMxMODH TPMxMODL。

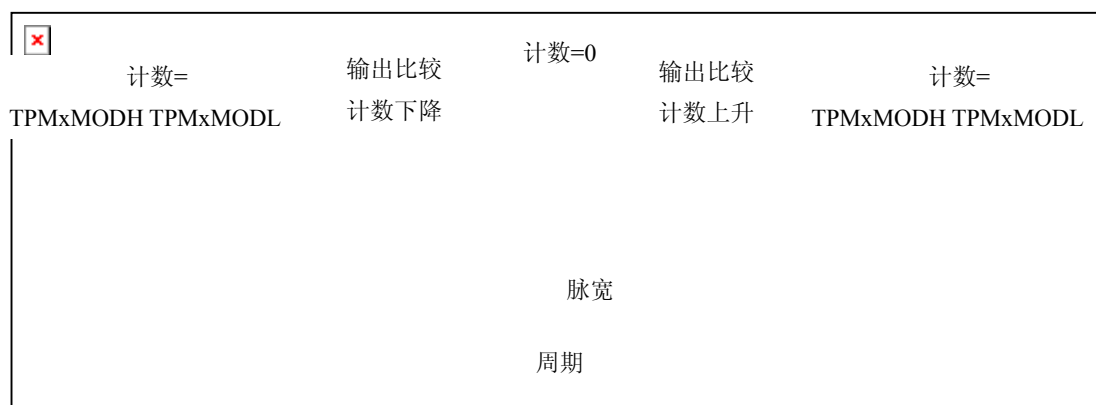


图10-16CPWM周期和脉冲宽度（ELSnA=0）

中心对齐 PWM 输出比边对齐 PWMs 模式产生更少的噪音因为较少的 I/O 引脚转换被列队在同一个系统时钟边沿。这种 PWM 类型也需要电机驱动的某些类型。

当计数器操作在增/减技术模式时，输入捕捉，输出比较和边对齐 PWM 功能无意义，所以这表明当 CPWMS=1 时所有带一个 TPM 的活跃通道必须被用在 CPWM 模式下。

TPM 可被用在 8 位 MCU。定时器通道寄存器中的设置被缓存以确保 16 位一致更新和避免产生不可预料的 PWM 脉宽。写 TPMxMODH、TPMxMODL、TPMxCnVH 和 TPMxCnVL 实际上就是写缓冲寄存器。

在中心对齐 PWM 模式下，根据 CLKSb: CLKSA 位的值用写缓冲里的值来更新 TPMxCnVH:L 寄存器。所以：

- 如果 (CLKSB: CLKSA=0:0)，当第二个字节被写时寄存器被更新
- 如果 (CLKSB: CLKSA!=0:0)，在两个字节都被写后 TPM 计数器从 (TPMxMODH : TPMxMODL - 1) 变为 (TPMxMODH : TPMxMODL) 寄存器被更新。如果 TPM 计数器是一个自运行计数器则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时更新被执行。

写 TPMxSC 取消了写 TPMxMODH 和/或 TPMxMODL 的任意值，复位模数寄存器一致机制。写 TPMxCnSC 取消了写通道值寄存器的值，复位 TPMxCnVH : TPMxCnVL 的一致机制。

10.7 复位概述

10.7.1 概要

无论何时 MCU 复位产生 TPM 即复位。

10.7.2 复位操作描述

复位清零 TPMxSC 寄存器并禁止 TPM 时钟和定时器溢出中断（TOIE=0）。CPWMS，MSnB、MSnA、ELSnB、ELSnA 被清零，为不带 I/O 引脚逻辑的相关引脚的输入捕捉操作配置所有 TPM 通道（所以所有与 TPM 相关的 MCU 引脚恢复为通用 I/O 引脚）。

10.8 中断

10.8.1 概述

若核心计数器溢出 TPM 产生一个可选择中断且每个通道也产生中断。该通道中断的方式取决于每个通道的操作模式。如果通道被配置为输入捕捉，若每次被选中的输入捕捉边沿被确认时中断标志被置位。若果通道被配置为输出比较或 PWM 模式，若核心定时器计数器等于 16 位通道值寄存器值中断被置位。

所有 TPM 中断被列在表 10-10 包括中断名称，中断使能名称它可阻塞离开 TPM 中断请求和通过独立中断处理逻辑获得确认。

表 10-10 中断摘要

中断名称	中断使能名称	触发源	描述
TOF	TOIE	计数器溢出	每次定时器计数器达到它的终点计数值时置位（可转向下个计数值通常为0X0000）
CHnF	ChnIE	通道事件	一个输入捕捉或输出比较事件发生在通道n

TPM 模块将提供一个高保真中断信号。中断向量和优先级由终端模块芯片集成时间决定所以可参考用户中断模块指南或芯片完整文件细节。

10.8.2 中断操作描述

对于每个在 TPM 的中断源，一个标志位被建立以确定中断情形如定时器溢出，通道输入捕捉，或输出比较事件。该标志位可通过软件读取以确定事件已发生或相关使能位被设置来允许硬件中断产生。当中断使能位被置位，无论何时相关中断标志等于 1 产生一个固定中断。用户软件必须执行一些步骤序列在终端服务例程返回之前清空中断标志。

TPM 中断标志通过一个两步处理来清空中断标志即当标志位为 1 时读出它然后将它置 0。如果在这两步处理之间一个新事件被检测到，该序列被复位，在第二步后中断标志位被保持置位以避免丢失新事件。

10.8.2.1 定时器溢出中断（TOF）描述

TOF 中断操作的细节和方式略有不同取决于 TPM 系统操作的模式（通用定时功能对应于中心对齐 PWM）。该标志位可通过上述两步处理来清零。

10.8.2.1.1 通常情形

通常情况下当定时器计数器从 0XFFFF 变为 0X0000 时 TOF 被置位。当 TPM 未被配置

为中心对齐 PWM 模式（CPWMS=0）时，在定时器计数值从终点值（模数寄存器中的值）变为 0X0000 时 TOF 置位。这种情形为计数器溢出一般方式。

10.8.2.1.2 中心对齐PWM情形

CPWMS=1 时，当定时器计数器在终点值（模数寄存器中的值）处改变计数方向从增转为减时 TOF 置位。这种情形下对应一个 PWM 周期的末尾。

10.8.2.2 通道事件中断描述

通道中断方式取决于通道当前模式（输入捕捉，输出比较，边对齐 PWM。中心对齐 PWM）。

10.8.2.2.1 输入捕捉事件

当通道被配置为输入捕捉通道时，ELSnB:ELSnA 控制位选择无边沿，上升沿，下降沿或任意边沿作为触发输入捕捉事件边沿。当被选择边沿被检测到时，中断标志位被置位。该标志位可通过在 10.8.2 章节“中断操作描述”中的两步处理来清零。

10.8.2.2.2 输出比较事件

当通道被配置为输出比较通道时，每当核心定时器计数器值等于通道值寄存器 16 位值时中断标志位被置位。该标志位可通过在 10.8.2 章节“中断操作描述”中的两步处理来清零。

10.8.2.2.3 PWM结束占空比事件

对于通道被配置为 PWM 操作有两种可能性。当通道被配置为边沿对齐 PWM 时，在定时器计数器等于通道值寄存器(标记活跃占空比周期末尾)时通道标志被置位。当通道被配置为中心对齐 PWM 时在每个 PWM 周期内定时器计数器等于通道值寄存器两倍。在这种 CPWM 情形下，当定时器计数器等于通道寄存器值时在占空比周期的开始和结尾时通道标志被置位。该标志位可通过在 10.8.2 章节“中断操作描述”中的两步处理来清零。

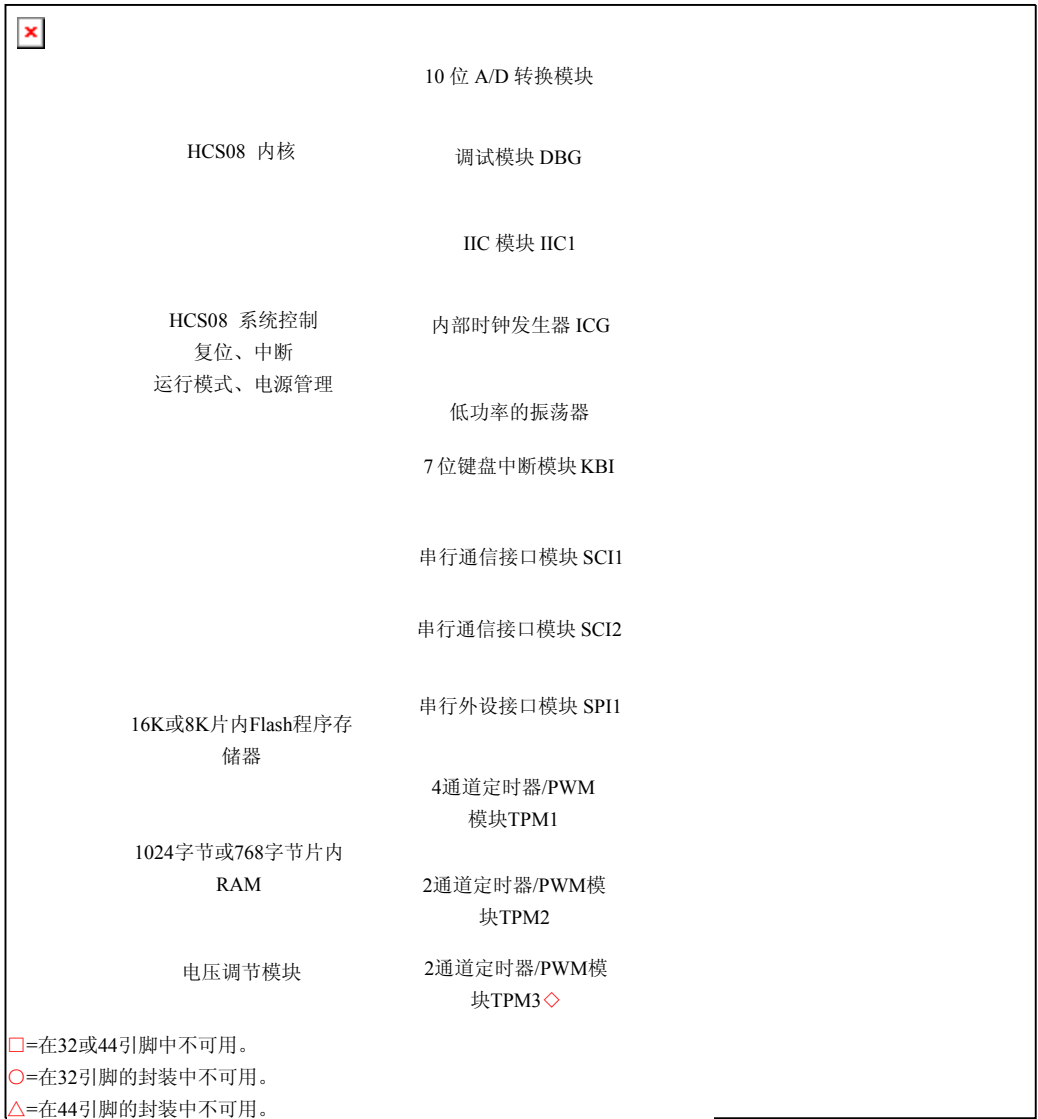
第十一章 串行通信接口（S08SCIV4）

11.1 简介

MC9S08AC16 系列包含两个独立的串行通信接口模块（SCI），也被称为通用异步收发器（UART）。通常，这些系统连接到个人计算机或工作站的 RS232 串行输入输出（I/O），但也可以和其他嵌入式控制器通信。

灵活的 13 位基于取模的波特率产生器支持超过 115.2k 波特的标准波特率。SCI 中的发送和接收使用相同的波特率，并且每个 SCI 模块都有各自独立的波特率产生器。

这个 SCI 提供了很多高级特性，这些特性很少在其他嵌入式控制器的异步串行（I/O）设备中出现。接收器使用先进的数据采样技术确保可靠的通信和噪音检测。也包括硬件奇偶校验，接收器唤醒，发送和接收的双缓冲。



□=在32或44引脚中不可用。
○=在32引脚的封装中不可用。
△=在44引脚的封装中不可用。
◇=在S9S08AWxxA设备中不可用。

注意:

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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测（IRQEDG=1），下拉使能。
3. IRQ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
4. 引脚包含集成的上拉设备。
5. PTD3,PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能（KBIPEn=1）而且上升沿被选择（KBEDGn=1），下拉使能。

图11-1 MC9S08AC16框图

11.1.1 特征

SCI 模块的特征:

- 全双工，标准的不归零（NRZ）格式。
- 发送器和接收器内各自存在双缓冲。
- 可编程的波特率（13 位取模因子）。
- 中断驱动或轮询操作：
 - 发送数据寄存器空和发送完成
 - 接收数据寄存器满

- 接收溢出，奇偶校验错误，帧错误，和干扰错误
- 接收器空闲检测
- 接收引脚上的有效边沿
- 终止检测支持 LIN
- 硬件奇偶产生和校验。
- 可编程的 8 位或 9 位的字符长度。
- 空闲线或地址标记唤醒接收器。
- 可选的 13 位终止字符产生或 11 位的终止字符检测。
- 可选的发送器输出极性。

11.1.2 操作模式

在下列模式中 SCI 操作的细节，参见第 11.3 节功能描述。

- 8 位或 9 位的数据模式
- 停止模式操作
- 轮询模式
- 单线模式

11.1.3 框图

图 11-2 说明了 SCI 发送器部分。

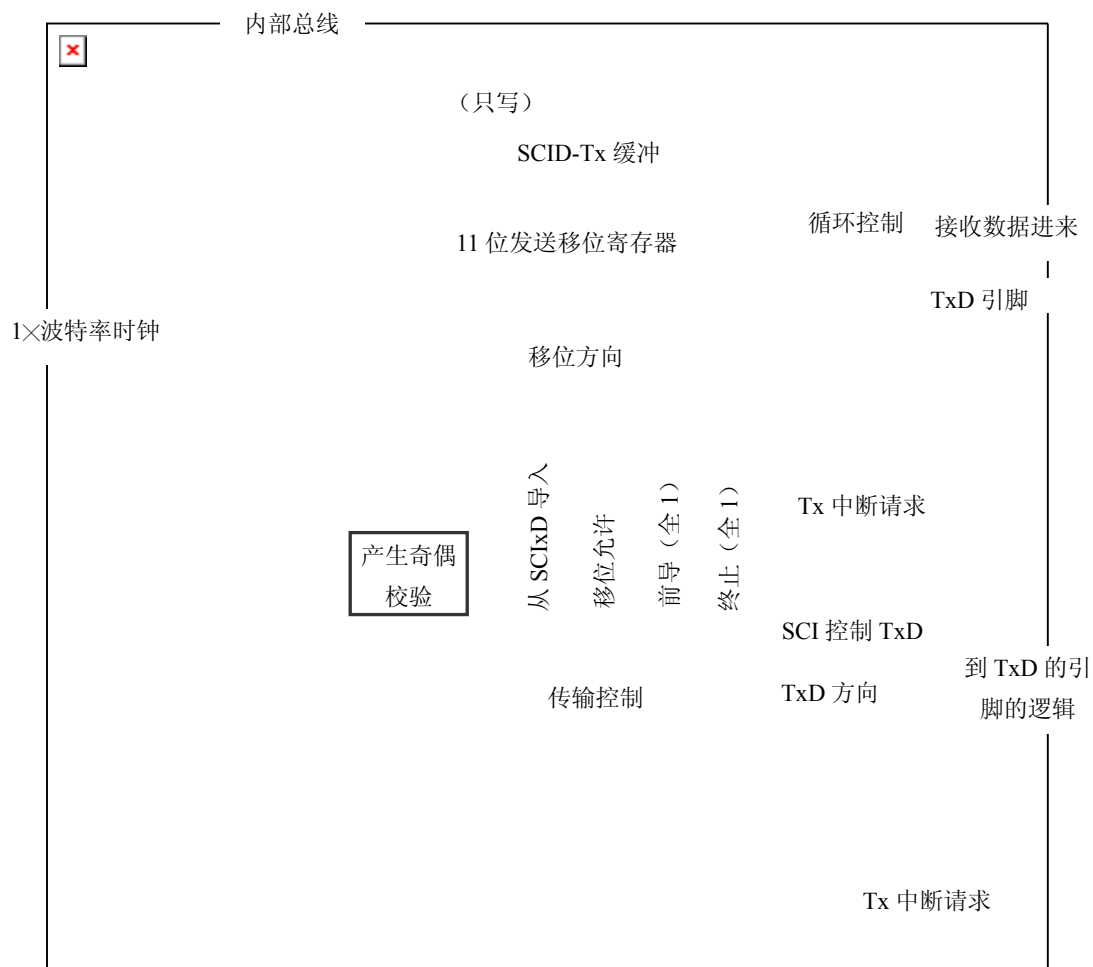


图11-2 SCI发送器框图

图 11-3 说明了 SCI 的接收器部分。

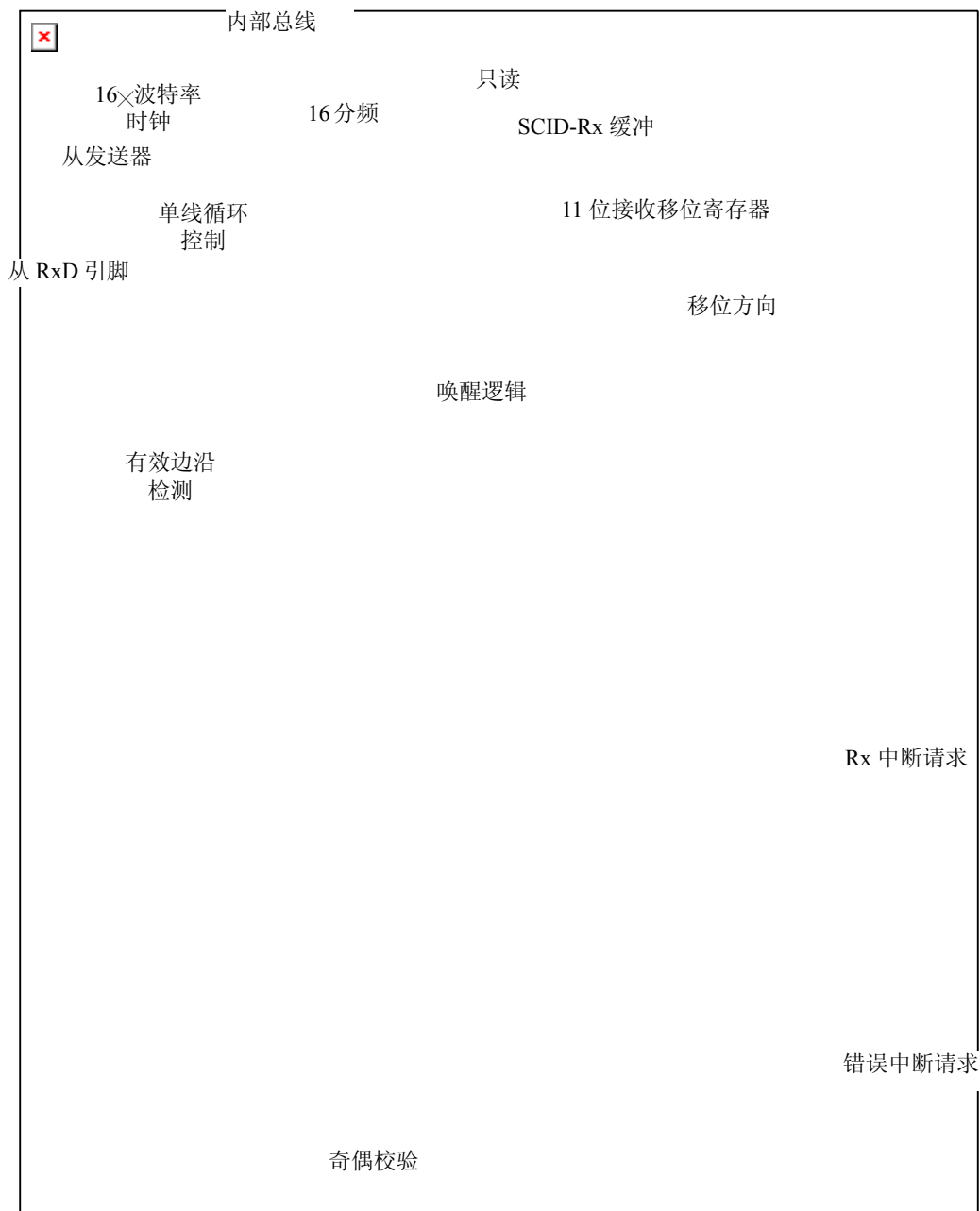


图11-3 SCI接收器框图

11.2 寄存器定义

SCI 有 8 个 8 位寄存器，分别用于控制波特率，选择 SCI 选项，报告 SCI 状态和发送/接收数据。

SCI 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。通常，Freescall 提供一个头文件把它们的名称翻译为绝对地址。

11.2.1 SCI波特率寄存器（SCIxBDH、SCIxBDL）

这两个寄存器控制产生 SCI 波特率的预分频因子。设置 13 位的[SBR12:SBR0]，先写新

值的高半部分到 SCIxBDH，然后写 SCIxBDL。直到写完 SCIxBDL，SCIxBDH 的值才改变。

SCIxBDL 复位后的值非零，所以复位后波特率产生器仍然禁止，直到接收或发送第一次被允许（SCIxC2 的 RE 或 TE 置 1）。

	第7位	6	5	4	3	2	1	第0位
读	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
写								
复位	0	0	0	0	0	0	0	0

图11-4 SCI波特率寄存器（SCIxBDH）

表 11-1 SCIxBDH 寄存器域描述

域	描述
7 LBKDIE	LIN中止检测中断使能（对于LBKDIF） 0 禁止来自LBKDIF的硬件中断（使用轮询）。 1 当LBKDIF标志为1时，发送硬件中断请求。
6 RXEDGIE	RxD输入有效边中断使能（对RXEDGIF） 0 禁止来自RXEDGIF的硬件中断（使用轮询）。 1 当RXEDGIF标志为1时，发送硬件中断请求。
4:0 SBR[12:8]	波特率分配因子——13位的SBR[12:0]记作BR，可以设置SCI波特率产生器的分频率。当BR=0时，SCI波特率产生器被禁止以减少电路消耗。BR可以从1到8191，SCI波特率=BUSCLK/(16*BR)。参见表11-2。

	第7位	6	5	4	3	2	1	第0位
读	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
写								
复位	0	0	0	0	0	1	0	0

图11-5 SCI波特率寄存器（SCIxBDL）

表 11-2 SCIxBDL 寄存器域描述

域	描述
7:0 SBR[7:0]	波特率分配因子——13位的SBR[12:0]记作BR，可以设置SCI波特率产生器的分频率。当BR=0时，SCI波特率产生器被禁止以减少电路消耗。BR可以从1到8191，SCI波特率=BUSCLK/(16*BR)。参见表11-1。

11.2.2 SCI控制寄存器1（SCIxC1）

可读写寄存器，用于控制多个 SCI 的可选功能。

	第7位	6	5	4	3	2	1	第0位
读:	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
写:								
复位:	0	0	0	0	0	0	0	0

图11-6 SCI控制寄存器1（SCIxC1）

表 11-3 SCIxC1 寄存器域描述

域	描述
7 LOOPS	循环模式选择——选择循环模式和正常的全双工模式。当LOOPS=1时，发送数据的输出在内部连接到接收器的输入。 0 禁止来自RXEDGIF的硬件中断（使用轮询）。 1 当RXEDGIF标志为1时，发送硬件中断请求。

6 SCISWAI	等待模式下SCI停止位 0 在等待模式下，SCI时钟继续运行，所以SCI可以作为唤醒CPU的中断源。 1 在等待模式下，SCI时钟停止运行。
5 RSRC	接收源选择——仅在LOOPS位为1时有意义。当LOOPS=1时，接收器输入在内部连接到TxD引脚上，RSRC决定引脚是否连接到发送器输出。 0 如果LOOPS=1，RSRC=0时，内部循环模式和SCI不使用RxD引脚。 1 单线SCI模式，TxD引脚连接到发送器输出和接收器输入。
4 M	9位或8位模式选择 0 正常——开始位+8位数据位（首先传输最低有效位LSB）+停止位。 1 接收和发送使用9位数据字符 开始位+8位数据位（首先传输最低有效位LSB）+9位数据位+停止位。
3 WAKE	接收器唤醒方式选择——更多信息参见11.3.3.2接收器唤醒操作。 0 空闲线唤醒。 1 地址符号唤醒。
2 ILT	空闲线类型选择——设置这个位为1保证字符后面的停止位和逻辑1的个数不超过10或11位（空闲线检测逻辑需要的高电平）。更多信息参见11.3.2.1空闲线唤醒 0 空闲字符位从“开始位”开始计数。 1 空闲字符位从“停止位”开始计数。
1 PE	奇偶校验允许——允许硬件奇偶校验产生和校验。当允许奇偶校验时，数据字符（第8或9位）的最高有效位（MSB）作为校验位。 0 不允许奇偶校验。 1 允许奇偶校验。
0 PT	奇偶校验类型——如果奇偶校验允许（PE=1），这个位选择奇校验或偶校验。奇校验是数据字符中1的总数（包括奇偶位）是奇数。偶校验是数据字符中1的总数（包括奇偶位）是偶数。 0 偶校验。 1 奇校验。

11.2.3 SCI控制寄存器2（SCIxC2）

可读写寄存器，用于控制多个SCI的可选功能。

	第7位	6	5	4	3	2	1	第0位
读	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
写								
复位	0	0	0	0	0	0	0	0

图11-7 SCI控制寄存器2（SCIxC2）

表 11-4 SCIxC2 寄存器域描述

域	描述
7 TIE	发送中断允许（用于TDRE） 0 TDRE的硬件中断禁止（使用轮询）。 1 当TDRE标志位是1，发送硬件中断请求。
6 TCIE	发送完成中断允许（用于TC） 0 TC的硬件中断禁止（使用轮询）。 1 当TC标志位是1，发送硬件中断请求。
5 RIE	接收中断允许（用于RDRE） 0 RDRE的硬件中断禁止（使用轮询）。 1 当RDRE标志位是1，发送硬件中断请求。
4 ILIE	空闲线中断允许（用于IDLE） 0 IDLE的硬件中断禁止（使用轮询）。 1 当IDLE标志位是1，发送硬件中断请求。

3 TE	发送允许 0 发送器关。 1 发送器开。 使用SCI发送器则TE要置1，当TE=1时，SCI强制TxD引脚作为SCI系统的输出。 当SCI设置位单线操作（LOOPS=RSRC=1），TXDIR控制单个SCI传输线（TxD引脚）的传输方向。 TE=0也可以表示一个空闲字符在排队，TE=1则传输在处理中。更多细节参见11.3.2.1发送终止和等待空闲。 当TE=0时，在引脚变回通用I/O引脚前，发送器一直控制TxD引脚，直到有数据，等待空闲或等待终止符排队完成传输。
2 RE	接收允许——当接收器关时，RxD引脚变回通用I/O引脚。如果LOOPS=1，即使RE等于1RxD也会变回通用I/O引脚。 0 接收关。 1 接收开。
1 RWU	接收唤醒控制——该位置1，SCI接收器进入待命状态，等待扫描到唤醒条件。唤醒条件是信息之间的空闲线（WAKE=0，空闲线唤醒）或者字符的最高有效位是逻辑1（WAKE=1，地址符号唤醒）。应用软件置位RWU，（通常）硬件唤醒条件自动清零RWU。更多信息参见11.3.3.2接收唤醒操作。 0 正常SCI接收操作。 1 待命的SCI接收器等待唤醒条件。
0 SBK	发送终止——向SBK写一个1然后一个0，一个终止符插入传输数据流。只要SBK=1、10或11（13或14）个逻辑0的终止符也会插入传输数据流。第二个终止符可能在软件清SBK前产生，这依赖于SBK在正传输的信息时复位和清零的时间选择。更多信息参见11.3.2.1发送终止和等待空闲。 0 正常发送操作。 1 发送对了终止字符。

11.2.4 SCI状态寄存器1（SCIxS1）

该寄存器有8个只读状态标志。写无影响。专门的软件序列（不是写该寄存器）用来清零这些状态位。

	第7位	6	5	4	3	2	1	第0位
读	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
写								
复位	1	1	0	0	0	1	0	0

图11-8 SCI状态寄存器1（SCIxS1）

表 11-5 SCIxS1 域描述

域	描述
7 TDRE	发送缓冲区空标志——TDRE在复位时置位。而且当发送数据从发送数据缓冲区传到发送移位寄存器为新字符留出空间时，TDRE也置位。SCIxS1的TDRE=1然后写SCI数据寄存器可以清零TDRE。 0 发送数据寄存器（缓冲区）满。 1 发送数据寄存器（缓冲区）空。
6 TC	发送完成标志——TC在复位时置位。而且当TDRE=1并且没有数据，前导符或终止符发送时，TC也置位。 0 发送器忙（发送数据，前导符或终止符）。 1 发送器空闲。 读SCIxS1的TC=1然后做下面的3件事情之一，TC就会自动清零。 ● 写SCI数据寄存器（SCIxD）发送新数据。 ● 改变TE从0到1插入前导符。 ● 置1SCIxC2的SBK插入终止符。

5 RDRF	接收器满标志——当字符从接受移位寄存器传输到接收数据寄存器（SCIxD）时，RDRF置位。读SCIxS1的RDRF=1然后读SCI数据寄存器（SCIxD）可以清RDRF。 0 接收数据寄存器空。 1 接收数据寄存器满。
4 IDLE	接收器空闲标志——SCI工作一段时间后，如果接收线空闲，IDLE则置位。当ILT=0时，接收器开始计数开始位后的空闲时间。所以如果接收字符全1，这些位的时间和停止位的时间达到了一个字符的逻辑高（10或11位依赖于M控制位），这会导致接收器检测空闲线，当ILT=1时，接收器直到停止位后才开始数空闲时间。所以停止位和先前字符的逻辑高不会到达一个字符逻辑高的时间。 读SCIxS1的IDLE=1然后读SCI数据寄存器（SCIxD）可以清零IDLE。IDLE清零后，直到接收到一个新的字符并且RDRF被置位，IDLE才能再次置位。即使接收线长时间保持空闲，IDLE也只会置位一次。 0 未检测到空闲线。 1 检测到空闲线。
3 OR	接收器溢出标志——当新的字符准备传到接收数据寄存器时，但是前面接收的数据还没有从SCIxD中读走，OR置位。在这种情况下，因为没有空间把这些数据移到SCIxD中，所以新的字符（和所有的相关错误信息）丢失。读SCIxS1的OR=1然后读SCI数据寄存器（SCIxD）可以清零OR。 0 未溢出。 1 接收溢出（新数据丢失）。
2 NF	噪音标志位——在接受器中使用先进的采样技术，开始位采样7次，每个数据位和停止位采样3次。在接收到数据时RDRF置位的同时，如果在帧中某个采样和其他的采样不同则标志NF置位。读SCIxS1然后读SCI数据寄存器（SCIxD）可以清零NF。 0 未检测到噪音。 1 SCIxD中接收到的字符检测到噪音。
1 FE	帧错误标志——接收器在停止位检测到逻辑0时，FE和RDRF置位。这表明接收器没有对齐字符帧。读SCIxS1的FE=1然后读SCI数据寄存器（SCIxD）可以清零FE。 0 未检测到帧错误。这并不确保帧时正确的。 1 帧错误。
0 PF	奇偶错误标志——当奇偶校验使能（PE=1）并且接收到的数据中的校验位和正确的校验位不同，PE和RDRF置位。读SCIxS1然后读SCI数据寄存器（SCIxD）可以清零PF。 0 无奇偶校验错误。 1 有奇偶校验错误。

11.2.5 SCI状态寄存器2（SCIxS2）

该寄存器有1个只读状态标志。

	第7位	6	5	4	3	2	1	第0位
读	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
写								
复位	0	0	0	0	0	0	0	0

图11-9 SCI状态寄存器2（SCIxS2）

表 11-6 SCIxS2 域描述

域	描述
7 LBKDIF	LIN终止检测中断标志——当使能LIN终止电路并且检测到一个LIN终止符，LBKDIF置位。向LBKDIF写1，LBKDIF清零。 0 未检测到LIN终止符。 1 检测到LIN终止符。
6 RXEDGIF	RxD引脚有效边沿中断标志——当RxD引脚上一个有效边沿发生时（如果RXINV=0，下降沿，如果RXINV=1，上升沿）RXEDGIF置位。向其写1，RXEDGIF清零。 0 在接收引脚上无有效边沿发生。 1 在接收引脚上有效边沿发生。

4 RXINV1 (1)	接收数据反转——该位置位则接受的数据输入的极性反转。 0 接收的数据未反转。 1 接收的数据反转。
3 RWUID	接收唤醒空闲检测——RWUID控制空闲字符是否置位IDLE位。 0 在接收待命状态（RWU=1）期间，检测到空闲字符时，IDLE不置位。 1 在接收待命状态（RWU=1）期间，检测到空闲字符时，IDLE置位。
2 BRK13	中断符长度——BRK13可以选择更长的发送终止符长度。该位的状态不影响帧错误的检测。 0 中断符长度是10字节（M=1则为11）。 1 中断符长度是13字节（M=1则为14）。
1 LBKDE	LIN终止检测使能——LBKDE可以选择更长的终止符检测长度。当LBKDE置位时，可以防止帧错误（FE）和接收数据寄存器满（RDRF）标志置位。 0 终止符的检测长度是10字节（M=1则为11）。 1 终止符的检测长度是11字节（M=1则为12）。
0 RAF	接收器有效标志——当SCI接收器检测到有效位的开始，RAF置位；当接收器检测到空闲线，RAF清零。该状态位可以用来检查MCU进入停止模式前，SCI字符是否正被接收。 0 SCI接收器空闲等待开始位。 1 SCI处于活动中（RxD输入无效）。

（1）置位RXINV反转RxD输入的各种情况：数据位，开始位和停止位，终止和空闲。

当在 LIN 系统中使用内部振荡器时，有必要增加一位检测阈。在最坏的情况下，LIN 允许的定时条件可能把 0x00 数据字符看做 10 位字节，在从机上运行比主机快 14%。这将触发正常的终止检测电路（用来检测 10 位终止标志）。当 LBKDE 位置位时，帧错误可以减少，终止检测阈从 10 位变成 11 位以防止把 0x00 数据字符错误地检测成 LIN 终止符号

11.2.6 SCI控制寄存器3（SCIxC3）

	第7位	6	5	4	3	2	1	第0位
读	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
写								
复位	0	0	0	0	0	0	0	0

图11-10 SCI控制寄存器3（SCIxC3）

表 11-7 SCIxC3 域描述

域	描述
7 R8	接收器的第九数据位——当SCI设置为9位数据时（M=1），R8可以看作SCIxD寄存器中数据的最高有效位左边的第九位接收数据。当读9位数据时，因为读SCIxD会自动完成清零标志（允许在R8和SCIxD中写入新数据），在读SCIxD之前先读R8。
6 T8	发送器的第九数据位——当SCI设置为9位数据时（M=1），T8可以看作SCIxD寄存器中数据的最高有效位左边的第九位发送数据。当写9位数据时，在写SCIxD之后整个9位值传到SCI移位寄存器，所以T8应该在写SCIxD之前先写（如果T8需要改变它的值）。如果T8不需要改变值（比如当它用于产生标记或空间奇偶校验）则不需要每次写SCIxD都重写。
5 TXDIR	单线模式TxD引脚方向——当SCI被设置为单线半双工操作时（LOOPS=RSRC=1），该位决定TxD引脚上的数据方向。 0 单线模式下TxD引脚为输入。 1 单线模式下TxD引脚位输出。
4 TXINV1	发送数据反转——该位置位则发送数据的输出的极性反转。 0 发送数据未反转。 1 发送数据反转。
3 ORIE	溢出中断使能——该位使能溢出标志（OR），产生硬件中断请求。 0 禁止OR中断（使用轮询）。 1 当OR=1时，发出硬件中断请求。

2 NEIE	噪音错误中断使能——该位使能溢出标志（NF），产生硬件中断请求。 0 禁止NF中断（使用轮询）。 1 当NF=1时，发出硬件中断请求
1 FEIE	帧错误中断使能——该位使能帧错误标志（FE），产生硬件中断请求。 0 禁止FE中断（使用轮询）。 1 当FE=1时，发出硬件中断请求
0 PEIE	奇偶校验错误中断使能——该位使能奇偶校验错误标志（PF），产生硬件中断请求。 0 禁止PF中断（使用轮询）。 1 当PF=1时，发出硬件中断请求

（1）置位TXINV反转TxD输出的各种情况：数据位，开始位和停止位，终止和空闲。

11.2.7 SCI数据寄存器（SCIxD）

该寄存器实际上是两个独立的寄存器。读得到只读接收数据缓冲区的内容，写只写发送数据缓冲区。该寄存器的读和写也和 SCI 状态标志的标识自动清零机制有关。

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

图11-11 SCI数据寄存器（SCIxD）

11.3 功能描述

SCI 是一种全双工，异步，NRZ 的串行通信。用于 MCU 和其他远程设备（包括其他 MCU）通信。SCI 包括一个波特率产生器，发送器和接收器。虽然使用相同的波特率产生器，发送器和接收器独立工作。正常工作时，MCU 监控 SCI 的状态，写数据发送，处理接收的数据。接下来描述每个 SCI 模块。

11.3.1 波特率产生

如图 11-12 所示，SCI 波特率产生器的时钟源是总线时钟。

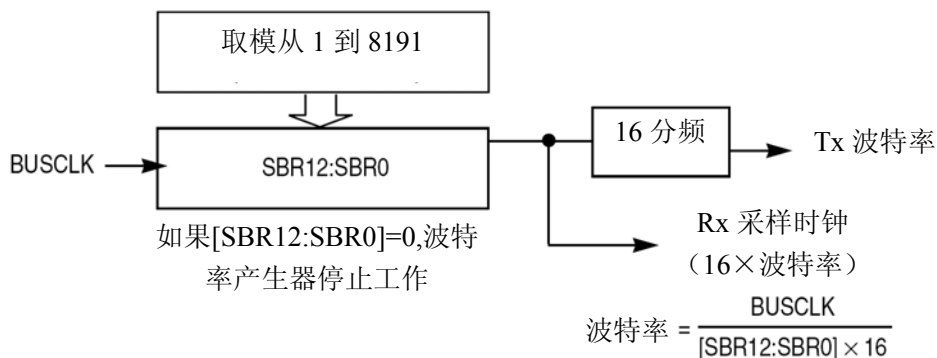


图11-12 SCI波特率产生

SCI 通信要求发送器和接收器（通常从独立的时钟源获得波特率）使用相同的波特率。所允许的波特频率由两个部分决定，接收器怎样同步开始位的有效沿和位采样怎样操作。

MCU 在每个高到低转换时都会重新同步位边界。但是在最坏的情况下，在 10 或 11 位的字符帧中没有这种转换，所以在整个字符期间任何波特率的不匹配都会积累起来。在飞思卡尔半导体的 SCI 系统中，其总线频率有一个晶振得到，所允许的波特率不匹配是 4.5%（8 位数据格式）和 4%（9 位数据格式）。虽然波特率分频因子的设置并不总是产生正好匹配标

准的波特率，但是通常偏差都在很小的百分比，可以保障可靠的通信。

11.3.2 发送功能描述

本节描述 SCI 接收器的框图和发送终止和空闲字符的功能。发送器的框图如 11-2 所示。

发送器输出（TxD）空闲状态默认是逻辑高（复位后 TXINV=0）。设置 TXINV=1，发送器输出被反转。置位 SCIxC2 的 TE，发送器使能。发送器发出一个前导符（一个空闲状态的满字符帧）。直到发送数据缓冲区的数据准备好，发送器保持空闲。写 SCI 数据寄存器（SCIxD）可以将数据存到发送数据缓冲区。

SCI 发送器的主要部分是发送移位寄存器，10 位或 11 位长（由 M 控制位决定）。在本节以后的部分，我们假定 M=0（选择正常的 8 位数据模式）。在 8 位数据模式下，移位寄存器包含 1 个开始位，8 个数据位，1 个停止位。当发送移位寄存器允许接受新的 SCI 字符时，等待在发送数据寄存器的值传到移位寄存器（由波特率时钟同步）并且发送数据空（TDRE）状态标志置位以表示 SCIxD 内的发送数据缓冲区可以写入另一个字符。

停止位移出 TxD 引脚后，如果没有数据等待发送数据缓冲区，发送器置位发送完成标志并进入空闲状态，TxD 逻辑高，等待传输字符。

写 0 到 TE 并不能立即使引脚成为通用 I/O 引脚。任何正在进行的发送过程必须先完成。这包括正在处理的数据，等待空闲字符和等待终止符。

11.3.2.1 发送终止和等待空闲

SCIxC2 的 SBK 用于发送终止符（原来用于引起电传打字机的注意）。终止符是一个字符的逻辑 0（10 位包含开始位和停止位）。置位 BRK13=1，则终止符可以达到 13 位的长度。通常，程序等待 TDRE 置位（表示信息的最后一个字符移到发送移位寄存器），然后向 SBK 位先后写 1 和 0。这样的话，只要移位寄存器可用，就会发送终止符。如果当终止符移到移位寄存器（由波特率时钟同步）SBK 仍为 1，则有另外一个终止符在等待。如果接收设备是另一个飞思卡尔半导体的 SCI，则接受的 8 位终止符全 0，并且产生帧错误（FE=1）。

当使用空闲线唤醒时，两个信息之间需要一个空闲（逻辑 1）的字符，以唤醒处于睡眠状态的接收器。通常，程序等待 TDRE 置位（表示信息的最后一个字符移到发送移位寄存器），然后向 TE 位先后写 0 和 1。这样的话，只要移位寄存器可用，就会发送空闲符。只要移位寄存器的字符没有完成（TE=0），SCI 发送器就不会释放对 TxD 引脚的控制。如果移位寄存器有完成的可能性（TE=0），设置为通用 I/O 控制，则引脚和 TxD 复用，输出逻辑 1。这确保了 TxD 线看起来像正常的空闲线，即使 SCI 在向 TE 写 0 和写 1 之间失去对引脚的控制。

终止符的长度受 BRK13 和 M 位影响，如下所示。

表 11-8 终止符长度

BRK13	M	终止符长度
0	0	10位
0	1	11位
1	0	13位
1	1	14位

11.3.3 接收功能描述

在该节中，接收器框图（图 11-3）是全面描述接收器的功能一个指南。然后，更详细地描述数据采样技术（用于得到接收器的数据）。最后，解释了两种接收器唤醒功能。

通过置位 RXINV=1，接收器输入被反转。置位 SCIxC2 的 RE，接收器使能。字符帧包

括一个开始位（逻辑 0），8（或 9）数据位（首先最低有效位），和停止位（逻辑 1）。关于 9 位数据模式的信息，参见 11.3.5.1 节“8 位和 9 位数据模式”。在以后的讨论中，我们假定 SCI 设置为正常的 8 位数据模式。

接受移位寄存器接收到停止位后，如果接收数据还未满，数据字符传到接收数据寄存器并且接收数据寄存器满（RDRF）状态标志置位。如果 RDRF 已经置位表明接收数据寄存器（缓冲区）已经满了，溢出（OR），状态标志置位并且新数据丢失。因为 SCI 接收器时双缓冲，在 RDRF 置位后和读接收数据缓冲区的数据前，程序有一个字符的时间避免接收器溢出。

当程序检测到接收数据寄存器满时（RDRF=1），读 SCIxD 可以得到接收数据寄存器的数据。RDRF 通过两步骤的序列（通常在控制接收数据的用户程序中）自动清零，更多关于标志清零的信息参见 11.3.4 节的中断和状态标志。

11.3.3.1 数据采样技术

SCI 接收器使用 16 倍的波特率时钟进行采样。接收器以 16 倍波特率采样逻辑电平，发现 RxD 串行输入引脚上的下降沿。下降沿被定义为 3 个逻辑 1 采样后的逻辑 0 采样。16 倍波特率时钟把位时间分成 16 段（从 RT1 到 RT16）。当定位了一个下降沿时，有 3 个采样（RT3、RT5、RT7）确保这是开始位而不是噪音。如果 3 个采样中不少于 2 个是 0，则接收器假定与接收字符同步。

然后接收器采样每个位时间，包括开始位和停止位（在 RT8、RT9、RT10 决定改有的电平）。逻辑电平由位时间的采样结果的多数采样决定。采样开始位时，如果 RT3、RT5、RT7 这三个中至少两个是 0，则可以认为该位为 0，即使 RT8、RT9、RT10 中的一个或全部为 1。如果在字符帧中任何一个位时间（包括开始位和停止位）的任何一个采样无法得到一致的逻辑电平，当接收的字符传到接收数据缓冲区时，噪音标志（NF）置位。

下降沿检测逻辑不断检测下降沿，如果检测到了，采样时钟同步到位时间。这提高了接收器在噪音和波特率不匹配存在时的可靠性。这不能提高最糟情况的分析，因为有一些字符在字符帧中都不会出现下降沿。

在帧错误的情况下，如果接收的字符不是终止符，查找下降沿的采样逻辑填充了 3 个逻辑 1，如此以至于几乎可以立即检测一个新的开始位。

在帧错误的情况下，直到帧错误清零，接收器都会被阻止接收新的字符。接受移位寄存器继续工作，但是如果 FE 置位，一个完成的字符不能转移到接收数据缓冲区。

11.3.3.2 接收唤醒操作

接收器唤醒是一种硬件机制，它允许 SCI 接收器忽略那些供不同 SCI 接收器使用的信息中的字符。在这个系统中，所有的接收器判断每个信息的第一个字符，只要认为信息是提供给不同 SCI 的，则将 SCIxC2 中的接受唤醒（RWU）置 1。当 RWU 置位，和接收器相关的状态（除了空闲位（IDLE），当 RWUID 位置位）禁止置位，因此消除控制不重要字符的软件开支。在信息的最后，或者是下一个信息的开始，所有的接收器自动强制 RWU 为 0，所以所有的接收器按时唤醒，查看下一个信息的第一个字符。

1. 空闲线唤醒

当 WAKE=0，接收器被设置为空闲线唤醒。在这种模式中，当接收器检测到一个满字符的空闲线电平时，RWU 自动清零。M 控制位选择 8 位或 9 位数据模式，这决定了多少个空闲位组成了一个满字符时间（加上开始位和停止位共 10 位或 11 位）。

当 RWU 为 1 和 RWUID 为 0 时，唤醒接收器的空闲条件不置位 IDLE。接收器唤醒并等待下一个信息的第一个字符（将会置位 RDRF，如果允许可以产生一个中断），当 RWUID 为 1 时，无论 RWU 是 0 还是 1，任何空闲条件置位 IDLE 标志并产生一个中断（如果允许）。

空闲线类型（ILT）控制位选择两种检测空闲线方法中的一种。ILT=0，空闲位从开始位开始计数，停止位和字符后面的逻辑 1 的个数组成了空闲线满字符。ILT=1，空闲位不是从开始位而是停止位，所以空闲线检测不受前一个信息的最后的字符的数据的影响。

2. 地址标志唤醒

当 WAKE=1，接收器被设置为地址标志唤醒。在这种模式中，当接收器检测到接收数据的大多数有意义位是逻辑 1 时（M=0，第 8 位；M=1，第 9 位），RWU 自动清零。

地址标识唤醒允许信息包含空闲位，但是需要 MSB 保留为地址帧使用。在接收到停止位并且置位 RDRF 标志前，地址帧的 MSB 逻辑 1 清零 RWU 位。在这种情况下，即使接收器在大多数字符时间都在休眠，MSB 字符仍被接收。

11.3.4 中断和状态标志

SCI 系统有三个独立的中断向量，减少了分析中断原因的软件编程。一个中断向量和发送器的 TDRE 和 TC 事件相关。另一个中断向量和接收器的 RDRF、IDLE、RXEDGIF 和 LBKDIF 事件相关。第三个向量用于 OR、NF、FE 和 PD 错误条件。通过本地中断使能掩码，这十个中断源可以独立掩码。当本地掩码被清零，禁止中断请求的产生时，这个标志可以通过软件轮询的方式得到。

SCI 发送器有两个状态标志，可以产生硬件中断请求。发送数据寄存器空（TDRE）表示什么时候发送数据缓冲区有空间可以写另一个发送字符到 SCIxD。如果发送中断使能（TIE）位置位，只要 TDRE=1，就会产生一个硬件中断请求。发送完成（TC）表示所有的数据，前导符和终止符都发送完成，发送器空闲（TxD 处于无效电平）。该标志位通常用于有调制解调器的系统，用于决定何时关闭调制解调器是安全的。如果发送完成中断使能位（TCIE）置位，只要 TC=1，就会产生一个硬件中断请求。如果 TIE 或 TCIE 本地中断掩码是 0，软件轮询可以代替硬件中断请求，可以用于检测 TDRE 和 TC 状态位。

当程序检测到接收数据寄存器满（RDRF=1），程序读 SCIxD 可以得到接收数据寄存器的数据。当 RDRF=1 并且读了 SCIxD 后，读 SCIxS1 可以清零 RDRF 标志。

当使用轮询方式时，这个顺序可以很自然的满足用户程序的一般过程。如果使用硬件中断方式，SCIxS1 必须在中断服务例程（ISR）中读。通常这必须在 ISR 中完成已检测接收错误，所以这个顺序可以自动满足。

当 RxD 线在一段时间保持空闲，IDLE 状态标志包含了可以避免重复置位的情况。当 IDLE=1 并且读了 SCIxD 后，读 SCIxS1 可以清零 IDLE。IDLE 清零后，直到接收器至少接收了一个新字符并且 RDRF 置位，才能再次置位。

如果在导致 RDRF 置位的接收字符中检测到了相应的错误，错误标志——噪音标志（NF），帧错误（FE）和奇偶校验错误标志（PF）将会和 RDRF 同时置位。在溢出情况，这些标志不置位。

如果 RDRF 置位，当一个新字符准备从接收移位寄存器到接收数据缓冲区，溢出标志（OR）置位，和数据相关的 NF、FE 或 PF 条件丢失。

在任何时候，RxD 串行数据输入引脚上的有效边沿会导致 RXEDGIF 标志置位。向 RXEDGIF 标志写 1，可以清零 RXEDGIF。这个功能不依赖于接收器被允许（RE=1）。

11.3.5 其他SCI功能

下面描述其他的 SCI 功能。

11.3.5.1 8位和9位数据模式

通过置位 SCIxC1 的 M 控制位，SCI 系统（发送器和接收器）可以被设置为 9 位数据模

式。在 9 位数据模式中，第九位数据位在 SCI 数据寄存器的最高有效位的左边。对于发送数据缓冲区，该位存储在 SCIxC3 的 T8。对于接收器，该位存储在 SCIxC3 的 T8。

为了一致地写发送数据缓冲区，在写 SCIxD 以前写 T8 位。

如果要发送的新字符的第九位的值和前一个字符相同，没有必要再写 T8。当数据从发送数据缓冲区传到发送移位寄存器中时，在数据从 SCIxD 传到移位寄存器的同时，T8 的值被复制。

9 位数据模式通常用于连接奇偶校验位，可以允许 8 位的数据加上第九位的奇偶校验位。或用于地址标志唤醒，所以第九位也可以作为唤醒位。在定制协议中，第九位也作为一个由软件控制的掩码器。

11.3.5.2 停止模式操作

在所有的停止模式中，SCI 模块的时钟暂停。

在 stop1 和 stop2 模式中，所有的 SCI 寄存器数据丢失。当从这两种停止模式中恢复时，所有的寄存器必须重新初始化。在 stop3 模式中，SCI 模块寄存器不会受影响。

在 stop3 模式中，接收输入有效边沿检测电路仍然有效，但是在 stop2 模式下无效。如果中断没有被屏蔽（RXEDGIE=1），接收输入上的一个可以将 CPU 从 stop3 模式中唤醒。

注意，因为时钟暂停，当从停止模式（仅在 stop3 模式中）中退出，SCI 模块将重新开始操作。当有字符正在被发送出或接收进 SCI 模块，软件应该确保不进入停止模式。

11.3.5.3 循环模式

当 LOOPS=1 时，同一个寄存器的 RSRC 位选择循环模式（RSRC=0）或单线模式（RSRC=1）。循环模式有时用于检测软件，独立于外部系统的连接，可以帮助分析系统问题。在这个模式中，发送器的输出在内部连接到接收器的输入，RxD 不使用，所以它作为通用 I/O 引脚。

11.3.5.4 单线操作

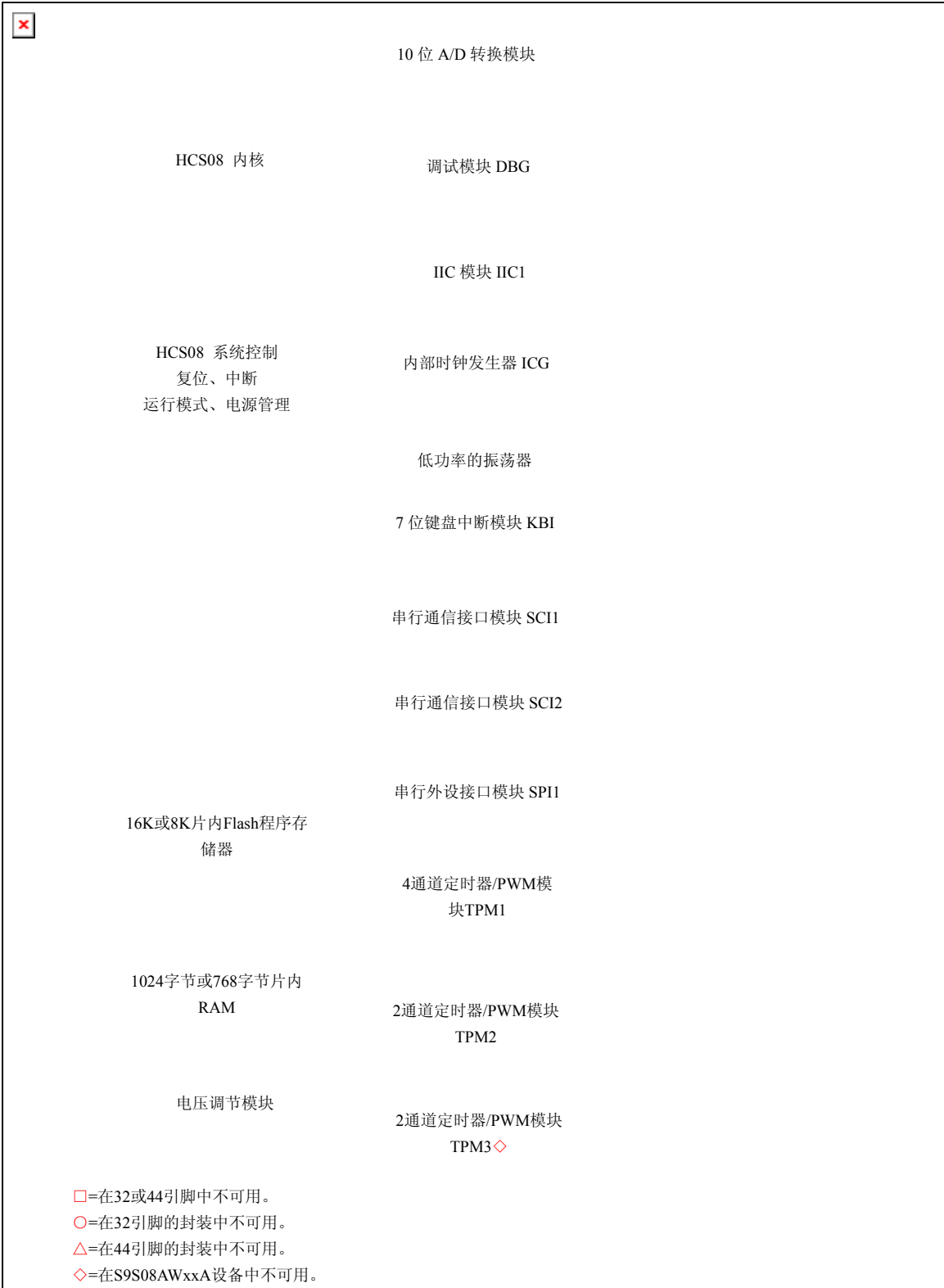
当 LOOPS=1 时，同一个寄存器的 RSRC 位选择循环模式（RSRC=0）或单线模式（RSRC=1）。单线模式半双工的串行连接。接收器在内部连接到发送器的输出 TxD 引脚。RxD 引脚不使用，作为通用 I/O 引脚。

在单线模式中，SCIxC3 的 TXDIR 位控制 TxD 引脚上数据的方向。当 TXDIR=0，TxD 引脚作为接收器的输入并且发送器和 TxD 引脚临时断开，所以一个外部设备可以向接收器发送数据。当 TXDIR=1，TxD 作为接收器的输出。在单线模式中，从发送器到接收器的内部环路使接收器接收发送器所发送的字符。

第十二章 串行外设接口（S08SPIV3）

12.1 简介

MC9S08AC16 系列含有一个串行外设接口（SPI）模块。与 SPI 功能相关的四个引脚复用端口 E 引脚 4-7。见附录 A，电气参数信息的“电气特性和定时说明”。



注：

1. 若为输入端口，端口引脚是用上拉设备软件配置
2. 若 IRQ 允许（IRQPE=1）引脚包含软件配置的上拉/下拉设备。如果上升沿检测有效（IRQEDG=1）下拉允许
3. IRQ 不需要通过钳二极管连接到 VDD。供电电压不超过 VDD。

4. 引脚包含整个上拉设备。
5. PTD3、PTD2、PTG4 包含上拉和下拉设备。当 KBI 允许时下拉有效（KBIPEn=1），上升沿检测有效（KBEDGn=1）

图 12-1 MC9S08AC16 框图

12.1.1 特征

SPI 模块特征如下：

- 主从工作方式
- 全双工或半双工
- 可编程的发送比特率
- 双缓存发送和接受
- 串行时钟相位和极性选择
- 从选择输出
- MSB 优先或 LSB 优先移位

12.1.2 框图

这部分包括框图显示 SPI 系统连接，模块内部组织，和控制主模式比特率的 SPI 始终分频器。

12.1.2.1 SPI 系统框图

图 12-2 显示两个 MCU 的 SPI 模块以主从方式连接。主设备发起所有 SPI 数据发送。在一个发送过程中，主机送出数据（在主出从入引脚上）给从机，但同时从机数据发送（在主入从出）到主机。发送有效地交换了两个 SPI 系统中 SPI 移位寄存器的数据。SPSCK 信号来自主机的一个时钟输出，是从机的一个时钟输入。从机设备必须由从机选择输出脚上的低电平来被选中。在这种系统中，主机设备配置它的 $\overline{SS}$ 引脚作为一个可选择的从机选择输出。



图 12-2 SPI 系统连接

SPI 系统最常见的用途包括连接简单的移位寄存器用于加入输入或输出端口内容或连接一个小型的外围设备如串行 A/D，D/A 转换。虽然图 12-2 显示了一个如何在两个 MCU

之间交换数据的系统，但在很多实际应用系统中在主从机之间的数据单向交换更为简单。

12.1.2.2 SPI模块框图

图 12-3 是一个 SPI 模块框图。SPI 的中心组成部分是 SPI 移位寄存器。数据写入双缓存发射器（写给 SPI1D）并在一个数据发送开始时获得数据传送给 SPI 移位寄存器。在移出一个字节数据后，数据被送入双缓存接收器（从 SPI1D 读出）在这数据可被读出。引脚多路技术逻辑控制 MCU 引脚和 SPI 模块之间的联系。

当把 SPI 作为主机，该时钟输入被发送到 SPSCCK 引脚，移位寄存器的输出发送到主出从入引脚（MOSI），移位寄存器的输入来自主入从出引脚（MISO）。

当把 SPI 作为从机，SPI 引脚作为 SPI 的时钟输入，移位寄存器输出发送到主入从出引脚（MISO），移位寄存器输入来自主出从入引脚（MOSI）引脚。

在外围 SPI 系统中，所有 SPSCCK 引脚简单地相互连接，所有主入从出引脚在一起，主出从入引脚在一起。外围设备经常用略有不同的名字命名这些引脚。

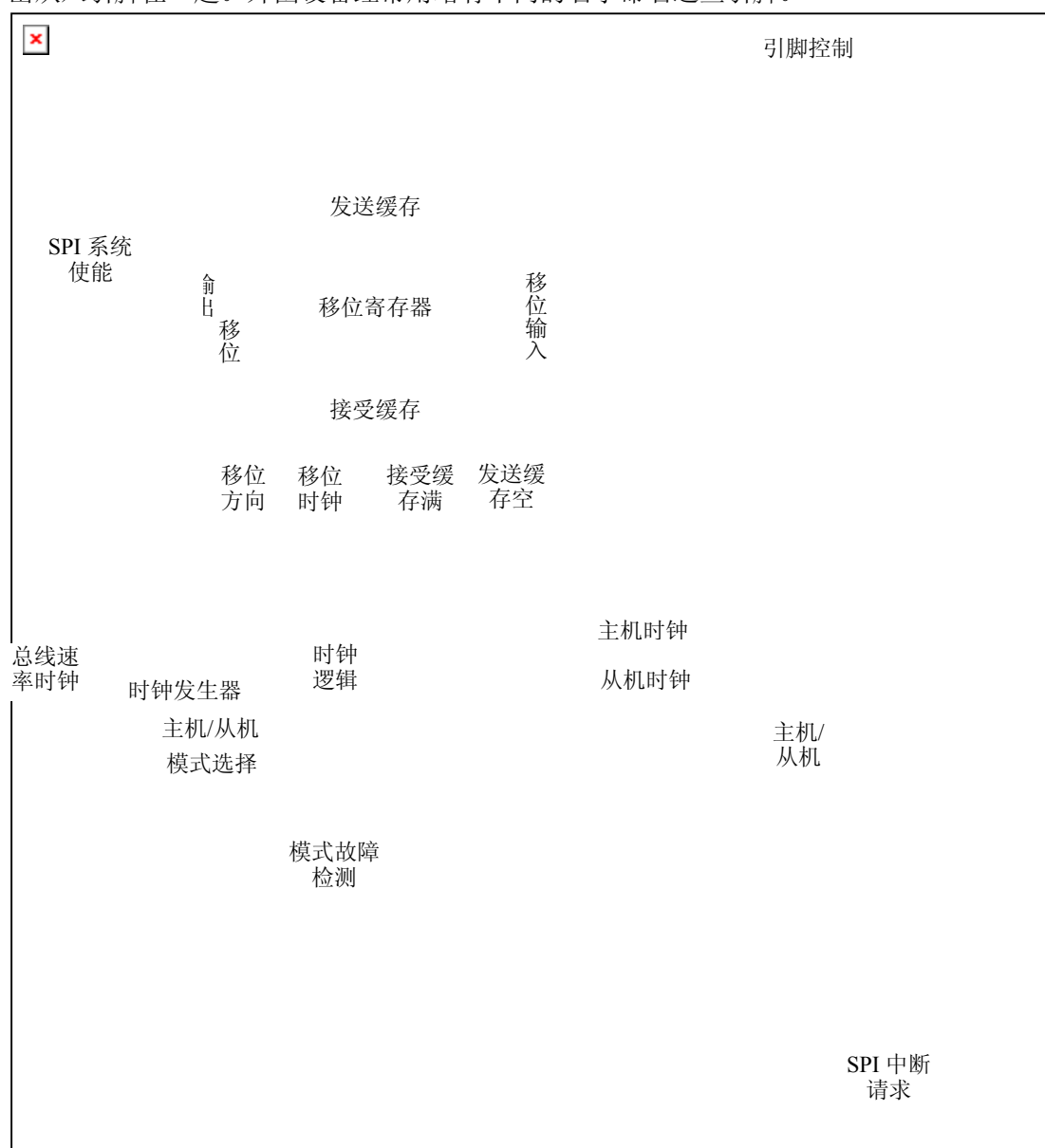


图 12-3 SPI 模块框图

12.1.3 SPI波特率发生器

如图 12-4 所示 SPI 波特率发生器时钟源是总线时钟。三个分频位（SPPR2 :SPPR21:SPPR0）选择一个分频约数 1、2、3、4、5、6、7 和 8。该三位速率位 SPPR2 :SPPR21:SPPR0 组合在 2、4、8、16、32、64 和 128，256 选择分频系数获得 SPI 内部主机模式位率时钟。

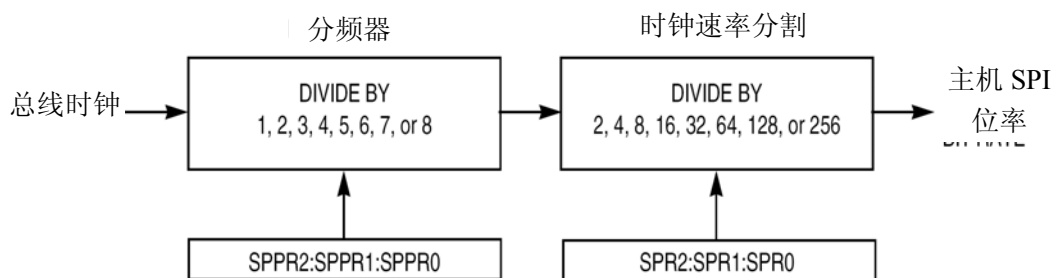


图 12-4 SPI 波特率发生器

12.2 外部信号描述

该 SPI 可复用四个端口引脚。这些引脚的功能取决于 SPI 控制位的设置。当 SPI 禁止时（SPE=0），该四个引脚作为通用 I/O 端口引脚，不被 SPI 控制。

12.2.1 SPSCCK — SPI串行时钟

当 SPI 工作于从机方式，该引脚为串行时钟输入。当 SPI 工作于主机方式，该引脚为串行时钟输出。

12.2.2 MOSI — 主入从出引脚

当 SPI 工作于主机方式时 SPI 引脚控制的零脚为 0（非双向模式），该引脚为串行数据输出。

当 SPI 工作于从机方式且 SPC0=0 时，该引脚为串行数据输入。如果 SPC0=1 选择单线双向模式，选择为主机模式，该引脚转变为双向数据 I/O（MOMI）引脚。同样双向模式输出使能位决定该引脚是作为输出还是输入。如果 SPC0=1 并选择从机模式，该引脚不被 SPI 使用并作为通用 I/O 端口引脚。

12.2.3 MISO — 主入从出引脚

当 SPI 工作于主机方式且 SPI 引脚控制零脚 SPC0 为 0（非双向模式），该引脚为串行数据输入。当 SPI 工作于从机方式且 SPC0=0 时，该引脚为串行数据输出。如果 SPC0=1 选择单线双向模式，从机模式被选，该引脚变为双向数据 I/O 引脚（SISO）同时双向模式使能位决定该引脚为输出（BIDIROE=0）还是输入（BIDIROE=1）。如果 SPC0=1 且主机模式被选中，该引脚不被 SPI 使用作为一个通用 I/O 端口引脚。

12.2.4 $\overline{ss}$ — 从机选择引脚

当 SPI 工作于从机方式时，该引脚为低电平选择输出脚。当 SPI 工作于主机方式且模式默认使能禁止（MODFEN=0），该引脚不被 SPI 使用作为一个通用 I/O 端口引脚。当 SPI 工作于主机方式且 NODFEN=1 时，从机选择输出使能位决定该引脚是否作为默认输入（SSOE=0）或作为从机选择输出（SSOE=1）。

12.3 操作模式

12.3.1 停止模式中的SPI

SPI 在所有停止模式中别禁止，无论在执行停止指令之前作了什么设置。在停止模式 1 或停止模式 2 期间，该 SPI 模块被完全拉底。若从停止模式 1 或停止模式 2 中唤醒，SPI 模块将处于复位状态。在停止模式 3 期间，SPI 模块时钟被禁止。没有寄存器被影响。如果停止模式 3 通过复位退出，SPI 将进入复位状态。如果停止模式 3 通过一个中断退出，当处于停止模式 3 时 SPI 继续保持原态。

12.4 寄存器定义

SPI 有五个 8 位寄存器来选择 SPI 选项，控制比特率，报告 SPI 状态，发送/接受数据。参阅存储器章节所有 SPI 寄存器绝对地址分配数据表中的直接页寄存器摘要。这部分通过名称来检索寄存器和控制位，一个 a Freescale-provided equate 或头文件被用来翻译这些名称为其相应绝对地址。

12.4.1 SPI控制寄存器1（SPI1C1）

该读/写寄存器包括 SPI 使能控制，中断使能和配置选项

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

图 12-5 SPI 控制寄存器 1（SPI1C1）

表 12-1 SPI1C1 位描述

位	描述
7 SPIE	SPI中断使能位（SPRF和MODF）— 这是一个中断使能位用来控制SPI接受缓存满和模式错误时件。 0 来自SPRF和MODF抑制中断（使用轮询检测） 1 当SPRF或MODF为1时，请求一个硬件中断
6 SPE	SPI系统允许位 — 禁止SPI停止进程中任何传送，清除数据缓存，初始化内部状态。SPRF被清零，SPTIEF被置位已表示SPI传送数据缓存为空。 0 SPI系统禁止 1 SPI系统允许

5 SPTIE	SPI发送中断允许位 — 该中断使能位用于SPI发送缓存空 0 来自SPTE抑制中断（使用轮询检测） 1 当SPTE为1时，硬件中断请求
4 MSTR	主机/从机模式选择位 0 SPI模块配置为一个从机SPI设备 SPI模块配置为一个主机SPI设备
3 CPOL	时钟极性位 — 该位有效放置一个变极器串联来自主机SPI或给从机SPI的时钟信号。详情参阅12.5.1章节“SPI时钟格式” 0 SPSCK空闲时为低电平 1 SPSCK空闲时为高电平
2 CPHA	时钟相位位 — 该位为不同类型的同步串行外设选择一个时钟形式。详情参阅12.5.1章节“SPI时钟格式” 0 SPSCK第一次沿跳变发生在8个数据发送周期的第一个周期中期 1 SPSCK第一次沿跳变发生在8个数据发送周期的第一个周期开始
1 SSOE	从机选择输出使能位 — 该位用于连接SPCR2模式故障使能位（MODFEN）和主机/从机控制位（MSTR）决定表12-2中 $\overline{SS}$ 引脚功能
0 LSBFE	LSB优先（移位寄存器方向） 0 开始时即以最多显著位进行SPI串行数据传送 1 开始时即以最少显著位进行SPI串行数据传送

表 12-2 $\overline{SS}$ 引脚功能

MODFEN	SSOE	主机模式	从机模式
0	0	通用I/O（非SPI）	从机选择输入
0	1	通用I/O（非SPI）	从机选择输入
1	0	$\overline{SS}$ 引脚输入用于模式故障	从机选择输入
1	1	$\overline{SS}$ 引脚自动输出	从机选择输入

注：

确保 SPI 被禁止（SPE=0）同时改变 CPHA 位。这些改变应表现为独立操作或突发行发生。

12.4.2 SPI控制寄存器2（SPI1C2）

该读/写寄存器用于控制 SPI 系统选项特征。位 7，6，5，2 是无效的，一直读为 0

	7	6	5	4	3	2	1	0
读	0	0	0			0		
写				MODFEN	BIDIROE		SPISWAI	SPC0
复位	0	0	0	0	0	0	0	0

 =未定义的

图 12-6 SPI 控制寄存器 2（SPI1C2）

表 12-3 SPI1C2 寄存器位描述

位	描述
---	----

4 MODFEN	主机模式-故障功能使能位—当SPI工作于从机模式，该位无意义，（ SS 引脚是从机选择输出引脚）在主机模式下该位决定 SS 引脚如何使用（详情参阅表12-2） 0 模式故障功能禁止，主机 SS 引脚作为通用I/O引脚不受SPI控制 1模式故障功能允许，主机 SS 引脚作为模式故障输入或从机选择输出
3 BIDIROE	双向模式输出允许—当双向模式通过SPI引脚控制SPC0=1被允许，BIDIROE决定该SPI数据输出驱动是否允许单线双向SPI I/O引脚。根据SPI工作于主机模式还是从机模式，它使用MOSI(MOMI)或MISO(SISO)引脚中的一个，作为SPI数据I/O引脚。 当SPC0=0时，BIDIROE无意义。 0输出驱动禁止SPI数据I/O引脚作为输入引脚 1 SPI I/O引脚允许作为输出引脚
1 SPISWAI	在等待模式下SPI停止 0 SPI时钟继续工作于等待模式下 1 当MCU进入等待模式时，SPI时钟停止
0 SPC0	SPI控制引脚0—SPC0引脚选择单线双向模式。如果MSTR=0（从机模式），该SPI 使用MISO引脚用于双向SPI数据发送。如果MSTR=1，BIDIROE用来允许或禁止用于单线双向SPI I/O引脚的输出驱动。 0 SPI使用独立引脚用于数据输入，输出 1 SPI配置为单线双向模式

12.4.3 SPI比特率寄存器（SPI1BR）

该寄存器为工作于主机方式 SPI 设置分频器和位率因子，可在任何时候被读或写。

	7	6	5	4	3	2	1	0
读	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
写								
复位	0	0	0	0	0	0	0	0

 =未定义的

图 12-7 SPI 比特率寄存器（SPI1BR）

表 12-4 SPI1BR 寄存器位描述

位	描述
6:4 SPPR[2:0]	SPI波特率分频因子—这三位为SPI波特率分频器选择表12-5中的八个因子中的一个。该分频器的输入是总线速率时钟（BUSCLK）。该分频器的输出驱动SPI波特率分割器的输入（见图12-4）
2:0 SPR[2:0]	SPI波特率因子—这三位为SPI波特率分割器选择表12-6中的八个因子中的一个。该分割器的输入来自SPI波特率分频器（见图12-4）。该分割器的输出为主机模式下的SPI位率时钟。

表 12-5 SPI 波特率分频器因子

SPPR2:SPPR1:SPPR0	分频器因子
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

表 12-6 SPI 波特率因子

SPPR2:SPPR1:SPPR0	位率因子
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

12.4.4 SPI状态寄存器（SPI1S）

该寄存器有三个只读状态位。位 6, 3, 2, 0 不被执行，通常读为 0。对它们写无意义或影响。

	7	6	5	4	3	2	1	0
读	SPRF	0	SPTEF	MODF	0	0	0	0
写								
复位	0	0	1	0	0	0	0	0

 =未定义的

图 12-8 SPI 状态寄存器（SPI1S）

表 12-7 SPI1S 寄存器位描述

位	描述
7 SPRF	SPI读缓冲区满标志—SPRF在SPI完成数据发送时被置位表明收到的数据可能来自SPI数据寄存器（SPI1D）。通过读SPRF可以清零它，然后读SPI数据寄存器。 0 在接受数据缓存中无可用数据 1 在接受数据缓存中数据可用
5 SPTEF	SPI发送缓冲区空标志—当传送数据缓冲区里有空间时该标志位被置位。它可通过读SPI1S来清零，然后写一个数据值给SPI1D中的传送缓冲区。在SPTEF=1，写数据给SPI1D之前必须读SPI1S否则写SPI1D将被忽略。如果SPI1C1中的SPTIE位也被置位SPTEF产生一个SPTEF CPU中断请求。当一个字节的数据从传送缓冲区里发送到发送移位寄存器时，SPTEF被自动置位。对于闲置的SPI（无数据在发送缓冲区或移位寄存器中，无发送动作），被写到SPI1D的数据被立即发送到移位寄存器，所以SPTEF在两个总线周期内被置位允许第二个8位值排队进入发送缓冲区。完成移位寄存器中的值的发送后，来自发送缓冲区的排队值将自动移入到移位寄存器，SPTEF将被置位以表明发送缓冲区里还有空间存放新数据。如果没有新数据在缓冲区里，SPTEF保持置位，无数据从缓冲区移到移位寄存器。 0 SPI发送缓冲区非空 1 SPI发送缓冲区空
4 MODF	主机模式故障标志—如果SPI工作于主机方式且从机选择输入脚为低，MODF被置位，表明其它一些SPI设备也工作于方式。仅当MSTR=1，MODFEN=1，SSOE=0时该SS引脚用于模式故障输入；否则MODF将从不被置位。通过读MODF可以清零它，然后写SPI控制寄存器1（SPI1C1） 0 无模式故障错误 1 模式故障错误检测

12.4.5 SPI数据寄存器（SPI1D）

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

图 12-9 SPI 数据寄存器（SPI1D）

读该寄存器返回来自接受数据缓冲区里的数据。写该寄存器写数据到发送数据缓冲区。当 SPI 工作于主机方式时，写数据到发送数据缓冲区表明一个 SPI 发送开始。

数据不应该被写入发送数据缓冲区除非该 SPI 发送缓冲区空标志（SPTEF）置位，表明发送缓冲区里有空间来排队一个新的发送字节。

在 SPRF 被置位后和在另一个发送结束之前可在任意时间从 SPI1D 读取数据。在一个新的发送尾期导致接受溢出和新发送的数据丢失之前读出接受数据缓冲区中过的数据失败。

12.5 功能性描述

通过检测 SPI 发送缓冲区空标志（SPTEF=1）开始一个 SPI 传送，然后写一个字节数据给工作于主机方式 SPI 数据寄存器。当 SPI 移位寄存器可用时，该字节数据从发送数据缓冲区移到移位寄存器，SPTEF 被置位以表明缓冲区里还有空间可利用，SPI 串行发送开始。在 SPI 发送数据期间，数据在 MISO 脚上的一个 SPSCCK 时钟边沿被读取和转移，改变 MOSI 引脚上的位值在稍后的一个半 SPSCCK 时钟周期。在八个 SPSCCK 时钟周期后主机移位寄存器中的数据通过 MOSI 引脚发送给从机，同时八位的数据通过 MISO 引脚从从机转移到主机移位寄存器。在这次传送过程最后，接受到的数据从移位寄存器移到接收数据缓冲区里且 SPRF 被置位以表明可通过读 SPI1D 来获取数据。如果另一个字节的数据在发送缓冲区里等待，它将被转移到移位寄存器，SPTEF 被置位且一个新的传送开始。

通常 SPI 数据首先被发送最高有效位（MSB）。如果最低有效位（LSB）优先允许，则优先发送最小位。

当 SPI 工作于从机方式，其 $\overline{ss}$ 引脚被拉低并在整个传送过程中保持低电平。如果 CPHA=0，在连续传送时 $\overline{ss}$ 引脚被置位逻辑 1。如果 CPHA=1，在连续传送时 $\overline{ss}$ 引脚保持低电平。详情参阅 12.5.1 “SPI 时钟格式”。

因为传送者和接受者都是双缓冲，所以第二个字节可以被立即移出排队进入发送数据缓冲区，先前接受到的数据放到接收数据缓冲区里同时一个新数据正在移入。SPTEF 标志表明发送缓冲区是否有空间给新数据使用。接收到的数据必须在下个传送结束之前从接收缓冲区里读出，否则产生一个接受溢出错误。

在接收溢出情况下，新数据将丢失因为接收缓冲区仍然保持先前的数据还未准备接收新数据。目前尚无现象来显示这样一个溢出情况，因此应用系统的设计者必须确保以前的数据在一个新的传送过程开始之前已经被读出接收缓冲区。

12.5.1 SPI时钟格式

为适应来自不同厂商不同种类的同步串行外设，SPI 系统含有一个时钟极性位（CPOL）和一个时钟相位控制位（CPHA）来选择四种数据传输形式中的一种。CPOL 选择性的插入以个变极器串联时钟。CPHA 在时钟和数据之间选择两个不同时钟相位关系。

图 12-10 显示 CPHA=1 时的时钟形式。在该图的顶部，八个位时间被显示位 1 在第一个 SPSCCK 边沿开始第十六个 SPSCCK 边沿后的一个半 SPSCCK 周期位 8 结束。最高有效

位优先和最低有效位优先排列显示 SPI 数据位的顺序，取决于 LSBFE 中的设置。SPSCK 极性的

两种不同变化被显示出来，但这些波形中只有一种应用于具体的传送，取决于 CPOL 的值。SAMPLE IN 波形应用于一个从机的 MOSI 引脚输入或一个主机的 MISO 引脚输出。MOSI 波形应用于来自主机的 MOSI 输出引脚而 MISO 波形应用于来自从机的 MISO 输出引脚。该 $\overline{ss}$ 输出波形应用于来自主机的从机选择输出（假设 MODFEN=SSOE=1）。该主机的 $\overline{ss}$ 引脚输出在传送的第一个位时间开始为低电平，在八个位时间后变为高电平并保持一个半 SPSCK 时钟周期。 $\overline{ss}$ 输入波形应用于一个从机的从机选择输入。

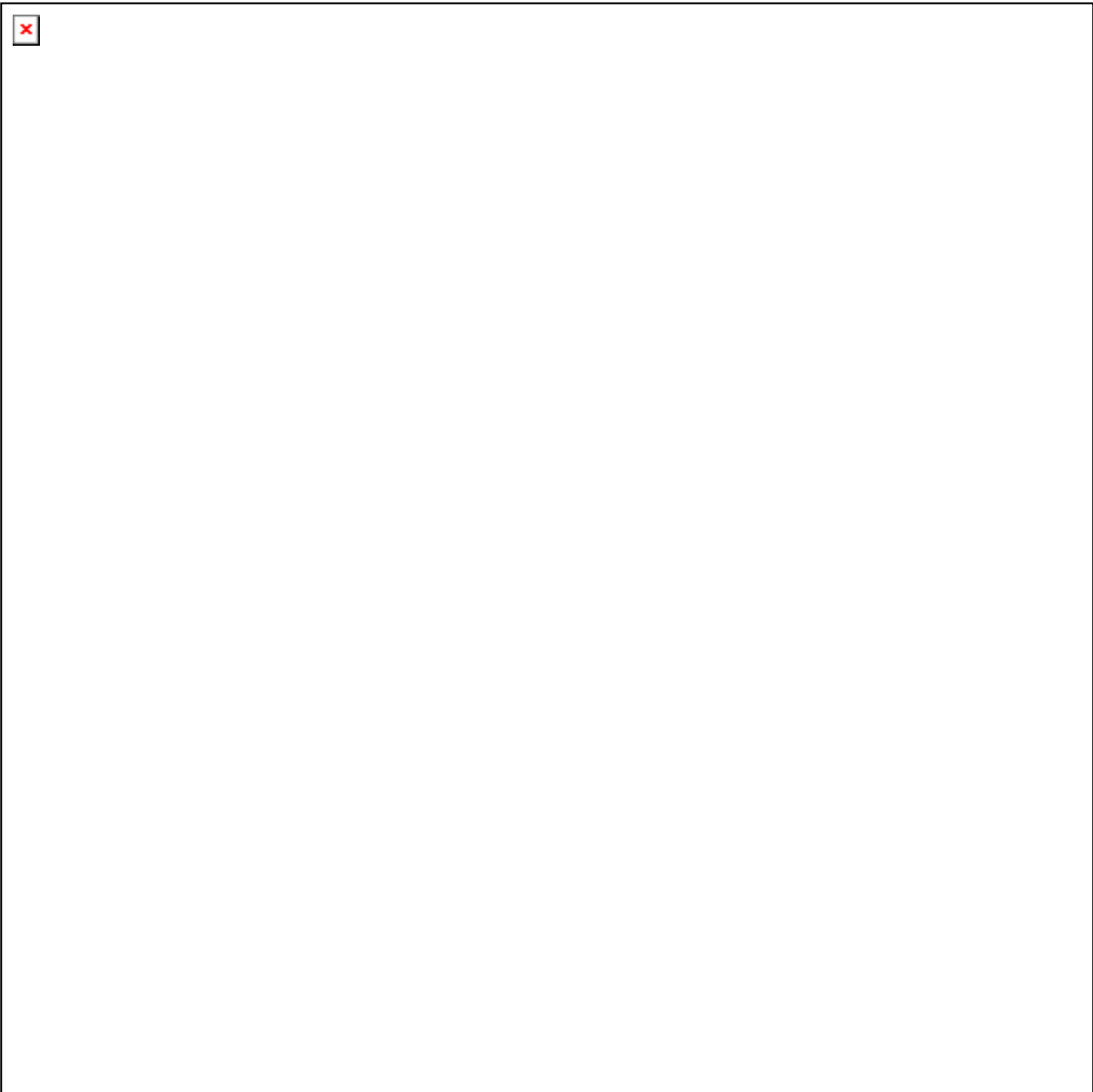


图 12-10 SPI 时钟格式（CPHA=1）

当 CPHA=1 时， $\overline{ss}$ 引脚走低时从机开始驱动其 MISO 引脚输出，当直到第一个 SPSCK 边沿数据才被识别。在第一个 SPSCK 边沿从移位寄存器移出数据的第一位到主机的 MOSI 输出引脚和从机的 MISO 输出引脚。下个 SPSCK 边沿主机和从机分别采样 MISO 和 MOSI 输入引脚上的数据。在第三个 SPSCK 边沿，该 SPI 移位寄存器移出一位，移入该位的值恰是移位寄存器采样和移位得到的第二个数据位值分别输给主机和从机的 MOSI 和 MISO 输出引脚。当 CPHA=1 时，该从机的 $\overline{ss}$ 引脚输入在传送期间不需要变为高电平。

图 12-11 显示 CPHA=0 时的时钟格式。在该图的顶部，八个位时间被显示当从机被选

中（ $\overline{ss}$ 引脚输入拉底）时位 1 开始，在最后的 SPSCCK 边沿位 8 结束。MSB 优先和 LSB 优先显示 SPI 数据位的顺序，这些取决于 LSBFE 中的设置。SPSCCK 极性的两种变化都被显示，但只只有其中的一种波形应用于具体的传送过程，该波形取决于 CPOL 的值。SAMPLE IN 波形应用于一个从机的 MOSI 引脚输入或一个主机的 MISO 输入。MOSI 波形应用于来自主机的 MOSI 引脚输出，MISO 波形应用于来自从机的 MISO 引脚输出。 $\overline{ss}$ OUT 波形应用于来自主机（假设 MODFEN=SSOE=1）的从机选择输出。该主机的 $\overline{ss}$ 引脚输出在传送的第一个位时间开始时拉低，在八个传送位时间之后的一个半 SPSCCK 周期被拉高。 $\overline{ss}$ IN 引脚波形应用于一个从机的从机选择输入。

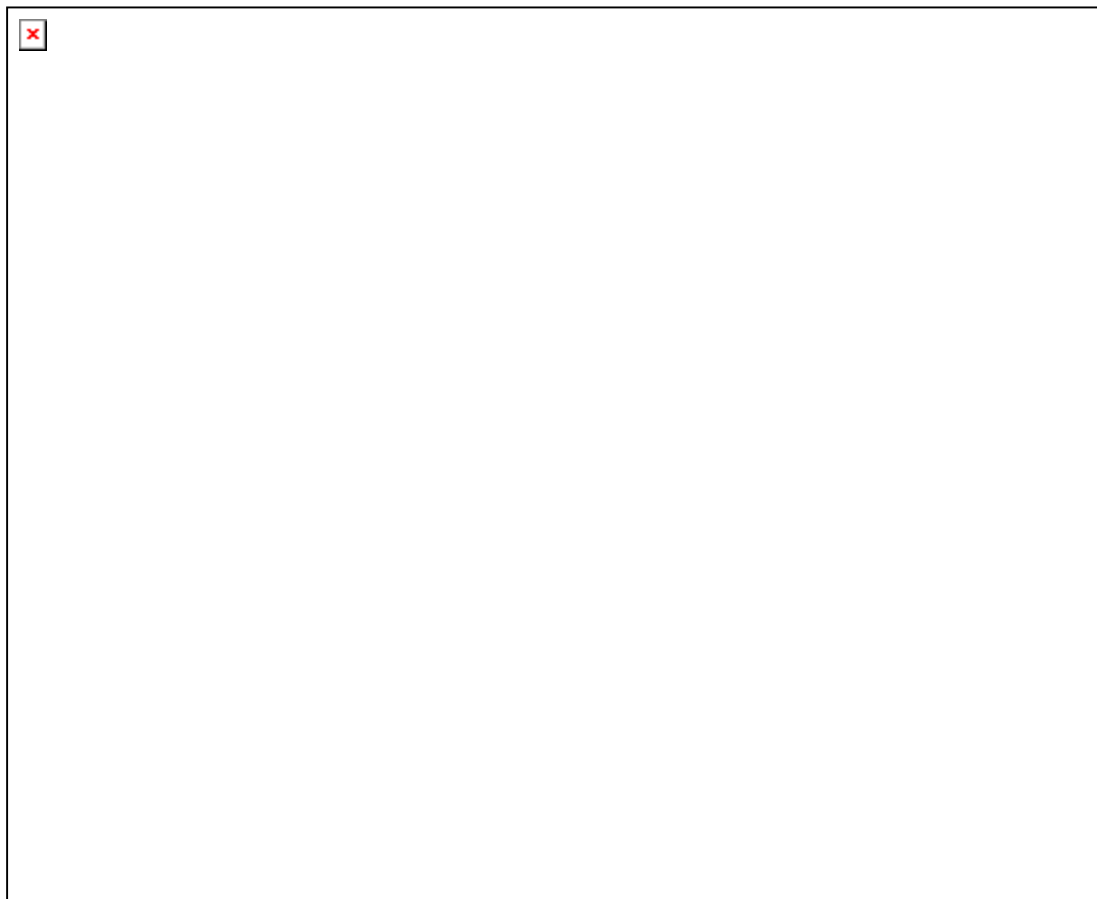


图 12-11 SPI 时钟格式（CPHA=0）

当 CPHA=0， $\overline{ss}$ 引脚为低时，从机开始驱动它的 MISO 引脚输出第一个数据位值（取决于 LSBFE 的 MSB 或 LSB）。第一个 SPSCCK 边沿时主机和从机各自采样 MISO 和 MOSI 引脚上的数据。在第二个 SPSCCK 边沿，该 SPI 移位寄存器移出一位，移入该位的值恰是移位寄存器采样和移位得到的第二个数据位值分别输给主机和从机的 MOSI 和 MISO 引脚。当 CPHA=0 时，该从机的 $\overline{ss}$ 引脚在传送时必须为高电平。

12.5.2 SPI 中断

与 SPI 系统相关的有三个标志位，两个中断屏蔽位和一个中断向量。SPI 中断屏蔽使能位（SPIE）允许来自 SPI 接受缓存满标志（SPRF）中断和模式故障标志（MODF）中断。SPI 发送中断使能屏蔽位（SPTIE）允许来自 SPI 发送换缓存空标志（SPTEF）中断。的那个这些标志位中的一个被置位和相应的中断屏蔽位被置位时，一个硬件中断被发送给 CPU。如果中断屏蔽位被清零，软件可以轮询相关的标志位来代替使用中断。该 SPI 中断服务例程

（ISR）应该检查其标志位以获知是什么事件引发的中断。在中断服务例程返回之前应该清零这些标志位。

12.5.3 模式故障检测

一个模式故障发生，模式故障标志位被置位在一个主机 SPI 设备检测到 $\overline{ss}$ 引脚上的错误时（假设该 $\overline{ss}$ 引脚被设置成模式故障输入信号）。当 MSTR=1 时该 $\overline{ss}$ 引脚被配置为模式故障输入信号，模式故障使能位被置位（MODFEN=1），从机选择输出使能位被清零（SSOE=0）。

模式故障检测特征可被用于多 SPI 设备系统。当主机的 $\overline{ss}$ 引脚为低时错误被检测到，表明其它一些先前为从机的 SPI 设备正试图掌握这个主机。这可能表明一种有害的输出驱动冲突，因此当这样的一个错误被检测到时，模式故障逻辑被设计用来停止所有的 SPI 输出驱动器。

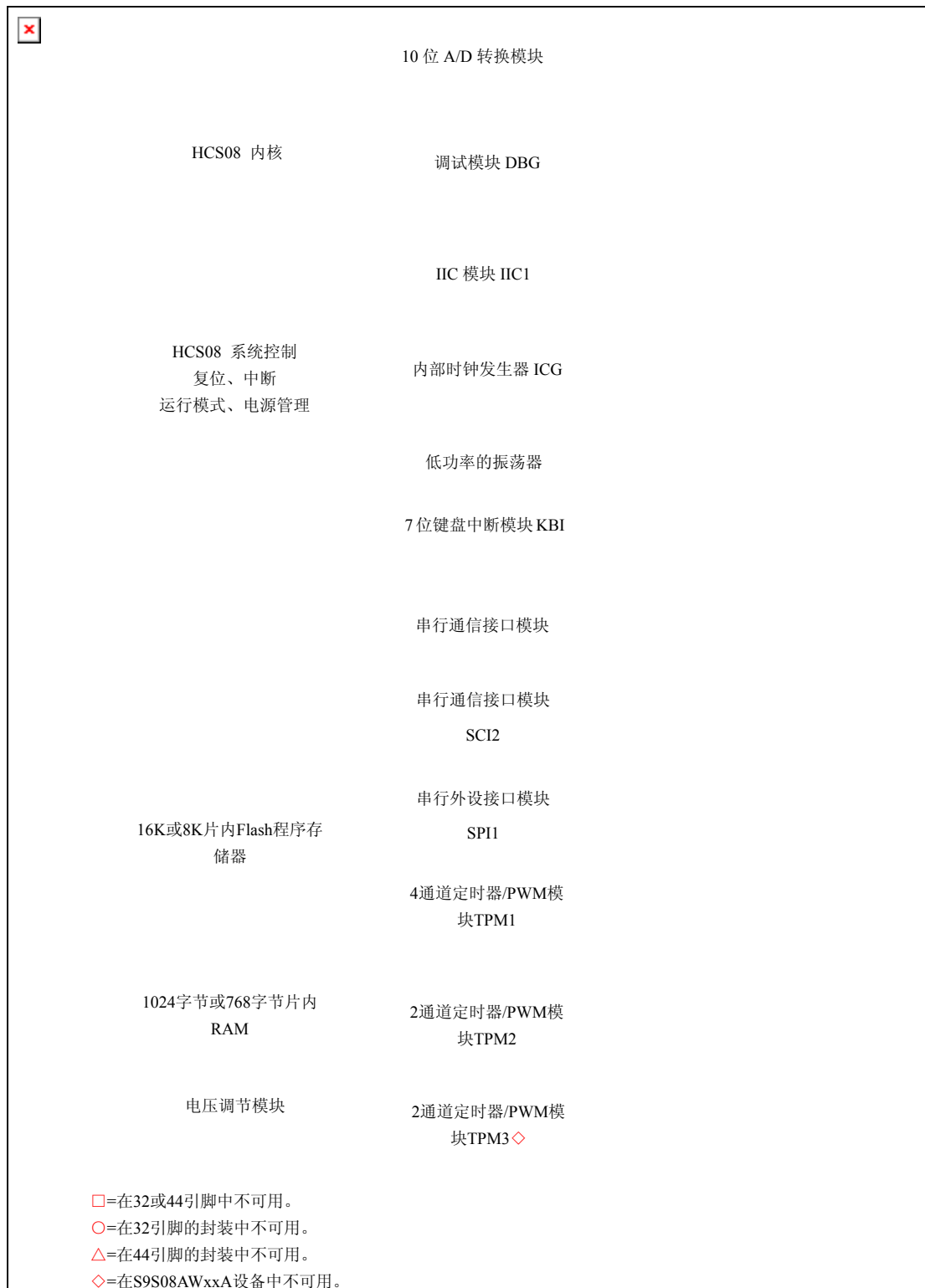
当一个模式故障被检测到时，MODF 被置位且 MSTR 被清零以改变 SPI 配置返回到从机模式。SPSCK、MOSI、MISO（如果不是双向模式）引脚上的输出驱动被禁止。

通过读 MODF 可清零 MODF，饭后写 SPI 控制寄存器 1（SPI1C1）。在把 SPI 改回到主机模式之前用户软件应核实误差的情况已得到纠正。

第十三章 IIC总线（S08IICV2）

13.1 简介

MC9S08AC16 系列微控制器包含一个 IIC 模块用于内部电路的通信连接。该模块相关的两个引脚 SCL 和 SDA 复用端口 C 引脚 0 和 1。



注:

- 1.如果是输入端口，端口引脚用上拉设备进行软件配置。
- 2.若 IRQ 允许（IRQPE=1）引脚包含软件配置的上拉/下拉设备。如果上升沿检测有效（IRQEDG=1）下拉允许。
- 3.IRQ 不需要通过钳二极管连接到 VDD。供电电压不超过 VDD。
- 4.引脚包含整个上拉设备。

5.PTD3、PTD2、PTG4 包含上拉和下拉设备。当 KBI 允许时下拉有效（KBIPEn=1），上升沿检测有效（KBEDGn=1）

图 13-1 MC9S08AC16 框图

13.1.1 特征

IIC 以下显著特征：

- 兼容 IIC 总线标准
- 支持多主控
- 软件可编程 64 种不同串行时钟频率
- 软件可选择应答位
- 字节传送中断
- 仲裁自动模式转换从主机到从机丢失的中断
- 调用地址识别中断
- 开始和停止信号发生/检测
- 重复产生开始信号
- 识别位发生/检测
- 总线忙检测
- 一般调用识别
- 10 位地址扩展

13.1.2 操作模式

下面简述不同 MCU 模式下的 IIC：

- 运行模式 — 这是基本的操作模式。为了节约能量，在此模式下，禁用该模块。
- 等待模式 — 该模块继续运作，而 MCU 是在等待模式，并能提供一个唤醒中断。
- 停止模式 — IIC 在停止模式 3 下是停止的以降低功耗。停止指令不影响 IIC 寄存器状态。停止模式 2 和停止模式 1 复位该寄存器内容。

13.1.3 框图

图 13-2 是 IIC 框图

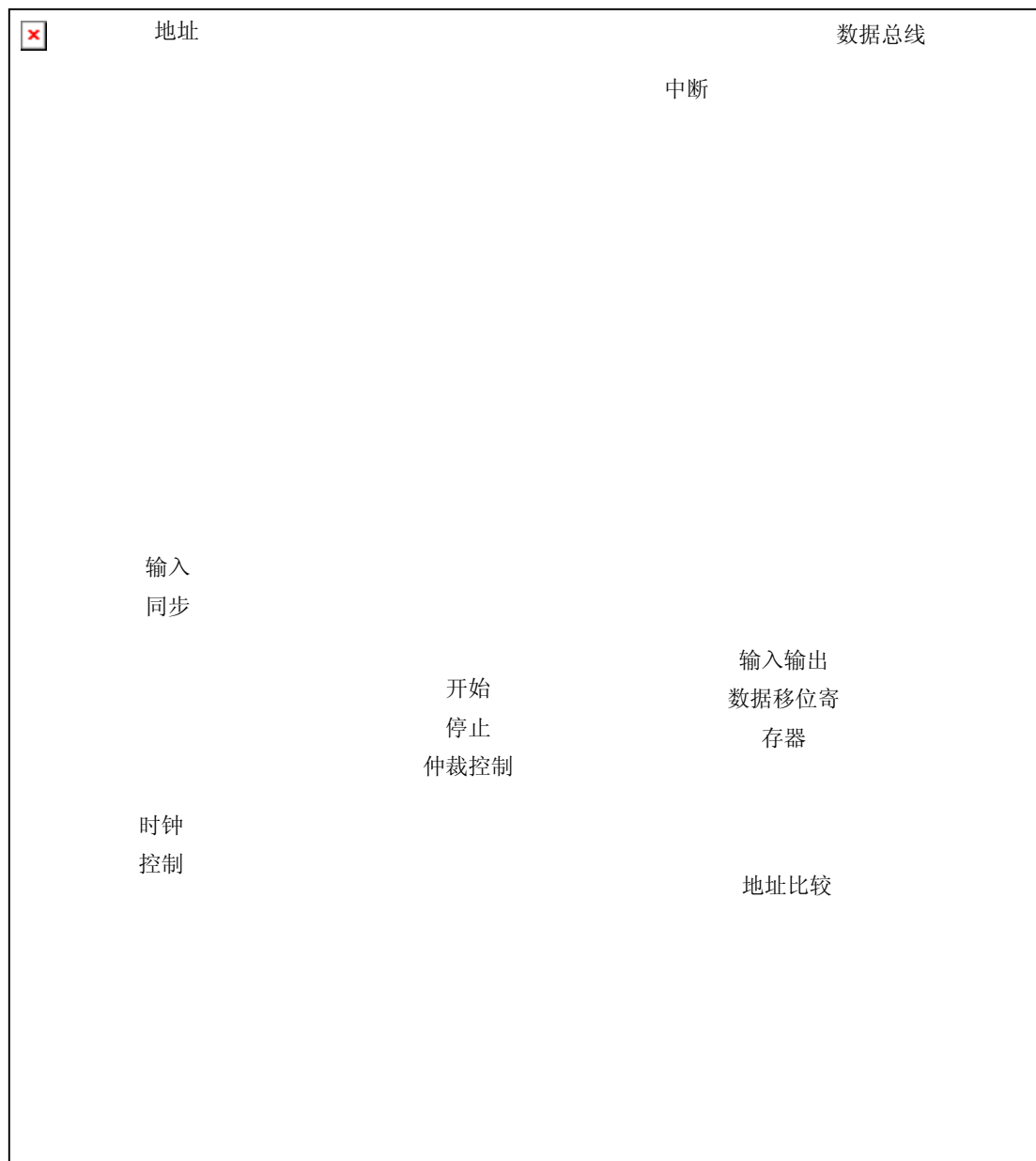


图 13-2 IIC 功能框图

13.2 外部信号描述

这部分描述每个用户可访问的引脚信号。

13.3.1 SCL—串行时钟线

该双向 SCL 是 IIC 系统的串行时钟线

13.2.2 SDA—串行数据线

该双向 SDA 是 IIC 系统的串行数据线

13.3 寄存器定义

这部分按地址顺序描述 IIC 寄存器。参阅存储器章节的所有 IIC 寄存器绝对地址分配文件关于直接页寄存器摘要。这部分通过寄存器和控制位的名字来使用。飞思卡尔提供头文件用来把这些名字转换成相应的绝对地址。

13.3.1 IIC地址寄存器（IIC1A）

	7	6	5	4	3	2	1	0
读	AD7	AD6	AD5	AD4	AD3	AD2	AD1	0
写								
复位	0	0	0	0	0	0	0	0

 =未定义的

图 13-3 IIC 地址寄存器（IIC1A）

表 13-1 IIC1A 位描述

位	描述
7-1 AD[7:1]	从机地址。AD[7:1]包含被IIC模块使用的从机地址。该位使用7位地址方案和10位地址方案中的低七位。

13.3.2 IIC频率分频寄存器（IIC1F）

	7	6	5	4	3	2	1	0
读	MULT		ICR					
写								
复位	0	0	0	0	0	0	0	0

图 13-4 IIC 频率分频寄存器

表 13-2 IIC1F 位描述

位	描述
7-6 MULT	IIC增频因子。MULT位定义了增频因子，mul.mul和SCL配合使用产生IIC波特率。 Mul定义如下： 00 mul=01 01 mul=02 10 mul=04 11 不使用
5-0 ICR	IIC 时钟速率。ICR被用来分频总线时钟化作位率选择。这些位和MULT决定了IIC波特率，SDA保持时间，SCL开始保持时间和SCL停止保持时间。 表13-4提供了SCL因子和保持值。 SCL分频因子乘以MULT增频因子得到IIC波特率 IC波特率=CPU总线频率/mul*SCL分频因子 SDA保持时间是从SCL线上时钟的下降沿开始到SDA线上数据稳定这段时间。 SDA保持时间=CPU总线周期*mul*SDA保持值 SCL开始保持时间是从SDA下降沿开始，从高（开始条件）到SCL下降沿(IIC时钟) SCL开始保持时间=CPU总线周期*mul*SCL开始保持值 SCL停止保持时间是从SCL上升沿开始到SDA上升沿，此时SCL为高（停止条件）。 SCL停止保持时间=CPU总线周期*mul*SCL停止保持值

例如，如果总线速度是 8MHz，下表显示可能的保持时间值用不同的 ICR 和 MULT 选择获得 100kbps 的 IIC 波特率。

表 13-3 8MHz 总线速度保持时间值

MULT	ICR	保持时间（微秒）		
		SDA	SCL开始	SCL停止
0x2	0x00	3.500	3.000	5.500
0x1	0x07	2.500	4.000	5.250
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.126	4.25	5.125
0x0	0x18	1.125	4.750	5.125

表 13-4 IIC 分频因子和保持值

ICR (hex)	SCL 分频 因子	SDA 保持 值	SCL 保持 值(开 始)	SCL保 持值 (停 止)	ICR (hex)	SCL 分频 因子	SDA 保持 值	SCL 保持 值(开 始)	SCL 保持 值(停 止)
00	20	7	6	11	20	160	17	78	81
01	22	7	7	12	21	192	17	94	97
02	24	8	8	13	22	224	33	110	113
03	26	8	9	14	23	256	33	126	129
04	28	9	10	15	24	288	49	142	145
05	30	9	11	16	25	320	49	158	161
06	34	10	13	18	26	384	65	190	193
07	40	10	16	21	27	480	65	238	241
08	28	7	10	15	28	320	33	158	161
09	32	7	12	17	19	384	33	190	193
0A	36	9	14	19	2A	448	65	222	225
0B	40	9	16	21	2B	512	65	254	257
0C	44	11	18	23	2C	576	97	286	289
0D	48	11	20	25	2D	640	97	318	321
0E	56	13	24	29	2E	768	129	382	385
0F	68	13	30	35	2F	960	129	478	481
10	48	9	18	25	30	640	65	318	321
11	56	9	22	29	31	768	65	382	385
12	64	13	26	33	32	896	129	510	513
13	72	13	30	37	33	1024	129	510	513
14	80	17	34	41	34	1152	193	638	641
15	88	17	38	45	35	1280	193	574	577
16	104	21	46	53	36	1536	257	766	769
17	128	21	58	65	37	1920	257	958	961
18	80	9	38	41	38	1280	129	638	641
19	96	9	46	49	39	1536	129	766	769
1A	112	17	54	57	3A	1792	257	894	897
1B	128	17	62	65	3B	2048	257	1022	1025
1C	144	25	70	73	3C	2304	385	1150	1153
1D	160	25	78	81	3D	2560	385	1278	1281
1E	192	33	94	97	3E	3072	513	1534	1537
1F	240	33	118	121	3F	3840	513	1918	1921

13.3.3 IIC控制寄存器（IIC1C1）

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

读 写	IICEN	IICIE	MST	TX	TXAK	0	0	0
						RSTA		
复位	0	0	0	0	0	0	0	0

 =未定义的

图 13-5 IIC 控制寄存器（IIC1C1）

表 13-5 IIC1C1 位描述

位	描述
7 IICEN	IIC使能位。该IICEN位决定IIC模块是否允许。 0 IIC不允许 1 IIC允许
6 IICIE	IIC中断请求允许位。IICIE位决定IIC中断请求是否被允许。 0 IIC中断请求不允许 1 IIC中断请求允许
5 MST	主模式选作位。当该位从0变到1，IIC总线上产生一个开始信号，MCU被确定为主模式。 当该位从1到0，MCU将在IIC总线上产生停止信号，操作模式也敬爱那个从注模式变为从模式。 0 从机模式 1 主机模式
4 TX	发送模式选择位。该位用来选择主机和从机的数据传送方向。在主机模式下，该位可根据数据传送需求来确定，虽然在整个地址周期，该位始终为1.当MCU为从机时，该位将根据状态寄存器的SRW位来确定 0 接受模式 1 发送模式
3 TXAK	发送应答使能位。在主机和从机接受缓存的数据识别期间该位指定驱动到SDA上的值 0 在接受完一个字节后发送应答信号 1 不发送应答信号
2 RSTA	重新开始（repeat START）。若MCU在主机模式下，写一个1到该位将产生一个重新开始信号。该位读出时总为0。在错误的时间尝试重新开始将丢失仲裁。

13.3.4 IIC状态寄存器（IIC1S）

读 写	7	6	5	4	3	2	1	0
	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
复位	1	0	0	0	0	0	0	0

 =未定义的

图 13-6 IIC 状态寄存器（IIC1S）

表 13-6 IIC 位描述

位	描述
7 TCF	发送完成标志位。一个字节传送完毕后，该位将被置位。该位只在进行IIC模块传送时有效。在接收模式下，读IIC数据寄存器IIC1D或在发送模式下写IIC1D清除该位。 0 发送中 1 发送已完成
6 IAAS	地址被选择标志位。当MCU作为从机，并且它的地址与IIC总线上由主机发送的寻址地址吻合时，该位被置位。写IIC1C寄存器清零该位。

	0 MCU未被寻址 1 MCU作为从机被主机寻址
5 BUSY	总线忙标志。该位表明总线的状态无论是在主机模式还是从机模式。当检测到开始信号，该位被置位。当检测到停止信号时，该位被清零 0 总线空闲 1 总线忙
4 ARBL	仲裁丢失标志位。当仲裁处理丢失，该位被硬件置位。必须通过软件的方法向该位写1来清除该位。 0 正常总线操作 1 仲裁丢失
2 SRW	从机读写标志位。当MCU作为从机被寻址时，该位被设为主机发送的寻址地址的读/写命令的值。 0 从机为接收模式，主机向从机写数据 1 从机为发送模式，主机从从机读数据
1 IICIF	IIC中断标志位。当一个中断发生该位被置位。必须通过软件方法来向该位写1清除该位。下列事件之一可以置位IICIF：一个字节传送完毕，从地址被选中，仲裁丢失。 0 无中断未决 1 有中断未决
0 RXAK	接收应答标志。当该位为0时表示在总线上传送完一个字节后接收到了应答信号。如果为1则表示没有接收到应答信号 0 接收应答信号 1 无接收应答信号

13.3.5 IIC数据输入输出寄存器（IIC1D）

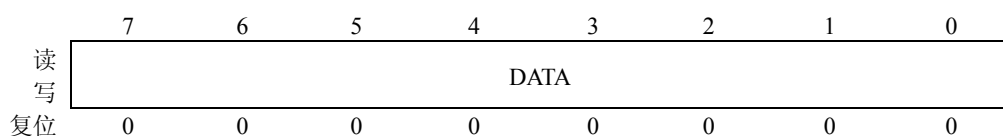


图 13-7 IIC 数据输入输出寄存器（IIC1D）

表 13-7 IIC1D 位描述

位	描述
7-0 DATA	数据—在主机发送模式下，将数据写入IIC1D就初始化了数据发送，先发送数据字节的最高位。在主机接收模式下，读该寄存器开始接受下个字节数据

注：当 MCU 要从主接收模式变为主发送模式时，应在从 IIC1D 寄存器读取数据之前，进行模式转换。

在从机模式下，在一个地址匹配发生后相同的功能是可用的。

在主机和从机模式下传送一开始时，IIC1C 中的 TX 位必须正确反映传送目标方向。例如，如果 IIC 被配置为主机发送但一个主机接收被请求时，读 IIC1D 并不能开始接收。

当 IIC 被配置为主机或从机接收模式时，读 IIC1D 返回接收到最后一个字节数据。IIC1D 并不影响在 IIC 总线上传送的每个字节，或通过读出已写入到 IIC1D 中的一个字节数据来校验。

在主机发送模式下，在 MST 被用于地址传送后数据的第一个字节被写到 IIC1D，应该包括与必需的 R/W 位（位 0）相关的寻址地址（位 7-位 1）

13.3.6 IIC控制寄存器2（IIC1C2）

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

读	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8
写								
复位	0	0	0	0	0	0	0	0
	=未定义的							

图 13-8 IIC 控制寄存器 2（IIC1C2）

表 13-8 IIC1C2 位描述

位	描述
7 GCAEN	一般寻址使能位。GCAEN位允许或禁止一般寻址。 0 一般寻址禁止 1一般寻址允许
6 ADEXT	地址扩展。ADEXT位控制用于从机地址的位数 0 7位地址方案 1 10位地址方案
2-0 AD[10:8]	从机地址。AD[10:8]包含10位地址方案从机地址的高三位。仅当ADEXT被置位该位才有效。

13.4 功能性描述

这部分将提供 IIC 模块一个完整的功能描述。

13.4.1 IIC协议

IIC 总线系统使用一个串行数据线（SDA）和一个串行时钟线(SCL)用于数据传送。所有连接到它的设备必须具有公开通道或公开手机输入。带有外部上拉电阻的两条线可执行逻辑与操作。电阻值由系统决定。

通常情况下，一个标准的通信包含以下四个部分：

- 开始信号
- 从机地址传输
- 数据传输
- 停止信号

停止信号不应被 CPU 停止指令所干扰。IIC 总线系统通信将在下面进行简要描述，并附上图 13-9。



图 13-9 IIC 总线发送信号

13.4.1.1 开始信号

当总线空闲时，无主机设备使用总线（SCL 和 SDA 逻辑高），一个主机可以通过发送一个开始信号开始一个通信。如图 13-9 所示，开始信号是当 SCL 为高时 SDA 从高变为低时产生的信号。该信号表明一个新的数据传送的开始，清除所有从机空闲状态。

13.4.1.2 从机地址传送

在开始信号之后被传输数据的第一个字节是从机地址。这是跟随在 R/W 位之后的七位寻址地址。该 R/W 位告诉从机数据传送的方向。

1=读发送，从机发送数据到主机。

0=写发送，主机发送数据到从机。

仅当主机地址和从机地址相匹配时才返回一个应答信号。这通过在第 9 个时钟拉低 SDA 来完成（见图 13-9）。

系统中没有两个从机地址是相同的。如果 IIC 工作于主机方式，它绝不会发送一个与它自身从机地址相同的地址。IIC 不能同时作为主机和从机方式使用。不论怎样，如果在一个地址周期内仲裁丢失，IIC 将恢复到从机模式并正常操作即使被另外一个主机寻址。

13.4.1.3 数据发送

在成功地完成从机寻址之前，又寻址主机发送的数据以字节为单位发送，方向由读写位（R/W）确定。

在一个地址周期后的所有传输都可看做是数据传输，即使当中包含从机设备的地址。每字节数据是 8 位。SDA 上的数据在 SCL 为低时发生改变为高时保持不变，见图 13-9。SCL 上一个时钟脉冲对应一个数据位，最高位被先发送。每个数据字节跟随在第 9（应答）位后，应答位是由接收设备标识返回的。通过在第 9 个时钟拉低 SDA 标识一个应答信号。简而言之，一个完整的数据通信需要九个时钟脉冲。

如果从机接收器在第九个位时间没有识别主机，那么 SDA 数据线被置高电平。主机认为该识别的失败是一次不成功的数据传输。

如果主机接收器在一个字节数据被发送后没有识别从机发送器，那么从机认为数据传输过程结束并释放 SDA 数据线。

无论哪一种情况，数据传送是失败的，主机将会做下面两件事情之一：

- 产生一个停止信号从而放弃总线
- 产生一个重新开始信号开始一个新的传输

13.4.1.4 停止信号

主机可以通过产生一个停止信号结束本次数据通信以释放总线。随后产生一个开始信号。这别称为重新开始。停止信号被定义为当 SCL 为逻辑 1 且 SDA 由低电平转变为高电平产生停止信号。（见图 13-9）即使从机已经产生一个应答信号主机也可产生一个停止信号结束数据通信。

13.4.1.5 重新开始信号

如图 13-9 所示，重新开始信号是一个不带结束通信停止信号的开始信号。它被用于主机与保持总线的另一个从机或不同模式下相同的从机相互通信。

13.4.1.6 仲裁程序

IIC 总线是多主机总线允许多个主机与它连接。如果在同一时间有两个或更多主机试图取得总线控制权，一个时钟同步程序决定该总线时钟，该总线时钟低电平周期等于最长时钟的低电平周期，高电平周期等于最短时钟时钟的高电平周期。这些竞争主机的相对优先级是由一个数据仲裁程序，一个总线主机丢失仲裁来决定的。未取得总先控制权的主机立即转为从机接收模式并停止 SDA 输出。在此情况下从主机模式到从机模式的跳变不产生停止条件。同时一个状态位通过硬件被置位表明仲裁丢失。

13.4.1.7 时钟同步

因为 SDA 数据实行线与逻辑，SCL 时钟线上的一个由高到低的跳变将影响所有连接到总线上的设备。设备开始计数其低电平周期并在一个设备时钟变低后，它将保持 SCL 时钟线为低电平直到高电平状态来到。不论怎样，如果另一个设备时钟仍然处于低电平那么该设备时钟由低变高的转变可能不会改变 SCL 时钟线的状态。因此同步时钟 SCL 被保持于最长低电平周期。在这期间带有较短低电平周期的设备进入高等待状态（见图 13-10）。当所有设备结束其低电平周期时，同步时钟 SCL 时钟线被释放并拉高。设备时钟，SCL 时钟线状态和设备高电平周期意义上是没有区别的。第一个设备完成其高电平周期再次拉低 SCL 时钟线。

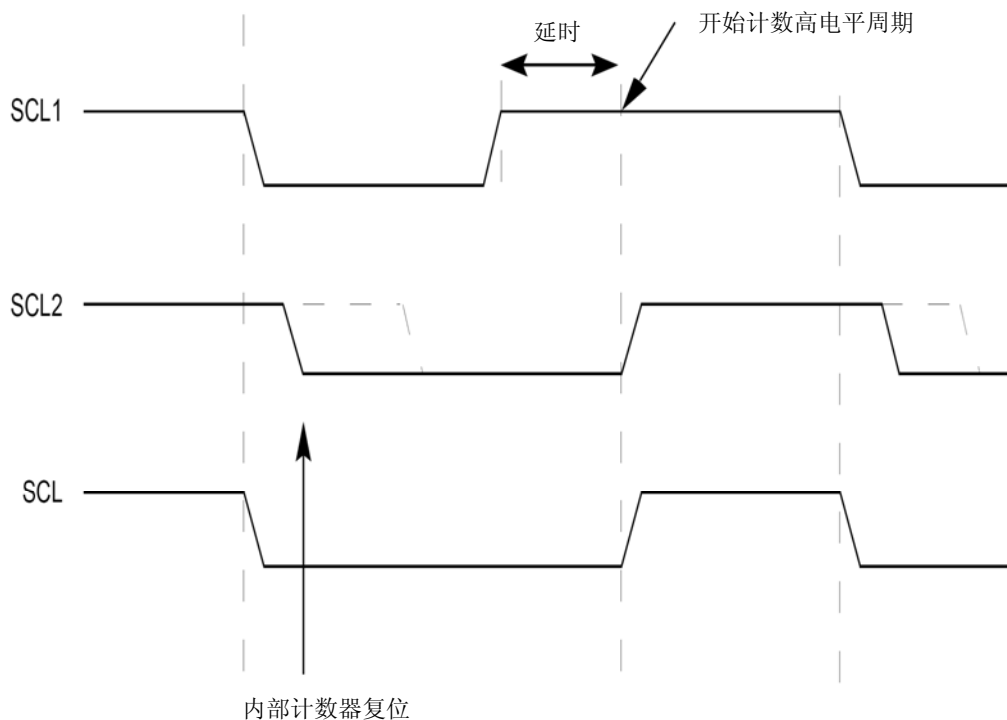


图 13-10 IIC 时钟同步

13.4.1.8 握手

时钟同步机制在数据发送中可以被用作握手机制。从设备在完成一个字节（9 位）发送后保持 SCL 低电平。在这种情况下，握手信号暂停总线时钟并强制主机时钟进入等待状态直到该从机释放 SCL 时钟线。

13.4.1.9 时钟扩展

时钟同步机制可被从机使用来降低发送的位速度。在主机完成驱动 SCL 时钟线为低后从机在必需的情况下可以驱动它为低电平然后释放它。如果从机时钟线的低电平周期比主机的长那么意味着 SCL 总线信号低电平周期被延伸。

13.4.2 10-位地址

对于 10 位寻址，0X11110 被用于第一个地址的头 5 位。包含 10 位寻址的传送的读/写形式的不同组合是可能的。

13.4.2.1 主机发送从机接收

发送数据的方向不改变（见表 13-9）。当在开始条件后发送一个 10 位地址时，每个从机将该地址的头七位与自己的地址相比并测试第八位（读写方向位）是否为 0。超过一个设备能够匹配该地址并返回一个应答信号（A1）。然后每个从机用自身地址与第二个字节的八位相比较。仅有一个从机匹配并产生一个应答信号（A2）。该匹配的从机保持被主机寻址直到它接收到一个停止信号（P）或一个重新开始信号（Sr）。

表 13-9 10 位地址主机发送从机接收

S	从机地址头7位 11110+AD10+AD9	R/W 0	A1	从机地址第 二个字节 AD[8:1]	A2	数据	A	...	数 据	A/A	P
---	---------------------------	----------	----	--------------------------	----	----	---	-----	--------	-----	---

在主机发送器发送 10 位地址的第一个字节,从机接收器见 IIC 中断。软件必须确保 IICD 的内容被忽略且不被这次中断的有效数据。

13.4.2.2 主机接收从机发送

发送方向在第二个读写位后被改变（见表 13-10）。截至并包括应答位 A2，程序与主机发送从机接收方式相同。在重新开始信号后（Sr），一个匹配的从机记住它已被寻址。从机稍后检测从机地址的第一个字节的头七位是否与他们的相同，然后测试第八位（读写位）是否为 1。如果存在匹配，从机认为它已被作为发送方被寻址并产生应答信号 A3。该从机发送器保持寻址直到它接收到停止信号（P）或重新开始信号（Sr）。

在重新开始信号之后（Sr），所有其它从设备也用自身地址与该从机地址第一个字节的头七位相比较并测试第八位（读写位）。如果读写位为 1（10 位地址设备）或从机地址 11110XX 未匹配的话，他们不能被寻址。

表 13-10 10 位地址主机接收从机发送

S	从机地址头7位 11110+ AD10+ AD9	R/ W 0	A1	从机地址第二个字节 AD[8:1]	A2	Sr	从机地址头7位 11110+ AD10+ AD9	R/ W 1	A3	数据	A	...	数据	A/A	P
---	-----------------------------------	--------------	----	----------------------	----	----	-----------------------------------	--------------	----	----	---	-----	----	-----	---

在主机接收器发送 10 位地址的第一个字节后，从机发送器注意到一个 IIC 中断。软件必须确保 IICD 的内容被忽略且不被这次中断的有效数据。

13.4.3 一般寻址

一般寻址被要求是 7 位或 10 位。如果 GCAEN 位被置位，IIC 匹配一般寻址即它自身的从机地址。当 IIC 响应一般寻址时，它可以作为从机接收器且在该地址周期后 IAAS 位被置位。在第一个字节传送后软件必须读取 IICD 寄存器来决定该地址是否等于它自身的从机地址或一个一般寻址。如果该值是 00，该匹配是一般寻址。如果 GCAEN 位被清零，IIC 忽略任何来自不带应答信号的一般寻址数据。

13.5 复位

IIC 在复位后被禁止。IIC 不能触发一个 MCU 复位。

13.6 中断

IIC 产生一个单一中断。

若 IICIE 位被置位，当表 13-11 中任何事件发生时产生一个 IIC 中断。中断由 IICIF 位驱动，用 IICIE 位屏蔽。在终端服务例程中 IICIF 位必须通过软件方法写一个 1 给它来清零。你可以通过读状态寄存器来决定中断类型。

表 13-11 中断摘要

中断源	状态	标志	使能
完成一个字节传送	TCF	IICIF	IICIE
标准寻址匹配	IAAS	IICIF	IICIE
仲裁丢失	ARBL	IICIF	IICIE

13.6.1 字节传送中断

TCF 位（传送完成标志）在第九个时钟的下降沿被置位以表明一个字节传送完成。

13.6.2 地址检测中断

当寻址地址等于预设的从机地址（IIC 地址寄存器）或 GCAEN 位被置位接收到一般寻址时，状态寄存器中的 IAAS 位被置位。若 IICIE 被置位，则 CPU 被中断。CPU 必须检测 SRW 位并进入发送模式。

13.6.3 仲裁丢失中断

IIC 是一个多主机总线允许超过一个主机与它连接。如果在同一时间有两个或更多的主机试图取得总线控制权，这些竞争主机的相对优先级由一个数据仲裁程序决定。当它丢失该数据仲裁程序且状态寄存器中的 ARBL 位被置位时，IIC 模块确认中断。

在下列情形下仲裁丢失：

- 在一个地址或数据传送周期内主机为高电平时，SDA 数据线采样为低电平。
- 在一个数据接收周期的应答位内主机为高电平时，SDA 数据线采样为低电平。
- 当总线忙时试图开始一个周期。
- 在从机模式下一个重新开始周期被请求。
- 当主机没有请求停止信号时一个停止信号被检测到。

该位必须通过软件方法写 1 来清零。

13.7 初始化/应用信息

模块初始化（从机）

1. 写IICC2
 - 禁止或允许一般寻址
 - 选择10位或7位寻址模式
2. 写IICA
 - 设置从机地址
3. 写IICCC1
 - 使能IIC和中断
4. 为传送数据初始化RAM变量（IICEN=1，IICIE=1）
5. 初始化RAM变量用来完成图13-12中所示的例程

模块初始化（主机）

1. 写：IICF
 - 设置IIC波特率
2. 写IICCC1
 - 使能IIC和中断
3. 为传送数据初始化RAM变量（IICEN=1，IICIE=1）
4. 初始化RAM变量用来完成图13-12中所示的例程
5. 写IICC1
 - 使能TX
6. 写：IICCC1
 - 使能MST（主机模式）
7. 写：IICD
 - 目标从机的地址

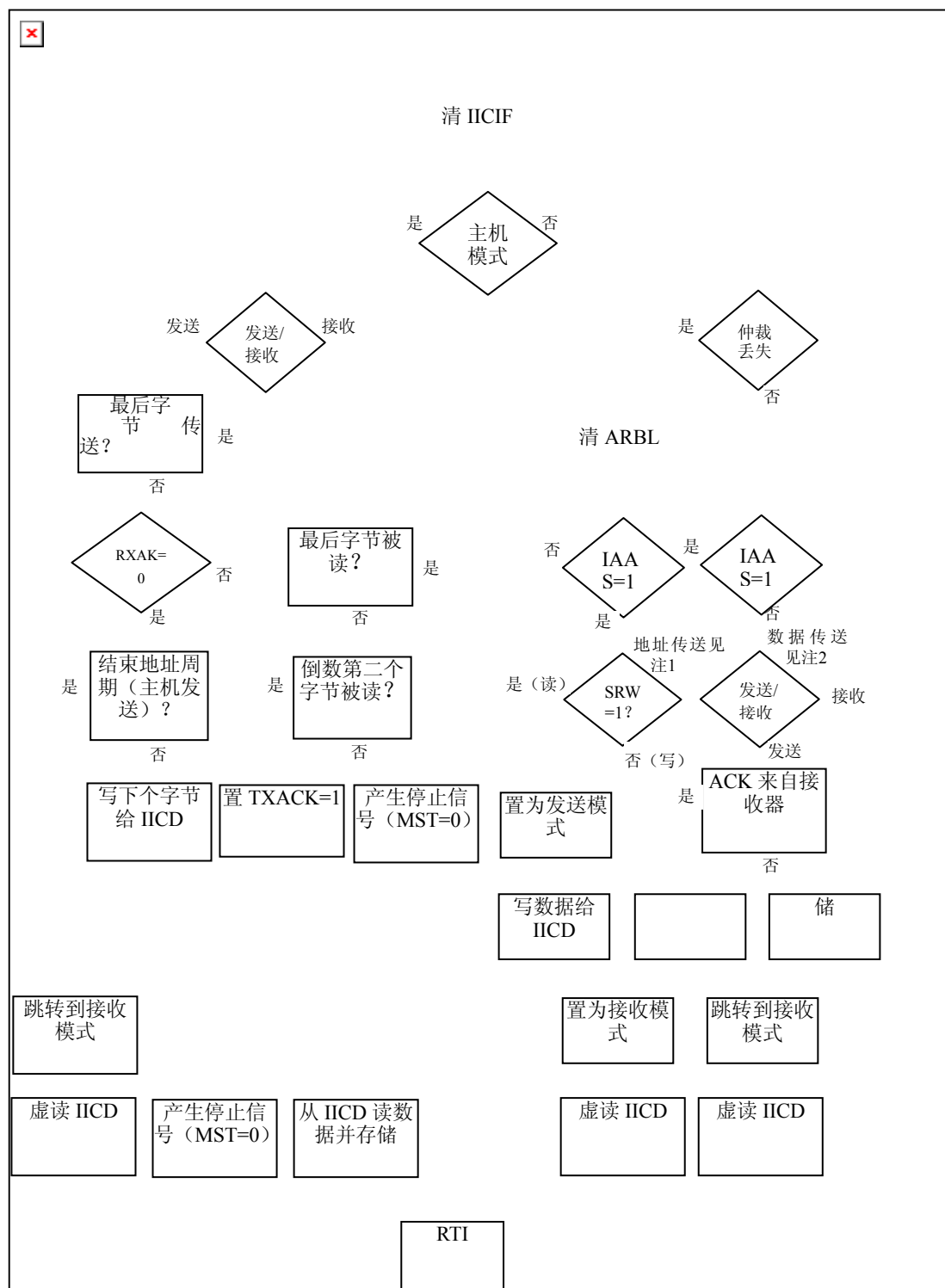
模块使用

图13-12所示的例程可以处理主机和从机IIC操作。对于从机操作来说，一个输入非IIC信息包含适当的地址开始IIC数据通信。对于主机操作来说，数据通信必须通过想写IICD寄存器来开始。

寄存器模块

IICA										AD[7:1]																									
当作为从机被寻址时（在从机模式下），该模块响应此地址																																			
IICF				MULT				ICR																											
波特率= BUSCLK / (2 x MULT x (SCL DIVIDER))																																			
IICC1				IICEN				IICIE				MST				TX				TXAK				RSTA				0		0					
模块配置																																			
IICS				TCF				IAAS				BUSY				ARBL				0				SRW				IICIF				RXAK			
模块状态标志																																			
IICD				DATA																															
数据寄存器：发送和读取IIC数据																																			
IICC2				GCAEN				ADEXT				0				0				0				AD10				AD9				AD8			

图13-11 IIC模块快速启动



注:

- 1.如果一般调用允许，必须执行一个检测来决定接收到的地址是否是一个一般寻址（0X00）。如果是的话，然后由用户软件来处理它。
- 2.当 10 位地址被用来寻址从机，在扩展地址的第一个字节后从机注意到一个中断。用户软件必须确保此中断，IICD 的内容被忽略且不作为一个有效数据传送

第十四章 AD转换器(S08ADC10V1)

14.1 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 MC9S08AC16 系列的 MCU 中只实现了九个输入（AD0-AD3、AD8-AD11、AD27）。这些输入可以通过 ADCH 位选择。如图 14-1 所示，一些输入和 I/O 复用。表 14-1 概述了 MC9S08AC16 系列设备的 ADC 中所有通道的分配。

14.2 通道分配

MC9S08AC16 系列设备的 ADC 通道分配如表 14-1 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。连接到 I/O 引脚上的通道有一个相应的引脚控制位。

表 14-1 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	PTB0/AD1P0	ADPC0	10000	AD16	V_{REFL}	N/A
00001	AD1	PTB1/AD1P1	ADPC1	10001	AD17	V_{REFL}	N/A
00010	AD2	PTB2/AD1P2	ADPC2	10010	AD18	V_{REFL}	N/A
00011	AD3	PTB3/AD1P3	ADPC3	10011	AD19	V_{REFL}	N/A
00100	AD4	V_{REFL}	N/A	10100	AD20	V_{REFL}	N/A
00101	AD5	V_{REFL}	N/A	10101	AD21	V_{REFL}	N/A
00110	AD6	V_{REFL}	N/A	10110	AD22	Reserved	N/A
00111	AD7	V_{REFL}	N/A	10111	AD23	Reserved	N/A
01000	AD8	PTD0/AD1P8	ADPC8	11000	AD24	Reserved	N/A
01001	AD9	PTD1/AD1P9	ADPC9	11001	AD25	Reserved	N/A
01010	AD10	PTD2/AD1P10	ADPC10	11010	AD26	Temp Sensor	N/A
01011	AD11	PTD3/AD1P11	ADPC11	11011	AD27	Internal Bandgap	N/A
01100	AD12	V_{REFL}	N/A	11100		Reserved	N/A
01101	AD13	V_{REFL}	N/A	11101	V_{REFH}	V_{REFH}	N/A
01110	AD14	V_{REFL}	N/A	11110	V_{REFL}	V_{REFL}	N/A
01111	AD15	V_{REFL}	N/A	11111	module disable	None	N/A

注：

选择内部的能带隙通道要求 SPMSC1 中的 BGBE=1，参见 5.9.8 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”。

14.2.1 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），或者交替时钟，ALTCLK。MC9S08AC16 系列 MCU 设备的交替时钟是内部时

钟产生器模块的外部参考时钟（ICGERCLK）。

因为 ICGERCLK 只有在外部时钟源允许时才有效，ICG 必须设置成 FBE 或 FEE 模块（CLKS1=1）。ICGERCLK 必须运行在一个频率，所以 ADC 转换时钟（ADCK）运行在具体范围（ f_{ADCK} ）的频率，在 ALTCLK 输入分频（由 ADIV 位决定）之后。例如，如果 ADIV 位被设置到 4 分频，然后 ALTCLK（ICGERCLK）的最小频率是 f_{ADCK} 的最小值的四倍。最大值是 f_{ADCK} 的最大值的四倍。因为最小频率需求，当一个振荡器电路使用时，它必须设置为高范围操作（RANGE=1）。

当 MCU 处于等待模式时，如果以上描述的条件满足，ALTCLK 有效。当 MCU 处于等待模式时，这允许 ALTCLK 可以作为 ADC 的转换时钟源使用。当 MCU 处于 stop3，ALTCLK 不能用作 ADC 转换时钟源

14.2.2 硬件触发器

ADC 硬件触发器，ADHWT，是实时中断计数器的输出。RTI 计数器可以使用 ICGERCLK 或 RTI 时钟内的 1kHz 时钟源作为时钟。MCU 处于运行，等待，stop3 模式时，1kHz 时钟源可以使用。ICG 设置为 FBE 或 FEE 模式，MCU 处于允许或等待模式时，ICGERCLK 可以使用。

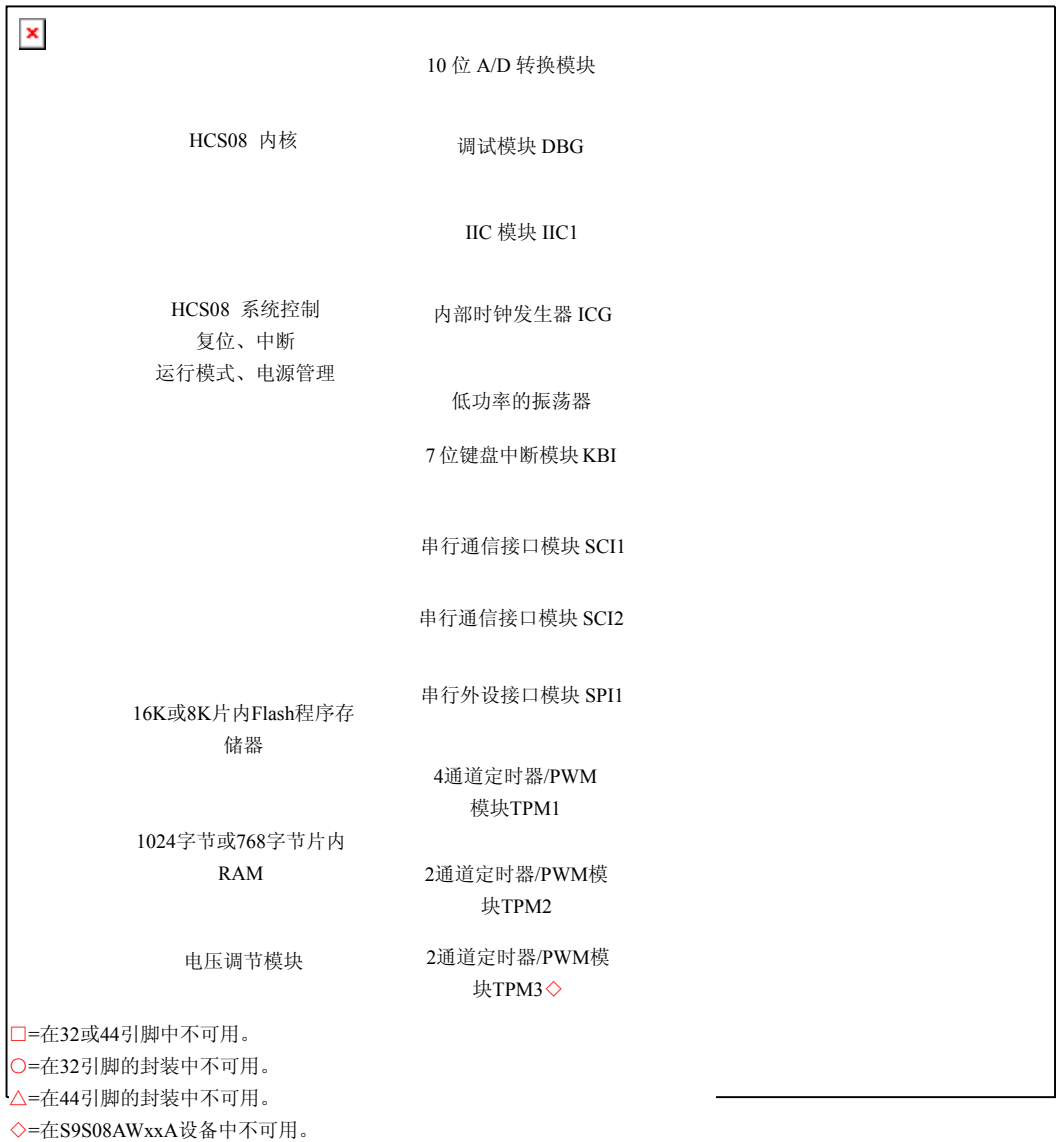
RTI 的周期可以由输入时钟和 RTIS 位决定。当允许 ADC 硬件触发器时，RTI 计数器溢出时，初始化一个转换。RTI 计数器是一个自由运行计数器，按照 RTIS 位决定的 RTI 率产生一个溢出。

14.2.2.1 模拟引脚使能

MC9S08AC16 系列的 ADC 只包含两个模拟引脚使能寄存器，APCTL1 和 APCTL2。

14.2.2.2 低功耗模式操作

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



- 注意：
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. 若IRQ使能(IRQPE=1)，引脚包括可软件配置的上下拉设备。若选择了上升沿检测（IRQEDG=1），下拉使能。
 3. IRQ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
 4. 引脚包含集成的上拉设备。
 5. PTD3,PTD2和PTG4引脚包含上拉/下拉设备。当KBI使能（KBIPEn=1）而且上升沿被选择（KBEDGn=1），下拉使能。

图14-1 MC9S08AC16框图

14.2.3 温度传感器

ADC1 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。等式 14-1 提供了温度传感器的近似转换函数。

Temp=25- ((VTEMP -VTEMP25) /m) （等式 14-1）

其中：

- VTEMP 是温度传感器通道在周围温度时的电压。
- VTEMP25 是温度传感器通道在 25℃时的电压。
- M 是冷或热温度斜率（V/℃）。

计算温度时，使用 ADC 电气表中的 V_{TEMP25} 和 m 的值。

在应用代码中，用户读温度传感器通道，计算 V_{TEMP} 并且和 V_{TEMP25} 比较。如果 V_{TEMP} 大于 V_{TEMP25} 在等式 14-1 中使用冷斜率值。如果 V_{TEMP} 小于 V_{TEMP25} 在等式 14-1 中使用热斜率值。

更多使用温度传感器的信息参考 AN3031。

14.2.4 特点

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。
- 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
- 温度传感器

14.2.5 框图

图 14-2 提供了 ADC 模块的框图。

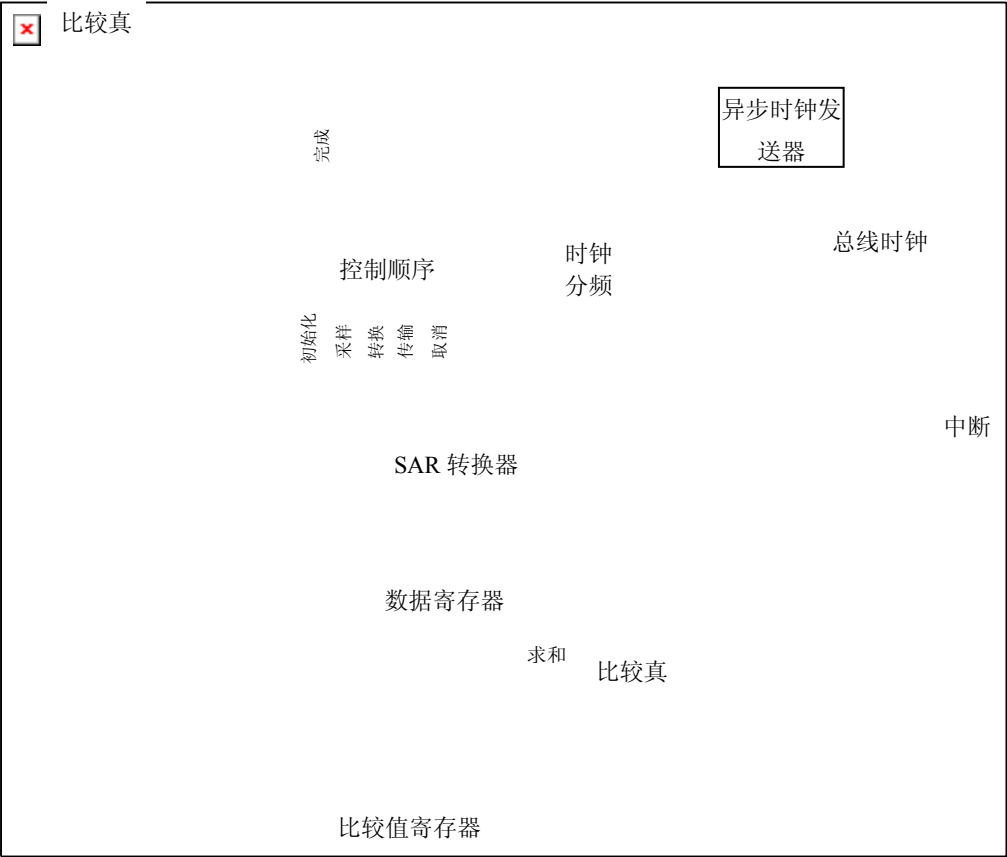


图14-2 ADC框图

14.3 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 14-2 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

14.3.1 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 VDD。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。

14.3.2 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。

14.3.3 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。

14.3.4 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。

14.3.5 模拟通道输入（ ADx ）

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。

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

14.4.1 状态和控制寄存器1（ $ADC1SC1$ ）

本节描述 ADC 状态和控制寄存器（ $ADC1SC1$ ）的功能。写 $ADC1SC1$ 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14-3 状态和控制寄存器($ADC1SC1$)

表 14-3 $ADC1SC1$ 寄存器域描述

域	描述
7 COCO	转换完成标志——COCO标志是一个只读位。当比较功能禁止（ $ACFE=0$ ）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 $ADC1SC1$ 或读 $ADC1RL$ ，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ADC1SC1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ADC1SC1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14-4中描述。 当通道选择位设置为全1时，逐次逼近转换器子系统关闭。这个特点允许禁止ADC和从所有的源中孤立输入通道。

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14-4 输入通道选择

14.4.2 状态和控制寄存器2（ADC1SC2）

ADC1SC2 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

图14-5 状态和控制寄存器(ADC1SC2)

表 14-4 ADC1SC2 寄存器域描述

域	描述
7 ADACT	转换行为——ADACT表示转换正在进行中。当初始化转换时，ADACT置位；当转换完成或终止时，ADACT清零。 0 转换未进行。 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ADC1SC1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。

14.4.3 数据高结果寄存器（ADC1RH）

ADC1RH 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，ADC1RH 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 ADC1RL 被读，再读 ADC1RH 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 ADC1RL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADC1RL 的互锁。在这种情况下，MODE 位被改变，ADC1RH 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14-6 数据高结果寄存器(ADC1RH)

14.4.4 数据低结果寄存器（ADC1RL）

ADC1RL 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 ADC1RL 被读，再读 ADC1RH 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 ADC1RL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADC1RH 的互锁。在这种情况下，MODE 位被改变，ADC1RL 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14-7 数据低结果寄存器(ADC1RL)

14.4.5 比较值高寄存器（ADC1CVH）

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，ADC1CVH 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14-8 比较值高寄存器(ADC1CVH)

14.4.6 比较值低寄存器（ADC1CVL）

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	ACVR4	ADCV3	ADCV2	ADCV1	ADCV0
写								
复位	0	0	0	0	0	0	0	0

图14-9 比较值低寄存器(ADC1CVL)

14.4.7 配置寄存器（ADC1CFG）

ADC1CFG 用于选择操作模式，时钟源，时钟分频和低功耗或长采样时间的设置。

	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14-10 配置寄存器(ADC1CFG)

表 14-5 ADC1CFG 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14-6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14-7。
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14-8。

表 14-6 时钟分频选择

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

表 14-7 转换模式

模式	模式描述
00	8位转换（N=8）
01	保留
10	10位转换（N=10）
11	保留

表 14-8 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)

14.4.8 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14-11 引脚控制1寄存器(APCTL1)

表 14-9 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。

14.4.9 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14-12 引脚控制2寄存器(APCTL2)

表 14-10 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。

14.4.10 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14-13 引脚控制1寄存器(APCTL2)

表 14-11 APCTL2 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。

14.5 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（ADC1RH 和 ADC1RL）。在 10 位模式中，结果四舍五入成 10 位放在 ADC1RH 和 ADC1RL 中。在 8 位模式中，结果四舍五入成 8 位放在 ADC1RL 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。

14.5.1 时钟选择和分频控制

可以选择 4 个时钟源中的一个作为 ADC 模块的时钟源。这个时钟源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 ADICLK 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。

14.5.2 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。

14.5.3 硬件触发

ADC 模块有一个可选的异步硬件转换触发器，ADHWT，当 ADTRG 位置位时，它被允许。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。

14.5.4 转换控制

可以使用 8 位或 10 位模式转换，由 MODE 位决定。一个软件或硬件触发可以初始化转换。另外，ADC 模块可以设置为低功耗操作，长采样时间，连续采样，自动比较转换值和软件决定的比较值。

14.5.4.1 初始化转换

满足以下条件，即初始化转换：

- 如果选择软件触发操作，在写 ADC1SC1 之后（ADCH 不是全 1）。
- 如果选择硬件触发操作，在一个硬件触发（ADHWT）事件之后。
- 当允许连续转换时，在将数据传到数据寄存器之后。

如果允许连续转换，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 ADC1SC1 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。

14.5.4.2 完成转换

当转换的结果传到数据结果寄存器，ADC1RH 和 ADC1RL 后，转换完成。通过置位 COCO 表示。如果 AIEN 是高，在 COCO 置位时会产生一个中断。

在 10 位模式中，如果数据正在被读（ADC1RH 已经被读但是 ADC1RL 还未被读），闭锁机制保护新的数据不会重写在 ADC1RH 和 ADC1RL 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。

14.5.4.3 终止转换

下列情况发生时，任何正在出来的转换都会终止：

- 写 ADC1SC1 发生（如果 ADCH 不是全 1，当前的转换被取消并开始了一个新的转

换,)。

- 写 ADC1SC2、ADC1CFG、ADC1CVH 或 ADC1CVL 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 (ADC1RH 和 ADC1RL) 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, ADC1RH 和 ADC1RL 返回到它们的复位值。

14.5.4.4 电源控制

直到初始化一个转换, ADC 模块都保持空闲。如果 ADACK 被选作转换时钟源, ADACK 时钟产生器也被允许。

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)。

14.5.4.5 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 ADC1RH 和 ADC1RL。

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 14-12 中总结了不同条件下的最大的总转换时间。

表 14-12 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, 100	0	20ADCK周期+5总线时钟周期
单个或第一个连续转换10位	0x, 100	0	23ADCK周期+5总线时钟周期
单个或第一个连续转换8位	0x, 100	1	40ADCK周期+5总线时钟周期
单个或第一个连续转换10位	0x, 100	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:

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

$$\text{总线时钟的个数} = 3.5 \mu\text{s} \times 8 \text{ MHz} = 28 \text{ (周期)}$$

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。

14.5.5 自动比较功能

比较功能可以设置为检测上限或下限。采样和转换输入后，结果和比较值（ADC1CVH 和 ADC1CVL）的补数相加。比较上限时（ACFGT=1），如果结果大于或等于比较值，COCO 置位。比较下限时（ACFGT=0），如果结果小于比较值，COCO 置位。转换结果和比较值的补数相加后产生的值传送到 ADC1RH 和 ADC1RL。

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。

14.5.6 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。

14.5.7 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。

14.5.7.1 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 ADC1RH 和 ADC1RL 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。

14.5.7.2 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。

14.5.8 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。

14.6 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 14-6、表 14-7 和表 14-8。

注：十六进制的前缀是 0x，二进制的前缀是%，十进制没有前缀。

14.6.1 ADC模块初始化举例

14.6.1.1 初始化顺序

在 ADC 模块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。

14.6.1.2 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Freescale使用；写为0

ADCSC1=0x41（%01000001）

Bit 7	COCO	0	只读标志当转换完成时置位
Bit 6	AIEN	1	转换完成中断使能
Bit 5	ADCO	0	仅一次转换（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。

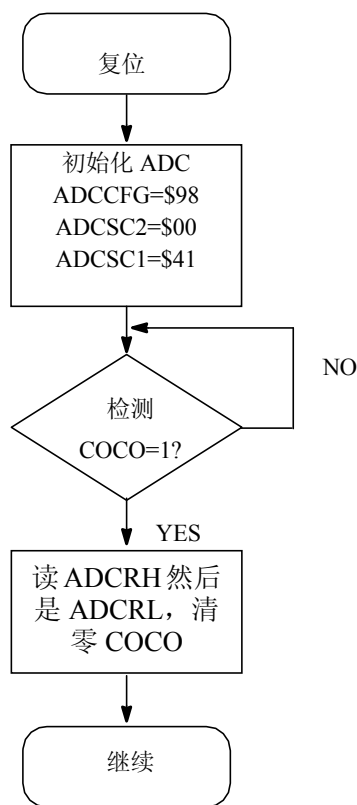


图14-14 该例的初始化流程

14.7 应用信息

该节包含了在应用中使用 ADC 模块的信息。ADC 被设计集成了一个微控制器可以使用在需要 A/D 转换器的嵌入式控制应用中。

14.7.1 外部引脚和安排

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。

14.7.1.1 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。

14.7.1.2 模拟参考引脚

除了模拟电源，ADC 模块连接了两个参考电压输入。高参考电源是 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最小化。

14.7.1.3 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。试验 $0.01\mu F$ 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。

14.7.2 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。

14.7.2.1 采样错误

为正确转换，输入必须被采样足够长时间才能达到合适的精度。如果最大输入电阻 $7k\Omega$ 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。

14.7.2.2 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 $N=10$ ）

14.7.2.3 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。
- V_{DDAD} 和 V_{SSAD} 之间有一个 $0.1\ \mu F$ 的低内阻电容。
- 如果电源使用感应隔离，一个 $1\ \mu F$ 的电容放在 V_{DDAD} 和 V_{SSAD} 之间。
- V_{SSAD} （和 V_{REFL} ，如果连接）连接到 V_{SS} （连着地平面的点）。
- 初始化（硬件触发转换）前或刚初始化（软件或硬件触发转换）ADC 转换，MCU 处于等待或 stop3 状态。
 - 对于软件触发转换，用 WAIT 或 STOP 指令写 ADC1SC1 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 VDD 噪音。在这些情况，或当 MCU 不能在等待状态或 I/O 行为停止，这些推荐的行为可能减少影响正确性的噪音：

- 在选择输入通道和 V_{REFH} 或 V_{REFL} 之间放置一个 $0.01\ \mu F$ 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。

14.7.2.4 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：

$$1\text{LSB} = (V_{REFH} - V_{REFL}) / 2^N \quad (\text{等式 14-2})$$

数字化结果存在固有的量化误差。对于 8 位或 10 位转换。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时 1.5LSB 。

14.7.2.5 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。

14.7.2.6 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 14.7.3.2 节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。

第十五章 开发支持

15.1 简介

HCS08 系列的开发支持系统包括背景调试控制器(BDC)和片上调试模块(DBG)。BDC 提供一个单线调试界面给目标 MCU 为在线 FLASH 和其它非易失性存储器编程提供方便。BDC 也是开发首要选择的调试接口，允许非插入存取内存数据和传统的调试功能如 CPU 寄存器内容修改，断点，单步跟踪。

在 HCS08 系列中，地址和数据总线信号在外部引脚上不可用（甚至在调试模式下）。通过单线背景调试接口输入指令到目标 MCU 完成调试。该调试模块提供一种方式去有选择地触发和捕捉总线信息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。

MC9S08AC16 系列 MCU 的 BDC 时钟源是 ICGLCLK 提供。关于 ICGLCLK 如何选择时钟源详情参阅第八章“内部时钟发生器”。

15.1.1 特征

BDC 模块特征如下：

- 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下时看门狗禁止。

ICE 系统特征如下：

- 两个触发比较器：两个地址+读 / 写(R/W)或一个全址+数据+R/W。
- 灵活的 8 字(每字 16 位)的 FIFO 缓冲用于捕捉信息：
 - 地址改变
 - 事件数据
- 两种类型的断点：
 - 标记指令操作码的断点
 - 用于地址访问的强制断点
- 9 种触发模式：只事件 A、A 或 B、A 然后 B、A 与 B 数据(全模式)、A 与非 B 数据(全模式)、只事件 B(存储数据)、A 然后只事件 B(存储数据)、内部范围($A \leq \text{地址} \leq B$)、外部范围(地址 $< A$ 或 地址 $> B$)。

15.2 背景调试是控制器

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，如以太网或通用串行总线(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、RESET (如果有复位引脚)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。

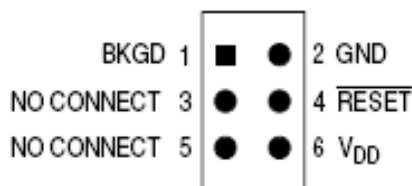


图15-1 BDM接头

15.2.1 BKGD引脚描述

BKGD 是单线的背景调试接口引脚。背景调试系统中的所有指令和双向数据都是通过 BKGD 引脚进行通信的。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。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。

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。关于通信协议的详细描述参阅 15.2.2 节“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。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 是一个虚开漏引脚且芯片上有上拉电阻所以不需要外部上拉电阻。不像传统的开漏引脚，外部 RC 时间在这个引脚上是一个常量，它还要受到外部电容的影响，对信号上升时间基本没有影响。在这个引脚上通用协议可简洁地使用驱动加速脉冲强制加快周期，而不需要冒着与标准驱动冲突的风险。更多的细节请看 15.2.2 “通信细节”。

当没有调试器连接到 6 针 BDM 的接口连接器上时, BKGD 上的内部上拉选择正常操作模式。当调试器相连时, 它可能强制 MCU 在复位后进入背景调试模式。强制进入背景调试模式的具体条件取决于 HCS08 核(参阅“开发支持系统”)。复位目标 MCU 后通过背景调试接口与它通信不是必需的。

15.2.2 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 的操作模式标准的串行通信协议要求调试头识别 BDC 通信时钟速度。

BDC 状态和控制寄存器中的时钟转换控制位 (CLKSW) 允许用户选择 BDC 时钟源。可以是总线时钟或交替的 BDC 时钟。

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 15-2 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, 在主机到目标的发送期间, 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。

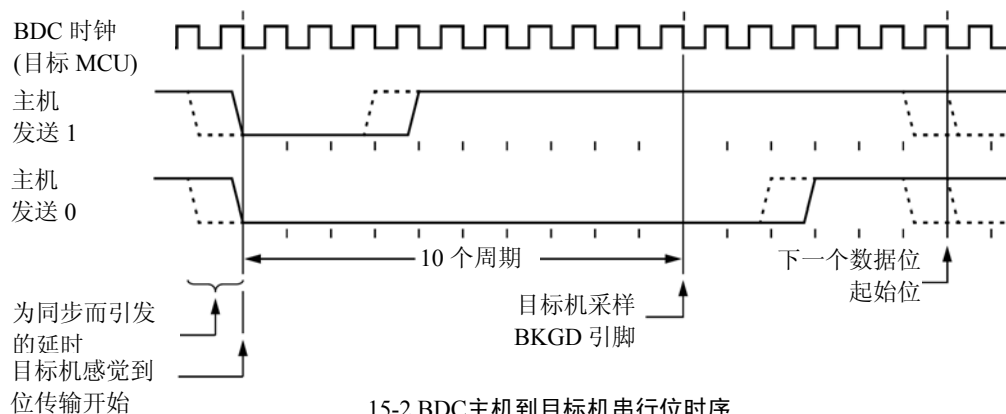
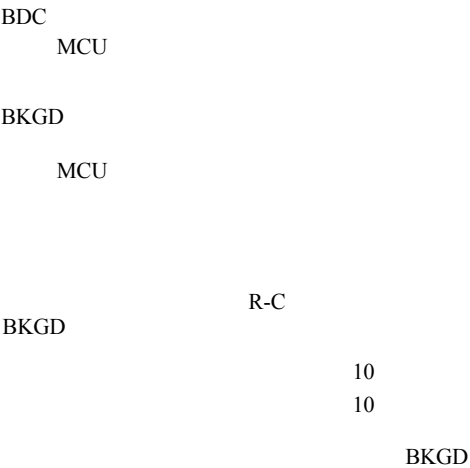
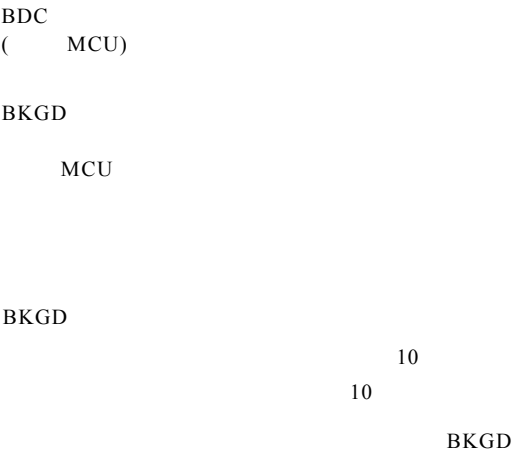


图 15-3 表示主机接收来自目标 HCS08MCU 逻辑 1 的图例。对于目标机来说主机是异步的, 所以从主机在 BKDG 引脚上产生下降沿到目标机感知到位开始有 0 ~ 1 个周期的延迟。主机在 BKGD 引脚上长时间保持低电平以便目标机来识别它(至少两个目标 BDC 周期)。从目标 MCU 感知到位时间开始后的第七个周期, 目标 MCU 将主动驱动一个简短的高电平加速脉冲, 在此之前主机必须释放低电平驱动。主机在位时间开始后的第 10 个周期对位电平进行采样。



15-3 BDC目标机到主机串行位时序(逻辑1)

图 15-4 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。



15-4 BDC目标机到主机串行位时序(逻辑0)

15.2.3 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 15-1 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。后面的文章更详细的描述了每条指令。

译码结构命名：

这些命名被用在表 15-1 中描述 BDC 指令的代码结构。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 7-1 BDC 命令汇总

命令命名	背景模式/非介入式	指令组成 ⁽¹⁾	描述
SYNC	非介入式	n/a ⁽²⁾	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

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 命令确认, 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(总线时钟或派生的具体的后备时钟源)
- 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。

15.2.4 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)两中类型的断点来实现。

片上调试模块包括两路额外的硬件断点电路，这样比 BDC 模块的简单断点要更加灵活。

15.3 片上调试系统（DBG）

由于 HCS08 设备没有外部地址和数据总线，故电路仿真功能作为最主要的模块会被集成在 MCU 中。调试系统包含有 8 级 FIFO 队列用于存放地址总线 and 数据总线的信息，和一个灵活的触发式系统用于决定何时捕获总线信息和捕获哪些总线信息。该系统通过背景调试系统(或 CPU)来访问调试控制寄存器，并读取 8 级 FIFO 队列的值。

与背景调试控制器不同，这个调试模块包含了在用户的存储映像图中可以访问到的控制和状态寄存器。这些寄存器被定位在高地址空间，以避免使用宝贵的直接页的地址空间。大多数调试模块的功能被用于开发期间，而且用户程序很少会访问调试模块的控制和状态寄存器。但有一个特例是一个 ROM 修补(ROM patching)。关于 ROM 修补内容的更多细节也将在 15.3.6 硬件断点讨论。

15.3.1 比较器A和B

通过 R/W 信号和操作码跟踪电路可以对两个 16 位比较器(A 和 B)进行访问。独立的控制位能够使各个控制器忽略 R/W 信号。如果在指定地址的操作码是可执行的，操作跟踪电

路能指定一个触发器触发。比较器同样能在内部和外部触发模式之间进行大量的比较操作。通过 BDC 访问可以使比较器暂时禁止。

A 比较器总与 16 位 CPU 地址有关。通过所选择的触发模式，B 比较器可以判断出是 16 位的 CPU 地址还是 8 位 CPU 数据总线。由于 CPU 数据总线分为读数据总线和写数据总线，RWAEN 和 RWA 控制位就是用来决定哪个总线用于比较。RWAEN=1(enable)并且 RWA=0(write)，则 CPU 的写数据总线被使用。否则，CPU 的读数据总线被使用。

当比较器检测到一个匹配条件时，当前选择的触发模式将决定哪种调试逻辑。匹配条件有：

- CPU 断点的产生
- 数据总线数值存入 FIFO
- 开始将流程改变地址(change-of-flow addresses)存入 FIFO(起始类型标记)
- 停止将流程改变地址(change-of-flow addresses)存入 FIFO(结束类型标记)

15.3.2 总线信息捕捉和FIFO操作

使用 FIFO 的通常做法是设置触发器模式或选择另外控制方法，然后启动调试器。当 FIFO 满或调试器停止向 FIFO 存数据时，可以按先前存入的顺序从 FIFO 中取出数据。状态位表示 FIFO 存入数据时数据的有效信息。如果一个跟踪运行在 FIFO 满之前 (CNT=1:0:0:0) 通过写 0 给 ARM 自动停止，那么数据移一个位置和主机必须执行 ((8-CNT)-1) 虚读 FIFO 以置它为最高优先级进入 FIFO。

在大多数触发模式，存在 FIFO 中的信息包含有流程改变地址(16 位)。在这种情况下，读 DBGFL 时 DBGFL 将从 FIFO 中取得一个字的信息。读 DBGFL(FIFO 数据端口的低半段)能使 FIFO 移位，使得下一字信息移到 FIFO 数据口。在事件触发模式(event-only trigger modes)下，8 位数据存于 FIFO。这时 FIFO 的高字节(DBGFH)没有被使用(总是保存和读取 0s)，数据只是简单的通过 DBGFL 从 FIFO 中取出。每次 DBGFL 被读时 FIFO 都会移位，移位后 DBGFL 的数据就是下一次要读的数据。

在触发模式下，FIFO 用来存储流程改变地址，在 CPU 地址放入 FIFO 时会会有一个延时在输入方之间产生。这种延时的结果是：如果触发事件自身是一个流程改变，或者当一个触发事件触发 FIFO 后一个流程改变出现在两个总线周期之后，在这种情况下，这个延时不会被保存在 FIFO 中。在跟踪结束的情况下，如果触发的事件是流程改变，它将被保存在上一个的流程改变入口，以便被调试运行。

当调试器未被保护时，FIFO 也可以产生被执行指令的地址仿真。当 ARM=0 时，读 DBGFL 将使最近使用的操作码的地址存储到 FIFO。使用仿真功能，主机调试器可通过读 DBGFH 从 FIFO 里取地址然后读 DBGFL。头八个值将被丢失因为读相应八位 DBGFL 需要先填满 FIFO。另外周期性读 DBGFH 和 DBGFL 返回关于已执行指令的延时信息所以主机调试器可以开发已执行指令的仿真。

15.3.3 流程改变信息

为了使存于 FIFO 的信息量最小化，只有涉及到能使正常的顺序执行的指令改变的信息才被存储。将源码和目标代码程序存于目标系统中，外部调试系统就能通过存储于 FIFO 的流程改变指令来重新构造执行路径。

对于条件分支指令当条件成立时，条件分支指令的源地址被存储。如果外部调试器在 FIFO 中找到了这样一个地址，就可以认为已经发生转移。因为 BRA 和 BRN 指令是可预见的，它们不会产生流程改变信息存于 FIFO。

间接 JMP 和 JSR 指令使用 H:X 变址寄存器来取得目标地址。因为这个原因，对于 JMP 或 JSR 指令，由调试系统来存储运行时的目标地址。然而对于其他 JMP 和 JSR 指令。对于中断 RTI、RTS 来说这些目标地址被当作流程改变信息保存到 FIFO。

15.3.4 标记与强制断点和触发器

标记是一个术语，它指示出指令操作码被取到指令队列时的识别方法，它不做任何动作直到指令被 CPU 执行。这点比较重要，因为任何来自跳转指令、分支语句、子程序或中断所导致的流程改变会使有些指令被指令队列扔掉而不执行。

强制断点会等当前指令完成后，再发出断点请求。通常断点的响应动作是转入背景模式而不是继续执行下一用户程序的指令。

标记断点与强制断点相比有两种不同的上下文。第一种是到来自 CPU 调试模式的断点请求。而第二种是来自于调试控制逻辑的比较器发出的信号。当标记中断请求发向 CPU 后，此信号会随其操作码被放入指令队列，若此操作码曾被执行过，CPU 将用 BGND 操作码代替标记操作码，因此 CPU 可以继续处在背景模式下而不需退出来执行标记指令。

当 DBGT 寄存器中的 TRGSEL 控制位被设置成选择标记断点操作时，来自 A 和 B 比较器的输出通过调试模块的逻辑块进行验证，只有比较地址被执行时，跟踪操作码和调试器才引起触发。每个比较器都有操作码跟踪逻辑，所以不止一个比较事件能在同一时间通过重建的指令队列被跟踪到。

15.3.5 触发模式

触发模式控制着所有的调试动作。在 DBGT 寄存器中的 4 位 TRG 字段可用于选择 9 个触发模式中的一个。当 DBGT 寄存中 TRGSEL 控制位为 1 时，比较器的输出必须在 FIFO 动作执行之前使用操作码跟踪电路比较。DBGT 中的 BEGIN 位决定，当检测到匹配触发时(追踪开始)FIFO 开始储存数据，或是 FIFO 以循环方式存储数据直到匹配触发被检测到(触发结束)。

调试运行开始于设置 DBGT 寄存器，并且在寄存器 DBGCR 的 ARM 位写入一个 1 来设置 ARMF 标记，并且清零标记位 A 和 B 和 DBGS 的 CNT 位。当 FIFO 满时一个开始标记调试运行结束。当被选择的触发事件发生时一个结束标记会结束。任何调试运行都可通过写 0 到 DBGCR 的 ARM 位或 DBGCR 的 ARM 位来人工的结束。

在所有的触发模式下，比较器 A 和 B 的匹配情况是由读写和通道重建逻辑(rebuild logic)的情况决定。R/W 比较功能是通过 RWxEN 控制位来设置，并且它可被认为是一个连接到比较器的外加输入。在整个触发模式下，RWAEN/RWA 可用来进行读写的比较，并且控制数据比较器来读写 CPU 的数据总线，此时 RWBEN/RWB 被忽略。当 TRGSEL = 1，这个 R/W 限制匹配条件被写入到指令通道重建逻辑(instruction pipe rebuild logic)，那么这个触发不会产生直到标记操作码达到通道重建逻辑的尾部。在事件触发模式下，TRGSEL 被忽略，并且匹配信息不会通过通道重建逻辑被认定。

起始标记运行后，当满足触发条件时 FIFO 将被填入信息，而当 FIFO 将被填满(CNT[3:0] = 1:0:0:0)时运行结束。结束标记运行后，当 ARM 位设为 1 时，运行以循环方式开始填充 FIFO，并且当遇见触发状态时结束。结束标记可在 FIFO 满之前结束运行。在一个结束标记运行期间，如果有超过八个输入被存到 FIFO 中，新输入会覆盖老的输入，这样将使得当调试运行结束时，在 FIFO 的信息是八个最近的流程改变地址。

在除了事件触发模式外所有的触发模式，FIFO 将存储流程改变的地址。在事件触发模式下，FIFO 将数据存储到 FIFO 中的低八位。

BEGIN 控制位在事件触发模式下被忽略且所有此类调试动作开始类型跟踪。当 TRGSEL=1 时选择操作码触发时，不一定要使用比较器中的读/写位因为操作码标记仅使用在操作码存取上，通常一直为读周期。在使用一种全触发方式时指明 trgsel = 1 将是不寻常的，因为操作码值通常是已知在某一地址了。

下面的触发模式描述仅列举了引起触发的首要比较条件。通过设置 RWAEN(RWBEN) 比较器可做 R/W 使用且相应的 RWA(RWB)再次与 R/W 匹配。如果 BRKEN=1 来自带选择 R/W 资格比较器的信号被用来请求一个 CPU 断点，TAG 决定 CPU 请求是一个标记请求还是强制请求。

- 单独 A 触发模式 — 当地址匹配比较器 A 中的值时触发
- A 或 B 触发模式 — 当地址匹配比较器 A 或比较器 B 中的值时触发
- A 然后 B 触发模式 — 当地址匹配比较器 B 中的值但仅在该地址另一个周期之后匹配比较器 A 中的值时触发。在匹配后 B 匹配之前存在一个周期。
- A 和 B 数据触发(全模式) — 所谓全模式是指为了导致一个触发地址，数据和可选的 R/W 必须在同一地址总线周期都匹配。比较器 A 检查地址，比较器 B 的低字节检查数据，若 RWAEN=1 读/写检查反 RWA。比较器 B 的高位不使用。

在全模式下指定一个标记类型的 CPU 断点是无用的 (BRKWN=TAG=1) 但是如果那样做了，因为确认 CPU 的标记请求比较器 B 的数据匹配将被忽略，当比较器 A 地址匹配时 CPU 断点被确认。

- A 与非 B 数据触发(全模式) — 地址必须与比较器 A 相匹配，数据必须与比较器 B 低位相匹配，若 RWAEN=1 读/写必须与 RWA 相匹配。在同一个总线周期三个条件必须都被满足才能触发。

在全模式下指定一个标记类型的 CPU 断点是无用的 (BRKWN=TAG=1)，但是如果那样做了，因为确认 CPU 的标记请求比较器 B 的数据匹配将被忽略，当比较器 A 地址匹配时 CPU 断点被确认。

- 事件 B 触发模式(存储数据) — 触发发生在每次地址与比较器 B 中的值相匹配时。触发事件引起数据被捕捉并存入 FIFO。当 FIFO 空时调试动作结束。
- A 然后事件 B 触发模式(存储数据) — 在地址与比较器 A 中的值相匹配之后，触发发生在每次地址与比较器 B 中的值相匹配时。触发事件引起数据被捕捉并存入到 FIFO。当 FIFO 满时调试动作结束。
- 触发范围内: $A \leq \text{地址} \leq B$ — 当地址同时大于或等于比较器 A 中的值且小于或等于比较器 B 中的值时触发发生。
- 触发范围外: 地址 < A 或者 地址 > B — 当地址小于比较器 A 中的值或大于比较器 B 中的值时触发发生。

15.3.6 硬件断点

DBGC 寄存器中的 BRKEN 控制位可被置位 1 来允许 15.3.5 节描述的触发条件用来产生一个硬件断点请求给 CPU。标记 DBGC 控制位以辨别断点请求将被作为一个标记类型断点还是强制类型断点。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。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。

如果背景模式被通过 BKGD 引脚上的串行 WRITE_CONTROL 命令禁止的话，CPU 将执行 SWI 指令而不是进入背景模式。

15.4 寄存器定义

这部分包含对 BDC 和 DBG 寄存器和控制位的描述。

参阅 DBG 寄存器绝对地址分配数据表中高地址页寄存器摘要。这部分所涉及到的寄存器和控制位均以名字检索。飞思卡尔提供头文件将名字映射成绝对地址。

15.4.1 BDC寄存器和控制位

BDC 有两个寄存器：

- BDC 状态和控制寄存器(BDCSCR)是一个包含控制和状态位的 8 位寄存器，这些控制和状态位用于背景调试控制器。
- BDC 断点寄存器(BDCBKPT)是用于保存一个 16 位断点匹配地址。

这些寄存器由一系列专门的 BDC 命令访问，但并不位于目标 MCU 的存储空间内(因此它们没有地址也不能由用户程序访问)。

BDCSCR 中的一些位有写限制；另外，这些寄存器可以在任意时刻被读和写。例如，当 MCU 在背景模式时，ENBDM 控制位是不能被写的(当 MCU 已经处于背景模式时，这将阻止控制位的不确定状态影响到背景模式)。同样，四个状态位(BDMACT、WS、WSF 和 DVF)也是只读状态位，它们永远不能由 BDC 命令的写控制写数。时钟开关(CLKSW)控制位在任何时刻都可以被读写。

15.4.1.1 BDC状态和控制寄存器（BDCSCR）

BDC 状态和控制寄存器(BDCSCR)可由 BDC 命令进行读写，但由于没有位于 MCU 的正常的内存中，所以用户程序不能访问。

	7	6	5	4	3	2	1	0
读		BDMACT	BKPTN	FTS	CLKSW	WS	WSF	DVF
写								
正常复位	0	0	0	0	0	0	0	0
激活BDM下复位	1	1	0	0	1	0	0	0

 =未定义的

图15-5 BDC状态和控制寄存器（BDCSCR）

表 15-2 BDCSCR 寄存器位描述

位	描述
7 ENBDM	BDM允许位(允许背景调试模式) 一典型的, 该位在调试开始后不久, 由调试主机写为1, 或者当调试主机复位目标机并保留1直到正常复位清零它。 1 = 激活BDM并允许背景调试模式指令。 0 = 不激活BDM(仍然允许非介入指令)。
6 BDMACT	BDMACT——背景模式状态位这是一个只读状态位。 1 = 激活BDM并等待一系列指令。 0 = BDM没有激活。
5 BKPTEN	BKPTEN——BDC断点允许位如果该位被清零, 则BDC断点不可用, 同时FTS控制位和BDCBKPT匹配寄存器将被忽略。 1 = 允许BDC断点。 0 = 禁止BDC断点。
4 FTS	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。 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)。 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。
3 CLKSW	CLKSW——BDC通信时钟源选择位, 默认为0, 可以选择可变的BDC时钟源。 1 = CPU总线时钟。 0 = 明确的后备频率源
2 WS	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, 在此所以的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。 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。 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 WSF	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。 1 = 由于CPU进入等待或停止模式, 内存存取指令失败。 0 = 内存存取没有与等待或停止指令发生冲突。
0 DVF	DVF——数据有效失败状态位, 该状态位不使用在MC9S08AC16系列因为它不支持低速存取。 1 = CPU在带有低速内存存取时内存进入指令失败。 0 = 内存进入没有和低速内存存取发生冲突。

15.4.1.2 BDC断点匹配寄存器（BDCBKPT）

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 15.2.4 BDC 硬件断点。

15.4.2 系统背景调试强制复位寄存器（SBDFR）

该寄存器包含一个单写控制位。一个背景模式命令如 `WRITE_BYTE` 必须被写到 SBDFR。从用户程序尝试去写该寄存器被忽略。通常读为 0X00。

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

 =未定义的

¹BDFR 仅通过串行背景模式调试命令可写而不是来自用户程序。

图 15-6 系统背景调试强制复位寄存器（SBDFR）

表 15-3 SBDFR 寄存器位描述

位	描述
0 BDFR	背景调试强制复位——一个背景模式命令如 <code>WRITE_BYTE</code> 允许一个外部调试主机强制一个目标系统复位。给这位写1强制MCU复位。该位不能被用户程序改写。

15.4.3 DBG寄存器和控制位

调试模块包含有 9 个字节的寄存器空间，用于安排 3 个 16 位寄存器和 3 个 8 位控制和状态寄存器。这些寄存器位于通用内存映射图的高地址空间中，所以能被普通用户程序访问。这些寄存器是很少被用户应用程序使用的，除非是一个基于 ROM 调试监控或者使用断点逻辑的 ROM 修补机制。

15.4.3.1 调试比较器 A 的高地址页寄存(DBGCAH)

这个寄存器存有比较器 A 的高 8 位值。复位后被强制设置为 \$00，任何时候可被读出，但只有 DBGCR 中的 ARM 位没有被置位时才能被写。

15.4.3.2 调试比较器 A 的低位寄存器(DBGCAL)

这个寄存器存有比较器 A 的低 8 位值。复位后被强制设置为 \$00，任何时候可被读出，但只有 DBGCR 中的 ARM 位没有被置位时才能被写。

15.4.3.3 调试比较器 B 的高地址页寄存(DBGCAH)

这个寄存器存有比较器 B 的高 8 位值。复位后被强制设置为 \$00，任何时候可被读出，但只有 DBGCR 中的 ARM 位没有被置位时才能被写。

15.4.3.4 调试比较器 B 的低位寄存器(DBGCAL)

这个寄存器存有比较器 B 的低 8 位值。复位后被强制设置为 \$00，任何时候可被读出，但只有 DBGCR 中的 ARM 位没有被置位时才能被写。

15.4.3.5 调试FIFO高地址页寄存器(DBGFHF)

这个寄存器在表示了 FIFO 的高 8 位，只能被读取。写这个寄存器时无效。在事件触发模式，FIFO 只用低 8 位存储信息，故而此时该寄存器无用，且读出来数据始终为 \$00。读取 DBGFHF 不能让 FIFO 执行移位动作。当从 FIFO 中读出 16 位数据时，DBGFHF 要比 DBGFL 先读，因为读 DBGFL 会使得 FIFO 移位。

15.4.3.6 调试FIFO低位寄存器(DBGFL)

这个寄存器提供对 FIFO 低 8 位值的只读访问。写这个寄存器没有意义和影响。

读取 DBGFL 将使得 FIFO 移位而获取下一字节的值。当调试模块工作于事件模式下时，只有 8 位数据存于 FIFO(每个 FIFO 字的高半字节没有被使用)。此时读 FIFO 相当于读 DBGFL 寄存器。在这种情况下并不需要读取 DBGFLH，数据全部来源于 DBGFL。

当 FIFO 锁定时(在 FIFO 被填满或 ARMF 被清零前没有解锁)不要尝试读数据。因为 FIFO 在读 DBGFL 期间受到保护。这可能导致从 FIFO 取数据的动作顺序不正确。

在调试器并未锁定的情况下读 DBGFL 将导致最新的操作码地址被存于 FIFO 的最后位置。通过定期地读取 DBGFLH 和 DBGFL，外部主机调试软件能获取程序执行的概况。经过 8 次 FIFO 读数后第 9 次将返回第一次读取结果的信息。使用其辅助的特性，可读 FIFO 前 8 次数据，而不需用新数据去填充 FIFO 序列，然后使用这些数据取得被执行地址处的延迟。

这个可以被 DBGFL 读取的信息被保存在 FIFO(FIFO 没有锁定)，该信息是最近被执行的操作码地址。

15.4.3.7 调试控制寄存器 (DBGC)

该寄存器可在任意时间被读写。

	7	6	5	4	3	2	1	0
读	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
写								
复位	0	0	0	0	0	0	0	0

图 15-7 调试控制寄存器 (DBGC)

表 15-4 DBGC 寄存器位描述

位	描述
7 DBGEN	调试模块使能—用于使能调试模块，当MCU保护时DBGEN不能置1。 1=DBG使能 0=DBG禁止
6 ARM	锁定控制—控制调试器是否比较和存储FIFO中的信息。当向这位置1时调试器完成后会自动清这一位。当向ARM或DBGEN写0时任何调试将被人为的停止。 1=调试器锁定 0=调试器不锁定
5 TAG	标记/强制选择—决定发向CPU的中断请求是标记请求还是强制请求。当BRKEN为0时，此位无效。 1=CPU中断请求是标记类型 0=CPU中断请求是强制类型
4 BPKEN	中断使能位—决定触发事件是否会向CPU产生中断请求。触发事件能让信息存于FIFO中而不需产生中断请求。当比较器和R/W符合条件时，CPU会发出一个CPU中断。CPU标记响应与读取操作码对应的，所以TGRSEL将在CPU发出中断时不受影响。 1=触发引起CPU中断请求 0=CPU中断请求禁止
3 RWA	比较器A的R/W比较值—当RWAEN为1时，这位决定比较器A的读写验证。当RWAEN为0时，RWA和读写信号不影响比较器A 1=比较器A只在读周期比较 0=比较器A只在写周期比较
2 RWAEN	使能比较器A的读写—控制读写信号是否与比较器A比较 1=读写信号用于比较器A 0=读写信号不用于比较器A
1 RWB	用于比较器B的读写比较值—当RWAEN为1时，这位决定比较器B的读写验证。当RWAEN为0时，RWA和读写信号不影响比较器B 1=比较器B只在读周期比较 0=比较器B只在写周期比较

0 RWBEN	使能比较器B的读写—控制读写信号是否与比较器B比较 1=读写信号用于比较器B 0=读写信号不用于比较器B
------------	--

15.4.3.8 调试触发寄存器（DBGT）

该寄存器任意时间可读但仅当 ARM=0 时可写，其中第 4 位和第 5 位被硬件固定为 0。

	7	6	5	4	3	2	1	0
读	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
写								
复位	0	0	0	0	0	0	0	0

 =未定义的

图 15-8 调试触发寄存器（DBGT）

表 15-5 DBGT 寄存器位描述

位	描述
7 TRGSEL	触发类型选择位—这一位控制在调试模式中，来自比较器A和B的匹配的输出是否满足代码跟踪逻辑的要求。如果这一位被置1，那么一个来自比较器A和B的匹配信号必须被传递到代码跟踪逻辑，同时如果地址匹配的代码真正的被执行，那么一个触发事件信号将只被传递到先进先出逻辑。 1=如果比较地址的代码被执行，那么触发 0=如果对比较地址进行存储，那么触发
6 BEGIN	开始/结束触发选择位—这一位控制先进先出逻辑是在触发事件开始填充还是以循环的方式来填充直到一个触发事件结束捕获信息。在事件触发模式下，这一位会被忽略而且所有的调试运行会被视为刚开始跟踪。 1=触发事件产生数据存储(开始跟踪) 0=数据一直存储直到有触发事件才停止(停止跟踪)
3:0 TRG[3:0]	选择触发模式—选择九中触发模式中的一种，如下所述。 0000 仅A 0001 A或B 0010 先A后B 0011 仅事件B(存储数据) 0101 先A后仅事件B(存储数据) 0110 A和B数据(全模式) 0111 A和非B数据(全模式) 1000 范围：A<=地址<=B 1001-1111（无触发）

15.4.3.9 调试状态寄存器（DBGS）

这是一个只读状态寄存器。

	7	6	5	4	3	2	1	0
读	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
写								
复位	0	0	0	0	0	0	0	0

 =未定义的

图 15-9 调试状态寄存器（DBGS）

表 15-6 DBGS 寄存器位描述

位	描述
7	触发匹配A标志—当调试开始时，这一位会被清零。同时，该位表明ARM生效后，是

AF	否触发匹配A的条件被满足。 1=比较器A匹配 0=比较器A没有被匹配
6 BF	触发匹配B标志—当调试开始时，这一位会被清零。同时，该位表明ARM生效后，是否触发匹配B的条件被满足。 1=比较器B匹配 0=比较器B没有被匹配
5 ARMF	Arm标志位—当DBGEN为1时，这一状态位是DBGC寄存器的ARM位的一个只读映像。当DBGEN为1时，通过对DBGC寄存器的ARM位写1可以置ARMF位。当调试运行结束时，这一位会被自动清零。当先进先出逻辑满时，或者当检测到触发事件时，调试运行会被结束。可以通过对DBGC的ARM位或者DBGEN位写0来人为地结束调试运行。 1=调试器锁定 0=调试器没有被锁定
3:0 CNT[3:0]	先进先出逻辑的有效计数—这些位在调试运行开始时会被清零。这些位表明了当调试运行结束时，先进先出逻辑里的有效的数据字的个数。当先进先出逻辑里的数据被读出时，CNT里的值不会减小。当先进先出逻辑里的数据被读出时，外部的调试主机负责跟踪CNT里的值。 0000先进先出逻辑的有效计数=0 0001先进先出逻辑的有效计数=1 0010先进先出逻辑的有效计数=2 0011先进先出逻辑的有效计数=3 0100先进先出逻辑的有效计数=4 0101先进先出逻辑的有效计数=5 0110先进先出逻辑的有效计数=6 0111先进先出逻辑的有效计数=7 1000先进先出逻辑的有效计数=8

附录A 电气特性和时序规范

A.1 概述

该节包含了电气和时序规范。

A.2 参数分类

在该附录的电气参数被多种方法保证。为了让客户更好的理解，使用了下面的分类，在适当的地方使用表中相应的标记。

表 A-1 参数分类

P	这些参数被保证，在每个产品设备上测试产品
C	这些参数通过设计描述完成，测量了统计的相关试样量（包含了过程变化）。
T	这些参数通过设计描述完成，测量一个小的典型条件的试样量，除非另有说明，所有显示在典型列里面的数据在此分类中。
D	这些参数主要通过仿真得到。

注：

在表中用适当的地方“C”标记来标注。

A.3 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-2 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-2 最大绝对额定值

额定值	标志	值	单位
电源电压	V_{DD}	-0.3 到 +5.8	V
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 的瞬间最大电流	I_{DD}	± 25	mA
V_{DD} 的最大电流	I_{DD}	120	mA
存储温度	T_{stg}	-55 to +150	°C
最大连接温度	T_J	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} 的瞬间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.4 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 $P_{I/O}$ ，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-3 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	$^{\circ}\text{C}$
热敏电阻 ^{1, 2, 3, 4} 48 引脚 QFN	θ_{JA}	- 0.3 to $V_{DD} + 0.3$	$^{\circ}\text{C/W}$
1s		84	
2s2p		27	
44 引脚 LQFP			
1s		73	
2s2p		56	
32 引脚 LQFP			
1s		85	
2s2p		56	

- (1) 连接温度是模具，片上功耗，封装的热敏电阻，底座（板）温度，周围环境温度，气流，板上其他元件的功耗和板上热敏电阻之间的功能。
- (2) 外部自然转换的连接。
- (3) 1s——单层板，单层信号层。
- (4) 2s2p——4 层板，2 层信号层和 2 层电源层。

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

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

其中：

T_J = 周围环境温度， $^{\circ}\text{C}$

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

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

$P_{int} = I_{DD} \times V_{DD}$ ，瓦特——芯片内部电源

$P_{I/O}$ = 输入和输出引脚上的功耗——用户决定

在大多数应用中， $P_{I/O} \ll P_{int}$ ， $P_{I/O}$ 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 A-1 和 A-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 A-1 和 A-2 可以得到 P_D 和 T_J 。

A.5 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-4 ESD 和闭锁测试条件

模型	描述	标志	值	单位
人体	串联电阻	R1	1500	Ω
	存储电容	C	100	PF
	每个引脚的脉冲数	-	3	
机器	串联电阻	R1	0	Ω
	存储电容	C	200	PF
	每个引脚的脉冲数	-	3	
闭锁	最小输入电压限制		-2.5	V
	最大输入电压限制		7.5	V

表 A-5 ESD 和闭锁保护特性

编号	C	额定值	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 $T_A=125^\circ\text{C}$ 时，闭锁电流	I_{LAT}	± 100	-	mA

A.6 DC特性

该节包含各种电源条件，I/O 引脚特性和不同操作模式的电源电流的信息。

表 A-6 DC 特性

编号	C	参数	标志	最小	典型 ¹	最大	单位
1	P	输出高电压——低驱动($PTxDSn=0$)	V_{OH}	$V_{DD}-1.5$	-	-	V
		5V, $I_{load}=-2\text{mA}$		$V_{DD}-1.5$	-	-	
		3V, $I_{load}=-0.6\text{mA}$		$V_{DD}-0.8$	-	-	
		5V, $I_{load}=-0.4\text{mA}$		$V_{DD}-0.8$	-	-	

2	P	输出低电压——高驱动(PTxDSn=1) 5V,Iload=-10mA 3V,Iload=-3mA 5V,Iload=-2mA 3V,Iload=-0.4 mA		$V_{DD}-1.5$ $V_{DD}-1.5$ $V_{DD}-0.8$ $V_{DD}-0.8$	- - - -	- - - -	V
		输出低电压——低驱动(PTxDSn=0) 5V,Iload=-2mA 3V,Iload=-0.6mA 5V,Iload=-0.4mA 3V,Iload=-0.24mA		1.5 1.5 0.8 0.8	- - - -	- - - -	
	C	输出高电压——高驱动(PTxDSn=1) 5V,Iload=-10mA 3V,Iload=-3mA 5V,Iload=-2mA 3V,Iload=-0.4 mA	V_{OL}	1.5 1.5 0.8 0.8	- - - -	- - - -	
3	P	输出高电流——所有端口最大总 I_{OH} 5V 3V	I_{OHT}	-	-	100 60	mA
4	P	输出高电流——所有端口最大总 I_{OH} 5V 3V	I_{OLT}	-	-	100 60	mA
5	P	输入高电压；所有的数字输入 $2.7V \leq V_{DD} \leq 4.5V$ $4.5V \leq V_{DD} \leq 5.5V$	V_{IH} V_{IH}	$0.75 \times V_{DD}$ $0.65 \times V_{DD}$	- -	- -	V
6	P	输入低电压；所有的数字输入	V_{IL}	-	-	$0.35 \times V_{DD}$	
7	P	输入滞后；所有的数字输入	V_{hys}	$0.06 \times V_{DD}$			V
8	P	输入泄露电流；仅输入引脚 ²	$ V_{in} $	-	0.1	1	μA
9	P	高阻抗（掉电状态）泄露电流 ²	$ V_{OZ} $	-	0.1	1	μA
10	P	内部上拉电阻 ³	R_{PU}	20	45	65	k Ω
11	P	内部下拉电阻 ⁴	R_{PD}	20	45	65	k Ω
12	C	输入电容；所有的非电源引脚	C_{In}	-	-	8	pF
13	P	POR 电压	V_{POR}	0.9	1.4	2.0	V
14	D	POR 时间	t_{POR}	10	-	-	μs
15	P	低压检测阈——大范围 V_{DD} 下降 V_{DD} 上升	V_{LVDH}	4.2 4.3	4.3 4.4	4.4 4.5	V
16	P	低压检测阈——小范围 V_{DD} 下降 V_{DD} 上升	V_{LVDL}	2.48 2.54	2.56 2.62	2.64 2.7	V
17	P	低压警告阈——大范围 V_{DD} 下降 V_{DD} 上升	V_{LVWH}	4.2 4.3	4.3 4.4	4.4 4.5	V

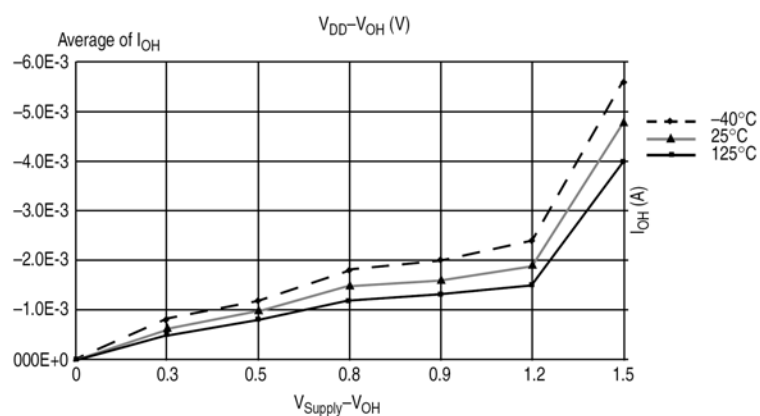
18	P	低压警告阈——小范围 V_{DD} 下降 V_{DD} 上升	V_{LVWL}	2.48 2.54	2.56 2.62	2.64 2.7	V
19	P	低压禁止复位/还原滞后 5V 3V	V_{hys}	- -	100 35	- -	mV

(1) 除非说明，典型值为25℃描述的数据。

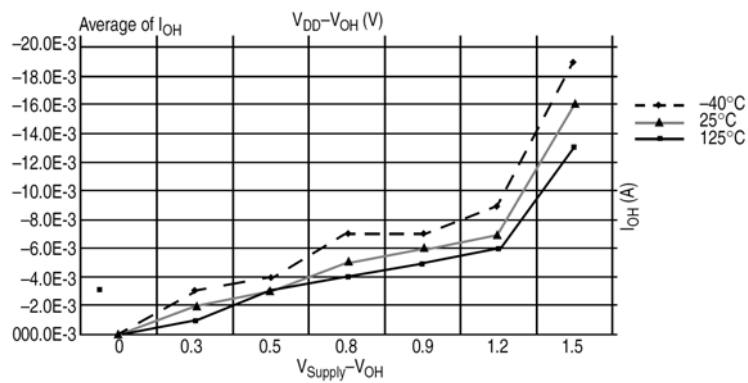
(2) 用 $V_{In} = V_{DD}$ 或 V_{SS} 测量。

(3) 用 $V_{In} = V_{DD}$ 测量。

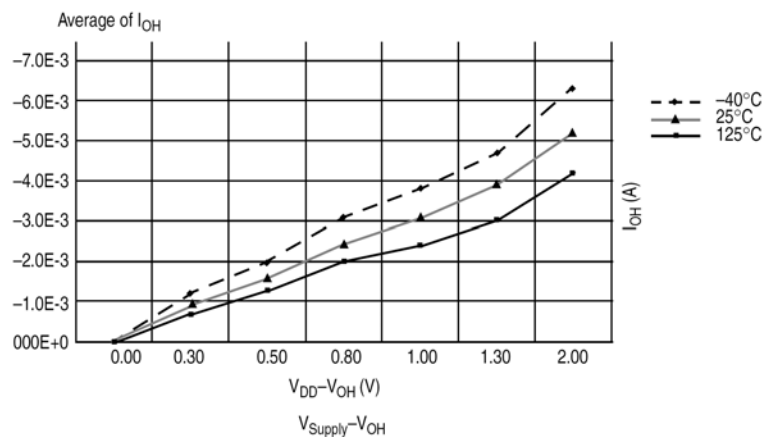
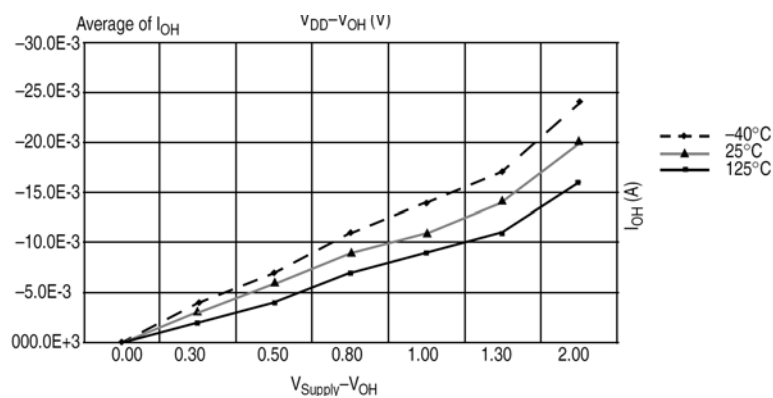
(4) 用 $V_{In} = V_{DD}$ 测量。



图A-1 $V_{DD}=3V$ 时, I_{OH} (低驱动)和 $V_{DD}-V_{OH}$ 的典型值



图A-2 $V_{DD}=3V$ 时, I_{OH} (低驱动)和 $V_{DD}-V_{OH}$ 的典型值

图A-3 $V_{DD}=5V$ 时, I_{OH} (低驱动)和 $V_{DD}-V_{OH}$ 的典型值图A-4 $V_{DD}=3V$ 时, I_{OH} (低驱动)和 $V_{DD}-V_{OH}$ 的典型值

A.7 电源电流特性

表 A-7 电源电流特性

编号	C	参数	标志	V_{DD} (V)	典型 ¹	最大 ²	单位	温度
1	C	运行时电源电流 ³ (CPU 时钟=2MHz, $f_{Bus}=1MHz$)	RI_{DD}	5	0.988	1.2 ⁴	mA	-40 到 125°C
				3	0.570	0.770		
2	C	运行时电源电流 ⁵ (CPU 时钟=2MHz, $f_{Bus}=1MHz$)	RI_{DD}	5	6.84	8 ⁶	mA	-40 到 125°C
				3	3.5	3.70		
3	C	Stop2 模式电源电流	$S2I_{DD}$	5	1.36	11 23 130 ⁴	μA	0 到 70°C -40 到 85°C -40 到 125°C
				3	1.05	10.5 21 110	μA	
4	C	Stop3 模式电源电流	$S3I_{DD}$	5	1.49	12 28 140 ⁴	μA	0 到 70°C -40 到 85°C -40 到 125°C

				3	1.15	11 26 120	μA	
5	C	带有 RTI 的 Stop2 或 stop3 ⁷	S3I _{DDRTI}	5	300	500 500	nA	-40 到 85℃ -40 到 125℃
				3	300	500 500	nA	-40 到 85℃ -40 到 125℃
6	C	带有 LVD 的 stop3 (LVDE=LVDSE=1)	S3I _{DDLVD}	5	116	150 180	μA	-40 到 85℃ -40 到 125℃
				3	90	TBD 160	μA	-40 到 85℃ -40 到 125℃
7	C	带有振荡器的 stop3 ⁸ (OSCSETN=1)	S3I _{DDOSC}	5,3	5	6	μA	-40 到 85℃
						TBD	μA	-40 到 125℃

- (1) 基于 25℃ 所描述的典型值，除非另有说明。对于电压和温度的典型曲线参见图 A-5，A-6，A-7。
- (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。等待模式通常 560 μA ，3V， f_{Bus} =1MHz。
- (8) 在下列条件下得到值，使用 32.768kHz 晶振小范围操作（RANGE=0），低功耗模式（HGO=0），禁止时钟监控（LOCD=1）。



注：外部时钟由功能产生器产生方波。对于FEE模式，外部参考频率是4MHz

图A-5 FBE和FEE模式中的典型运行 I_{DD} ， I_{DD} 和 V_{DD}



图A-6 典型的stop2 I_{DD}

图A-7典型的stop3 I_{DD}

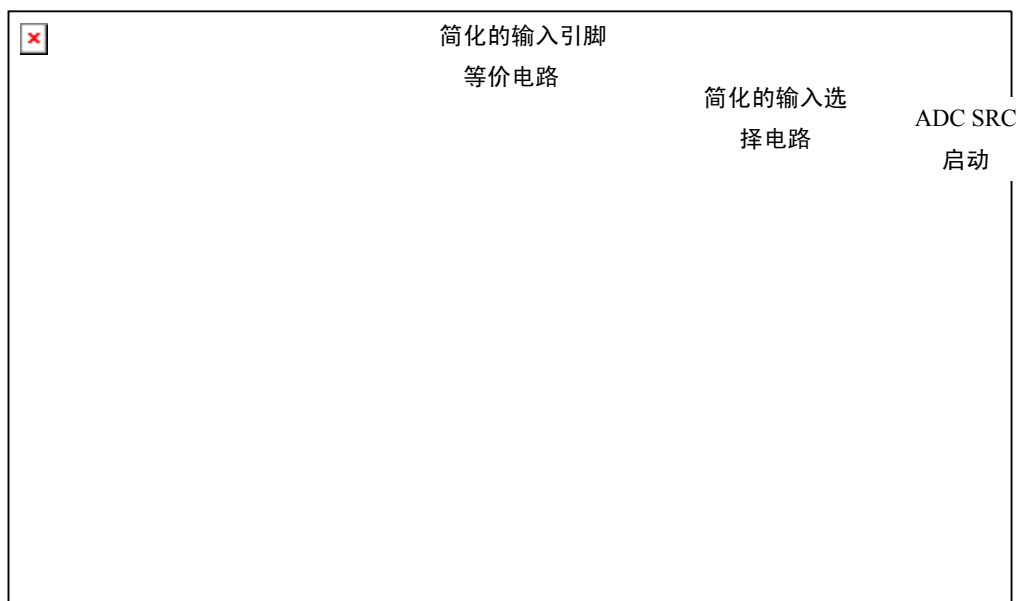
A.8 ADC特性

表 A-8 5V10 位 ADC 转换条件

特性	条件	标志	最小	典型 ¹	最大	单位
电源电压	绝对	V_{DDAD}	2.7	-	5.5	V
	Delta 到 $V_{DD} (V_{DD} - V_{DDAD})^2$	ΔV_{DDAD}	-100	0	+100	mV
地电压	Delta 到 $V_{SS} (V_{SS} - V_{SSAD})^2$	ΔV_{SSAD}	-100	0	+100	mV
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
电源电流	Stop 复位，关闭模块	I_{DDAD}	-	0.011	1	μA
输入电压		V_{ADIN}	V_{REFL}	-	V_{REFH}	V
输入电容		C_{ADIN}	-	4.5	5.5	pF
输入电阻		R_{ADIN}	-	3	5	k Ω
外接 MCU 的模拟源电阻	10 位模式	R_{AS}	-	-	5	k Ω
	$f_{ADCK} > 4\text{MHz}$		-	-	10	
	$f_{ADCK} < 4\text{MHz}$		-	-	10	
ADC 转换时钟频率	8 位模式（所有有效 f_{ADCK} ）		-	-	10	
	高速（ADLPC=0）	f_{ADCK}	0.4		8.0	MHz
	低速（ADLPC=1）		0.4		8.0	

(1) 除非另有说明，假定 $V_{DDAD}=5\text{V}$ ，温度=25℃， $f_{ADCK}=1.0\text{MHz}$ ，典型值指示举例，没有在产品中测试。

(2) DC 位差不同。



图A-8 ADC输入电阻框图

表 A-9 5V10 位 ADC 转换特性 ($V_{REFL}=V_{DDAD}$, $V_{REFL}=V_{SSAD}$)

特性	条件	C	标志	最小	典型 ¹	最大	单位
电源电流 ADLPC = 1 ADLSMP = 1 ADCO = 1		T	I_{DDAD}	-	133	-	A
电源电流 ADLPC = 1 ADLSMP = 0 ADCO = 1		T	I_{DDAD}	-	218	-	A
电源电流 ADLPC = 1 ADLSMP = 0 ADCO = 1		T	I_{DDAD}	-	327	-	A
电源电流 ADLPC = 0 ADLSMP = 0 ADCO = 1		T	I_{DDAD}	—	582	—	μA
	$V \leq 5.5 V$	P		—	—	1	mA
ADC 异步时钟源 $t_{ADACK}=1/f_{ADACK}$	高速 (ADLPC=0)	P	f_{ADACK}	2	3.3	5	MHz
	低功耗 (ADLPC=1)			1.25	2	3.3	
转换时间（包括采样时间）	短采样 (ADLSMP=0)	P	t_{ADC}	—	20	—	ADCK 周期
	长采样 (ADLSMP=1)			—	40	—	
采样时间	短采样 (ADLSMP=0)	P	t_{ADS}	—	3.5	—	ADCK 周期
	长采样 (ADLSMP=1)			—	23.5	—	
总的未调整误差包括量化	10 位模式	P	E_{TUE}	—	± 1	± 2.5	LSB ²

	8 位模式			—	±0.5	±1.0	
微分非线性	10 位模式	P	DNL	—	±0.5	±1.0	LSB ²
	8 位模式			—	±0.3	±0.5	
	确保单调且无遗失编码						
积分非线性	10 位模式	C	INL	—	±0.5	±1.0	LSB ²
	8 位模式			—	±0.3	±0.5	
归零错误 V _{ADIN} = V _{SSA}	10 位模式	P	E _{ZS}	—	±0.5	±1.5	LSB ²
	8 位模式			—	±0.5	±0.5	
满标误差 V _{ADIN} = V _{DDA}	10 位模式	P	E _{FS}	—	±0.5	±1.5	LSB ²
	8 位模式			—	±0.5	±0.5	
量化误差	10 位模式	D	E _Q	—	—	±0.5	LSB ²
	8 位模式			—	—	±0.5	
输入泄露误差 焊盘泄露 ³ *R _{AS}	10 位模式	D	E _{IL}	—	±0.2	±2.5	LSB ²
	8 位模式			—	±0.1	±1	
温度传感器斜率	-40℃-25℃	D	m	—	3.266	—	mV/℃
	25℃– 125℃			—	3.636	—	
温度传感器电压	25℃	D	V _{TEMP25}	—	1.41	—	mV

(1) 如果没有其他说明, 假定 $V_{DDAD}=5V$, 温度=25°C, $f_{ADCK}=1MHz$ 典型值, 典型值仅供参考未在产品中测试。

(2) $1LSB = (V_{REFH} - V_{REFL}) / 2^N$.

(3) 基于输入焊盘泄漏电流, 参考焊盘电气。

A.9 内部时钟产生模块特性

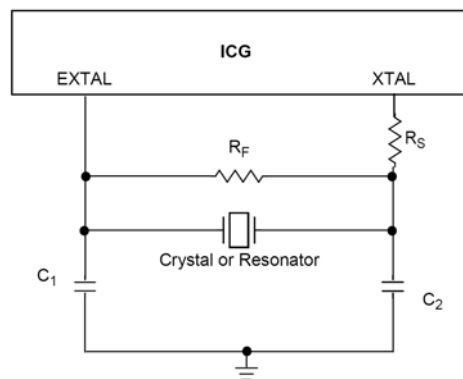


表 A-10 ICG DC 电气描述（温度范围=-40 到 125℃）

特性	标志	最小	典型 ¹	最大	单位
负载电容	C ₁ C ₂	参见下注 ²			
反馈电阻	R _F		10		MΩ
低范围（32k 到 100kHz）					
高范围(1M-16MHz)					
串联电阻	R _S		0	-	kΩ
低范围					
低增益（HGO=0）					
高增益(HGO=1)					
高范围					
低增益（HGO=0）					
高增益(HGO=1)					
≥8MHz					
4MHz					
1MHz					

(1) 典型值是在 V_{DD}=5V, 25℃条件下取得。

(2) 参见晶振或共鸣器厂商的建议。

A.9.1 ICG频率规格

表 A-11 ICG 频率规格

(V_{DDA}=V_{DDA(min)}到 V_{DDA(max)}, 温度范围=-40 到 125℃)

编号	C	特性	标志	最小	典型 ¹	最大	单位				
1		振荡器晶体或共鸣器 (REFS=1)(基谐波晶体 或陶瓷电容)	f _{lo}	32	-	100	kHz				
		低范围									
		高范围									
		高增益						f _{hi_byp}	1	16	MHz
		高增益						f _{hi_eng}	2	10	MHz
		低功耗						f _{lp_byp}	1	8	MHz
		低功耗						f _{lp_eng}	2	8	MHz
2		输入时钟频率 (CLKS=11,REFS=0)	f _{lo}	32	-	100	kHz				
		低范围									
		高范围						f _{hi_eng}	2	10	MHz
3		输入时钟频率 (CLKS=10,REFS=0)	f _{Extal}	0	-	40	MHz				
4		内部参考频率（未对齐）	f _{ICGIRCLK}	182.25	243	303.75	kHz				
5		输入时钟的占空因子 (REFS=0)	tdc	40	-	60	%				

6	输出时钟 ICGOUT 频率 CLKS=10,REFS=0 其他情况	f_{ICGOUT}	$f_{\text{Extal}} (\text{min})$ $f_{\text{lo}} (\text{min})$	- -	$f_{\text{Extal}} (\text{max})$ $f_{\text{ICGDCLKmax}} (\text{max})$	MHz
7	最小 DCO 时钟 (ICGDCLK) 频率	$f_{\text{ICGDCLKmin}}$	8	-		MHz
8	最小 DCO 时钟 (ICGDCLK) 频率	$f_{\text{ICGDCLKmax}}$		-	40	MHz
9	自时钟模式 (ICGOUT) 频率 2	f_{Self}	$f_{\text{ICGDCLKmin}}$		$f_{\text{ICGDCLKmax}}$	MHz
10	自时钟模式复位 (ICGOUT) 频率	$f_{\text{Self_reset}}$	5.5	8	10.5	MHz
11	丢失参考频率 3 低范围 高范围	f_{LOR}	5 50		25 500	kHz
12	丢失 DCO 频率 4	f_{LOD}	0.5		1.5	MHz
13	晶振开始时序 5,6 低范围 高范围	t_{CSTL} t_{CSTH}	— —	430 4	— —	ms
14	FLL 锁时序 低范围 高范围	t t_{Lockh}	— —		2 2	ms
15	FLL 频率解锁范围	n_{Unlock}	$-4*N$		$4*N$	counts
16	FLL 频率上锁范围	n_{Lock}	$-2*N$		$2*N$	counts
17	ICGOUT 周期抖动。8 以 f_{ICGOUT} 测量最大抖动 (平均 2ms 间隔)	C_{Jitter}	—		0.2	% f_{ICG}
18	MC9S08ACxx: 内部振荡 器背离对齐的频率9 $V_{\text{DD}} = 2.7 - 5.5 \text{ V}$, (恒 温) $V_{\text{DD}} = 5.0 \text{ V} \pm 10\%$, -40°C to 125°C	ACC_{int}	— —	± 0.5 ± 0.5	± 2 ± 2	%
	S9S08AWxxA: 内部振 荡器背离对齐的频率10 $V_{\text{DD}} = 2.7 - 5.5 \text{ V}$, (恒 温) $V_{\text{DD}} = 5.0 \text{ V} \pm 10\%$, -40°C to 125°C	ACC_{int}	— —	± 0.5 ± 0.5	± 1.5 ± 1.5	%

- (1) 除非另有说明，典型值是在 $V_{DD}=5V$ ， $25^{\circ}C$ 取得。
- (2) 自时钟模式频率是当 FLL 开环时，DCO 产生的频率。
- (3) 丢失参考频率是参考频率在内部被检测，如果不在要求的范围，将使 ICG 进入自时钟模式。
- (4) 丢失 DCO 频率是 DCO 频率在内部被检测，如果不在要求的范围，将使 ICG 进入 FLL 旁路外部模式（如果外部参考频率存在）。
- (5) 该参数未 100% 的测试就被描述。
- (6) 适当的 PC 布板程序可能会遵循到达规范。
- (7) 进入到 FLL 忙碌的内部或外部模式后，该规则应用到对于 FLL 需要锁的时序
- (8) 抖动背离了编程的频率，经测量超过了规范的内部最大 f_{ICGOUT} 。测量是在有过滤供电和稳定外部时钟信号的设备上进行。噪音经 $VDDAVSSA$ 进入 FLL 电路，并且晶体振荡器频率的变化增加了 C_{jitter} 的百分比（对于给定的间隔）。
- (9) 参见图 A-9。
- (10) 参见图 A-9。

内部振荡器背离对齐的频率



设备在 $25^{\circ}C$ ，3V 对齐。

图A-9 内部振荡器背离对齐的频率

A.10 AC特性

该节描述了每个外设系统的交流时序特性。关于总线如何产生时钟的更多信息，参见第八章“内部时钟产生器（S08ICGV4）”。

A.10.1 控制时序

表 A-7 电源电流特性

编号	C	参数	标志	最小	典型 ¹	最大	单位
1		总线频率($t_{cyc}=1/f_{Bus}$)	f_{Bus}	dc	-	20	MHz
2		实时中断内部振荡器周期	t_{RTI}	700		1300	μs

3		外部复位脉宽 ² ($t_{\text{cyc}} = 1/f_{\text{Self_reset}}$)	t_{extrst}	$1.5 \times t_{\text{Self_reset}}$	-	ns
4		复位低驱动 ³	t_{rstdrv}	$34 \times t_{\text{cyc}}$	-	ns
5		有效北京调试模式闭锁准备时间	t_{MSSU}	25	-	ns
6		有效北京调试模式闭锁保持时间	t_{MSH}	25	-	ns
7		IRQ 脉宽 异步路径 ² 同步路径 ⁴	$t_{\text{ILIH}},$ t_{IHIL}	100 $1.5 \times t_{\text{cyc}}$	- -	ns
8		KBIPx 脉宽 异步路径 ² 同步路径 ³	$t_{\text{ILIH}},$ t_{IHIL}	100 $1.5 \times t_{\text{cyc}}$	- -	ns
9		端口上升和下降时间 (load = 50 pF) ⁵ 禁止转换速率控制 (PTxSE=0) 允许转换速率控制 (PTxSE=1)	$t_{\text{Rise}},$ t_{Fall}	— —	3 30	ns

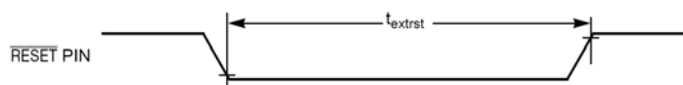
(1) 除非另有说明，典型值是在 $V_{\text{DD}}=5\text{V}$ ， 25°C 取得。

(2) 这是最短脉冲，可以确保作为复位引脚请求。更短的脉冲不能确保从内部时钟产生复位请求。

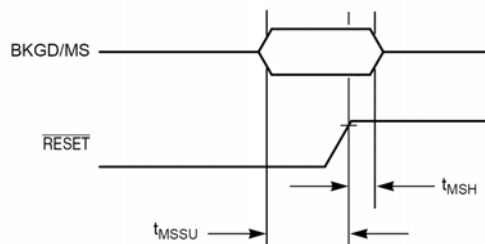
(3) 当任何复位初始化，内部驱动复位引脚低 34 个总线周期，然后在复位引脚采样 38 个周期来从内部复位请求区分外部复位请求。

(4) 这是最小的脉冲宽度，可以确保通过引脚同步电路。更短的脉冲可能或可能不会被认出。在 stop 模式，同步器被设旁路，如此以至于在那种情况下，更短的脉冲可以被认出。

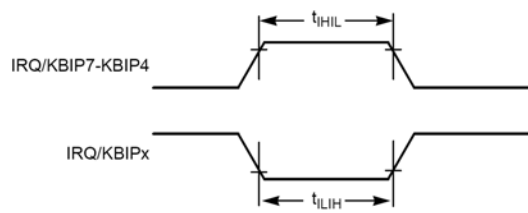
(5) 所有的时序关系 $V_{\text{DD}}20\%$ 和 70% ；温度范围 -40°C to 125°C 。



图A-10 复位时序



图A-11 有效的背景调试模式的闭锁时序



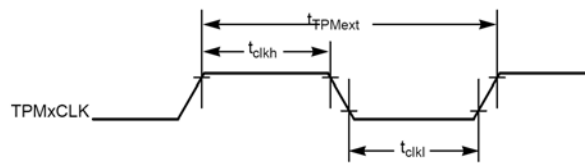
图A-12 IRQ/KBIPx 时序

A.10.2 定时器/PWM(TPM)模块时序

同步器电路决定了被公认为最短输入脉冲或可以用作定时器计数器的外部源的最快的时钟。这可以同当前总线时钟同步。

表 A-10 ICG DC 电气描述（温度范围=-40 到 125℃）

特性	标志	最小	最大	单位
外部时钟频率	f_{TPMext}	dc	$f_{Bus}/4$	MHz
外部时钟周期	t_{TPMext}	4	—	t_{cyc}
外部时钟高时间	t_{clkh}	1.5	—	t_{cyc}
外部时钟低时间	t_{clkl}	1.5	—	t_{cyc}
输入捕捉脉宽	t_{ICPW}	1.5	—	t_{cyc}



图A-13 定时器外部时钟

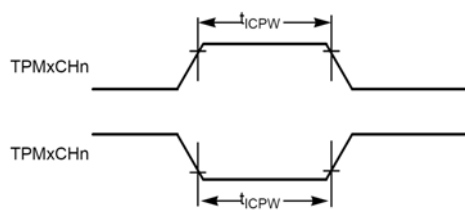


图 A-14 =3V 时, IOH(低驱动)和 VDD-VOH 的典型值

A.11 SPI特性

表 A-14 和图 A-15 到图 A-18 描述 SPI 系统的时序要求。

表 A-14 SPI 电气特性

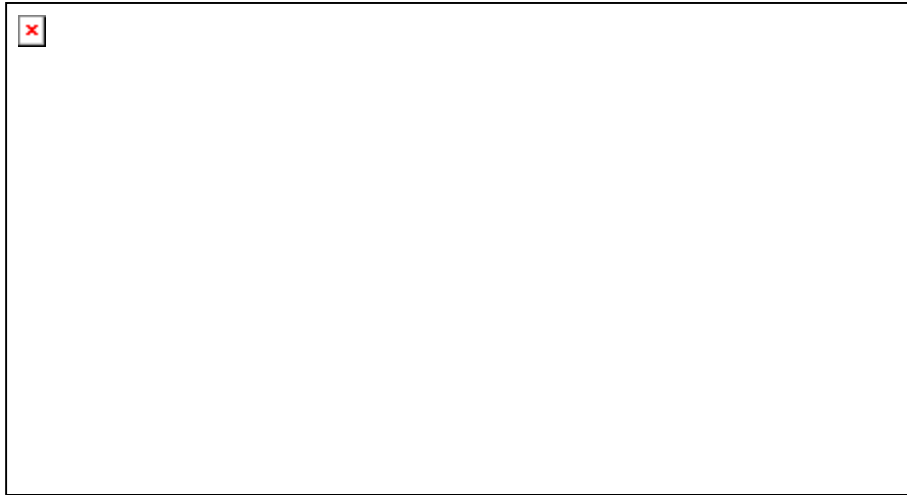
编号	C	特性 ¹	标志	最小	最大	单位
1		操作频率				
		主机	f_{op}	$f_{Bus}/2048$		Hz
		从机	f_{op}	dc		

2		周期时间	主机	t_{SCK}	2	2048	t_{cyc}
			从机	t_{SCK}	4	-	t_{cyc}
3		允许持续时间		t_{Lead}	-	1/2	t_{SCK}
				t_{LLead}	1/2		t_{SCK}
4		允许滞后时间		t_{Lead}	-	1/2	t_{SCK}
				t_{LLead}	1/2		t_{SCK}
5		时钟（SPSCK）高时间 主机和从机		t_{SCKH}	1/2 $t_{SCK}-25$	-	ns
6		时钟（SPSCK）低时间 主机和从机		t_{SCKL}	1/2 $t_{SCK}-25$	-	ns
7		数据准备时间（输入） 主机 从机		$t_{SI}(M)$	30	-	ns
				$t_{SI}(S)$	30	-	ns
8		数据保持时间（输入） 主机 从机		$t_{HI}(M)$	30	-	ns
				$t_{HI}(S)$	30	-	ns
9		访问时间，从机 ²		t_A	0	40	ns
10		禁止时间，从机 ³		t_{dis}	-	40	ns
11		数据准备时间（输出） 主机 从机		t_{SO}	25	-	ns
				t_{SO}	25	-	ns
12		数据保持时间（输出） 主机 从机		t_{HO}	-10	-	ns
				t_{HO}	-10	-	ns

(1) 除非说明，所有的时序关系 $V_{DD}20\%$ 和 70% ；100pF 在所有的 SPI 引脚上。所有的时序假定输出引脚禁止转换速率控制和允许高驱动强度。

(2) 数据有效的时序来自高阻抗状态。

(3) 保持时间，高阻抗状态



注：1 SS 输出模式（MODFEN=1、SSOE=1）。

2 LSBF=0，对于 LSBF=1 位的顺序是 LSB，位 1，...，位 6，MSB。

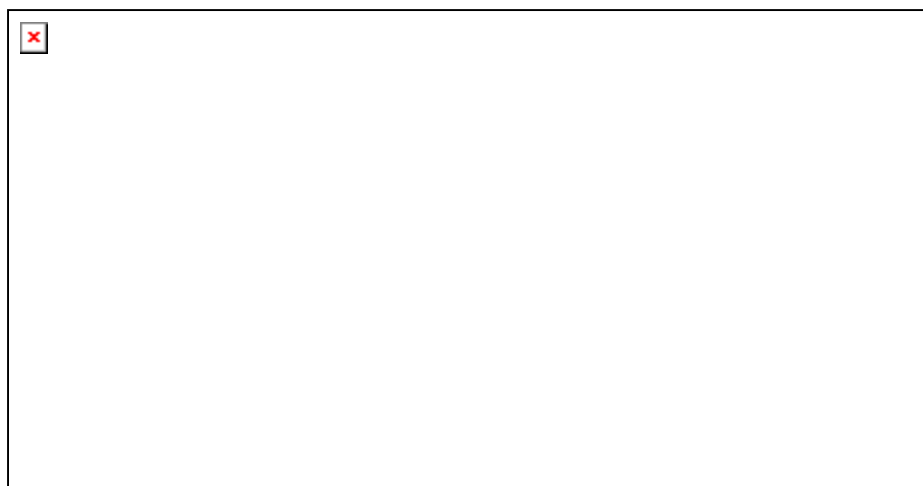
图 A-15 SPI 主机时序（CPHA=0）



注：1 SS 输出模式（MODFEN=1、SSOE=1）。

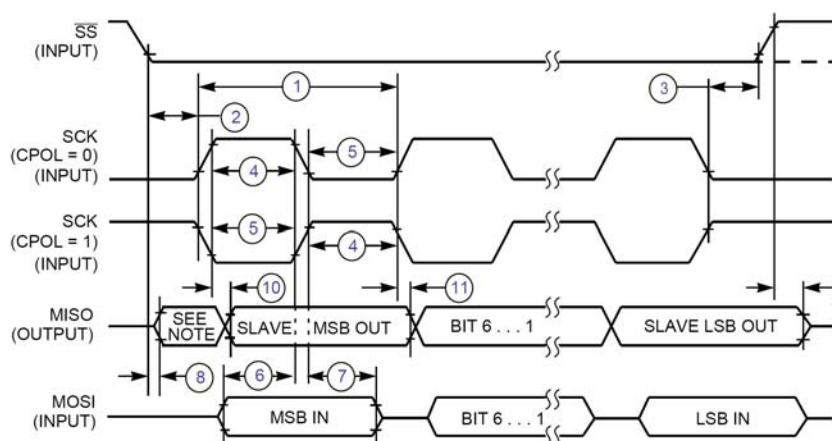
2 LSBF=0，对于 LSBF=1 位的顺序是 LSB，位 1，...，位 6，MSB。

图 A-16 SPI 主机时序（CPHA=1）



注：字符的 MSB 没有定义，仅被接收。

图A-17 SPI从机时序（CPHA=0）



注：字符的 LSB 没有定义但是仅仅被接收。

图 A-18 SPI 从机时序（CPHA=1）

A.12 FLASH规格

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。更多关于擦除和编程操作的信息，参见第四章“内存”。

表 A-15 FLASH 特性

编号	C	特性	标志	最小	典型 ¹	最大	单位
1		写入/擦除电压	$V_{\text{prog/erase}}$	2.7		5.5	V
2		读操作电压	V_{read}	2.7		5.5	V
3		内部 FCLK 频率 ²	V_{FCLK}	150		200	KHz
4		内部 FCLK 周期 (1/FCLK)	t_{Fcyc}	5		6.67	μs

5		自己写入时间（随机地址） ⁽²⁾	t_{prog}	9			t_{FcyC}
6		自己写入时间（突发模式） ⁽²⁾	t_{Bust}	4			t_{FcyC}
7		页擦除时间 ³	t_{Page}	4000			t_{FcyC}
8		整体擦除时间 ⁽²⁾	t_{Mass}	20000			t_{FcyC}
9	C	写入/擦除强度 ⁴ TL到TH=-40℃ 到+125℃ T=25		10000 -	- 10000	- -	周期
10		数据保持 ⁵	$t_{\text{D-ret}}$	15	100	-	年

(1) 除非有其他说明，典型值 VDD=5V，25℃取得。

(2) 时钟的频率通过软件设置来控制。

(3) 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。

(4) 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”。

(5) 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”。

A.13 EMC性能

电磁适应性（EMC）性能高度依赖于 MCU 所处的环境。板的设计和布板，电路布局选择，外部元件的放置和特性，MCU 软件操作都对 EMC 的性能有很大影响。系统设计者应该参考 Freescale 应用笔记 AN2321、AN1050、AN1263、AN2764 和 AN1259 的建议和指导来优化 EMC 性能。

附录B 订购须知和机械制图

B.1 订购须知

该节包含了 MC9S08AC16 系列设备的订购须知。参见下面一个设备标号的例子。

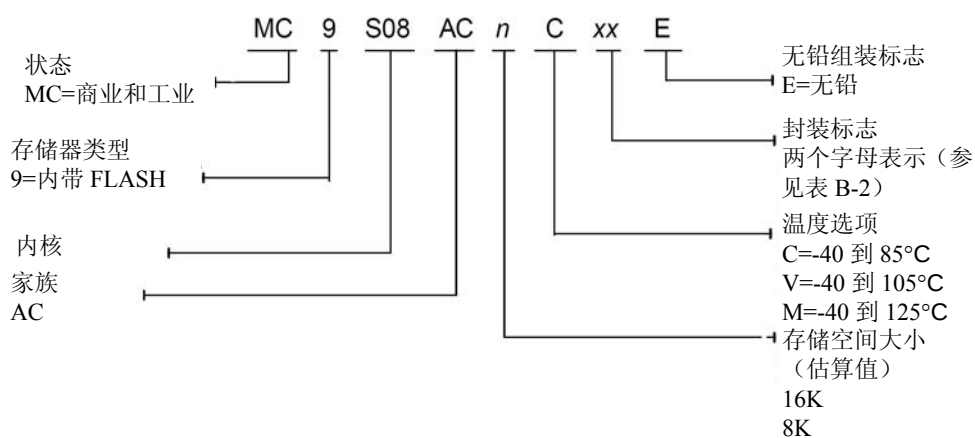
表 B-1 设备比较

设备编号 ¹	存储器		可用封装 ²	用途
	FLASH	RAM	类型	类型
MC9S08AC16	16,384	1024	48 引脚 QFN	商业
MC9S08AC8	8192	768	44 引脚 LQFP	工业
			32 引脚 LQFP	
MC9S08AW16A	16,384	1024	48 引脚 QFN	汽车
MC9S08AW16A	8192	768	44 引脚 LQFP	
			32 引脚 LQFP	

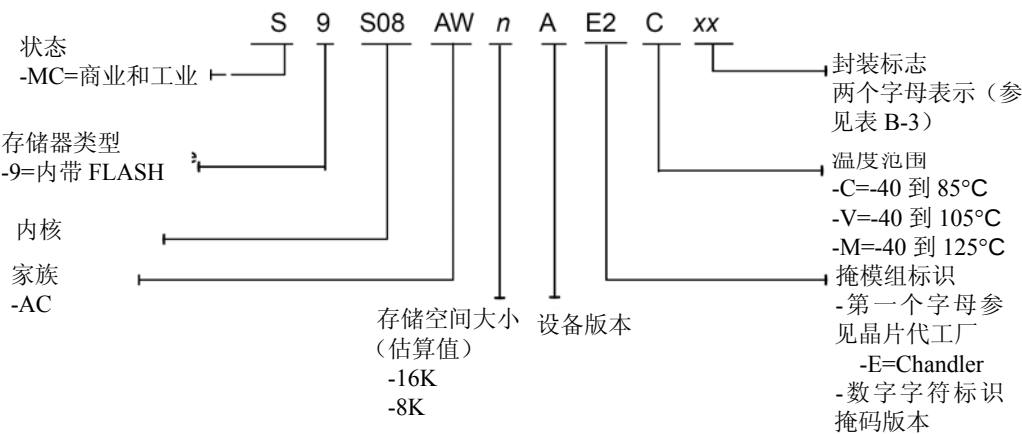
(1) 参见表 1-1 有每个设备所包含模块的详细描述。

(2) 封装信息参见表 B-2 和表 B-3。

按照该设备家族，MC9S08AC16 系列设备的订购零件编号也是多样的。参见图 B-1 和图 B-2 的例子。



图B-1 商业和工业设备编号系统



图B-1 商业和工业设备编号系统

B.2 机械制图

接下来的内容包含了 MC9S08AC16 系列封装选项的机械说明。参见接下来的表中文档号与每个封装类型相对应。

表 B-2 MC9S08AC16 和 MC9S08AC8 商业和工业封装信息

引脚数	类型	标注	文档号
48	QFN	FD	98ARH99048A
44	LQFP	FG	98ASS23225W
32	LQFP	FJ	98ASH70029A

表 B-3 MC9S08AW16A 和 MC9S08AW8A 汽车封装信息

控制位		时钟源	COP 溢出计时
COPCLKS	COPT		
48	QFN	FT	98ARH99048A
44	LQFP	LS	98ASS23225W
32	LQFP	LM	98ASH70029A



Freescall 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: F

热增强型四方扁平贴片封装(QFN)

案例号: 1314-05

2005-12-5

48 引脚, 间距 0.5 (7×7×1)

标准: JEDEC MO-220 VKKD-2



Freescall 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: F

热增强型四方扁平贴片封装(QFN)

案例号: 1314-05

2005-12-5

48 引脚, 间距 0.5 (7×7×1)

标准: JEDEC MO-220 VKKD-2



Freescall 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: F

热增强型四方扁平贴片封装(QFN)

案例号: 1314-05

2005-12-5

48 引脚, 间距 0.5 (7×7×1)

标准: JEDEC MO-220 VKKD-2



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

热增强型四方扁平贴片封装(QFN)

48 引脚, 间距 0.5 (7×7×1)

文档号: 98ARH99048A

案例号: 1314-05

标准: JEDEC MO-220 VKKD-2

修订版本: F

2005-12-5



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ASS23225W

修订版本: D

44引脚 (LQFP)

案例号: 824D-02

2007-2-26

10×10×1.4 封装, 间距 0.5, 厚度 1.4

标准: JEDEC MS-026 -BCB



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ASS23225W

修订版本: D

44引脚 (LQFP)

案例号: 824D-02

2007-2-26

10×10×1.4 封装, 间距 0.5, 厚度 1.4

标准: JEDEC MS-026 -BCB



Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ASS23225W

修订版本: D

44引脚 (LQFP)

案例号: 824D-02

2007-2-26

10×10×1.4 封装, 间距 0.5, 厚度 1.4

标准: JEDEC MS-026 -BCB



Freescale 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ASH70029A

修订版本: D

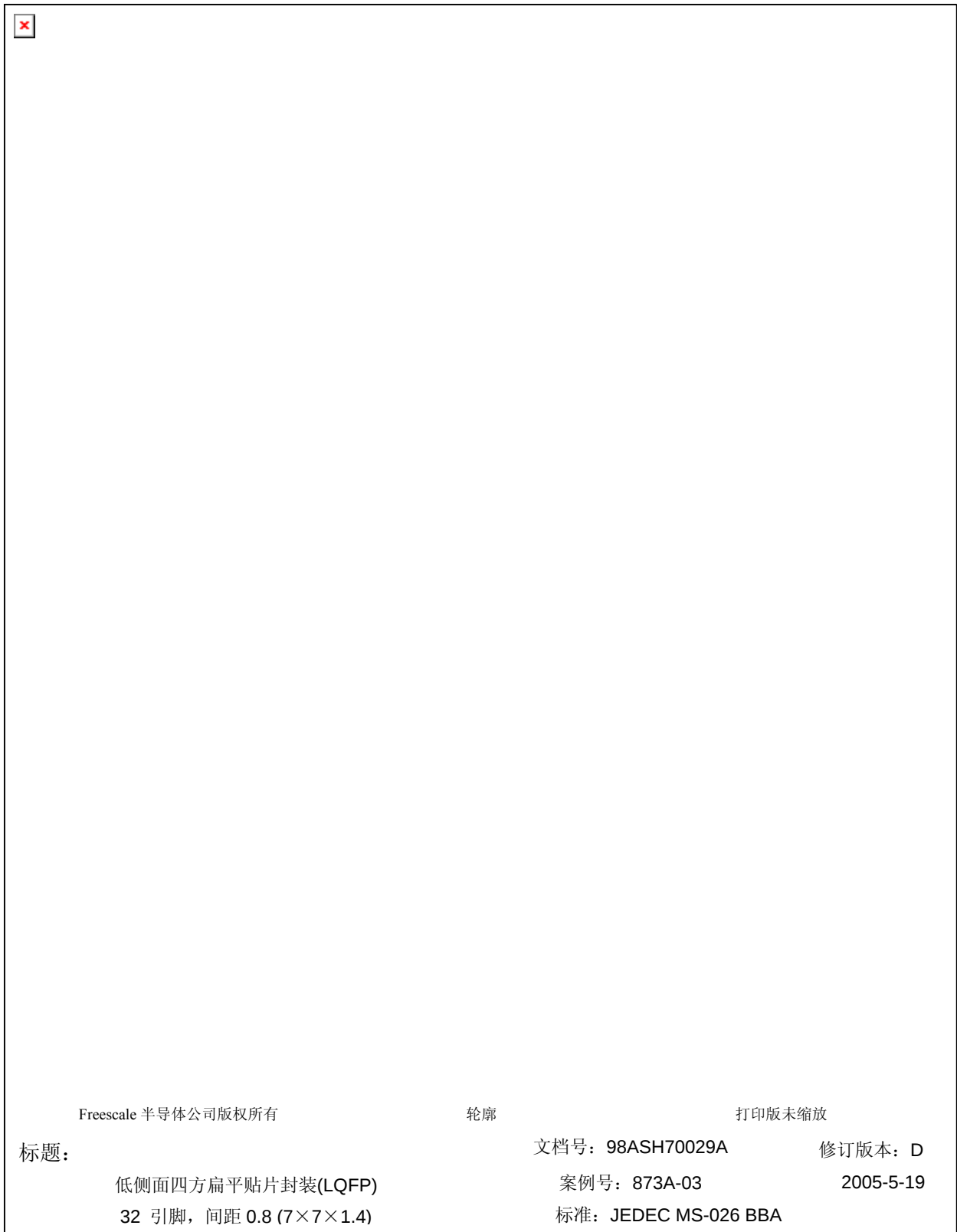
低侧面四方扁平贴片封装(LQFP)

案例号: 873A-03

2005-5-19

32 引脚, 间距 0.8 (7×7×1.4)

标准: JEDEC MS-026 BBA





Freescall 半导体公司版权所有

轮廓

打印版未缩放

标题:

文档号: 98ASH70029A

修订版本: D

低侧面四方扁平贴片封装(LQFP)

案例号: 873A-03

2005-5-19

32 引脚, 间距 0.8 (7×7×1.4)

标准: JEDEC MS-026 BBA
