

SC667027

数据手册：技术数据

HCS08
微控制器

SC667027ZHS

第 1 版

2008 年 9 月

飞思卡尔半导体

SC667027 系列产品的特性

8 位 HCS08 中央处理单元 (CPU)

- 20 MHz HCS08CPU(中央处理单元)
- 10 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再增加两个断点)
- 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 16 KB 的片上在线可编程 flash 存储器, 有块保护和选项
- 高达 1 KB 的片上 RAM

时钟源选项

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

系统保护

- 可选的看门狗 (COP) 复位, 使用运行于独立的内部时钟源或总线时钟
- 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

- 两种停止模式

外围设备

- **ADC**-8 通道, 8 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 位中断字符通讯模式
- **定时器**-3 个 16 位定时器 /PWM (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边沿对齐。每一个定时器 /PWM (TPM) 可以在其所有通道产生中心对齐的 PWM (CPWM)。
- **KBI**-7 引脚键盘中断模块

输入 / 输出

- 高达 34 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输出斜率控制

- 输出时, 每个端口都有软件选择的驱动强度
- 主复位引脚和上电复位 (POR)
- RESET, IRQ, BKGD/MS 引脚在内部上拉以减小客户系统成本。

封装选择

- 44 引脚 LQFP
- 32 引脚 LQFP

正常工作环境

- 电压: 2.7 V 到 5.5 V
- 温度: -40 °C 到 85 °C

SC667027 数据手册

SC667027ZHS

第 1 版

2008 年 9 月



版本记录

我们在互联网上提供手册的最新版本，而您的纸件版本可能是较早版本。请登录以下网站来确认您拥有最新的产品信息：

<http://freescale.com/>

以下为本手册的修订记录：

版本号	出版日期	更新描述
1	9/8/2008	第一次正式发布

This product incorporates SuperFlash[®] technology licensed from SST.

Freescale[™] and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
© Freescale Semiconductor, Inc., 2008. All rights reserved.

章节号	标题	页
第 1 章	绪论	1
第 2 章	引脚及其连接	5
第 3 章	工作模式	11
第 4 章	片上存储器	17
第 5 章	复位、中断和系统配置	37
第 6 章	并行输入和输出	53
第 7 章	中央处理单元 (S08CPUV2)	75
第 8 章	内部时钟发生器 (S08ICGV4)	93
第 9 章	定时器 /PWM (S08TPMV3)	117
第 10 章	串行通信接口 (S08SCIV4)	139
第 11 章	AD 转换器 (S08ADC10V1)	157
第 12 章	开发支持	181
附录 A	电气特性和时序规范	199
附录 B	订购信息和机械图	215



手册目录

章节号	标题	页码
第 1 章		
绪论		
1.1	概述	1
1.2	MCU 结构框图	1
1.3	系统时钟分配	3
第 2 章		
引脚及其连接		
2.1	简介	5
2.2	MCU 的引脚分配	5
2.3	推荐的系统连接	7
2.3.1	电源 (V_{DD} 、 $2 \times V_{SS}$ 、 V_{DDAD} 和 V_{SSAD})	9
2.3.2	振荡器 (XTAL 与 EXTAL)	9
2.3.3	复位	9
2.3.4	背景 / 模式选择 (BKGD/MS)	9
2.3.5	ADC 参考引脚 (V_{REFH} 与 V_{REFL})	10
2.3.6	外部的中断引脚 (IRQ)	10
2.3.7	通用 I/O 及外设端口	10
第 3 章		
工作模式		
3.1	简介	11
3.2	特性	11
3.3	运行模式	11
3.4	背景调试模式	11
3.5	停止模式	12
3.5.1	Stop2 模式	12
3.5.2	Stop3 模式	13
3.5.3	停止模式下激活 BDM 使能	13
3.5.4	停止模式下 LVD 使能	14
3.5.5	停止模式下的片上外设模块	14
第 4 章		
片上存储器		
4.1	SC667027 系列存储映像简介	17
4.1.1	复位和中断向量分配	18
4.2	寄存器地址和位分配	19
4.3	RAM	24
4.4	Flash	25

4.4.1	特性	25
4.4.2	写入和擦除时间	25
4.4.3	写入和擦除命令的执行	26
4.4.4	突发模式写入的执行	27
4.4.5	访问错误	28
4.4.6	Flash 块保护	29
4.4.7	向量重定向	29
4.5	保护机制	30
4.6	Flash 寄存器和控制位	31
4.6.1	Flash 时钟分频寄存器 (FCDIV)	31
4.6.2	Flash 选项寄存器 (FOPT 和 NVOPT)	32
4.6.3	Flash 配置寄存器 (FCNFG)	33
4.6.4	Flash 保护寄存器 (FPROT 和 NVPROT)	33
4.6.5	Flash 状态寄存器 (FSTAT)	33
4.6.6	Flash 命令寄存器 (FCMD)	34

第 5 章 复位、中断和系统配置

5.1	简介	37
5.2	特性	37
5.3	MCU 复位	37
5.4	计算机正常操作监控模块 (COP) 看门狗	38
5.5	中断	39
5.5.1	中断堆栈结构	39
5.5.2	外部中断请求 (IRQ) 引脚	40
5.5.3	中断向量、中断源和局部屏蔽	41
5.6	低电压检测系统 (LVD)	43
5.6.1	上电复位操作	43
5.6.2	LVD 复位操作	43
5.6.3	LVD 中断操作	43
5.6.4	低电压警告 (LVW)	43
5.7	实时中断 (RTI)	43
5.8	MCLK 输出	44
5.9	复位、中断以及系统控制寄存器和控制位	44
5.9.1	中断请求状态和控制寄存器 (IRQSC)	44
5.9.2	系统复位状态寄存器 (SRS)	45
5.9.3	系统背景调试强制复位寄存器 (SBDFR)	46
5.9.4	系统选项寄存器 (SOPT)	46
5.9.5	系统 MCLK 控制寄存器 (SMCLK)	47
5.9.6	系统设备识别寄存器 (SDIDH、SDIDL)	48
5.9.7	系统实时中断状态和控制寄存器 (SRTISC)	48
5.9.8	系统电源管理状态控制寄存器 1 (SPMSC1)	49
5.9.9	系统电源管理状态和控制寄存器 2 (SPMSC2)	50
5.9.10	系统系统选项寄存器 2 (STOP2)	51

第 6 章 并行输入和输出

6.1	简介	53
6.2	特性	55
6.3	引脚描述	55
6.3.1	A 口	55
6.3.2	B 口	55
6.3.3	C 口	56
6.3.4	D 口	56
6.3.5	E 口	57
6.3.6	F 口	57
6.3.7	G 口	57
6.4	并行 I/O 控制	58
6.5	引脚控制	59
6.5.1	内部上拉电阻使能	59
6.5.2	输出斜率控制使能	59
6.5.3	输出驱动强度选择	59
6.6	停止模式下的 I/O 口	59
6.7	并行 I/O 与引脚控制寄存器	59
6.7.1	A 口 I/O 寄存器 (PTAD 和 PTADD)	59
6.7.2	A 口引脚控制寄存器 (PTAPE、PTASE、PTADS)	60
6.7.3	B 口 I/O 寄存器 (PTBD 和 PTBDD)	61
6.7.4	B 口引脚控制寄存器 (PTBPE、PTBSE、PTBDS)	62
6.7.5	C 口 I/O 寄存器 (PTCD 和 PTCDD)	63
6.7.6	C 口引脚控制寄存器 (PTCPE、PTCSE、PTCDS)	64
6.7.7	D 口 I/O 寄存器 (PTDD 和 PTDDD)	65
6.7.8	D 口引脚控制寄存器 (PTDPE、PTDSE、PTDDS)	66
6.7.9	E 口 I/O 寄存器 (PTED 和 PTEDD)	67
6.7.10	E 口引脚控制寄存器 (PTEPE、PTESE、PTEDS)	68
6.7.11	F 口 I/O 寄存器 (PTFD 和 PTFDD)	69
6.7.12	F 口引脚控制寄存器 (PTFPE、PTFSE、PTFDS)	70
6.7.13	G 口 I/O 寄存器 (PTGD 和 PTGDD)	71
6.7.14	G 口引脚控制寄存器 (PTGPE、PTGSE、PTGDS)	72

第 7 章 中央处理单元 (S08CPUV2)

7.1	概述	75
7.1.1	特性	75
7.2	编程模型和 CPU 寄存器	75
7.2.1	累加器 (A)	76
7.2.2	变址寄存器 (H:X)	76
7.2.3	堆栈指针 (SP)	76
7.2.4	程序计数器 (PC)	77
7.2.5	条件码寄存器 (CCR)	77
7.3	寻址模式	78

7.3.1	固有寻址模式 (INH)	78
7.3.2	相对寻址模式 (REL)	78
7.3.3	立即寻址模式 (IMM)	78
7.3.4	直接寻址模式 (DIR)	78
7.3.5	扩展寻址模式 (EXT)	78
7.3.6	变址寻址模式	78
7.4	特殊操作	79
7.4.1	复位序列	79
7.4.2	中断序列	79
7.4.3	等待模式	80
7.4.4	停止模式	80
7.4.5	背景模式	80
7.5	HCS08 指令设置摘要	81

第 8 章 内部时钟发生器 (S08ICGV4)

8.1	概述	95
8.1.1	特性	95
8.1.2	操作模式	95
8.1.3	功能结构图	96
8.2	外部信号描述	97
8.2.1	EXTAL—外部参考时钟 / 振荡器输入	97
8.2.2	XTAL—振荡器输出	97
8.2.3	外部时钟连接	97
8.2.4	外部晶振 / 谐振器连接	97
8.3	寄存器定义	98
8.3.1	ICG 控制寄存器 1 (ICGC1)	98
8.3.2	ICG 控制寄存器 2 (ICGC2)	99
8.3.3	ICG 状态寄存器 1 (ICGS1)	100
8.3.4	ICG 状态寄存器 2 (ICGS2)	101
8.3.5	ICG 滤波器寄存器 (ICGFLTU、ICGFLTL)	101
8.3.6	ICG 调整寄存器 (ICGTRM)	102
8.4	功能描述	102
8.4.1	OFF 模式	103
8.4.2	自时钟模式 (SCM)	103
8.4.3	FLL 内部时钟模式 (FEI)	104
8.4.4	FLL 内部未锁定	104
8.4.5	FLL 内部锁定	105
8.4.6	FLL 旁路外部时钟模式 (FBE)	105
8.4.7	FLL 外部时钟模式 (FEE)	105
8.4.8	FLL 锁定和失锁检测	105
8.4.9	FLL 时钟丢失检测	106
8.4.10	时钟模式必要条件	107
8.4.11	固定频率时钟	108
8.4.12	高增益振荡器	108

8.5	初始化 / 应用信息	108
8.5.1	概述	108
8.5.2	例 1: 外部晶振 = 32 kHz, 总线频率 = 4.19 MHz	110
8.5.3	例 2: 外部晶振 = 4 MHz, 总线频率 = 20 MHz	111
8.5.4	例 3: 无外部晶振, 总线频率 = 5.4 MHz	113
8.5.5	例 4: 内部时钟产生器调整	114

第 9 章 定时器 / PWM (S08TPMV3)

9.1	概述	117
9.2	TPMV3 与以前版本的区别	119
9.2.1	从 TPMV1 移植	120
9.3	特性	121
9.3.1	操作模式	122
9.3.2	结构框图	122
9.4	信号描述	124
9.4.1	信号详细描述	124
9.5	寄存器定义	126
9.5.1	TPM 状态和控制寄存器 (TPMxSC)	126
9.5.2	TPM 计数器寄存器 (TPMxCNTH:TPMxCNHTL)	128
9.5.3	TPM 计数器模寄存器 (TPMxMODH:TPMxMODL)	128
9.5.4	TPM 通道 n 状态和控制寄存器 (TPMxCnSC)	129
9.5.5	TPM 通道值寄存器 (TPMxCnVH : TPMxCnVL)	131
9.6	功能描述	132
9.6.1	计数器	132
9.6.2	通道模式选择	133
9.7	复位概述	136
9.7.1	概要	136
9.7.2	复位操作	136
9.8	中断	136
9.8.1	概述	136
9.8.2	中断操作描述	136

第 10 章 串行通信接口 (S08SCIV4)

10.1	概述	139
10.1.1	特性	141
10.1.2	操作模式	141
10.1.3	框图	141
10.2	寄存器定义	144
10.2.1	SCI 波特率寄存器 (SCIxBDH、SCIxBDL)	144
10.2.2	SCI 控制寄存器 1 (SCIxC1)	145
10.2.3	SCI 控制寄存器 2 (SCIxC2)	146
10.2.4	SCI 状态寄存器 1 (SCIxS1)	147
10.2.5	SCI 状态寄存器 2 (SCIxS2)	148

10.2.6 SCI 控制寄存器 3 (SCIxC3)	149
10.2.7 SCI 数据寄存器 (SCIxD)	150
10.3 功能描述	150
10.3.1 波特率产生	151
10.3.2 发送功能描述	151
10.3.3 接收功能描述	152
10.3.4 中断和状态标志	153
10.3.5 其他 SCI 功能	154

第 11 章 AD 转换器 (S08ADC10V1)

11.1 概述	157
11.2 通道分配	159
11.2.1 交替时钟	159
11.2.2 硬件触发器	160
11.2.3 温度传感器	160
11.2.4 特性	161
11.2.5 框图	161
11.3 外部信号描述	162
11.3.1 模拟电源 (V_{DDAD})	162
11.3.2 模拟地 (V_{SSAD})	163
11.3.3 参考高电压 (V_{REFH})	163
11.3.4 参考低电压 (V_{REFL})	163
11.3.5 模拟通道输入 (ADx)	163
11.4 寄存器定义	163
11.4.1 状态和控制寄存器 1 (ADC1SC1)	163
11.4.2 状态和控制寄存器 2 (ADC1SC2)	165
11.4.3 数据高结果寄存器 (ADC1RH)	166
11.4.4 数据低结果寄存器 (ADC1RL)	166
11.4.5 比较值高寄存器 (ADC1CVH)	167
11.4.6 比较值低寄存器 (ADC1CVL)	167
11.4.7 配置寄存器 (ADC1CFG)	167
11.4.8 引脚控制 1 寄存器 (APCTL1)	169
11.4.9 引脚控制 2 寄存器 (APCTL2)	169
11.4.10 引脚控制 3 寄存器 (APCTL3)	170
11.5 功能描述	171
11.5.1 时钟选择和分频控制	171
11.5.2 输入选择和引脚控制	172
11.5.3 硬件触发	172
11.5.4 转换控制	172
11.5.5 自动比较功能	174
11.5.6 MCU 等待模式操作	174
11.5.7 MCU stop3 模式操作	174
11.5.8 MCU stop1 和 stop2 模式操作	175
11.6 初始化信息	175
11.6.1 ADC 模块初始化举例	175

11.7 应用信息	177
11.7.1 外部引脚和安排	177
11.7.2 错误源	178

第 12 章开发支持

12.1 介绍	181
12.1.1 强制激活背景调试	181
12.1.2 特性	182
12.2 背景调试控制器 (BDC)	182
12.2.1 BKGD 引脚描述	183
12.2.2 通信详细介绍	183
12.2.3 BDC 命令	185
12.2.4 BDC 硬件断点	188
12.3 片上调试系统 (DBG)	189
12.3.1 比较器 A 和 B	189
12.3.2 总线捕获信息和 FIFO 操作	189
12.3.3 流变化信息	190
12.3.4 标记 vs. 强制断点和触发器	190
12.3.5 触发模式	190
12.3.6 硬件断点	191
12.4 寄存器定义	192
12.4.1 BDC 寄存器和控制位	192
12.4.2 系统背景调试强制复位寄存器 (SBDFR)	194
12.4.3 DBG 寄存器和控制位	194

附录 A 电气特性和时序规范

A.1 概述	199
A.2 参数分类	199
A.3 最大绝对额定值	199
A.4 热特性	200
A.5 DC 特性	201
A.6 电源电流特性	203
A.7 ADC 特性	206
A.8 内部时钟产生模块特性	208
A.8.1 ICG 频率规格	209
A.9 AC 特性	211
A.9.1 控制时序	211
A.9.2 定时器 /PWM(TPM) 模块时序	212
A.10 Flash 规格	213
A.11 EMC 性能	214

附录 B 订购信息和机械图

B.1 订购信息	215
----------------	-----

B.2 机械图.....	215
--------------	-----

第 1 章 绪论

1.1 概述

SC667027 系列属于 8 位微控制单元（MCU），是高性能低成本的 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。

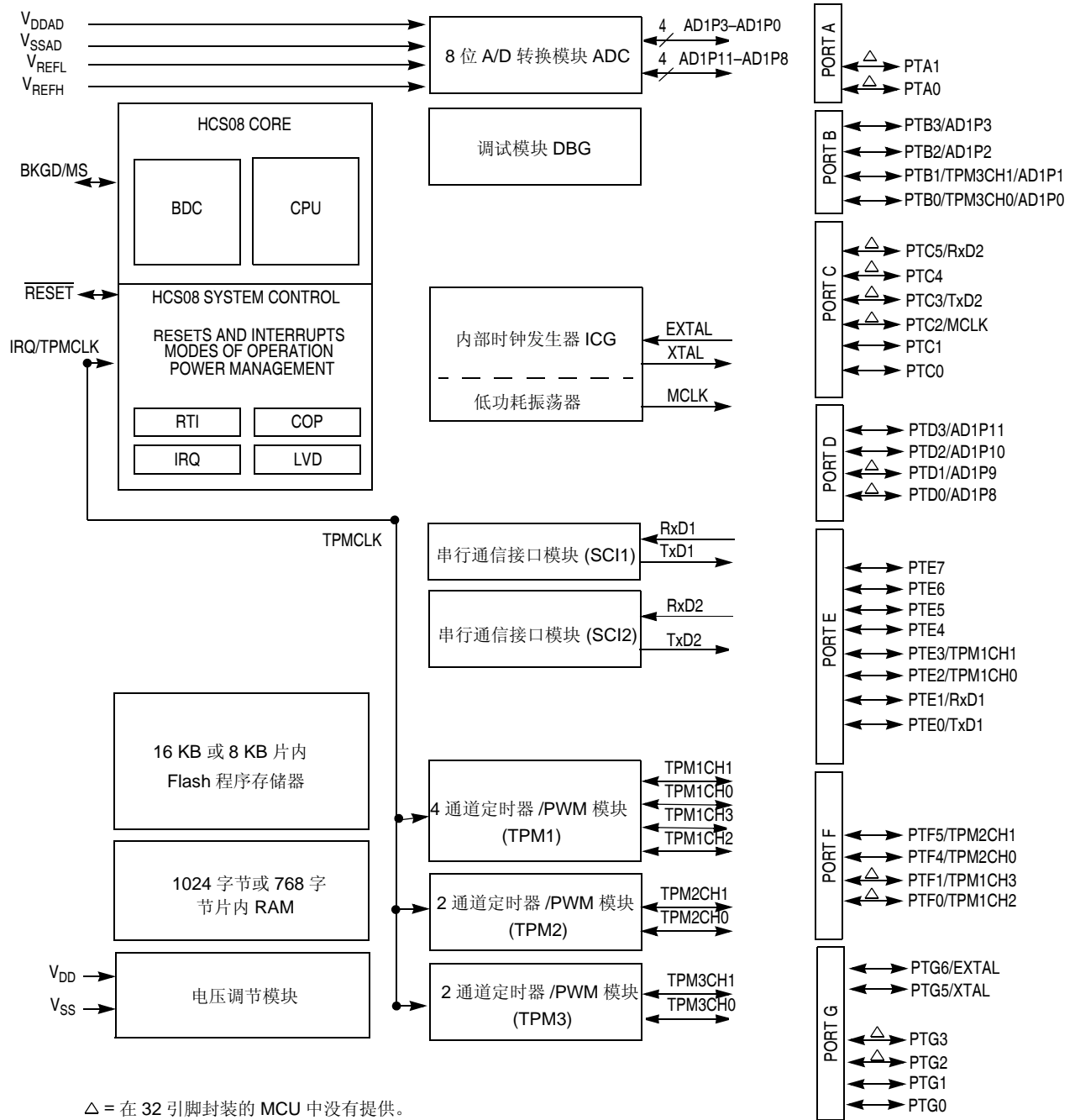
表 1-1 概括了各种 MCU 性能配置。

表 1-1. MCU 的特征和封装

性能	SC667027	
flash 容量（字节）	16K	
内存容量（字节）	1024	
引脚数量	44	32
ADC 通道	8	6
TPM1 通道	4	2
TPM2 通道	2	2
TPM3 通道	2	2
GPIO 引脚	34	22

1.2 MCU 结构框图

SC667027 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. SC667027 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
串行通信接口 (SCI)	4
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

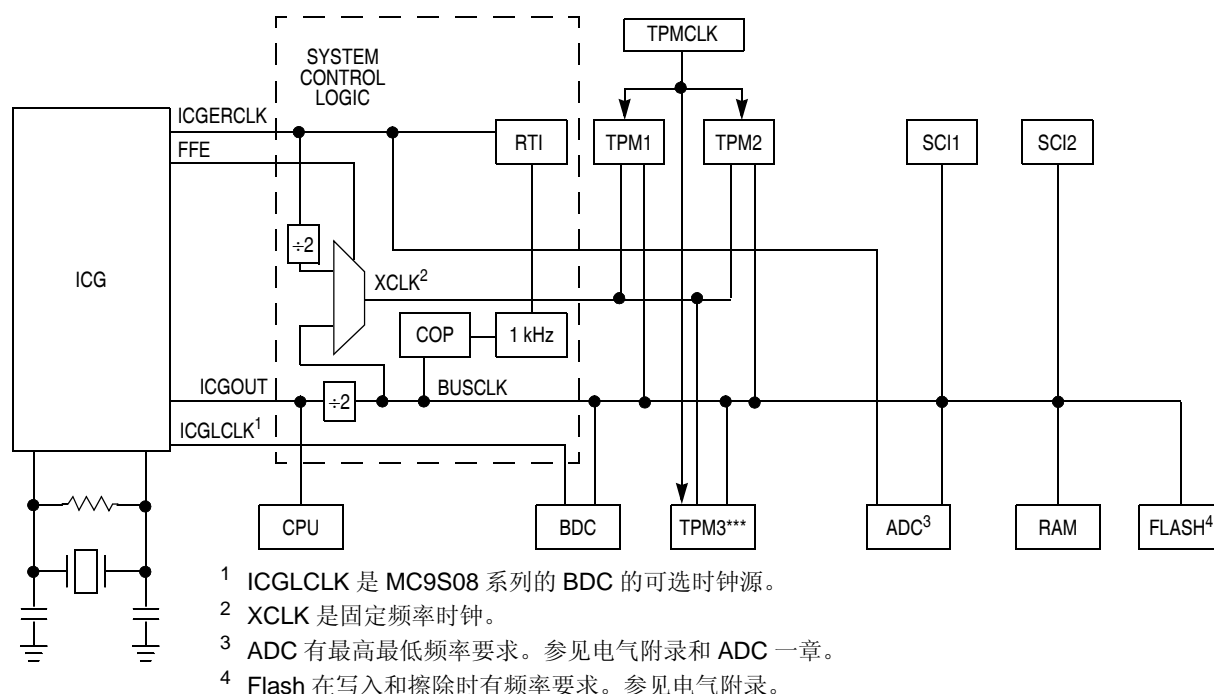


图 1-2. 系统时钟分配图

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

- 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
 - 锁频环子模块中数控振荡器 (DCO) 的输出
 - ICG 内的控制位决定它与那个时钟源相连
- FFE 是在 ICG 里面产生的一个控制信号。如果 ICGOUT 的频率 $>4*$ (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 ALTCLK 的输入

第 2 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，包括一张引脚图，一张信号表和信号的具体描述。

2.2 MCU 的引脚分配

图 2-1 为 SC667027 系列设备的 44 引脚 LQFP 引脚排列。

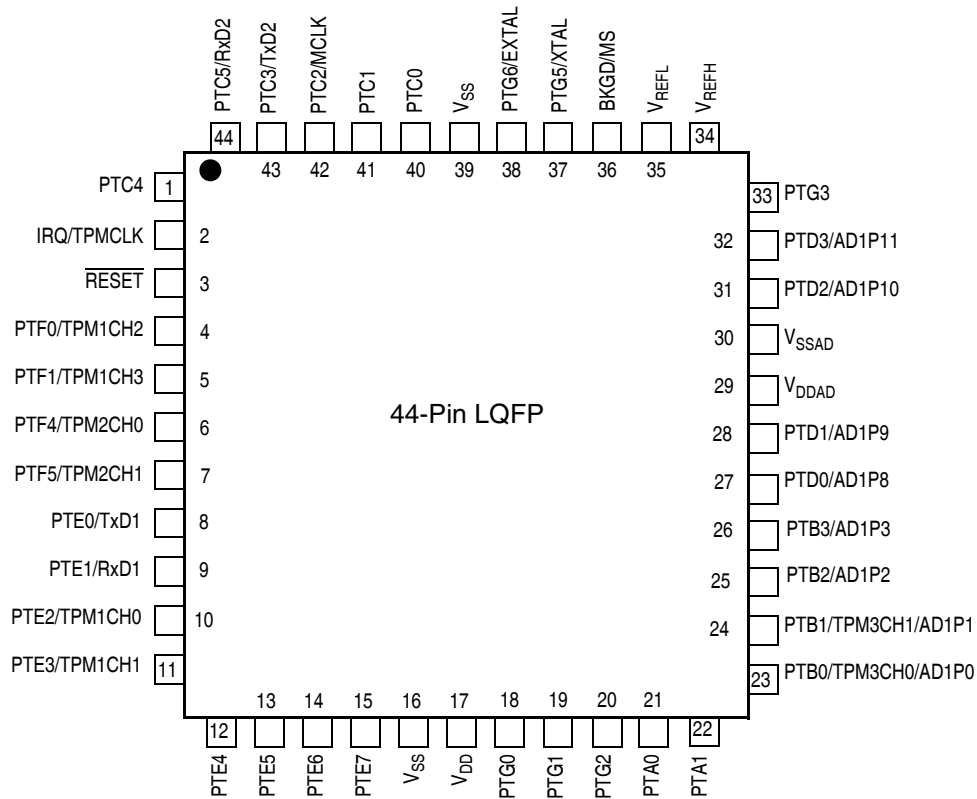


图 2-1. SC667027 系列 44 引脚 LQFP 封装

图 2-2 为 SC667027 系列 MCU 的 32 引脚 LQFP 引脚排列。

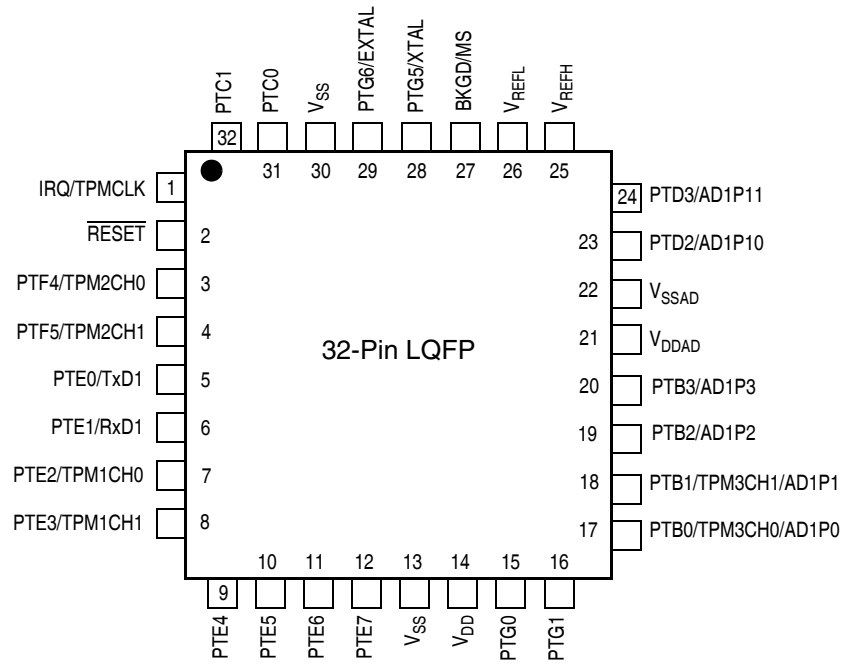


图 2-2. SC667027 系列 32 引脚 LQFP 封装

表 2-1. 各封装的引脚列表

管脚号		最低 <-- 优先级 --> 最高		
44	32	Port Pin	Alt 1	Alt 2
1	—	PTC4		
2	1		IRQ	TPMCLK
3	2			RESET
4	—	PTF0	TPM1CH2	
5	—	PTF1	TPM1CH3	
6	3	PTF4	TPM2CH0	
7	4	PTF5	TPM2CH1	
8	5	PTE0		TxD1
9	6	PTE1		RxD1
10	7	PTE2	TPM1CH0	
11	8	PTE3	TPM1CH1	
12	9	PTE4		
13	10	PTE5		
14	11	PTE6		
15	12	PTE7		
16	13			V _{SS}
17	14			V _{DD}
18	15	PTG0		
19	16	PTG1		
20	—	PTG2		
21	—	PTA0		
22	—	PTA1		

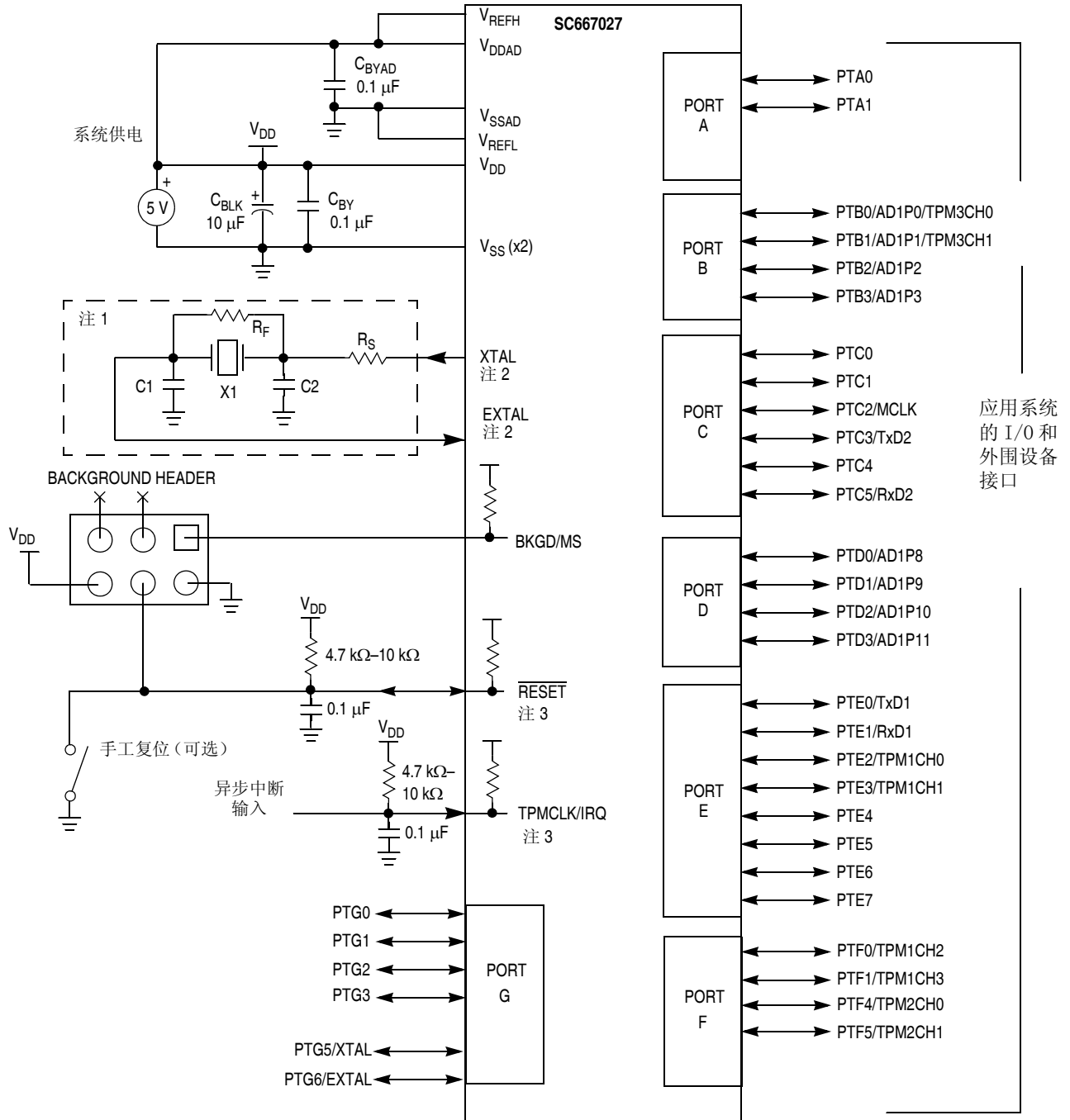
管脚号		最低 <-- 优先级 --> 最高		
44	32	Port Pin	Alt 1	Alt 2
23	17	PTB0	TPM3CH0	AD1P0
24	18	PTB1	TPM3CH1	AD1P1
25	19	PTB2	AD1P2	
26	20	PTB3	AD1P3	
27	—	PTD0	AD1P8	
28	—	PTD1	AD1P9	
29	21			V _{DDAD}
30	22			V _{SSAD}
31	23	PTD2	AD1P10	
32	24	PTD3	AD1P11	
33	—	PTG3		
34	25			V _{REFH}
35	26			V _{REFL}
36	27	BKGD	MS	
37	28	PTG5	XTAL	
38	29	PTG6	EXTAL	
39	30			V _{SS}
40	31	PTC0		
41	32	PTC1		
42	—	PTC2		MCLK
43	—	PTC3		TxD2
44	—	PTC5		RxD2

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”
定时器 / PWM	TCLK、TPMCHx	第 10 章 “定时器 / PWM”
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 2-3 显示了适用于几乎所有 SC667027 系列应用系统典型的引脚连接。



注意：

- 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 为了得到较好的电磁兼容性能，推荐将 RESET 和 IRQ 接上 RC 过滤器。

图 2-3. 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 $10 \mu\text{F}$ 的钽电容, 用来为全体系统提供大电流充电存储, 和一个 $0.1 \mu\text{F}$ 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。SC667027 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)

复位后, MCU 采用内部产生的频率 (自时钟模式) 相当于 8 MHz 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振启动延时。MCU 也包含了一个可微调的内部时钟产生器 (ICG) 模块, 这个模块用来运行 MCU。关于 ICG 的更多信息参考第 8 章 内部时钟发生器 (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, 也可采用外部振荡器和 EXTAL 输入引脚相连。

下面的讨论将参考图 2-3。 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 $10 \text{ M}\Omega$, 对湿度和低值减小增益很敏感, (在极端地情况下) 可能导致无法启动。

$C1$ 和 $C2$ 通常在 5 pF 到 25 pF 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, 每个振荡器引脚 (EXTAL 和 XTAL) 与 PCB 的组合电容约为 10 pF 。

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, 没有输出斜率控制。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 2-3 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斜率控制。

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决定 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。如果没有使能 IRQ，该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 2-3 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/输出，内部上拉设备无效。

注意

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 6 章 并行输入和输出。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 6 章 并行输入和输出获取更多信息。

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 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沿。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沿给较低优先级模块。建议在使能一个模块前关闭其他共用同一引脚的所有模块。

第 3 章 工作模式

3.1 简介

本章讲述了 SC667027 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 特性

- 普通用户在运行模式下操作
- 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 SC667027 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 0xFFFF:0xFFFF 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 (bootloader) 或者用户应用程序。当 SC667027 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 12 章 开发支持。

3.5 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），也包括停止模式 1。SC667027 系列 MCU 不包括停止模式 1。

表 3-1 总结了 MCU 在每种停止模式下的行为。

表 3-1. 停止模式下的行为

模式	PPDC	CPU, 数字 外设, flash	RAM	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。

3.5.1 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模式 2（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模式 3。

进入 STOP2 模式之前，用户可以在 RAM 中保存 I/O 口寄存器的内容，和其它想在退出 STOP2 模式后恢复的寄存器的内容。在离开 STOP2 模式后，这些内容可在闭锁的端口开放之前被用户软件恢复。

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

通过 MCU 的唤醒端口 ($\overline{\text{RESET}}$ 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寄存器控制。

3.5.2 Stop3 模式

为了进入 STOP3 模式，在执行 STOP 指令前必须选择停止模式 3（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 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 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 1 kHz 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。

3.5.3 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 12 章 开发支持介绍。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

3.5.4 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位 SPMSC1 寄存器中的 LVDE 和 LVDSE 位使能 LVD，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 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

3.5.5 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 3.5.2 节 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。
- **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。

- **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 4 章 片上存储器

4.1 SC667027 系列存储映像简介

图 4-1 描述了 SC667027 系列 MCU 的存储映像，SC667027 的片上存储器由 RAM、非易失的 flash 存储器、输入 / 输出 (I/O) 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)

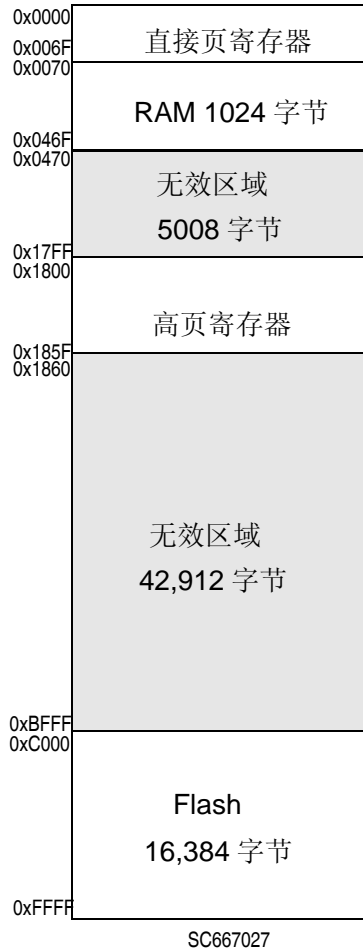


图 4-1. SC667027 系列内存映像

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 Freescale 公司提供的 SC667027 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、中断和系统配置。

表 4-1. 复位和中断向量表

地址 (高 / 低)	向量	向量名称
0xFFC0:FFC1 至 0xFFC4:FFC5	不用的向量空间 (可用于用户程序)	—
0xFFC6:FFC7	TPM3 溢出	Vtpm3ovf
0xFFC8:FFC9	TPM3 通道 1	Vtpm3ch1
0xFFCA:FFCB	TPM3 通道 0	Vtpm3ch0
0xFFCC:FFCD	RTI	Vrti
0xFFCE:FFCF	未使用	—
0xFFD0:FFD1	ADC 模数转换	Vadc1
0xFFD2:FFD3	未使用	—
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	未使用	—
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

表 4-1. 复位和中断向量表 (continued)

0xFFFA:FFFB	IRQ	Virq
0xFFFC:FFFD	SWI	Vswi
0xFFFE:FFFF	RESET	Vreset

4.2 寄存器地址和位分配

SC667027 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 0x1800 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 0xFFB0~0xFFBF 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三个值
 - 一个 8 字节的密码，用于取得加密存储器的使用权

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，以与右边的位名称区分。没有定义的位单元格以阴影表示。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无效或保留的字节，读到的值是 1 或 0。

表 4-2. 直接页寄存器概述 (Sheet 1 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x0000	PTAD	R	R	R	R	R	R	PTAD1	PTAD0
0x0001	PTADD	R	R	R	R	R	R	PTADD1	PTADD0
0x0002	PTBD	R	R	R	R	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	PTBDD	R	R	R	R	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	PTCD	0	R	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	PTCDD	0	R	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006	PTDD	R	R	R	R	PTDD3	PTDD2	PTDD1	PTDD0
0x0007	PTDDD	R	R	R	R	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x000A	PTFD	R	R	PTFD5	PTFD4	R	R	PTFD1	PTFD0
0x000B	PTFDD	R	R	PTFDD5	PTFDD4	R	R	PTFDD1	PTFDD0
0x000C	PTGD	0	PTGD6	PTGD5	R	PTGD3	PTGD2	PTGD1	PTGD0
0x000D	PTGDD	0	PTGDD6	PTGDD5	R	PTGDD3	PTGDD2	PTGDD1	PTGDD0
0x000E– 0x000F	保留	—	—	—	—	—	—	—	—
0x0010	ADCSC1	COCO	AIEN	ADCO	ADCH				
0x0011	ADCSC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
0x0012	ADCRH	0	0	0	0	0	0	0	0
0x0013	ADCRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	ADCCVH	0	0	0	0	0	0	0	0
0x0015	ADCCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	ADCCFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0x0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
0x001A– 0x001B	保留	—	—	—	—	—	—	—	—
0x001C	IRQSC	0	IRQPDD	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
0x001D	保留	—	—	—	—	—	—	—	—
0x001E	保留	—	—	—	—	—	—	—	—
0x001F	保留	—	—	—	—	—	—	—	—
0x0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0

表 4-2. 直接页寄存器概述 (Sheet 2 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x0028	TPM1C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0029	TPM1C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x002A	TPM1C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x002B	TPM1C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
0x002C	TPM1C2VH	Bit 15	14	13	12	11	10	9	Bit 8
0x002D	TPM1C2VL	Bit 7	6	5	4	3	2	1	Bit 0
0x002E	TPM1C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
0x002F	TPM1C3VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0030	TPM1C3VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0031– 0x0037	保留	—	—	—	—	—	—	—	—
0x0038	SCI1BDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0039	SCI1BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x003A	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– 0x005F	保留	—	—	—	—	—	—	—	—
0x0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8

表 4-2. 直接页寄存器概述 (Sheet 3 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x006B– 0x006F	保留	—	—	—	—	—	—	—	—

表 4-3 中所示的高地址页寄存器比其他的 I/O 和控制寄存器的访问频率低，所以它们位于可直接寻址的存储器空间之外，开始于 0x1800。

表 4-3. 高地址页寄存器概要 (Sheet 1 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x1800	SRS	POR	PIN	COP	ILOP	ILAD	ICG	LVD	0
0x1801	SBDFR	0	0	0	0	0	0	0	BDFR
0x1802	SOPT	COPE	COPT	STOPE	—	0	0	—	—
0x1803	SMCLK	0	0	0	MPE	0	MCSEL		
0x1804– 0x1805	保留	—	—	—	—	—	—	—	—
0x1806	SDIDH	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
0x1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
0x1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0 ¹	BGBE
0x180A	SPMSC2	LVWF	LVWACK	LVDV	LVWV	PPDF	PPDACK	—	PPDC
0x180B	保留	—	—	—	—	—	—	—	—
0x180C	SOPT2	COPCLKS	—	—	—	—	—	—	—
0x180D– 0x180F	保留	—	—	—	—	—	—	—	—
0x1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
0x1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
0x1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
0x1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
0x1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
0x1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
0x1816	DBGC	DBGGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
0x1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
0x1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
0x1819– 0x181F	保留	—	—	—	—	—	—	—	—

表 4-3. 高地址页寄存器概要 (Sheet 2 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
0x1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
0x1822	Reserved	—	—	—	—	—	—	—	—
0x1823	FCNFG	0	0	KEYACC	0	0	0	0	0
0x1824	FPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
0x1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
0x1826	FCMD	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
0x1827– 0x182F	保留	—	—	—	—	—	—	—	—
0x1830	TPM3SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x1831	TPM3CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x1832	TPM3CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x1833	TPM3MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x1834	TPM3MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x1835	TPM3C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x1836	TPM3C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x1837	TPM3C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x1838	TPM3C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x1839	TPM3C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x183A	TPM3C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x183B 0x183F	保留	—	—	—	—	—	—	—	—
0x1840	PTAPE	R	R	R	R	R	R	PTAPE1	PTAPE0
0x1841	PTASE	R	R	R	R	R	R	PTASE1	PTASE0
0x1842	PTADS	R	R	R	R	R	R	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

表 4-3. 高地址页寄存器概要 (Sheet 3 of 3)

地址	寄存器名称	第 7 位	6	5	4	3	2	1	第 0 位
0x1852	PTEDS	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
0x1853	保留	—	—	—	—	—	—	—	—
0x1854	PTFPE	R	R	PTFPE5	PTFPE4	R	R	PTFPE1	PTFPE0
0x1855	PTFSE	R	R	PTFSE5	PTFSE4	R	R	PTFSE1	PTFSE0
0x1856	PTFDS	R	R	PTFDS5	PTFDS4	R	R	PTFDS1	PTFDS0
0x1857	保留	—	—	—	—	—	—	—	—
0x1858	PTGPE	0	PTGPE6	PTGPE5	0	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x1859	PTGSE	0	PTGSE6	PTGSE5	0	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x185A	PTGDS	0	PTGDS6	PTGDS5	0	PTGDS3	PTGDS2	PTGDS1	PTGDS0
0x185B– 0x185F	保留	—	—	—	—	—	—	—	—

¹ 保留位 必须写 0。

表 4-4 中所示的非易失性 flash 寄存器位于 flash 存储器中。它包括一个 8 字节的后门密钥，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，从而控制加密存储块保护功能。

表 4-4. 非易失性寄存器概要

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0xFFB0 – 0xFFB7	NVBACKKEY	8-Byte Comparison Key							
0xFFB8 – 0xFFBC	保留	—	—	—	—	—	—	—	—
0xFFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
0xFFBE	保留 ¹	—	—	—	—	—	—	—	—
0xFFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

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解除存储器的安全保护机制。密钥机制只能通过安全存储区域的用户代码访问 (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。如果密钥失效，去除安全保护机制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) 进入非安全保护状态即可。

4.3 RAM

SC667027 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。通常，将访问最频繁的程序变量放在这个区域。

当 MCU 处于低功耗的等待、STOP2、STOP3 模式时，RAM 保持数据。上电时，RAM 的内容不会被重新初始化。如果所提供的电压高于 RAM 所需的最小电压，RAM 数据不会受任何复位的影响。

为了和以前的 M68HC05 MCU 兼容，HCS08 初始化时栈指针定为 0x00FF。在 SC667027 中，通常重新初始化栈指针指向 RAM 的顶端，这样直接页 RAM 可以用于存储频繁访问的 RAM 变量和可位寻址的程序变量。在重启初始化程序中添加下来的两条指令（在 Freescale 提供的文件中 RamLast 等于 RAM 的最高地址）。

```
LDHX #RamLast+1 ;指向 RAM 最高地址 +1
TXS      ;SP<-(H:X-1)
```

在使能保护机制的情况下，RAM 被视为受保护的存储资源，BDM 或者来自非受保护的存储区的代码不能对其进行访问。对于保护机制的特征，请参考 4.5 节 保护机制。

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汇编编译以后下载到 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。

4.4.1 特性

Flash 有以下特性：

- Flash 容量：16384 字节（每页 512 字节，共 32 页）
- 单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 的保护机制
- 自动低功耗的低频率读访问

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 150 kHz 到 200 kHz 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 禁止写入。所以，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。命令处理器完成一个擦除或者写入操作的时间是周期的整数倍。

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 $t_{FCLK}=1/f_{FCLK}$ 。表中各操作的时间分别由所需的 FCLK 时钟周期以及 $t_{FCLK} = 5 \mu s$ 的绝对时间来表示。这个时间也包括了命令状态机的消耗，以及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=200 kHz)
字节写入	9	45 μ s
字节写入 (突发)	4	20 μ s ¹
页擦除	4000	20 ms
整体擦除	20000	100 ms

¹ 不包括开始 / 结束的时间消耗

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操作。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后的第一页。

注意

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 (0x05)，写入一个字节命令 (0x20)，突发模式写入命令 (0x25)，页擦除命令 (0x40)，整体擦除命令 (0x41)。这些命令锁存到命令缓冲区。
- 向 FSTAT 寄存器的 FCBEF 位写 1，可以清 FCBEF 位和启动命令 (包括它的地址和数据信息)。

在写存储器之后，写 1 清零 FCBEF 和命令完成之前的任何时候可以向 FCBEF 手动写 0，取消部分命令序列。用这种方式取消命令可使 FACCERR 存取错误标志位置位，该标志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。这降低了意外改变 flash 内容的可能性。命令完成标志 (FCCF) 标志完成一个命令。命令队列必须通过清零 FCBEF 来开始执行命令。图 4-2 是除了突发模式写入以外，所有命令执行的流程图。在使用任何 flash 命令前，FCDIV 寄存器必须被初始化 (复位后，只能执行一次操作)。

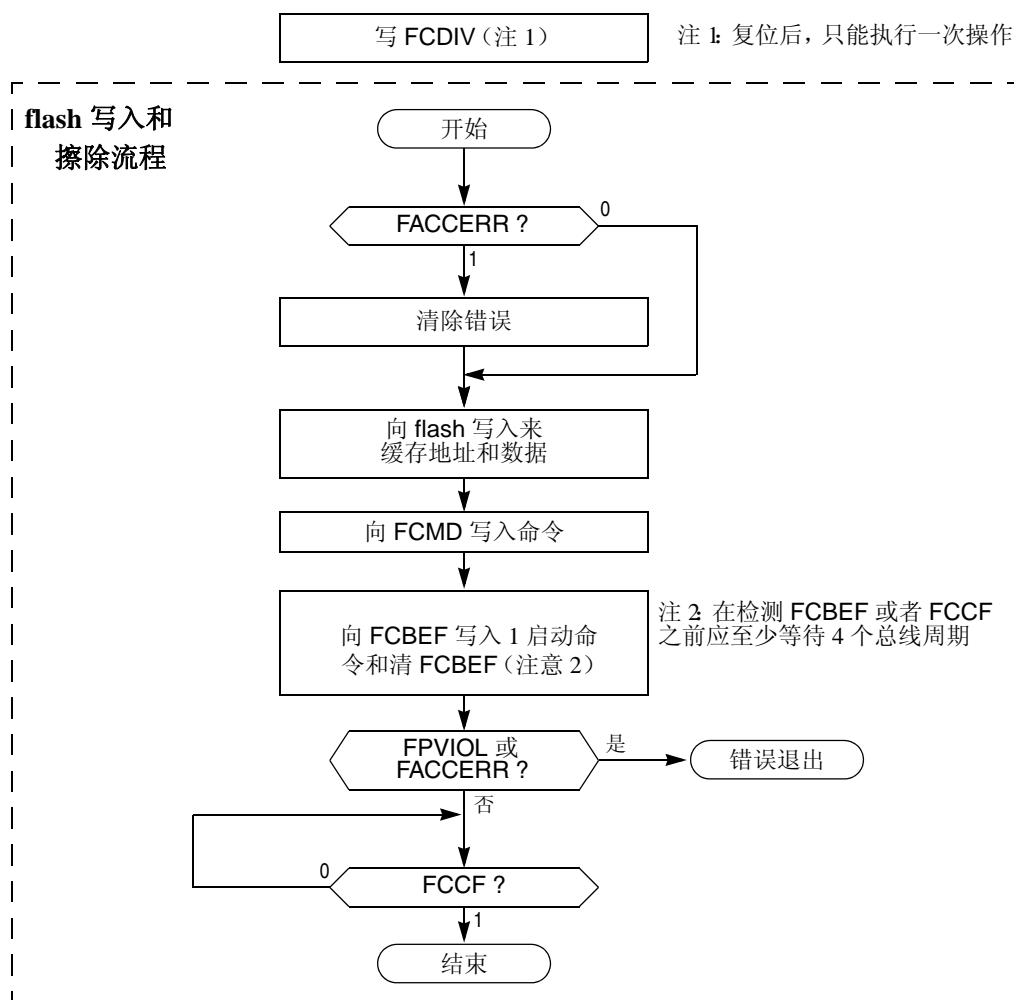


图 4-2. Flash 命令流程图

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 阵列中的高电压持续有效。通常，当发出写入或擦除命令时，必须使能 flash 存储器相关的内部电荷泵来为阵列提供高电压。命令完成后电荷泵被关闭。在突发模式写入命令中，电荷泵被启动，并在满足下面两个条件的情况下，电荷泵保持打开，直到突发模式写入操作完成。

- 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。

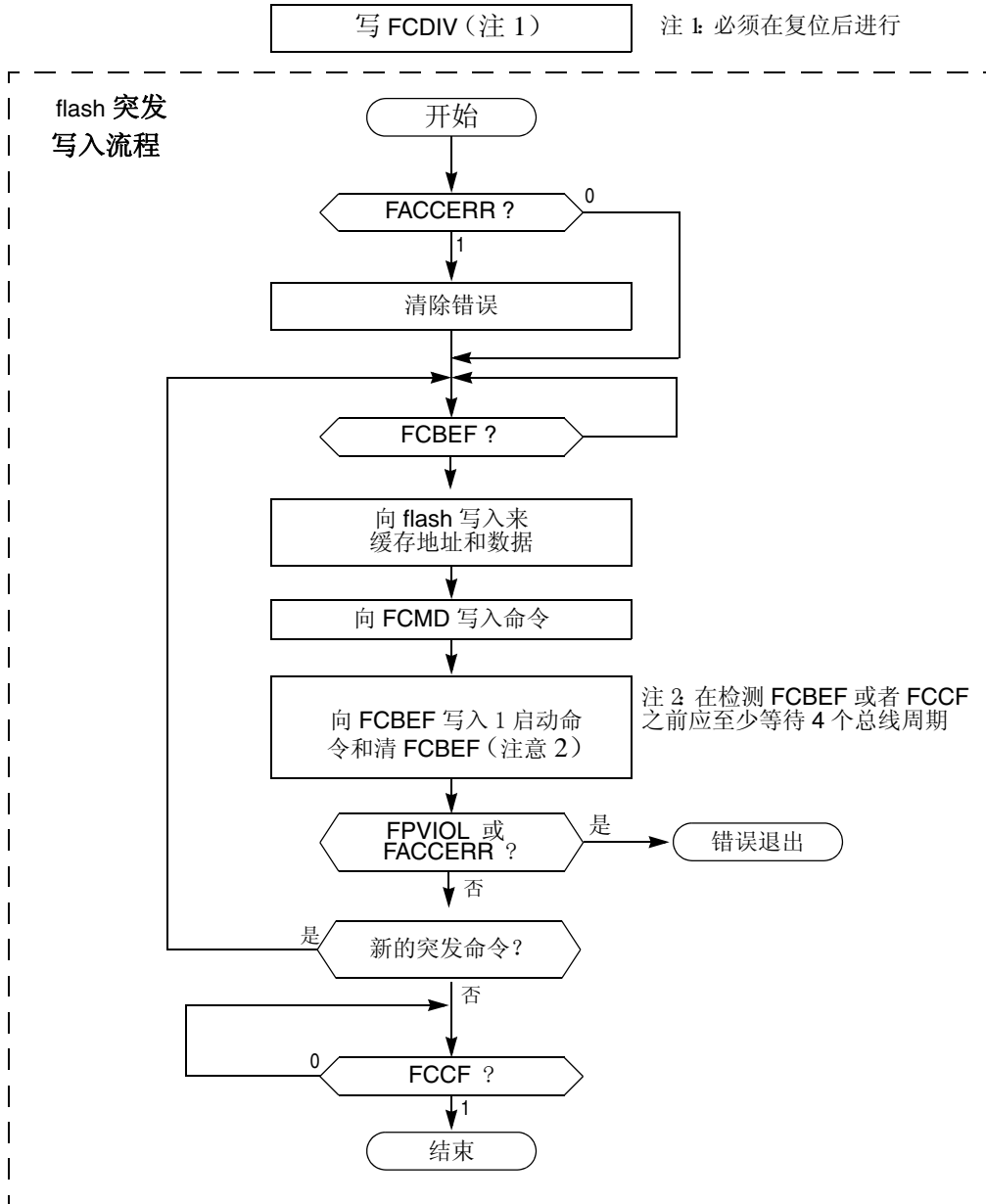


图 4-3. Flash 突发写入流程图

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。如果满足上述条件，剩下的字节也会在突发模式下写入。如果在下一个序列的地址指向新的行，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置位 FSTAT 中的访问错误标志 (FACCERR)。一旦发生了访问错误, 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)。
- 在启动命令之前, 再一次写入 flash 地址 (每一个命令只写一次 flash)。
- 在启动命令之前, 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 和启动命令) 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, 块保护可以开始于任意 512 字节的边界, 这个边界低于 flash 最后的地址 (0xFFFF) (参考 4.6.4 节 Flash 保护寄存器 (FPROT 和 NVPROT))。

复位后, 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 定义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。

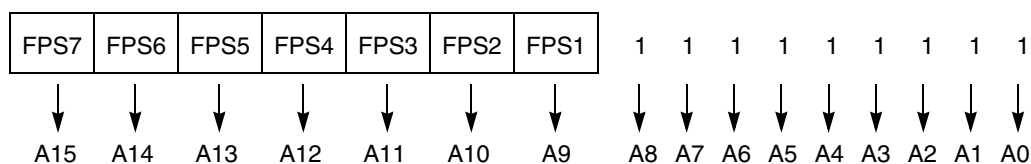


图 4-4. 块保护机制

块保护的一个用途是为引导程序 (Bootloader) 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以及重新写剩余的 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取消对引导程序和复位向量空间的保护。向 NVOPT 内地址为 0xFFBF 的寄存器的 FNORED 位写入 0, 可以使能向量重定向。为了实现重定向操作, 必须设置位于 0xFFBD 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 (0xFFC0 - 0xFFFFD) 都会重定向, 除了复位向量 (0xFFFFE:0xFFFFF)。

举个例子, 如果 flash 的 512 个字节被保护, 保护地址区域是从 0xFE00 到 0xFFFF。则中断向量 0xFFC0 - 0xFFFFD 被重定向到 0xFDC0 - 0xFDFD。以一个 SPI 中断来举例, 地址 0xFDE0: FDE1 将取代地址 0xFFE0: FFE1 的值作为中断向量使用。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 保护机制

SC667027 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。当使能安全机制时, 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被禁止的 (写操作被忽略, 读时总为 0)。

是否使用保护机制是由 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) 使能 MCU 保护机制。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 处于保护模式时, 芯片的调试模式是被禁止的。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 就没有方法离开保护模式。如果 KEYEN 是 1, 一个在保护模式下的用户程序可以通过以下的方法暂时离开保护模式:

- 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, 在下次复位前都会保护模式被取消。

安全密钥只能从保护存储器 (RAM 或 flash) 写入。如果没有保护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 4-2 和表 4-3 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。在进行任何擦除或写入操作之前，向该寄存器写入内容用以设置有效范围内非易失性存储系统时钟的频率。

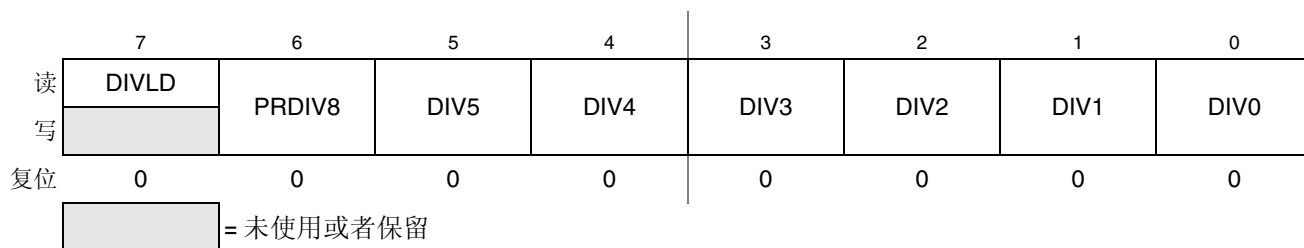


图 4-5. Flash 时钟分频寄存器 (FCDIV)

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 复位后，FCDIV 没有被写过，不可以对 flash 进行擦除和写入操作。 1 复位后，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。

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

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

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

表 4-7. Flash 时钟分频器设置

f_{Bus}	PRDIV8 (二进制)	DIV5:DIV0 (十进制)	f_{CLK}	擦除 / 写入脉冲时间 (最少 5 μs, 最多 6.7 μs)
20 MHz	1	12	192.3 kHz	5.2 μs
10 MHz	0	49	200 kHz	5 μs

表 4-7. Flash 时钟分频器设置 (continued)

8 MHz	0	39	200 kHz	5 μ s
4 MHz	0	19	200 kHz	5 μ s
2 MHz	0	9	200 kHz	5 μ s
1 MHz	0	4	200 kHz	5 μ s
200KHz	0	0	200 kHz	5 μ s
150KHz	0	0	150 kHz	6.7 μ s

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。

	7	6	5	4	3	2	1	0
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

= 未使用或保留

图 4-7. 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保护模式。后门密钥机制只能被保护的固件使用。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 保护机制。 0 禁止 flash 保护密钥。 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，保护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]	保护状态码——这两位决定 MCU 的保护状态，详见表 4-9。当 MCU 在保护模式时，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。保护模式更详细的信息参见 4.5 节 保护机制。

表 4-9. 保护状态码

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。

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


 = 保留或者未使用

图 4-8. Flash 配置寄存器 (FCNFG)

表 4-10. FCNFG 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 4.5 节 保护机制。 0 写 0xFFB0 - 0xFFB7 被认为是 flash 写入或擦写命令的开始。 1 写 NVBACKKEY(0xFFB0 - 0xFFB7) 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。

	7	6	5	4	3	2	1	0	
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 4-9. 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。

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


 = 未使用或保留

图 4-10. 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 可以写入新的突发写入命令到命令缓冲区。
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。对 FCCF 的写操作没有意义。 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 表示侵犯保护区域。 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执行时（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 0xFF。

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 写入和擦除命令的执行。

	7	6	5	4	3	2	1	0
读	0	0	0	0	0	0	0	0
写	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
复位	0	0	0	0	0	0	0	0

图 4-11. Flash 命令寄存器 (FCMD)

表 4-13. FCMD 寄存器域描述

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	0x05	mBlank
字节写入	0x20	mByteProg
字节写入—快速模式	0x25	mBurstProg
页擦除（每页 512 字节）	0x40	mPageErase
整体擦除（整个 flash）	0x41	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 5 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 SC667027 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 特性

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 0x00FF。

SC667027 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知的开始点。

在任何一个复位之后，置位 SOPT 寄存器 (更多信息见 5.9.4 节 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 节 系统系统选项寄存器 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 (32 ms) ¹
0	1	~1kHz	2^8 周期 (256 ms)
1	0	总线	2^{13} 周期
1	1	总线	2^{18} 周期

¹ 表中数据基于 $t_{RTI} = 1$ ms 计算的。关于 t_{RTI} 以及表中数据的允许误差，参考 A.9.1 节 控制时序。

即使应用程序使用 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 (使能中断) 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。

注意

为了能够和 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。

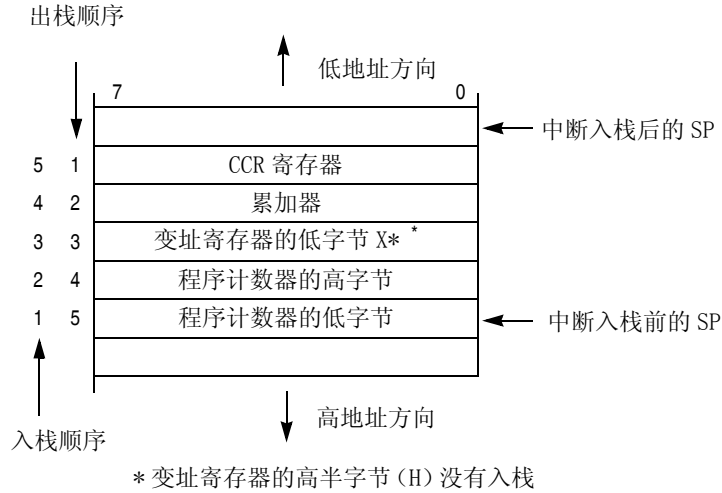


图 5-1. 中断堆栈框架

当一条 RTI 指令被执行时，这些值就会按照出栈的顺序从堆栈中恢复过来。作为 RTI 序列的一部分，CPU 通过读取三个字节的程序信息来装满流水线，从由堆栈中恢复的 PC 地址开始。

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

5.5.2 外部中断请求 (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，IRQ(如果被使能) 通过异步通道可以将 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上拉 / 下拉设备 (IRQPDD=0)，是上拉还是下拉由选择的极性决定。如果用户需要使用外部上拉或下拉，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。

注意

引脚到 V_{DD} 不包含箝位二极管，因此不能输入超过 V_{DD} 的电压。上拉 IRQ 引脚标准电压必须低于 $V_{DD} - 0.7V$ 。所有与该引脚相连的内部始终与 V_{DD} 相连。

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 5 V 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，所以它负责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	
从低到高	29-31	0xFFC0/FFC1– 0xFFC4/FFC5	未使用的向量空间 (对用户程序可用)					
	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	未使用的向量空间					
	23	0xFFD0/FFD1	Vatd	ATD	COCO	AIEN	ADC1	
	22	0xFFD2/FFD3	未使用的向量空间					
	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 SPTF	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	未使用的向量空间					
	9	0xFFEC/FFED	未使用的向量空间					
	8	0xFFEE/FFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 通道 3	
	7	0xFFFF0/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)

SC667027 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 VLVDH 或者低电压 VLVDL)。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。如果 LVDSE 位没有被置位，停止模式中不能使用 LVD。如果 LVDSE 和 LVDE 都被置位，则 MCU 不能进入停止模式 1 和停止模式 2，并且由于 LVD 被激活的停止模式 3 的功耗将会变大。

5.6.1 上电复位操作

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

5.6.2 LVD 复位操作

此时 LVD 能被设置为在侦测到一个低电压条件时 (即 LVDRE 为 1 时) 产生一个复位。LVD 复位发生后，LVD 系统将会保持 MCU 在复位状态直到供应电压升高到高于由 LVDV 设定的电压。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。引脚的输出斜率和驱动强度分别由 PTCSE2 和 PTCDS2 控制。如果输出斜率控制使能，最大时钟输出频率被限制，输出斜率使能时引脚上升和下降的时间附录 A 电气特性和时序规范。

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 片上存储器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 3 章 工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 的无效位，4 个可读写位，一个只读状态位，和一个只写位。这些位都是被用于设置 IRQ 功能，报告状态和确认 IRQ 事件。

	7	6	5	4	3	2	1	0
读	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IQRMOD
写						IRQACK		
复位	0	0	0	0	0	0	0	0
	= 保留或未使用							

表 5-3. 中断请求状态和控制寄存器 (IRQSC)

表 5-4. IRQSC 寄存器域描述

域	描述
6 IRQPDD	中断请求上下拉设备禁止位——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 上下拉设备允许。 1 如果 IRQPE=1，IRQ 上下拉设备禁止。
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。IQRMOD 控制位决定 IRQ 引脚是边沿 / 电平触发还是仅仅为边沿触发。当 IRQ 引脚作为 IRQ 输入引脚使能，并且设置为侦测引脚上升沿时，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。

表 5-4. IRQSC 寄存器域描述 (continued)

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 检测到 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 IRQF 置位不产生中断请求 (使用软件查询)。 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 边沿 / 电平触发有更加详细的介绍。 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 SBD FR 寄存器的 BDFR 位写入 1 强制复位时，SRS 的各标志位都不会被置位。向该寄存器地址写入任意值可清零 COP 看门狗计数器但不影响寄存器的内容。这些位的复位状态取决于 MCU 复位的原因。

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

表 5-5. 系统复位状态 (SRS)

表 5-6. 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。

表 5-6. SRS 寄存器域描述 (continued)

域	描述
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。 SC667027 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。

	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

= 保留或未使用

¹ BDFR 只能通过背景调试命令而不能通过用户程序写入

图 5-2. 系统背景调试强制复位寄存器 (SBDFR)

表 5-7. 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。

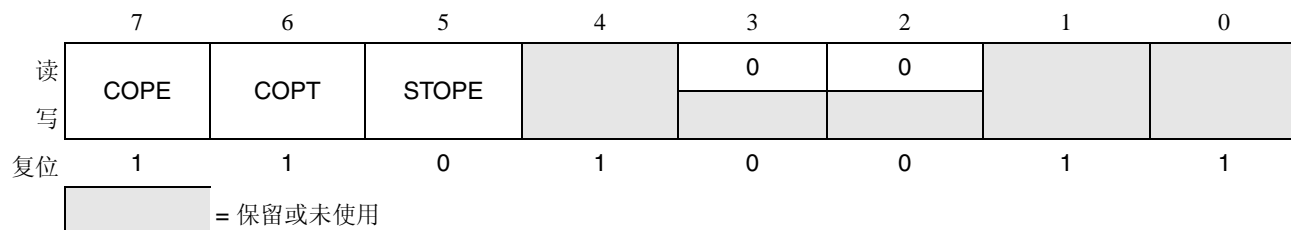


图 5-3. 系统选项寄存器 (SOPT)

表 5-8. 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。

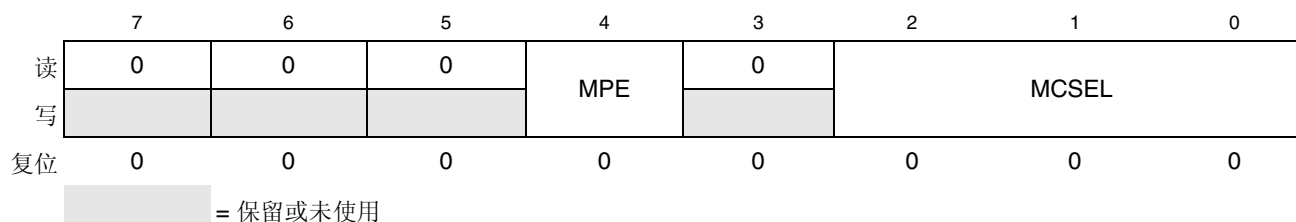


表 5-9. 系统 MCLK 控制寄存器 (SMCLK)

表 5-10. 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

$$\text{MCLK 频率} = \text{总线频率} \div (2 \times \text{MCSEL})$$

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。

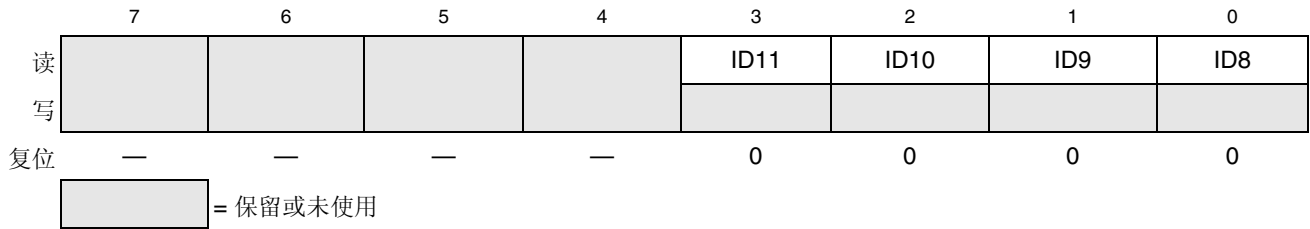


图 5-4. 系统标识寄存器 — 高 (SDIDH)

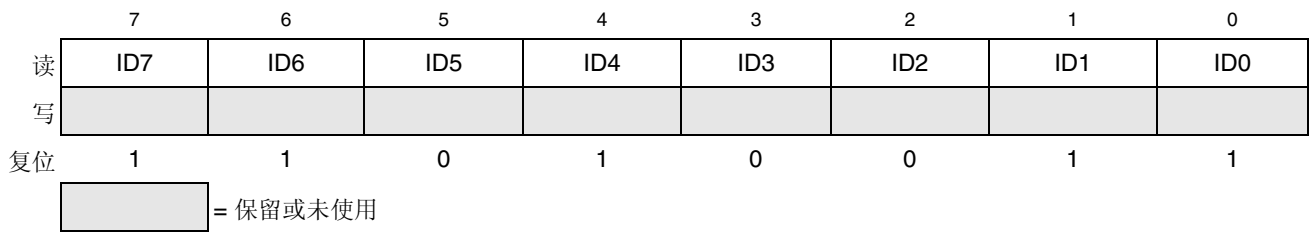


图 5-5. 系统标识寄存器 - 低 (SDIDL)

表 5-11. 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。SC667027 系列编码位 0x012。参见表 5-9 的 ID 位。

表 5-12. 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。SC667027 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。

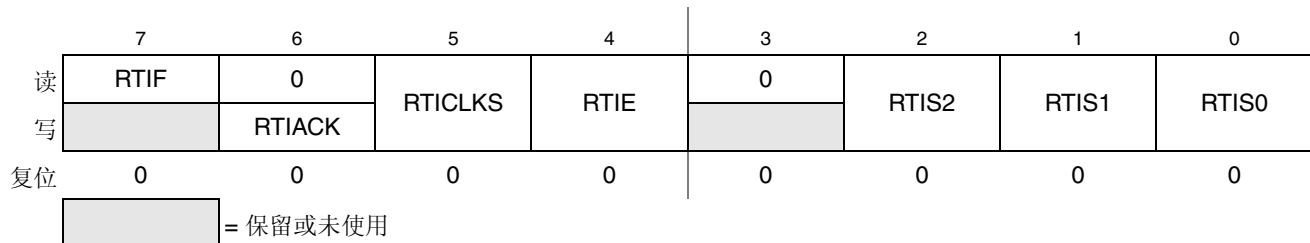


表 5-13. SRTISC 寄存器域描述

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真实时中断请求 (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——该可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 1 kHz 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——该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 1 kHz 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 5-14. 实时中断频率

RTIS2:RTIS1:RTIS0	1kHz 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	8 ms	divide by 256
0:1:0	32 ms	divide by 1024
0:1:1	64 ms	divide by 2048
1:0:0	128 ms	divide by 4096
1:0:1	256 ms	divide by 8192
1:1:0	512 ms	divide by 16384
1:1:1	1.024 s	divide by 32768

¹ 列中的值以 $f_{RTI} = 1 \text{ kHz}$ 为基准，关于这些值的规定参见附录 A 电气特性和时序规范

5.9.8 系统电源管理状态控制寄存器 1 (SPMSC1)

	7	6	5	4	3	2	1 ¹	0
读	LVDF	0	LVDIE	LVDRE ²	LVDSE ²	LVDE(2)	保留	BGBE
写	保留	保留						
复位	0	0	0	1	1	1	0	0

保留 = 保留或未使用

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 5-6. 系统电源管理状态控制寄存器 1 (SPMSC1)

表 5-15. 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。

	7	6	5	4	3	2	1	0
读	LVWF	0	LVDV	LVWV	PPDF	0		PPDC ¹
写		LVWACK						
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

= 保留或未使用

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 5-7. 系统电源管理状态和控制寄存器 2 (SPMSC2)

表 5-16. 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。

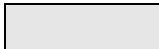
表 5-16. SPMSC2 寄存器域描述

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 (V_{LVD})。 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 (V_{LVW})。 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 SC667027 系列设备的具体特征的位。

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

 = 保留或未使用

¹ 这个位复位后只能写一次。其他的写入被忽略。

图 5-8. 系统系统选项寄存器 2 (STOP2)

表 5-17. 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 6 章

并行输入和输出

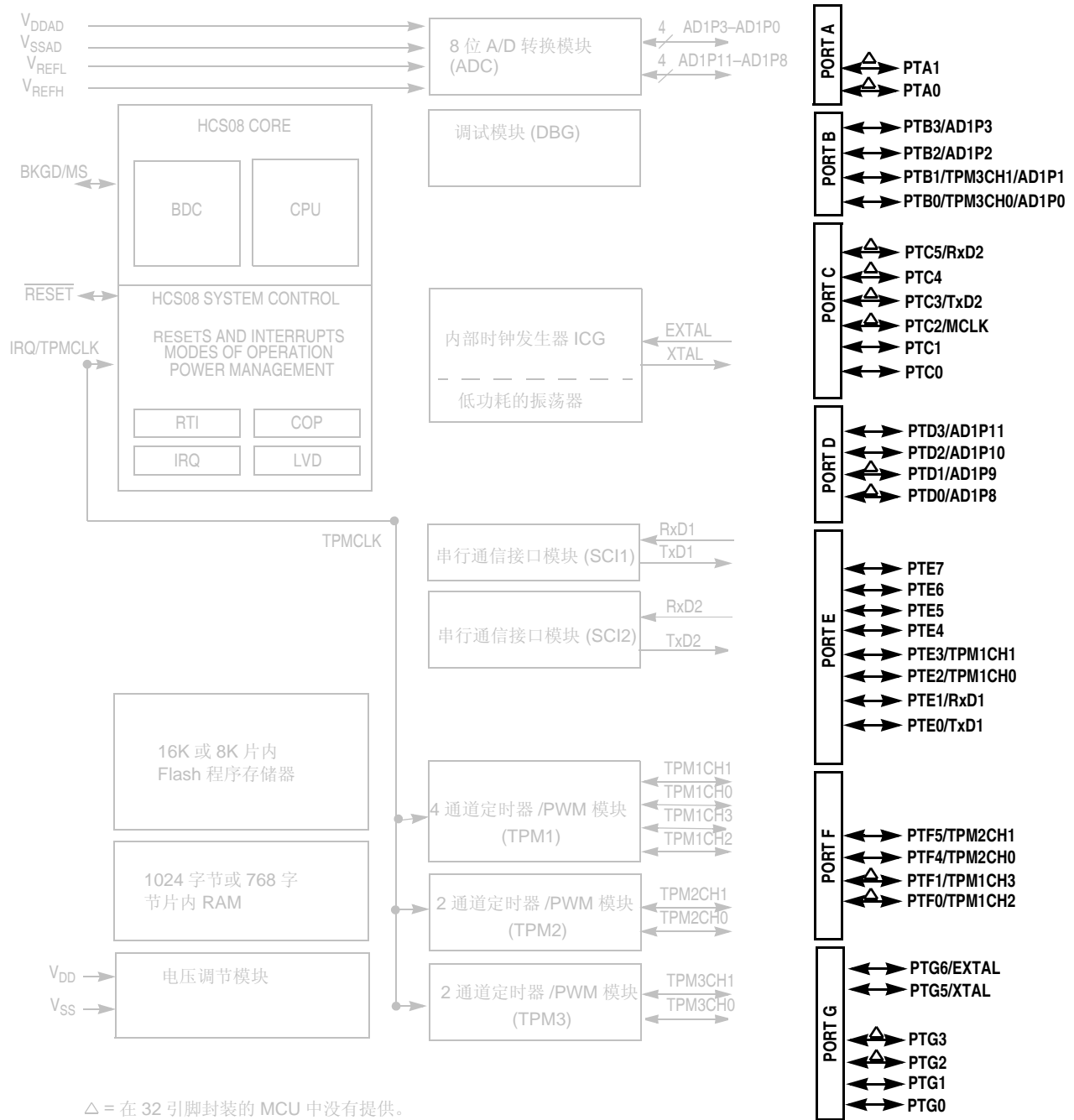
6.1 简介

本章解释了软件控制相关的并行输入输出（I/O）。SC667027 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 2 章 [引脚及其连接](#)。

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。

注意

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:

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. 显亮并行输入 / 输出引脚的 SC667027 模块图

6.2 特性

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 34 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
- 软件可控制的压摆率输出缓存
- 2 个 A 口引脚
- 4 个 B 口引脚与 ADC 和 TPM3 复用
- 6 个 C 口引脚与 SCI2 和 MCLK 复用
- 4 个 D 口引脚与 ADC
- 8 个 E 口引脚与 SCI1 和 TPM1 复用。
- 5 个 F 口引脚与 TPM1 和 TPM2 复用。
- 7 个 G 口引脚与 XTAL 和 EXTAL 复用。

6.3 引脚描述

SC667027 系列总共有 7 个端口（PTA-PTG）共 34 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：输出斜率控制使能（PtxSEn=1），选择低驱动强度（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

	7	6	5	4	3	2	1	0
MCU 引脚	R	R	R	R	R	R	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），输出斜率控制（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

	7	6	5	4	3	2	1	0
MCU 引脚	R	R	R	R	PTB3/ AD1P3	PTB2/ AD1P2	PTB1/ TPM3CH1/ AD1P1	PTB0/ TPM3CH0/ 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），输出斜率控制（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 9 章 定时器 /PWM (S08TPMV3)，更多关于 B 口用作模拟输入的信息参见第 11 章 AD 转换器 (S08ADC10V1)。

6.3.3 C 口

	7	6	5	4	3	2	1	0
MCU 引脚	0	R	PTC5/ RxD2	PTC4	PTC3/ TxD2	PTC2/ MCLK	PTC1	PTC0

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），输出斜率控制（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 10 章 串行通信接口 (S08SCIV4)。

关于把 PTC2 引脚用作 MCLK 引脚的更多信息参见第 5 章 复位、中断和系统配置。

6.3.4 D 口

	7	6	5	4	3	2	1	0
MCU 引脚	R	R	R	R	PTD3/ AD1P10	PTD2/ AD1P10	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），输出斜率控制（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 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 9 章 定时器 /PWM (S08TPMV3)。

关于把 D 口引脚用作模拟输入的更多信息参见第 11 章 AD 转换器 (S08ADC10V1)。

6.3.5 E 口

	7	6	5	4	3	2	1	0
MCU 引脚	PTE7	PTE6	PTE5	PTE4/	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)，输出斜率控制 (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 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 10 章 串行通信接口 (S08SCIV4)。

关于把 E 口引脚用作 TPM 通道引脚的更多信息参见第 9 章 定时器 /PWM (S08TPMV3)。

6.3.6 F 口

	7	6	5	4	3	2	1	0
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)，输出斜率控制 (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 9 章 定时器 /PWM (S08TPMV3)。

6.3.7 G 口

	7	6	5	4	3	2	1	0
MCU 引脚	0	PTG6/ EXTAL	PTG5/ XTAL	R	PTG3	PTG2	PTG1	PTG0/

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），输出斜率控制（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 6.4 节 并行 I/O 控制。

G 口通用 I/O 与 XTAL 和 EXTAL 复用。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 8 章 内部时钟发生器 (S08ICGV4)。

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 中说明。

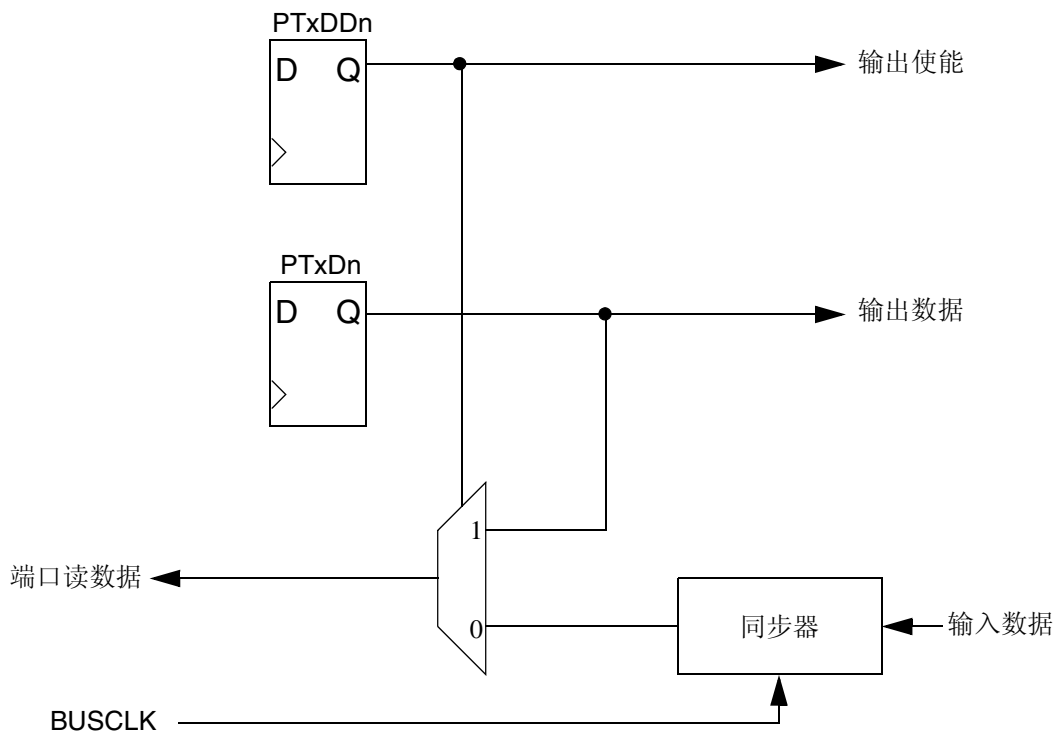


图 6-9. 并行 I/O 框图

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前，先写端口数据寄存器，这是好的编程方法。这样防止 I/O 口输出寄存器中遗留的值。

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，输出斜率和驱动强度。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 输出斜率控制使能

通过设置输出斜率控制寄存器（PTxSEn）的相应位可以使能引脚的输出斜率控制。在使能情况下，限制输出斜率减少 EMC 辐射。输出斜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 I/O 口

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：

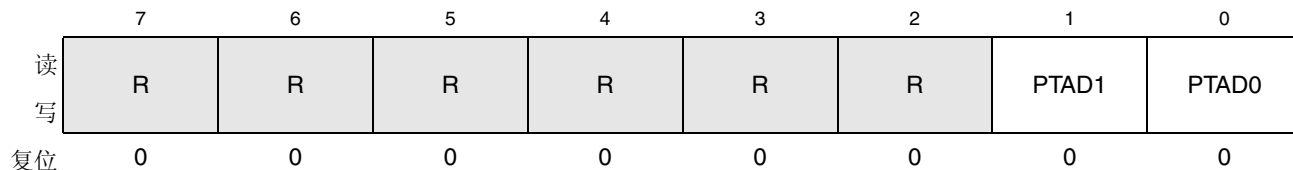
- stop2 模式是局部掉电模式，在 STOP 指令执行以前 I/O 保持它的状态。在 STOP 指令执行 MCU 进入 stop2 模式前，CPU 寄存器状态和 I/O 寄存器状态保存在 RAM 中。从 stop2 模式返回时，访问任何 I/O 前，用户检测 SPMSC2 寄存器的 PPDF 位的状态。如果 PPDF 为 0，I/O 必须像上电复位一样重新初始化。如果 PPDF 为 1，I/O 数据先前存入 RAM，在执行 STOP 指令以前，外围设备必须重新初始化，重新载入先前的数据。用户必须置 1 SPMSC2 寄存器的 PPDACK 位。然后用户应用程序就可以访问 I/O 了。
- 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

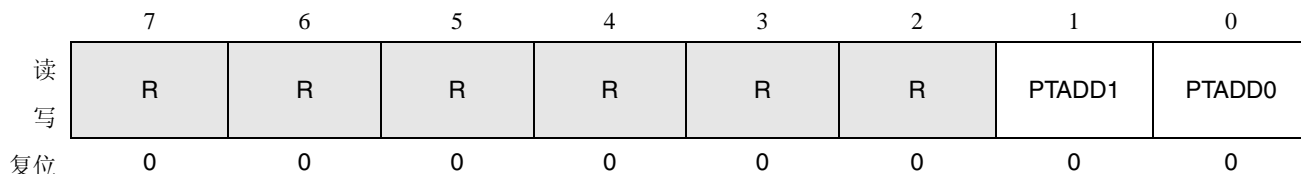


¹ 2-7 位是保留位，必须被写为 0。

图 6-10. A 口数据寄存器 (PTAD)

表 6-1. PTAD 寄存器域描述

域	描述
1:0 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。



¹ 2-7 位是保留位，必须被写为 0。

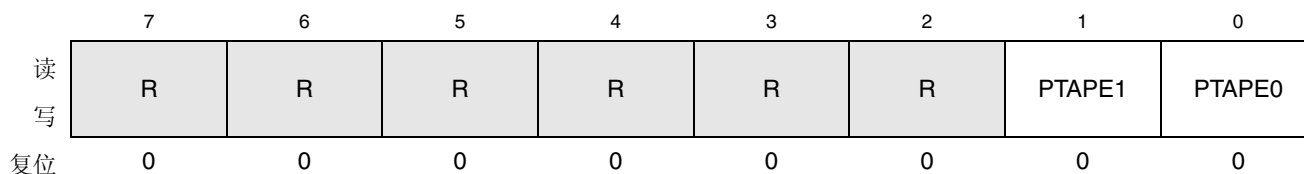
图 6-11. A 口数据方向寄存器 (PTADD)

表 6-2. PTADD 寄存器域描述

域	描述
1:0 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：



¹ 2-7 位是保留位，必须被写为 0。

图 6-12. A 口内部上拉电阻使能 (PTAPE)

表 6-3. PTAPE 寄存器域描述

域	描述
1:0 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。

	7	6	5	4	3	2	1	0
读	R	R	R	R	R	R	PTASE1	PTASE0
写								
复位	0	0	0	0	0	0	0	0

¹ 2-7 位是保留位，必须被写为 0。

图 6-13. A 口输出斜率控制使能寄存器 (PTASE)

表 6-4. PTASE 寄存器域描述

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

	7	6	5	4	3	2	1	0
读	R	R	R	R	R	R	PTADS1	PTADS0
写								
复位	0	0	0	0	0	0	0	0

¹ 2-7 位是保留位，必须被写为 0。

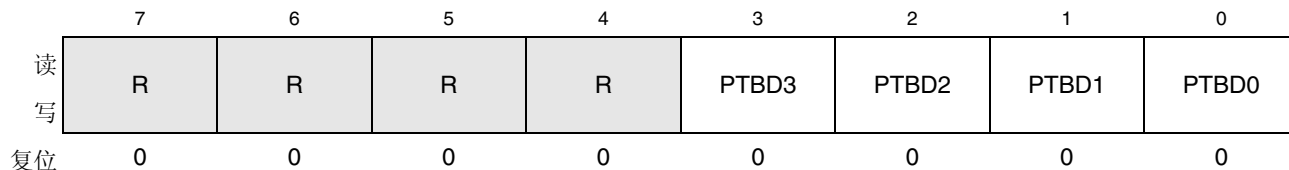
表 6-5. A 口输出驱动强度寄存器 (PTADS)

表 6-6. PTADS 寄存器域描述

域	描述
1:0 PTADSn	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：

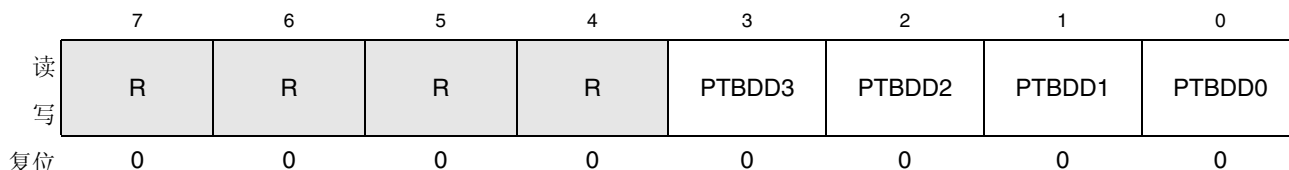


¹ 4-7 位是保留位，必须被写为 0。

图 6-14. B 口数据寄存器 (PTBD)

表 6-7. PTBD 寄存器域描述

域	描述
3:0 PTBD[3:0]	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。



¹ 4-7 位是保留位，必须被写为 0。

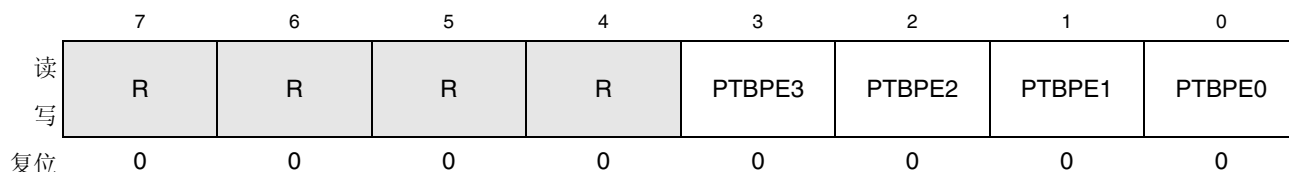
图 6-15. B 口数据方向寄存器 (PTBDD)

表 6-8. 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 PTBDn 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：



¹ 4-7 位是保留位，必须被写为 0。

图 6-16. B 口内部上拉电阻使能 (PTBPE)

表 6-9. 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。

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

¹ 4-7 位是保留位，必须被写为 0。

图 6-17. B 口输出斜率控制使能寄存器 (PTBSE)

表 6-10. PTBSE 寄存器域描述

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

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

¹ 4-7 位是保留位，必须被写为 0。

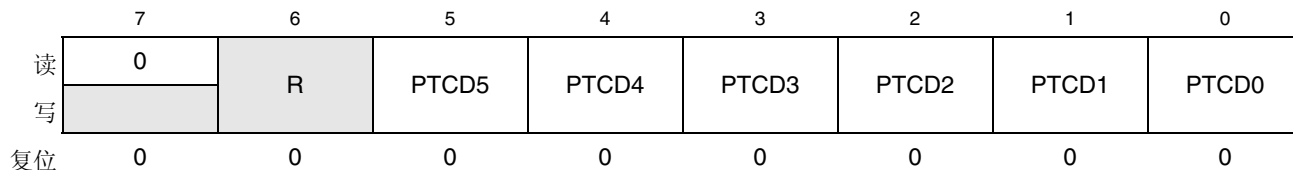
图 6-18. B 口输出驱动强度寄存器 (PTBDS)

表 6-11. 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 PTCD)

C 口并行 I/O 功能被下列寄存器控制：

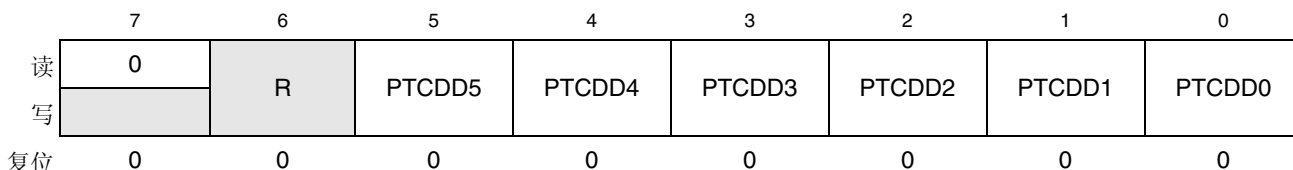


¹ 6 位是保留位，必须被写为 0。

图 6-19. C 口数据寄存器 (PTCD)

表 6-12. PTCD 寄存器域描述

域	描述
5:0 PTCD[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 PTCD 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。



¹ 6 位是保留位，必须被写为 0。

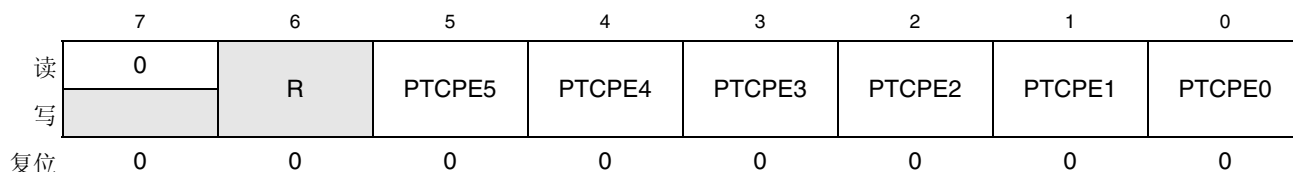
图 6-20. C 口数据方向寄存器 (PTCDD)

表 6-13. 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 PTCD 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 PTCD 读返回 PTCDn 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：

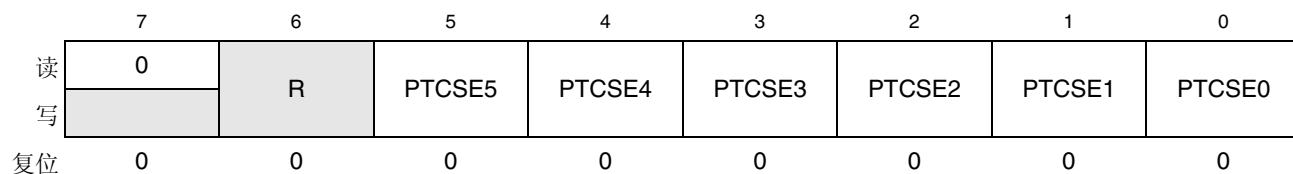


¹ 6 位是保留位，必须被写为 0。

图 6-21. C 口内部上拉电阻使能 (PTCPE)

表 6-14. 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。

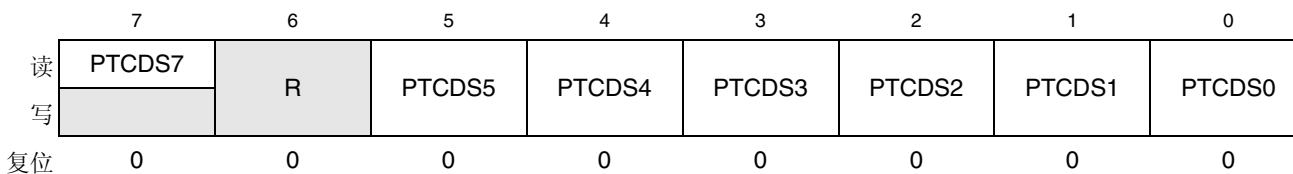


¹ 6 位是保留位，必须被写为 0。

图 6-22. C 口输出斜率控制使能寄存器 (PTCSE)

表 6-15. PTCSE 寄存器域描述

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



¹ 6 位是保留位，必须被写为 0。

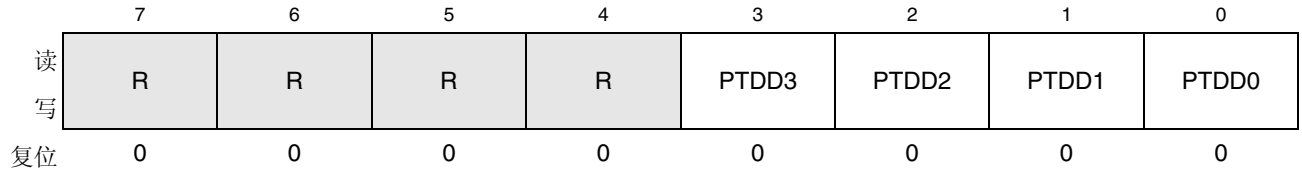
图 6-23. C 口输出驱动强度寄存器 (PTCDS)

表 6-16. 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：

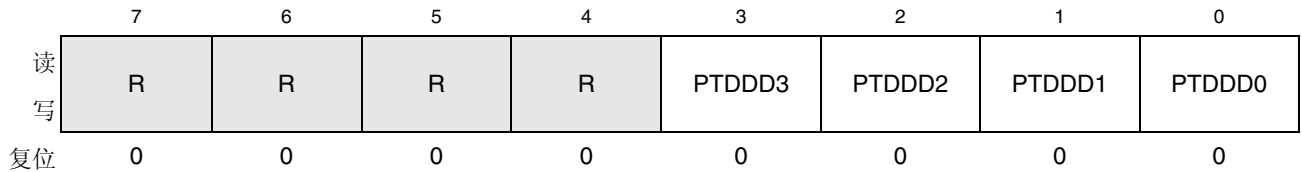


¹ 4-7 位是保留位，必须被写为 0。

图 6-24. D 口数据寄存器 (PTDD)

表 6-17. 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。



¹ 4-7 位是保留位，必须被写为 0。

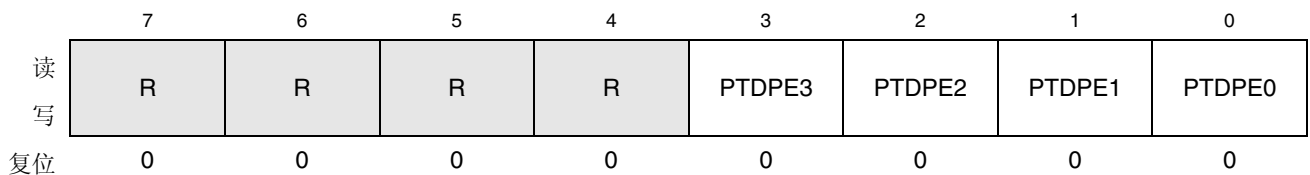
图 6-25. D 口数据方向寄存器 (PTDDD)

表 6-18. 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：



¹ 4-7 位是保留位，必须被写为 0。

图 6-26. D 口内部上拉电阻使能 (PTDPE)

表 6-19. 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。

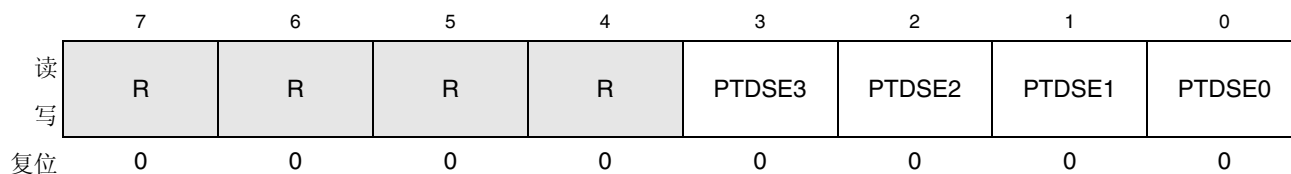


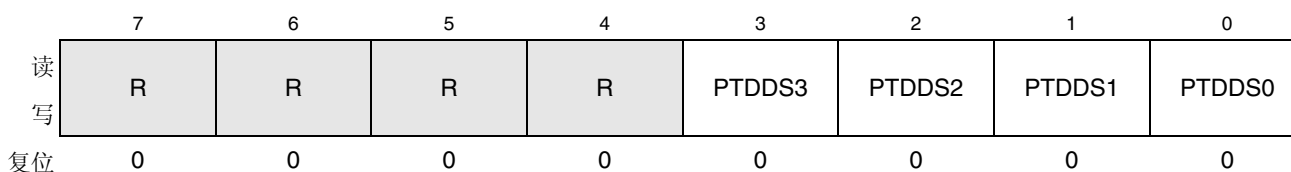
图 6-28 (1)

¹ 4-7 位是保留位，必须被写为 0。

图 6-27. D 口输出斜率控制使能寄存器 (PTDSE)

表 6-20. PTDSE 寄存器域描述

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



¹ 4-7 位是保留位，必须被写为 0。

图 6-28. D 口输出驱动强度寄存器 (PTDDS)

表 6-21. 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：

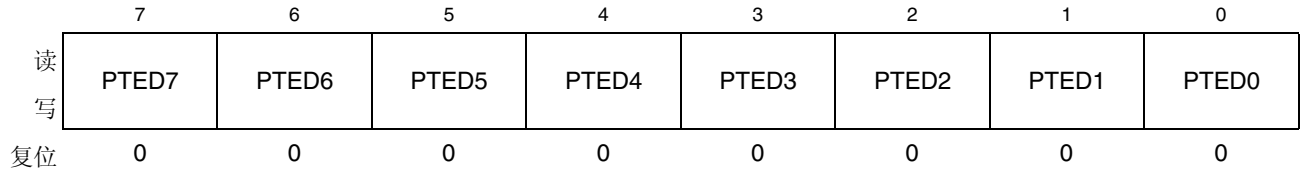


图 6-29. E 口数据寄存器 (PTED)

表 6-22. 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。

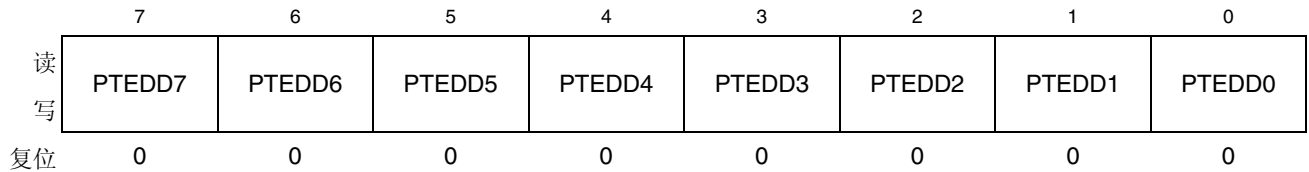


图 6-30. E 口数据方向寄存器 (PTEDD)

表 6-23. 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：

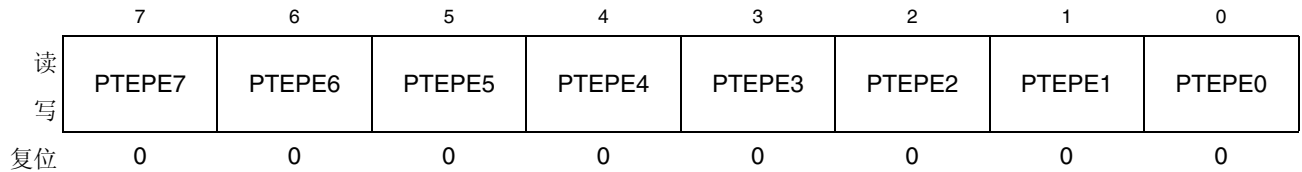


图 6-31. E 口内部上拉电阻使能 (PTEPE)

表 6-24. 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。

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

图 6-32. E 口输出斜率控制使能寄存器 (PTESE)

表 6-25. PTESE 寄存器域描述

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

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

图 6-33. E 口输出驱动强度寄存器 (PTEDS)

表 6-26. 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：

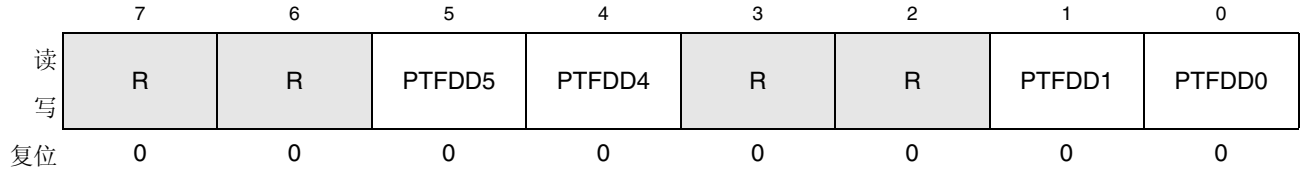


¹ 7、6、3、2 位是保留位，必须被写为 0。

图 6-34. F 口数据寄存器 (PTFD)

表 6-27. 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。



¹ 7、6、3、2 位是保留位，必须被写为 0。

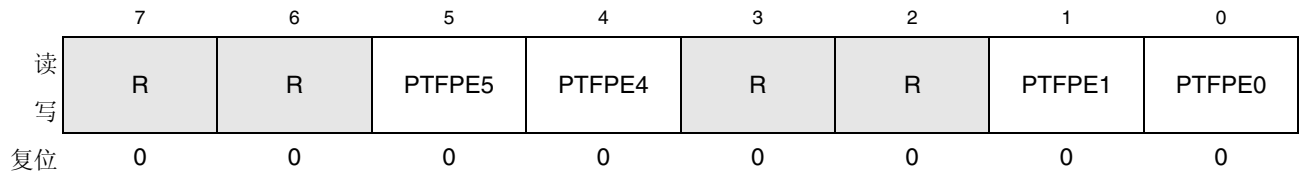
图 6-35. F 口数据方向寄存器 (PTFDD)

表 6-28. 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：



¹ 7、6、3、2 位是保留位，必须被写为 0。

图 6-36. F 口内部上拉电阻使能 (PTFPE)

表 6-29. PTFPE 寄存器域描述

域	描述
6:4, 1:0 PTFPE _n	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。

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

¹ 7、6、3、2 位是保留位，必须被写为 0。

图 6-37. F 口输出斜率控制使能寄存器 (PTFSE)

表 6-30. PTFSE 寄存器域描述

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

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

¹ 7、6、3、2 位是保留位，必须被写为 0。

图 6-38. F 口输出驱动强度寄存器 (PTFDS)

表 6-31. PTFDS 寄存器域描述

域	描述
6:4, 1:0 PTFDS _n	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：

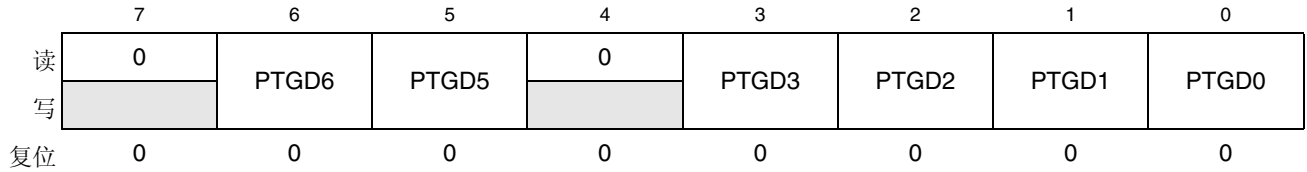


图 6-39. G 口数据寄存器 (PTGD)

表 6-32. PTGD 寄存器域描述

域	描述
6:5, 3:0 PTGD[6:5, 3:0]	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。

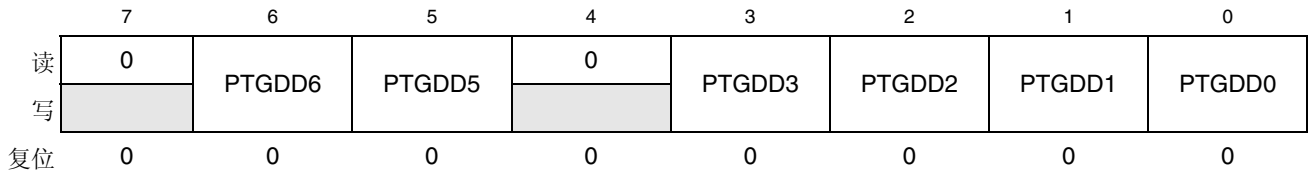


图 6-40. G 口数据方向寄存器 (PTGDD)

表 6-33. PTGDD 寄存器域描述

域	描述
6:5, 3:0 PTGDD[6:5, 3:0]	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：

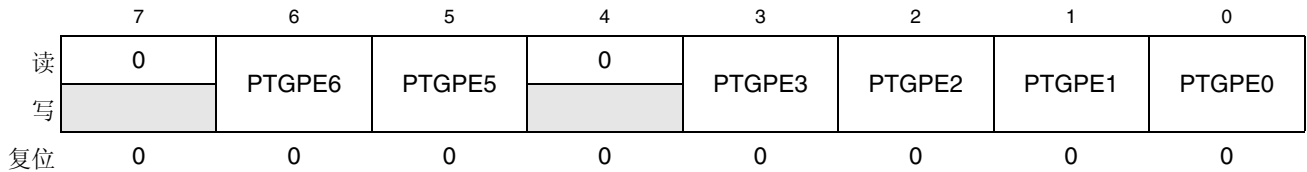


图 6-41. G 口内部上拉电阻使能 (PTGPE)

表 6-34. PTGPE 寄存器域描述

域	描述
6:5, 3:0 PTGPE[6:5, 3:0]	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。

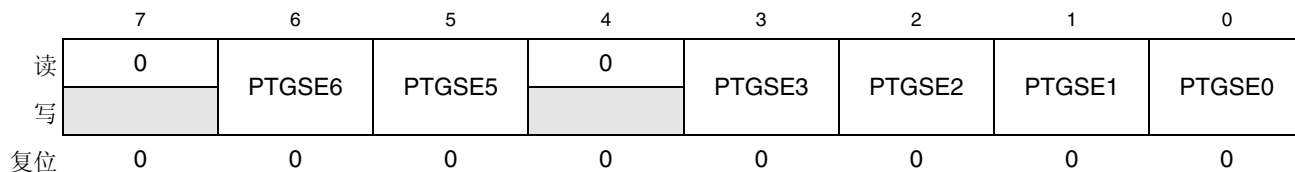


图 6-42. G 口输出斜率控制使能寄存器 (PTGSE)

表 6-35. PTGSE 寄存器域描述

域	描述
6:5, 3:0 PTGSE[6:5, 3:0]	G 口输出输出斜率使能位——每个控制位决定相应的 PTG 引脚是否允许输出输出斜率控制。当 G 口设置为输入时，这些位不起作用。 0 G 口 n 位禁止输出输出斜率控制。 1 G 口 n 位允许输出输出斜率控制。

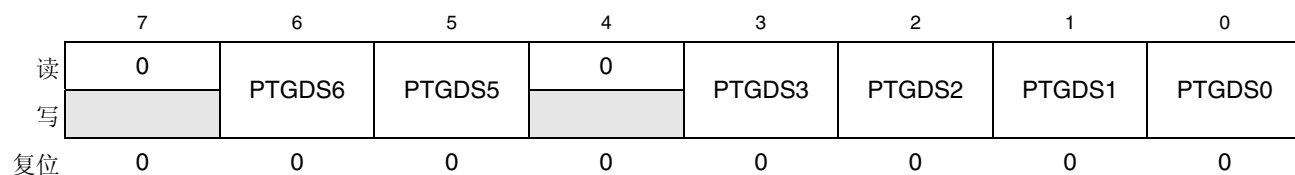


图 6-43. G 口输出驱动强度寄存器 (PTGDS)

表 6-36. PTGDS 寄存器域描述

域	描述
6:5, 3:0 PTGDS[6:5, 3:0]	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 7 章

中央处理单元（S08CPUV2）

7.1 概述

本节介绍了 HCS08 系列 CPU 的寄存器、寻址方式和指令集。更详细的信息请参考《HCS08 Family Reference Manual》，该文件序号为 HCS08RMV1/D。

HCS08 CPU 与 M68HC08 CPU 的源和目标代码完全兼容。新增的指令和增强寻址模式提高了 C 编译器的效率，并且支持新的后台调试系统以代替早期 68HC08 的监控模式。

7.1.1 特性

HCS08 CPU 具有以下特性：

- 目标代码完全兼容 M68HC05 和 M68HC08 家族
- 所有寄存器和存储器映射到一个独立的 64 KB 的地址空间
- 16 位堆栈指针（64 K 字节地址空间内任意大小、任意地址的堆栈）
- 16 位变址寄存器（H:X）支持强大的索引地址模式
- 8 位累加器（A）
- 许多指令把 X 作为第二个通用 8 位寄存器
- 7 种寻址模式：
 - 固有寻址模式 — 操作数存于内部寄存器
 - 相对寻址模式 — 8 位有符号偏移量的分支地址
 - 立即寻址模式 — 操作数位于下一个目标代码
 - 直接寻址模式 — 操作数位于 0x0000 到 0x00FF 之间
 - 扩展寻址模式 — 操作数位于 64K 字节地址空间内
 - H:X 相对变址寻址模式 — 提供包括自动增量在内的 5 种子模式
 - SP 相对变址寻址模式 — 大大提高 C 语言编译的效率
- 提供四种寻址模式组合的寄存器 - 寄存器数据转移指令
- 溢出、半进位、负、零和进位状况码支持根据带符号、无符号、BCD 码操作的结果进行条件转移
- 高效率的位操作指令
- 快速 8 位乘 8 位和 16 位除 8 位指令
- STOP 和 WAIT 指令调用低功耗运行模式

7.2 编程模型和 CPU 寄存器

图 7-1 为 5 个 CPU 寄存器，微处理器的存储器不包括 CPU 寄存器。

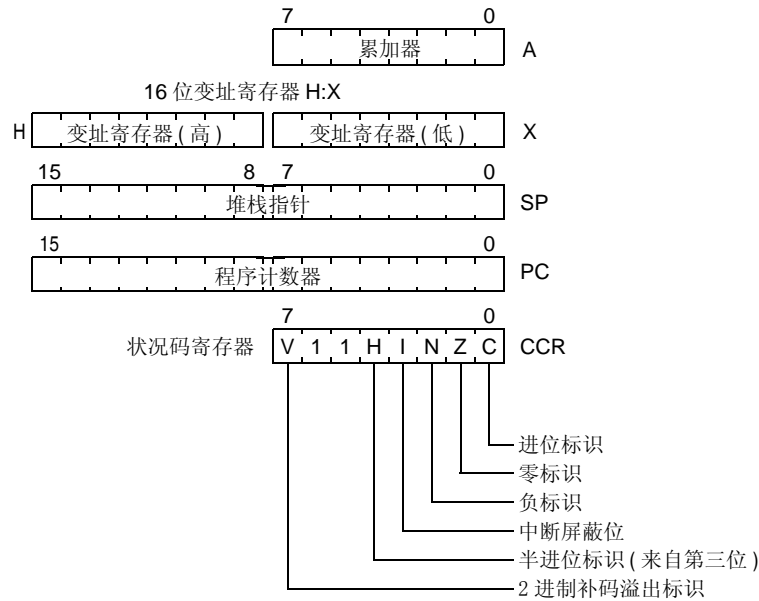


图 7-1. CPU 寄存器

7.2.1 累加器 (A)

这个累加器 (A) 是通用的 8 位寄存器。一个操作数经过累加器输入到算术逻辑单元 (ALU)，在算术和逻辑运算后，ALU 的结果通常存放入 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装载数据，或者累加器 A 中的数据可以根据不同的寻址模式指定的地址装入到存储器中。

复位对累加器中的数据无影响。

7.2.2 变址寄存器 (H:X)

这个 16 位寄存器实际上是由两个 8 位寄存器 (H,X) 组成，常一起作为一个 16 位地址指针。其中，H 装地址的高字节，X 装地址的低字节。所有变址寻址方式指令用 H:X 中的 16 位的值作为索引参考指针。但是，为了和早期 M68HC05 系列兼容，部分指令只能在低 8 位 (X) 上运行。

许多指令把 X 作为第二个通用 8 位寄存器来存储 8 位数据。X 可以被清零，增加，递减，取反，移位，循环移位。在算术逻辑操作时，转移指令用于 X 与累加器 A 进行数据交换。

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，该 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 的 64K 字节空间内的任意地址，它可以与所提供的 RAM 大小相同。堆栈可以自动保存子程序调用的返回地址，中断操作中的返回地址和 CPU 寄存器值，以及本地变量。AIS (Add Immediate to Stack pointer) 指令赋予一个 8 位有符号立即数给 SP。这种方法经常被用于为本地堆栈变量在堆栈中分配或回收空间。

为了兼容早期的 M68HC05 系列，SP 复位后被设为 0xFF。在复位初始化时，HCS08 程序通常将 SP 的值改为 RAM 最后地址值 (最高地址)，来释放直接页上的 RAM (片上寄存器结束处一直到 0x00FF)。

RSP (复位堆栈指针) 是与 M68HC05 系列兼容的指令，因为它仅影响堆栈指针的低位地址，所以很少用在 HCS08 程序中。

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、转移、中断和返回操作向程序计数器装载一个地址而不是下个连续位置。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 5 个标识最近执行指令结果的状态标志位。第 6 位和第 5 位恒为逻辑 1。下图简要描述了 CCR 每个位的信息及功能。关于如何使用指令设置 CCR 的每个位，请参阅 HCS08 系列参考手册，第 1 卷，其文档号为 HCS08RMv1。

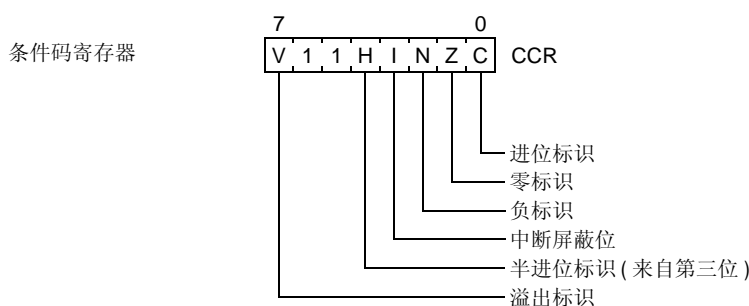


图 7-2. 条件码寄存器 (CCR)

表 7-1. CCR 寄存器位功能描述

位置	描述
7 V	二进制补码溢出标志位—当二进制补码溢出时，CPU 将设置溢出标志位。有符号指令 BGT、BGE、BLE、BLT 使用溢出标志。 0 无溢出 1 溢出
4 H	半进位标志—执行加法指令 (ADD) 和带进位加法指令 (ADC) 时，如果累加器第 3 和第 4 位有进位，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 和 C 的状态来自动调整前面 ADD 或者 ADC 的结果，修正结果得到有效的 BCD 值。 0 第 3 位和第 4 位之间无进位 1 第 3 位和第 4 位之间进位
3 I	中断屏蔽位—当中断屏蔽位被置位时，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，在第一个中断服务例程被执行之前，CPU 寄存器值被保存到堆栈后中断屏蔽被自动设置。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 的下一个指令执行时不被干扰。 0 允许中断 1 禁止中断
2 N	负标志—CPU 运算时，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。

表 7-1. CCR 寄存器位功能描述

1 Z	零标志—CPU 进行运算过程中，如果数据或运算结果为零，零标志置 1，否则置零。 0 无零结果 1 有零结果
0 C	进位 / 借位标志—当进行加法时，在最高位 D7 上有进位；或在进行减法运算时需要向更高位借位，则 CPU 将进位 / 借位标志 C 置 1。一些指令如位测试，跳转，移位指令等也会影响该标志。 0 无进位 / 借位 1 有进位 / 借位

7.3 寻址模式

寻址模式决定 CPU 存取数据和操作数的方法。在 HCS08 系列芯片中，所有的存储器、状态和控制寄存器和 I/O 口共享一个 64K 字节线性地址空间，所以一个 16 位二进制地址可以唯一确定一个存储位置。这样，访问 RAM 变量的指令同时也能访问 I/O 和控制寄存器或非易失性程序空间。

某些指令的寻址方式不止一种。例如，MOVE 指令用一种寻址方式来指定源操作数，又用另一种寻址方式来确定目标地址。BRCLR、BRSET、CBEQ、DBNZ 等指令用一种寻址方式来确定一个操作数地址，并进行测试，如果假设条件为真时，就用相对寻址方式来确定分支的目标地址。对于 BRCLR、BRSET、CBEQ、DBNZ，列在指令设置表上的寻址方式用于定位需要测试的操作数，相对寻址方式用来定位分支目标地址。

7.3.1 固有寻址模式 (INH)

固有寻址模式所需的操作数已经存于 CPU 寄存器中，所以 CPU 不需要访问存储器来获取操作数。

7.3.2 相对寻址模式 (REL)

相对寻址模式用于确定分支指令的目的地址。8 位有符号的偏移值在存储器中位于操作码之后。在执行指令时，如果分支条件为真，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，这样程序转移到分支目标地址处运行。

7.3.3 立即寻址模式 (IMM)

立即寻址模式中，用以完成指令的操作数包含在目标代码内，并紧跟着指令操作码存放在存储器中。在 16 位立即操作数情况下，高位字节存放在操作码之后的第一个存储位置，低位字节存放在下一个存储位置。

7.3.4 直接寻址模式 (DIR)

直接寻址模式中，指令包含的地址低 8 位存放在直接页 (0x0000-0x00FF)。在执行指令时，通过连接 0x00 作为高位地址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完整的 16 位地址要更快，存取效率更高。

7.3.5 扩展寻址模式 (EXT)

在扩展寻址模式中，操作数的 16 位地址位于操作码后的两字节（高位在先）。

7.3.6 变址寻址模式

变址寻址模式包含七种，其中五种使用 16 位 H:X 变址寄存器，另两种使用堆栈作为基值参考。

7.3.6.1 无偏移量变址模式 (IX)

在这种变址模式中，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、变址自动加 1 寻址模式 (IX+)

在这种变址模式中，H:X 变址寄存器存储指令的操作数。当取出地址后，H:X 寄存器自动加 1 ($H:X=H:X+0x0001$)。这种寻址模式只应用于 CBEQ 和 MOV 指令。

7.3.6.3 8 位偏移量变址模式 (IX1)

在这种变址模式中，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、变址自动加 1 寻址模式 (IX1+)

在这种变址模式中，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。在操作数被取出后变址寄存器自动加 1。CBEQ 是唯一使用这种寻址模式的指令。

7.3.6.5 16 位偏移量变址模式 (IX2)

在这种变址模式中，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 位偏移量堆栈寻址模式 (SP1)

在这种变址模式中，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 位偏移量堆栈寻址模式 (SP2)

在这种变址模式中，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 可以执行一些操作，这些操作与其他的 CPU 指令相似，只是没有操作码。另外一些指令如 STOP，WAIT 直接影响其他 MCU 的电路。本节详细将讲述这些操作。

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。当复位事件发生时，复位事件强制 CPU 立即停止正在执行的任何事件 (MCU 不会等待指令边界，再响应复位)。关于 MCU 如何确认复位及其来源的信息，可参阅复位、中断和系统配置章节。

当用以确定复位源是否来自内部的序列被执行，同时复位引脚不再有强制复位信号时，复位事件被认为结束。然后，CPU 执行一个 6 周期的序列从 0XFFFE 和 0XFFF 处取出复位向量填充到指令队列，为执行程序作好准备。

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条指令的开始位置，这也就是 CPU 中断返回地址。CPU 对中断的响应是执行与软中断 (SWI) 相同的操作序列，除非当中断序列开始时，由挂起的高优先级中断决定向量获取的地址。

CPU 中断序列如下:

1. 按顺序存储 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. 将 3 字节的程序信息 (其首地址由中断向量决定) 填入指令队列为执行中断服务例程第一条指令作准备。

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 位防止响应其它中断。虽然也可以通过指令在中断服务例程中清 I, 但是有可能导致嵌套中断 (不推荐采用这种方法, 因为这将增加程序调试和维护的难度)。

为了和早期的 M68HC05MCU 兼容, H:X 变址寄存器中的 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任何用 PSHH 指令保存 H 内容, 然后在 RTI 返回之前用 PULH 指令将 H 取出。如果能确定中断服务例程不使用任何与 H 相关的指令或自增寻址方式, H 值也可以不保存。

软中断除了不能被 CCR 中的 I 位屏蔽, 其它和硬中断一样。并且, 它和程序中的一个指令操作码相关联, 所以它不是异步程序操作。

7.4.3 等待模式

WAIT 指令通过清零 CCR 的 I 位使能中断。然后关闭 CPU 时钟以节省功耗, CPU 维持在低功耗状态直到被中断或复位唤醒。当中断或复位发生时, CPU 从等待模式被唤醒。此时, CPU 重启内部时钟, 并响应中断以及复位。

当 CPU 在等待模式时, 如果一个串行的后台调试指令通过后台调试界面发送到 MCU, CPU 时钟将被重启并进入能处理其它串行背景指令的活跃背景模式。这样, 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实现最小功耗。在这样的系统中, 需要一个外部电路来计算时间, 用于在指定的时间产生信号唤醒 MCU。和早期的 M68HC05 和 M68HC08 不同, HCS08 在停止模式中可以设定运行最少时钟。这就允许一个内部周期信号从停止模式中唤醒 MCU。

当一个主机调试系统连接到后台调试引脚 (BKGD) 且通过后台调试接口串行命令设置 ENBDM 位时 (或者 MCU 在复位时进入活跃的后台调试模式), MCU 进入停止模式时振荡器被强制活跃。在这种情况下, 当 CPU 在停止模式下通过后台调试接口向 MCU 发送后台调试指令, CPU 时钟将被开启并进入能处理其它串行背景指令的活跃背景模式。这样, 即使在停止模式中主机开发系统仍能进入目标 MCU。

从停止模式恢复的方式取决于具体的 HCS08 型号以及振荡器是否在停止模式时停止。详细信息请参阅操作模式章节。

7.4.5 背景模式

与 M68HC08 相比, HCS08 新增了 BGND 指令。BGND 通常用在普通用户程序中, 强迫 CPU 停止处理当前指令进入活跃的背景模式。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后台调试接口发出 GO、TRACE1 或者 TAGGO 串行命令。

可以用后台调试操作码替换目标断点地址操作码来设置软断点。当程序到达断点地址时，CPU 强制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

表 7-2 包含了 HCS08 所有寻址模式的指令集。本表提供了各操作数的结构、在内部总线周期的执行时间、以及每个指令在每种寻址模式的循环方式。

表 7-2. 指令集摘要 (Sheet 1 of 9)

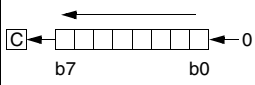
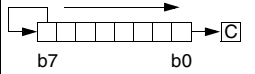
源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
ADC #opr8i ADC opr8a ADC opr16a ADC oprx16,X ADC oprx8,X ADC ,X ADC oprx16,SP ADC oprx8,SP	进位加 $A \leftarrow (A) + (M) + (C)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A9 ii B9 dd C9 hh ll D9 ee ff E9 ff F9 9E D9 ee ff 9E E9 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rpp prpp prpp	$\uparrow 1 1 \uparrow$	$- \uparrow \uparrow \uparrow$
ADD #opr8i ADD opr8a ADD opr16a ADD oprx16,X ADD oprx8,X ADD ,X ADD oprx16,SP ADD oprx8,SP	无进位加 $A \leftarrow (A) + (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AB ii BB dd CB hh ll DB ee ff EB ff FB 9E DB ee ff 9E EB ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rpp prpp prpp	$\uparrow 1 1 \uparrow$	$- \uparrow \uparrow \uparrow$
AIS #opr8i	将立即数（有符号）压入堆栈 $SP \leftarrow (SP) + (M)$	IMM	A7 ii	2	pp	- 1 1 -	- - - - -
AIX #opr8i	将立即数（有符号）压入 编制寄存器（H:X） $H:X \leftarrow (H:X) + (M)$	IMM	AF ii	2	pp	- 1 1 -	- - - - -
AND #opr8i AND opr8a AND opr16a AND oprx16,X AND oprx8,X AND ,X AND oprx16,SP AND oprx8,SP	逻辑与 $A \leftarrow (A) \& (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 ii B4 dd C4 hh ll D4 ee ff E4 ff F4 9E D4 ee ff 9E E4 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rpp prpp prpp	0 1 1 -	$- \uparrow \uparrow -$
ASL opr8a ASLA ASLX ASL oprx8,X ASL ,X ASL oprx8,SP	算术左移  (和 LSL 相同)	DIR INH INH IX1 IX SP1	38 dd 48 58 68 ff 78 9E 68 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow 1 1 -$	$- \uparrow \uparrow \uparrow$
ASR opr8a ASRA ASRX ASR oprx8,X ASR ,X ASR oprx8,SP	算术右移 	DIR INH INH IX1 IX SP1	37 dd 47 57 67 ff 77 9E 67 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow 1 1 -$	$- \uparrow \uparrow \uparrow$

表 7-2. 指令集摘要 (Sheet 2 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
BCC <i>rel</i>	进位位清零则转移 (如果 C = 0)	REL	24 rr	3	ppp	- 1 1 -	- - - - -
BCLR <i>n,opr8a</i>	存储器中位 n 清零 (Mn ← 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 dd 13 dd 15 dd 17 dd 19 dd 1B dd 1D dd 1F dd	5 5 5 5 5 5 5 5	rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp	- 1 1 -	- - - - -
BCS <i>rel</i>	进位位置 1 则转移 (如果 C = 1) (和 BLO 相同)	REL	25 rr	3	ppp	- 1 1 -	- - - - -
BEQ <i>rel</i>	相等则转移 (如果 Z = 1)	REL	27 rr	3	ppp	- 1 1 -	- - - - -
BGE <i>rel</i>	大于等于则转移 (if N ⊕ V = 0) (带符号)	REL	90 rr	3	ppp	- 1 1 -	- - - - -
BGND	ENBDM = 1 则进入活跃后台调试状态 等待和处理 BDM 指令直到 GO, TRACE1, TAGGO	INH	82	5+	fp...ppp	- 1 1 -	- - - - -
BGT <i>rel</i>	大于则转移 (如果 Z (N ⊕ V) = 0) (带符号)	REL	92 rr	3	ppp	- 1 1 -	- - - - -
BHCC <i>rel</i>	半进位清零则转移 (如果 H = 0)	REL	28 rr	3	ppp	- 1 1 -	- - - - -
BHCS <i>rel</i>	半进位位置 1 则转移 (如果 H = 1)	REL	29 rr	3	ppp	- 1 1 -	- - - - -
BHI <i>rel</i>	为高则转移 (如果 C Z = 0)	REL	22 rr	3	ppp	- 1 1 -	- - - - -
BHS <i>rel</i>	高于或相同则转移 (如果 C = 0) (和 BCC 相同)	REL	24 rr	3	ppp	- 1 1 -	- - - - -
BIH <i>rel</i>	IRQ 引脚为高则转移 (如果 IRQ pin = 1)	REL	2F rr	3	ppp	- 1 1 -	- - - - -
BIL <i>rel</i>	IRQ 引脚为低则转移 (如果 IRQ pin = 0)	REL	2E rr	3	ppp	- 1 1 -	- - - - -
BIT # <i>opr8i</i> BIT <i>opr8a</i> BIT <i>opr16a</i> BIT <i>opr16,X</i> BIT <i>opr8,X</i> BIT <i>,X</i> BIT <i>opr16,SP</i> BIT <i>opr8,SP</i>	位测试 (A) & (M) (CCR 更新但操作数不变)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 ii B5 dd C5 hh ll D5 ee ff E5 ff F5 9E D5 ee ff 9E E5 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- ↕ ↕ -
BLE <i>rel</i>	小于等于则转移 (如果 Z (N ⊕ V) = 1) (带符号)	REL	93 rr	3	ppp	- 1 1 -	- - - - -
BLO <i>rel</i>	为低则转移 (如果 C = 1) (和 BCS 相同)	REL	25 rr	3	ppp	- 1 1 -	- - - - -
BLS <i>rel</i>	为低或相同则转移 (如果 C Z = 1)	REL	23 rr	3	ppp	- 1 1 -	- - - - -
BLT <i>rel</i>	小于则转移 (如果 N ⊕ V = 1) (带符号)	REL	91 rr	3	ppp	- 1 1 -	- - - - -
BMC <i>rel</i>	中断屏蔽位清零则转移 (如果 I = 0)	REL	2C rr	3	ppp	- 1 1 -	- - - - -
BMI <i>rel</i>	负则转移 (如果 N = 1)	REL	2B rr	3	ppp	- 1 1 -	- - - - -
BMS <i>rel</i>	中断屏蔽位置 1 则转移 (如果 I = 1)	REL	2D rr	3	ppp	- 1 1 -	- - - - -
BNE <i>rel</i>	不等则转移 (如果 Z = 0)	REL	26 rr	3	ppp	- 1 1 -	- - - - -

表 7-2. 指令集摘要 (Sheet 3 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
BPL <i>rel</i>	正则转移 (如果 N = 0)	REL	2A rr	3	ppp	- 1 1 -	- - - -
BRA <i>rel</i>	一直转移 (如果 I = 1)	REL	20 rr	3	ppp	- 1 1 -	- - - -
BRCLR <i>n,opr8a,rel</i>	存储器中位 n 清零 (如果 (Mn) = 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 dd rr 03 dd rr 05 dd rr 07 dd rr 09 dd rr 0B dd rr 0D dd rr 0F dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	- 1 1 -	- - - - ↓
BRN <i>rel</i>	从不转移 (如果 I = 0)	REL	21 rr	3	ppp	- 1 1 -	- - - -
BRSET <i>n,opr8a,rel</i>	存储器中位 n 置 1 则转移 (如果 (Mn) = 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 dd rr 02 dd rr 04 dd rr 06 dd rr 08 dd rr 0A dd rr 0C dd rr 0E dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	- 1 1 -	- - - - ↓
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 dd 12 dd 14 dd 16 dd 18 dd 1A dd 1C dd 1E dd	5 5 5 5 5 5 5 5	rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp	- 1 1 -	- - - -
BSR <i>rel</i>	转移到子程序 PC ← (PC) + \$0002 push (PCL); SP ← (SP) - \$0001 push (PCH); SP ← (SP) - \$0001 PC ← (PC) + <i>rel</i>	REL	AD rr	5	ssppp	- 1 1 -	- - - -
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 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移 如果 (X) = (M) 则转移 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移	DIR IMM IMM IX1+ IX+ SP1	31 dd rr 41 ii rr 51 ii rr 61 ff rr 71 rr 9E 61 ff rr	5 4 4 5 5 6	rpppp pppp pppp rpppp rfppp prpppp	- 1 1 -	- - - -
CLC	清进位位 (C ← 0)	INH	98	1	p	- 1 1 -	- - - - 0
CLI	清中断屏蔽位 (I ← 0)	INH	9A	1	p	- 1 1 -	0 - - -
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 ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	DIR INH INH INH IX1 IX SP1	3F dd 4F 5F 8C 6F ff 7F 9E 6F ff	5 1 1 1 5 4 6	rfwpp p p p rfwpp rfwp prfwpp	0 1 1 -	- 0 1 -

表 7-2. 指令集摘要 (Sheet 4 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V I 1 H	I N Z C
CMP #opr8i CMP opr8a CMP opr16a CMP oprx16,X CMP oprx8,X CMP ,X CMP oprx16,SP CMP oprx8,SP	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 ii B1 dd C1 hh ll D1 ee ff E1 ff F1 9E D1 ee ff 9E E1 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↑ 1 1 -	- ↓ ↓ ↓
COM opr8a COMA COMX COM oprx8,X COM ,X COM oprx8,SP	求补 $M \leftarrow \overline{(M)} = \$FF - (M)$ (一个数的补) $A \leftarrow \overline{(A)} = \$FF - (A)$ $X \leftarrow \overline{(X)} = \$FF - (X)$ $M \leftarrow \overline{(M)} = \$FF - (M)$ $M \leftarrow \overline{(M)} = \$FF - (M)$ $M \leftarrow \overline{(M)} = \$FF - (M)$	DIR INH INH IX1 IX SP1	33 dd 43 53 63 ff 73 9E 63 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	0 1 1 -	- ↓ ↓ ↓ 1
CPHX opr16a CPHX #opr16i CPHX opr8a CPHX oprx8,SP	比较变址寄存器 (H:X) 和存储器 (H:X) - (M:M + \$0001) (CCR 更新但操作数不变)	EXT IMM DIR SP1	3E hh ll 65 jj kk 75 dd 9E F3 ff	6 3 5 6	prrfpp ppp rrfpp prrfpp	↑ 1 1 -	- ↓ ↓ ↓
CPX #opr8i CPX opr8a CPX opr16a CPX oprx16,X CPX oprx8,X CPX ,X CPX oprx16,SP CPX oprx8,SP	比较 X (低位变址寄存器) 和存储器 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 ll D3 ee ff E3 ff F3 9E D3 ee ff 9E E3 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↑ 1 1 -	- ↓ ↓ ↓
DAA	在 BCD 码 ADD,ADC 操作后转换累加器内容到十进制	INH	72	1	p	U 1 1 -	- ↓ ↓ ↓
DBNZ opr8a,rel DBNZA rel DBNZX rel DBNZ oprx8,X,rel DBNZ ,X,rel DBNZ oprx8,SP,rel	如果不为零, 则 A, X, 或者 M 自减并且转移 (如果 (result) ≠ 0) DBNZX 影响 X 但不影响 H	DIR INH INH IX1 IX SP1	3B dd rr 4B rr 5B rr 6B ff rr 7B rr 9E 6B ff rr	7 4 4 7 6 8	rfwpppp fppp fppp rfwpppp rfwppp prfwpppp	- 1 1 -	- - - -
DEC opr8a DECA DECX DEC oprx8,X DEC ,X DEC oprx8,SP	自减 $M \leftarrow (M) - \$01$ $A \leftarrow (A) - \$01$ $X \leftarrow (X) - \$01$ $M \leftarrow (M) - \$01$ $M \leftarrow (M) - \$01$ $M \leftarrow (M) - \$01$	DIR INH INH IX1 IX SP1	3A dd 4A 5A 6A ff 7A 9E 6A ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↑ 1 1 -	- ↓ ↓ -
DIV	除 $A \leftarrow (H:A) \div (X); H \leftarrow \text{余数}$	INH	52	6	ffffp	- 1 1 -	- - ↓ ↓
EOR #opr8i EOR opr8a EOR opr16a EOR oprx16,X EOR oprx8,X EOR ,X EOR oprx16,SP EOR oprx8,SP	带累加器的存储器异或 $A \leftarrow (A \oplus M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 ii B8 dd C8 hh ll D8 ee ff E8 ff F8 9E D8 ee ff 9E E8 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- ↓ ↓ -

表 7-2. 指令集摘要 (Sheet 5 of 9)

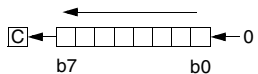
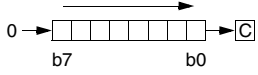
源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
INC <i>opr8a</i> INCA INCX INC <i>opr8,X</i> INC <i>,X</i> INC <i>opr8,SP</i>	自增 $M \leftarrow (M) + \$01$ $A \leftarrow (A) + \$01$ $X \leftarrow (X) + \$01$ $M \leftarrow (M) + \$01$ $M \leftarrow (M) + \$01$ $M \leftarrow (M) + \$01$	DIR INH INH IX1 IX SP1	3C dd 4C 5C 6C ff 7C 9E 6C ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow 1 1 -$	$- \downarrow \downarrow -$
JMP <i>opr8a</i> JMP <i>opr16a</i> JMP <i>opr8,X</i> JMP <i>opr16,X</i> JMP <i>,X</i>	跳转 PC ← 跳转地址	DIR EXT IX2 IX1 IX	BC dd CC hh ll DC ee ff EC ff FC	3 4 4 3 3	ppp pppp pppp ppp ppp	$- 1 1 -$	$- - - - -$
JSR <i>opr8a</i> JSR <i>opr16a</i> JSR <i>opr8,X</i> JSR <i>opr16,X</i> JSR <i>,X</i>	跳转到子程序 $PC \leftarrow (PC) + n$ ($n = 1, 2, \text{ or } 3$) Push (PCL); $SP \leftarrow (SP) - \$0001$ Push (PCH); $SP \leftarrow (SP) - \$0001$ PC ← Unconditional Address	DIR EXT IX2 IX1 IX	BD dd CD hh ll DD ee ff ED ff FD	5 6 6 5 5	ssppp pssppp pssppp ssppp ssppp	$- 1 1 -$	$- - - - -$
LDA # <i>opr8i</i> LDA <i>opr8a</i> LDA <i>opr16a</i> LDA <i>opr8,X</i> LDA <i>opr16,X</i> LDA <i>,X</i> LDA <i>opr8,SP</i> LDA <i>opr16,SP</i> LDA <i>opr8,SP</i>	从存储器装入累加器 $A \leftarrow (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 ii B6 dd C6 hh ll D6 ee ff E6 ff F6 9E D6 ee ff 9E E6 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rpp pprpp prpp	0 1 1 -	$- \downarrow \downarrow -$
LDHX # <i>opr16i</i> LDHX <i>opr8a</i> LDHX <i>opr16a</i> LDHX <i>,X</i> LDHX <i>opr8,X</i> LDHX <i>opr16,X</i> LDHX <i>opr8,X</i> LDHX <i>opr8,SP</i>	装入变址寄存器 (H:X) $H:X \leftarrow (M:M + \$0001)$	IMM DIR EXT IX IX2 IX1 SP1	45 jj kk 55 dd 32 hh ll 9E AE 9E BE ee ff 9E CE ff 9E FE ff	3 4 5 5 6 5 5	ppp rrpp prpp prpp prpp pprpp prpp prpp	0 1 1 -	$- \downarrow \downarrow -$
LDX # <i>opr8i</i> LDX <i>opr8a</i> LDX <i>opr16a</i> LDX <i>opr8,X</i> LDX <i>opr16,X</i> LDX <i>,X</i> LDX <i>opr8,SP</i> LDX <i>opr16,SP</i> LDX <i>opr8,SP</i>	从存储器中装入 X (低位变址寄存器) $X \leftarrow (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE ii BE dd CE hh ll DE ee ff EE ff FE 9E DE ee ff 9E EE ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rpp pprpp prpp	0 1 1 -	$- \downarrow \downarrow -$
LSL <i>opr8a</i> LSLA LSLX LSL <i>opr8,X</i> LSL <i>,X</i> LSL <i>opr8,SP</i>	逻辑左移  (和 ASL 相同)	DIR INH INH IX1 IX SP1	38 dd 48 58 68 ff 78 9E 68 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow 1 1 -$	$- \downarrow \downarrow \downarrow$
LSR <i>opr8a</i> LSRA LSRX LSR <i>opr8,X</i> LSR <i>,X</i> LSR <i>opr8,SP</i>	逻辑右移 	DIR INH INH IX1 IX SP1	34 dd 44 54 64 ff 74 9E 64 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow 1 1 -$	$- 0 \uparrow \uparrow$

表 7-2. 指令集摘要 (Sheet 6 of 9)

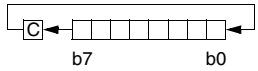
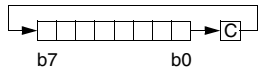
源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	移动 $(M)_{destination} \leftarrow (M)_{source}$ In IX+/DIR and DIR/IX+ Modes, $H:X \leftarrow (H:X) + \$0001$	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rpwpp rfwpp pwpp rfwpp	0 1 1 -	- $\uparrow\downarrow$ -
MUL	无符号相乘 $X:A \leftarrow (X) \times (A)$	INH	42	5	ffffp	- 1 1 0	- - - - 0
NEG <i>opr8a</i> NEGA NEGX NEG <i>opr8,X</i> NEG <i>,X</i> NEG <i>opr8,SP</i>	取负 $M \leftarrow -(M) = \$00 - (M)$ (二进制补码) $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	\uparrow 1 1 -	- $\uparrow\downarrow\downarrow$
NOP	空操作 — 使用一个总线周期	INH	9D	1	p	- 1 1 -	- - - - -
NSA	累加器半位元组交换 $A \leftarrow (A[3:0]:A[7:4])$	INH	62	1	p	- 1 1 -	- - - - -
ORA <i>#opr8i</i> ORA <i>opr8a</i> ORA <i>opr16a</i> ORA <i>opr8,X</i> ORA <i>opr8,X</i> ORA <i>,X</i> ORA <i>opr8,SP</i> ORA <i>opr8,SP</i>	累加器或存储器 $A \leftarrow (A) (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- $\uparrow\downarrow$ -
PSHA	把累加器压入堆栈 Push (A); $SP \leftarrow (SP) - \$0001$	INH	87	2	sp	- 1 1 -	- - - - -
PSHH	把 H (高位变址寄存器) 压入堆栈 Push (H); $SP \leftarrow (SP) - \$0001$	INH	8B	2	sp	- 1 1 -	- - - - -
PSHX	把 X (低位变址寄存器) 压入堆栈 Push (X); $SP \leftarrow (SP) - \$0001$	INH	89	2	sp	- 1 1 -	- - - - -
PULA	累加器出栈 $SP \leftarrow (SP + \$0001)$; Pull (A)	INH	86	3	ufp	- 1 1 -	- - - - -
PULH	H (高位变址寄存器) 出栈 $SP \leftarrow (SP + \$0001)$; Pull (H)	INH	8A	3	ufp	- 1 1 -	- - - - -
PULX	X (低位变址寄存器) 出栈 $SP \leftarrow (SP + \$0001)$; Pull (X)	INH	88	3	ufp	- 1 1 -	- - - - -
ROL <i>opr8a</i> ROLA ROLX ROL <i>opr8,X</i> ROL <i>,X</i> ROL <i>opr8,SP</i>	进位循环左移 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	\uparrow 1 1 -	- $\uparrow\downarrow\downarrow$
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 INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	\uparrow 1 1 -	- $\uparrow\downarrow\downarrow$

表 7-2. 指令集摘要 (Sheet 7 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
RSP	堆栈复位 (低位) SPL ← \$FF (不影响高字节)	INH	9C	1	p	- 1 1 -	- - - -
RTI	中断返回 SP ← (SP) + \$0001; Pull (CCR) SP ← (SP) + \$0001; Pull (A) SP ← (SP) + \$0001; Pull (X) SP ← (SP) + \$0001; Pull (PCH) SP ← (SP) + \$0001; Pull (PCL)	INH	80	9	uuuuufppp	↑ 1 1 ↓	↑ ↓ ↓ ↓
RTS	从子程序返回 SP ← SP + \$0001; Pull (PCH) SP ← SP + \$0001; Pull (PCL)	INH	81	5	ufppp	- 1 1 -	- - - -
SBC #opr8i SBC opr8a SBC opr16a SBC oprx16,X SBC oprx8,X SBC ,X SBC oprx16,SP SBC oprx8,SP	带进位减 A ← (A) - (M) - (C)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 ii B2 dd C2 hh ll D2 ee ff E2 ff F2 9E D2 ee ff 9E E2 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↑ 1 1 -	- ↓ ↓ ↓
SEC	进位位置位 (C ← 1)	INH	99	1	p	- 1 1 -	- - - 1
SEI	中断屏蔽位置位 (I ← 1)	INH	9B	1	p	- 1 1 -	1 - - -
STA opr8a STA opr16a STA oprx16,X STA oprx8,X STA ,X STA oprx16,SP STA oprx8,SP	将累加器中内容存储到存储器 M ← (A)	DIR EXT IX2 IX1 IX SP2 SP1	B7 dd C7 hh ll D7 ee ff E7 ff F7 9E D7 ee ff 9E E7 ff	3 4 4 3 2 5 4	wpp pwpp pwpp wpp wp ppwpp pwpp	0 1 1 -	- ↓ ↓ ↓ -
STHX opr8a STHX opr16a STHX oprx8,SP	存储 H:X (变址寄存器) (M:M + \$0001) ← (H:X)	DIR EXT SP1	35 dd 96 hh ll 9E FF ff	4 5 5	wwpp pwpp pwpp	0 1 1 -	- ↓ ↓ ↓ -
STOP	中断使能 停止处理 参考 MCU 文档 I bit ← 0; Stop Processing	INH	8E	2	fp...	- 1 1 -	0 - - -
STX opr8a STX opr16a STX oprx16,X STX oprx8,X STX ,X STX oprx16,SP STX oprx8,SP	存储 X (变址寄存器低 8 位) 到存储器 M ← (X)	DIR EXT IX2 IX1 IX SP2 SP1	BF dd CF hh ll DF ee ff EF ff FF 9E DF ee ff 9E EF ff	3 4 4 3 2 5 4	wpp pwpp pwpp wpp wp ppwpp pwpp	0 1 1 -	- ↓ ↓ ↓ -

表 7-2. 指令集摘要 (Sheet 8 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
SUB #opr8i SUB opr8a SUB opr16a SUB oprx16,X SUB oprx8,X SUB ,X SUB oprx16,SP SUB oprx8,SP	减 A ← (A) – (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 ii B0 dd C0 hh ll D0 ee ff E0 ff F0 9E D0 ee ff 9E E0 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp prpp prpp	↑ 1 1 -	- ↓ ↓ ↓
SWI	软中断 PC ← (PC) + \$0001 Push (PCL); SP ← (SP) – \$0001 Push (PCH); SP ← (SP) – \$0001 Push (X); SP ← (SP) – \$0001 Push (A); SP ← (SP) – \$0001 Push (CCR); SP ← (SP) – \$0001 I ← 1; PCH ← 高字节中断矢量 PCL ← 低字节中断矢量	INH	83	11	sssssvvfppp	- 1 1 -	1 - - -
TAP	转移累加器到 CCR CCR ← (A)	INH	84	1	p	↑ 1 1 ↓	↑ ↓ ↓ ↓
TAX	转移累加器到 X (低位变址寄存器) X ← (A)	INH	97	1	p	- 1 1 -	- - - -
TPA	转移 CCR 到累加器 A ← (CCR)	INH	85	1	p	- 1 1 -	- - - -
TST opr8a TSTA TSTX TST oprx8,X TST ,X TST oprx8,SP	测试零或负数 (M) – \$00 (A) – \$00 (X) – \$00 (M) – \$00 (M) – \$00 (M) – \$00	DIR INH INH IX1 IX SP1	3D dd 4D 5D 6D ff 7D 9E 6D ff	4 1 1 4 3 5	rfpp p p rfpp rfp prfpp	0 1 1 -	- ↓ ↓ -
TSX	转移 SP 到变址寄存器. H:X ← (SP) + \$0001	INH	95	2	fp	- 1 1 -	- - - -
TXA	转移 X (低位变址寄存器) 到累加器 A ← (X)	INH	9F	1	p	- 1 1 -	- - - -

表 7-2. 指令集摘要 (Sheet 9 of 9)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
TXS	转移变址寄存器到 SP SP ← (H:X) - \$0001	INH	94	2	fp	- 1 1 -	- - - -
WAIT	中断使能, 等待中断 I bit ← 0; Halt CPU	INH	8F	2+	fp...	- 1 1 -	0 - - -

源格式栏中, 除了用斜体表示的表达式外, 其它信息必须以相同的格式出现在汇编源文件中。原始的 3 至 5 个助记符是一种文字表达。所有的逗号, 井号 (#), 圆括号, 加号 (+) 都是文字字符。

n — 任何估算 0-7 之间的单整形数的标识或表达式

opr8i — 任何估算 8 位立即数值的标识或表达式

opr16i — 任何估算 16 位立即数值的标识或表达式

opr8a — 任何估算 8 位直接页地址 (0x00XX) 的标识或表达式。

opr16a — 任何估算 16 位地址的标识或表达式。

opr8 — 任何估算无符号 8 位立即数值的标识或表达式, 用于变址寻址

opr16 — 任何用来估值 16 位立即数值的标识或表达式, 用于变址寻址

rel — 任何标识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。

操作符:

A 累加器
CCR 条件码寄存器
H 变址寄存器高 8 位
M 寄存器地址
n 任何位
opr 操作数 (1 或 2 字节)
PC 程序计数器
PCH 程序计数器高 8 位
PCL 程序计数器低 8 位
rel 相对程序计数器偏移字节
SP 堆栈指针
SPL 堆栈指针低 8 位
X 变址寄存器低 8 位
& 布尔与
| 布尔或
⊕ 布尔异或
() 寄存器或存储器内容显示在圆括号里
+ 加
- 减
x 乘
÷ 除
立即值
" 被装入 (读: 获得)
: 连接

条件寄存器 (CCR) 位

V 溢出位
H 半进位位
I 中断屏蔽标识
N 负标识位
Z 零标识
C 进位 / 借位标识

寻址模式:

DIR 直接寻址模式
EXT 扩展寻址模式
IMM 立即寻址模式
INH 内在寻址模式
IX 无偏移量变址模式
IX 1 8 位偏移量变址模式
IX 2 16 位偏移量变址模式
IX+ 无偏移量变址、变址加 1 寻址模式
IX1+ 8 位偏移量变址、变址加 1 寻址模式
REL 相对寻址模式
SP1 8 位偏移量堆栈寻址模式
SP2 16 位偏移量堆栈寻址模式

循环码:

f 自由周期。在自由周期, CPU 不需要使用系统总线, 一个自由周期通常是系统总线时钟的一个周期并且通常是一个只读周期。
p 程序取回; 从程序寄存器的下个连续地址读取。
r 读 8 位操作码
s 把一个字节推入 (写入) 堆栈
u 从堆栈弹出 (读出) 一个字节
v 从 0xFFxx 读矢量 (从高字节开始)
w 写 8 位操作码

CCR 常用符号:

↓ 设置或清零
- 不影响
U 未定义

表 7-3. Opcode Map (Sheet 1 of 2)

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

INH 隐含寻址 **REL** 相对寻址 **SP1** 8 位偏移量堆栈寻址
IMM 立即寻址 **IX** 无偏移量变址寻址 **SP2** 16 位偏移量堆栈寻址
DIR 直接寻址 **IX1** 8 位偏移量变址寻址 **IX+** 无偏移量自动加 1 变址寻址
EXT 扩展寻址 **IX2** 16 位偏移量变址寻址 **IX+D** 变址 - 直接 - 变址加 1 寻址
DD 直接 - 直接寻址 **IMD** 立即 - 直接寻址 **IX1+** 8 位偏移量自动加 1 变址寻址
DIX+ 直接 - 自动加 1 变址寻址

用 16 进制表示
 操作码 F0 3
 字节数 1 SUB IX HCS08 指令周期
 助记寻址方式

表 7-3. Opcode Map (Sheet 2 of 2)

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

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
 DIX+ 直接 - 自动加 1 变址寻址

REL 相对寻址
 IX 无偏移量变址寻址
 IX1 8 位偏移量变址寻址
 IX2 16 位偏移量变址寻址
 IMD 立即 - 直接寻址

SP1 8 位偏移量堆栈寻址
 SP2 16 位偏移量堆栈寻址
 IX+ 无偏移量自动加 1 变址寻址
 IX+D 变址 - 直接 - 变址加 1 寻址
 IX1+ 8 位偏移量自动加 1 变址寻址

用 16 进制表示
 操作码 和前置位 (9E)

9E60 6 NEG 3 SP1	HCS08 指令周期 助记寻址方式
------------------------	----------------------

字节数

第 8 章

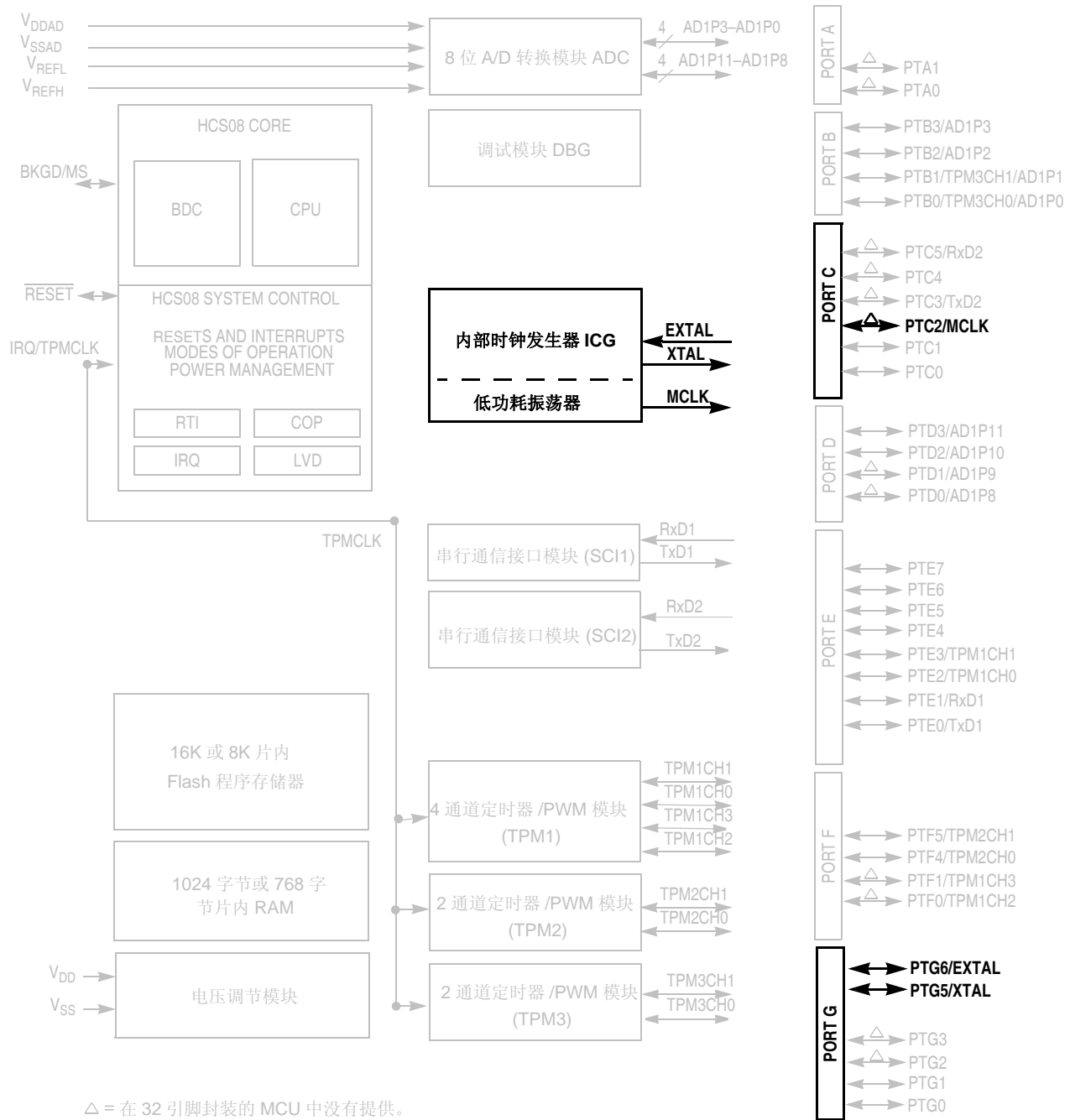
内部时钟发生器 (S08ICGV4)

内部时钟发生器模块 (ICG) 为 SC667027 系列单片机提供系统时钟。其模拟供电线路 V_{DD} 和 V_{SS} 源自 MCU 的 V_{DD} 和 V_{SS} 的引脚。ICG 的电气参数可参考附录 A 电气特性和时序规范。

注意

Freescale 半导体公司建议 flash 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，以便用户访问。

SC667027 的内部总线频率最高可达 10 MHz。本章中描述为最高 20 MHz 适用于飞思卡尔 8 位微控制器的其他产品。



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 8-1. 显亮 ICG 模块和引脚的 SC667027 模块结构图

8.1 概述

ICG 提供多个时钟源。这为用户在均衡成本、精度、电流消耗量以及性能时，提供了极大的灵活性，如图 8-2 所示，ICG 由四个功能模块组成。下面将对这些模块进行简要的描述，详细描述请参见后续章节。

- 振荡器模块 — 振荡器模块用于连接外部晶振或振荡器。它提供两种可软件选择的频率范围，以获得最佳的启动和稳定性。振荡器模块可以产生一个外部方波给系统时钟。外部时钟源可以提供非常精确的时钟。HGO 可以配置振荡器为低功耗模式或高幅模式。
- 内部辅助发生器 — 内部辅助发生器由两个受控时钟源组成。一个为大约 8 MHz 作为背景调试控制器的本地时钟。另一个时钟源是典型的 243 kHz，当一个精确的定时事件输入到 MCU 时，它可通过软件进行微调 (Trim) 以提高其精确度。这将提供一个非常可靠，低成本时钟源。
- 锁频环 — 对内部或外部时钟源进行倍频的时候，需要用到锁频环。当电路频率锁定或失锁时，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信号是否有效。
- 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 FLL 的输出倍频时钟，ICGERCLK 是源自晶振或外部时钟源的参考时钟频率，FFE(固定频率使能)是一个用来控制系统固定频率时钟 (XCLK) 的控制信号。ICGLCLK 是背景调试控制器的时钟源。

8.1.1 特性

该模块非常易用，许多功能无需用户干预自动实现。若需要快速配置此模块，请参考 8.5 节 初始化 / 应用信息，并选取一个与应用相似的例子。

ICG 和时钟分配系统的特性如下：

- 几种可选主时钟源提供灵活的成本，频率和精确度选择。
 - 32 kHz ~ 100 kHz 的晶振或振荡器
 - 1 MHz ~ 16 MHz 的晶振或振荡器
 - 外部时钟
 - 内部参考发生器
- 缺省设置为自时钟模式以获得最小的启动延时
- 频率锁定回路 (FLL) 产生 8 MHz ~ 40 MHz 时钟 (总线时钟最高至 20 MHz) — 用外部或内部时钟作参考频率。
- 自动停止非工作状态的时钟源。
- 丢失时钟或者 FLL 失锁会产生复位或中断
- 数字控制振荡器 (DCO) 保留先前的频率设定，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。
- 后 FLL 分频器选择 8 个总线分频约数 (/1 到 /128) 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。
- 锁获得后自动 FLL 接触。
- 外部振荡器可选择低功率或高增益

8.1.2 操作模式

本节对操作模式进行简单描述。详细描述参见 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)

模式 2 是缺省模式，在复位后立即进入此模式。内部时钟发生器的 FLL 是开环的，数字控制振荡器自由运行在设置的频率。

- 模式 3 — FLL 参与模式 (FEI)

在此模式下，FLL 对内部时钟源进行可编程倍频来产生频率。

- 当 FLL 试图上锁时，未上锁的 FEI 是一个暂态。此时，FLL 的数字控制振荡器 (DCO) 频率偏离目标频率，而 FLL 调整数字控制振荡器以达到目标频率。
- 当 FLL 检测到数字控制振荡器锁定为目标倍频频率时，FLL 将上锁。

- 模式 4—FLL 外部旁路

在这种模式下，ICG 被配置为旁路 FLL，而使用外部时钟作为系统时钟源。

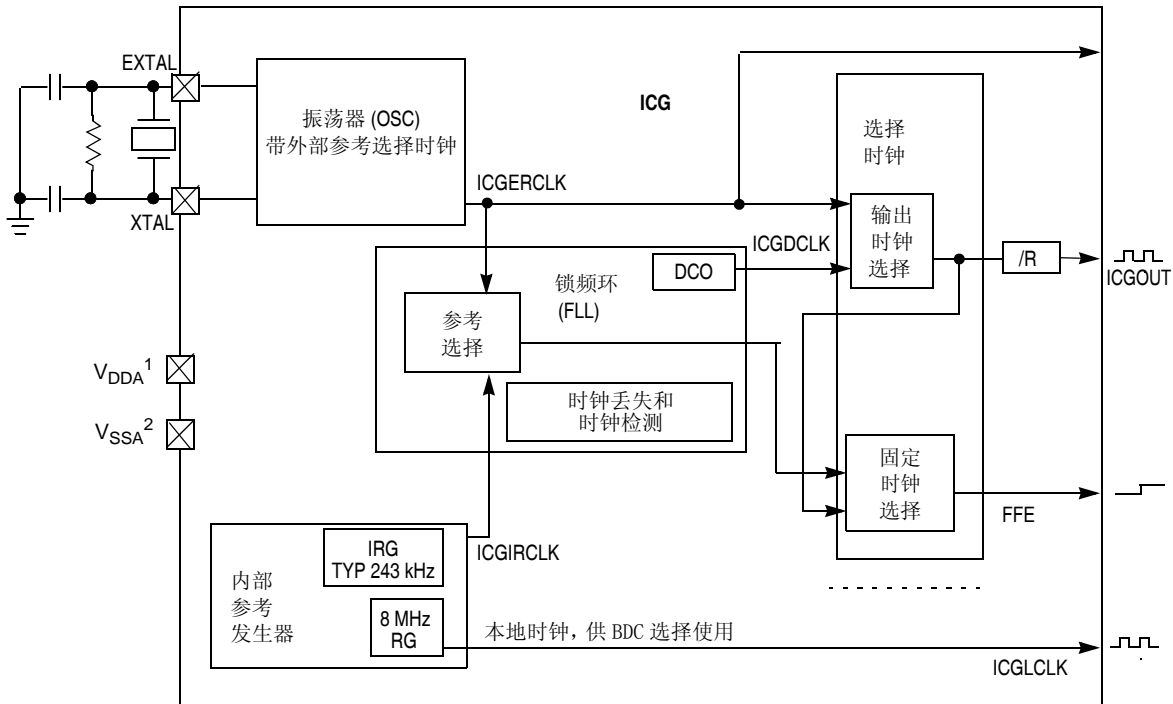
- 模式 5—外部使用的 FLL (FEE)

内部时钟发生器的 FLL 对外部时钟源进行倍频

- 当 FLL 试图上锁时，未上锁的 FEI 是一个暂态。此时，FLL 的数字控制振荡器 (DCO) 频率偏离目标频率，而 FLL 调整数字控制振荡器以达到目标频率。
- 当 FLL 检测到数字控制振荡器锁定为目标倍频频率时，FLL 将上锁。

8.1.3 功能结构图

图 8-2 是内部时钟发生器 (ICG) 模块的功能结构图。



¹ ICGOUT、FFE、ICGCLK、ICGERCLK 具体使用方法参阅表 8-1

² 不是所有的 HCS08 芯片都含有 ICG 单独的电源引脚。参阅设备引脚分配。

图 8-2. 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（缺省）其增益受控。在低功耗模式下振荡器振幅限制为约 1 V（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，无论选择 FEE 模式或者 FBE 模式，该引脚将由 REFS 决定作为外部时钟输入或振荡器电路输入。若是第一次写 ICGC1，无论选择 FEI 模式或者 SCM 模式，该引脚都不被 ICG 使用。

8.2.2 XTAL—振荡器输出

若是第一次写 ICGC1，无论选择 FEE 模式或者 FBE 模式，该引脚将被作为振荡器电路输出。若是第一次写 ICGC1，无论选择 FEI 模式或者 SCM 模式，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

若使用外部时钟，则引脚连接如图 8-3 所示。

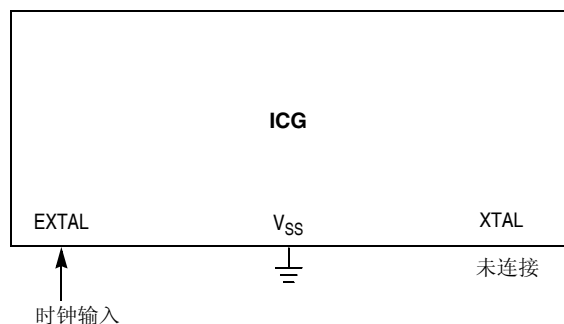


图 8-3. 外部时钟连接

8.2.4 外部晶振 / 谐振器连接

如果使用外部晶振 / 谐振器，那么引脚可按图 8-4 连接。推荐的元件值被列在“电气特性”章节。

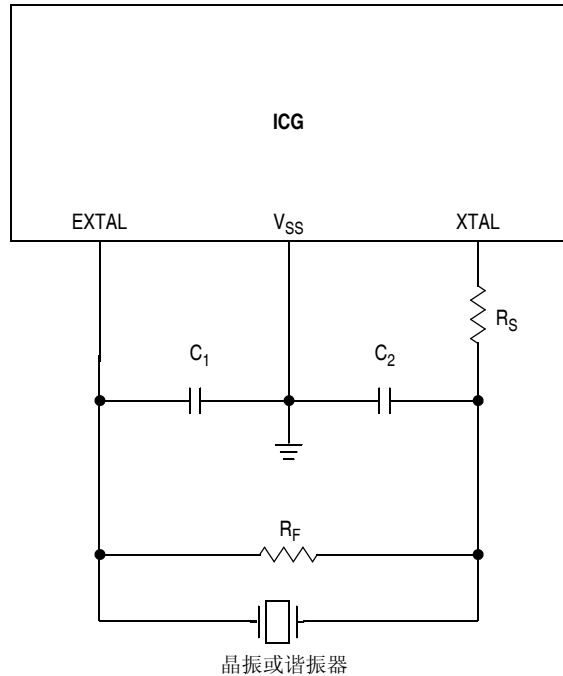
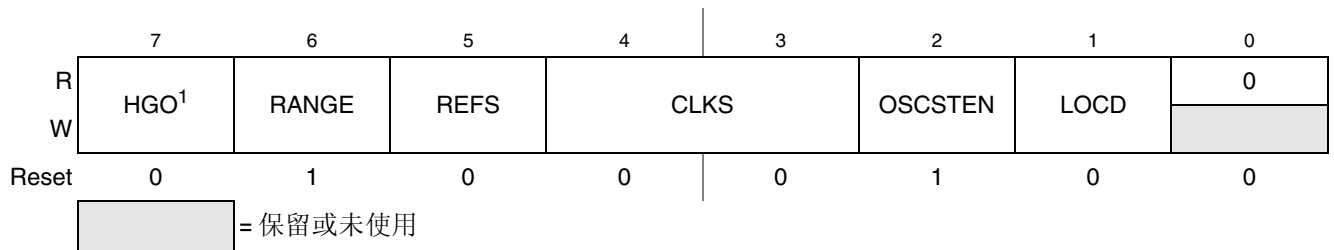


图 8-4. 使用外部参考频率的连接

8.3 寄存器定义

参阅存储器节的直接页寄存器，可以找到所有 ICG 寄存器绝对地址分配表。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)



¹ 该位在复位后仅可写 1 次。之后的写操作被忽略。

图 8-5. ICG 控制寄存器 1 (ICGC1)

表 8-1. ICGC1 寄存器位描述

位	描述
7 HGO	高增益振荡器的选择位—该位用来选择低功耗操作或高增益操作以降低噪声影响。该位复位后只可写一次。 0 振荡器配置为低功耗操作 1 振荡器配置为高增益操作

表 8-1. ICGC1 寄存器位描述

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 选择频率范围。该位复位后只可写一次。该位仅在外使用 FLL(FEE) 和 FLL 外部旁路 (FBE) 模式有效。 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 REFS	外部参考源选择—该位控制 ICGERCLK 的外部参考时钟源。该位复位后只可写一次。 0 外部时钟请求 1 振荡器使用晶振或振荡器请求
4:3 CLKS	时钟模式选择—该位控制时钟模式。如果请求 FLL 外部旁路模式 (FBE)，除非 ERCS=1，否则该位将不能被选择。如果 ICG 关闭，该位将保持不变。如果先前写未完成，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 OFF 模式下振荡器使能位—当 ICG 关闭时，该位控制振荡器电路是否保持使能。如果 HGO = 1 并且 RANGE = 1，该位无效。 0 当 ICG 处在 OFF 模式时，振荡器禁止，除非使能位为高，CLKS = 10 并且 REFST = 1。 1 当 ICG 处于 OFF 模式时，振荡器被使能，CLKS = 1X 并且 REFST = 1。
1 LOCD	禁用时钟丢失检测 0 时钟丢失检测允许 1 时钟丢失检测禁止

8.3.2 ICG 控制寄存器 2 (ICGC2)

	7	6	5	4	3	2	1	0
R								
W	L	L	M	M	L	L	R	R
Reset	0	0	0	0	0	0	0	0

图 8-6. ICG 控制寄存器 2 (ICGC2)

表 8-2. ICGC2 寄存器位描述

位	描述
7 LOLRE	失锁复位使能位—该位决定当失锁发生的时候，ICG 发送的请求。只有当 LOLS 被置位时，该位才有效。 0 失锁产生一个中断请求 1 失锁产生一个复位请求
6:4 MFD	倍频因子—该位控制环路 FLL 中可编程倍频因子。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以确保 f_{ICGCLK} 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

表 8-2. ICGC2 寄存器位描述

3 LOCRE	时钟丢失复位使能—该位决定系统如何管理一个时钟丢失。 0 时钟丢失产生一个中断请求 1 时钟丢失产生一个复位请求
2:0 RFD	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定的输出时钟源。如果先前的写操作未完成，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)

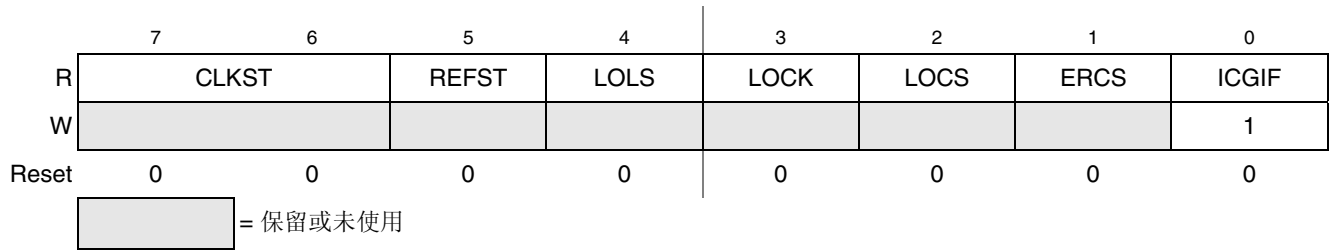


图 8-7. ICG 状态寄存器 1 (ICGS1)

表 8-3. ICGS1 寄存器位描述

位	描述
7:6 CLKST	时钟模式状态位—该位标识当前时钟模式。在写 CLKS 位后由于需要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—该位标识时钟参考选择电路当前使用的时钟参考源。 0 选择外部时钟 1 选择晶振 / 振荡器
4 LOLS	FLL 失锁状态—该位是 FLL 锁状态的关联指示。 0 自从 LOCS 最近一次被清零后，FLL 没有意外丢失锁。 1 自从 LOCS 最近一次被清零后，FLL 出现意外失锁，LOLRE 决定将采取的行动。
3 LOCK	FLL 锁状态—该位标识 FLL 是否已获锁定。在关闭、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—该位标识 ICG 时钟状态丢失。 0 自从 LOCS 最近一次被清零后，ICG 没有丢失时钟。 1 自从 LOCS 最后一次被清零后，ICG 出现时钟丢失，LOCRES 决定后续采取的行动。

表 8-3. ICGS1 寄存器位描述

位	描述
1 ERCS	外部参考时钟状态—该位标识外部参考时钟 (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 中断标识—当出现挂起的 ICG 中断请求时, ICGIF 读/写标识被置位。当 ICGIF 被置位时可通过复位或读取 ICG 状态寄存器, 并向 ICGIF 位写 1 来清零。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, 因此在清零上一个中断的序列后, ICGIF 还将保持原设置。写逻辑 0 给 ICGIF 无影响。 0 无挂起的 ICG 中断请求 1 有一个 ICG 中断请求挂起

8.3.4 ICG 状态寄存器 2 (ICGS2)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	DCOS
W								
Reset	0	0	0	0	0	0	0	0


 = 保留或未使用

图 8-8. 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 (ICG2DCLK) 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 nunlock 所改变, 并且该 DCO 时钟不是静止的。当退出 OFF 模式, 如果 CLKS=X1, 该位用来决定何时转换到请求的时钟模式。它同时也用在自时钟模式来决定何时开始监视 DCO 时钟。进入 OFF 模式该位被清零。 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
R	0	0	0	0	FLT			
W								
Reset	0	0	0	0	0	0	0	0

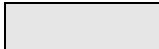
 = 保留或未使用

图 8-9. 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—表明当前滤波器值，该值控制 DCO 频率。除了自时钟模式（CLKS = 00），该位可读。在自时钟模式下任何写 ICGFLTU 将更新当前 12 位滤波器值。如果前一个关锁序列未完成，写 ICGFLTU 寄存器不会影响 FLT。

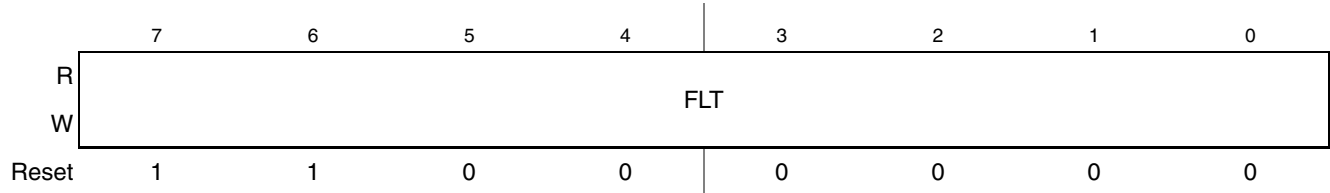
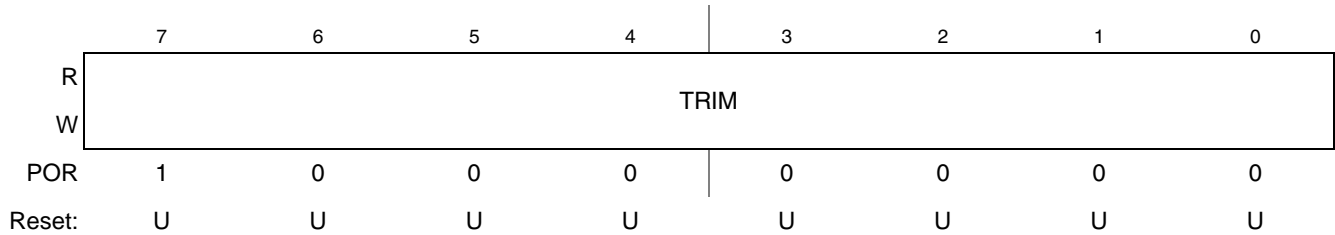


图 8-10. ICG 滤波寄存器低八位 (ICGFLTU)

表 8-6. ICGFLTU 寄存器位描述

位	描述
7:0 FLT	滤波器值—表明当前滤波器值，该值控制 DCO 频率。除了自时钟模式（CLKS=00），该位可读。在自时钟模式下任何写 ICGFLTU 将更新当前 12 位滤波器值。如果前一个关锁序列未完成，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)



U = 不受 MCU 复位影响

图 8-11. ICG 调整寄存器 (ICGTRM)

表 8-7. ICGTRM 寄存器位描述

位	描述
7: 0 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。该位通过二进制加权改变周期（例如，位 1 调整是位 0 调整的两倍）。在调整时，增加其二进制值将会增长周期，反之，减少值将缩短周期。

8.4 功能描述

本节将描述 ICG 五个工作模式的功能。同时讨论时钟丢失和失锁错误及进入每个模式的必要条件。ICG 非常具有灵活性，在某些配置下它可能超越某个时钟规定。当使用 FLL 时，配置 ICG，则 ICGDCLK 的频率不会超出它的最大值，确保 MCU 正常工作。

8.4.1 OFF 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 OFF 模式。当然，在 STOP 模式下，有两种情况时钟继续工作。

8.4.1.1 活跃 BDM

当 BDM 允许时，ICG 继续根据最初设置运行。这将允许通过 BDC 控制器访问存储器和控制寄存器。

8.4.1.2 OSCSTEN 位设置

当在停止模式下允许振荡器时（OSCSTEN=1），除了单独的时钟发生器工作外，MCU 其他部分所需的时钟源都被关闭。这种方式避免了冗长的振荡器启动时间，或在停止模式 3 下使用振荡器来运行 RTI。

8.4.1.3 停止 /OFF 模式恢复

如果中断从停止模式唤醒 CPU，则先前设置的控制位有效，且系统时钟恢复工作。如果是 FEE 模式，ICG 将以内部参考时钟为源时钟直到外部时钟稳定。如果是 FBE 模式，在使能 ICGOUT 之前 ICG 将等待外部时钟稳定。

如果由复位从停止模式唤醒 CPU，先前设置的 ICG 控制位将被忽略，而使用缺省的复位值。因此退出 Stop 停止模式后，ICG 进入 SCM 模式，并被配置为一个大约 8MHz 的 DCO 输出。如果使用外部晶振，在使能 ICGERCLK 之前，将先检测 4096 个时钟信号。这包含在晶振的启动时间里。

8.4.2 自时钟模式（SCM）

自时钟模式（SCM）是缺省的操作模式，当以下条件中任何一个发生时就将进入 SCM 模式：

- 任何复位。
- 当 CLKS 不等于 10 时，从 OFF 模式退出。如果 CLKS = X1，ICG 将临时进入该状态直到 DCO 稳定（DCOS=1）。
- CLKS 由 X1 改写到 00
- CLKS = 1X 且 ICGERCLK 未被检测到（ERCS = 0、LOCS = 1）。

在这模式下，FLL 环路是敞开的。DCO 正在运行，其输出时钟信号 ICGOUT 频率为 $f_{ICGDCLK} / R$ 。ICGDCLK 频率在 8 MHz 到 40 MHz 之间变化，由滤波器寄存器（ICGFLTH 和 ICGFLTL）决定。滤波器寄存器仅在此模式下可写。

如果由于复位进入此模式， $f_{ICGDCLK}$ 将缺省为 8 MHz。如果从 FEI 模式进入此模式， $f_{ICGDCLK}$ 将保持先前的频率，但如果从 FEE 模式进入（不管是设置 CLKS，还是外部参考信号丢失）， $f_{ICGDCLK}$ 将保持先前的频率，而如果 FLL 失锁，ICGOUT 将变为其两倍。如果从 OFF 模式 (OFF) 进入此模式，在进入 OFF 模式前 $f_{ICGDCLK}$ 将等于 ICGDCLK 的频率。如果在退出 OFF 模式时 CLKS 位被置为 01 或 11，那么直到 ICGDCLK 稳定 (DCOS=1)，ICG 才进入自时钟模式。在 ICGDCLK 稳定后，ICG 自动关闭环路，并切换到 FLL 参与的模式，由 CLKS 决定内部信号或外部信号使用。

8.4.5 FLL 内部锁定

对于给定数目的样本，如果减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，将从 FEI 失锁进入本状态。此时时钟信号输出 ICGOUT 频率等于 f_{ICGDCLK}/R 。在锁定的 FEI 模式下，滤波器值仅在每四个比较周期更新一次。更新的值为前四次比较取样误差的平均值。

8.4.6 FLL 旁路外部时钟模式 (FBE)

当下列任何一种条件发生时进入 FLL 旁路外部时钟模式：

- 当 $\text{CLKS} = 10$ 且 ERCS 为高时从 SCM 进入
- 当 $\text{CLKS} = 10$ 时， $\text{ERCS} = 1$ 进入 OFF 模式，然后退出 OFF 模式
- 如果 DCO 时钟丢失，而外部时钟参考保持有效 ($\text{LOCS} = 1$ 、 $\text{ERCS} = 1$) 从 FLL 外部时钟模式进入

在这种模式下，DCO 和 IRG 关闭，参考时钟为外部参考时钟 ICGERCLK。该输出时钟信号 ICGOUT 频率为 f_{ICGDCLK}/R 。如果使用外部时钟源 ($\text{REFS} = 0$)，那么 EXTAL 引脚上的输入频率范围：0 MHz ~ 40 MHz。如果使用晶振或振荡器，那么频率范围可为低频范围 ($\text{RANGE} = 0$) 或为高频范围 ($\text{RANGE} = 1$)。

8.4.7 FLL 外部时钟模式 (FEE)

当下列任何一种条件发生时进入 FEE 模式：

- $\text{CLKS} = 11$ 并且 ERCS 和 DCOS 都为高
- 当退出 OFF 模式时， $\text{CLKS} = 11$ 。并进入自时钟模式，DCO 时钟稳定 ($\text{DCOS} = 1$)

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 频率以达到预设值。为了在 FEE 模式下运行，必须有一个 32 kHz ~ 100 kHz 或 2 MHz ~ 10 MHz 的外部时钟源。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 的最小倍频因子从表 8-12 可知是 4。因为 $4 \times 10 \text{ MHz}$ 是 40 MHz，这是 DCO 的运行的极限，所以参考时钟不能超过 10 MHz。

8.4.7.1 FLL 外部失锁

当进入 FEE 模式，对于给定数目的样本，减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，将进入失锁的 FEE 模式。

减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，ICG 将一直处于本模式。

在这种状态下，脉冲计数器、减数器、数字环路滤波器，以及 DCO 组成一个回路，根据其运行条件 (本章后面的部分将进行讲解) 试图进行锁定。从进入该状态，到锁定 FLL，ICGOUT 的频率为 $f_{\text{ICGDCLK}}/(2 \times R)$ 。这额外的除 2，可以防止在锁定过程中，频率超过范围。在成功锁定 FLL 后，如果在 FEE 模式下，发生意外失锁，而导致重新进入该状态，此时 ICGOUT 的输出频率为 f_{ICGDCLK}/R 。

8.4.7.2 FLL 外部锁定

对于给定数目的样本，当减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，就从 FEE 未锁定模式进入 FEE 锁定模式。此时时钟信号输出 ICGOUT 频率为 f_{ICGDCLK}/R 。在 FLL 外部锁定模式，仅在每四个比较周期，采用前四次比较取样误差的平均值更新滤波器。

8.4.8 FLL 锁定和失锁检测

为决定 FLL 锁定和失锁条件，脉冲计数器累加 DCO 一个比较周期的脉冲数 (见表 8-9 比较周期的解释)，并把该值传给减法器。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 MCU 复位。LOLS 清除方法: 读 ICGS1 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 ICG 由停止模式进入 OFF 模式, 则进入 FLL 失锁状态 (LOCK 被清零), 但 LOLS 保持不变因为这不是一个意外的失锁条件。如果在停止模式下, ENBDM 被清为 0, ICG 进入 OFF 模式, 尽管这不太常见。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, 将进入失锁状态。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

监控参考时钟和 DCO 时钟的条件不同 (见表 8-8)。假定参考频率正被监控, 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 (分别为 f_{LOR} 和 f_{LOD}), LOLS 将通过被置位来指示错误。LOCS 将保持置位直到它被清除或直到 MCU 复位。通过读 ICGS1 然后写 1 给 ICGIF (LOCRES=0) 可清除 LOCS, 或通过时钟丢失引起复位, 或通过任意 MCU 复位。

如果 ICG 处于 FEE 模式, 参考时钟的丢失将导致 ICG 进入 SCM 模式, DCO 时钟丢失将导致 ICG 进入 FBE 模式。如果 ICG 处于 FBE 模式下, 参考时钟的丢失将导致 ICG 进入 SCM 模式。在上述情况下, CLKST 和 CLKS 位将被自动改变以反映当前状态。

如果时钟丢失发生在 FEE 模式, 同时 ERCS 仍为 1, 那么 CLKST 将被设置为 10, 同时 ICG、转换到 FBE 模式。

在 FEE 或 FEI 模式, 时钟丢失将导致失锁。因为清 LOCS 和 LOLS 位的方法相同, 极少出现的 LOLRE = 1 但是 LOCRES = 0 将是一个问题。在这种情形下, 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?
OFF 模式	0X 或 11	X	强制低	无	无
	10	0	强制低	无	无
	10	1	实时 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	有	有 ²
	11	X	实时	有	有 ²
FEI (CLKST=01)	0X	X	强制低	无	有
	11	X	实时	有	有

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?
FBE (CLKST=10)	10	0	强制高	无	无
	10	1	实时	有	无
FEE (CLKST=11)	11	X	实时	有	有

¹ 如果 ENABLE 为高 (在退出停止模式后等待外部晶振启动)。

² 从 OFF 模式或 FLL 旁路外部模式进入 SCM 模式, 在 DCO = 1 之前, DCO 时钟将不被监控。

8.4.10 时钟模式必要条件

通过写 CLKS1:CLKS0 请求时钟模式, 同时 CLKST1:CLKST0 指示当前的时钟模式。如果满足最小条件, CLKST1:CLKST0 指示的状态应该和 CLKS1:CLKS0 中请求的一致。表 8-9 列出了 CLKS 和 CLKST 之间的关系。它也列出了 CLKS = CLKST 的条件或 CLKS ≠ CLKST 的原因。

注意

如果在下次复位之前需要使用晶振, 在第一次写 ICGC1 寄存器前, 必须设置 REFS = 1 和 CLKS = 1x。否则, 将导致锁定 REFS = 0, 这将在下一次复位前禁用振荡放大器。

表 8-9. ICG 状态表

实际模式 CLKST	期望的模式 CLKS	范围	参考频率 (f _{REFERENCE})	比较周期时间	ICGOUT	CLKS = CLKST 的条件 ¹	CLKS1 ≠ CLKST 的原因
OFF 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}	ICGDCLK/R	不从 FBE 跳转到 SCM	—
	FEI(01)	0	f _{ICGIRCLK} / 7 ¹	8/f _{ICGIRCLK}	ICGDCLK/R	—	DCOS = 0
	FBE(10)	X	f _{ICGIRCLK} / 7 ¹	8/f _{ICGIRCLK}	ICGDCLK/R	—	ERCS = 0
	FEE(11)	X	f _{ICGIRCLK} / 7 ¹	8/f _{ICGIRCLK}	ICGDCLK/R	—	DCOS = 0 或 ERCS = 0
FEI(01)	FEI(01)	0	f _{ICGIRCLK} / 7	8/f _{ICGIRCLK}	ICGDCLK/R	DCOS=1	—
	FEE(11)	X	f _{ICGIRCLK} / 7	8/f _{ICGIRCLK}	ICGDCLK/R	—	ERCS = 0

表 8-9. ICG 状态表 (continued)

实际模式 CLKST	期望的模式 CLKS	范围	参考频率 ($f_{\text{REFERENCE}}$)	比较周期时间	ICGOUT	CLKS = CLKST 的条件 ¹	CLKS1 \neq CLKST 的原因
FBE (10)	FBE(10)	X	0	—	ICGDCLK/R	ERCS=1	—
	FEE(11)	X	0	—	ICGDCLK/R	—	LOCS = 1 & ERCS = 1
FEE (11)	FEE (11)	0	f_{ICGIRCLK}	$2/f_{\text{ICGIRCLK}}$	ICGDCLK/R ³	ERCS = 1 且 DCOS = 1	—
		1	f_{ICGIRCLK}	$128/f_{\text{ICGIRCLK}}$	ICGDCLK/R ²	ERCS = 1 且 DCOS = 1	—

¹ 写 CLKS 后 CLKST 不会立即更新。CLKST 更新到新值需要几个总线周期。

² 在 SCM 模式下参考频率对 ICGOUT 无效，但该参考频率仍然用作比较，其比较误差决定 DCOS 位。

³ 在初始锁定过程中， f_{ICGIRCLK} 为 ICGDCLK/2R。当 MFD 位被改变后，FLL 重新锁定，此时 f_{ICGIRCLK} 为 ICGDCLK/R。

8.4.11 固定频率时钟

ICG 提供一个固定频率时钟输出 XCLK 用于片上外设。在除 FEE 以外的所有模式下，该时钟输出等于内部总线时钟 BUSCLK。在 FEE 模式下，如满足下列条件，XCLK 等于 ICGERCLK 除以 2：

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

如果以上两个条件不能满足，那么 XCLK 等于 BUSCLK。

当 ICG 在 FEI 或 SCM 模式时，XCLK 被关闭，任何使用 XCLK 作为时钟源的外设不能使用 XCLK。

8.4.12 高增益振荡器

振荡器可以运行在高增益振荡器（HGO）模式，当在 FBE 或 FEE 模式下时，高增益振荡器（HGO）模式能提高振荡器对 EMC 噪声的抵抗力。通过写 1 到 ICGC1 寄存器中的 HGO 位使能该模式。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，选择的是标准低功耗振荡器。复位后该位只能写一次。

8.5 初始化 / 应用信息

8.5.1 概述

本节给出了初始化时的基本指导，以及用户可参考的配置。在一些应用中，串行通信连接需要准确的参考时钟。在另一些应用中，最低功耗可能是时钟设置主要考虑的事项。也有一些应用，低成本是最主要的目标。ICG 具有很大的灵活性，为不同的应用选择最佳的时钟。

表 8-10. ICG 配置原则

	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（如果震荡范围 = 高，将会小于 FEE） 较高的时钟精确度（在 IRG 调整后） 最低系统成本（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（如果震荡范围 = 低，将会小于 FEI） 高时钟精确度 中等或高系统成本（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。 $3 \text{ MHz} < f_{\text{Bus}} < 5 \text{ MHz}$ （缺省） $3 \text{ MHz} < f_{\text{Bus}} < 20 \text{ MHz}$ （通过过滤器位） 中等功率 低精确度 IRG 关闭。DCO 打开并且开环。	FBE 当使用晶振或蜂鸣器时， $f_{\text{Bus}} \leq 8 \text{ MHz}$ 。 最低功率 最高时钟精确度 中等或高系统成本（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 ~ 2.5 mA。为了最小功耗和最小抖动，选择尽可能小的 N 和 R。

下面将根据不同配置举例介绍。

注意

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	复位后通常 $f_{\text{ICGOUT}} = 8 \text{ MHz}$
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	÷ 1
001	6	001	÷ 2
010	8	010	÷ 4
011	10	011	÷ 8
100	12	100	÷ 16
101	14	101	÷ 32
110	16	110	÷ 64
111	18	111	÷ 128

8.5.2 例 1: 外部晶振 = 32 kHz, 总线频率 = 4.19 MHz

在该例中, FLL 将倍频外部 32 kHz 振荡器到 8.38 MHz, 来产生 4.19 MHz 的总线频率。

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

时钟配置是使用 FLL, 外部 (FEE)。所以有:

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

N/R:

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

为执行操作, 每个寄存器需要的值如下:

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 预分频因子为 64
Bit 5	REFS	1	振荡器需要使用晶振或谐振器
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	未使用或保留; 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	设置 MFD 倍乘因子为 4
Bit 3	LOCRE	0	丢失时钟失产生一个中断请求
Bits 2:0	RFD	000	设置分频因子为 ÷ 1

ICGS1 = \$xx

该位只读, 除了清中断标志

ICGS2 = \$xx

该位只读；执行任何时间临界任务前应该读 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

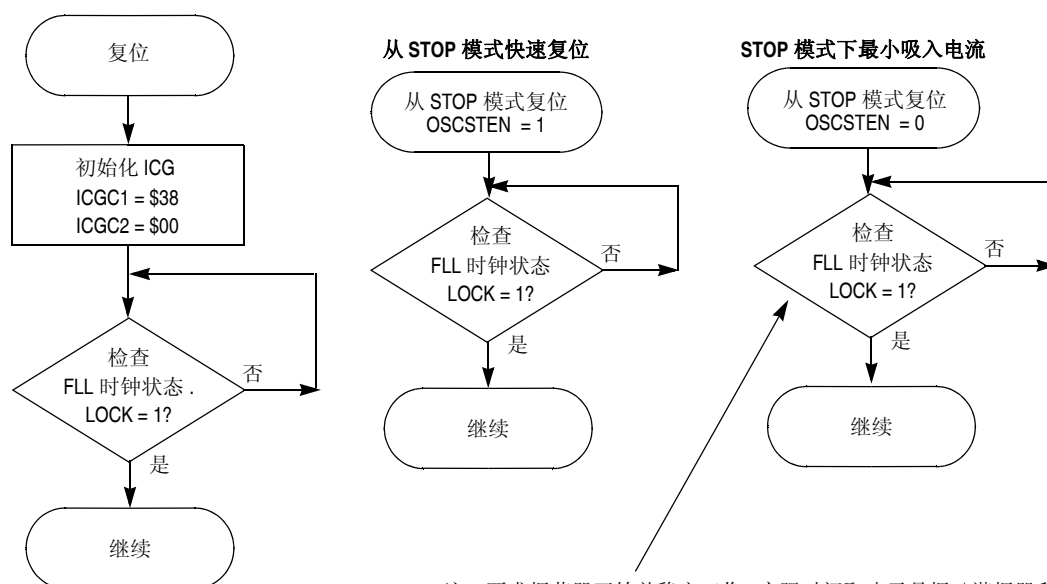
Bits 15:12 未使用 0000

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。采用外部晶振时不使用。

图 8-13 为需要 ICG 初始化的三个条件的流程图。



注：要求振荡器开始并稳定工作。实际时间取决于晶振 / 谐振器和外部电路

图 8-13. ICG 初始化用于 FEE 在例 1 情况下

8.5.3 例 2：外部晶振 = 4 MHz，总线频率 = 20 MHz

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

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

复位初始化软件时，时钟配置为使用 FLL，外部 (FEE)。所以

$$f_{ICGOUT} = f_{ext} * P * N/R ; P = 1, f_{ext} = 4.00 \text{ MHz} \quad \text{公式 8-3}$$

N/R:

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

为执行操作，每个寄存器需要的值如下：

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 预分频因子为 1
Bit 5	REFS	1	振荡器需要使用晶振或谐振器
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	保留或未使用；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	011	设置 MFD 倍乘因子为 10
Bit 3	LOCRE	0	丢失时钟产生一个中断请求
Bits 2:0	RFD	000	设置分频因子为 ÷ 1

ICGS1 = \$xx

该位只读，除了清中断标志

ICGS2 = \$xx

该位只读；执行任何时间临界任务前应该读 DCOS = 1

ICGFLTU/L = \$xx

在该例中未使用

ICGTRM = \$xx

在该例中未使用

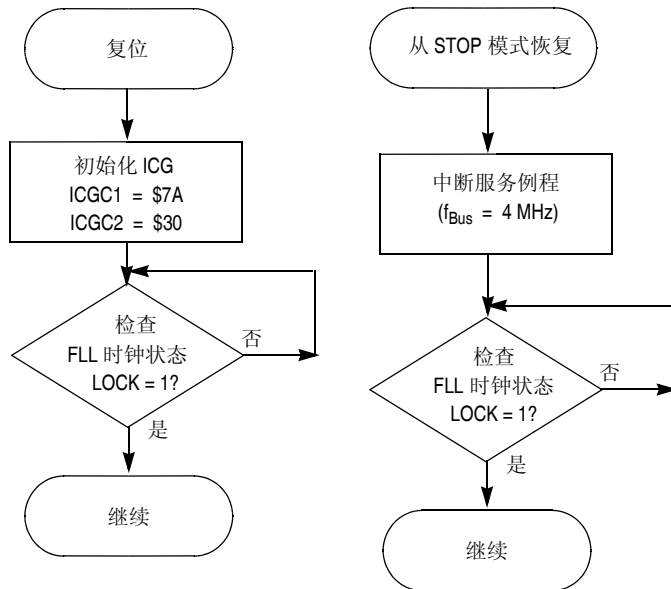


图 8-14. 用于例 2 情况下的 ICG 初始化和从 stop 模式复位

8.5.4 例 3: 无外部晶振, 总线频率 = 5.4 MHz

在本例中, FLL 将倍频内部 243 kHz (大约) 参考时钟到 10.8 MHz, 来产生 5.4 MHz 的总线频率。该系统使用微调功能基于外部参考信号获得更精确的频率。

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

时钟配置是使用 FLL, 内部 (FEI)。所以

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

N/R:

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

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

为执行操作, 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 预分频因子为 64
Bit 5	REFS	1	振荡器需要使用晶振或谐振器 (不使用该位)
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	保留或未使用; 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	设置 MFD 倍乘因子为 10
Bit 3	LOCRE	0	丢失时钟产生一个中断请求
Bits 2:0	RFD	001	设置分频因子为 ÷ 2

ICGS1 = \$xx

该位只读, 除了清中断标志

ICGS2 = \$xx

该位只读; 执行任何时间临界任务前建议读该位。

ICGFLTLU/L = \$xx

在本例中未使用

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)。

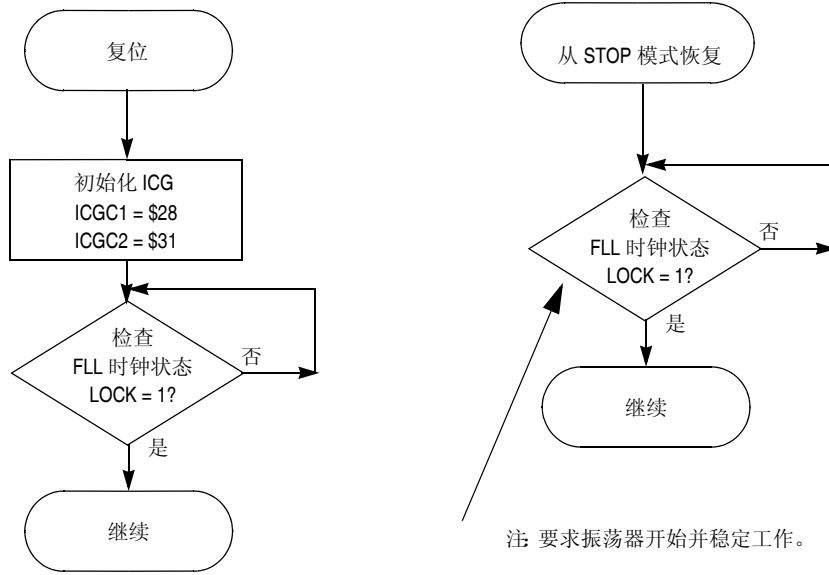


图 8-15. 用于例 3 情况下的 ICG 初始化和从 stop 模式复位

8.5.5 例 4: 内部时钟产生器调整

内部产生的时钟源有一个 $\pm 25\%$ 的浮动值。在一些情况，此精度能够满足要求。但在一些要求频率精度很高的应用中，需要调整程序帮助产生更准确的时钟源。该节给出了一个调整内部振荡器的例子。除此外，还有很多其他调整的方法可供使用。

初始条件:

- 1) ATE 提供的时钟工作周期为 500 μ s
- 2) ICG 设置为 4 MHz 内部总线频率

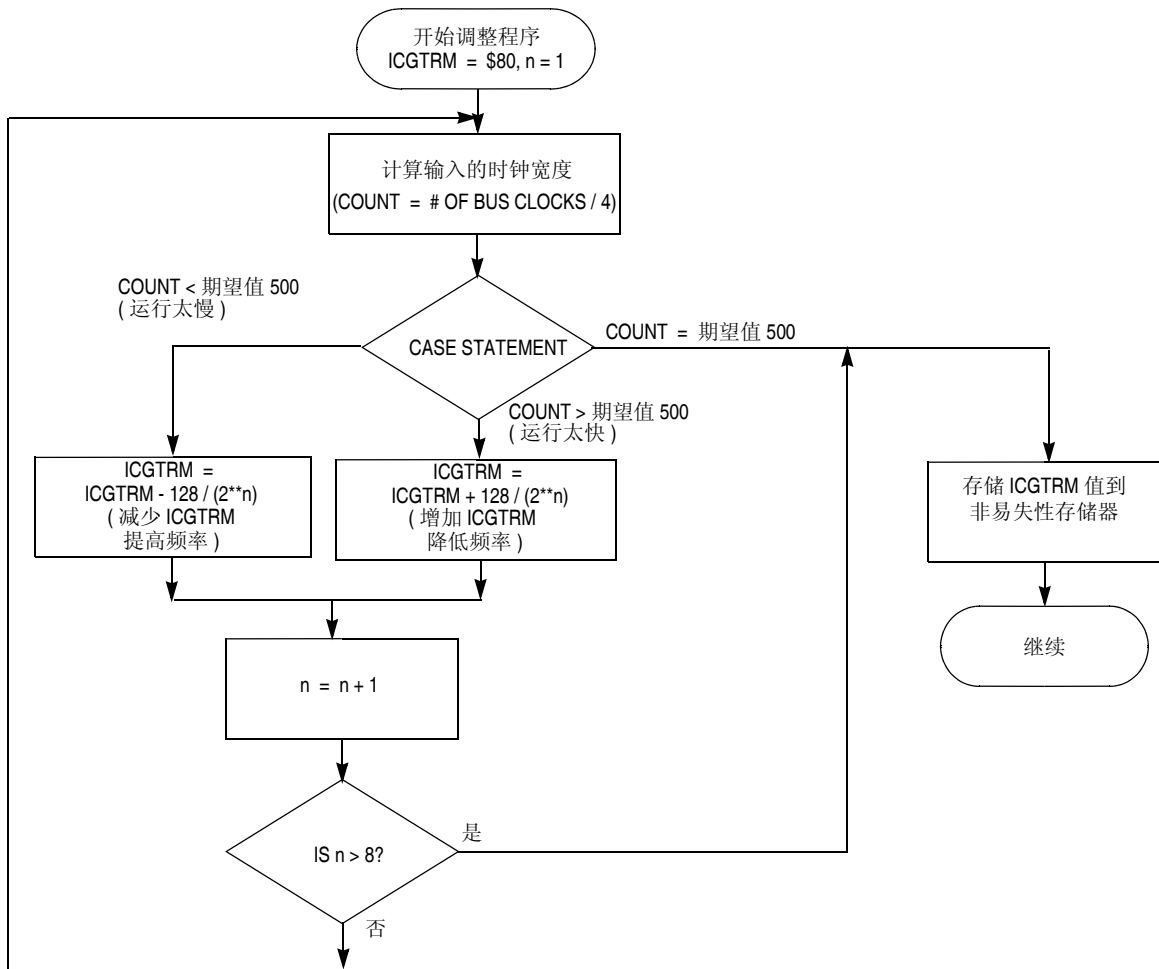


图 8-16. 调整程序

在这个特殊情况下，MCU 贴片到 PCB 上，整个装配正在用自动化测试工具进行最终测试。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 根据图 8-16 流程所示，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，推荐使用两倍于最终值的分频因子（R）。调整程序完成后，恢复分频因子。这可以避免意外地超过最大时钟频率。

第 9 章

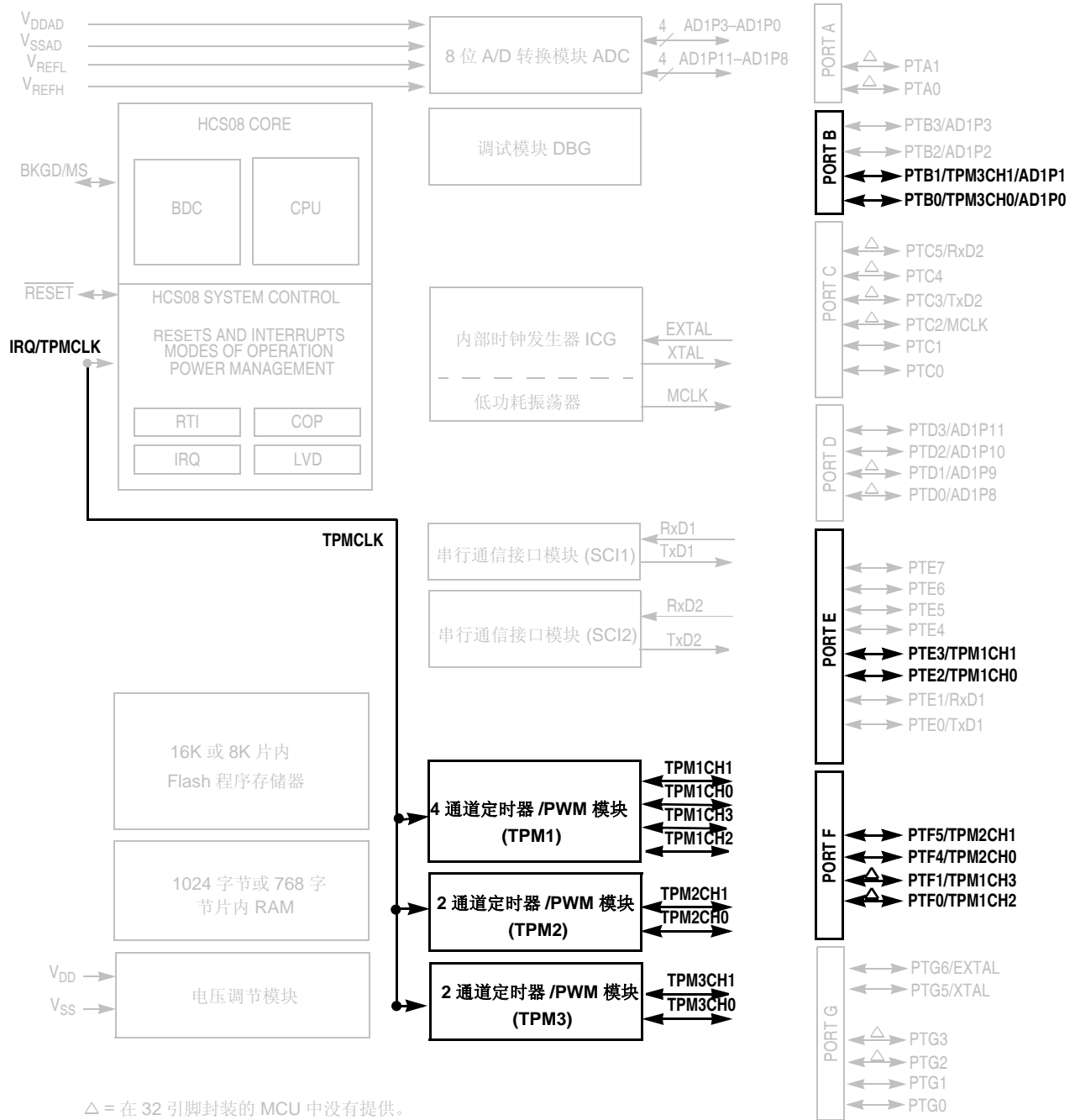
定时器 /PWM (S08TPMV3)

9.1 概述

SC667027 系列包含三个独立定时器 /PWM 模块，支持传统的输入捕捉，输出捕捉以及带缓冲的边沿对齐 PWM。每个 TPM 有一个控制位配置所有的通路产生中心对齐的 PWM 信号。在每一个 TPM 中，定时功能是基于对独立的带分频器以及模块化的 16 位计数器来控制频率和范围 (溢出周期)。这个计时系统非常适于多种控制应用，TPM 的中心对齐 PWM 能力使其能应用于更加广泛的领域，如小家电的电机控制。

由于可以采用固定的系统时钟 XCLK 作为任何 TPM 模块的时钟源，这使得 TPM 分频器可以运行于振荡器频率的一半 (ICGERCLK/2)。当然此时 ICG 必须是 FEE 模式配置且满足适当的条件 (见 8.4.11 节 固定频率时钟)。这样的选择在其它 ICG 模式下是多余的，因为 XCLK 等于 BUSCLK。

外部时钟源可以连接到 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 模块可独立选择 TPMCLK 作为时钟源。



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 9-1. 显亮 TPM 模块和引脚的 SC667027 模块结构图

9.2 TPMV3 与以前版本的区别

TPMV3 是定时 /PWM 模块的最新版本，它解决了先前版本勘误表中涉及的问题。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因素。

表 9-1. 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 计数器的返回值不可修改	在 BDM 模式有效之前，如果读取只有一字节的 TPMxCNTH:L 寄存器，读到的是缓冲区的锁存的值（而不是固定的 TPM 计数器值）
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，如果读取只有一字节的 TPMxCnVH:L 寄存器，读到的是缓冲区的锁存的值（而不是 TPMxCnVH:L 寄存器的值）
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 ³	不允许	允许
在输出比较模式下，当 (CLKSB:CLKSA ≠ 0:0)，写 TPMxCnVH:L 寄存器 ³	在写入第二个字节后，使用 TPM 计数器（分频计数结束）下个变化时的写缓冲区的值更新 TPMxCnVH:L 寄存器内容	在第二字节内容被写入后，立即更新其寄存器内容
在边沿对齐模式下，当 (CLKSB:CLKSA ≠ 00) 写 TPMxCnVH:L 寄存器	当 TPM 计数器值从 (TPMxMODH:L-1) 到 TPMxMODH:L，两个字节被写入后，用写缓冲区的值更新 TPMxCnH:L 值 注：如果 TPM 计数器用作自由运行计数器，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

表 9-1. TPMV2 和 TPMV3 移植条件

作用	TPMV3	TPMV2
在中心对齐模式下, 当 (CLKSB:CLKSA \neq 00), 写 TPMxCnVH:L 寄存器 ⁴	当 TPM 计数器值从 (TPMxMODH:L-1) TPMxMODH:L, 两个字节被写入后, 用写缓冲区的值更新 TPMxCnH:L 值 注: 如果 TPM 计数器用作自由运行的计数器, 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	在一个新的 PWM 周期中开始使用新的占空比设置	在当前 PWM 周期中期 (当计数为 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当前 PWM 周期	用新占空比设置完成当前 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 9.5.2 节 TPM 计数器寄存器 (TPMxCNTH:TPMxCNHTL)

² 更多信息参考 9.5.5 节 TPM 通道值寄存器 (TPMxCnVH : TPMxCnVL)

³ 更多信息参考 9.6.2.1 节 输入捕捉模式

⁴ 更多信息参考 9.6.2.4 节 中心对齐 PWM 模式

⁵ 更多信息参考 9.6.2.4 节 中心对齐 PWM 模式 【SE110-TPM 案例 1】

⁶ 更多信息参考 9.6.2.4 节 中心对齐 PWM 模式 【SE110-TPM 案例 2】

⁷ 更多信息参考 9.6.2.4 节 中心对齐 PWM 模式 【SE110-TPM 案例 3, 5】

⁸ 更多信息参考 9.6.2.4 节 中心对齐 PWM 模式 【SE110-TPM 案例 4】

9.2.1 从 TPMV1 移植

作为 9.2 节 TPMV3 与以前版本的区别的补充, 当从带 TPMV1 的 MCU 移植时, 必须考虑以下事项。

- 当定时器是 TPMV2 不在输入捕捉模式下, 而不是 TPMV3 时, 可以写通道值寄存器 (TPMxCnV)。
- 在边沿或中心对齐 PWM 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。
- 在配置 TPM 模块时, 最好先写 TPMxSC 再写 TPMxCnV, 因为写 TPMxSC 会复位 TPMxCnV 寄存器的一致机制。

表 9-2. 移植到 TPMV3 的条件表

条件	操作 / 经验教训
写入通道值寄存器 (TPMxCnV) 导入	定时器必须在输入捕捉模式下
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

9.3 特性

SC667027 系列定时器系统包括 4 通道 TPM1，一个单独的 2 通道 TPM2 和一个单独的 2 通道 TPM3。定时器系统特征如下：

- 总共最多能达到 8 个通道
 - 每个通道可用作输入捕捉，输出比较或带缓冲的边沿对齐 PWM
 - 上升沿，下降沿，或任何边沿输入捕捉触发
 - 置位，清零或触发输出比较
 - PWM 输出的极性可选
- 在所有通道上，每个 TPM 可配置带缓冲的，中心对齐 PWM（CPWM）
- 每个 TPM 可运行于独立的时钟源，这些时钟源可以是总线时钟、固定系统时钟或通过引脚引入的外部时钟
 - 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 位自由运行或累加 / 递减（CPWM）等计数操作
- 16 位模计数寄存器用来控制计数范围
- 计时系统使能
- 每个通道有一个中断，每一个 TPM 模块有一个终端计数中断

9.3.1 操作模式

一般来说，TPM 通道可以被独立设置为输入捕捉、输出比较或边沿对齐模式。设置一个控制位，可以使整个 TPM 通道转换为对齐 PWM 模式。当使能中心对齐 PWM 模式，在这个 TPM 模块的其它通道上，输入捕捉、输出比较和边沿对齐 PWM 功能将被禁用。

当微控制器处在活跃 BDM 后台调试模式或 BDM 前台调试模式下。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，因此直到时钟信号重启，TPM 都处于关闭状态。在等待模式下时，TPM 可正常工作。假如 TPM 不需产生一个实时参考时间或提供中断源从等待模式唤醒 MCU，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

当一个设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置位中断标志位。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

当定时计数寄存器的值匹配通道值寄存器值时，将置位中断标志位，且相应的 MCU 引脚产生一个可选择输出行为。该输出比较行为可被选来迫使引脚为 0，为 1，锁存它，或忽略该引脚（用于软件定时功能）。

- 边沿对齐 PWM 模式

16 位模块寄存器的值加 1 设置 PWM 输出信号的周期。通道值寄存器设置 PWM 输出信号的占空比。用户也可以选择 PWM 输出信号的极性。在该周期末尾和占空比转换临界点，可以产生中断。这种类型的 PWM 信号被称作边沿对齐 PWM 是因为所有 PWM 主要边沿信号与周期开始时对齐，同时一个 TPM 的所有通道都相同。

- 中心对齐 PWM 模式

16 位模块寄存器值的两倍设置 PWM 输出周期，同时通道值寄存器设置一半占空比的延时。定时器计数递增直到它达到模块值然后计数递减直到 0。当计数值递减并与通道值寄存器相匹配时，PWM 输出活跃。当计数值递增并与通道值寄存器相匹配时，PWM 输出不活跃。这种类型的 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 被广泛应用于小家电的电机。

这仅是一个概述，在稍后章节将进行详细介绍。

9.3.2 结构框图

TPM 每个通道使用一个输入 / 输出引脚，TPMxCHn（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（请参考相应微控制器的输入输出引脚描述）。图 9-2 是 TPM 结构图。TPM 的主要组成部分是 16 位计数器，可以作为一个自运行计数器或一个递增 / 递减计数器。TPM 计数器（运行于递增模式时）为输入捕捉，输出比较，边沿对齐 PWM 功能提供参考时钟。定时器计数模块寄存器 TPMxMODH:TPMxMODL 控制计数器因子值（值 0x0000 或 0xFFFF 可使计数器自运行）。软件可以任意读取计数器值，而不影响计数器的值。TPMxCNT 计数器的任意一半被写入将复位计数器，而不管写入的是什么。

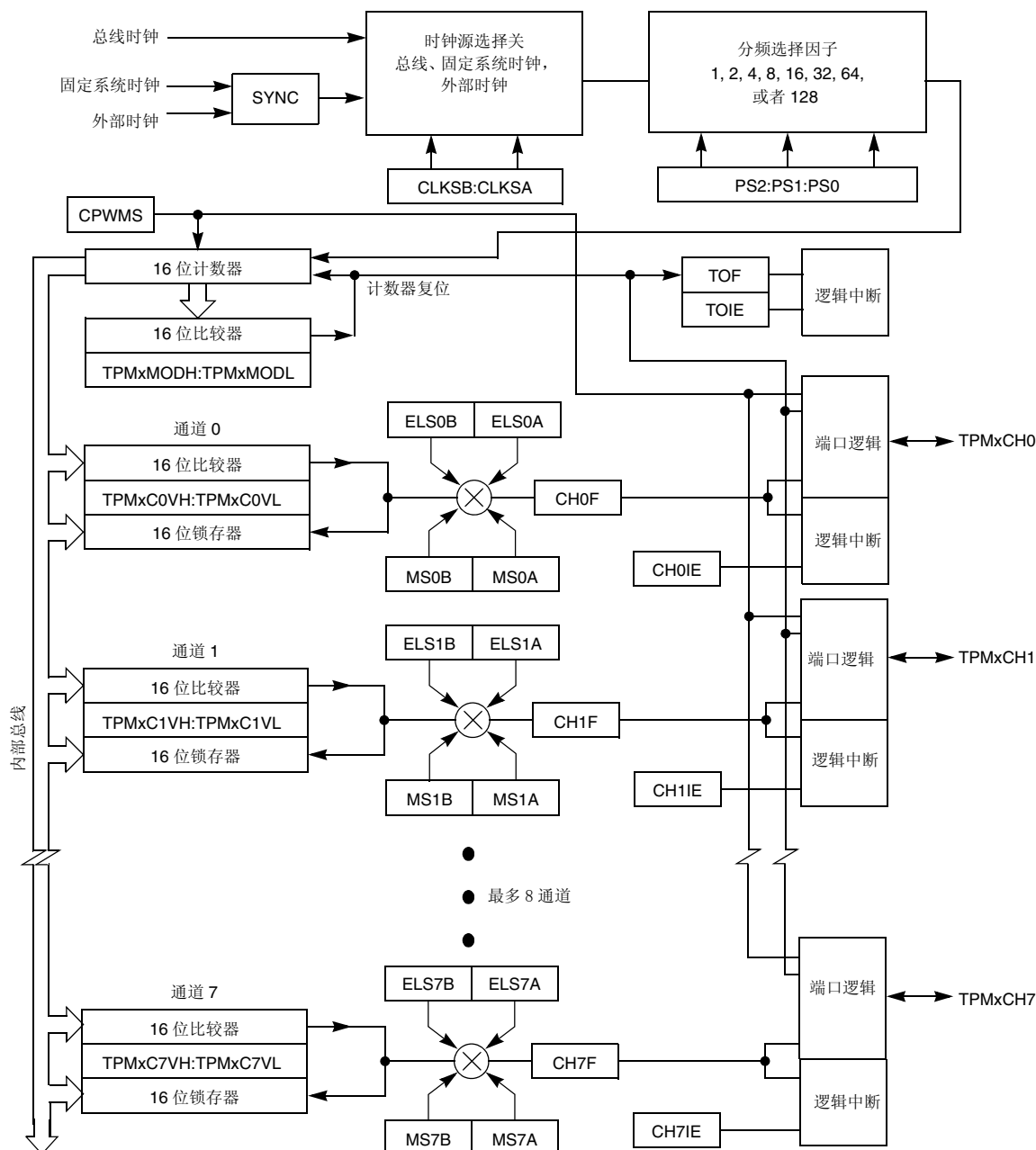


图 9-2. TPM 结构图

TPM 通道和输入捕捉，输出比较或边沿对齐 PWM 通道一样是可独立编程设置的。另一种情况是，TPM 可将所有通道配置成 CPWM 输出。当 TPM 被配置成 CPWM 时，递增计数器 / 递减计数器，输入捕捉，输出比较、EPWM 等功能无效。

如果通道被配置成输入捕捉模式，则内接一个上拉设备。因为 I/O 口引脚和相关通用 I/O 口控制位不是模块的一部分，模块如何利用引脚相互作用主要取决于芯片的执行情况。可参阅微控制器关于输入输出端口逻辑的描述。

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，所以通常使用三通道或六通道。

9.4 信号描述

表 9-3 显示用户可访问的 TPM 信号。通道的数量可以是 1 到 8 个。所有的 TPM 通道都可以复用同一个外部时钟，而且外部时钟只需连接到单独的输入引脚。具体实现可参阅微控制器 I/O 口的描述。

表 9-3. 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，该信号可复用给任何通道引脚；然而根据微控制器的实现情况，该信号可被连接到独立的外部引脚。

² n = 通道数量 (1 ~ 8)

关于复位状态，端口连接，和这些引脚是否有上拉电阻的详情请参阅该微控制器的说明。TPM 通道引脚可与通用 I/O 口引脚相关联且有被动上拉电阻，当 TPM 或通用 I/O 控制配置相关引脚为输入时，上拉电阻可以用一个控制位使能。当没有使能 TPM，相应的引脚可应用于 I/O 控制，包括端口数据，数据方向寄存器。复位后，默认没有使能 TPM，所以所有相关引脚被通用 I/O 控制。

9.4.1 信号详细描述

本节将详细描述每个用户可访问的引脚信号。虽然表 9-3 已经列出所有引脚，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本微控制器的说明。

9.4.1.1 EXTCLK——外部时钟源

位于定时器状态和控制寄存器中的控制位无效（此时定时器无效），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，外部时钟源是同步的。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。

外部时钟信号和通道 I/O 引脚复用相同引脚。所以当被选做外部时钟源时，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用作软件定时器（ELSnB:ELSnA=0:0）。

9.4.1.2 TPMxCHn——TPM 通道 n 输入输出引脚

MCU 的每个 TPM 通道与一个 I/O 引脚相关联。引脚功能依赖于通道配置。TPM 引脚与通用 I/O 引脚复用，它的每个引脚对应一个端口数据寄存器位，一个数据方向控制位，并且作为输入时，端口具有可使能的被动上拉电阻。

当（ELSnB:ELSnA = 0:0）或（CLKSB:CLKSA = 0:0）时，TPM 通道不能控制 I/O 引脚。所以通常转为通用 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 控制。当 CPWMS = 0 时，MSnB:MSnA 控制位决定该通道配置为输入捕捉，输出比较或边沿对齐 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 的边沿有效输入。ELSnB:ELSnA 控制位决定触发输入捕捉事件的极性边沿或者边沿。基于总线时钟的同步器用于将输入边沿同步到总线时钟上。这意味着在一个输入捕捉引脚上能被可靠检

测到的最小脉冲宽度是总线时钟周期的四倍（能检测接近两个总线时钟的理想时钟脉宽）。TPM 把这个引脚当作输入捕捉的输入，而忽略端口数据和数据方向控制。

当通道被配置为输出比较模式（CPWMS = 0，MSnB: MsnA = 0:1，ELSnB:ELSnA = 0:0）时，忽略相关数据方向控制，TPMxCHn 引脚被认为是由 TPM 控制输出，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沿对齐 PWM 模式（CPWMS = 0，MSnB = 1，ELSnB:ELSnA ≠ 0:0）时，数据方向被忽略，TPMxCHn 引脚被强制为由 TPM 控制输出，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

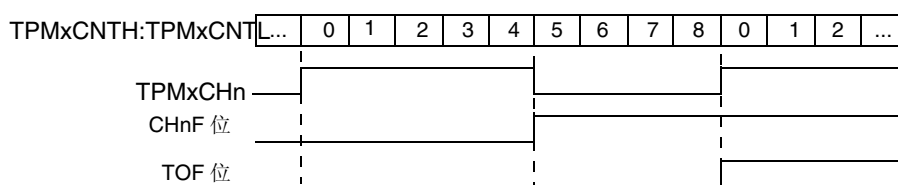


图 9-3. 边缘对齐 PWM 模式的高真脉冲

TPMxMODH:TPMxMODL = 0x0008
TPMxMODH:TPMxMODL = 0x0005

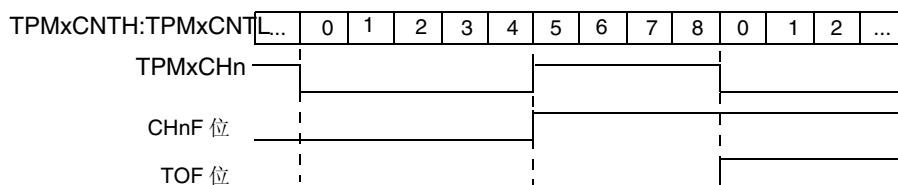


图 9-4. 边缘对齐 PWM 模式的低真脉冲

当 TPM 被配置为中心对齐 PWM 模式（ELSnB:ELSnA ≠ 0:0），在 TPM 模式下，所有通道的数据方向被忽略，TPMxCHn 引脚被强制转换为由 TPM 控制输出，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。

定时器 /PWM (S08TPMV3)

TPMxMODH:TPMxMODL = 0x0008
 TPMxMODH:TPMxMODL = 0x0005

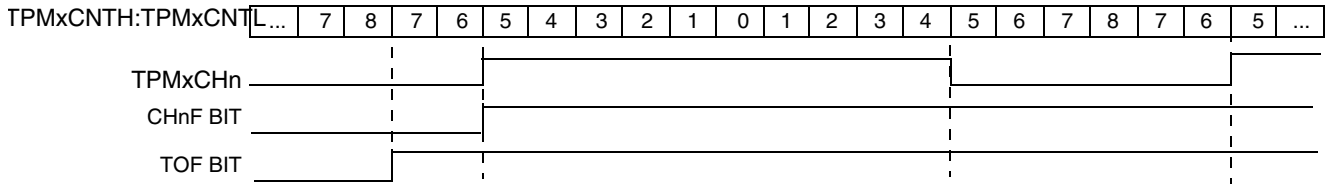


图 9-5. 中心对齐 PWM 模式的高真脉冲

TPMxMODH:TPMxMODL = 0x0008
 TPMxMODH:TPMxMODL = 0x0005

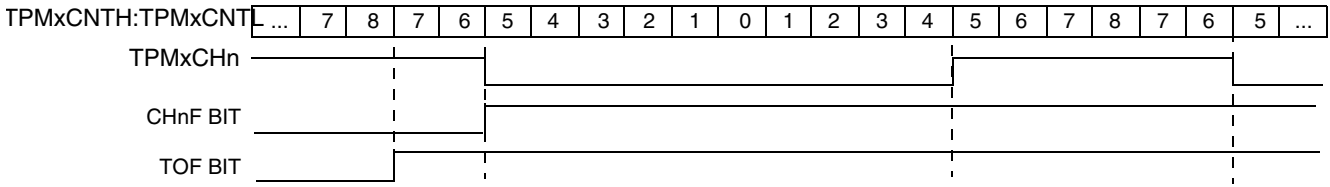


图 9-6. 中心对齐 PWM 模式的低真脉冲

9.5 寄存器定义

本节将按地址顺序描述寄存器。一个典型 MCU 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哪个 TPM 和哪个通道被选用了。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。

9.5.1 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、TPM 配置、时钟源、分频因子的控制位。这些控制涉及到此定时器模块的所有通道。

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

图 9-7. TPM 状态和控制寄存器 (TPMxSC)

表 9-4. TPMxSC 位描述

位	描述
7 TOF	定时器溢出标志，当 TPM 计数器达到在 TPM 计数器模寄存器中预设值后，复位为 0x0000，然后置位这个读写标志。当 TOF 被置位，可以通过读 TPM 状态和控制寄存器，然后向 TOF 写 0 清除 TOF。如果在彻底清除序列前发生了另一个 TPM 溢出，该序列复位，并在先前清除序列完成后，TOF 还将保持置位。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。 0 TPM 计数器未达到模值或未溢出 1 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。 0 TOF 中断禁止（用于软件轮询） 1 TOF 中断使能
5 CPWMS	中心对齐 PWM 选择位。这个读写位选择 CPWM 操作模式。默认情况下，TPM 提供输入捕捉，输出比较和边沿对齐 PWM 功能，并提供递增计数器操作。设置 CPWMS 重新配置 TPM 操作于上升 / 下降模式计数器，实现 CPWM 功能。复位清零 CPWMS。 0 所有通道可作输入捕捉，输出比较，边沿对齐 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 1 所有通道都可运行在中心对齐 PWM 模式下
4-3 CLKS[B:A]	时钟源选择位，如表 9-5 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，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，固定系统时钟源通过片上同步电路同步总线时钟。当有 PLL 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 个分频因子中的一个用来分频 TPM 时钟输入，如表 9-6 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后，新的分频因子将对下个系统时钟周期产生影响。

表 9-5. 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 9-6. 分频因子选择

PS2:PS1:PS0	分频因子
000	1
001	2
010	4
011	8

表 9-6. 分频因子选择 (continued)

110	16
101	32
110	64
111	128

9.5.2 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低位字节内容。在读 TPMxCNTH 或 TPMxCNHTL 中的一个字节时，两字节内容都被锁存到一个缓冲区，直到剩下的一个字节内容也被读出。这使得 16 位连读而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，不管是否涉及到写数据。

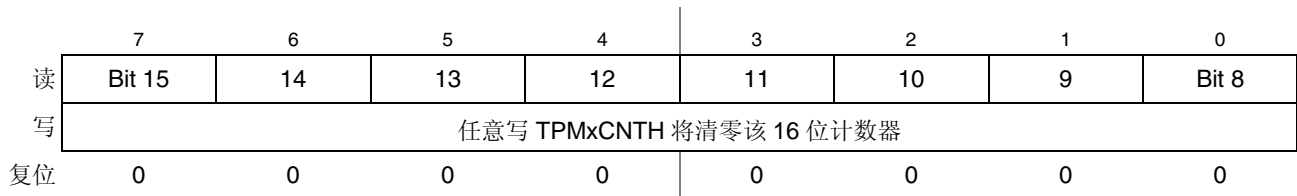


图 9-8. TPM 计数器寄存器高位 (TPMxCNTH)



图 9-9. TPM 计数器寄存器低位 (TPMxCNTL)

当 BDM 被激活，定时器计数器被暂停（这个值将被用户读取）；当 BDM 被激活时，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在 BDM 活跃时被读取。这保证了当 BDM 被激活时如果用户正在读 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 寄存器将复位 TPMxCNTH:L 寄存器读一致机制，不管是否涉及写的的数据。

9.5.3 TPM 计数器模寄存器 (TPMxMODH:TPMxMODL)

该读写 TPM 模寄存器包含其模值。当 TPM 计数器达到模值之后，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 禁止 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), 即使模寄存器的一半或者全部在 BDM 活跃时被写, 缓冲锁仍保持在 BDM 激活时的状态。当 BDM 活跃时, 任何写模块寄存器将越过缓冲锁存而直接写模块寄存器。

	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 9-10. TPM 计数器模寄存器高位 (TPMxMODH)

	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 9-11. 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。

9.5.4 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。

	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

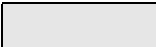
 = 保留或未使用

图 9-12. TPM 通道 n 状态和控制寄存器 (TPMxCnSC)

表 9-7. TPMxCnSC 位描述

位	描述
7 CHnF	<p>通道 n 标志。</p> <p>当通道 n 为输入捕捉通道时，通道 n 上的有效边沿触发该位置位。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 为边沿对齐 / 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，产生相应的中断请求。当 CHnF 被置位时通过读 TPMxCnSC，并向 CHnF 写 0 清空 CHnF。在清空序列完成前如果另一个中断请求发生，该序列复位，并且在清空前一个中断序列之后，CHnF 仍保持置位。这样，不会因为清除前一个中断标志，而丢失新的 CHnF 中断请求。复位清空 CHnF 位。向 CHnF 写 1 不产生影响。</p> <p>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</p>
6 ChnIE	<p>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。</p> <p>0 禁止通道 n 中断请求 1 允许通道 n 中断请求</p>
5 MSnB	TPM 通道 n 的 B 模式。当 CPWMS = 0, MSnB = 1 时配置 TPM 通道 n 为边沿对齐 PWM 模式。参阅表 9-8 中通道模式和设置控制。
4 MSnA	<p>TPM 通道 n 的 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 9-8 中通道模式和设置控制。</p> <p>注：在转变为输入捕捉模式之前，如果相关端口引脚在至少两个总线时钟周期内不稳定，将会出现意外的边沿触发。</p>
3-2 ELSnB ELSnA	<p>边沿 / 电平选择位。</p> <p>取决于定时器通道的操作模式，通过 CPWMS:MSnB:MSnA 来设置，详细可参照表 9-8。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作为通用 I/O 引脚。这种功能用于暂时禁止输入捕捉，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。</p>

表 9-8. 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	边沿对齐 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
X1			低保真脉宽 (比较时设置输出)	

9.5.5 TPM 通道值寄存器 (TPMxCnVH : TPMxCnVL)

这些读写寄存器包含通过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。

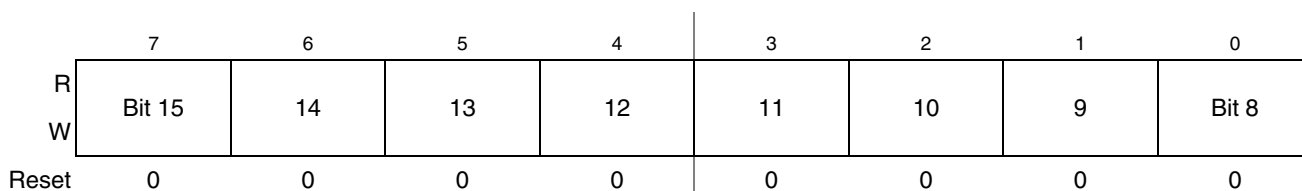


图 9-13. TPM 通道值寄存器高位 (TPMxCnVH)

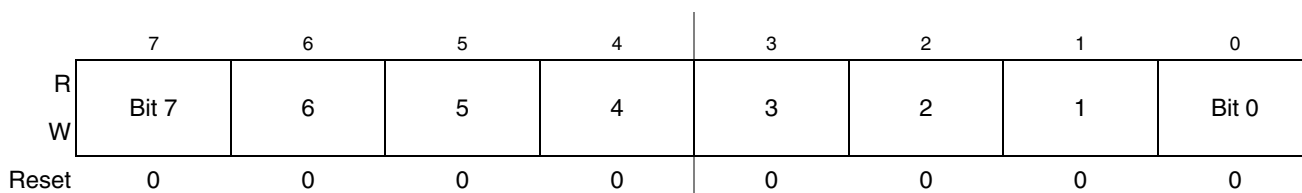


图 9-14. TPM 通道值寄存器低位 (TPMxCnVL)

在输入捕捉模式下，在读 (TPMxCnVH : TPMxCnVL) 任一字节过程中，寄存器两个字节的內容将被锁存到缓冲器中，直至读取另外一个字节。当 TPMxCnSC 寄存器被写 (BDM 模式是否活跃) 时，这个锁机制可被复位 (变为非锁定)。任何对通道寄存器的写操作在输入捕捉模式下将被忽略。当 BDM 活跃时，该一致机制被暂停 (除非通过写 TPMxCnSC 寄存器复位)，这样，即使读部分或全部通道寄存器，缓冲器锁存值保持在 BDM 变为活跃时的状态。这确保了正在读 16 位寄存器时，用户使能 BDM，而在返回到正常执行后，仍能正确读取 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) 其中的一个字节，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 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，并且在 TPM 计数器下个变化时 (分频计数结束) 寄存器被更新。
- 如果 CLKS_B:CLKS_A≠0:0 并且在 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 0xFFFFE 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。该锁机制允许以升序或降序写入 16 位值，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)，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。当 BDM 活跃时，任何写通道寄存器将越过缓冲锁存而直接写通道寄存器。在 BDM 活跃时且恢复正常执行时，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。

9.6 功能描述

所有的 TPM 功能与 16 位计数器相关，此计数器允许灵活的选择时钟源和分频因子。同时还有一个 16 位模寄存器相应于此主计数器。

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、输出比较或边沿对齐 PWM 模式下。该 CPWMS 控制位位于主 TPM 状态和控制寄存器，因为它对影响 TPM 的所有通道，并决定主计数器的操作方式。(在 CPWM 模式下，作为通用计数器功能计数器通常为增 / 减模式)。

下面各节将描述核心计数器和每个定时器的操作模式 (输入捕捉，输出比较，边沿对齐 PWM，中心对齐 PWM)。因为引脚操作的细节和中断活跃取决于操作模式，相应的内容在相关模式章节进行讲述。

9.6.1 计数器

所有定时器功能建立在核心 16 位计数器基础之上 (TPMxCNTH : TPMxCNTL)。本节将讨论时钟源的选择、计数结束溢出、计数增 / 减、计数器复位。

9.6.1.1 计数器时钟源

CLKSB:CLKSA 位于定时器状态和控制寄存器 (TPMxSC)，用来选择三个时钟源中的一个或 OFF (能有效禁止 TPM)。见表 9-5。MCU 在复位后，CLKSB:CLKSA = 0:0，没有时钟源被选择，TPM 处于低功耗状态。这些控制位可在任意时间被读或写，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 9-9. TPM 时钟源选择

CLKSB:CLKSA	用于分频输入的 TPM 时钟源
00	无时钟被选择 (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 中或不使用 PLL 电路时，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。当使用 PLL 或者 FLL 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，以保证计数器转换和总线时钟转换相一致。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。该时钟源通过一个同步装置来确保计数器转换和总线时钟转换能一致。总线速率时钟驱动同步装置；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。理想情况下外部时钟等于总线时钟除以 4。

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，不能再把它配置成输入捕捉 (用户应尽量避免)。TPM 通道仍然能在输出比较模式中使用软件定时功能 (引脚控制设置不影响 TPM 通道引脚)。

9.6.1.2 计数器溢出和模复位

一个中断标志和中断使能与 16 位核心计数器相关。软件可访问的 TOF 用来标识定时器计数器溢出。该使能信号可选择软件轮询 (TOIE = 0, 即无硬件中断产生) 或中断驱动 (TOIE = 1, 即一个硬件中断产生)。

TOF 是否置位取决于 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, 在下一个计数时钟溢出为 0x0000。当从 0xFFF 变为 0x000 时 TOF 置位。当 TPM 处于中心对齐 PWM 模式时 (CPWMS = 1), 当计数器达到模数寄存器中到预设值, 并改变计数方向时, TOF 置位。这依据于 PWM 周期结束时的值。(0x000 计数值对应于一个周期的中间值)。

9.6.1.3 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, 计数器用作单增计数器。作为单增计数器, 定时器计数器从 0x0000 计数直到终点值然后再从 0x0000 重新开始计数。该终端计数是 0xFFFF 或是 TPMxMODH : TPMxMODL 设定的模数值。

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) 在终点计数周期结束时 (计数变为下个更小的计数值时) 被置位。

9.6.1.4 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。

9.6.2 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, 输出比较和边沿对齐 PWM。

9.6.2.1 输入捕捉模式

在输入捕捉模式中, 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保证 16 位被一致存取。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。

在 BDM 模式时, 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。

9.6.2.2 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的通道值寄存器中设置的值时, 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，数值被传递到相应定时器通道寄存器。所以：

- 如果 (CLKSB: CLKSA=0:0)，当写入第二个字节，寄存器被更新
- 如果 (CLKSB: CLKSA!=0:0)，第二个字节被写后，在 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。

9.6.2.3 边沿对齐 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 TPM 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 PWM 信号周期由模寄存器值 (TPMxMODH : TPMxMODL) 加 1 决定。占空比由设置在定时器通道寄存器 (TPMxCnVH : TPMxCnVL) 的值来决定。该 PWM 信号的极性由 ELSnA 控制位来决定。1% 和 100% 占空比是可能的。

在 TPM 通道值寄存器中的输出比较值决定 PWM 信号的脉宽 (占空比) (图 9-15)。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。

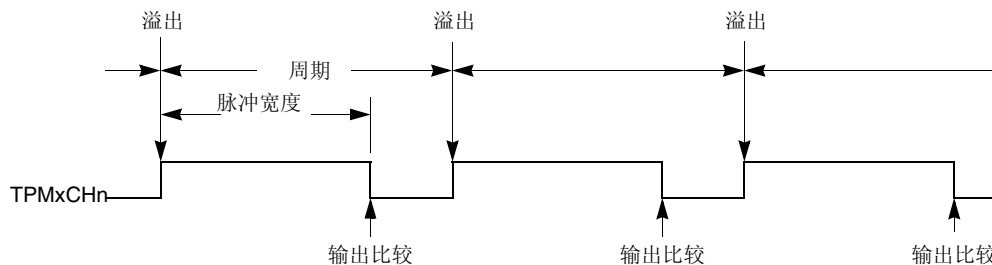


图 9-15. PWM 周期和脉冲宽度 (ELSnA=0)

当通道值寄存器被设为 0x0000 时，占空比为 0%。通过设置定时器寄存器 (TPMxCnVH : TPMxCnVL) 为一个大于模数设置的值，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。在边沿对齐 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)，当写第二个字节时，寄存器被更新
- 如果 (CLKSB: CLKSA != 0:0)，在两个字节都被写入后，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 0xFFFFE 变为 0xFFFF 时，更新被执行。

9.6.2.4 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 决定 CPWM 输出的极性。

脉宽 = 2 × (TPMxCnVH : TPMxCnVL)

周期 = $2 \times (\text{TPMxMODH}:\text{TPMxMODL})$; $\text{TPMxMODH}:\text{TPMxMODL} = 0x0001 - 0x7FFF$

如果通道值寄存器 $\text{TPMxCnVH}:\text{TPMxCnVL}$ 为零或负数（位 15 置位），占空比将为 0%。如果 $\text{TPMxCnVH}:\text{TPMxCnVL}$ 为正数（位 15 清空）且大于模数设定值（非负），占空比为 100%，这是因为绝不会发生占空比相比。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一般应用的周期长很多。

$\text{TPMxMODH}:\text{TPMxMODL} = 0x0000$ 是一个特殊情况，通常不用于中心对齐 PWM 模式。当 $\text{CPWMS} = 0$ 时，计数器将从 $0x0000$ 到 $0xFFFF$ 自动运行，但当 $\text{CPWMS} = 1$ 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 9-16）的脉宽（占空比）。如果 $\text{ELSnA} = 0$ ，当计数增大强制 CPWM 输出信号为低时进行一个比较，当计数减小强制输出为低时进行一个比较。计数器计数增加直到它达到 $\text{TPMxMODH}:\text{TPMxMODL}$ 中的模数设定值，然后计数下降直至为零。这将设置周期等于两倍 $\text{TPMxMODH}:\text{TPMxMODL}$ 。

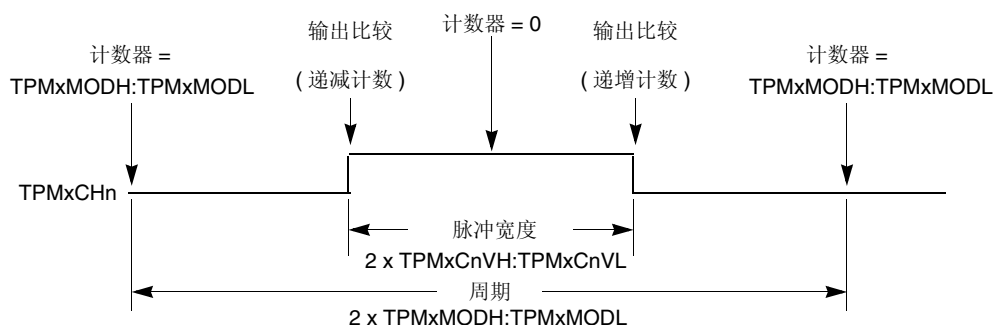


图 9-16. 16CPWM 周期和脉冲宽度 ($\text{ELSnA}=0$)

中心对齐 PWM 输出比边沿对齐 PWM 模式产生更少的噪音，这是因为较少的 I/O 引脚转换被列队在同一系统时钟边沿。某些需要电机驱动的类型也需要这种 PWM 类型。

当计数器操作在增 / 减计数模式时，输入捕捉、输出比较和边沿对齐 PWM 功能无意义，这表明当 $\text{CPWMS} = 1$ 时，所有带 TPM 的活跃通道必须运行于 CPWM 模式。

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 $\text{CLKSB}:\text{CLKSA}$ 位的值用写缓冲里的值来更新 $\text{TPMxCnVH}:\text{L}$ 寄存器。所以：

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

写 TPMxSC 会取消任何写入 TPMxMODH 和 / 或 TPMxMODL 的值，并且复位模寄存器一致机制。写 TPMxCnSC 会取消写入通道值寄存器的值，并且复位 $\text{TPMxCnVH}:\text{TPMxCnVL}$ 的一致机制。

9.7 复位概述

9.7.1 概要

MCU 复位会导致 TPM 复位。

9.7.2 复位操作

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)。

9.8 中断

9.8.1 概述

TPM 给核心计数器提供一个溢出中断, 并给每个通道提供一个中断。该通道中断意义取决于每个通道的操作模式。如果通道被配置为输入捕捉, 则每次设定的输入捕捉边沿被确认时中断标志被置位。如果通道被配置为输出比较或 PWM 模式, 当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 9-10, 包括中断名称, 中断使能。这些中断使能可阻塞离开 TPM 的中断请求和通过独立中断处理逻辑获得确认。

表 9-10. 中断

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最终计数值时置位 (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, 可参考用户手册的中断模块章节或芯片完整文档了解相关的细节。

9.8.2 中断操作描述

对于 TPM 的每个中断源, 当中断情形, 如定时器溢出, 通道输入捕捉, 或输出比较事件被确定, 将对一个标志位置位。该标志位可通过软件读取, 以确定事件已发生, 或者设置相关使能为允许硬件中断。当中断使能位被置位, 只要相关中断标志等于 1, 将产生一个相应的中断。用户软件必须执行一些步骤序列, 用以在终端服务例程返回之前清空中断标志。

清空 TPM 中断标志的流程包括两个步骤, 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检测到新事件, 该序列被复位, 在第二步后中断标志位保持置位以避免丢失新事件。

9.8.2.1 定时器溢出中断 (TOF)

TOF 中断操作的细节和方式取决于 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。

9.8.2.1.1 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。

9.8.2.1.2 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) 处改变计数方向从递增转为递减时，TOF 置位。这种情形下，TOF 对应一个 PWM 周期的末尾。

9.8.2.2 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)。

9.8.2.2.1 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。当检测到被选择边沿时，中断标志位被置位。该标志位可通过在 9.8.2 节 [中断操作描述](#) 中的两步流程来清零。

9.8.2.2.2 输出比较事件

若通道被配置为输出比较通道，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 9.8.2 节 [中断操作描述](#) 中的两步流程来清零。

9.8.2.2.3 PWM 结束占空比事件

若通道被配置为 PWM，则可能的操作有两种。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，定时器计数器两次等于通道值寄存器的值。在这种 CPWM 情形下，在占空比周期的开始和结尾时通道标志被置位，此时，定时器计数器的值等于通道寄存器的值。该标志位可通过在 9.8.2 节 [中断操作描述](#) 中的两步流程来清零。

第 10 章

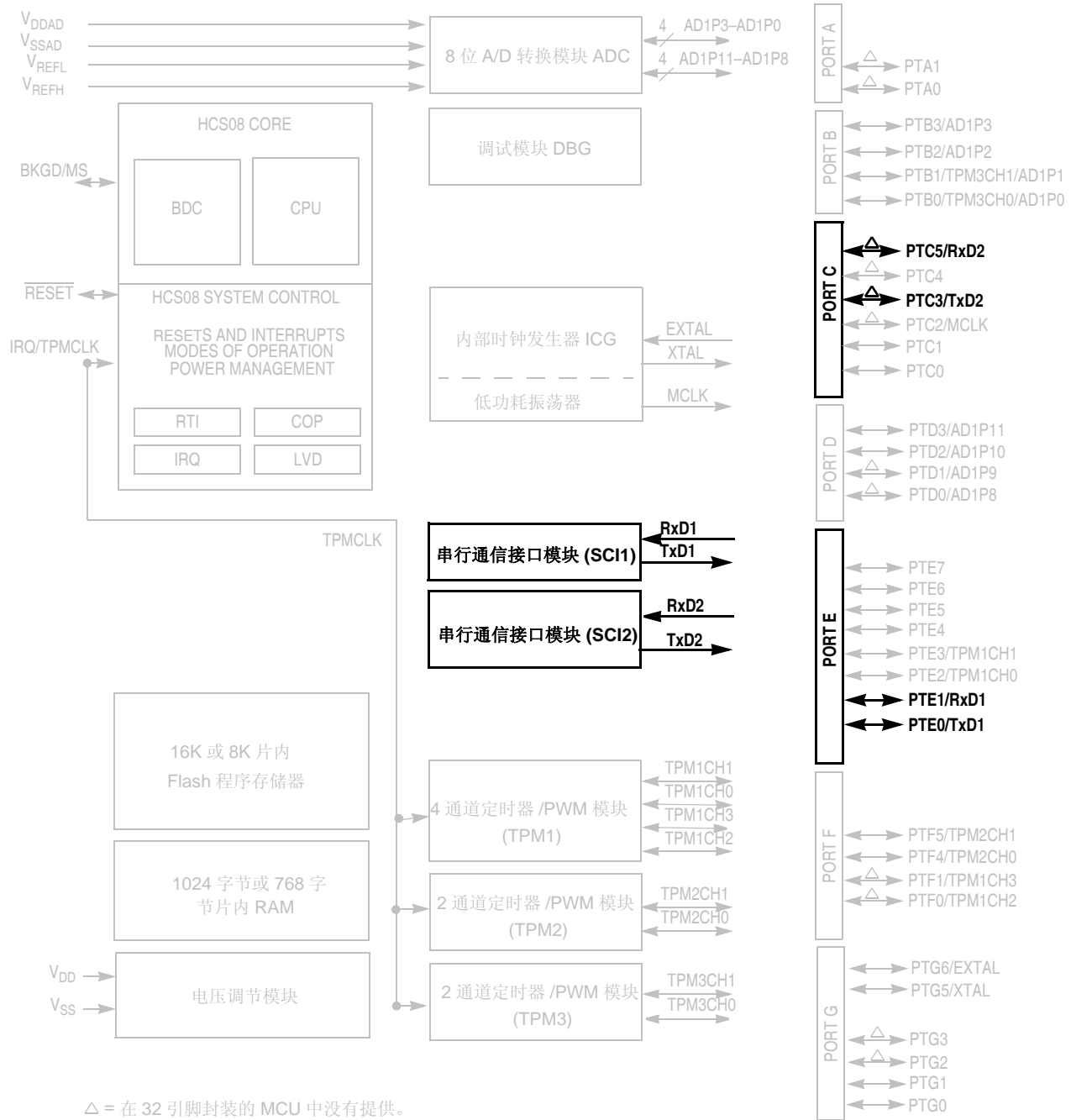
串行通信接口 (S08SCIV4)

10.1 概述

SC667027 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 位模块化的波特率产生器支持超过 115.2 kB 的标准波特率。SCI 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。

本 SCI 提供了多种高级特性，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。



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 10-1. 显亮 SCI 模块和引脚的 SC667027 模块结构图

10.1.1 特性

SCI 模块具有以下特性：

- 全双工，标准的不归零（NRZ）格式。
- 发送器和接收器内具有各自的双缓冲，并可分别使能。
- 可编程的波特率（13 位模因子）。
- 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
 - 间断检测支持 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 位间断字符产生或 11 位的间断字符检测。
- 可选的发送器输出极性。

10.1.2 操作模式

在下列模式中 SCI 操作的细节，参见第 10.3 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

10.1.3 框图

图 10-2 为 SCI 的发送器部分。

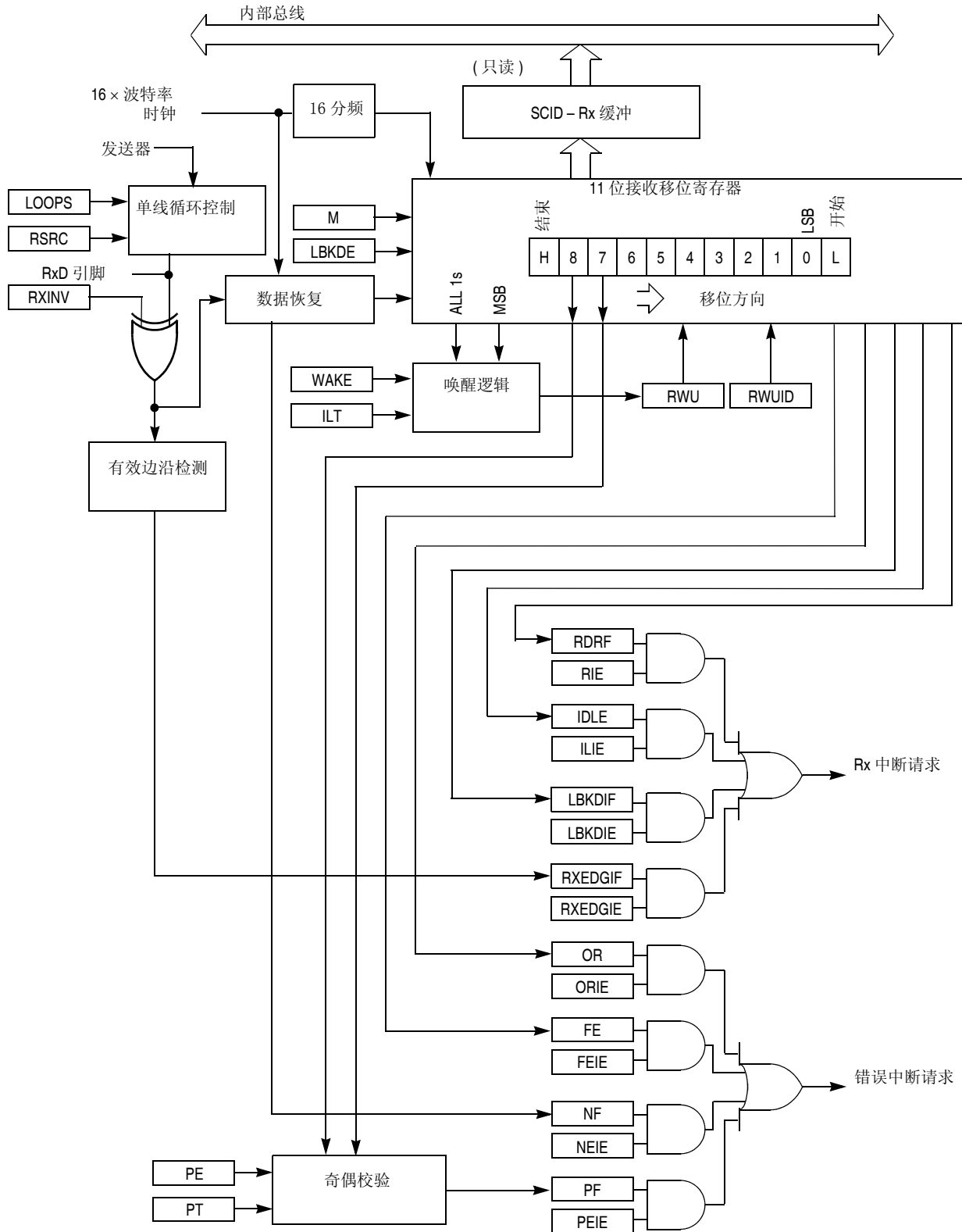


图 10-3. SCI 接收器原理结构图

10.2 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，Freescale 提供一个头文件把它们名称翻译为绝对地址。

10.2.1 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 位的波特率 [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，波特率才改变。

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)。

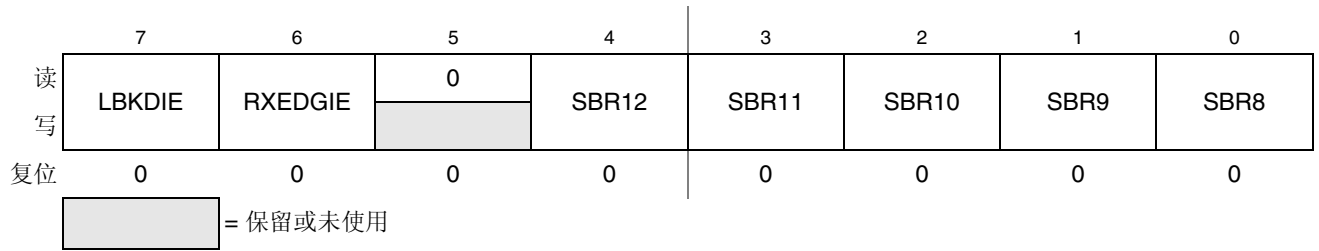


图 10-4. SCI 波特率寄存器 (SCIxBDH)

表 10-1. 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)。其他的 BR 位请参见表 10-2。

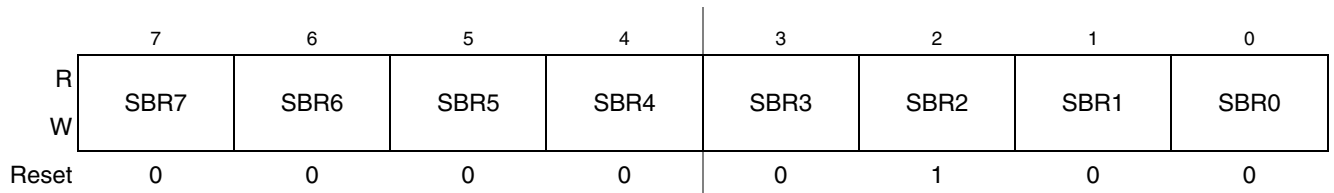


图 10-5. SCI 波特率寄存器 (SCIxBDL)

表 10-2. 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)。其他的 BR 位请参考表 10-1。

10.2.2 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。

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

图 10-6. SCI 控制寄存器 1 (SCIxC1)

表 10-3. 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，此时发送器的输出连接到接收器的输入。 0 正常操作 --- RxD 和 TxD 采用不同的引脚。 1 循环或者单线模式，此时发送器的输出连接到接收器的输入。
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 10.3.3.2 节 接收唤醒。 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 10.3.3.2.1 节 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。

表 10-3. SCIxC1 寄存器域描述

域	描述
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。

10.2.3 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。

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

图 10-7. SCI 控制寄存器 2 (SCIxC2)

表 10-4. 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 10.3.2.1 节 发送间隔和等待空闲。 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，等待空闲或等待间隔符排队完成传输。

表 10-4. SCiXC2 寄存器域描述

域	描述
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 10.3.3.2 节 接收唤醒。 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， 一个间隔符插入传输数据流。只要 SBK=1、 10 或 11 (13 或 14) 个逻辑 0 的间隔符也会插入传输数据流。第二个间隔符可能在软件清 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 10.3.2.1 节 发送间隔和等待空闲。 0 正常发送操作。 1 发送对了终止字符。

10.2.4 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。

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


 = 保留或未使用

图 10-8. SCI 状态寄存器 1 (SCiXS1)

表 10-5. 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， 前导符或间隔符发送时， TC 也置位。 0 发送器忙 (发送数据， 前导符或间隔符)。 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 插入间隔符。
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。

表 10-5. SCIxS1 域描述 (continued)

域	描述
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），这会导致接收器认为检测到空闲，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，从而不会被检测为空闲线。 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，而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。

10.2.5 SCI 状态寄存器 2 (SCiXS2)

该寄存器有 1 个只读状态标志。

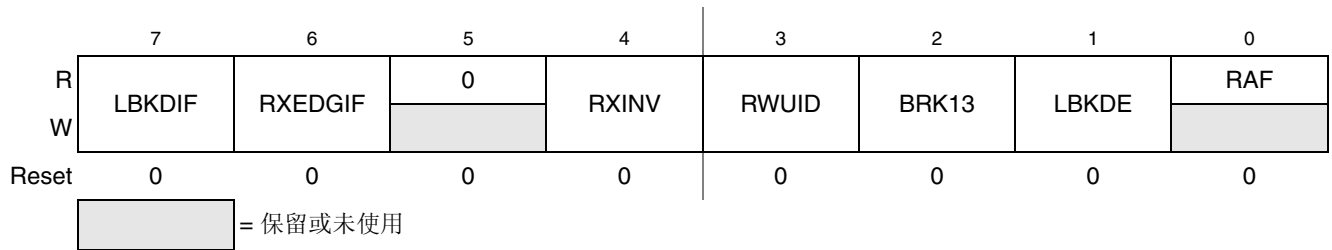


图 10-9. SCI 状态寄存器 2 (SCiXS2)

表 10-6. SCiXs2 域描述

域	描述
7 LBKDIF	LIN 终止检测中断标志——当使能 LIN 间隔检测电路，并且检测到一个 LIN 间隔符，LBKDIF 置位。向 LBKDIF 写 1，LBKDIF 清零。 0 未检测到 LIN 间隔符。 1 检测到 LIN 间隔符。
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 RXINV 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	间隔符长度——BRK13 可以选择更长的发送间隔符长度。该位的状态不影响帧错误的检测。 0 间隔符长度是 10 字节（M=1 则为 11）。 1 间隔符长度是 13 字节（M=1 则为 14）。
1 LBKDE	LIN 间隔检测使能——LBKDE 可以选择更长的间隔符检测长度。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 间隔符的检测长度是 10 字节（M=1 则为 11）。 1 间隔符的检测长度是 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，有必要把间断检测极限值增加一位。在最坏的情况下，LIN 允许的的定时条件可能把 0x00 数据字符看做 10.26 位字节，因为在从机上运行比主机快 14%。这将触发正常的间隔检测电路（用来检测 10 位间隔标志）。当 LBKDE 位置位时，帧错误可以减少，间隔检测阈从 10 位变成 11 位以防止把 0x00 数据字符错误地检测成 LIN 间隔符号。

10.2.6 SCI 控制寄存器 3 (SCiXC3)

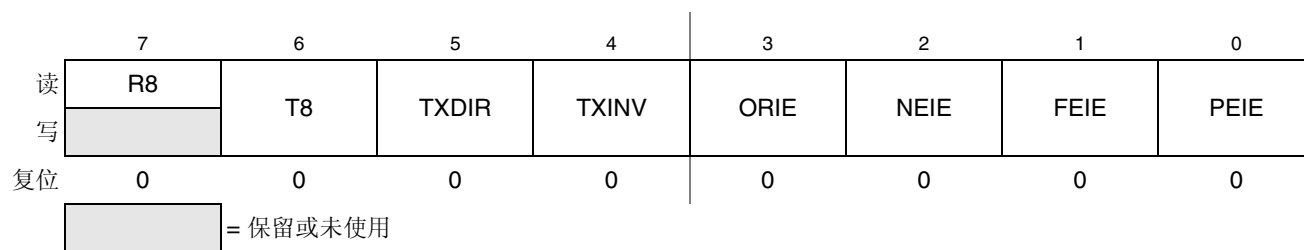


表 10-7. SCI 控制寄存器 3 (SCiXC3)

表 10-8. SCIxCS 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 TXINV ¹	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。

10.2.7 SCI 数据寄存器 (SCIxD)

该寄存器实际上是两个独立的寄存器。读操作读到只读接收数据缓冲区的内容，写操作写入发送数据缓冲区。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。

	7	6	5	4	3	2	1	0
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 10-10. SCI 数据寄存器 (SCIxD)

10.3 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。

10.3.1 波特率产生

如图 10-11 所示，SCI 波特率产生器的时钟源是总线时钟。

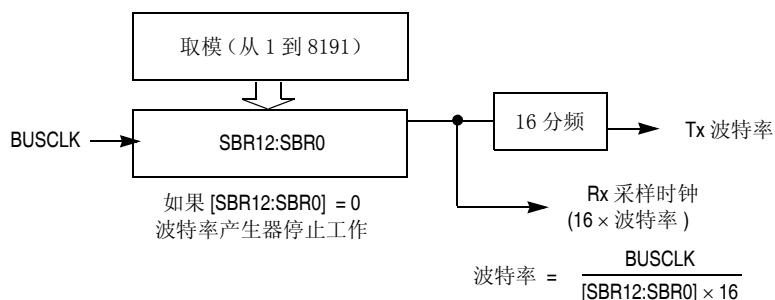


图 10-11. SCI 波特率产生

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。

10.3.2 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。发送器的框图如图 10-2 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。

TxD 引脚发出停止位后，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 并不能即使引脚成为通用 I/O 引脚，必须先完成正在进行的发送过程。这包括正在处理的数据，等待空闲字符和等待间隔符。

10.3.2.1 发送间隔和等待空闲

SCIXC2 的 SBK 用于发送间隔符（原来用于与电传打字机进行通信初始化）。间隔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，则间隔符可以达到 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，就会发送间隔符。如果当间隔符进入移位寄存器（由波特率时钟同步）SBK 仍为 1，则有另外

一个间隔符在等待。如果接收设备是另一个飞思卡尔半导体的 SCI，则接受的 8 位间隔符全 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。

间隔符的长度受 BRK13 和 M 位影响，如表 10-9 所示。

表 10-9. 间隔符长度

BRK13	M	间隔符长度
0	0	10 位
0	1	11 位
1	0	13 位
1	1	14 位

10.3.3 接收功能描述

在该节中，接收器框图（图 10-3）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 SC1xC2 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 10.3.5.1 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 SC1xD 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 10.3.4 节 中断和状态标志。

10.3.3.1 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间隔符,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。

10.3.3.2 接收唤醒

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SCIC2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。

10.3.3.2.1 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RWUIN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。

10.3.3.2.2 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。

10.3.4 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，前导符和间隔符都发送完成，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)。

10.3.5 其他 SCI 功能

下面描述其他的 SCI 功能。

10.3.5.1 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。

10.3.5.2 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。

10.3.5.3 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。

10.3.5.4 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 11 章

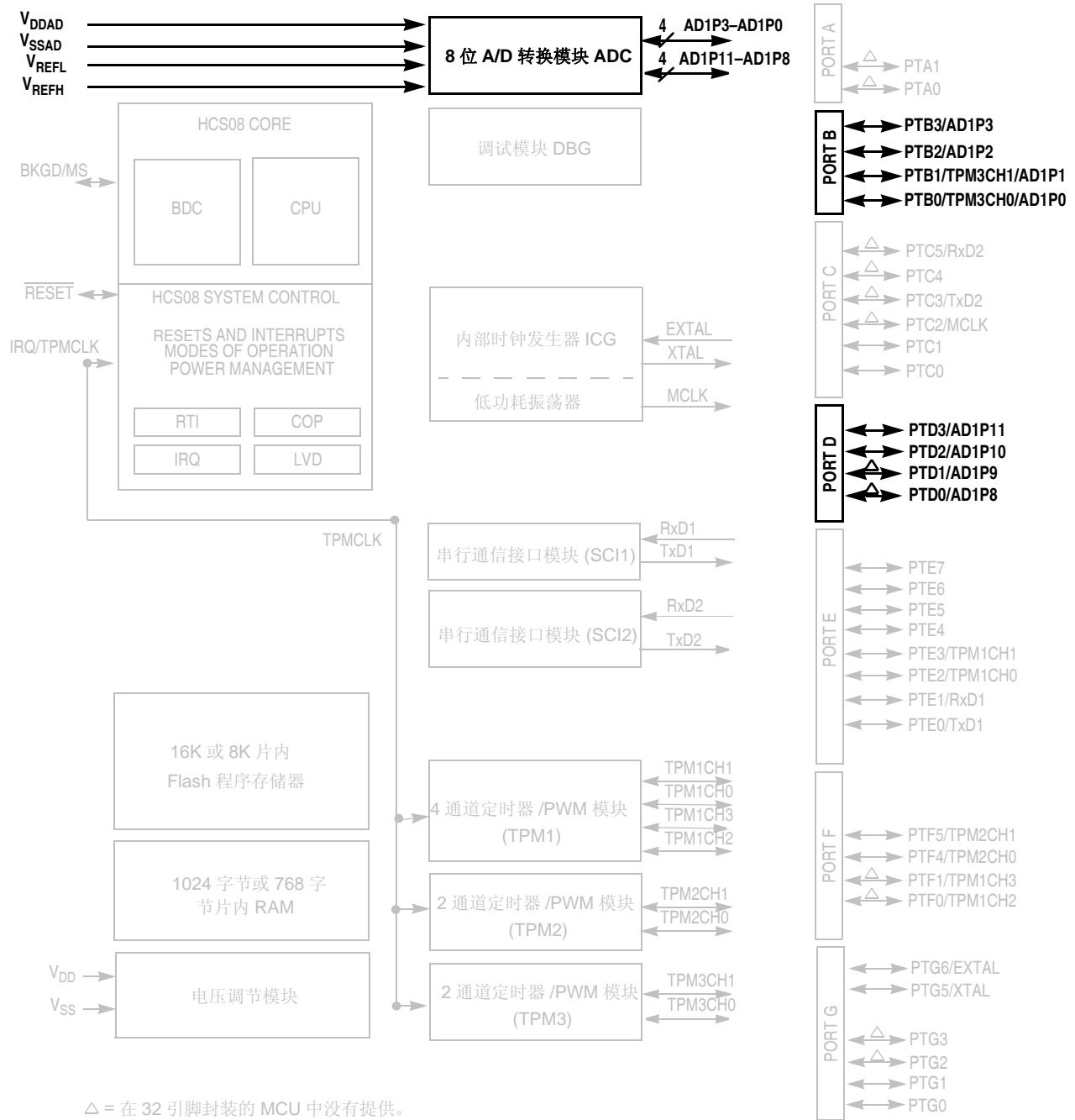
AD 转换器 (S08ADC10V1)

11.1 概述

8 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 SC667027 系列的 MCU 中只实现了 8 个输入（AD0-AD3、AD8-AD11）。这些输入可以通过 ADCH 位选择。如图 11-1 所示，一些输入和 I/O 复用。表 11-1 概述了 SC667027 系列设备的 ADC 中所有通道的分配。

注意

SC667027 的 ADC 只提供 8 位，不支持 8 位和 12 位。本章中关于 8 位和 12 位的描述适用于飞思卡尔提供的其他 8 位微控制器。



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 11-1. 显亮 ADC 模块和引脚的 SC667027 模块结构图

11.2 通道分配

SC667027 系列设备的 ADC 通道分配如表 11-1 所示。未实现的通道在内部连接到 VREFL。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 11-1. ADC 通道分频

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

注意

选择内部的能带隙通道要求 SPMSC1 中的 BGBE=1, 参见 5.9.8 节 系统电源管理状态控制寄存器 1 (SPMSC1)。更多关于能带隙电压参见 A.5 节 DC 特性。

11.2.1 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。SC667027 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 只有在外时钟源允许时才有效，ICG 必须设置成 FBE 或 FEE 模式 (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。

11.2.2 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。

11.2.2.1 模拟引脚使能

SC667027 系列的 ADC 只包含两个模拟引脚使能寄存器，APCTL1 和 APCTL2。

11.2.2.2 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。

11.2.3 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 11-1 提供了温度传感器的近似转换函数。

$$\text{Temp} = 25 - ((V_{\text{TEMP}} - V_{\text{TEMP25}}) \div m) \quad \text{等式 11-1}$$

其中：

- V_{TEMP} 是温度传感器通道在周围温度时的电压。
- V_{TEMP25} 是温度传感器通道在 25 °C 时的电压。
- M 是冷或热温度斜率 ($V/^\circ\text{C}$)。

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 11-1 中使用冷斜率值。如果 V_{TEMP} 小于 V_{TEMP25} 在等式 11-1 中使用热斜率值。

更多使用温度传感器的信息参考 AN3031。

11.2.4 特性

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

11.2.5 框图

图 11-2 提供了 ADC 模块的框图。

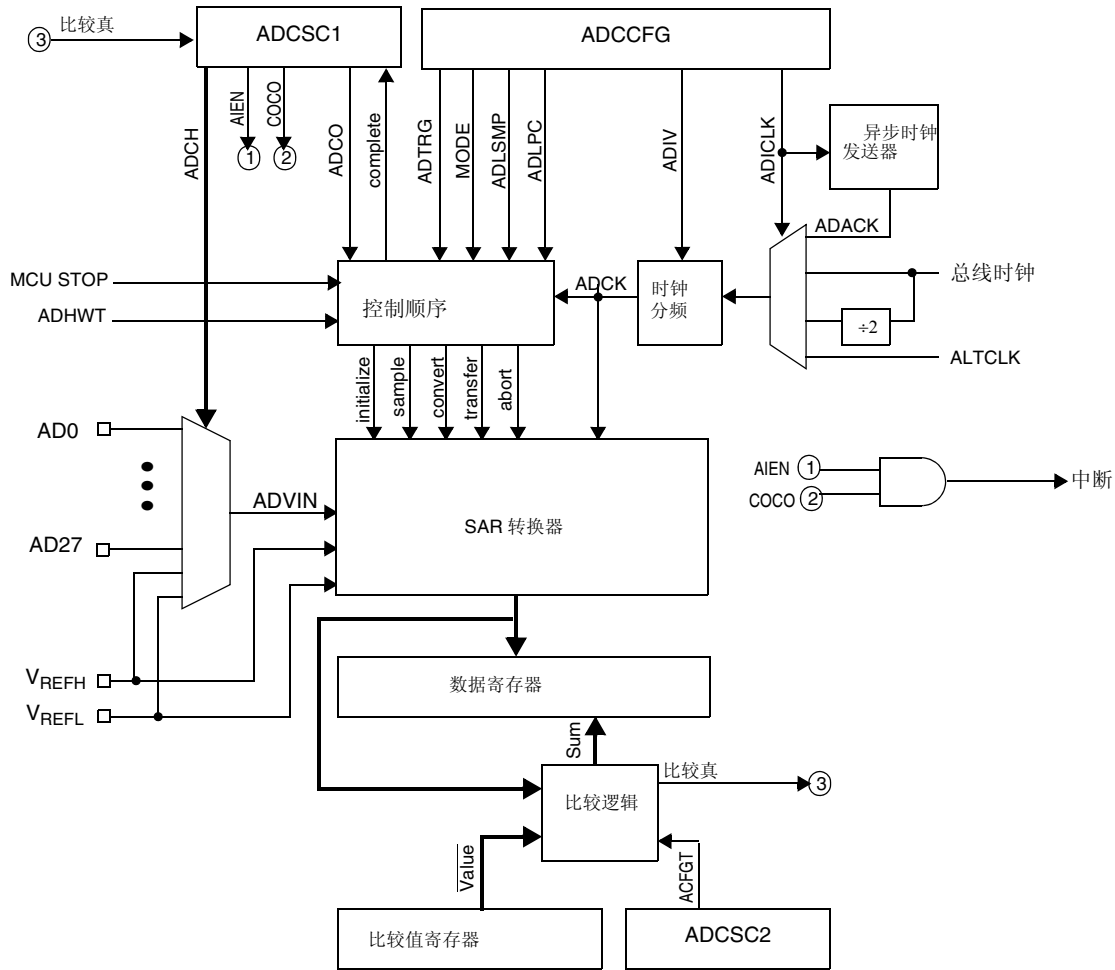


图 11-2. 2 ADC 框图

11.3 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 11-2. 信号属性

名称	功能
AD27-AD0	模拟通道输入
V _{REFH}	高参考电压
V _{REFL}	低参考电压
V _{DDAD}	模拟电压供电
V _{SSAD}	模拟地

11.3.1 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 V_{DD}。如果外部可能，连接到 V_{DDAD} 的引脚和 V_{DD} 到相同电压。外部滤波对 V_{DDAD} 可能是必要的。

11.3.2 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。

11.3.3 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。

11.3.4 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。

11.3.5 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。

11.4 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, $APCTL1$ 、 $APCTL2$ 、 $APCTL3$

11.4.1 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)。

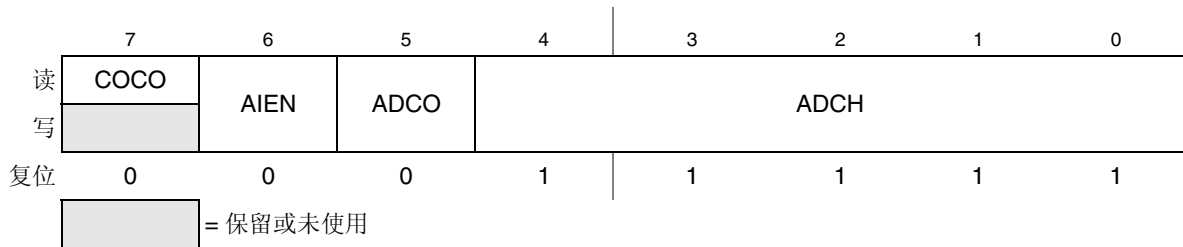


图 11-3. 状态和控制寄存器 ($ADC1SC1$)

表 11-3. 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。输入通道在表 11-4 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。

表 11-4. 输入通道选择

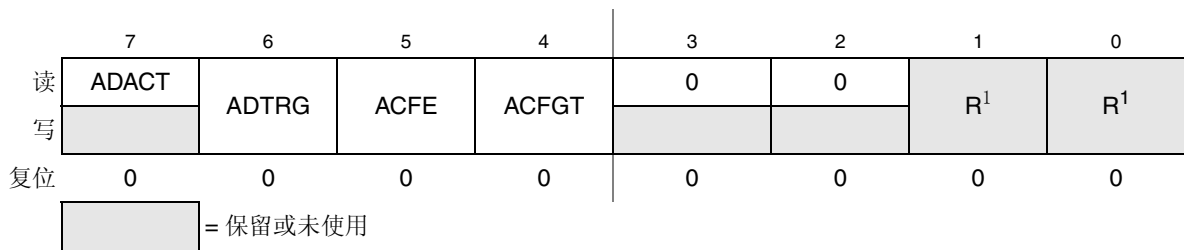
ADCH	输入选择
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7
01000	AD8
01001	AD9
01010	AD10
01011	AD11
01100	AD12
01101	AD13
01110	AD14
01111	AD15
10000	AD16
10001	AD17

表 11-4. 输入通道选择 (continued)

ADCH	输入选择
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

11.4.2 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。



¹ 位 1 和位 0 为保留位，必须写为 0。

图 11-4. 状态和控制寄存器 (ADC1SC2)

表 11-5. 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。

表 11-5. ADC1SC2 寄存器域描述

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。

11.4.3 数据高结果寄存器 (ADC1RH)

ADC1RH 包含 10 位转换结果的高 2 位。当设置为 8 位转换时，ADR8 和 ADR9 都等于 0。每次转换完成，除非自动比较被允许而且不满足比较结果，ADC1RH 将被更新。在 10 位数据模式中，读 ADC1RH 将暂时禁止下一次转换，直到读取了 ADC1RL 中的内容。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。

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

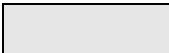
 = 保留或未使用

图 11-5. 数据高结果寄存器 (ADC1RH)

11.4.4 数据低结果寄存器 (ADC1RL)

ADC1RL 包含 10 位转换结果的低 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，读 ADC1RH 将暂时禁止下一次转换，直到读取了 ADC1RL 中的内容。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。

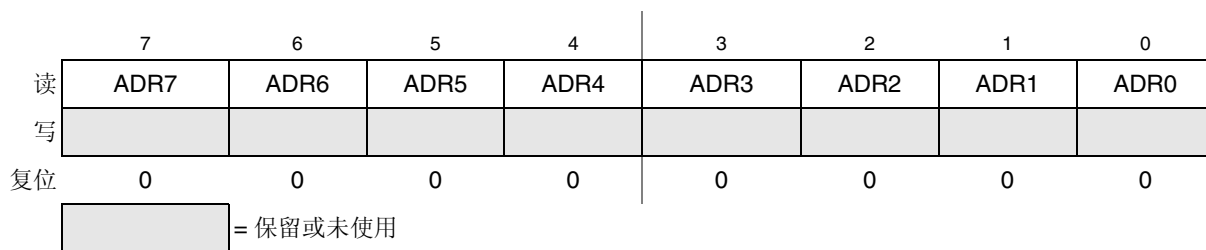


图 11-6. 数据低结果寄存器 (ADC1RL)

11.4.5 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。

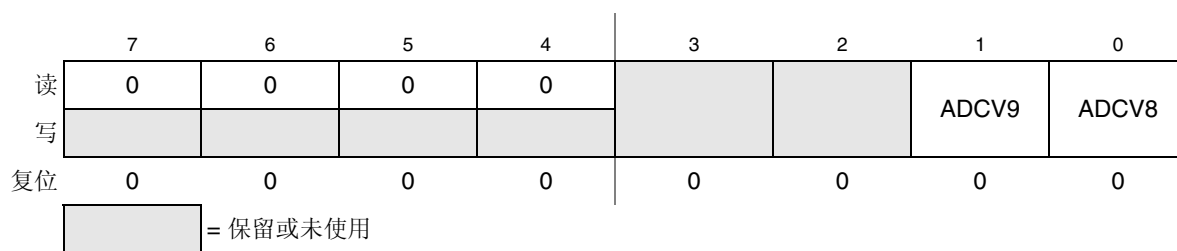


图 11-7. 比较值高寄存器 (ADC1CVH)

11.4.6 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。

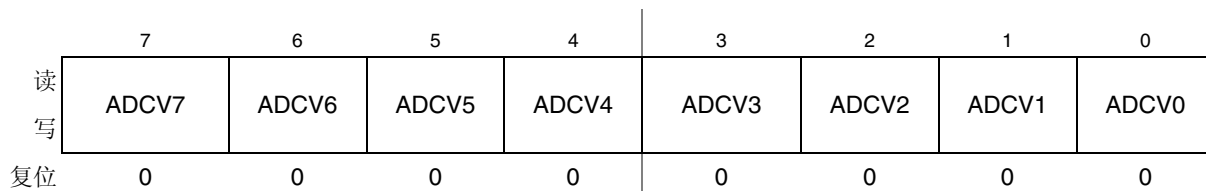


图 11-8. 比较值低寄存器 (ADC1CVL)

11.4.7 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。

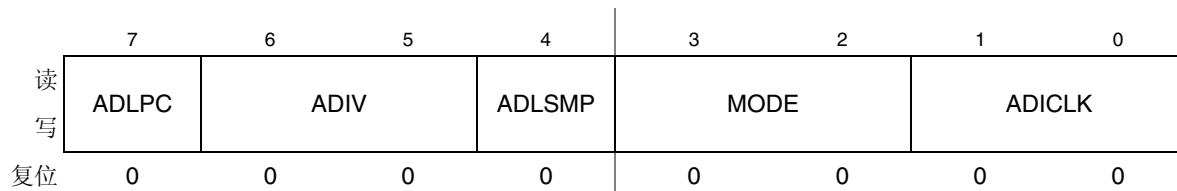


图 11-9. 配置寄存器 (ADC1CFG)

表 11-6. 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 11-7 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输出斜率，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 11-8。
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 11-9。

表 11-7. 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 11-8. 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 11-9. 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)

11.4.8 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。

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

图 11-10. 引脚控制 1 寄存器 (APCTL1)

表 11-10. 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。

11.4.9 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。

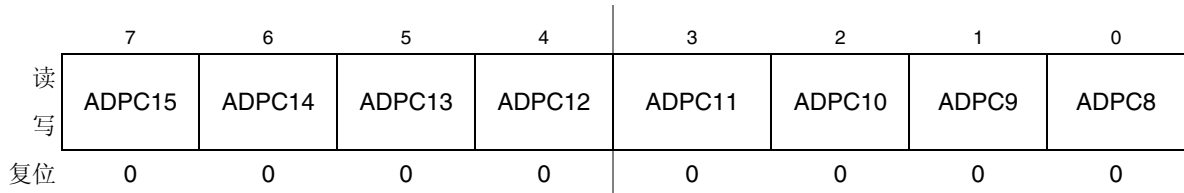


图 11-11. 引脚控制 2 寄存器 (APCTL2)

表 11-11. 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。

11.4.10 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。

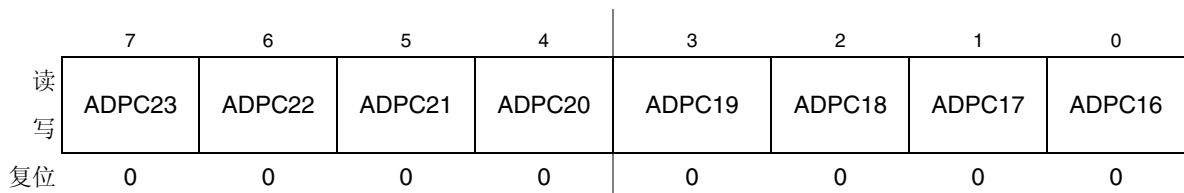


图 11-12. 引脚控制 3 寄存器 (APCTL3)

表 11-12. APCTL3 寄存器域描述

域	描述
7 ADPC23	ADC 引脚控制 23——ADPC23 用于控制和通道 AD23 相关的位。 0 允许 AD23 引脚 I/O 控制。 1 禁止 AD23 引脚 I/O 控制。
6 ADPC22	ADC 引脚控制 22——ADPC22 用于控制和通道 AD22 相关的位。 0 允许 AD22 引脚 I/O 控制。 1 禁止 AD22 引脚 I/O 控制。
5 ADPC21	ADC 引脚控制 21——ADPC21 用于控制和通道 AD21 相关的位。 0 允许 AD21 引脚 I/O 控制。 1 禁止 AD21 引脚 I/O 控制。
4 ADPC20	ADC 引脚控制 20——ADPC20 用于控制和通道 AD20 相关的位。 0 允许 AD20 引脚 I/O 控制。 1 禁止 AD20 引脚 I/O 控制。
3 ADPC19	ADC 引脚控制 19——ADPC19 用于控制和通道 AD19 相关的位。 0 允许 AD19 引脚 I/O 控制。 1 禁止 AD19 引脚 I/O 控制。
2 ADPC18	ADC 引脚控制 18——ADPC18 用于控制和通道 AD18 相关的位。 0 允许 AD18 引脚 I/O 控制。 1 禁止 AD18 引脚 I/O 控制。
1 ADPC17	ADC 引脚控制 17——ADPC17 用于控制和通道 AD17 相关的位。 0 允许 AD17 引脚 I/O 控制。 1 禁止 AD17 引脚 I/O 控制。
0 ADPC16	ADC 引脚控制 16——ADPC16 用于控制和通道 AD16 相关的位。 0 允许 AD16 引脚 I/O 控制。 1 禁止 AD16 引脚 I/O 控制。

11.5 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。

11.5.1 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。

11.5.2 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。

11.5.3 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。

11.5.4 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。

11.5.4.1 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。

11.5.4.2 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。

11.5.4.3 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。

11.5.4.4 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）。

11.5.4.5 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_{ADCD} ）。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 11-13 中总结了不同条件下的最大的总转换时间。

表 11-13. 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	5s+20ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	0	5s+23ADCK +5 总线时钟周期

表 11-13. 不同控制条件的总转换时间 (continued)

单个或第一个连续转换 8 位	11	1	5s+40ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	1	5s+43ADCK +5 总线时钟周期
后来的连续转换 8 位 $f_{\text{BUS}} \geq f_{\text{ADCK}}$	xx	0	17ADCK 周期
后来的连续转换 10 位 $f_{\text{BUS}} \geq f_{\text{ADCK}}$	xx	0	20ADCK 周期
后来的连续转换 8 位 $f_{\text{BUS}} \geq f_{\text{ADCK}}/11$	xx	1	37ADCK 周期
后来的连续转换 10 位 $f_{\text{BUS}} \geq f_{\text{ADCK}}/11$	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：

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

公式 11-2

总线时钟的个数 = $3.5 \mu\text{s} \times 8 \text{ MHz} = 28$ (周期)

注意

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。

11.5.5 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。

注意

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。

11.5.6 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。

11.5.7 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。

11.5.7.1 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。

11.5.7.2 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。

注意

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 11.5.4.2 节 完成转换）清零。

11.5.8 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。

11.6 初始化信息

本节举例说明了用户如何初始化和配置 ADC 模块的一些基本方用法。用户可以灵活地选择配置模块，8 位或 10 位精度，单个或连续转换，循环或中断方式，还有其他选项。在该例中的信息可以参见表 11-7、表 11-8 和表 11-9。

注意

十六进制的前缀是 0x，二进制的前缀是 %，十进制没有前缀。

11.6.1 ADC 模块初始化举例

11.6.1.1 初始化顺序

在 ADC 模块执行转换操作前，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。

11.6.1.2 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。

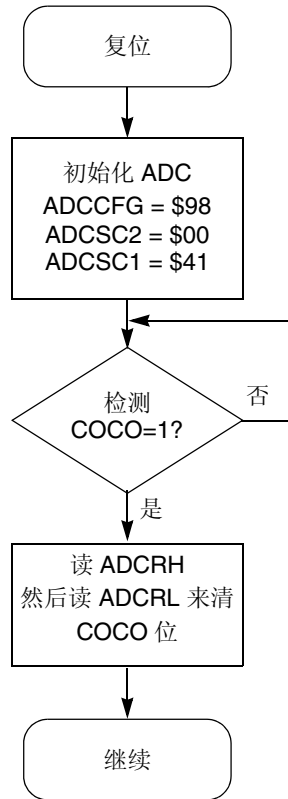


图 11-13. 举例的初始化流程

11.7 应用信息

本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。

11.7.1 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。

11.7.1.1 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 MCU V_{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隔离。

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，谨慎布线避免干扰，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。

11.7.1.2 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 $0.1\mu\text{F}$ 的电容（高频特性）。这个电容连接到 V_{REFH} 和 V_{REFL} 之间，离封装尽可能近。不推荐使用电阻，因为电流导致电压泄露，这可能导致转换错误。该路径上的电磁感应应最小化。

11.7.1.3 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，可以进行转换操作。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 $0.01\mu\text{F}$ 的电容（有高频特性）完全可以满足。这些电容并不是在所有的情况下都需要，但是它们必须放在离封装尽可能近的地方，作为 V_{SSA} 的参考。

为正确转换，输入电压必须在 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 $\$3\text{FF}$ （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。

11.7.2 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。

11.7.2.1 采样错误

为正确转换，输入必须被采样足够长时间才能达到合适的精度。如果最大输入电阻 $7\text{ k}\Omega$ 和输入电容 5.5 pF ，外部模拟源（RAS）的电阻小于 $5\text{ k}\Omega$ ，则采样 $1/4\text{LSB}$ （10 位精度）可以在最小的采样窗口完成（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。

11.7.2.2 引脚漏电流误差

如果外部模拟源（RAS）为高，I/O 引脚上的漏电流导致转换误差。如果在应用中不能容忍这个错误，保持 RAS 小于 $V_{DDAD}/(2N \cdot I_{LEAK})$ 将会后更少漏电流误差（8 位模式 $N = 8$ ，10 位模式 $N = 10$ ）

11.7.2.3 噪音误差

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 $0.1 \mu\text{F}$ 的低内阻电容。
- V_{DDAD} 和 V_{SSAD} 之间有一个 $0.1 \mu\text{F}$ 的低内阻电容。
- 如果电源使用感应隔离，一个 $1 \mu\text{F}$ 的电容放在 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 V_{DD} 噪音但是因为停止复原增加了有效转换时间。
- MCU 处于转换时, 没有 I/O 选择, 输入或输出。

在一些情况, 外部系统行为导致辐射或噪音发射或伴随 ADC 的过多 V_{DD} 噪音。在这些情况, 或当 MCU 不能在等待状态或 I/O 行为停止, 上述操作可能减少影响正确性的噪音:

- 在选择的输入通道和 V_{REFH} 或 V_{REFL} 之间放置一个 $0.01 \mu\text{F}$ 的电容 (CAS) (这将增加噪音问题但是影响基于外部的模拟源电阻的采样率)。
- 求多次转换模拟值的平均。需要四次采样减少 1LSB 的影响以及一次误差。
- 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。

11.7.2.4 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 $d\text{leta}$, 在一个代码到下一个之间。N 位转换器 (N 是 8 或 10) 的理想代码宽度, 定义为 1LSB, 为:

$$1\text{LSB} = (V_{REFH} - V_{REFL}) / 2^N \quad \text{公式 11-3}$$

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 。

11.7.2.5 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:

- 归零误差 (EVS) (也称作偏移量) ——指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, 则采用了然后实际的 \$001 编码宽度和理想情况 (1LSB) 的不同。
- 满标误差 (EFS) ——指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, 采用了实际的 \$001 编码宽度和理想情况 (1LSB) 的不同。
- 微分非线性 (DNL) ——指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) —— 指 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。

11.7.2.6 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。这个误差可以通过重复采样和对结果求均值来减少。另外在 [11.7.2.3 节 噪音误差](#) 节中的技巧可以减少该误差。

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 12 章 开发支持

12.1 介绍

HCS08 中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 提供单线调试接口，与目标连接，通过这个接口可以方便地进行片上闪存和其它非易失性存储器的编程。BDC 也是开发用的主要调试接口，允许以非侵入式方式存取存储器数据和传统调试功能，如 CPU 寄存器修改、断点和单指令跟踪命令等。

在 HCS08 产品系列中，外部引脚不包括地址和数据总线信号（即使在测试模式也不包括）。调试的执行是通过单线背景调式接口向目标 MCU 传输命令来实现的。调试模块提供了一种有选择性地触发和捕获总线信息的方式，这样外部开发系统可以对 MCU 内发生的事件按周期进行重现，而不需要从外部存取 MCU 的地址和数据信号。

12.1.1 强制激活背景调试

ICGLCLK 是 SC667027 的可选 BDC 时钟源。关于此时钟源的详细信息以及如何选择时钟源请参见第 8 章 内部时钟发生器 (S08ICGV4)。

12.1.2 特性

BDC 模块的特性包括:

- 单引脚进行模式选择和背景调试通信
- BDC 的寄存器不位于存储器地址中
- SYNC 命令确定目标通信速率
- 非侵入式命令进行存储器存取
- 供 CPU 寄存器存取的激活背景调试模式命令
- GO 和 TRACE1 命令
- 背景调试命令可以将 CPU 从停止模式或等待模式中唤醒
- BDC 内置一个硬件地址断点
- 如果 BDC 使能, 则振荡器运行在停止模式
- 处于激活背景调试模式时, COP 看门狗禁止

ICE 系统的特性包括:

- 两个触发比较器: 两个地址 + 读 / 写 (R/W) 或一个完整地址 + 数据 + R/W
- 灵活的 8-word x 16-bit FIFO (先进先出) 缓存, 用于捕获信息:
 - 流程变化的地址 或
 - 纯事件数据
- 两个类型的断点:
 - 指令操作码的标记断点
 - 任何地址存取的强制断点
- 九个触发模式:
 - 基本: 只有 A, A 或 B
 - 顺序: A 然后 B
 - 全部: A 和 B 数据, A 和非 B 数据
 - 事件 (存储数据): 纯事件 B, A 然后纯事件 B
 - 范围: 在范围以内 ($A \leq \text{地址} \leq B$), 在范围以外 ($\text{地址} < A$ 或 $\text{地址} > B$)

12.2 背景调试控制器 (BDC)

HCS08 系列中的所有 MCU 都包含一个单线背景调试接口, 它支持片上非易失性存储器的在线编程和先进的非侵入式调试功能。与早期的 8- 位 MCU 的调试接口不同, 这个系统不干扰正常的应用资源。它不使用任何用户存储器或存储器映射中的地址, 也不分享任何片上外设。

BDC 命令分为两大组:

- 激活背景调试模式命令要求目标 MCU 处于激活背景调试模式 (用户程序未运行)。激活背景调试模式命令允许读写 CPU 寄存器, 允许用户一次跟踪一个用户指令, 或从激活背景调试模式进入用户程序。
- 非侵入式命令可以随时执行, 即使用户的程序正在运行。
非侵入式命令允许用户在背景调试控制器中读写 MCU 存储器位置或存取状态和控制寄存器。

一般地, 可以用相当简单的接口盒将来自主机的命令转换为与单线背景调试系统连接所需的串行命令。根据开发工具供应商的不同, 这个接口盒可以采用标准 232 串行端口, 或是并行打印端口, 或是其它类型的

通信端口，如用来与 PC 通信的 USB 接口。这个接口盒一般通过接地、BKGD 引脚、RESET，有时还有 VDD 与目标系统连接。RESET 引脚的开漏连接允许主机强制目标系统复位，这有助于重新获得对已失目标系统的控制，或在片上非易失性存储器重新编程之前，控制目标系统的启动。有时可以用 VDD 来允许接口盒使用目标系统的电源，避免再使用一个电源。但是，如果单独对接口盒供电，它可以连接到一个正在运行的目标系统，而不必强制目标系统复位，否则会干扰正在运行的应用程序。

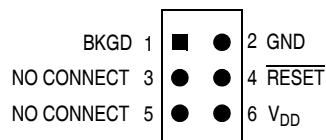


图 12-1. 工具接口

12.2.1 BKGD 引脚描述

BKGD 是单线背景调试接口引脚。这个引脚的主要功能是实现激活背景调试模式命令和数据的双向串行通信。在复位过程中，这个引脚用来选择激活背景调试模式启动或启用用户的应用程序。这个引脚还用来请求定时同步响应脉冲，允许主机开发工具确定背景调试串行通信的正确时钟频率。

BDC 串行通信采用首先引入在微处理器 M68HC12 系列上的定制串行协议。这个协议假定主机知道通信时钟速率，这个速率按目标 BDC 所有通信通过主机启动和控制所有通信，主机驱动高到低边沿发出每个位时间开始信号。命令和数据以最重的位先发 (MSB 先发) 的方式发送。有关通信协议的详细信息，请参见 12.2.2 节 通信详细介绍。

如果主机尝试与 BDC 时钟速率未知的目标 MCU 沟通，可以发送 SYNC 命令给目标 MCU，请求定时同步响应信号，通过这个信号，主机可以判断正确的通信速率。

BKGD 是伪开漏引脚，有一个片上上拉，因此不需要外部上拉电阻。与典型的开漏引脚不同，引脚上的外部 RC 时间常数（受外部容性的影响），在信号上升时间上几乎不起作用。定制协议提供瞬态加速脉冲，强制提高这个引脚的上升时间，而没有驱动电平冲突风险。参见 12.2.2 节 通信详细介绍，了解更多详情。

当没有调试盒连接 6- 引脚的 BDM 接口连接器时，BKGD 的内部上拉会选择正常的操作模式。当调试盒连接到 BKGD 时，可以在 MCU 复位后强制它进入激活背景调试模式。强制激活背景调试的具体条件取决于 HCS08 衍生产品（参见“开发支持”小节的介绍）。不必复位目标 MCU 来通过背景调试接口来与之通信。

12.2.2 通信详细介绍

BDC 串行接口需要外部控制器来生成 BKGD 引脚上的下降沿，指示每个位时间的开始。无论数据是发送或接收，外部控制器都会提供这个下降边沿。

BKGD 是伪开漏引脚，可以被外部控制器或 MCU 来驱动。数据以 MSB 先发的形式且以每位 16 个 BDC 时钟周期的速率（标定速率）发送。如果来自主机的下降边沿之间产生 512 BDC 时钟周期，则该接口超时。如果出现超时，任何正在进行的 BDC 命令被中止，对目标 MCU 系统的存储器或操作模式没有影响。

定制串行协议要求调试盒知道目标 BDC 通信时钟速率。

BDC 状态和控制寄存器中的时钟开关 (CLKSW) 控制位允许用户选择 BDC 时钟源。BDC 时钟源可以是总线，或备用的 BDC 时钟源。

BKGD 引脚可以接收高或低电平，或发送高或低电平。下图显示了每种情况的时序。接口时序与目标 BDC 中的时钟同步，但是与外部主机异步。显示的内部 BDC 时钟信号是计数周期的参考。

图 12-2 显示了外部主机将逻辑 1 或 0 发送到目标 HCS08 MCU 的 BKGD 引脚。主机与目标异步，因此主机生成的 BKGD 下降边沿与目标所认为的位时间起始点有 0- 到 -1 周期的延迟。10 个目标 BDC 时钟周期后，目标获得 BKGD 引脚的电平。一般地，主机在主机到目标方向的传输过程中驱动 BKGD 引脚，以加快上升边沿。由于目标在主机至目标方向的传输周期中不驱动 BKGD 引脚，因此没有必要在此期间将线路作为开漏信号。

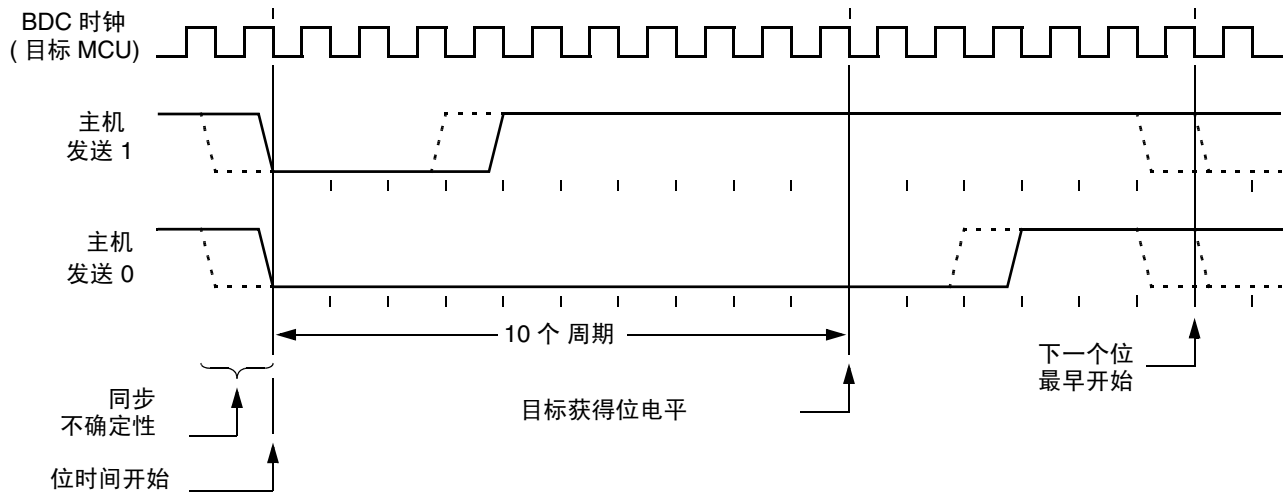


图 12-2. BDC 主机到目标方向串行位时序

图 12-3 显示主机从目标 HCS08 MCU 收到逻辑 1。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机保持低 BKGD 引脚足够长的时间，使目标识别它（至少两个目标 BDC 周期）。主机必须在目标 MCU 在其认为的位计时开始后驱动瞬时高态加速脉冲七个周期前，释放低电平驱动。主机应该在其启动位时间约 10 个周期后采样位电平。

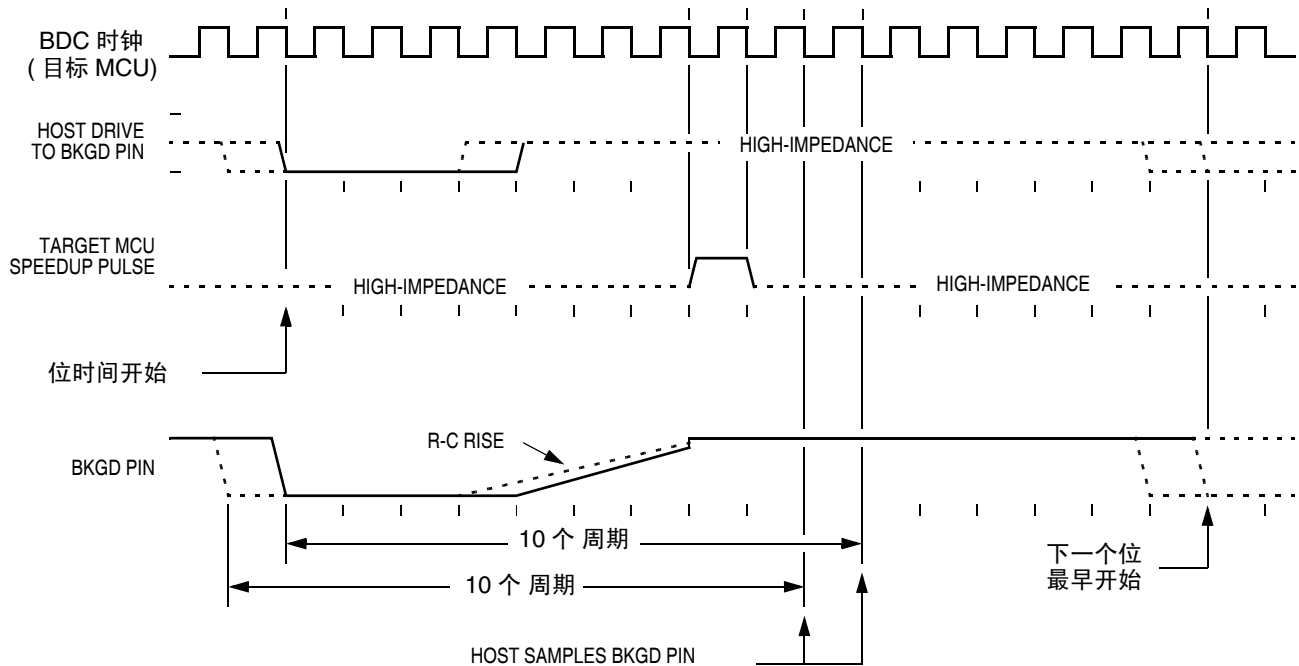


图 12-3. BDC 目标 - 到 - 主机串行位时序 (逻辑 1)

图 12-4 显示了主机从目标 HCS08 MCU 收到逻辑 0。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机启动位时间，但是目标 HCS08 MCU 完成它。由于目标希望主机接收逻辑 0，它保持低 BKGD 引脚 13 个 BKGD 引脚周期，然后驱动引脚置高，加速上升沿。主机在启动位时间约 10 个周期后采样位电平。

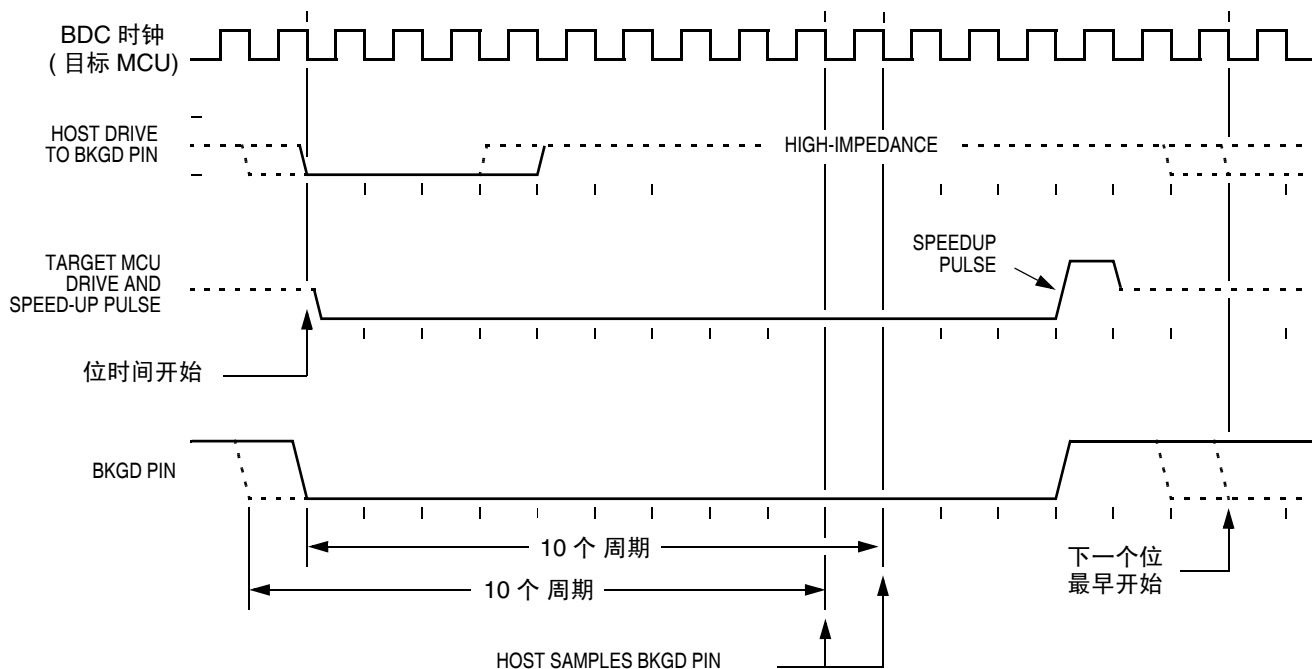


图 12-4. BDM 目标 - 到 - 主机串行位时序 (逻辑 0)

12.2.3 BDC 命令

BDC 命令以串行形式从主机发送到目标 HCS08 MCU 的 BKGD 引脚。所有命令和数据都采用定制 BDC 通信协议以 MSB-先发的形式发送。激活背景调试模式命令要求目标 MCU 当前处于激活背景调试模式，而非侵入式命令可以随时发出，无论目标 MCU 是处于激活背景调试模式还是运行用户应用程序。

表 12-1 显示了所有 HCS08 BDC 命令，并简要描述了它们的编码结构，以及每个命令的含义。

12.2.3.1 编码结构术语

表 12-1 中所用的术语描述了 BDC 命令的编码结构。

- 命令在主机 - 到 - 目标方向上始于一个 8- 位十六进制命令代码 (MSB 先发)
- / = 将命令的各部分分开
 - d = 延迟 16 个目标 BDC 时钟周期
 - AAAA = 主机 - 到 - 目标方向上的一个 16- 位地址
 - RD = 目标 - 到 - 主机方向上的 8- 位读数据
 - WD = 主机 - 到 - 目标方向上的 8- 位写数据
 - RD16 = 目标 - 到 - 主机方向上的 16 位读数据
 - WD16 = 主机 - 到 - 目标方向上的 16 位写数据
 - SS = 目标 - 到 - 主机方向 (STATUS) 上的 BDCSCR 内容
 - CC = 主机 - 到 - 目标方向 (CONTROL) 方向上的 8 位写数据
 - RBKP = 目标 - 到 - 主机方向 (从 BDCBKPT 断点寄存器) 上的 16 位读数据
 - WBKP = 主机 - 到 - 目标方向 (至 BDCBKPT 断点寄存器) 16 位写数据

表 12-1. BDC 命令一览

命令术语	激活 BDM/ 非侵入式	编码结构	描述
SYNC	非侵入式	n/a ¹	要求定时参考脉冲来确定目标 BDC 通信速率
ACK_ENABLE	非侵入式	D5/d	使能响应协议。参见飞思卡尔文档编号 HCS08RMv1/D。
ACK_DISABLE	非侵入式	D6/d	禁止响应协议。参见飞思卡尔文档编号 HCS08RMv1/D。
BACKGROUND	非侵入式	90/d	如果使能，则进入激活背景调试模式（如果 ENBDM 位等于 0，则忽略）
READ_STATUS	非侵入式	E4/SS	从 BDCSCR 读取 BDC 状态
WRITE_CONTROL	非侵入式	C4/CC	向 BDCSCR 中写入对 BDC 的控制
READ_BYTE	非侵入式	E0/AAAA/d/RD	从目标存储器读取字节
READ_BYTE_WS	非侵入式	E1/AAAA/d/SS/RD	读字节和报告状态
READ_LAST	非侵入式	E8/SS/RD	从地址重新读字节，仅读和报告状态
WRITE_BYTE	非侵入式	C0/AAAA/WD/d	将字节写入到目标存储器
WRITE_BYTE_WS	非侵入式	C1/AAAA/WD/d/SS	写入字节和报告状态
READ_BKPT	非侵入式	E2/RBKP	读 BDCBKPT 断点寄存器
WRITE_BKPT	非侵入式	C2/WBKP	写 BDCBKPT 断点寄存器
GO	激活 BDM	08/d	从 PC 当前的地址执行用户应用程序
TRACE1	激活 BDM	10/d	在 PC 的地址跟踪 1 条用户指令，然后返回到激活后台模式
TAGGO	激活 BDM	18/d	与 GO 相同，但激活外部标签（HCS08 器件没有外部标签引脚）
READ_A	激活 BDM	68/d/RD	读累积器 (A)
READ_CCR	激活 BDM	69/d/RD	读条件代码寄存器 (CCR)
READ_PC	激活 BDM	6B/d/RD16	程序计数器 (PC)
READ_HX	激活 BDM	6C/d/RD16	读 H 和 X 寄存器对 (H:X)
READ_SP	激活 BDM	6F/d/RD16	读堆栈指针 (SP)
READ_NEXT	激活 BDM	70/d/RD	以 1 为基数递增 H:X，然后读位于 H:X 的存储器字节
READ_NEXT_WS	激活 BDM	71/d/SS/RD	以 1 为基数递增 H:X，然后读位于 H:X 的存储器字节。报告状态和数据。
WRITE_A	激活 BDM	48/WD/d	写累积器 (A)
WRITE_CCR	激活 BDM	49/WD/d	写条件代码寄存器 (CCR)
WRITE_PC	激活 BDM	4B/WD16/d	写程序计数器 (PC)
WRITE_HX	激活 BDM	4C/WD16/d	写 H 和 X 寄存器对 (H:X)
WRITE_SP	激活 BDM	4F/WD16/d	写堆栈指针 (SP)
WRITE_NEXT	激活 BDM	50/WD/d	以 1 为基数递增 H:X，然后写位于 H:X 的存储器字节。
WRITE_NEXT_WS	激活 BDM	51/WD/d/SS	以 1 为基数递增 H:X，然后写位于 H:X 的存储器字节。报告状态和数据。1 SYNC 命令是特殊操作，不需要命令代码。

¹ SYNC 命令是特殊操作，不需要命令代码。

SYNC 命令与其它 BDC 命令不同，因为主机不必要知道 BDC 通信的正确通信速率，直到它分析完 SYNC 命令的响应后。

要发出 SYNC 命令，主机：

- 保持 BKGD 引脚为低电平至少 128 周期，而且是以最慢的 BDC 时钟来计（最慢的时钟一般是参考振荡器 /64 或自时钟速率 /64。）
- 驱动 BKGD 达到高电平，实现瞬态加速，快速上升时间（这个加速脉冲一般是系统中最快的时钟的一个周期）
- 去除 BKGD 引脚的所有驱动，这样它可回复到高阻抗。
- 监视 BKGD 引脚得到同步响应脉冲

当检测到主机的 SYNC 请求（比在正常 BDC 通信过程中发生的慢时钟要长），则目标：

- 等待 BKGD 返回到逻辑高电平
- 延迟 16 个周期，允许主机停止驱动高电平加速脉冲
- 驱动 BKGD 低态 128 BDC 时钟周期
- 驱动一个周期的高电平加速脉冲，在 BKGD 上实现快速上升时间
- 去除 BKGD 引脚的所有驱动，这样它可回复到高阻抗。

主机测量这个 128 周期的响应脉冲的低电平时间，判断速率，进行后续的 BDC 通信。主机一般可以确定正确的通信速率，与实际目标速率的误差只有百分之几，通信协议能够接受百分之几的速率误差。

12.2.4 BDC 硬件断点

BDC 包括一个相对简单的硬件断点，将 CPU 地址总线与 BDCBKPT 寄存器中的 16- 位匹配值进行比较。这个断点可以生成强制断点或标记断点。强制断点使 CPU 在存取断点地址后的第一个指令边界进入激活背景调试模式。标记的断点使指令操作码在断点地址被标记，这样当 CPU 到达指令队列的终点时，将进入激活后台模式，而不是执行该指令。这意味着标记的断点可能放置在指令操作代码的地址上，而强制断点可以设置在任意地址。

BDC 状态和控制寄存器 (BDCSCR) 中的断点使能 (BKPTEN) 控制位用来激活断点逻辑 (BKPTEN = 1)。当 BKPTEN = 0（复位后它的默认值），断点逻辑禁止，无论其它 BDC 断点中的值是多少，也不管控制位如何，均不请求断点。BDCSCR 中的强制 / 标记选择 (FTS) 控制位用来选择强制 (FTS = 1) 或标记 (FTS = 0) 类型断点。

片上调试模块 (DBG) 包括两个额外的硬件断点的电路，这两个硬件断点比 BDC 模块中的简单断点更灵活。

12.3 片上调试系统 (DBG)

由于 HCS08 器件没有外部地址和数据总线，在线仿真器最重要的功能已经构建在 MCU 的芯片上。这种调试系统包含可以灵活地存储地址或数据总线信息的 8- 级 FIFO，和一个确定何时捕获总线信息以及捕获哪些总线信息的灵活触发系统。这个系统依赖单线背景调试系统来存取调试控制寄存器，读取 8 级阶 FIFO 的结果。

调试模块包括控制和状态寄存器，可以在用户存储器映射中存取。这些寄存器位于高地址寄存器空间中，避免使用宝贵的直接页面存储器空间。

大多数调试模块的功能在开发过程使用，用户程序很少存取调试模块的任何控制和状态寄存器。一个例外就是调试系统可以提供一种手段来实施某种形式的 ROM 补丁。12.3.6 节 [硬件断点](#) 中对此有更详细的描述。

12.3.1 比较器 A 和 B

两个 16- 位比较器 (A 和 B) 可以选择用 R/W 信号或一个操作码跟踪电路来鉴定。比较器单独的控制位允许你忽略每个比较器的 R/W。操作码跟踪电路可选地允许你规定，如果操作码在规定的地址实际执行，而不是只从存储器读到指令队列中，则触发将发生。比较器还能够进行庞大的比较，支持范围内和范围外触发模式。在所有 BDC 存取过程中，比较器临时禁止。

比较器 A 总是与 16- 位 CPU 地址相关联。比较器 B 根据所选的触发模式比较 CPU 地址或 8- 位 CPU 数据总线。由于 CPU 数据总线分为只读数据和写数据总线，RWAEN 和 RWA 控制位有一个额外的目的，在完整地地址加上数据比较中，它们被用来确定其中哪些总线用在比较器 B 数据总线比较中。如 RWAEN = 1 (激活)，RWA = 0 (写)，则使用 CPU 的写数据总线，否则用 CPU 的只读数据。

当前选择触发模式确定当比较器检测到合格的匹配条件时，调试器逻辑做什么。匹配可以导致以下情况：

- 生成 CPU 断点
- 将数据总线值存储到 FIFO 中
- 开始将流变化地址存储到 FIFO 中 (开始类型跟踪)
- 停止将流变化地址存储到 FIFO 中 (结束类型跟踪)

12.3.2 总线捕获信息和 FIFO 操作

使用 FIFO 的通常方式是建立触发模式和其它控制选项，然后打开调试器。当 FIFO 填满后，或调试器停止将数据存储到 FIFO 后，你可以按信息存储的顺序从中读取信息。状态位指示数据所在的 FIFO 中的有效信息的字数。如果在满 (CNT = 1:0:0:0) 之前将 ARM 写为 0，以人工停止跟踪，信息移动一个位置，主机必须执行 ((8 - CNT) - 1)FIFO 虚读操作，使信息进入到 FIFO 中的第一个重要入口。

在大多数触发模式中，存储在 FIFO 中的信息包含 16- 位流变化地址。在这些情况中，先读 DBGFH 然后读 DBGFL，从 FIFO 中获得一个一致的信息字。读 DBGFL (FIFO 数据端口的低阶字节) 会使 FIFO 移动，这样下一个信息字可以在 FIFO 数据端口提供。在纯事件触发模式 (参见 12.3.5 节 [触发模式](#)) 中，8 位数据信息存储在 FIFO 中。在这些情况中，FIFO(DBGFH) 的上半部分没有被使用，仅仅通过读 DBGFL 来从 FIFO 中读出数据。每次读 DBGFL 时，FIFO 都会移动，这样通过 DBGFL 的 FIFO 数据端口可以获得下一个数据值。

在触发模式中，FIFO 保存流变化地址，CPU 地址与 FIFO 的输入端有一个延迟。由于这个延迟，如果触发事件本身是一个流变化地址或在触发事件启动 FIFO 后下两个周期中出现了流变化地址，它将不保存在 FIFO 中。如果是结束 - 跟踪的情况，当触发事件是一个流变化，则它将保存为运行的调试器的最后一个流变化入口。

当调试器没有打开时，FIFO 还可以用来生成所执行指令地址的分析。当 ARM = 0, 读 DBGFL 会使最近获取的操作码的地址保存在 FIFO 中。采用分析功能，主机调试器将从 FIFO 中读取地址，即以常规的间隔先读 DBGFH 然后读 DBGFL。前 8 个值将被丢弃，因为它们对应于初始需要填充 FIFO 的 8 个 DBGFL 读取。DBGFH 和 DBGFL 的其它周期读取则返回关于所执行指令的延迟信息，这样主机调试器可以对执行指令地址进行分析。

12.3.3 流变化信息

为了减少存储在 FIFO 中的信息数量，只保存与使正常的指令执行顺序发生变化的指令相关的信息。知道存储在目标系统中的源和对象代码程序后，外部调试器可以通过来自 FIFO 中存储的大量流变化信息的许多指令来重现执行路径。

对于采用了分支的条件分支指令（分支条件为真），则保存源地址（条件分支操作码的地址）。由于 BRA 和 BRN 指令不是条件的，这些事件不会使流变化信息存储在 FIFO 中。

间接 JMP 和 JSR 指令采用 H:X 间址寄存器对的当前内容，确定目的地址，这样调试系统为任何间接 JMP 或 JSR 保存运行时的目的地址。对于中断，RTI 或 RTS, 目的地址作为流变化信息存储在 FIFO 中。

12.3.4 标记 vs. 强制断点和触发器

标记一词指当指令操作码被取到指令队列时识别它，但是不采取任何其它操作，直到且除非指令被 CPU 真正执行。这种区分非常重要，因为任何因跳转、分支、子例程调用、或中断而发生的流变化都会导致一些指令被取到指令队列，未执行就被丢弃。

强制类型的断点等待当前指令完成，然后执行断点请求操作。通常操作是进入激活背景调试模式，而不是继续用户应用程序中的下一个指令。

标记 vs. 强制这一术语在调试模块的两种情况下使用。第一种情况指从调试模块向 CPU 发送断点请求。第二种情况指从比较器向调试控制逻辑发送匹配信号。当标记类断点发送给 CPU 时，信号与操作码一起进入指令队列，这样当这个操作码被执行时，CPU 将有效地用 BGND 操作码代替被标记的操作码，这样 CPU 进入激活背景调试模式，而不是执行被标记的指令。当 DBGTT 寄存器中的 TRGSEL 控制位被设置为选择标记类操作，比较器 A 或 B 的输出被调试模块中的逻辑块鉴定，这个逻辑块跟踪操作码，如果比较地址的操作码被实际执行，则只向该调试器生成一个触发。每个比较器都有单独的操作码跟踪逻辑，这样整个指令队列一次不只一个比较事件被跟踪。

12.3.5 触发模式

触发模式控制调试器运行的整体行为。DBGTT 寄存器中的 4- 位 TRG 字段选择九个触发模块中的一个。当 DBGTT 寄存器中的 TRGSEL = 1, 比较器的输出必须在触发 FIFO 操作前通过操作码跟踪电路传播。DBGTT 中的 BEGIN 位选择当检测到合格的触发时 FIFO 是否开始存储数据（开始跟踪），或 FIFO 从其打开之时开始循环存储数据，直到检测到合格的触发（结束触发）。

将 1 写入到寄存器中的 ARM 位便可启动调试运行，它设置 DBGS 中的 ARMF 标记，并清除 AF 和 BF 标记及 CNT 位。开始跟踪调试运行当 FIFO 满时结束。结束跟踪运行则在所选触发事件发生时结束。任何调试运行均可通过将 0 写入到 DBGIC 中的 ARM 或 DBGEN 位停止。

除纯事件模式外的所有触发模式中，FIFO 都存储流变化地址。在纯事件触发模式中，FIFO 将数据存储在 FIFO 的八低八位。

控制位在纯事件触发模式中被忽略，而且所有这样的调试运行都是开始类型跟踪。当 TRGSEL = 1 选择操作码获取触发器，没有必要在比较中使用 R/W，因为操作码标签只应用于操作码获取，而这一直都是读周期。在采用全模式触发器时，规定 TRGSEL = 1 也是不正常的，因为操作码的值通常在特定的地址可以知道。

下面的触发模式描述只说明了导致触发的主要比较器条件。比较器 A 或 B 通常都可以被 R/W 进一步鉴定，通过将 RWAEN (RWBEN) 和相应的 RWA (RWB) 值设置为与 R/W 相匹配。如果 BRKEN = 1，来自比较器的带可选 R/W 鉴定的信号，用来请求 CPU 断点，TAG 决定 CPU 请求是标记请求还是强制请求。

只 A — 当地址匹配比较器 A 的值时触发

A 或 B — 当地址匹配比较器 A 或 B 的值时触发

A 然后 B — 当地址匹配比较器 B 但只能在另一个周期的地址匹配比较器 A 的值以后，触发。可能在 A 匹配后 B 匹配前有许多周期。

A 和 B 数据 (全模式) — 这称为全模式，因为地址，数据和 R/W (可选) 必须在同一个总线周期内匹配，才能产生触发事件。比较器 A 检查地址，比较器的低阶字节检查数据，如果 RWAEN = 1，R/W 对照 RWA 进行检查。比较器 B 的高半部分没有使用。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

A 但非 B 数据 (全模式) — 地址必须匹配比较器 A，数据必须不能匹配比较器 B 的低阶部分，如果 RWAEN = 1，R/W 必须匹配 RWA。所有三个条件必须在同一个总线周期中达到才能引起触发。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

纯事件 B (存储数据) — 当地址每次匹配比较器 B 的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。

A 然后纯事件 B (存储数据) — 当地址匹配比较器 A 中的值后，每次地址匹配比较器 B 中的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。

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

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

12.3.6 硬件断点

DBGIC 寄存器中的 BRKEN 控制位可以设置为 1，来允许使用 12.3.5 节 触发模式所描述的任何触发条件，向 CPU 生成硬件断点请求。DBGIC 中的 TAG 控制断点请求是否处理为标记类断点或强制类断点。标记断点使当前的操作码进入指令队列时被标记。如果标记的操作码达到队列的末端，CPU 执行 BGND 指令，进入激活背景调试模式，而不是执行被标记的操作码。强制类断点使 CPU 完成当前指令，然后进入激活背景调试模式。

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

12.4 寄存器定义

本小节描述了 BDC 和 DBG 寄存器及控制位。

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

12.4.1 BDC 寄存器和控制位

BDC 有两个寄存器:

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

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

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

12.4.1.1 BDC 状态和控制寄存器 (BDCSCR)

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

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
W								
正常 复位	0	0	0	0	0	0	0	0
在激活 BDM 中复位	1	1	0	0	1	0	0	0


 = 未实施或预留

图 12-5. BDC 状态和控制寄存器 (BDCSCR)

表 12-2. BDCSCR 寄存器字段描述

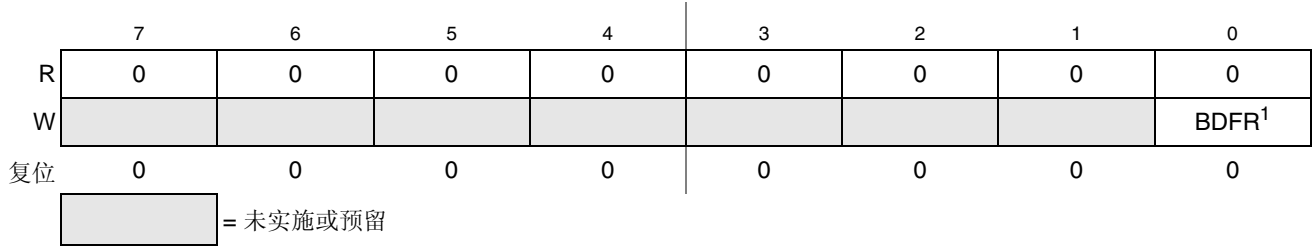
字段	描述
7 ENBDM	激活 BDM (允许激活背景调试模式) — 一般而言, 这个位在调试开始后不久, 或只要调试主机复位目标, 由调试主机写为 1, 并保留 1, 直到通过正常的复位清除它。 0 BDM 不能激活 (非侵入式命令仍然被允许) 1 BDM 可以激活, 允许激活后台模式命令
6 BDMACT	背景调试模式激活状态 — 这是只读状态位。 0 BDM 未激活 (用户应用程序运行) 1 BDM 激活并等待串行命令
5 BKPTEN	BDC 断点激活 — 如果这个位清零, BDC 断点处于处活状态, FTS (强制标签选择) 控制位和 BDCBKPT 匹配寄存器被忽略。 0 BDC 断点禁止 1 BDC 断点激活
4 FTS	强制 / 标签选择 — 当 FTS = 1, 只要 CPU 地址总线匹配 BDCBKPT 匹配寄存器, 则请求断点。当 FTS = 0, CPU 地址总线与 BDCBKPT 寄存器之间的匹配会造成获取的操作码被标记。如果标记的操作码到达指令队列的末端, CPU 则进入激活后台模式, 而不是执行标记的操作码。 0 在断点地址标记操作码, 如果 CPU 试图执行该指令, 则进入激活后台模式 1 断点匹配强制在下一个指令边界进入激活后台模式 (地址不必是操作码)
3 CLKSW	选择 BDC 通信时钟的源 — CLKSW 默认 0, 选择其它 BDC 时钟源。 0 其它 BDC 时钟源 1 MCU 总线时钟表
2 WS	等待或停止状态 — 当目标 CPU 处于等待或停止状态时, 大多数 BDC 命令不起作用。但是可以用后台命令来强制目标 CPU 从等待或停止状态进入激活后台模式, 这样所有 BDC 命令都可以起作用。只要主机强制目标 MCU 进入激活背景调试模式, 主机应该发出 READ_STATUS 命令, 在尝试其它 BDC 命令前, 检查 BDMACT = 1。 0 目标 CPU 运行用户应用程序代码, 或处于激活背景调试模式 (当后台激活时, 它不处于等待或停止模式) 1 目标 CPU 处于等待或停止模式, 或者后台命令用来将其从等待或停止状态改变为激活背景调试模式
1 WSF	等待或停止失败状态 — 如果这存储器存取命令因目标 CPU 在大约相同时间执行等待或停止指令而失败, 则设置这个状态位。通常的恢复策略是发出后台命令, 从等待或停止模式进入激活后台模式, 重复失败的命令, 然后返回到用户程序。(一般地, 主机应该恢复 CPU 寄存器, 准备值, 重新执行等待或停止指令。) 0 存储器存取与等待或停止指令不冲突 1 存储器存取命令失败, 因为 CPU 已进入等待或停止模式
0 DVF	数据有效失败状态 — 这个状态位没有在 MC9S08DZ60 系列中使用, 因为它没有慢存取存储器。 0 存储器存取与慢存储器接入不冲突 1 存储器存取命令失败, 因为 CPU 没有完成慢存储器接入

12.4.1.2 BDC 断点匹配寄存器 (BDCBKPT)

6-位寄存器保留 BDC 中的硬件断点的地址。BDCSCR 中的 BKPTEN 和 FTS 控制位用来使能和配置断点逻辑。专门的串行 BDC 命令 (READ_BKPT 和 WRITE_BKPT) 用来读和写 BDCBKPT 寄存器, 但是用户程序不能存取它, 因为它不位于 MCU 的普通存储器映射空间中。当目标 MCU 处于激活背景调试模式时, 断点一般在运行用户应用程序前设置。关于建立和使用 BDC 中的硬件断点逻辑的更多信息, 请参见 [12.2.4 节 BDC 硬件断点](#)。

12.4.2 系统背景调试强制复位寄存器 (SBDFR)

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



¹ BDFR 只有通过串行后台模式调试命令才可写，不能通过用户程序来写。

图 12-6. 系统背景调试强制复位寄存器 (SBDFR)

表 12-3. 寄存器字段描述

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

12.4.3 DBG 寄存器和控制位

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

12.4.3.1 调试比较器 A 高寄存器 (DBGCAH)

这个寄存器包含比较器 A 的高 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

12.4.3.2 调试比较器 A 低寄存器 (DBGCAL)

这个寄存器包含比较器 A 的低 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

12.4.3.3 调试比较器 B 高寄存器 (DBGCBH)

这个寄存器包含比较器 B 的高 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

12.4.3.4 调试比较器 B 低寄存器 (DBGCBL)

这个寄存器包含比较器 B 的低 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

12.4.3.5 调试 FIFO 高寄存器 (DBGFH)

这个寄存器提供对 FIFO 的高 8 位的只读接入。写到这个寄存器没有意义或无效果。在纯事件触发模式中，FIFO 只将数据存储在每个 FIFO 字的低字节，因此这个寄存器不能使用，将读 0x00。

读 DBGFH 不会导致 FIFO 移动到下一个字。当从 FIFO 中读出 16- 位字时，在读 DBGFL 前先读 DBGFH，因为读 DBGFL 会导致 FIFO 先于下个字的信息。

12.4.3.6 调试 FIFO 低寄存器 (DBGFL)

这个寄存器提供对 FIFO 的低 8 位的只读存取。写到这个寄存器没有意义或无效果。

读 DBGFL 会导致 FIFO 移动到下一个字的信息。当调试模块以纯事件模式运行时，只有 8- 位数据存储于 FIFO（每个 FIFO 字的高字节部分没有使用）。当从 FIFO 中读出 8- 位字时，只需重复地读 DBGFL，从 FIFO 中获得数据的连续的字节。在这种情况下，没有必要读 DBGFH。

当 FIFO 仍然打开时（打开后，但 FIFO 充满或 ARMF 被清除前）不要试图从其中读数据，因为在 DBGFL 读取过程中，FIFO 不能进一步操作。这可以干扰正常的 FIFO 的读取顺序。

在调试器没有打开的情况下读会使最近获取的操作码的地址存储到 FIFO 中的最后的位置。读取 DBGFL，然后定期 DBGFL，外部主机软件可以开发程序执行的概况。在对 FIFO 进行八次读取后，第九次读取将返回第一次读取结果的信息。要使用分析功能，则需要读取 FIFO 八次，且不使用启动顺序的数据，然后开始使用数据来获取已执行地址的延迟概况。存储在 FIFO 中的关于 DBGFL（且 FIFO 没有打开）读取的信息就是最近所获操作码的地址。

12.4.3.7 调试控制寄存器 (DBGC)

这个寄存器可以在任何时间读或写。

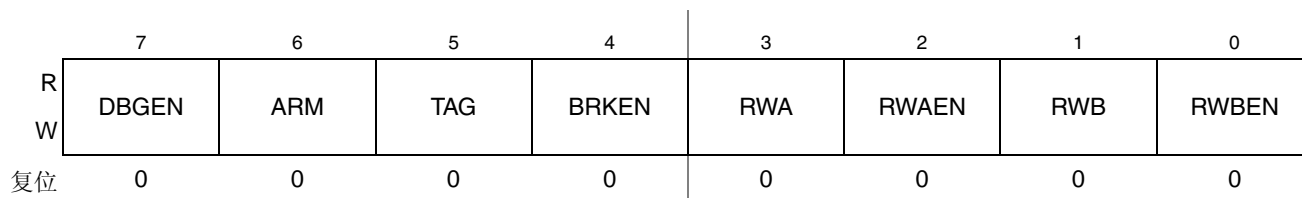


图 12-7. 调试控制寄存器 (DBGC)

表 12-4. DBGC 寄存器字段描述

字段	描述
7 DBGEN	调试模块启用 — 用来启用调试模块。DBGEN 不能设置为 1, 如果 MCU 是安全的。 0 DBG 禁用 1 DBG 启用
6 ARM	打开控制 — 控制调试器是否在 FIFO 中比较和存储信息。采用写操作来设置该位 (和 ARMF), 完成调试运行就是自动清除它。将 ARM 或 DBGEN 写为 0, 可以停止任何调试运行。 0 调试器没有打开 1 调试器被打开
5 TAG	标记 / 强制选择 — 控制送到 CPU 的中断请求是否为标签或强制型请求。如果 BRKEN = 0, 这个位就没有意义或无效。 0 CPU 中断请求作为强制型请求 1 CPU 中断请求作为标签型请求
4 BRKEN	中断启用 — 控制触发事件是否向 CPU 生成中断请求。触发事件可以使信息存储在 FIFO 中而不必向 CP 生成中断请求。对于结束跟踪, 如果比较器 (s) 和 R/W 满足触发条件, 则发出 CPU 中断请求。对于起始跟踪, 则当 FIFO 满时发出 CPU 中断请求。TRGSEL 不影响 CPU 中断请求的定时。 0 CPU 不断请求未启用 1 触发器触发向 CPU 发出中断请求
3 RWA	比较器 A 的 R/W 比较值 — 当 RWAEN = 1, 这个位确定是否用读或写接入来鉴定比较器 A, 当 RWAEN = 0, RWA 和 R/W 信号不影响比较器 A。 0 比较器 A 只在写周期上匹配 1 比较器 A 只在读周期上匹配
2 RWAEN	启用比较器 A 的 R/W — 控制比较器 A 的匹配是否考虑这个水平的 R/W。 0 R/W 未用在比较 A 中 1 R/W 用在比较 A 中
1 RWB	比较器 B 的 R/W 比较值 — 当 RWBEN = 1, 这个位确定是否用读或写接入来鉴定比较器 B。当 RWBEN = 0, RWA 和 R/W 信号不影响比较器 B。 0 比较器 B 只在写周期上匹配 1 比较器 B 只在读周期上匹配
0 RWBEN	启用比较器 B 的 R/W B — 控制比较器 B 的匹配是否考虑这个水平的 R/W。 0 R/W 未用在比较 B 中 1 R/W 用在比较 B 中

12.4.3.8 调试触发寄存器 (DBGT)

这个寄存器在任何时候都可以读，但是只有当 $ARM = 0$ 时才可以写，除非位 4 和位 5 硬件线与至 0。

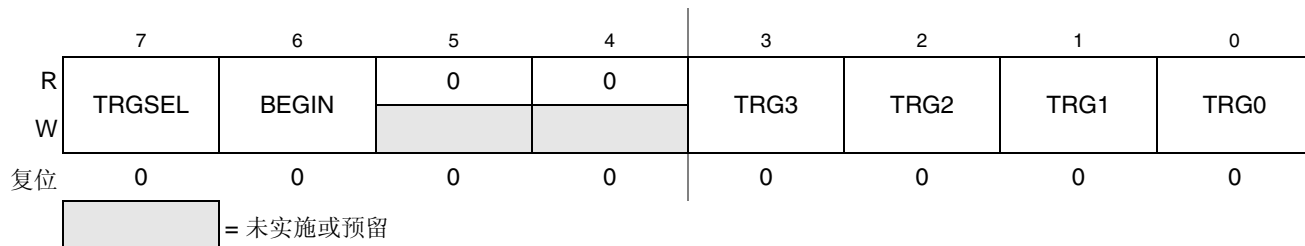


图 12-8. 调试触发寄存器 (DBGT)

表 12-5. DBGT 寄存器字段描述

字段	描述
7 TRGSEL	触发类型 — 控制比较器 A 和 B 的匹配输入是否与调试模块中的操作码跟踪逻辑匹配。如果 TRGSEL 已设置，比较器 A 或 B 的匹配信号必须通过操作码跟踪逻辑传播，如果匹配地址的操作码实际已执行，则只有触发事件发送到 FIFO 逻辑。 0 存取比较地址时触发 (强制) 1 如果比较地址的操作码已执行 (标签)，则触发
6 BEGIN	开始 / 结束触发选择 — 控制 FIFO 在触发时开始填充还是以循环形式填充直到触发结束信息的捕获。在纯事件触发模式中，忽略这个位，所有调试运用都假定为起始跟踪。 0 数据存储存储在 FIFO，直到触发 (结束跟踪) 1 触发启动数据存储 (起始跟踪)
3:0 TRG[3:0]	选择触发模式 — 选择下面 9 个触发模式中的一个。 0000 只有 A 0001 A 或 B 0010 A 然后 B 0011 只有事件 B (存储数据) 0100 A 然后只有事件 B (存储数据) 0101 A 和 B 数据 (满模式) 0110 A 和非 B 数据 (满模式) 0111 1 范围内: $A \leq \text{地址} \leq B$ 1000 范围外: 地址 $< A$ 或 地址 $> B$ 1001 – 1111 (无触发)

12.4.3.9 调试状态寄存器 (DBGS)

这是一个只读状态寄存器。

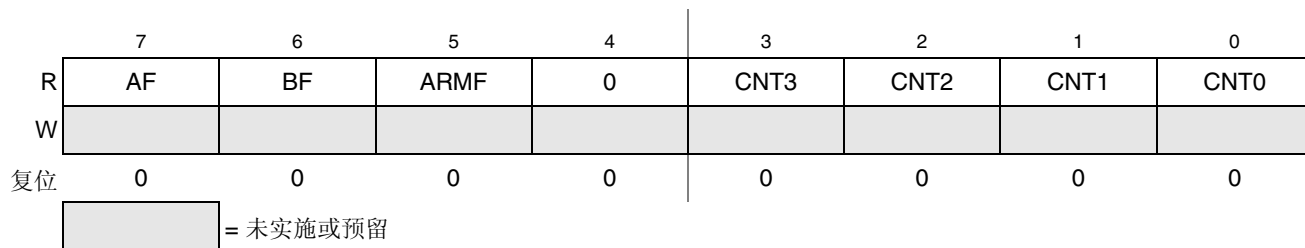


图 12-9. 调试状态寄存器 (DBGS)

表 12-6. DBGS 寄存器字段描述

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

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。

注意

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 所有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	T_L 到 T_H -40 to 85	°C
热敏电阻 ^{1,2,3,4} 44 引脚 LQFP 1s 2s2p 32 引脚 LQFP 1s 2s2p	θ_{JA}	73 56 85 56	°C/W

- 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 °C 为单位）从等式 A-1 中得到：

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

其中：

T_J = 周围环境温度, °C

T_A = 封装的热敏电阻, 连接到周围环境, °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 和等式 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 A-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 和等式 A-2 可以得到 P_D 和 T_J 。

A.5 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-4. 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) 5 V, $I_{load} = -2$ mA	V_{OH}	$V_{DD}-1.5$	—	—	V
		输出低电压——高驱动 (PTxDSn=1) 5 V, $I_{load} = -10$ mA		$V_{DD}-1.5$	—	—	
2	P	输出低电压——低驱动 (PTxDSn=0) 5 V, $I_{load} = -2$ mA	V_{OL}	1.5	—	—	V
		输出高电压——高驱动 (PTxDSn=1) 5 V, $I_{load} = -10$ mA		1.5	—	—	
3	P	输出高电流——所有端口最大总 I_{OH} 5 V	I_{OHT}	—	—	100	mA
4	P	输出高电流——所有端口最大总 I_{OH} 5 V	I_{OLT}	—	—	100	mA
5	P	输入高电压; 所有的数字输入 $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	V_{IH}	$0.65 \times V_{DD}$	—	—	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	—	μA
9	P	高阻抗 (掉电状态) 泄露电流 ²	$ V_{OZ} $	—	0.1	—	μA
10	P	内部上拉电阻 ³	R_{PU}	—	45	—	k Ω
11	P	内部下拉电阻 ⁴	R_{PD}	—	45	—	k Ω
12	C	输入电容; 所有的非电源引脚	C_{In}	—	—	8	pF

表 A-4. DC 特性 (continued)

编号	C	参数	标志	最小	典型 ¹	最大	单位
13	P	POR 电压	V_{POR}	—	1.4	—	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}	—	4.3 4.4	—	V
16	P	低压警告阈——大范围 V_{DD} 下降 V_{DD} 上升	V_{LVWH}	—	4.3 4.4	—	V
17	P	低压禁止复位 / 还原滞后 5 V	V_{hys}	—	100	—	mV

¹ 除非说明, 典型值为 25 °C 描述的数据。

² 用 $V_{In} = V_{DD}$ 或 V_{SS} 测量。

³ 用 $V_{In} = V_{SS}$ 测量。

⁴ 用 $V_{In} = V_{DD}$ 测量。

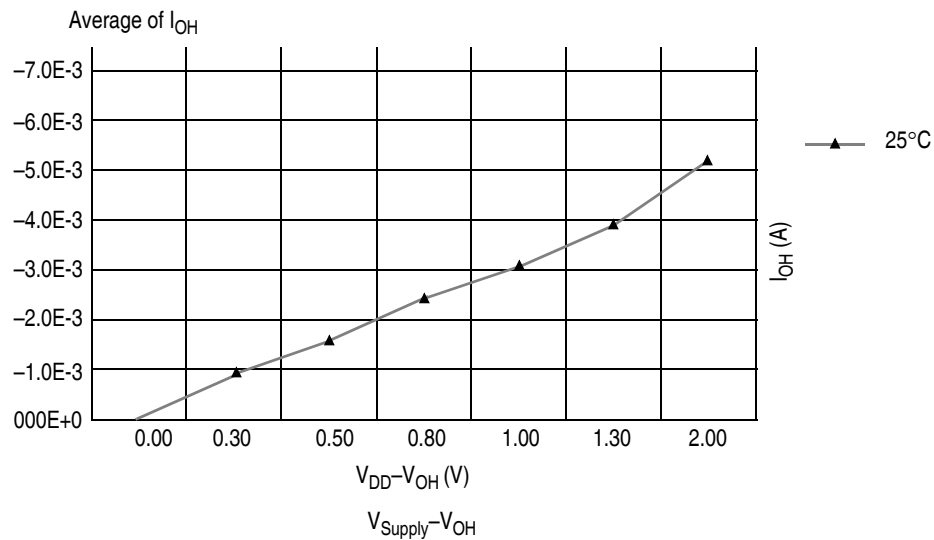
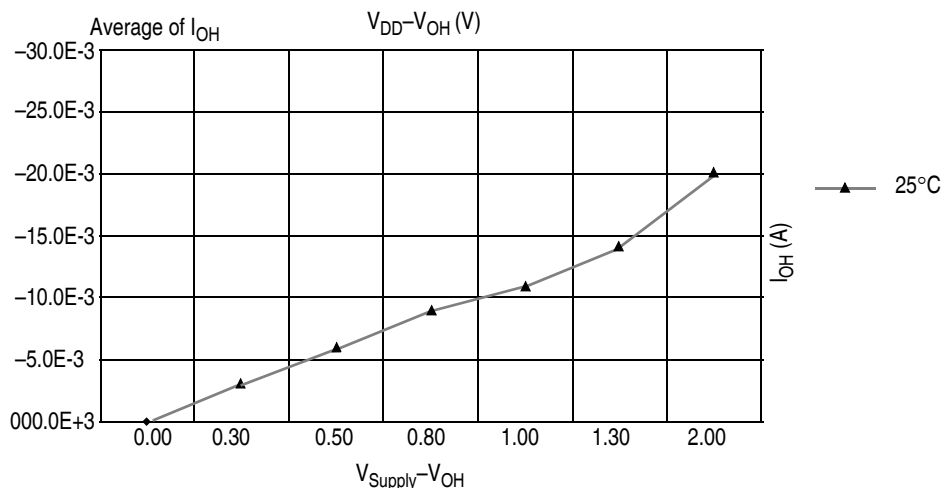


图 A-1. $V_{DD} = 5\text{ V}$ 时, I_{OH} (低驱动) 和 $V_{DD} - V_{OH}$ 的典型值

图 A-2. $V_{DD} = 5\text{ V}$ 时, I_{OH} (高驱动) 和 $V_{DD}-V_{OH}$ 的典型值

A.6 电源电流特性

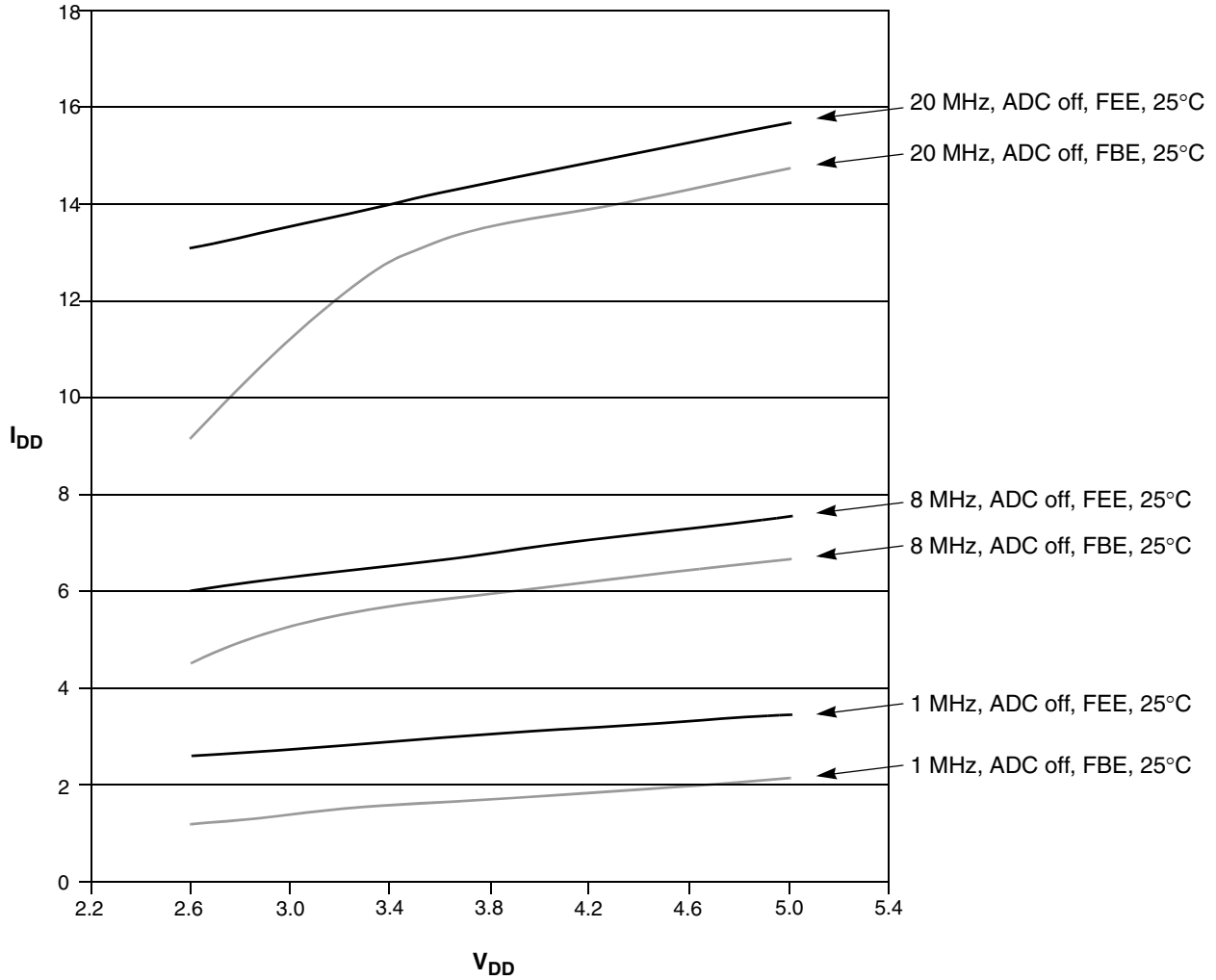
表 A-5. 电源电流特性

编号	C	参数	标志	V_{DD} (V)	典型 ¹	单位	温度
1	C	运行时电源电流 ² (CPU 时钟 =2 MHz, $f_{Bus}=1\text{ MHz}$)	$R I_{DD}$	5	0.988	mA	-40 到 85 °C
2	C	运行时电源电流 ³ (CPU 时钟 =2MHz, $f_{Bus}=1\text{ MHz}$)	$R I_{DD}$	5	6.84	mA	-40 到 85 °C
3	C	Stop2 模式电源电流	$S2 I_{DD}$	5	1.36	μA	-40 到 85 °C
4	C	Stop3 模式电源电流	$S3 I_{DD}$	5	1.49	μA	-40 到 85 °C
5	C	带有 RTI 的 Stop2 或 stop3	$S3 I_{DDRTI}$	5	300	nA	-40 到 85 °C
6	C	带有 LVD 的 stop3 ($LVDE=LV DSE=1$)	$S3 I_{DDLVD}$	5	116	μA	-40 到 85 °C
7	C	带有振荡器的 stop3 ($OSCSETN=1$)	$S3 I_{DDOSC}$	5	5	μA	-40 到 85 °C

¹ 基于 25 °C 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-3, 图 A-4, 图 A-5。

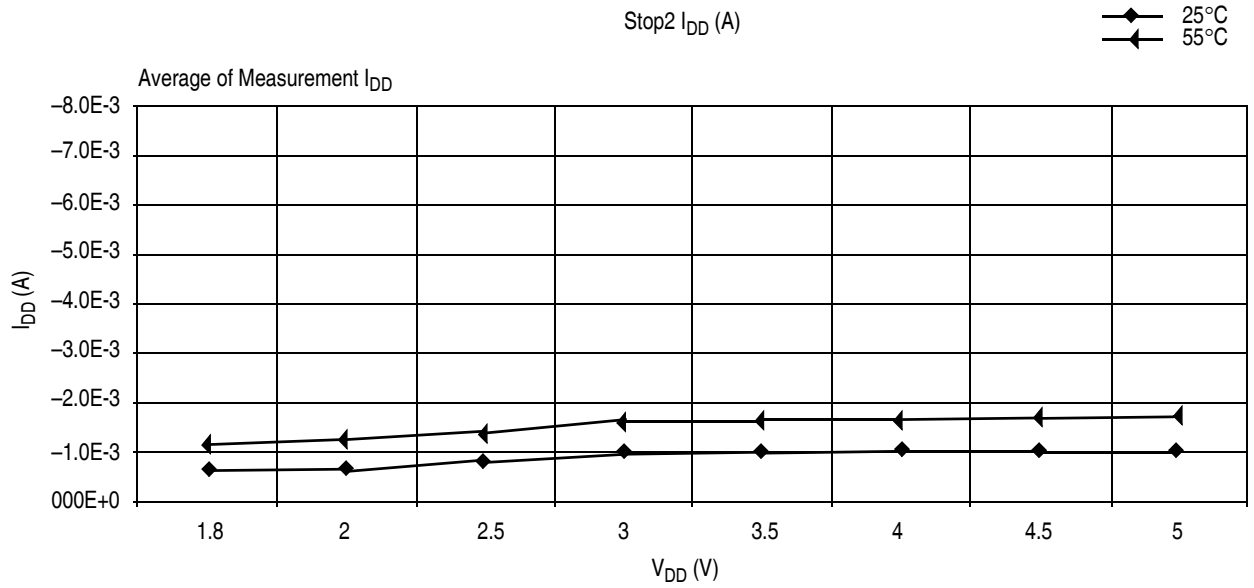
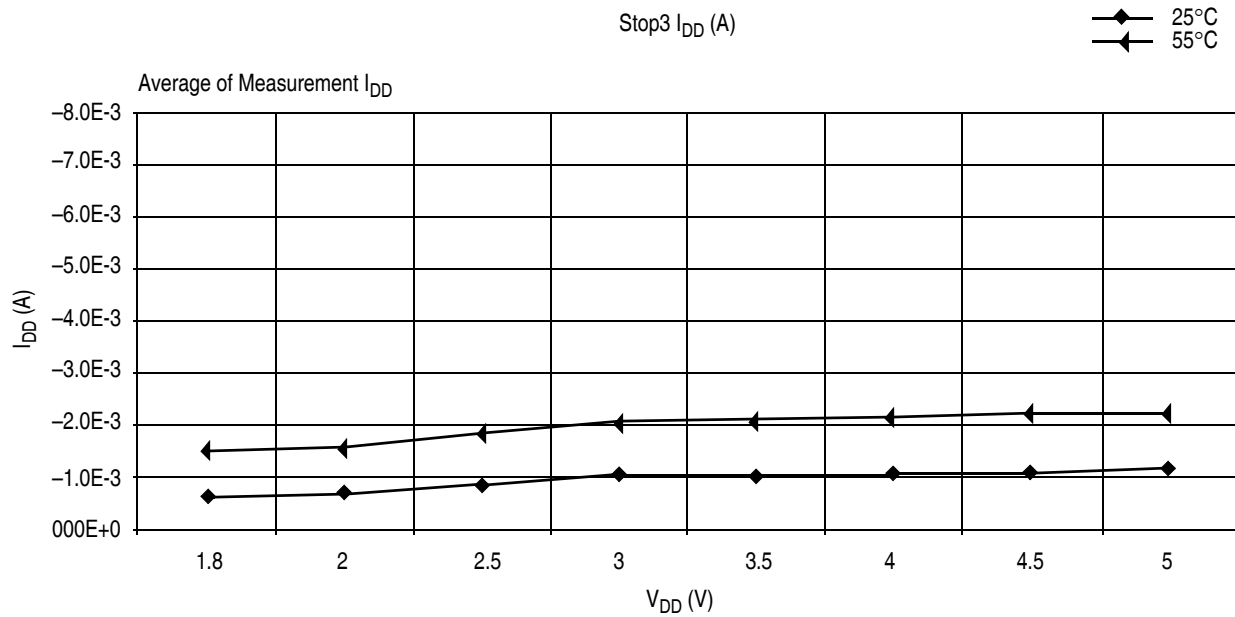
² 除了 ADC, 配置为 FBE 的 ICG, 所有的模块在引脚上不包含任何直流电输入。

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。



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-3. FBE 和 FEE 模式中的典型运行 I_{DD} , I_{DD} 和 V_{DD}

图 A-4. 典型的 stop2 I_{DD} 图 A-5. 典型的 stop3 I_{DD}

A.7 ADC 特性

表 A-6. 5 V 8 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	—	pF
输入电阻	—	R_{ADIN}	—	3	—	k Ω
外接 MCU 的模拟源电阻	8 位模式 (所有有效 f_{ADCK})	R_{AS}	—	—	10	k Ω
ADC 转换时钟频率	高速 (ADLPC=0)	f_{ADCK}	0.4	—	8.0	MHz
	低速 (ADLPC=1)		0.4	—	4.0	

¹ 除非另有说明, 假定 $V_{DDAD} = 5 V$, 温度 = 25 °C, $f_{ADCK} = 1.0 MHz$, 典型值指示举例, 没有在产品中测试。

² DC 位差不同。

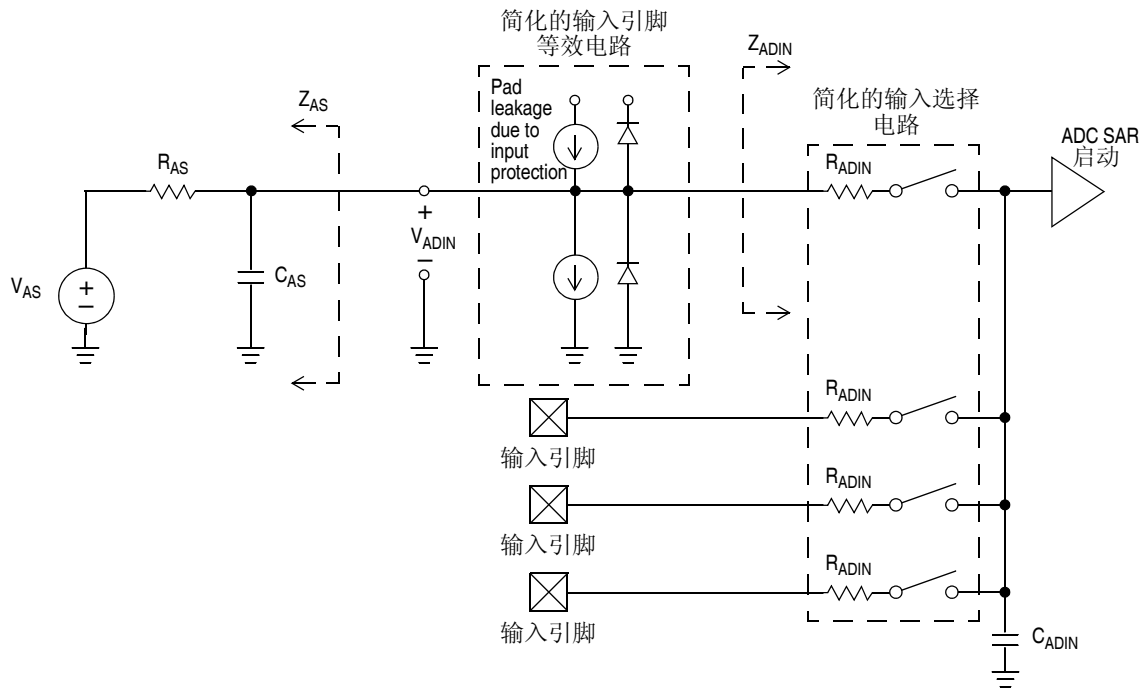


图 A-6. ADC 输入电阻框图

表 A-7. 5 V 10 位 ADC 转换特性 ($V_{REFL}=V_{DDAD}$, $V_{REFL}=V_{SSAD}$)

特性	条件	C	标志	典型值 ¹	单位
电源电流 ADLPC = 1 ADLSMP = 1 ADCO = 1	—	T	I_{DDAD}	133	μA
电源电流 ADLPC = 1 ADLSMP = 0 ADCO = 1	—	T	I_{DDAD}	218	μA
电源电流 ADLPC = 1 ADLSMP = 0 ADCO = 1	—	T	I_{DDAD}	327	μA
电源电流 ADLPC = 0 ADLSMP = 0 ADCO = 1	—	T	I_{DDAD}	582	μA
ADC 异步时钟源 $t_{ADACK} = 1/f_{ADACK}$	高速 (ADLPC = 0)	P	f_{ADACK}	3.3	MHz
	低功耗 (ADLPC = 1)			2	
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	8 位模式	P	E_{TUE}	± 0.5	LSB^2
微分非线性	8 位模式	P	DNL	± 0.3	LSB^2
	确保单调且无遗失编码				
积分非线性	8 位模式	C	INL	± 0.3	LSB^2
归零误差 $V_{ADIN} = V_{SSA}$	8 位模式	P	E_{ZS}	± 0.5	LSB^2
满标误差 $V_{ADIN} = V_{DDA}$	8 位模式	P	E_{FS}	± 0.5	LSB^2
量化误差	8 位模式	D	E_Q	—	LSB^2
输入泄露误差 焊盘泄露 ³ *RAS	8 位模式	D	E_{IL}	± 0.1	LSB^2
温度传感器斜率	-40 °C – 25 °C	D	m	3.266	mV/°C
	25 °C – 85 °C			3.636	
温度传感器电压	25 °C	D	V_{TEMP25}	1.41	mV

¹ 如果没有其他说明, 假定 $V_{DDAD} = 5 \text{ V}$, 温度 = 25 °C, $f_{ADCK} = 1 \text{ MHz}$ 典型值, 典型值仅供参考未在产品中测试。

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

³ 基于输入焊盘泄漏电流, 参考焊盘电气。

A.8 内部时钟产生模块特性

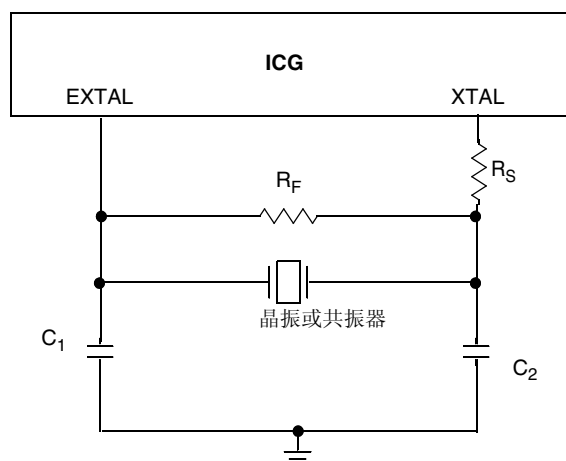


表 A-8. ICG DC 电气描述 (温度范围 = -40 到 85 °C)

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 低范围 (32 k 到 100 kHz) 高范围 (1 M-16 MHz)	R _F	—	10 1	—	MΩ
串联电阻 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) ≥ 8 MHz 4 MHz 1 MHz	R _S	—	0 100 0 10 20	—	kΩ

¹ 典型值是在 V_{DD} = 5 V, 25 °C 条件下取得。

² 参见晶振或共鸣器厂商的建议。

A.8.1 ICG 频率规格

表 A-9. ICG 频率规格
 $V_{DDA}=V_{DDA}(\min)$ 到 $V_{DDA}(\max)$, 温度范围 = -40 to 85 °C

编号	C	特性	标志	最小	典型 ¹	最大	单位
1	C	振荡器晶体或谐振器 (REFS=1)(基谐波晶体 或陶瓷电容) 低范围 高范围 高增益 高增益 低功耗 低功耗	f_{lo}	32	—	100	kHz
			f_{hi_byp}	1	—	16	MHz
			f_{hi_eng}	2	—	10	MHz
			f_{lp_byp}	1	—	8	MHz
			f_{lp_eng}	2	—	8	MHz
2	C	输入时钟频率 (CLKS=11,REFS=0) 低范围 高范围	f_{lo}	32	—	100	kHz
			f_{hi_eng}	2	—	10	MHz
3	C	输入时钟频率 (CLKS=10,REFS=0)	f_{Extal}	0	—	40	MHz
4	C	内部参考频率 (未对齐)	$f_{ICGIRCLK}$	182.25	243	303.75	kHz
5	C	输入时钟的占空因子 (REFS=0)	t_{dc}	40	—	60	%
6	C	输出时钟 ICGOUT 频率 CLKS=10,REFS=0 其他情况	f_{ICGOUT}	$f_{Extal}(\min)$ $f_{lo}(\min)$	—	$f_{Extal}(\max)$ $f_{ICGDCLKmax}(\max)$	MHz
7	C	最小 DCO 时钟 (ICGDCLK) 频率	$f_{ICGDCLKmin}$	8	—	—	MHz
8	C	最小 DCO 时钟 (ICGDCLK) 频率	$f_{ICGDCLKmax}$	—	—	40	MHz
9	C	自时钟模式 (ICGOUT) 频率 ²	f_{Self}	$f_{ICGDCLKmin}$	—	$f_{ICGDCLKmax}$	MHz
10	C	自时钟模式复位 (ICGOUT) 频率	f_{Self_reset}	5.5	8	10.5	MHz
11	C	丢失参考频率 ³ 低范围 高范围	f_{LOR}	5	—	25	kHz
				50	—	500	
12	C	丢失 DCO 频率 ⁴	f_{LOD}	0.5	—	1.5	MHz
13	C	晶振开始时序 ^{5,6} 低范围 高范围	t_{CSTL}	—	430	—	ms
			t_{CSTH}	—	4	—	

表 A-9. ICG 频率规格
 $V_{DDA}=V_{DDA(\min)}$ 到 $V_{DDA(\max)}$, 温度范围 = -40 to 85 °C)

编号	C	特性	标志	最小	典型 ¹	最大	单位
14	C	FLL 锁时序 ⁷ 低范围 高范围	t_{Lockl} t_{Lockh}	—	—	2 2	ms
15	C	FLL 频率解锁范围	n_{Unlock}	$-4*N$	—	$4*N$	counts
16	C	FLL 频率上锁范围	n_{Lock}	$-2*N$	—	$2*N$	counts
17	C	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	C	内部振荡器背离对齐的频率 ⁹ $V_{DD} = 2.7 - 5.5$ V, (恒温) $V_{DD} = 5.0$ V \pm 10%, -40 °C to 85 °C	ACC_{int}	—	± 0.5 ± 0.5	—	%

¹ 除非另有说明, 典型值是在 $V_{DD}=5$ V, 25 °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)。

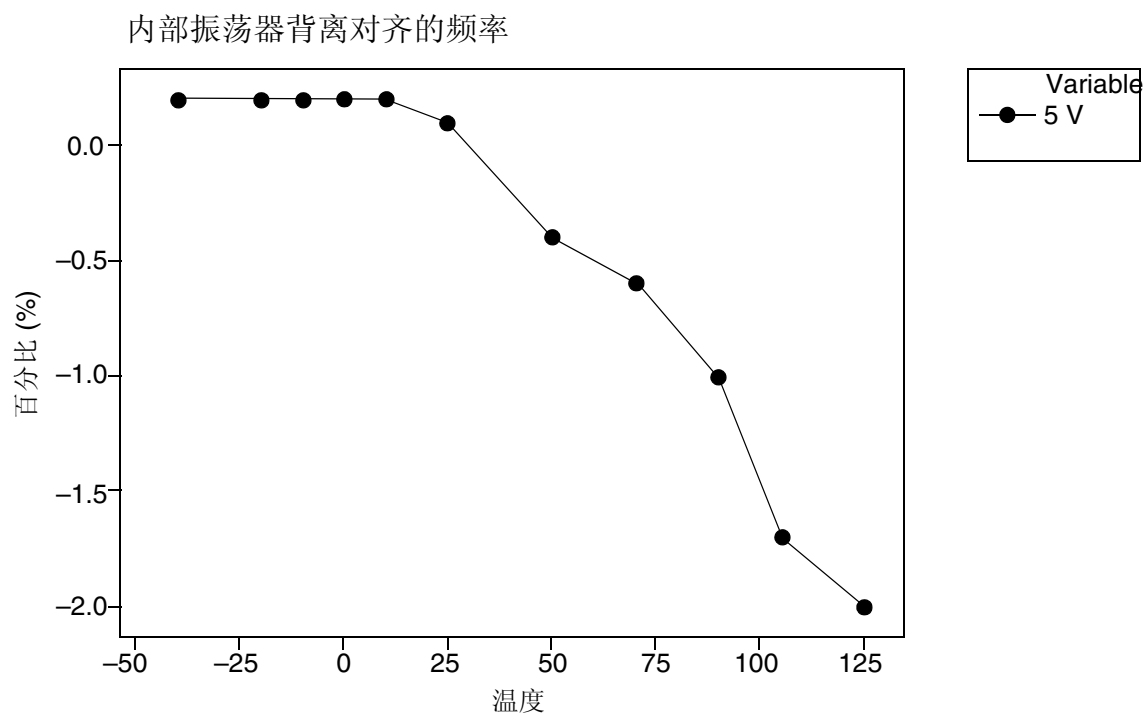
⁵ 该参数未 100% 的测试就被描述。

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 $V_{DDA}V_{SSA}$ 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-7。



设备在 25 °C，3 V 对齐。

图 A-7. 内部振荡器背离对齐的频率

A.9 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 8 章 内部时钟发生器 (S08ICGV4)。

A.9.1 控制时序

表 A-10. 电源电流特性

编号	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 ($t_{cyc} = 1/f_{Self_reset}$)	t_{extrst}	$1.5 \times t_{Self_reset}$	—	—	ns
4	复位低驱动 ³	t_{rstdrv}	$34 \times t_{cyc}$	—	—	ns
5	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

表 A-10. 电源电流特性

编号	参数	标志	最小	典型 ¹	最大	单位
7	IRQ 脉宽 异步路径 ² 同步路径 ⁴	t_{LIH} , t_{HIL}	100 $1.5 \times t_{cyc}$	—	—	ns
8	KBIPx 脉宽 异步路径 ² 同步路径 ³	t_{LIH} , t_{HIL}	100 $1.5 \times t_{cyc}$	—	—	ns
9	端口上升和下降时间 (load = 50 pF) ⁵ 禁止输出斜率控制 (PTxSE=0) 允许输出斜率控制 (PtxSE=1)	t_{Rise} , t_{Fall}	—	3 30	—	ns

¹ 除非另有说明，典型值是在 $V_{DD} = 5\text{ V}$ ， $25\text{ }^\circ\text{C}$ 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_{DD} 20% 和 70%；温度范围 $-40\text{ }^\circ\text{C}$ to $85\text{ }^\circ\text{C}$ 。

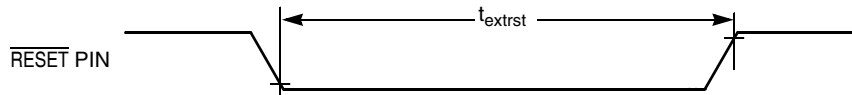


图 A-8. 复位时序

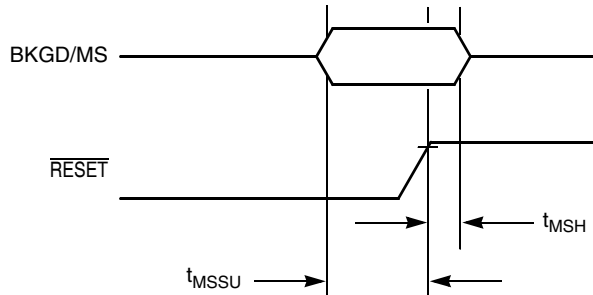


图 A-9. 有效的背景调试模式的闭锁时序

A.9.2 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-11. ICG DC 电气描述 (温度范围 -40 到 125 °C)

特性	标志	最小	最大	单位
外部时钟频率	f_{TPMext}	dc	$f_{\text{Bus}}/4$	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}

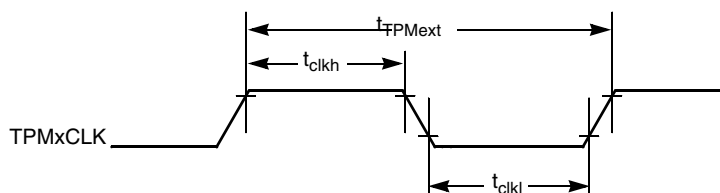


图 A-10. 定时器外部时钟

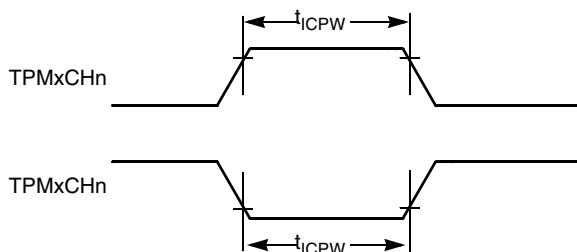


图 A-11. 定时器输入捕获脉冲

A.10 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-12. Flash 特性

编号	C	特性	标志	最小	典型 ¹	最大	单位
1	D	写入 / 擦除电压	$V_{\text{prog/erase}}$	2.7	—	5.5	V
2	D	读操作电压	V_{read}	2.7	—	5.5	V
3	D	内部 FCLK 频率 ²	f_{FCLK}	150	—	200	KHz

表 A-12. Flash 特性 (continued)

编号	C	特性	标志	最小	典型 ¹	最大	单位
4	D	内部 FCLK 周期 (1/FCLK)	$t_{F_{cyc}}$	5	—	6.67	μs
5	D	自己写入时间 (随机地址) ²	t_{prog}	9			$t_{F_{cyc}}$
6	D	自己写入时间 (突发模式) ²	t_{Bust}	4			$t_{F_{cyc}}$
7	D	页擦除时间 ³	t_{Page}	4000			$t_{F_{cyc}}$
8	D	整体擦除时间 ²	t_{Mass}	20000			$t_{F_{cyc}}$
9	D	写入 / 擦除强度 ⁴ T_L 到 $T_H = -40\text{ }^\circ\text{C}$ 到 $+125\text{ }^\circ\text{C}$ $T = 25$		10000 —	— 10000	—	周期
10	D	数据保持 ⁵	t_{D_ret}	15	100	—	年

¹ 除非有其他说明, 典型值 $V_{DD} = 5V$, $25\text{ }^\circ\text{C}$ 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.11 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 订购信息和机械图

B.1 订购信息

该节包含了 SC667027 系列微控制器的订购信息。

表 B-1. 微处理器编号系统

设备编号 ¹	存储器		可用封装 ²
	Flash	RAM	类型
SC667027	16,384	1024	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。

按照该设备家族，SC667027 系列的订购零件编号也是多样的。参见图 B-1。

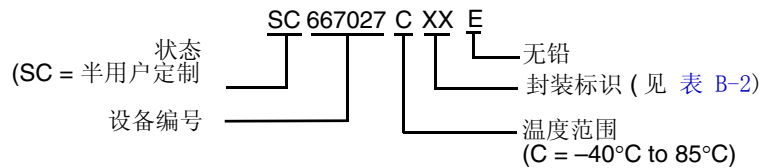


图 B-1. SC667027 系列的编号系统

B.2 机械图

接下来的内容包含了 SC667027 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. 封装信息

引脚数	类型	标识	文档号
44	LQFP	FG	98ASS23225W
32	LQFP	FJ	98ASH70029A

如何联系我们:

主页:
www.freescale.com

技术支持网页:
<http://www.freescale.com/support>

美国 / 欧洲或未列出的地方:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

欧洲、中东和非洲:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

日本:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

亚太地区:
飞思卡尔半导体(中国)有限公司 100022
北京市朝阳区建国路乙 118 号
京汇大厦 23 层
+86 10 5879 8000
support.asia@freescale.com

仅提供印刷品请求:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

文档号: SC667027ZHS
第 1 版
2008 年 9 月

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org
© Freescale Semiconductor, Inc. 2008. All rights reserved.