

MC9RS08KA8
MC9RS08KA4
参考手册

深圳市南天星

RS08

微控制器

相关文档:

- **MC9RS08KA8 (数据表)**
包含引脚分配和图表, 所有电气规格和机械图纸大纲。

查找所有文档的最新版本:
[Http://www.freescale.com](http://www.freescale.com)

MC9RS08KA8RM
修订版 4
2015 年

Freescale.com





深圳南天星

MC9RS08KA8 功能

8 位 RS08 中央处理器单元 (CPU)

- 高达 20 MHz 的 CPU，温度范围为 -40°C 至 85°C，电压范围为 1.8V 至 5.5V
- 添加 BGND 指令的 HC08 指令集子集

片上内存

- 8 KB 闪存读取/程序/擦除在全工作电压和温度下
 - KA4 有 4 KB 闪存
- 254 字节随机存取存储器 (RAM) -KA4 有 126 字节的 RAM
- 防止未经授权访问 RAM 和闪存内容的安全电路

节电模式

- 等等，然后停下来
- 使用实时中断 (RTI)、KBI 或 ACMP 从省电模式唤醒

时钟源选项

- 振荡器 (XOSC) - 环路控制穿孔振荡器；晶体或陶瓷谐振器
 - 范围为 31.25 kHz 至 39.0625 kHz 或 1 MHz 到 5 MHz
- 内部时钟源 (ICS) - 内部时钟源模块包含由内部或外部参考控制的频率锁定回路 (FLL)；内部参考的精确修剪允许 0.2% 的分辨率和 2% 的温度和电压偏差；支持高达 10 MHz 的总线频率

系统保护

- 看门狗计算机正常运行 (COP) 重置，可以选择从专用的 1 kHz 内部时钟源或总线时钟运行
- 带有重置或中断的低压检测
- 带有重置的非法操作码检测
- 通过重置进行非法地址检测
- 闪存块保护

开发支持

- 单线后台调试接口
- 断点功能允许在电路内调试期间设置单个断点

外围设备

- ADC—12 通道，10 位分辨率；2.5M 转换时间；自动比较功能；停止操作；功能齐全，从 2.7V 到 5.5V (16 针封装上提供 8 通道)
 - TPM——一个 2 通道；每个通道上可选的输入捕获、输出比较或缓冲边缘或中心对齐 PWM
 - IIC — 集成电路总线模块，能够运行高达 100 kbps，最大总线负载；能够以较低的负载实现更高的包德率
 - MTIM1 和 MTIM2——两个 8 位模定时器
 - KBI—带有上升或下降边缘检测的键盘中断；八个 KBI 端口
 - 16 针和 20 针封装
 - ACMP — 模拟比较器；完整轨道到轨道的供应操作；可选择与固定的内部带格参考电压进行比较；可以在停止模式下运行 **输入/输出**
-

- 14/18 GPIO，包括一个仅输出引脚和一个仅输入引脚
- 所有输入引脚上的滞后和可配置的上拉装置；所有输出引脚上可配置的减压率和驱动强度

套餐选项

- 16 针 SOIC、PDIP 或 TSSOP
- 20 针 SOIC 或 PDIP

深圳南天星



深圳南天星

MC9RS08KA8 系列参考手册

封面： MC9RS08KA8 MC9RS08KA4

MC9RS08KA8
修订版 4
2015 年 5 月

深圳南天星



修订历史

为了提供最新信息，我们对万维网上文件的修订将是最新的。您的打印副本可能是更早的修订版。要验证您是否拥有最新信息，请参阅：[Http://freescale.com](http://freescale.com)

(RS08IICV2) 131 第 13 章 模块计时器

(RS08MTIMV1) 161

第 15 章 发展支持 177

深圳市南天星



深圳南天星

内容

部分编号	标题	页
第 1 章		
MC9RS08KA8 设备概述		
1.1	概述	17
1.2	MCU 方框图	17
1.3	系统时钟分布	19
第 2 章		
引脚和连接		
2.1	导言	21
2.2	设备引脚分配	21
2.3	推荐的系统连接	22
2.4	别针细节	22
2.4.1	电源引脚	22
2.4.2	PTA5/TCLK/RESET/V 代表别 针	23
2.4.3	PTA4/ACMPO/BKGD/MS Pin	23
2.4.4	通用 I/O 和外围端口	23
第 3 章		
操作模式		
3.1	导言	25
3.2	特点	25
3.3	运行模 式	25
3.4	主动后台模式	25
3.5	等待模 式	26
3.6	停止模 式	27

3.6.1	在停止模式下启用主动 BDM	28
	LVD 在停止模式下启用	28

第 4 章 记忆

4.1	记忆图	29
4.2	未实现的内存	30
4.3	索引/间接寻址	30
4.4	RAM 和寄存器地址和位分配	31
4.5	RAM	34
4.6	闪光	35
4.6.1	特点	35
4.6.2	闪存编程程序	35
4.6.3	闪光大规模擦除操作	36
4.6.4	安全	36
4.7	闪存寄存器和控制位	37
	4.7.1 闪存选项寄存器 (FOPT 和 NVOPT)	37
	4.7.2 闪存控制寄存器 (FLCR)	38
4.8	页面选择寄存器 (PAGESEL)	38

第 5 章

重置、中断和通用系统控制

5.1	导言	
5.2	特点	41
5.3	MCU 重置	41
5.4	计算机正常运行 (COP) 看门狗	42
5.5	中断	42
5.6	低压检测 (LVD) 系统	43

5.6.1	开机重置操作	43	5.6.2	LVD 重置操作	43
5.6.3	LVD 中断操作	43			
5.7	实时中断 (RTI)	43			
5.8	重置、中断和系统控制寄存器和控制位	44			
	5.8.1 系统重置状态寄存器 (SRS)				
	5.8.2 系统选项寄存器 (SOPT)	45			
	5.8.3 系统设备识别寄存器 (SDIDH, SDIDL)	46			
	5.8.4 系统实时中断状态和控制寄存器 (SRTISC)	47			
5.8.5	系统电源管理状态和控制 1 寄存器 (SPMSC1)	49	5.8.6	系统中断待注册 (SIP1)	49

第 6 章

并行输入/输出控制

6.1	低功耗模式下的引脚行为	52			
6.2	并行 I/O 寄存器	52			
	6.2.1 端口 A 寄存器	52	6.2.2	端口 B 寄存器	53
	6.2.3 端口 C 寄存器	54			
6.3	引脚控制寄存器	55			
	6.3.1 端口 A 引脚控制寄存器	55			
	6.3.2 端口 B 引脚控制寄存器	57			
	6.3.3 端口 C 引脚控制寄存器	59			

第 7 章

键盘中断 (RS08KBIV1)

7.1	导言	63			
	7.1.1 特点	63			
	7.1.2 操作模式	63			
	7.1.3 框图	63			
7.2	外部信号描述	64			
7.3	注册定义	64			
	7.3.1 KBI 状态和控制登记册 (KBISC)	65			
	7.3.2 KBI 引脚启用寄存器 (KBIPE)	65			
	7.3.3 KBI 边缘选择寄存器 (KBIES)	66			

7.4	功能描述	66
7.4.1	仅边缘灵敏度	66
7.4.2	边缘和水平灵敏度	67
7.4.3	KBI上拉/下拉电阻	67
7.4.4	KBI初始化	67

第 8 章

中央处理器单元 (RS08CPUV1)

8.1	引言	69
8.2	程序员模型和 CPU 寄存器	69
8.2.1	累加器 (A)	70
8.2.2	程序计数器 (PC)	71
8.2.3	影子程序计数器 (SPC)	71
8.2.4	条件代码寄存器 (CCR)	71
8.2.5	索引数据寄存器 (D[X])	72
8.2.6	索引寄存器 (X)	72
8.2.7	页面选择注册 (PAGESEL)	73
8.3	寻址模式	73
8.3.1	固有寻址模式 (INH)	73
8.3.2	相对寻址模式 (REL)	73
8.3.3	即时寻址模式 (IMM)	74
8.3.4	微小寻址模式 (TNY)	74
8.3.5	短寻址模式 (SRT)	75
8.3.6	直接寻址模式 (DIR)	75
8.3.7	扩展寻址模式 (EXT)	75
8.3.8	索引寻址模式 (IX, 由伪指令实现)	75
8.4	特别行动	75
8.4.1	重置序列	76
8.4.2	中断	76
8.4.3	等待和停止模式	76
8.4.4	主动后台模式	76
8.5	汇总说明表	77

第 9 章

模拟比较器 (RS08ACMPV1)

MC9RS08KA8 Series Reference Manual, Rev. 4

10.4	功	能	描
	述		
10.4.1	时	钟	选
	制	择	和
		分	割
			控
			106
10.4.2	输	入	选
	制	择	和
			引
			脚
			控
			107
10.4.3	硬	件	触
	器		发
			1
			07
10.4.4	转	换	控
	制		1
			07
10.4.5	自	动	比
	能		较
			功
			110
10.4.6	MCU	等	待
		模	式
		操	作
			110
10.4.7	MCU	停	止
		模	式
		操	作
			111
10.5	初	始	化
	息		信
			111
10.5.1	ADC	模	块
		初	始
		化	示
		例	112
10.6	应	用	信
	息		113
10.6.1	外	部	引
			脚
			和
			路
			114
			10.6.2
			来
			115

第 11 章

内部时钟源 (RS08ICSOSCV1)

11.1	导	言	119
11.1.1	特	点	121
11.1.2	操	作	模
		式	121
11.1.3	方	框	图
			122
11.2	外	部	信
		号	描
		述	123
11.3	注	册	定
		义	123

11.3.1 ICS 控制寄存器	1
(ICSC1)	123
11.3.2 ICS 控制寄存器	2
(ICSC2)	125
11.3.3 ICS 修剪寄存器	
(ICSTRM)	126
11.3.4 ICS 状态和控制	
(ICSSC)	126
11.4 功能描述	127
11.4.1 操作模式	127
11.4.2 模式切换	129
11.4.3 总线分频器	129
11.4.4 低功耗位使用	129
11.4.5 内部参考时钟	130
11.4.6 可选的外部参考时钟	130
11.4.7 固定频率时钟	130

第 12 章 集成电路 (RS08IICV2)

12.1 导言	131
12.1.1 模块配置	131
12.1.2 特点	133
12.1.3 操作模式	133
12.1.4 方框图	133
12.2 外部信号描述	134
12.2.1 SCL — 串行时钟线	134
12.2.2 SDA — 串行数据线	134
12.3 注册定义	135
12.3.1 IIC 地址登记册 (IICA)	135
12.3.2 IIC 分频器寄存器 (IICF)	135
12.3.3 IIC 控制登记册	
(IICC1)	138
12.3.4 IIC 状态登记册 (IICS)	139
12.3.5 IIC 数据 I/O 寄存器	
(IICD)	140
12.3.6 IIC 控制寄存器	2
(IICC2)	141
12.4 功能描述	142

12.4.1 IIC 协 议	142
12.4.2 10 位地址	146
12.4.3 一般呼叫地址	147
12.5 重 置	147
12.6 中断	147
12.6.1 字节传输中断	147
12.6.2 地址检测中断	147
12.6.3 仲裁丢失中断	147
12.7 初始化/应用信息	149

第 13 章

模块化计时器 (RS08MTIMV1)

13.1 导言	151
13.1.1 特点	153
13.1.2 操作模式	153
13.1.3 方框图	154
13.2 外部信号描述	154
13.3 注册定义	154
13.3.1 MTIM 状态和控制登记册 (MTIMSC)	155
13.3.2 MTIM 时钟配置寄存器 (MTIMCLK)	156
13.3.3 MTIM 计 数 器 寄 存 器 (MTIMCNT)	156
13.3.4 MTIM 模 块 寄 存 器 (MTIMMOD)	157
13.4 功能描述	158
13.4.1 MTIM 操作示例	159

第 14 章

16 位定时器/PWM (RS08TPMV2)

14.1 导言	161
14.1.1 特点	163
14.1.2 框图	163
14.2 外部信号描述	165
14.2.1 外部 TPM 时钟源	165
14.2.2 TPMxCHn — TPMx 通 道 n I/O 引 脚	165

14.3 注册定义	165
14.3.1 计时器状态和控制寄存器 (TPMxSC)	166
14.3.2 计时器计数器寄存器 (TPMxCNTH : TPMxCNTL)	167
14.3.3 定时器计数器模寄存器 (TPMxMODH: TPMxMODL)	168
14.3.4 计时器通道 n 状态和控制寄存器 (TPMxCnSC)	169
14.3.5 计时器通道值寄存器 (TPMxCnVH: TPMxCnVL)	170
14.4 功能描述	171
14.4.1 柜台	171
14.4.2 通道模式选择	172
14.4.3 中心对齐 PWM 模式	174
14.5 TPM 中断	175
14.5.1 清除计时器中断标志	175
14.5.2 计时器溢出中断描述	175
14.5.3 通道事件中断描述	176
14.5.4 PWM 结束工作周期事件	176

第 15 章 开发支持

15.1 导言	177	
15.2 特点	177	
15.3 RS08 后台调试控制器 (BDC)	178	
15.3.1 BKGD 引脚描述	179	
15.3.2 通信详情	179	
15.3.3 SYNC 和串行通信超时	182	
15.4 BDC 寄存器和控制位	183	
15.4.1 BDC 状态和控制登记册 (BDCSCR)	183	
15.4.2 BDC 断点匹配注册	184	
15.5 命令	185	
RS08	BDC	命

深圳市南天星

第 1 章

MC9RS08KA8 设备概述

1.1 概述

MC9RS08KA8 系列微控制器单元 (MCU) 是一种极低成本、小针数、高性能设备，用于家用电器、医疗设备和通用微控制器。该设备由标准的片上模块组成，包括一个非常小且高效的 RS08 CPU 核心，254 字节 RAM，8K 字节闪存，两个 8 位模定时器，12 通道 10 位 ADC，2 通道 16 位定时器/PWM，集成电路总线模块，键盘中断和模拟比较器。德维 Ce 有 16 针和 20 针的封装。

1.2 MCU 方块图

方框图，图 1-1，显示 MC9RS08KA8 MCU 结构。

深圳市南天星

第 1 章 MC9RS08KA8 设备概述

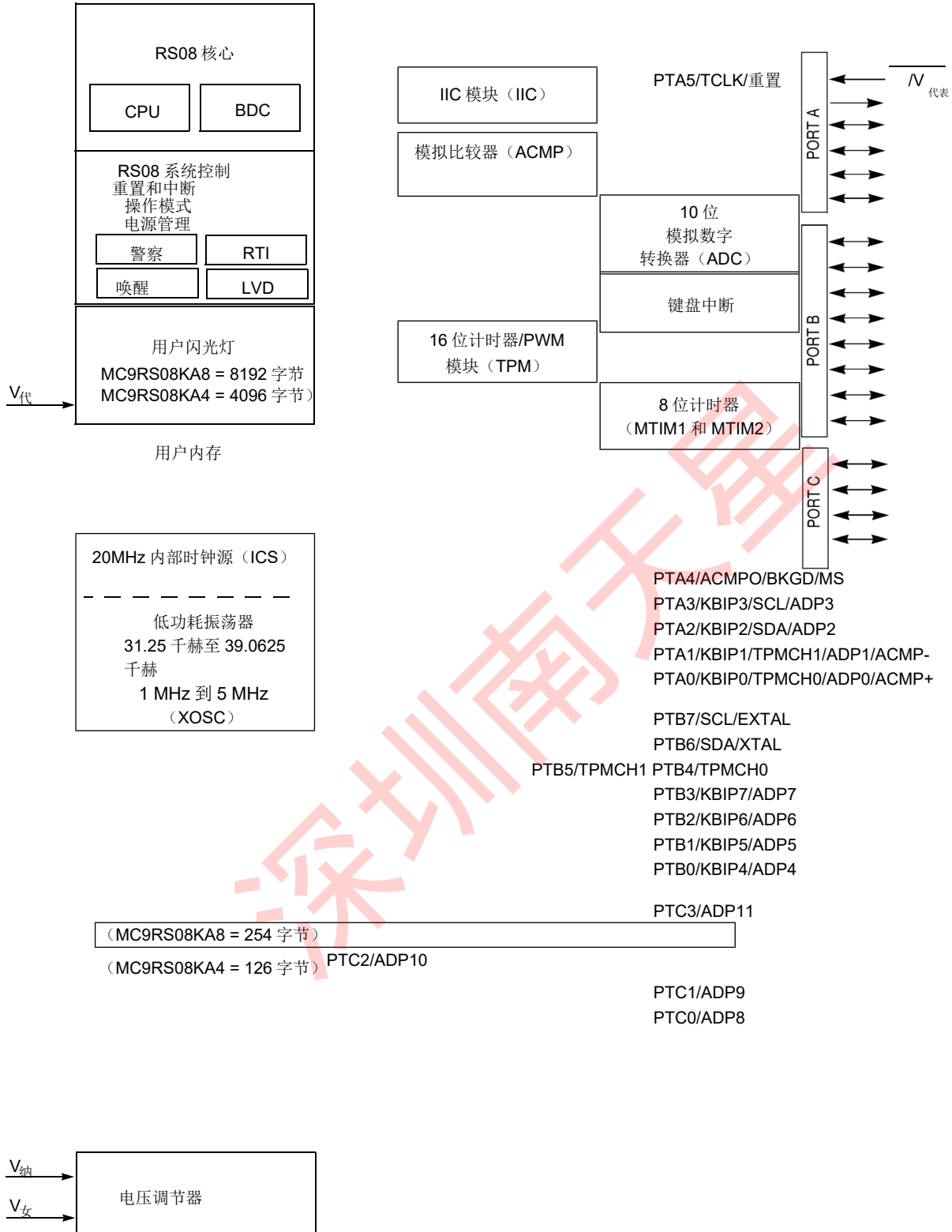


图 1-1. MC9RS08KA8 系列方框图

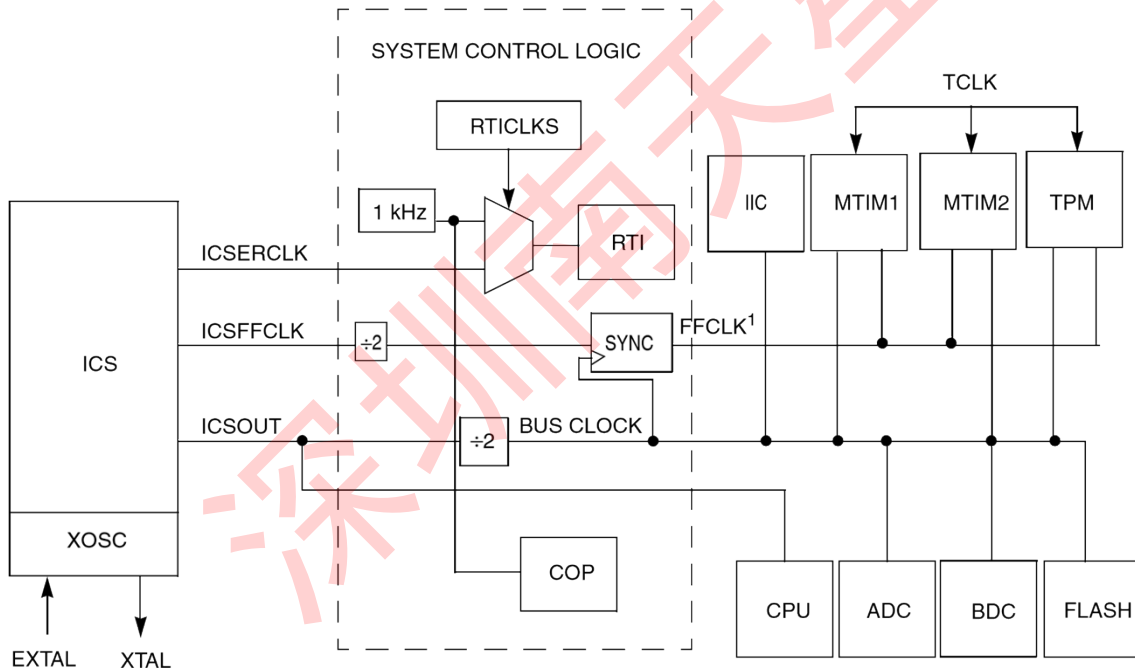
表 1-1 提供片上模块的功能版本。

表 1-1. 块版本

模块	变种
RS08 CPU	1
模拟比较器 (RS08 ACMP)	1
键盘中断 (RS08 KBI)	1
模块化计时器 (RS08 MTIM)	1
内部时钟源 (RS08 ICSOSC)	1
模拟数字转换器 (RS08 ADC10)	1
集成电路 (RS08 IIC)	2
16 位计时器/PWM (RS08 TPM)	2
XOSC	1

第 1 章 MC9RS08KA8 设备概述

1.3 系统时钟分配



¹ 固定频率时钟 (FFCLK) 在内部与总线时钟同步, 不得超过总线时钟频率的一半。

图 1-2. 系统时钟分布图

图 1-2 显示 MCU 的简化时钟连接图。总线时钟频率是 ICS 输出频率的一半, 并被所有内部模块使用。

第 1 章 MC9RS08KA8 设备概述

第 2 章 引脚和连接

2.1 简单介绍

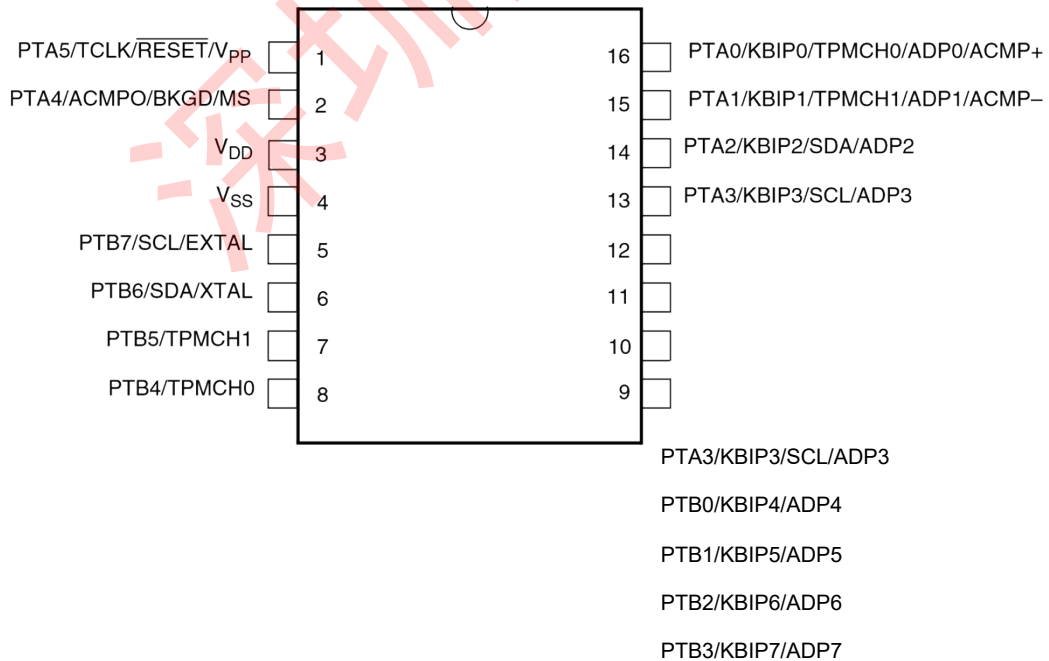
本章介绍了连接到封装引脚的信号。它包括引脚图、信号属性表和详细的信号讨论。

2.2 设备引脚分配

图 2-1 和图 2-2 在 MC9RS08KA8 系列的软件包中显示引脚分配。



Figure 2-1. MC9RS08KA8 Series 20-Pin PDIP/SOIC Package



PTC0/ADP8

PTC1/ADP9

PTB0/KBIP4/ADP4

PTB1/KBIP5/ADP5

PTB2/KBIP6/ADP6

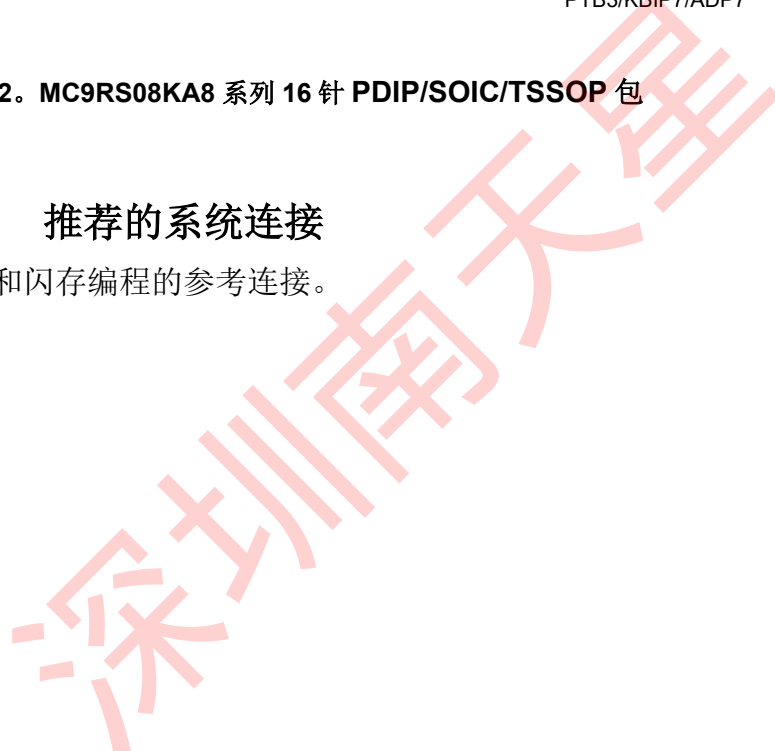
PTB3/KBIP7/ADP7

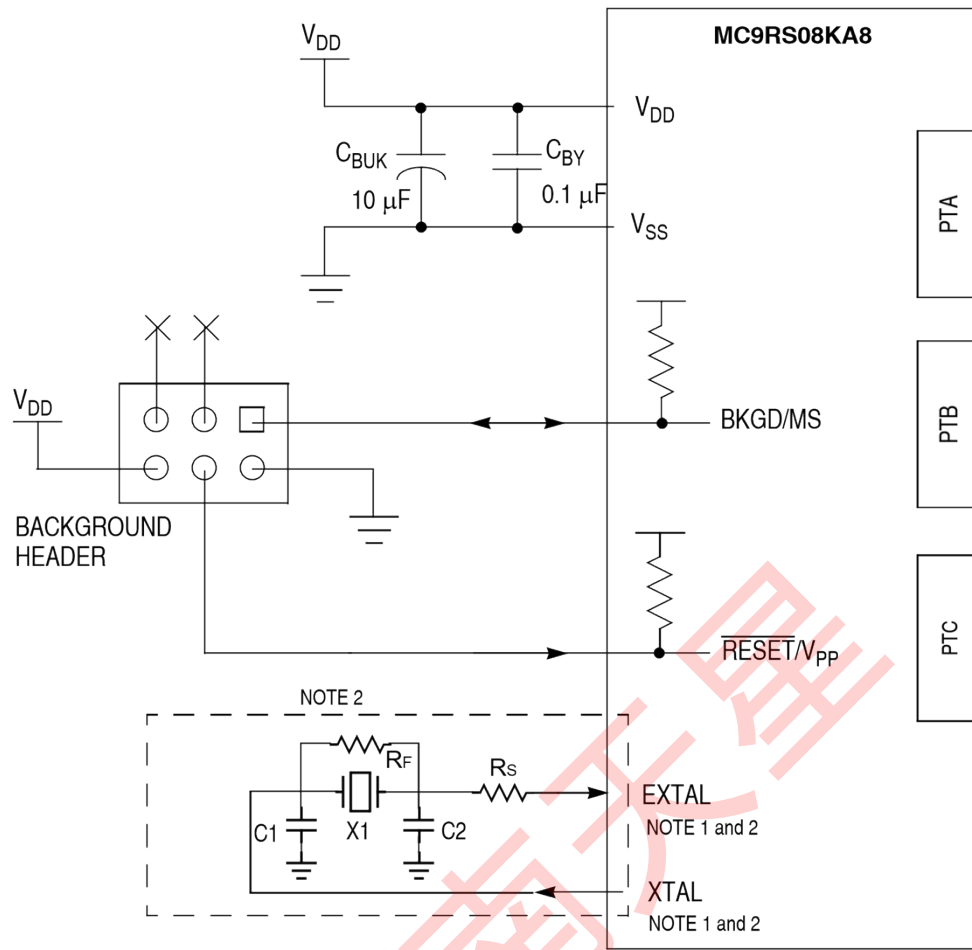
图 2-2。MC9RS08KA8 系列 16 针 PDIP/SOIC/TSSOP 包

第 2 章 引脚和连接

2.3 推荐的系统连接

图 2-3 显示后台调试和闪存编程的参考连接。





NOTES:
¹ Not required if using the internal clock option.
² 这些引脚与 PTB6 和 PTB7 相同。

图 2-3。参考系统连接图

2.4 别针细节

本节提供了系统连接的详细说明。

2.4.1 电源别针

V_{女儿} 和 V_{纳粹党卫军} 是 MCU 的主要电源引脚。该电压源为所有 I/O 缓冲电路和内部稳压器供电。内部电压调节器为 CPU 和其他 MCU 内部电路提供调节的低压源。

通常，应用系统在电源引脚上有两个单独的电容器：一个散装电解电容器，如 10MF 钽电容器，为整个系统提供批量充电存储，以及

第 2 章 引脚和连接

旁路电容器，如 0.1MF 陶瓷电容器，靠近 MCU 电源销，可以抑制高频噪音。

2.4.2 PTA5/TCLK/RESET/V_{代表} 别针

开机重置（POR）进入用户模式后，PTA5/TCLK/RESET/V_{代表} 引脚默认为通用输入端口引脚，

PTA5。在 SOPT 中设置 RSTPE 将引脚配置为 RESET 输入

别针。配置为 RESET 后，引脚将保持为 RESET，直到下一次重置。重置引脚可以

当引脚低时，用于从外部源重置 MCU。当启用为 RESET 时引脚（RSTPE = 1），内部上拉设备自动启用。

外部 V_{代表} 电压（通常为 12V，见 MC9RS08KA8 系列 *数据表*）在执行闪存编程或擦除时，此引脚上需要。V_{代表} 无论引脚功能如何，连接始终连接到内部闪存模块。为了避免对闪光灯造成过度压力，外部 V_{代表} 电压必须消除，电压必须高于 V_{女儿} 当不发生闪存编程或擦除时，必须避免。

笔记

此引脚不包含 V 的夹钳二极管 _{女儿} 不得开到 V 以上 _{女儿} 当没有发生闪存编程或擦除时。

2.4.3 PTA4/ACMPO/BKGD/MS 引脚

后台/模式选择功能与 PTA4 引脚上的仅输出引脚和可选的模拟比较器输出共享。在重置期间，引脚作为模式选择引脚。重置上升后，该大头针立即起到背景大头针的作用，并且可用于后台调试通信。在作为后台/模式选择引脚时，该引脚启用了内部上拉设备。要用作仅输出端口，请在 SOPT 中清除 BKGDPE。

如果没有连接到此引脚，MCU 将在重置的上升边缘进入正常运行模式。如果调试系统连接到 6 针标准后台调试头，它可以在开机重置期间将 BKGD/MS 保持低，从而强制 MCU 进入活动后台模式。

BKGD 引脚主要用于后台调试控制器（BDC）通信，使用自定义协议，每位时间使用目标 MCU 的 BDC 时钟的 16 个时钟周期。目标 MCU 的 BDC 时钟等于总线时钟速率；因此，显著的 c 电容不得连接到可能干扰后台串行通信的 BKGD/MS 引脚。

虽然 BKGD 引脚是一个伪开放引脚，但后台调试通信协议提供了简短、主动驱动、高速的脉冲，以确保快速上升时间。内部上拉装置的小电容不会影响 BKGD 引脚的上升和下降时间。

2.4.4 通用 I/O 和外围端口

其余引脚在通用 I/O 和片上外围功能之间共享，如计时器和模拟比较器。重置后，所有这些引脚都立即配置为高阻抗通用输入，内部上拉/下拉设备被禁用。

第 2 章 引脚和连接

笔记

为了避免来自浮动输入引脚的额外电流消耗，应用程序中的重置初始化例程必须启用片上拉/下拉设备或将未使用的引脚的方向更改为输出。

表 2-1. 按包引脚计数的引脚可用性

别针数		←—最低 优先地位 —→ 最高				
20	16	端口别针	Alt 1	阿尔特 2	备选 3	备选 4
1	1	PTA5		TCLK	调整	V 代表
2	2	PTA4	ACMPO	BKGD	女士	
3	3					V 女儿
4	4					V 纳粹党卫军
5	5	PTB7	SCL ¹			极端的
6	6	PTB6	SDA ¹			XTAL
7	7	PTB5	TPMCH1 ²			
8	8	PTB4	TPMCH0 ²			
9	—	PTC3			ADP11	
10	—	PTC2			ADP10	
11	—	PTC1			ADP9	
12	—	PTC0			ADP8	
13	9	PTB3	KBIP7		ADP7	

¹ IIC pins can be remapped to PTA3 and PTA2

² TPM pins can be remapped to PTA0 and PTA1

14	10	PTB2	KBIP6		ADP6	
15	11	PTB1	KBIP5		ADP5	
16	12	PTB0	KBIP4		ADP4	
17	13	PTA3	KBIP3	SCL ¹	ADP3	
18	14	PTA2	KBIP2	SDA ¹	ADP2	
19	15	PTA1	KBIP1	TPMCH1 ²	ADP1	ACMP-
20	16	PTA0	KBIP0	TPMCH0 ²	ADP0	ACMP+

第 3 章 操作模式

3.1 简单介绍

本章介绍了 MC9RS08KA8 系列操作模式。它还详细介绍了进入每个模式，退出每个模式，以及每种模式的功能。

3.2 特点

- 用于代码开发的主动后台模式
- 等待模式：
 - CPU 关闭以节省电力
 - 系统时钟继续运行 — 保持全电压调节
- 停止模式：
 - 系统时钟停止
 - 所有内部电路都保持通电，以快速恢复

3.3 运行模式

运行模式是 MC9RS08KA8 系列的正常操作模式。当 BKGD/MS 引脚在重置的上升边缘处高时，会选择此模式。在此模式下，CPU 从内部存储器执行代码，从地址 \$3FFD 开始。JMP 指令（opcOde \$BC）操作数位于 3FFE-\$3FFF，必须编程到用户应用程序中，以便正确重置操作。操作数定义了用户程序启动的位置。而不是像 HC08/S08 家族那样使用矢量获取过程，用户程序负责执行 JMP 指令，将程序计数器重新定位到正确的用户程序启动位置。

3.4 主动后台模式

活动后台模式功能通过 RS08 核心中的后台调试控制器（BDC）进行管理。BDC 为在软件开发期间分析 MCU 操作提供了手段。

主动后台模式以四种方式之一进入：

- 当 BKGD/MS 引脚在开机重置（POR）期间或发出后台调试强制重置（BDC_RESET）命令后立即低时
- 当通过 BKGD 引脚接收背景命令时
- 当 BGND 指令被执行时

第 3 章 操作模式

- 当遇到 BDC 断点时

进入活动后台模式后，CPU 将处于暂停状态，等待串行后台命令，而不是执行用户应用程序的指令。

后台命令有两种类型：

- 非侵入性命令——可以在用户程序运行时发出的命令，可以在 MCU 处于运行模式时通过 BKGD 引脚发出。当 MCU 处于活动后台模式时，也可以执行非侵入性命令。非侵入性 E 命令包括：
 - 内存访问命令
 - 带状态的内存访问命令
 - 背景命令
- 活动后台命令——只能在 MCU 处于活动后台模式时执行，包括以下命令：
 - 读取或写入 CPU 寄存器
 - 一次跟踪一个用户程序指令
 - 离开活动后台模式以返回到用户应用程序（GO）

主动后台模式用于在 MCU 首次以运行模式运行之前将用户应用程序代码编程到闪存程序内存中。当 MC9RS08KA8 系列发货时，闪存程序内存通常会被擦除，因此没有程序可以在运行模式下执行，直到闪存最初被编程。活动后台模式还可用于在编程后擦除和重新编程闪存。

有关活动后台模式的更多信息，请参阅第 15 章，“[发展支持](#)。”

3.5 等待模式

通过执行 WAIT 指令进入等待模式。在执行 WAIT 指令后，CPU 进入低功耗状态，其中没有时钟。程序计数器（PC）在 WAIT 指令执行的位置停止。当一个内部发生 rupt 请求：

1. MCU 退出等待模式并恢复处理。

2. PC 增加一个，并获取下一个要处理的指令。

用户程序必须探测唤醒 MCU 的相应中断源，因为不涉及矢量获取过程。

当 MCU 处于等待模式时，并非所有后台调试命令都可以使用。只有 BACKGROUND 命令和 memory-access-with-status 命令可用。Memory-access-with-status 命令不允许内存访问，但它们报告了一个错误防止 MCU 处于停止或等待模式。BACKGROUND 命令可用于将 MCU 从等待模式唤醒并进入活动后台模式。

表 3-1 总结了 MCU 在等待模式下的行为。

表 3-1. 等待模式行为

形式	CPU	数字的外围设备	ICS	ACMP	校准器	I/O 引脚	RTI	ADC
等待	备用物	可选地在	在...上	可选地在	在...上	各州举行	可选地在	可选地在

3.6 停止模式

当系统选项寄存器中设置 STOPE 位时，在执行 STOP 指令时进入停止模式。在停止模式下，CPU 和模块的所有内部时钟都停止。如果 CPU 执行 STOP 指令时没有设置 STOPE 位，E MCU 不会进入停止模式，并且强制进行非法的操作码重置。

表 3-2 总结了 MCU 在停止模式下的行为。

表 3-2. 停止模式行为

形式	CPU	数字的外围设备	ICS ¹	ACMP ²	校准器 ³	I/O 引脚	RTI	ADC ⁴
阻止	备用物	备用物	可选地在	可选地在	可选地在	各州举行	可选地在	可选地在

¹ ICS 要求 IREFSTEN = 1，并且必须设置 LVDE 和 LVDSE 以允许停止操作。

² 如果需要带隔引用，则在进入停止之前，必须同时设置 SPMS1 中的 LVDE 和 LVDSE 位。

³ 启用 BDM 时，调节器已打开。或者，只有当 LVDE 和 LVDSE 位都在 SPMS1 中设置时，调节器才会打开形式。

⁴ 需要设置 SPMS1 中的异步 ADC 时钟、LVDE 和 LVDSE 位，否则 ADC 处于待机状态形式。

进入停止模式后，MCU 中的所有时钟都会停止。当 IREFSTEN 位被清除并且电压调节器进入待机状态时，ICS 默认关闭。维护所有内部寄存器和逻辑的状态，以及 RAM 内容。保持 I/O 引脚状态。

停止退出是通过断言重置、任何启用的异步中断或实时中断来完成的。异步中断是 KBI 引脚、LVD 中断、ADC 中断或 ACMP 中断。

如果通过断言 RESET 引脚退出停止，则 MCU 将被重置，程序执行从 3FFD 的位置开始。如果通过异步中断或实时中断退出，则相应地执行 STOP 指令的位置之后的下一个指令。用户程序必须探测唤醒 CPU 的相应中断源。

一个单独的自时钟源 (≈ 1 kHz) 用于实时中断，允许在没有外部组件的情况下从停止模式唤醒。当 RTIS = 000 时，实时中断功能被禁用。当 MCU 处于 STOP 模式时，LVD 被禁用，RTICLKS = 1，内部 1 kHz 振荡器被禁用 LED 和功耗更低。

第 3 章 操作模式

外部时钟源也可以启用实时中断，以允许在没有外部组件的情况下从停止模式唤醒。设置 ERCLKEN=1 和 EREFSTEN=1 在停止模式下启用外部时钟源。

要在 STOP 模式下启用 ADC，必须通过设置 LVDE 和 LVDSE 来启用异步 ADC 时钟和 LVD，否则 ADC 处于待机状态。

要使 XOSC 在 STOP 模式下与外部参考时钟源一起运行，必须通过设置 LVDE 和 LVDSE 来启用 LVD。

3.6.1 在停止模式下启用活动 BDM

如果设置了 BDCSCR 中的 ENBDM 位，则会启用从运行模式进入活动后台模式。此寄存器描述在第 15 章，“发展支持。”如果在 CPU 执行 STOP 指令时设置了 ENBDM，则当 MCU 进入停止模式时，系统时钟到后台调试逻辑保持活动状态，因此后台调试通信仍然可能。电压调节器不会进入其低功耗待机状态。它保持完整的内部监管。

大多数后台命令在停止模式下不可用。带状态的内存访问命令不允许内存访问。他们报告了一个错误，表明 MCU 处于停止或等待模式。BACKGROUND 命令可用于将 MCU 从停止中唤醒如果设置了 ENBDM 位，则进入活动后台模式。进入活动后台模式后，所有后台命令都可用。表 3-3 总结启用进入活动后台模式时停止的 MCU 行为。

表 3-3. BDM 启用停止模式行为

形式	CPU	数字的外围设备	ICS	ACMP	校准器	I/O 引脚	RTI	ADC
阻止	备用物	备用物	在...上	可选地在	在...上	各州举行	可选地在	可选地在

3.6.2 在停止模式下启用 LVD

当电源电压低于 LVD 电压时，LVD 系统可以产生中断或复位。当 CPU 执行 STOP 指令时，如果在停止时启用了 LVD (SPMSC1 中的 LVDE 和 LVDSE 位都设置)，则电压调节器仍然处于活动状态。

表 3-4 总结了启用 LVD 时 MCU 在停止时的行为。

表 3-4。LVD 启用停止模式行为

形式	CPU	数字的外围设备	ICS	ACMP	校准器	I/O 引脚	RTI	ADC ¹
阻止	备用物	备用物	可选地在	可选地在	在...上	各州举行	可选地在	可选地在

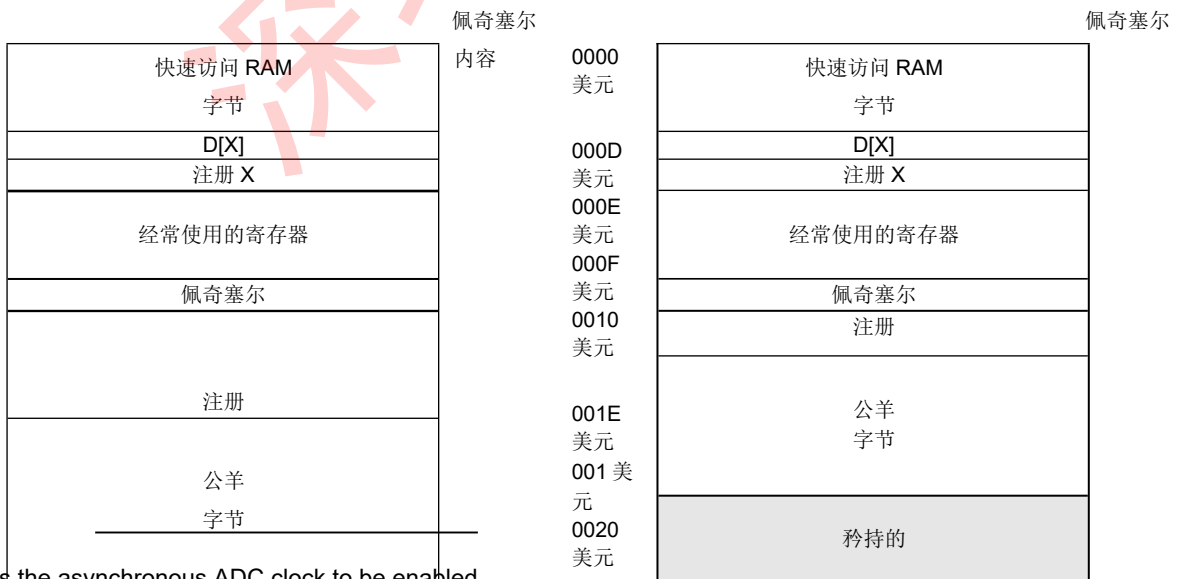
第 4 章 记忆

4.1 记忆地图

MCU 的内存映射分为以下组：

- 使用微小和短的指令快速访问 RAM (0000 美元-0000D)
- 间接数据访问 D[X] (000E)
- D[X]的索引寄存器 X (000F)
- 经常使用的外围寄存器 (0010 美元-001E 美元, \$0020 – \$002F)
- PAGESSEL 寄存器 (001F 美元)
- MC9RS08KA8 的 RAM (0030 美元-0000 美元 BF, 0100 美元-15F 美元)
- MC9RS08KA4 的 RAM (0030 美元-009F 美元)
- 寻呼窗口 (\$00C0 – \$00FF)
- 其他外围寄存器 (0200 美元至 023F 美元)
- MC9RS08KA8 的非易失性内存 (2000 美元-3FFF)
- MC9RS08KA4 的非易失性内存 (3000 美元-3FFF)

第 4 章 记忆



¹ Requires the asynchronous ADC clock to be enabled.

分页窗口	00 美元	002 美元 元 F	分页窗口
公羊 字节		009F 美元 00A0 美元	矜持的
完整的		00 美元 元 BF 00C0 美元	完整的
高页面寄存器			高页面寄存器
完整的			完整的
		00 美元 元 FF 1000 美元	矜持的
闪存 8 KBYTES NVOPT		0.15 美元	闪存 4 KBYTES NVOPT
闪光灯	08 美元 (重 置价值)	0200 美元 023F 美元	闪光灯
	80 美元	1 美元 FFF 2000 美元 2 美元 元 FFF 3000 美元 3 美元 FFB 3 美元 FFC 3 美元 FFD 3 美元 FFF	

\$0000CONTENT

00 美元

000D 美元
\$000E \$000F \$010

\$001E \$001F \$0020 \$002F

MC9RS08KA8 Series Reference Manual, Rev. 4

\$00BF \$00C0

00 美元 FF 0100 美元

0.15 美元

023F 美元

0200 美元 08 美元 (重置值)

1 美元 FFF
2000 美元

C0 美元

3 美元 FFB 3 美元 FFC 3 美元 FFD 3 美元 FFF
MC9RS08KA8

MC9RS08KA4

图 4-1。MC9RS08KA8 系列内存地图

4.2 未实现的内存

尝试在未实现的内存地址访问数据或指令会导致重置。

4.3 索引/间接寻址

寄存器 D[X]和寄存器 X 合并执行间接数据访问。寄存器 D[X]被映射到地址 000E 美元。寄存器 X 位于地址\$000F。8 位寄存器 X 包含注册时使用的地址访问 D[X]。重置时，寄存器 X 被清除为零。通过编程寄存器 X，任何位置

第一页（0000-\$00FF）可以通过寄存器 D[X]读/写。图 4-2 显示了 D[X]和寄存器 X 之间的关系。例如，在 HC08/S08 语法中 *Lda D[X]* 在 RS08 编码中，当寄存器 X 已用索引值编程时。

\$000E 的物理位置在 RAM 中。当寄存器 X 包含 0E 时，通过 D[X]访问该位置会返回\$000E RAM 内容。\$000F 的物理位置是寄存器 X 本身。通过 D[X]读取位置返回寄存器 X 内容。Writing 到位置修改寄存器 X。

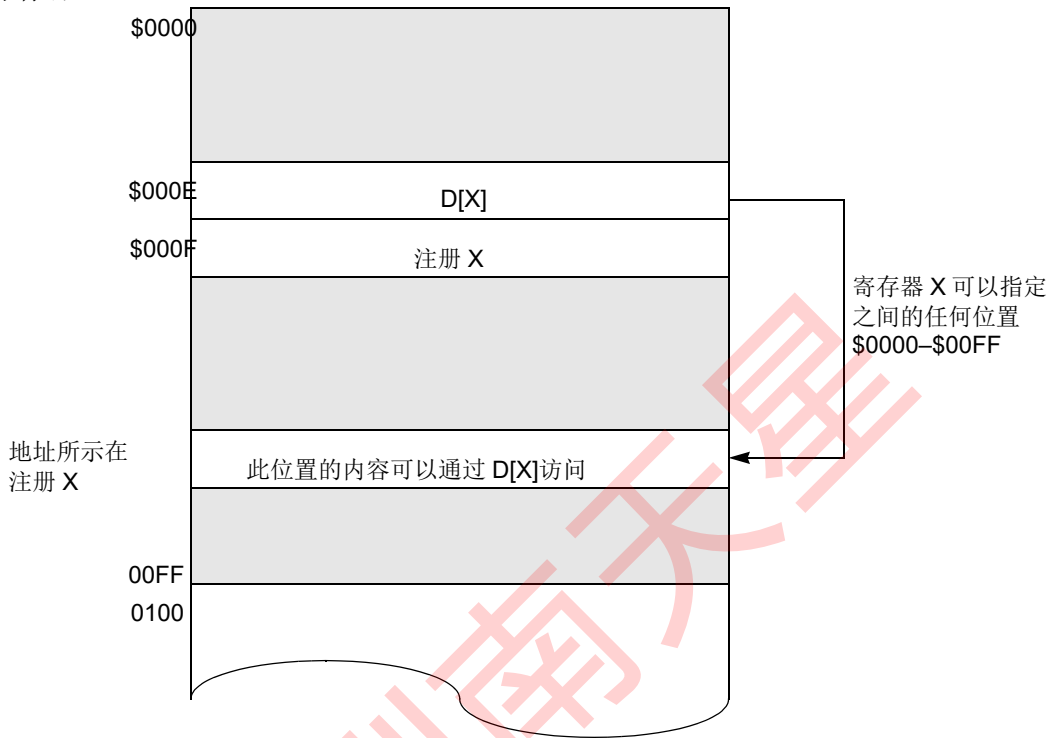


图 4-2。间接寻址寄存器

4.4 RAM 和寄存器地址以及位分配

使用简短的直接寻址模式指令来读取和写入快速访问 RAM 区域。对于小型寻址模式指令，操作数用操作码编码为单个字节。

常用的寄存器可以利用简短的寻址模式指令来加快加载、存储和清除操作。对于简短的寻址模式指令，操作数与操作码一起编码为单个字节。

表 4-1。注册摘要

地址注册名称	7 位	6	5	4	3	2	1	位 0
快速访问 RAM								
	7 位	6	5	4	3	2	1	位 0
	7 位	6	5	4	3	2	1	位 0

COCO	AIEN	ADCO	ADCH				
ADACT	ADTRG	ACFE	ACFGT	—	—	—	—
0	0	0	0	0	0	ADR9	ADR8
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0	0	0	0	KBF	KBACK	KBIE	KBMOD
CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
Bit15	14	13	12	11	10	9	Bit8
Bit7	6	5	4	3	2	1	Bit0
CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
Bit15	14	13	12	11	10	9	Bit8
Bit7	6	5	4	3	2	1	Bit0
顶点	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD	
0	0	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0	0	0	0	PTCD3	PTCD2	PTCD1	PTCD0
AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6
TOF	TOIE	TRST	TSTP	0	0	0	0
0	0	CLKS		记			
总数							
摩登派							
TOF	TOIE	TRST	TSTP	0	0	0	0
0	0	CLKS		记			
总数							
摩登派							

\$0000 - \$000D	\$000E	AD7	AD6	AD5	AD4	AD3	AD2	AD1	0					
D[X]1\$000F	X 00	MULT		ICR										
美元 10 ADCSC1	\$0011	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0					
ADCSC2	\$0012													
ADCRH	\$0013													
ADCRL	\$0014													
KBISC	\$0015													
TPMC0SC	\$0016	TCF	IAAS	无暇的	ARBL	0	SRW	IICIF	RXAK					
TPMC0VH	\$0017													
TPMC0VL	\$0018	数据												
TPMC1SC	\$0019													
TPMC1VH	\$001A													
TPMC1VL	\$001B	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8					
ACMPSC	\$001C													
PTAD	\$001D	—	—	—	—	—	—	—	—					
PTBD	\$001E													
PTCD	\$001F	—	—	—	—	—	—	—	—					
PAGESEL	\$0020													
MTIM1SC	\$0021													
MTIM1CLK	\$0022	公羊												
MTIM1CNT	\$0023													
MTIM1MOD	\$0024													
MTIM2SC	\$0025	分页窗口												
MTIM2CLK	\$0026													
MTIM2CNT	\$0027	MTIM2MOD	\$0028	IICA	\$0029	IICF	\$002A	IICC1	\$002B	IICS	\$002C	IICD	\$002D	IICC2

\$002E 未售出 \$002F 未售出 \$0030 -

\$00BF \$00C0 - \$00FF

表 4-1. 注册摘要 (续)

地址注册名称

7 位	6	5	4	3	2	1	位 0
公羊							
—	—	—	—	—	—	—	—
POR	别针	警察	ILOP	ILAD	0	LVD	0
长袍	科普特人	STOPE	IICPS	TPMCH1PS	TPMCH0PS	BKGDPE	RSTPE
IIC	KBI	ACMP	ADC	TPM	MTIM2	MTIM1	RTI
REV3	REV2	REV1	REV0	ID[11:8]			
ID[7:0]							
RTIF	RTIACK	RTICLKS	RTIE	0	RTIS		
LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	BGBE

0100 美元 – 015F								
\$0160 – 未售出 \$01FF								
0200 美元 SRS \$0201 SOPT \$0202 SIP1	—	—	—	—	—	—	—	—
0203 美元 未 售出 \$0204 保留 \$0205 未 得 到充分	0	0	0	0	0	0	0	SECD
	0	0	0	0	HVEN	质量	0	PGM
0206 美元 SDIDH \$0207 SDIDL \$0208 SRTISC \$0209 SPMSC1	0	0	0	0	0	0	ADCV9	ADCV8
020A 美元保 留 \$020B 保留 \$020C – 未售出	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
\$020F	ADLPC	ADIV		ADLSMP	形式		ADICLK	
\$0210 FOPT \$0211 FLCR	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
	0	0	0	0	ADPC11	ADPC10	ADPC9	ADPC8
0212 美元 – 保留 \$0213								
0214 美元 ADCCVH \$0215 ADCCVL \$0216 ADCCFG \$0217 APCTL1 \$0218 APCTL2	0	0	0	0	PTADD3	PTADD2	PTADD1	PTADD0
	0	0	PTAPE5	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0
	0	0	PTAPUD5	0	PTAPUD3	PTAPUD2	PTAPUD1	PTAPUD0
\$0219 – 未售出 \$021F	0	0	0	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
0220 美元 PTADD \$0221 PTAPE \$0222 PTAPUD \$0223 PTASE \$0224 PTBDD \$0225 PTBPE \$0226 PTBPUD \$0227 PTBSE \$0228 PTCDD \$0229 PTCPE \$022A PTCPUD \$022B PTCSE	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
	PTBPUD7	PTBPUD6	PTBPUD5	PTBPUD4	PTBPUD3	PTBPUD2	PTBPUD1	PTBPUD0
	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
	0	0	0	0	PTCDD3	PTCDD2	PTCDD1	PTCDD0
	0	0	0	0	PTCPE3	PTCPE2	PTCPE1	PTCPE0
	0	0	0	0	PTCPUD3	PTCPUD2	PTCPUD1	PTCPUD0
	0	0	0	0	PTCSE3	PTCSE2	PTCSE1	PTCSE0
022C 未 提供 022D 未售出 \$022E 未 得 到充分	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—

表 4-1。注册摘要 (续)

地址注册名称	7 位	6	5	4	3	2	1	位 0
	—	—	—	—	—	—	—	—
TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0	
Bit15	14	13	12	11	10	9	8	Bit8
Bit7	6	5	4	3	2	1	0	Bit0

022F 美元 未得到充分	Bit15	14	13	12	11	10	9	Bit8
0230 美元 TPMSC \$0231	Bit7	6	5	4	3	2	1	Bit0
TPMCNTH \$0232								
TPMCNTL \$0233	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
TPMMODH \$0234	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBEDG3	KBEDG2	KBEDG1	KBEDG0
TPMMODL \$0235	0	0	0	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
KBIPE \$0236								
KBIES \$0237	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
PTADS \$0238								
PTBDS \$0239	0	0	0	0	PTCDS3	PTCDS2	PTCDS1	PTCDS0
PTCDS								
023A 美元未售出 未得到充分								
\$023C ICSC1 \$023D	CLKS		RDIV		IREFS	IRCLKEN	IREFSTEN	
ICSC2 \$023E								
ICSTRM \$023F	BDIV		范围	HGO	LP	EREFS	ERCLKEN	EREFSTEN
ICSSC	修剪							
	0	0	0	0	CLKST		OSCINIT	FTRIM
\$3FF8 保留 \$3FF9 保留	—	—	—	—	—	—	—	—
\$3FFA ² 保留 \$3FFB ² 矜持的	—	—	—	—	—	—	—	—
	保留用于室温 ICS 装饰							
3FFC 美元 NVOPT	矜持的							FTRIM
	0	0	0	0	0	0	0	SECD
	=未充分或保留							

1 当索引寄存器 X 的内容为 0E 时，\$000E 的物理 RAM 可以通过 D[X] 寄存器访问。

2 如果使用未修剪的 MCU，应用程序可能会使用 3FFA 和 3FFB 美元。

ICS 工厂修剪后的值将存储在 0x3FFA 和 0x3FFB（位 0）中。工厂修剪的总线频率为 10 MHz。

4.5 公羊

该设备包括三个静态 RAM 部分。使用更高效的微小寻址模式指令和短寻址模式指令可以直接访问从 0000 美元到 000D 美元的位置。位置 \$000E RAM 可以通过 D[X] reg 访问当寄存器 X 为 0E 时，或者当 PAGESEL 寄存器为 00 美元时，通过寻呼窗口位置 \$00CE。RAM 的第二部分从 030 美元到 00BF 美元不等，可以使用直接寻址模式指令访问。RAM 的第三部分从 \$0100 至 015F。

当 MCU 处于低功耗等待和停止模式时，RAM 会保留数据。如果电源电压不低于 RAM 保留的最低值，RAM 数据不会受到任何重置影响。

4.6 闪光

闪存用于程序存储。电路内编程允许在应用产品最终组装后将操作程序加载到闪存中。您可以通过单线后台调试接口对整个阵列进行编程。由于该设备不包括片上充电泵电路，外部 V_{PP} 是编程和擦除操作所必需的。

4.6.1 特点

闪存功能包括：

- 在典型电压和温度下高达 1000 个程序/擦除循环
- 闪存的安全功能

4.6.2 闪存编程程序

闪存是按行编程的。一行由 64 个连续字节组成，从地址 $\$2X00$ 、 $\$2X40$ 、 $\$2X80$ 或 $\$2XC0$ 开始。要编程一行闪存：

1. 应用外部 V 代表。
2. 设置 PGM 位。这配置了用于程序操作的内存，并允许锁定用于编程的地址和数据。
3. 在要编程的行的地址范围内，通过高页面访问窗口 $\$00C0$ – $\$00FF$ 将任何数据写入任何闪存位置。（在数据写入操作之前，必须正确配置 PAGESEL 寄存器，以映射高页面访问 G 窗口到相应的闪存行。）
4. 等一会儿， t_{Nvs} 。
5. 设置 HVEN 位。
6. 等一会儿， t_{pgs} 。
7. 将数据写入要编程的闪存位置。
8. 等一会儿， t_{Prog} 。
9. 重复步骤 7 和 8，直到行中的所有字节都经过编程。
10. 清除 PGM 位。
11. 等一会儿， t_{Nvh} 。 12. 清除 HVEN 位。
13. 时间过后， t_{Rcv} ，内存可以在读取模式下再次访问。
14. 移除外部 V 代表。

这个程序序列在整个内存中重复，直到所有数据都编程。

笔记

从闪存位置执行的软件代码无法编程或擦除闪存。要编程或擦除闪存，必须从 RAM 或 BDC 命令执行命令。在擦除或程序序列期间，用户代码不得进入等待或停止。

这些操作必须按照所示顺序执行，否则步骤之间可能会发生其他无关的操作。

4.6.3 闪存大规模擦除操作

要大规模擦除整个闪存：

1. 应用外部 V 代表。
2. 在闪存控制寄存器中设置 MASS 位。
3. 通过高页面访问窗口 \$00C0-\$00FF 将任何数据写入任何闪存位置。（在数据写入操作之前，必须正确配置 PAGESSEL 寄存器，以将高页面访问窗口映射到任何闪存位置。）
4. 等一会儿， t_{Nvs} 。
5. 设置 HVEN 位。
6. 等一会儿 t_{we} 。
7. 清除 MASS 位。
8. 等一会儿， t_{Nvh1} 。
9. 清除 HVEN 位。
10. T 之后 R_{cv} 时间，内存可以在读取模式下再次访问。
11. 移除外部 V 代表。

笔记

从闪存位置执行的软件代码无法编程或擦除闪存。要编程或擦除闪存，必须从 RAM 或 BDC 命令执行命令。在擦除或程序序列期间，用户代码不得进入等待或停止。

这些操作必须按照所示顺序执行，否则步骤之间可能会发生其他不相关的操作。

4.6.4 安全

MC9RS08KA8 系列包括有助于防止未经授权访问闪存内容的电路。当安全性被启用时，闪光灯是一种安全的资源。RAM、直接页面寄存器和后台调试控制器是不安全的资源。尝试 t_0 如果通过后台调试接口或设置 BKGDPE，则访问安全内存位置将被阻止（读取返回所有 0）。

根据 FOPT 寄存器中非易失性寄存器位（SECD）的状态，进行或脱离安全性。在重置期间，非易失性位置 NVOPT 内容从闪存复制到高页寄存器空间中的工作 FOPT 寄存器中。参与 s 通过编

程 NVOPT 位置来确保安全。您可以在闪存编程时执行此操作。擦除状态 (SECD = 1) 使 MCU 不安全。当 NVOPT 中的 SECD 被编程时 (SECD = 0)，下次通过 POR、内部重置或外部重置，安全性已启动。为了解除安全性，必须通过 BDM 命令执行大规模擦除，然后进行任何重置。单独的后台调试控制器仍然可以用于寄存器和 RAM 访问。通过 BDM 命令，可以通过写入遵循中列出的闪存大规模擦除程序的闪存控制寄存器，可以实现闪存大规模擦除 4.6.3 节“闪存大规模擦除操作。”

当您出现以下情况时，始终可以通过后台调试界面解除安全性：

1. 通过后台 BDM 命令或 RAM 加载程序大规模擦除闪存。
2. 执行重置。该设备启动时安全脱离。

笔记

当设备启动到正常操作模式（重置期间 MS 引脚高），并编程了 SECD (SECD = 0) 时，闪存安全被启用。BKGDPPE 重置为 0，所有 BDM 通信被阻止，不允许后台调试。

4.7 闪存寄存器和控制位

闪存模块在闪存中有一个非易失性寄存器 NVOPT (3FFC)，在重置时将其复制到相应的控制寄存器 FOPT (0210 美元)。

4.7.1 闪存选项寄存器 (FOPT 和 NVOPT)

在重置过程中，非易失性位置 NVOPT 内容从闪存复制到 FOPT。7 到 1 位不使用，总是读 0。这个寄存器可以随时读取，但写入没有任何意义或效果。要更改此寄存器中的值，请擦除并重新编程闪存中的 NVOPT 位置，然后发出新的 MCU 重置。

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

字母 R

罗马字母的第 23 个字母

调整

此寄存器在重置期间从非易失性位置 NVOPT 加载。



= 未充分或保留

图 4-3. Flash 选项寄存器 (FOPT)

表 4-2. FOPT 字段描述

学科	描述
0 SECD	安全国家代码—此位字段决定了 MCU 的安全状态。当 MCU 受到保护时，闪存内容无法通过任何不安全来源（包括后台调试接口）的指令访问；请参阅第 4.6.4 节“安全。”0 安全人员参与。 1 安全脱离。

4.7.2 闪存控制寄存器 (FLCR)

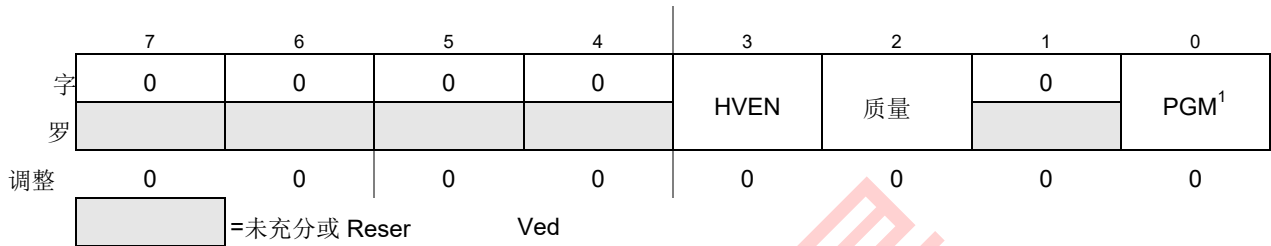


图 4-4. 闪存控制寄存器 (FLCR)

表 4-3. FLCR 字段描述

学科	描述
3 HVEN	高压启用—此读/写位使闪存阵列具有高电压，用于编程和擦除操作。只有当 PGM=1 或 MASS=1 并且遵循正确的程序或擦除顺序时，才能设置 HVEN。 0 高压禁用到阵列。 1 支持阵列的高压。
2 MASS	大规模擦除控制位—此读/写位配置内存以进行大规模擦除操作。0 未选择大规模擦除操作。 1 个批量擦除操作被选中。
0 PGM ¹	程序控制位—此读/写位为程序操作配置内存。PGM 与 MASS 位互锁，因此两个位不能等于 1 或同时设置为 1。 0 未选择程序操作。 1 选择了程序操作。

¹当闪存安全启用时，写入 PGM 位没有任何效果。因此，不允许闪存编程。

4.8 页面选择注册 (PAGESEL)

直接页面中有一个 64 字节的窗口 (00C0 美元 – \$00FF)，预留用于分页访问。编程页面选寄存器确定内存地图上相应的 64 字节块，用于直接页面访问。例如，当 PAGESEL 寄存器是专业的高页寄存器 (0200 美元至 023F 美元) 可以通过直接寻址模式指令通过寻呼窗口 (00C0 美元 – \$00FF) 访问价值为 08 美元。



当 PAGESEL 寄存器为 00 美元时，分页窗口会映射到第一页（00 美元-3F）。分页位置\$00C0 – \$00CE 映射到物理位置\$0000 – \$000E，即 RAM。分页位置\$00CF 映射到注册 X。因此，访问地址\$CE 返回\$000E 的物理 RAM 内容。访问地址\$000E 返回 D[X]注册内容。

深圳市南天星

第 5 章

重置、中断和一般系统控制

5.1 简单介绍

本章讨论了 MC9RS08KA8 系列中的基本重置、中断机制以及各种重置和中断源。本参考手册的其他章节详细讨论了来自外围模块的一些中断源。本章在一个地方收集有关所有重置和中断源的基本信息，以便于参考。一些重置和唤醒源，包括计算机正常运行（COP）监督和实时中断（RTI），不是片上外围系统的一部分 Ems 有自己的章节，但却是系统控制逻辑的一部分。

5.2 特点

重置和中断功能包括：

- 多个重置源，实现灵活的系统配置和可靠运行
- 系统重置状态寄存器（SRS）以指示最近重置的来源
- 系统中断待处理寄存器（SIP1）以指示待系统中断的状态
 - 带有启用的 Analog Comparator 中断
 - 启用键盘中断
 - 带启用的模块定时器中断
 - 启用的实时中断
 - ADC 中断与启用
 - IIC 中断启用
 - 启用 TPM 中断

5.3 MCU 重置

重置 MCU 提供了一种从一组已知条件下开始处理的方法。在重置期间，大多数控制和状态寄存器被迫达到初始值，程序计数器从位置启动

$\$3FFD$ 。操作数位于 $\$3FFE - \$3FFF$ 的 JMP 指令（opcode \$BC）必须编程到用户应用程序中才能正确重置操作。操作数定义用户程序启动的位置。片上外设模块被禁用 I/O 引脚最初配置为通用、高阻抗输入，并禁用上拉/下拉装置。

MC9RS08KA8 系列有七个重置源：

- 外部引脚重置（PIN）——在 SOPT 中使用 RSTPE 启用
- 开机重置（POR）

第 5 章重置、中断和一般系统控制

- 低压检测（LVD）

- 电脑正常运行（COP）计时器
- 非法操作码检测（ILOP）
- 非法地址检测（ILAD）
- 通过 BDC 命令 BDC_RESET 进行后台调试强制重置

除后台调试强制重置外，每个源在系统重置状态寄存器（SRS）中都有一个关联位。

5.4 电脑正常运行（COP）看门狗

如果应用软件无法按预期执行，COP 监督机构用于强制重置系统。为了防止系统从 COP 计时器重置（启用时），应用软件必须定期重置 COP 计数器。如果应用程序 program 丢失，未能在时间结束前重置 COP 计数器，系统重置生成，以迫使系统返回到已知起点。

任何重置后，COPE 都会在 SPT 中设置，这使得 COP 监督机构能够（见 5.8.2 节“系统选项寄存器（SOPT）”）。如果申请中没有使用 COP 监督机构，则通过清除 COPE 将其禁用。COP 计数器通过将任何值写入 SRS 地址来重置。此写入不会影响只读 SRS 中的数据。相反，给这个添加者写信的行为 ess 被解码并向 COP 计数器发送重置信号。

SOPT 中有一个相关的短期和长期超时，由 COPT 控制。表 5-1 总结了 COPT 位控制功能。COP 监督机构从 1kHz 时钟源运行，默认为相关的长时间超时（ 2^8 周期）。

表 5-1. COP 配置选项

科普特人	COP 溢出计数 ¹
0	2^5 周期 (32 ms)
1	2^8 周期 (256 毫秒)

¹ 此列中的值基于 $t_{RTI} \approx 1$ 毫秒 见 t_{RTI} 在“MC9RS08KA8 系列数据表”中，用于公差值。

即使应用程序使用 COPE 和 COPT 的重置默认设置，在重置初始化期间写入一次性 SOPT 寄存器以锁定设置，这样如果应用程序丢失，它们就不会被意外更改。最初的 write to SOPT 重置 COP 计数器。

在后台调试模式下，COP 计数器不会 increment。

当 MCU 进入停止模式时，COP 计数器在进入停止模式时重新初始化为零。一旦 MCU 退出停止模式，COP 计数器就从零开始。

5.5 中断

MC9RS08KA8 系列不包括 HC08 和 HCS08 设备中使用的带有矢量表查找机制的中断控制器。然而，LVD、KBI 等模块的中断源，

ADC、RTI 和 ACMP 仍然可以从等待或停止模式唤醒 CPU。用户应用程序负责轮询相应的模块以确定唤醒的来源。

该模块的每个唤醒源都与相应的中断启用位相关联。如果位被禁用，中断源将被门禁，该特定源无法将 CPU 从等待或停止模式唤醒。然而，相应的中断 flag 仍设置为指示发生了外部唤醒事件。

系统中断待处理寄存器（SIP1）指示系统待中断状态。当启用 S SIP1 的只读位时，它会显示要从指示模块提供服务的待处理中断。写到寄存器位没有效果。足够的当指定模块的所有相应中断标志被取消时，待处理中断标志会自动被取消。

5.6 低电压 Detect (LVD) 系统

MC9RS08KA8 系列包括一个系统，用于保护内存内容，并在电源电压变化期间控制 MCU 系统状态以应对低电压条件。该系统由开机复位（POR）电路和预定义跳闸电压的 LVD 电路组成。LVD 电路在 SPMSC1 中启用了 LVDE。除非 SPMSC1 中设置了 LVDSE，否则进入停止模式时 LVD 将被禁用。如果 LVDSE 和 LVDE 都设置，则启用 LVD 时停止的当前消耗会更大。

5.6.1 开机重置操作

当最初给 MCU 供电时，或者当电源电压降至 V 以下时 V_{POR} 水平，POR 电路导致重置条件。随着电源电压的上升，LVD 电路将 MCU 重置，直到电源上升到 V 以上 V_{LVD} 水平。SRS 中的 POR 位和 LVD 位是在 POR 之后设置的。

5.6.2 LVD 重置操作

LVD 可以通过将 LVDRE 设置为 1 来配置为在检测到低电压条件时生成重置。LVD 重置后，LVD 系统将保持 MCU 重置，直到电源电压上升到 V 级以上 V_{LVD} 。SRS 寄存器中的 LVD 位是在 LVD 重置或 POR 后设置的。

5.6.3 LVD 中断操作

当检测到低电压状态并使用 SPMSC1 配置 LVD 电路进行中断操作（LVDE 集、LVDIE 集和 LVDRE 清除）时，SPMSC1 中的 LVDF 被设置并发生 LVD 中断请求。

5.7 实时中断（RTI）

实时中断功能可用于生成周期性中断。RTI 由 1kHz 内部时钟调制器或外部时钟源驱动。SRTISC 中的 RTICLKS 位用于选择 RTI 时钟源。1 kHz 内部或前当 MCU 处于运行、等待或停止模式时，可以使用 RTI 的外部时钟源。在正常或等待模式下使用外部摆度时，设置 ERCLKEN=1。在停止模式下使用外部摆动器时，设置 ERCLKEN=1 和 ERREFSTEN=1。

第 5 章重置、中断和一般系统控制

SRTISC 寄存器包括一个只读状态标志、一个只写确认位和一个 3 位控制值 (RTIS)，用于选择七个唤醒周期之一或禁用 RTI。RTI 有一个本地中断，使 RTIE 允许屏蔽实时中断。可以通过将 RTIS 的每个位写入 0 来禁用 RTI，并且不会产生中断。看见 5.8.4 节“系统实时中断状态和控制寄存器 (SRTISC)”，以获取更多信息。

5.8 重置、中断和系统控制寄存器和控制位

请参阅直接页面注册摘要第 4 章“Memory”，用于所有寄存器的绝对地址分配。本节仅通过名称提及寄存器和控制位。Freescale 提供的等式或头文件用于将这些名称转换为适当的绝对地址。

SOPT 寄存器中的一些控制位与操作模式有关。虽然这里提供了这些位的简要描述，但相关功能将在第 3 章“操作模式。”

5.8.1 系统重置状态登记册 (SRS)

这个高页面寄存器包括只读状态标志，以指示最近重置的来源。当调试主机通过 BDC_RESET 命令强制重置时，SRS 中的所有状态位都会被解除。将任何值写入此寄存器地址都会清除 tCOP 监督员计时器不会影响本登记册的内容。这些位的重置状态取决于导致 MCU 重置的原因。

7	6	5	4	3	2	1	0
POR	别针	警察	ILOP	ILAD	0	LVD	0
Wr 对 SRS 地址的任何价值都会清除 COP 监督机构 timer。							
1	0	0	0	0	0	1	0

字母 R

罗马字母的第 23 个字母

POR:

LVR: 第二十个罗马字母 0 0 0 0 0 0 1 0

任何其他重置: 0 注 1 注 1 注 1 注 1 0 0 0

¹ 重置输入时处于活动状态的任何重置源都会设置相应的位；重置输入时与未激活的源对应的位将被刷新。

图 5-1. 系统重置状态 (SRS)

表 5-2. SRS 字段描述

学科	描述
7 POR	<p>开机重置—由开机检测逻辑引起的重置。由于当时内部电源电压正在上升，低压复位 (LVR) 状态位也被设置为表明重置发生在内部电源低于 LVR 阈值时发生的。</p> <p>0 重置不是由 POR 引起的。</p> <p>1 POR 导致重置。</p>

6 别针	外部重置引脚 —由外部重置引脚上的主动低电平引起的重置。0 重置不是由外部重置引脚引起的。 1 个外部复位引脚导致复位。
---------	------------------------------------------------------------------------

第 5 章重置、中断和一般系统控制

表 5-2. SRS 字段描述 (续)

学科	描述
5 COP	电脑正常运行 (COP) 看门狗 — COP 监督机构计时器超时导致的重置。COPE = 0 可以阻止重置源。 0 重置不是由 COP 超时引起的。 1 COP 超时导致重置。
4 ILOP	非法操作码 —重置是由试图执行未执行或非法的操作码引起的。如果 SOPT 寄存器中的 STOPE = 0 禁用停止模式, 则 STOP 指令被视为非法。如果 BDCSC 寄存器中的 ENBDM = 0 禁用停止模式, BGND 指令将被视为非法。 0 重置不是由非法操作码引起的。 1 非法操作码导致重置。
3 ILAD	非法地址 —因尝试访问未实现内存地址的数据或指令而导致的重置。 0 重置不是由非法地址引起的。 1 非法地址导致重置。
1 LVD	低电压 Detect —如果设置了 LVDRE 位, 电源下降到 LVD 跳闸电压以下, 则会发生 LVD 重置。POR 设置了这个位。 0 重置不是由 LVD 旅行或 POR 引起的。 1 LVD 旅行或 POR 导致重置。

5.8.2 系统选项注册 (SOPT)

这个高页寄存器是一个写一次寄存器, 因此只有重置后的第一次写入才会受到尊重。它随时都可以阅读。任何后续 (有意或无意) 写入 SOPT 的任何尝试都会被忽略, 以避免意外更改这些敏感设置。SOPT 必须在用户的重置初始化程序期间编写, 以设置所需的控件, 即使所需的设置和重置设置相同。

7	6	5	4	3	2	1	0
长袍	科普特人	STOPE	IICPS	TPMCH1PS	TPMCH0PS	BKGDPE	RSTPE
1	1	0	0	0	0	1 (注 1)	第二十一个 罗马字母

字母 R

罗马字母的第 23 个字母

重置:

POR: 1 1 0 0 0 0 1 (注1) 0 =未充分或保留 U = 未影响

¹当设备开机重置时，BKGEPE 重置为 1。当设备重置为正常运行模式（MS 在重置期间很高），如果闪存安全脱离，BKGDPE 将重置为 1（SECD = 1）。如果启用了闪存安全（SECD = 0），BKGDPE 将重置为 0。当设备重置为活动 BDM 模式（重置期间 MS 较低）时，BKGDPE 始终重置为 1，以便允许 BDM 通信。

图 5-2。系统选项注册 1 (SOPT)

深圳南天星

表 5-3. SOPT 注册字段描述

学科	描述
7 COPE	COP 监督机构启用 —此一次性写入位选择 COP 监督机构是否启用。 0 COP 监督员计时器已禁用。 1 COP 监督员计时器已启用（超时时强制重置）。
6 COPT	COP 看门狗超时 —此一次写入位选择 COP 的超时期。0 选择短超时期。 1 选择的长超时期。
5 STOPE	停止模式启用 —此写入一次位用于启用停止模式。如果停止模式被禁用，并且用户程序试图执行 STOP 指令，则会强制执行非法的操作码重置。 0 停止模式已禁用。 1 启用停止模式。
4 IICPS	IIC Pin Select —此位选择 IIC 模块的 SDA 和 SCL 引脚的位置。 0 PTA2 上的 SDA, PTA3 上的 SCL。 1 PTB6 上的 SDA, PTB7 上的 SCL。
3 TPMCH1PS	TPMCH1 引脚选择 —此位选择 TPM 模块的 TPMCH1 引脚的位置。PTA1 上的 0 TPMCH1。 PTB5 上的 1 TPMCH1。
2 TPMCH0PS	TPMCH0 引脚选择 —此位选择 TPM 模块的 TPMCH0 引脚的位置。 0 PTA0 上的 TPMCH0。 1 PTB4 上的 TPMCH0。
1 BKGDPE ^{1, 2}	后台调试模式引脚启用 —设置后，此一次性写入位使 PTA4/ACMPO/BKGD/MS 引脚能够作为 BKGD/MS 发挥作用。当清晰时，该引脚仅作为其输出的替代功能之一。任何 MCU 重置后，此引脚默认为 BKGD/MS 功能。 0 PTA4/ACMPO/BKGD/MS 引脚的功能是 PTA4 或 ACMPO。 1 PTA4/ACMPO/BKGD/MS 引脚的功能是 BKGD/MS。
0 RSTPE	重置引脚启用 —设置后，此写入一次位启用 PTA5/TCLK/RESET/V _{代表} 引脚作为 重置。当清晰时，该引脚作为其仅输入的替代功能之一。此引脚仅为输入端口 MCU POR 之后的功能。当设置 RSTPE 时，在 RESET 上启用内部上拉设备。 0 PTA5/TCLK/RESET/V _{代表} 作为 PTA5/TCLK/V 的引脚功能代表。

¹ When the device is power on reset, BKGEPE is reset to 1. When the device is reset into normal operating mode (MS is high during reset), BKGDPE is reset to 1 if flash security is disengaged (SECD = 1). BKGDPE is reset to 0 if flash security is engaged (SECD = 0). When the device is reset into active BDM mode (MS is low during reset), BKGDPE is reset to 1 so that BDM communication is allowed.

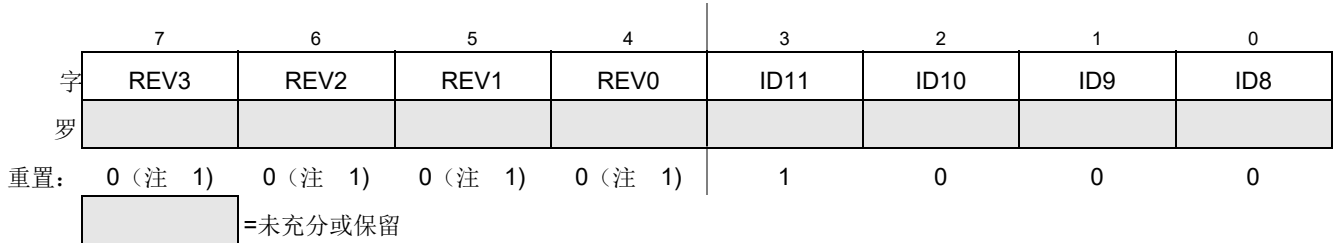
² BKGDPE can write only once from value 1 to 0. Writing from value 0 to 1 by user software is not allowed. BKGDPE can be changed back to 1 only by a POR or reset with proper condition as stated in Note 1.

5.8.3 System Device Identification Register (SDIDH, SDIDL)

These high-page, read-only registers are included so host development systems can identify the RS08 derivative and revision number. This allows the development software to recognize where specific memory blocks, registers, and control bits are located in a target MCU.

	<p>1 PTA5/TCLK/重置^N代表引脚功能为 RESET^N代表。</p>
--	---------------------------------------------------------------

第 5 章重置、中断和一般系统控制



¹硬编码到这些位的修订号反映了当前的硅修订级别。

图 5-3. 系统设备识别寄存器——高 (SDIDH)

表 5-4. SDIDH 寄存器字段描述

学科	描述
7:4 REV[3:0]	修订号—高阶 4 位地址 SDIDH 是硬编码的，以反映当前掩码集修订号 (0–F)。
3:0 ID[11:8]	部件识别号—RS08 家族中的每个衍生品都有一个唯一的识别号。MC9RS08KA8 被硬编码为 0803 美元。另请参阅 ID 位图 5-4。

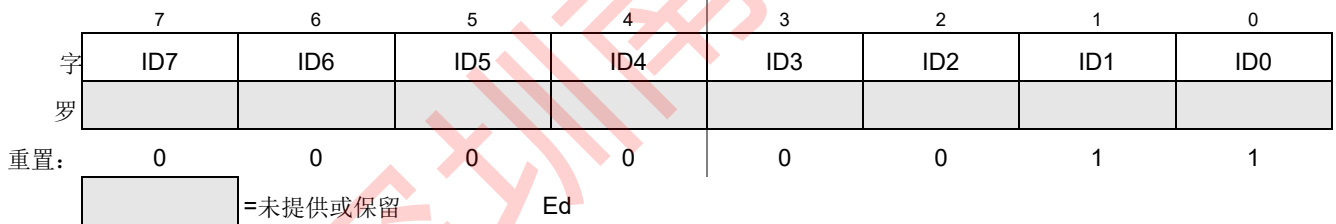


图 5-4. 系统设备识别寄存器——低 (SDIDL)

表 5-5. SDIDL 注册字段描述

学科	描述
7:0 ID[7:0]	部件识别号—RS08 家族中的每个衍生品都有一个唯一的识别号。MC9RS08KA8 被硬编码为 0803 美元。另请参阅 ID 位图 5-3。

5.8.4 系统实时中断状态和控制寄存器 (SRTISC)

这个高页寄存器包含 RTI 的状态和控制位。

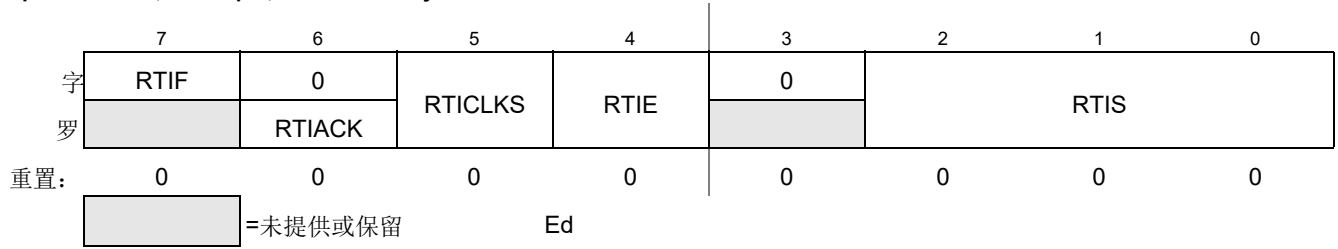


图 5-5. 系统 RTI 状态和控制寄存器 (SRTISC)

表 5-6. SRTISC 注册字段描述

学科	描述
7 RTIF	实时中断标志 —只读状态位表示定期唤醒计时器已过时。0 定期唤醒计时器未过时。 1 个定期唤醒计时器已过时。
6 RTIACK	实时中断确认 。一只写位确认实时中断请求 (写 1 以清除 RTIF)。写 0 没有意义或效果。读取总是返回 0。
5 RTICLKs	实时中断时钟选择 —读/写位选择实时中断的时钟源。 0 实时中断请求时钟源是内部 1kHz 的发声器。 1 实时中断请求时钟源是外部时钟。
4 RTIE	实时中断启用 —读写位允许实时中断。 0 禁用实时中断。 1 启用实时中断。
2:0 RTIS	实时中断延迟选择 —这些读/写位选择 RTI 的周期。看见表 5-7。

表 5-7. 实时中断期

RTIS	使用 1kHz 振荡器源 ¹	使用外部时钟输入
000	禁用并清除 RTI 计数器	禁用并清除 RTI 计数器
001	8 毫秒	$1/f_{Extclk} \times 256$
010	32 毫秒	$1/f_{Extclk} \times 1024$
011	64 毫秒	$1/f_{Extclk} \times 2048$
100	128 毫秒	$1/f_{Extclk} \times 4096$
101	256 毫秒	$1/f_{Extclk} \times 8192$
110	512 毫秒	$1/f_{Extclk} \times 16384$
111	1.024 秒	$1/f_{Extclk} \times 32768$

第 5 章重置、中断和一般系统控制

¹ Values are shown in this column based on $f_{RTI} = 1 \text{ kHz}$. Consult electricals specification from MC9RS08KA8 Series *Data Sheet*.

5.8.5 系统电源管理状态和控制 1 寄存器 (SPMSC1)

此高页寄存器包含状态和控制位，以支持低压检测功能，并启用带隙电压参考供 ACMP 和 LVD 模块使用。

7	6	5	4	3	2	1	0
LVDF	0	LVDIE	LVDRE ¹	LVDSE	LVDE ¹	0	BGBE
	LVDACK						
0	0	0	1	1	1	0	0

=未充分或保留
 重置后只能写一次。其他写入会被忽略。

字母 R

罗马字母的第 23 个字母

重置:

1

图 5-6. 系统电源管理状态和控制 1 寄存器 (SPMSC1)

表 5-8. SPMSC1 寄存器字段描述

学科	描述
7 LVDF	低电压 Detect Flag —如果 LVDE = 1，此只读状态位表示低压检测事件。
6 LVDACK	低压式识别确认 —只写位用于确认低电压检测错误（写 1 以清除 LVDF）。读取总是返回 0。
5 LVDIE	低电压 Detect Interrupt Enable —启用 LVDF 的硬件中断请求。 0 禁用硬件中断（使用轮询）。 1 当 LVDF = 1 时请求硬件中断。
4 LVDRE	启用低电压 Detect Reset —写一次位使低压检测事件生成硬件重置（前提是 LVDE = 1）。 0 LVDF 不会产生硬件重置。 1 LVDF = 1 时强制 MCU 重置。
3 LVDSE	低压 Detect Stop Enable —如果 LVDE = 1，该读/写位将确定低压检测功能在 MCU 处于停止模式时运行。 0 在停止模式下禁用低压检测。 1 在停止模式下启用低压检测。

2 LVDE	<p>启用低电压 Detect—一次性写入位可实现低压检测逻辑和该寄存器中其他位的操作。</p> <p>0 LVD 逻辑被禁用。</p> <p>1 LVD 逻辑启用。</p>
0 BGBE	<p>带隙缓冲区启用—启用带宽电压参考的内部缓冲区，供 ACMP 模块在其内部通道之一上使用。</p> <p>0 Bandgap 缓冲区已禁用。</p> <p>1 启用带宽缓冲区。</p>

5.8.6 系统中断待定寄存器 (SIP1)

这个高页寄存器包含模块待处理中断的状态。

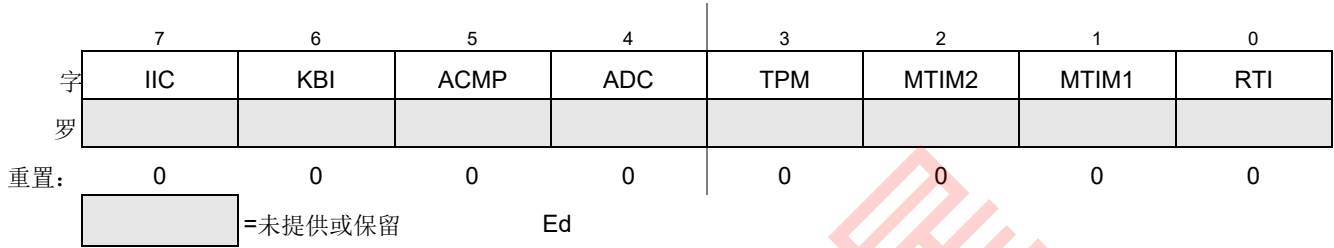


图 5-7. 系统中断待定寄存器 (SIP1)

表 5-9. SIP1 寄存器字段描述

学科	描述
7 IIC	<p>IIC 中断待决—只读位表示来自 IIC 模块的待处理中断。清除 IICCS 登记册的 IICIF 标志可以清除这一点。重置也会清除此位。</p> <p>0 没有待处理的 IIC 中断；即，IICIF 标志和/或 IICIE 位被取消。</p> <p>1 等待 IIC 中断；即设置了 IICIF 标志和 IICIE 位。</p>
6 KBI	<p>等待键盘中断—只读位表示 KBI 模块的待处理中断。清除 KBISC 寄存器的 KBF 标志可以清除此位。重置也会清除此位。</p> <p>0 没有悬而未决的 KBI 中断；即 KBF 标志和/或 KBIE 位被取消。</p> <p>1 等待 KBI 中断；即设置 KBF 标志和 KBIE 位。</p>
5 ACMP	<p>等待类似比较器中断—只读位表示来自 ACMP 模块的待处理中断。清除 ACMPSC 寄存器的 ACF 标志可以清除此位。重置也会清除此位。</p> <p>0 没有待处理的 ACMP 中断；即，ACHF 标志和/或 ACIE 位被取消。</p> <p>1 等待 ACMP 中断；即设置了 ACF 标志和 ACIE 位。</p>
4 ADC	<p>ADC 中断待处理—只读位表示 ADC 模块的待中断。清除 ADCSC1 寄存器的 COCO 标志可以清除此位。重置也会清除此位。</p> <p>0 没有未处理的 ADC 中断，即 COCO 标志和/或 AIEN 位被批准。</p> <p>1 P 结束 ADC 中断，即设置了 COCO 标志和 AIEN 位。</p>
3 TPM	<p>计时器/PWM 中断待处理—只读位表示来自 TPM 的待处理中断模块。清除 TPMSC 寄存器的 TOF 标志和/或 TPMCnSC 寄存器的 CHnF 标志可以清除此位。重置也会清除此位。</p> <p>0 没有未处理的计时器/PWM 中断，即 TOF 标志和/或 TOIE 位都被取消。；或 CH0F 标志和/或 CH0IE 位被 cleared。；或 CH1F 标志和/或 CH1IE 位被 cleared。</p> <p>1 等待计时器/PWM 中断，即设置 TOF 标志和 TOIE 位。；或者设置了 CH0F 标志和 CH0IE 位。；或者设置了 CH1F 标志和 CH1IE 位。</p>

2 MTIM2	Modulo Timer 2 中断待处理 —只读位表示 MTIM2 模块的待处理中断。清除 MTIM2SC 寄存器的 TOF 标志可以清除此位。重置也会清除此位。 0 没有待处理的 MTIM2 中断；即 TOF 标志和/或 TOIE 位被取消。 1 等待 MTIM2 中断；即设置了 TOF 标志和 TOIE 位。
1 MTIM1	Modulo 计时器 1 中断待处理 —只读位表示 MTIM1 模块的待处理中断。清除 MTIM1SC 寄存器的 TOF 标志可以清除此位。重置也会清除此位。 0 没有待处理的 MTIM1 中断；即 TOF 标志和/或 TOIE 位被取消。 1 等待 MTIM1 中断；即设置了 TOF 标志和 TOIE 位。
0 RTI	实时中断待处理 —只读位表示来自 RTI 的待处理中断。清除 SRTISC 寄存器的 RTIF 标志可以清除此位。重置也会清除此位。 0 没有待处理的 RTI 中断；即 RTIF 标志和/或 RTIE 位被取消。 1 等待 RTI 中断；即设置 RTIF 标志和 RTIE 位。

第 6 章

并行输入/输出控制

本章解释了与并行输入/输出 (I/O) 和引脚控制相关的软件控制。看见第 2 章, “引脚和连接”, 了解有关这些引脚分配和外部硬件注意事项的更多信息。

所有这些 I/O 引脚都与片上外围设备功能共享 (见表 2-1)。外围模块优先于 I/O。当外围设备启用时, 与共享引脚相关的 I/O 功能将被禁用。重置后, 共享的外设功能将被禁用, 因此 I/O 控制引脚。所有的 I/O 都是配置为输入 (PTADDn = 0), 下拉/下拉设备 (PTAPEn = 0), 但仅输出引脚 PTA4 除外, 该引脚默认为 BKGD/MS 功能。重置后选择所有引脚的默认低驱动器强度 (PTxDSn = 0)

通过端口数据寄存器读取和写入并行 I/O。方向, 无论是输入还是输出, 都通过端口数据方向寄存器控制。方框图图 6-1 说明了单个引脚的并行 I/O 端口功能。

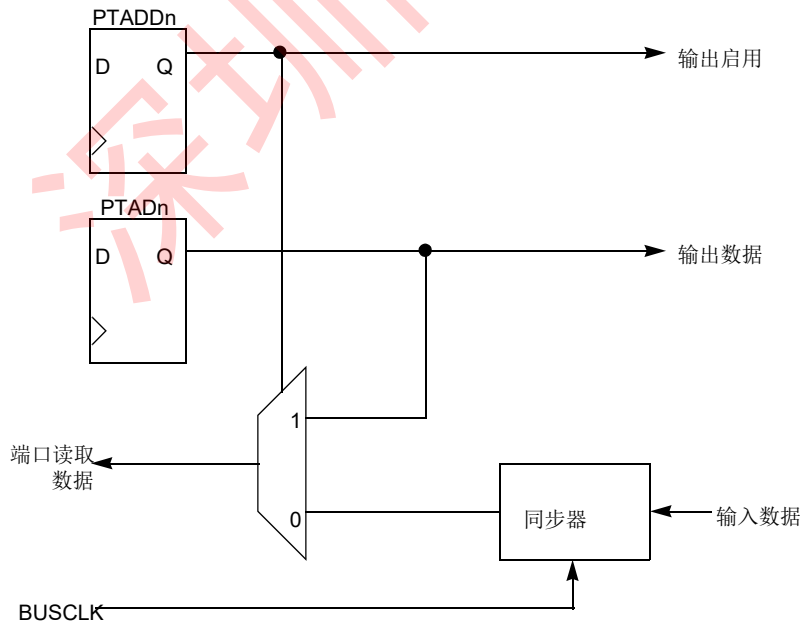


图 6-1. 并行 I/O 方框图

数据方向控制位（PTADD_n）确定关联引脚的输出缓冲区是否启用，并控制端口数据寄存器读取的源。关联引脚的输入缓冲区始终处于启用状态，除非该引脚被启用是模拟功能或仅输出的引脚。

当为引脚启用共享数字功能时，共享功能控制输出缓冲区。
然而，数据方向寄存器位继续控制端口数据寄存器的读取源。

深圳市南天星

当为引脚启用共享模拟功能时，输入和输出缓冲区将被禁用。对于位为输入（PTADDn = 0）且输入缓冲区被禁用的任何端口数据位，都会读取 0 的值。一般来说，每当与...共享一个针脚时替代数字功能和模拟功能，模拟功能具有优先权，如果启用数字和模拟功能，模拟功能将控制引脚。

在改变端口引脚的方向成为输出之前，写入端口数据寄存器很有用。这确保了引脚不会用恰好在端口数据寄存器中的旧数据值临时驱动。

与并行 I/O 端口关联的一组寄存器位于高页寄存器空间中，该寄存器独立于并行 I/O 寄存器运行。这些寄存器用于控制引脚的上拉/下拉和摆动率。看见第 6.3 节“引脚控制寄存器。”

6.1 低功耗模式下的引脚行为

在等待和停止模式下，所有引脚状态都保持不变，因为内部逻辑保持通电。恢复后，所有引脚功能都会恢复到进入停止模式之前的状态。

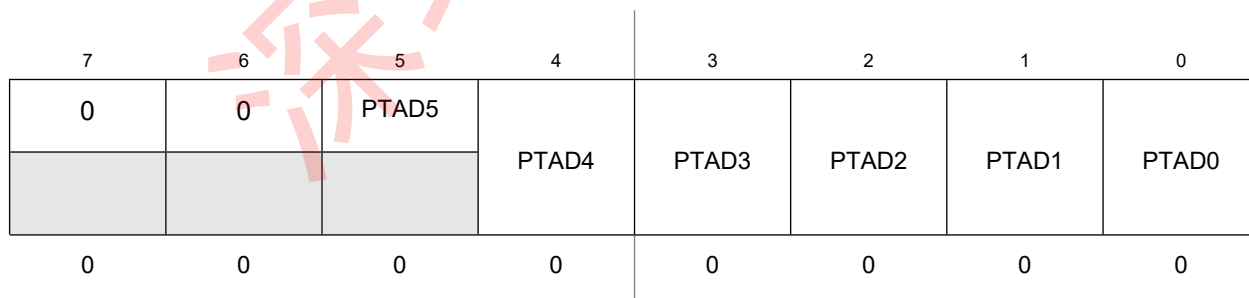
6.2 并行 I/O 寄存器

本节提供与并行 I/O 端口相关的寄存器的信息。并行 I/O 数据寄存器位于内存映射的\$001F 内存边界内，因此可以使用简短而直接的寻址模式指令。

请参阅表格第 4 章“Memory”，用于所有并行 I/O 的绝对地址分配。本节仅通过其名称来指寄存器和控制位。Freescale Semiconductor 提供的等式或头文件通常用于将这些名称转换为适当的绝对地址。

6.2.1 端口 A 寄存器

端口 并行 I/O 功能由本节中描述的数据和数据方向寄存器控制。



字母 R

罗马字母的第 23 个字母

重置:

图 6-2. Port A 数据寄存器 (PTAD)

表 6-1. PTAD 注册字段描述

R

W

7	6	5	4	3	2	1	0
0	0	0	0	PTADD3	PTADD2	PTADD1	PTADD0
0	0	0	0	0	0	0	0

学科	描述
5:0 PTAD[5:0]	端口 A 数据寄存器位 —对于作为输入的端口 A 引脚，读取返回引脚上的逻辑电平。对于配置为输出的端口 A 引脚，读取返回写入此寄存器的最后值。写入到这个寄存器的所有位中。对于配置为输出的端口 A 引脚，逻辑级别在相应的 MCU 引脚上驱动。 重置强制 PTAD 到所有 0，但这些 0 不会在相应的引脚上推出，因为重置还会将所有端口引脚配置为禁用上拉/下拉的高阻抗输入。

字母 R

罗马字母的第 23 个字母

重置:

图 6-3。端口 A 数据方向寄存器 (PTADD)

表 6-2。PTADD 注册字段描述

学科	描述
3:0 PTADD[3:0]	A 端口位的数据方向 —这些读/写位控制端口 A 引脚的方向和 PTAD 读取的内容。 0 输入（禁用输出驱动程序）和读取返回引脚值。 1 为端口 A 位 n 启用的输出驱动程序，PTAD 读取返回 PTADn 的内容。

6.2.2 B 端口登记册

B 端口并行 I/O 功能由本节中描述的数据和数据方向寄存器控制。

7	6	5	4	3	2	1	0
PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0	0	0	0	0	0	0	0

字母 R

罗马字母的第 23 个字母

重置:

图 6-4. B 端口数据寄存器 (PTBD)

表 6-3. PTBD 注册字段描述

7	6	5	4	3	2	1	0
PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0	0	0	0	0	0	0	0

学科	描述
7:0 PTBD[7:0]	<p>B 端口数据寄存器位—对于作为输入的端口 B 引脚，读取返回引脚上的逻辑电平。对于配置为输出的端口 B 引脚，读取返回写入此寄存器的最后值。写入到这个寄存器的所有位中。对于配置为输出的端口 B 引脚，逻辑级别在相应的 MCU 引脚上驱动。</p> <p>重置强制 PTBD 到所有 0，但这些 0 不会在相应的引脚上推出，因为重置还会将所有端口引脚配置为禁用上拉/下拉的高阻抗输入。</p>

重置:

图 6-5. B 端口数据方向寄存器 (PTBDD)

表 6-4. PTBDD 寄存器字段描述

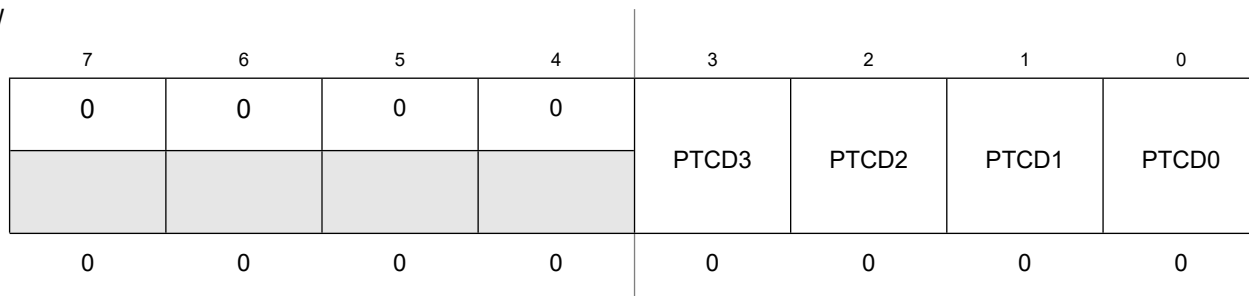
学科	描述
7:0 PTBDD[7:0]	<p>B 端口位的数据方向—这些读/写位控制端口 B 引脚的方向和 PTBD 读取的内容。</p> <p>0 输入（禁用输出驱动程序）和读取返回引脚值。</p> <p>1 为端口 B 位 n 启用的输出驱动程序和 PTBD 读取返回 PTBDn 的内容。</p>

6.2.3 C 端口寄存器

C 端口并行 I/O 功能由本节中描述的数据和数据方向寄存器控制。

R

W



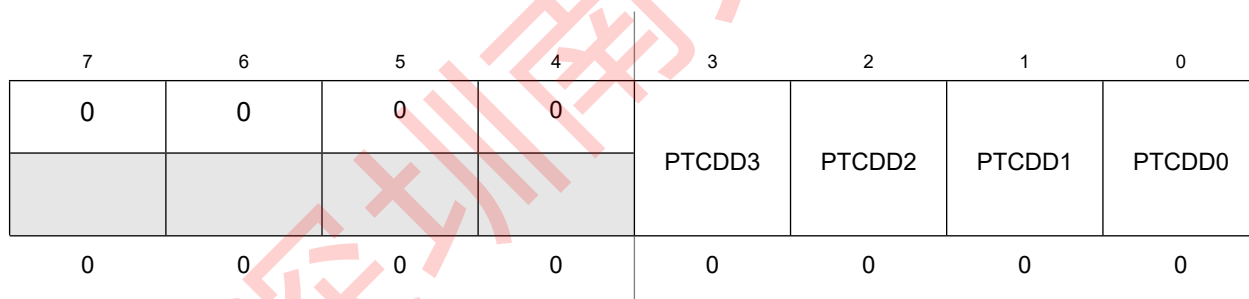
字母 R

罗马字母的第 23 个字母

重置:

图 6-6. C 端口数据寄存器 (PTCD)

表 6-5. PTCD 注册字段描述



学科	描述
3:0 PTCD[3:0]	C 端口数据寄存器位 —对于作为输入的端口 C 引脚，读取返回引脚上的逻辑级别。对于配置为输出的端口 C 引脚，读取返回写入此寄存器的最后值。写入到这个寄存器的所有位中。对于配置为输出的端口 C 引脚，逻辑级别在相应的 MCU 引脚上驱动。重置强制 PTCD 到所有 0，但这些 0 不会在相应的引脚上被驱除，因为重置也将所有端口引脚配置为禁用上拉/下拉的高阻抗输入。

字母 R

罗马字母的第 23 个字母

重置:

图 6-7. C 端口数据方向寄存器 (PTCDD)

表 6-6. PTCDD 寄存器字段描述

学科	描述
3:0 PTCDD[3:0]	C 端口位的数据方向 —这些读/写位控制端口 C 引脚的方向以及 PTCDD 读取的内容。 0 输入（禁用输出驱动程序）和读取返回引脚值。 1 为端口 C 位 n 启用的输出驱动程序和 PTCDD 读取返回 PTCDDn 的内容。

6.3 引脚控制寄存器

本节提供与用于引脚控制功能的并行 I/O 端口相关的寄存器的信息。

请参阅表格第 4 章“Memory”，用于引脚控制寄存器的绝对地址分配。本节仅通过名称提及寄存器和控制位。Freescale Semiconductor 提供的等式或标头文件通常用于将这些名称转换为适当的绝对地址。

笔记

通过在驱动器强度选择寄存器（PTxDSn）中设置相应的位，可以选择输出引脚来具有高输出驱动强度。当选择高驱动器时，引脚能够采购和沉入更大的电流。即使每个 I/O p 可以选择为高驱动，不超过 MCU 的总电流源和接收器限制。驱动强度选择会影响 I/O 引脚的直流行为。然而，AC 行为也会受到影响。高驱动允许引脚驱动更大的负载 w 它与低驱动器相同的开关速度，使引脚进入较小的负载。正因为如此，EMC 排放可能会受到使引脚成为高驱动的影响。

6.3.1 端口 A 引脚控制寄存器

与端口 A 相关的引脚由本节提供的寄存器控制。这些寄存器控制端口 A 引脚的引脚上拉/下拉和转速率，独立于并行 I/O 寄存器。

6.3.1.1 启用内部拉动装置

通过在
拉拔设
备启用
寄存器

7	6	5	4	3	2	1	0
0	0	PTAPE5	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0

（PTAPE_n）中设置相应的位，可以为每个端口引脚启用内部拉拔设备。如果引脚被并行 I/O 控制逻辑或任何共享配置为输出，则拉动装置将被禁用 D 外围输出功能，无论相应的拉拔设备启用寄存器位的状态如何。如果模拟功能控制销，拉动装置也会被禁用。

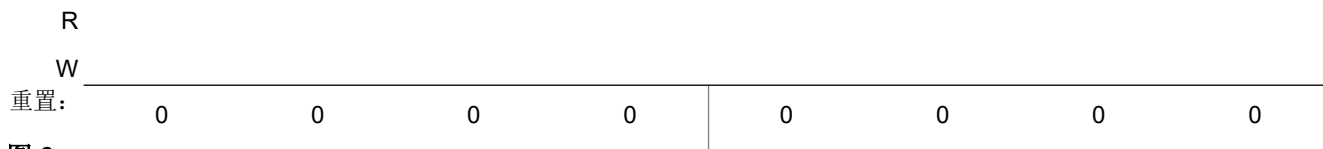


图 6-

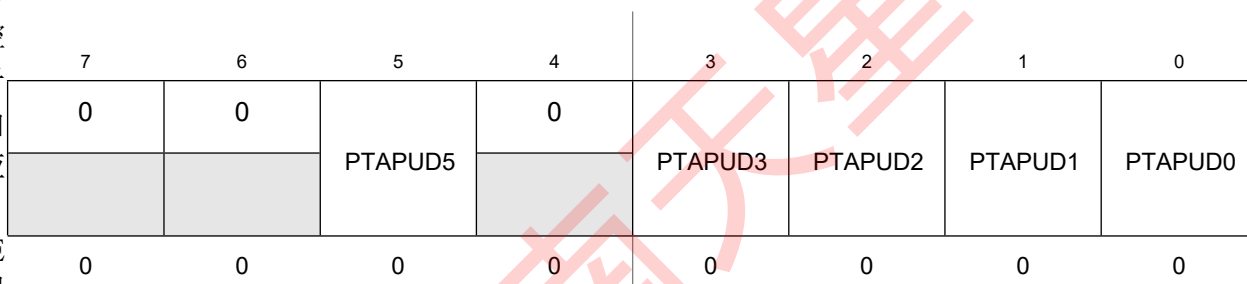
8. 内拉装置启用端口 A 寄存器 (PTAPE)

表 6-7. PTAPE 注册字段描述

学科	描述
5,3:0 PTAPE[5,3:0]	<p>内置拉拔设备启用端口 A 位—每个控制位决定了内部拉动装置是否为相关的 PTA 引脚启用。对于配置为输出的端口 A 引脚，这些位不起作用，内部上拉装置被禁用。</p> <p>0 端口 A 位 n 禁用内部拉动装置。</p> <p>1 为端口 A 位 n 启用了内部拉拔装置。</p>

6.3.1.2 拉起/下拉控制

拉起 / 下拉控制用于选择由相应的 PTAPE 位启用的拉起或下拉装置。



字母 R

罗马字母的第 23 个字母

重置:

图 6-9. A 端口的上拉/下拉设备控制 (PTAPUD)

表 6-8. PTAPUD 注册字段说明

学科	描述
5,3:0 PTAPUD[5,3:0]	<p>A 端口位的上拉/下拉装置控制—每个控制位都决定了是否为关联的 PTA 引脚选择内部上拉或下拉装置。只有启用关联的 PTAPE 位，才能启用实际的上拉/下拉设备。0 为端口 A 位 n 选择了内部上拉装置。</p> <p>1 为端口 A 位 n 选择了内部下拉装置。</p>

6.3.1.3 输出下降速率控制启用

通过在转速率控制寄存器（PTASEn）中设置相应的位，可以为每个端口引脚启用转速率控制。启用后，减排控制限制输出传输速率，以减少 EMC 排放。降低利率控制没有对配置为输入的引脚的影响。

深圳南天星

		5	4	3	2	1	0	
R	0	0	0	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
W 重置:								
	0	0	0	1	1	1	1	1

图 6-10. 启用端口

A 寄存器 (PTASE) 的降低率

表 6-9. PTASE 注册字段描述

学科	描述
4:0 PTASE[4:0]	为端口 A 位启用输出 Slew 速率—每个控制位决定了是否为关联的 PTA 引脚启用了输出 slew 速率控制。对于配置为输入的端口 A 引脚，这些位不起作用。 0 端口 A 位 n 禁用输出转速率控制。 1 为端口 A 位 n 启用输出 slew 速率控制。

6.3.1.4 端口 A 驱动器强度选择寄存器 (PTADS)

	7	6	5	4	3	2	1	0
	0	0	0	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
	0	0	0	0	0	0	0	0

字母 R

罗马字母的第 23 个字母

调整

图 6-11. 端口 A (PTASE) 的输出驱动器强度选择

表 6-10. PTASE 注册字段描述

学科	描述
4:0 PTADS[4:0]	A 端口位的输出驱动器强度选择—每个控制位都为相关的 PTA 引脚在低输出和高输出驱动器之间进行选择。 0 为端口 A 位 n 启用的低输出驱动器。 1 为端口 A 位 n 启用的高输出驱动器。

6.3.2 B 端口引脚控制寄存器

与端口 B 关联的引脚由本节提供的寄存器控制。这些寄存器控制端口 B 引脚的引脚上拉/下拉和下降速率，独立于并行 I/O 寄存器。

R

W

6.3.2.1 启用内部拉动装置

通过在拉拔装置启用寄存器

7	6	5	4	3	2	1	0
PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0	0	0	0	0	0	0	0

(PTBPE_n) 中设置相应的位，可以为每个端口引脚启用内部拉拔装置。如果引脚被并行 I/O 控制逻辑或任何共享配置为输出，则拉动装置将被禁用 D 外围输出功能，无论相应拉拔装置的状态如何，启用寄存器位。如果模拟功能控制销，拉动装置也会被禁用。

重置:

图 6-12. 启用 B 端口寄存器 (PTBPE) 的内部拉拔装置

表 6-11. PTBPE 注册字段描述

学科	描述
7:0 PTBPE[7:0]	<p>内置拉拔设备启用端口 A 位—这些控制位中的每一个都决定了内部拉动装置是否为关联的 PTB 引脚启用。对于配置为输出的端口 B 引脚，这些位没有影响，内部上拉装置被禁用。</p> <p>0 端口 B 位 n 禁用内部拉动装置。</p> <p>1 为端口 B 位 n 启用的内部拉拔装置。</p>

6.3.2.2 拉起/下拉控制

拉起 / 下拉控制用于选择由相应的 PTBPE 位启用的上拉或下拉装置。

7	6	5	4	3	2	1	0
PTBPUD7	PTBPUD6	PTBPUD5	PTBPUD4	PTBPUD3	PTBPUD2	PTBPUD1	PTBPUD0
0	0	0	0	0	0	0	0

字母 R

罗马字母的第 23 个字母

重置:

R

W

图 6-13。B 端口的上拉/下拉装置控制 (PTBPUD)

表 6-12。PTBPUD 注册字段描述

学科	描述
7:0 PTBPUD[7:0]	B 端口位的上拉/下拉装置控制 —每个控制位决定了是否为关联的 PTB 引脚选择内部上拉或下拉装置。只有启用相关的 PTBPE 位, 才能启用实际的上拉/下拉设备。 0 为端口 B 位 n 选择了内部上拉装置。 1 为端口 B 位 n 选择了内部下拉装置。

6.3.2.3 输出下降速率控制启用

通过在
速率
控制寄
存器

7	6	5	4	3	2	1	0
PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
1	1	1	1	1	1	1	1

(PTBSEn) 中设置相应的位, 可以为每个端口引脚启用速率控制。启用后, 减排控制限制输出传输速率, 以减少 EMC 排放。降低速率控制没有对配置为输入的引脚的影响。

重置:

图 6-14。启用 B 端口寄存器 (PTBSE) 的速率

表 6-13。PTBSE 注册字段描述

学科	描述
7:0 PTBSE[7:0]	为端口 B 位启用输出减去率 —这些控制位中的每一个都决定了是否为关联的 PTB 引脚启用了输出 slew 速率控制。对于配置为输入的端口 B 引脚, 这些位没有影响。 0 对端口 B 位 n 禁用输出速率控制。 1 为端口 B 位 n 启用了输出速率控制。

6.3.2.4 B 端口驱动强度选择寄存器 (PTBDS)

7	6	5	4	3	2	1	0
PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
0	0	0	0	0	0	0	0

字母 R

R

W

罗马字母的第 23 个字母

调整

图 6-15. B 端口 (PTBDS) 的输出驱动器强度选择

表 6-14. PTBDS 注册字段描述

学科	描述
7:0 PTBDS[7:0]	B 端口位的输出驱动强度选择 —每个控制位都为关联的 PTB 引脚在低输出和高输出驱动器之间进行选择。 0 为端口 B 位 n 启用的低输出驱动器。 1 为端口 B 位 n 启用的高输出驱动器。

6.3.3 C 端口引脚控制寄存器

与端口 C 关联的引脚由本节提供的寄存器控制。这些寄存器控制端口 C 引脚的引脚上拉/下拉和下降速率，独立于并行 I/O 寄存器。

6.3.3.1 启用内部拉动装置

通过在
拉动设
备启用
寄存器

7	6	5	4	3	2	1	0
0	0	0	0	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0	0	0	0	0	0	0	0

(PTCPE_n) 中设置相应的位，可以为每个端口引脚启用内部拉动装置。如果引脚被并行 I/O 控制逻辑或任何共享配置为输出，则拉动装置将被禁用 D 外围输出功能，无论相应拉拔装置的状态如何，启用寄存器位。如果模拟功能控制销，拉动装置也会被禁用。

重置:

图 6-16. 启用端口 C 寄存器 (PTCPE) 的内部拉拔装置

表 6-15. PTCPE 寄存器字段描述

学科	描述
----	----

R

W

3:0 PTCPE[3:0]	<p>内置拉拔装置启用端口 C 位—这些控制位中的每一个都决定了内部拉拔设备是否为关联的 PTC 引脚启用。对于配置为输出的端口 C 引脚，这些位没有影响，内部上拉装置被禁用。</p> <p>0 端口 C 位 n 禁用内部拉动装置。</p> <p>1 为端口 C 位 n 启用了内部拉动装置。</p>
-------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

6.3.3.2 拉起/下拉控制

下拉 / 下拉控制用于选择由相应的 PTCPE 位启用的上拉或下拉设备。

	7	6	5	4		3	2	1	0
选择由相应的 PTCPE 位启用的上拉或下拉设备。	0	0	0	0		PTCPUD3	PTCPUD2	PTCPUD1	PTCPUD0
	0	0	0	0		0	0	0	0

字母 R

罗马字母的第 23 个字母

重置:

图 6-17. C 端口 (PTCPUD) 的上拉/下拉设备控制

表 6-16. PTCPUD 注册字段描述

学科	描述
3:0 PTCPUD[3:0]	<p>C 端口位的上拉/下拉装置控制—这些控制位中的每一个都决定了是否为相关的 PTC 引脚选择内部上拉或下拉装置。实际的上拉/下拉设备仅通过启用相关的 PTCPE 位才能启用。</p> <p>0 为端口 C 位 n 选择内部上拉装置。</p> <p>1 为端口 C 位 n 选择了内部下拉装置。</p>

6.3.3.3 输出下降速率控制启用

通过在摆速率控制寄存器 (PTCSEn) 中设置相应的位，可以为每个端口引脚启用摆速率控制。启用后，减排控制限制输出传输速率，以减少 EMC 排放。降低利率控制没有对配置为输入的引脚的影响。

	7	6	5	4	3	2	1	0
字母 R	0	0	0	0	PTCSE3	PTCSE2	PTCSE1	PTCSE0
罗马字母的第 23 个字母	0	0	0	0	1	1	1	1

重置:

图 6-18. 为端口 C 寄存器 (PTCSE) 启用掉头速率

表 6-17. PTCSE 寄存器字段描述

字段	描述
3:0 PTCSE[3:0]	<p>为端口 C 位启用输出流速率—每个控制位都确定是否为相关的 PTC 引脚启用了输出平速率控制。对于配置为输入的端口 C 引脚，这些位没有效果。</p> <p>0 端口 C 位 n 的输出流速率控制被禁用。</p> <p>1 为端口 C 位 n 启用了输出流速率控制。</p>

6.3.3.4 端口 C 驱动强度选择寄存器 (PTCDS)

	7	6	5	4	3	2	1	0
字母 R	0	0	0	0	PTCDS3	PTCDS2	PTCDS1	PTCDS0
罗马字母的第 23 个字母	0	0	0	0	0	0	0	0

调整

图 6-19. 端口 C (PTCDS) 的输出驱动强度选择

表 6-18. PTCDS 寄存器字段描述

字段	描述
3:0 PTCDS[3:0]	<p>端口 C 位的输出驱动强度选择—每个控制位都是在关联的 PTC 引脚的低输出和高输出驱动器之间选择的。</p> <p>0 为端口 C 位 n 启用低输出驱动器。</p> <p>1 为端口 C 位 n 启用高输出驱动器。</p>

第 6 章 并行输入/输出控制

第 7 章

键盘中断（RS08KBIV1）

7.1 简单介绍

键盘中断（KBI）模块提供独立启用的外部中断源。

7.1.1 特点

KBI 的功能包括：

- 每个键盘中断引脚都有一个单独的引脚启用位
- 每个键盘中断引脚只能编程为下降边缘（或上升边缘），或同时下降边缘和低水平（或上升边缘和高水平）中断灵敏度
- 一个支持软件的键盘中断
- 退出低功耗模式

7.1.2 操作模式

本节定义了等待、停止和后台调试模式下的 KBI 操作。

7.1.2.1 在等待模式下操作

如果在执行 WAIT 指令之前启用，KBI 将继续在等待模式下运行。因此，如果启用了 KBI 中断（KBIE = 1），则可以使用已启用的 KBI 引脚（KBPE_n = 1）使 MCU 退出等待模式。

7.1.2.2 在停止模式下操作

如果在执行 STOP 指令之前启用，KBI 将在停止模式下异步运行。因此，如果启用了 KBI 中断（KBIE = 1），则可以使用已启用的 KBI 引脚（KBPE_n = 1）使 MCU 退出停止模式。

7.1.2.3 在活动后台模式下操作

当微控制器处于活动后台模式时，KBI 继续正常运行。

7.1.3 方框图

图 7-1 显示键盘中断模块的框图。

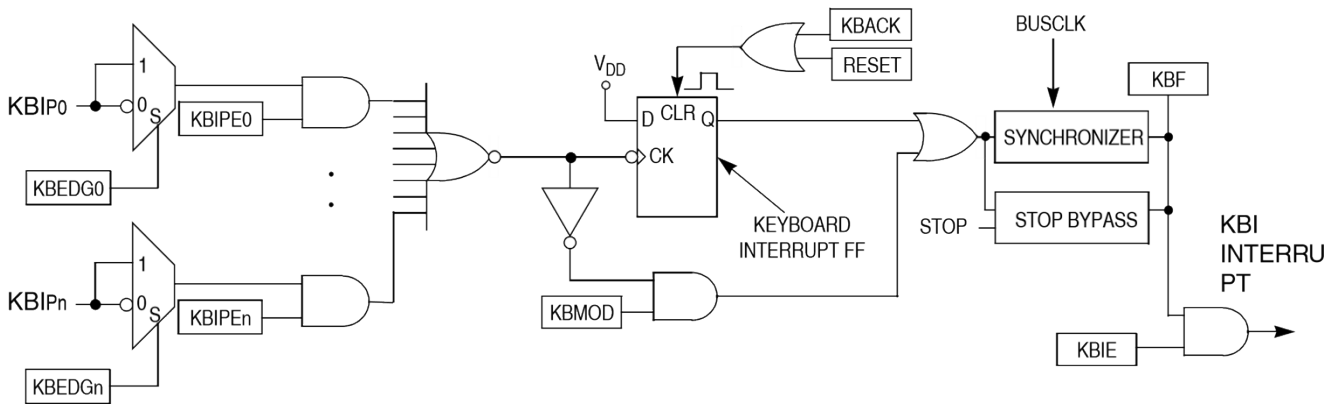


图 7-1. 键盘中断 (KBI) 块图

7.2 外部信号描述

KBI 输入引脚可用于检测下降边缘，或同时检测下降边缘和低电平中断请求。KBI 输入引脚还可用于检测上升边缘，或同时检测上升边缘和高级中断请求。

表 7-1 显示 KBI 信号属性。

表 7-1. 信号属性

信号	函数	I/O
KBIPn	键盘中断引脚	我

7.3 注册定义

KBI 包括三个寄存器：

- 8 位引脚状态和控制寄存器
- 8 位引脚启用寄存器
- 8 位边缘选择寄存器

请参阅直接页面注册摘要第 4 章，“记忆，”用于所有 KBI 寄存器的绝对地址分配。本节仅通过其名称来指寄存器和控制位。

KBI 寄存器汇总在表 7-2。

表 7-2. KBI 注册摘要

名字		7	6	5	4	3	2	1	0
KBISC	字母 R	0	0	0	0	KBF	0	KBIE	KBMOD
	罗马字						KBACK		

	母的第23个字母								
KBIPE	字母 R	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
	罗马字母的第23个字母								

第7章 键盘中断 (RS08KBIV1)

表 7-2. KBI 注册摘要 (续)

名字		7	6	5	4	3	2	1	0
KBIES	字母 R	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBEDG3	KBEDG2	KBEDG1	KBEDG0
	罗马字母的第23个字母								
PIN 名称	我	KBIP7	KBIP6	KBIP5	KBIP4	KBIP3	KBIP2	KBIP1	KBIP0

7.3.1 KBI 状态和控制登记册 (KBISC)

KBISC 包含用于配置 KBI 的状态标志和控制位。

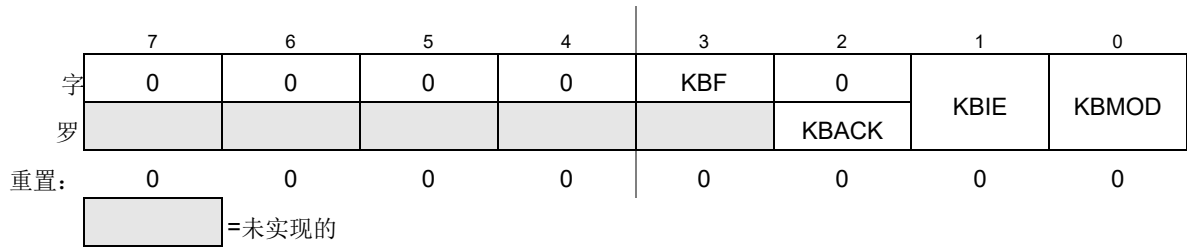


图 7-2。KBI 状态和控制寄存器 (KBISC)

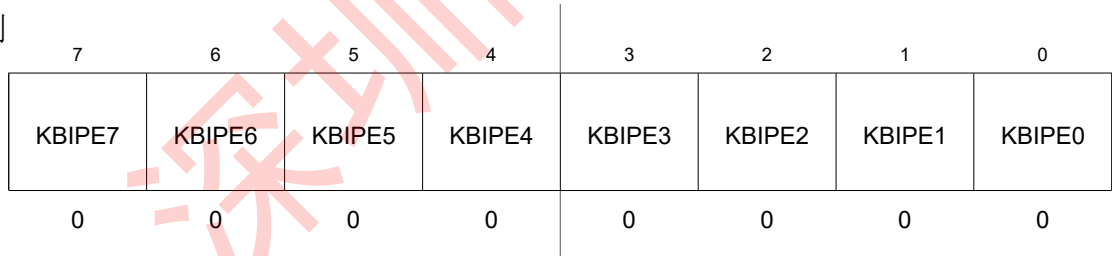
表 7-3。KBISC 寄存器字段描述

字段	描述
3 KBF	键盘中断标志 — KBF 表示检测到键盘中断。写入对 KBF 没有影响。0 未检测到键盘中断。 1 检测到键盘中断。
2 KBACK	键盘确认 — 将 1 写入 KBACK 是清除旗帜机制的一部分。KBACK 总是读作 0。
1 KBIE	键盘中断启用 — KBIE 启用键盘中断请求。 0 键盘中断请求未启用。 1 启用了键盘中断请求。
0 KBMOD	键盘检测模式 — KBMOD (以及 KBEDG 位) 控制键盘中断引脚的检测模式。 0 键盘仅检测边缘。 1 键盘可以检测边缘和水平。

7.3.2 KBI 引脚启用寄存器 (KBIPE)

KBIPE 包含脚启用控制

引位。



字母 R

罗马字母的第 23 个字母

重置:

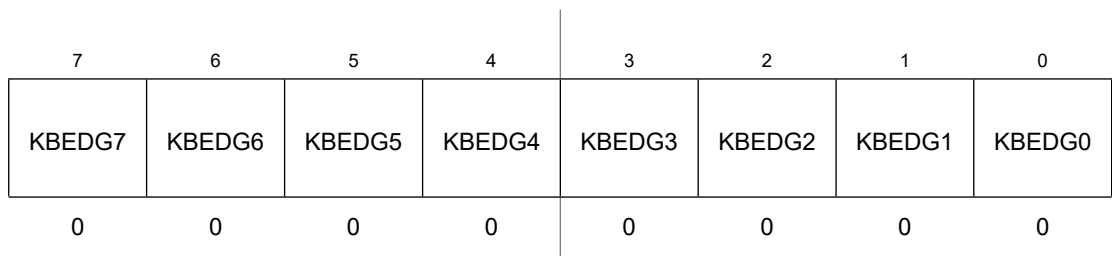
图 7-3。KBI 引脚启用寄存器 (KBIPE)

表 7-4。KBIPE 寄存器字段描述

字段	描述
7:0 KBIPEn	键盘引脚启用 — 每个 KBIPE _n 位都启用了相应的键盘中断引脚。0 相应引脚未作为键盘中断启用。 1 个相应的引脚启用为键盘中断。

7.3.3 KBI 边缘选择寄存器 (KBIES)

KBIES 包含边缘选择控制位。



字母 R

罗马字母的第 23 个字母

重置:

图 7-4. KBI 边缘选择寄存器 (KBIES)

表 7-5. KBIES 注册字段描述

字段	描述
7:0 KBEDGn	键盘边缘选择 —每个 KBEDGn 位选择相应引脚的下降边缘/低水平或上升边缘/高水平功能。 0 下降边缘/低水平。 1 上升边缘/高水平。

7.4 功能描述

这个片上外围模块被称为键盘中断 (KBI) 模块，因为它最初旨在简化键盘交换机的行列矩阵的连接和使用。然而，这些输入作为额外的外部中断也很有用输入和作为从停止或等待低功耗模式中唤醒 MCU 的外部手段。

KBI 模块允许其引脚作为额外的中断源。写入键盘中断引脚启用寄存器 (KBIPEn) 中的 KBIPEn 位会独立启用或禁用每个 KBI 引脚。每个 KBI 引脚可以配置为边缘敏感或边缘-和级别敏感基于键盘中断状态和控制寄存器 (KBISC) 中的 KBMOD 位。边缘敏感可以是软件，可以编程为下降或上升；水平可以是低或高。边缘或边缘和水平敏感的极性 *ity* 是使用键盘中断边缘选择寄存器 (KBIES) 中的 KBEDGn 位选择的。

7.4.1 仅边缘灵敏度

同步逻辑用于检测边缘。当启用的键盘中断 (KBIPEn=1) 输入信号在一个总线周期中被视为逻辑 1 (去断言水平)，然后在下一个周期中被视为逻辑 0 (断言水平) 时，检测到下降边缘。当输入信号在一个总线周期内被视为逻辑 0 (断言电平)，然后在下一个周期中被视为逻辑 1 (断言电平) 时，会检测到上升边缘。在检测到第一个边缘之前，所有启用的键盘都会中断输入信号必须在

断言的逻辑水平。检测到任何边缘后，所有启用的键盘中断输入信号必须返回到去断言的水平，然后才能检测到任何新的边缘。

已启用的 KBI 引脚上的有效边缘将在 KBISC 中设置 KBF。如果在 KBISC 中设置了 KBIE，将向 CPU 显示中断请求。清除 KBF 是通过在 KBISC 中写一个 1 到 KBACK 来完成的。

7.4.2 边缘和水平灵敏度

已启用的 KBI 引脚上的有效边缘或级别将在 KBISC 中设置 KBF。如果在 KBISC 中设置了 KBIE，将向 CPU 显示中断请求。清除 KBF 是通过在 KBISC 中向 KBACK 写入 1 来完成的，前提是所有启用的键盘输入都处于去断言的水平。如果在试图通过向 KBACK 写入 1 来清除时断言任何已启用的 KBI 引脚，KBF 将保持设置。

7.4.3 KBI 上拉/下拉下电阻器

KBI 引脚可以配置为使用内部上拉/下拉电阻，使用关联的 I/O 端口上拉启动寄存器。如果启用了内部电阻，则使用 KBIES 寄存器来选择电阻是上拉（KBEDGn = 0）还是下拉（KBEDGn = 1）。

7.4.4 KBI 初始化

首次启用键盘中断引脚时，可能会获得虚假的键盘中断标志。为了防止在键盘初始化期间出现错误的中断请求，用户必须执行以下操作：

1. 掩码键盘通过在 KBISC 中清除 KBIE 来中断。
2. 通过在 KBIES 中设置适当的 KBEDGn 位来启用 KBI 极性。
3. 如果使用内部上拉/下拉设备，请在 PTxPE 中配置相关的上拉启用位。
4. 通过在 KBIPE 中设置适当的 KBIPEn 位来启用 KBI 引脚。
5. 在 KBISC 中写信给 KBACK，以清除任何虚假中断。
6. 在 KBISC 中设置 KBIE 以启用中断。

第 8 章

中央处理器单元 (RS08CPUV1)

8.1 简单介绍

本章总结了有关 RS08 系列 CPU 的寄存器、寻址模式和指令集的信息。有关更详细的讨论，请参阅 RS08 核心参考手册，第 1 卷，Freescale Semiconductor 文档订单号 RS08RMv1。

RS08 CPU 的开发旨在使用独立于过程的设计方法针对成本极低的嵌入式应用程序，使其能够跟上硅处理技术的快速发展。

RS08 核心的主要特点是：

- 简化的程序员模型
- 带有次要指令扩展的 HCS08 指令集子集
- 成本敏感型嵌入式应用程序的最小指令集
- 影子程序计数器操作、SHA 和 SLA 的新说明
- 用于代码大小优化的新短和微小寻址模式
- 16K 字节可访问的内存空间
- 重置将从 3FFD 中获取第一个指令
- 通过执行 STOP 和 WAIT 指令来支持低功耗模式
- 使用后台调试控制器模块进行调试和 FLASH 编程支持
- 带有重置的非法地址和操作码检测

8.2 程序员模型和 CPU 寄存器

图 8-1 显示 RS08 CPU 的程序员模型。这些寄存器不位于微控制器的内存图中。它们直接构建在 CPU 逻辑中。

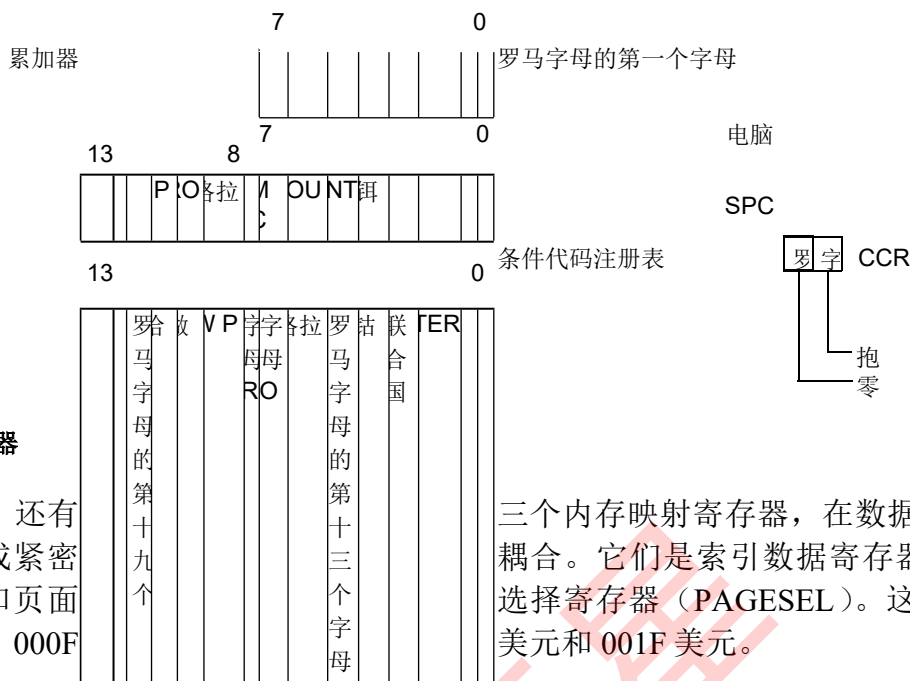


图 8-1。CPU 寄存器

除了 CPU 寄存器外，还有期间与核心地址生成紧密索引寄存器 (X) 和页面分别位于 000E 美元、000F

三个内存映射寄存器，在数据读写操作耦合。它们是索引数据寄存器 (D[X])、选择寄存器 (PAGESEL)。这些寄存器美元和 001F 美元。

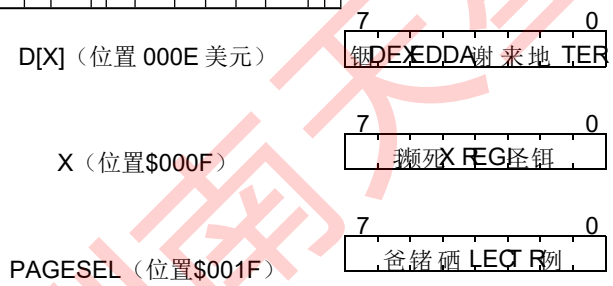


图 8-2。内存映射寄存器

8.2.1 累加器 (A)

这个通用的 8 位寄存器是 RS08 MCU 的主要数据寄存器。数据可以通过负载累加器 (LDA) 指令从内存读取到 A 中。A 中的数据可以通过存储累加器 (STA) 指令写入内存。各种添加器 Essing 模式变化允许在指定加载或存储指令中涉及的内存位置方面具有很大的灵活性。交换指令允许在 A 和 SPC 高 (SHA) 之间以及 A 和 SPC 低 (SLA) 之间交换值。

算术、移位和逻辑操作可以对 A 中的值执行，如 ADD、SUB、RORA，印加、德卡和、奥拉、EOR 等。在其中一些指令中，如 INCA 和 LSLA，A 中的值是唯一的输入操作数，结果取代了 A 中的值。在其他情况下，如 ADD 和 AND，有两个操作数：A 中的值和内存中的第二个值。算术或逻辑运算的结果取代了 A 中的值。

一些指令，如内存到内存移动指令 (MOV)，不使用累加器。DBNZ 还解除了 A，因为它允许在内存变量而不是累加器中实现循环计数器。

在重置期间，累加器加载了 00 美元。

8.2.2 程序计数器 (PC)

程序计数器是一个 14 位寄存器，包含要获取的下一个指令或操作数的地址。

在正常执行期间，每次获取指令或操作数时，程序计数器会自动增加到下一个顺序内存位置。跳转、分支和返回操作下一个顺序位置以外的地址加载程序计数器。这被称为改变流程。

在重置期间，程序计数器加载了 3FFD 美元，程序将从这个特定位置开始执行。

8.2.3 影子程序计数器 (SPC)

影子程序计数器是一个 14 位寄存器。在使用 JSR 或 BSR 指令的子程序调用期间，返回地址将保存到 SPC 中。子程序完成后，RTS 指令将恢复程序的内容来自影子程序计数器的计数器。

在重置期间，阴影程序计数器加载了 3 美元 FFD。

8.2.4 条件代码寄存器 (CCR)

2 位条件代码寄存器包含两个状态标志。RS08 中的 CCR 内容不可直接读取。CCR 位可以使用 BCC 和 BEQ 等条件分支指令进行测试。这两个寄存器位可以通过 BDC 接口直接访问。以下段落提供了有关 CCR 位及其使用方式的详细信息。图 8-3 识别 CCR 位及其位位置。



图 8-3. 条件代码寄存器 (CCR)

重置后，状态位 (Z 和 C) 被清除为 0。

这两个状态位表示算术和其他指令的结果。条件分支指令要么分支到新的程序位置，要么允许程序继续执行分支之后的下一个指令，具体取决于 CCR 状态位中的值。条件分支指令，如 BCC、BCS 和 BNE，根据单个 CCR 位的状态导致分支。

通常，条件分支立即遵循导致 CCR 位更新的指令，如下顺序：

厘米 #5 ;比较蓄能器 A 到 5 blo 较低的 ;分支，如果 A 小 5 更多：
deca;如果 A 不高于或相同于 5，请这样做：

测试和条件分支之间可以执行其他指令，只要使用的唯一指令是那些不干扰影响条件分支的 CCR 位的指令。例如，测试是在子程序或功能中执行的在子程序返回主程序之前，条件分支不会被执行。这是一种参数传递形式（即信息以条件代码位返回给调用程序）。

Z — 零旗

Z 位设置为表示操作结果为 00 美元。

分支如果相等（BEQ）和分支如果不相等（BNE）是简单的分支，仅基于 Z 位中的值进行分支。所有加载、存储、移动、算术、逻辑、移位和旋转指令都会导致 Z 位更新。

C — 携带

在加法操作后，如果源操作数都大于或等于 80 美元，或者如果其中一个操作数大于或等于 80 美元，并且结果小于 80 美元，则设置 C 位。这相当于无符号溢出。减法或比较从 CPU 寄存器的内容中执行内存操作数的减法，因此在减法操作后，如果内存操作数的无符号值大于 CPU 寄存器的无符号值，则设置 C 位。这是等价的 T 到未签名的借款或溢出。

分支如果携带清除（BCC）和分支如果携带集（BCS）是仅基于 C 位值的分支。C 位也被无符号分支 BLO 和 BHS 使用。添加、减去、移动和旋转指令导致 C 位更新。分支 if 位集（BRSET）和分支 if 位清除（BRCLR）指令将测试位复制到 C 位中，以促进高效的串行到并行转换算法。设置携带（SEC）和清除携带（CLC）允许设置或清除携带位直接红色。这与移位和旋转指令相结合，以及将状态信息从子例程传回 C 位的主程序的例程非常有用。

C 位包含在移位和旋转操作中，因此这些操作可以轻松扩展到多字节操作数。移位和旋转操作可以被视为 9 位移位，包括 8 位操作数或 CPU 寄存器和 CCR 的承载位。逻辑移位后，C 保存从 8 位操作数中移出的位。如果接下来使用旋转指令，则该 C 位移入操作数进行旋转，从操作数的另一端移出的位将取代 value in C，因此可以在后续旋转指令中使用。

8.2.5 索引数据寄存器（D[X]）

这个 8 位索引数据寄存器允许用户访问由 X 索引的直接页面地址空间中的数据。此寄存器驻留在内存映射位置 \$000E。有关 D[X] 寄存器的详细信息，请参阅第 8.3.8 节，“索引寻址模式（IX，由伪指令实现）。”

8.2.6 索引寄存器（X）

这个 8 位索引寄存器允许用户索引或寻址直接页面地址空间中的任何位置。

此寄存器位于内存映射位置 \$000F。有关 X 寄存器的详细信息，请参阅第 8.3.8 节，“索引寻址模式（IX，由伪指令实现）。”

8.2.7 页面选择注册（PAGESEL）

此 8 位页面选择寄存器允许用户通过位于 \$00C0 至 \$00FF 的页面窗口访问整个 16K 字节地址空间中的所有内存位置。此寄存器驻留在内存映射位置 \$001F。有关 PAGESEL 寄存器的详细信息，请参阅 RS08 核心参考手册。

8.3 寻址模式

每当 MCU 从内存中读取信息或将信息写入内存时，都会使用寻址模式来确定读取或写入信息的确切地址。本节解释了几种寻址模式以及每种模式的使用方式在不同的编程情况下。

每个操作码都告诉 CPU 以某种方式执行某种操作。许多指令，如负载累加器（LDA），允许几种不同的方法来指定要操作的内存位置，每个寻址模式变化都需要单独的操作码。所有这些变体都使用相同的指令助记符，汇编器根据操作数字段的语法和位置知道使用哪个操作码。在某些情况下，特殊字符用于指示特定的寻址模式（suCh 作为#[pound]符号，表示立即寻址模式）。在其他情况下，操作数的值告诉汇编器要使用哪种寻址模式。例如，汇编器选择短寻址模式，而不是直接寻址模式，如果操作数地址从\$0000 到\$001F。除了允许汇编器根据操作数地址选择寻址模式外，汇编器指令还可以使用“>”或“<”前缀强制直接或微小/短寻址模式 E 操作数，分别。

有些指令使用多种寻址模式。例如，移动指令使用一种寻址模式从内存访问源值，使用第二种寻址模式访问目标内存位置。对于这些移动说明，两者都文档中列出了寻址模式。所有分支指令都使用相对（REL）寻址模式来确定分支的目的地，但 BRCLR、BRSET、CBEQ 和 DBNZ 也必须访问内存操作数。这些说明是分类的通过用于内存操作数的寻址模式，并假设分支偏移的相对寻址模式。

以下段落中的讨论包括每个寻址模式的工作原理，以及指示汇编器使用特定寻址模式的语法线索。

8.3.1 固有寻址模式（INH）

当 CPU 本质上知道完成指令所需的一切，并且源代码中没有提供寻址信息时，则使用此寻址模式。通常，CPU 需要的操作数位于 CPU 的内部寄存器中，一个在 LSLA、CLRA、INCA、SLA、RTS 等中。一些固有指令，包括无操作（NOP）和背景（BGND），没有操作数。

8.3.2 相对寻址模式（REL）

相对寻址模式用于指定相对于程序计数器的分支指令的偏移地址。通常，程序员在分支指令的操作数字段中使用程序标签或表达式指定目的地；汇编器计算位置计数器（指向当时分支指令后的下一个地址）与操作数字段中标签或表达式表示的地址之间的差值。这种差异被称为 Fset 和是一个 8 位二的补码。汇编器将此偏移量存储在分支指令的对象代码中。

在执行过程中，CPU 评估控制分支的条件。如果分支条件为真，CPU 符号将偏移量扩展到 14 位值，将偏移量添加到当前 PC 中，并将其用作获取下一个 ins 的地址 Truction 并继续执行，而不是用分支之后的下一个指令继续执行。由于偏移量是 8 位二的补值，因此目的地必须在以下地址的 -128 到+127 位置范围内是分支指令的目标代码的最后一个字节。

创建一个简单的无限循环的常见方法是使用分支指令，该指令分支到自己。这有时用于在调试期间结束短代码段。通常，要摆脱这个无限循环，请使用调试主机（通过后台 comMands）停止程序，检查寄存器和内存，或从新位置开始执行。此结构不用于普通应用程序，除非程序检测

到错误并希望强制 COP 监督 Mer 到超时。（无限循环中的分支重复执行，直到看门狗计时器最终导致重置。）

8.3.3 即时寻址模式 (IMM)

在此寻址模式下，操作数位于指令流中操作码之后。当程序员希望使用编写程序时已知的显式值时，使用此寻址模式。#（磅）符号是用于告诉汇编器将操作数用作数据值，而不是访问所需值的地址。

即时操作数的大小总是 8 位。汇编程序将根据需要自动截断或扩展操作数，以匹配指令所需的大小。如果提供了 16 位操作数，大多数汇编程序都会生成警告。

程序员有责任使用#符号来告诉汇编者何时使用即时寻址。汇编器不认为省略#符号是错误，因为生成的语句仍然是有效的指令（alth 这可能意味着与程序员的意图不同）。

8.3.4 微小寻址模式 (TNY)

TNY 寻址模式只能寻址地址图中的前 16 字节，从 0000 美元到 00000F 美元。此寻址模式适用于 INC、DEC、ADD 和 SUB 指令。可以通过在这个内存领域放置最计算密集型的数据来优化系统。

由于 4 位地址嵌入在操作码中，因此指令中必须只包含最不重要的地址四位；这节省了程序空间和执行时间。在执行过程中，CPU 向 4 位操作数添加 10 个高阶 0 地址并使用组合的 14 位地址（000x）来访问预期的操作数。

8.3.5 短寻址模式 (SRT)

SRT 寻址模式只能寻址地址图中的前 32 字节，从 \$0000 到 \$001F。此寻址模式适用于 CLR、LDA 和 STA 指令。可以通过在这个内存领域放置最计算密集型的数据来优化系统。

由于 5 位地址嵌入在操作码中，因此指令中必须只包含最不重要的五位地址；这节省了程序空间和执行时间。在执行过程中，CPU 在 5 位操作数中添加了九个高阶 0D 地址并使用组合的 14 位地址（000x 或 001x 美元）来访问预期的操作数。

8.3.6 直接寻址模式 (DIR)

DIR 寻址模式用于访问位于直接地址空间（0000 美元至 00FF）的操作数。

在执行过程中，CPU 在操作码后面的直接地址操作数的低字节中添加了六个高阶 0。CPU 使用组合的 14 位地址（00xx 美元）来访问预期的操作数。

8.3.7 扩展寻址模式 (EXT)

在扩展寻址模式下，操作数的 14 位地址包含在操作码后接下来两个字节的低阶 14 位的对象代码中。此寻址模式仅用于 RS08 MCU 中跳转目标地址的 JSR 和 JMP 指令。

8.3.8 索引寻址模式 (IX, 通过伪指令实现)

索引寻址模式有时被称为间接寻址模式，因为索引寄存器被用作访问预期操作数的参考。

索引寻址模式的一个重要特征是，操作数地址在执行期间根据位于内存映射\$000F 的 X 索引寄存器的当前内容计算，而不是在程序汇编期间确定的常量地址位置。这允许编写一个程序，该程序根据早期程序指令的结果访问不同的操作数位置（而不是访问编写程序时确定的位置）。

RS08 系列支持的索引寻址模式使用位于\$000F 的寄存器 X 作为索引，位于\$000E 的 D[X]寄存器作为索引数据寄存器。通过对索引寄存器 X 进行编程，可以直接页面中的任何位置都可以读/写 n 通过索引数据寄存器 D[X]。

通过使用 D[X]作为操作数，这些伪指令可以与所有支持直接、短和微小寻址模式的指令一起使用。

8.4 特别行动

CPU 所做的大部分工作都由指令集描述，但必须考虑一些特殊操作，例如在电源优先后，CPU 如何在应用程序开始时启动

应用。程序开始运行后，当前指令通常会决定 CPU 接下来会做什么。两个特殊事件可能导致 CPU 暂时暂停正常程序执行：

- 重置事件迫使 CPU 在应用程序开始时重新开始，这迫使执行从 3FFD 开始。
- 主机开发系统可能导致 CPU 进入活动后台模式，而不是继续应用程序中的下一个指令。

8.4.1 重置序列

处理从重置事件的后缘开始。可能导致重置事件的事物数量可能因 RS08 衍生而略有不同；然而，最常见的来源是：开机

重置，外部重置引脚，低压重置，COP 看门狗超时，非法操作码检测和非法地址访问。有关 MCU 如何识别重置事件并确定内部和外部原因之间差异的更多信息，重新 Fer 到[重置和中断](#)章节。

重置事件迫使 MCU 立即停止它正在做的事情，并开始响应重置。任何正在处理的指令都将立即中止，而不会完成任何剩余的时钟周期。一系列简短的活动已经完成，以确定重置的来源是内部还是外部，并记录重置的原因。在剩余时间内，重置源保持活动状态，内部时钟停止以节省电力。在重置事件的后缘，时钟恢复，CPU 退出重置条件。程序计数器重置为 3FFD，重置发布后将启动指令获取。

为了使设备在重置后从 3FFD 开始从片上内存中执行代码，必须注意不要在重置结束时将 BKDG 引脚压低，因为这将迫使设备进入活动后台模式，CPU 将等待一个命令 D 来自后台通信界面。

8.4.2 中断

RS08 中的中断机制不用于中断指令的正常流；它用于从等待和停止模式中唤醒 RS08。在运行模式下，中断事件必须由 CPU 轮询。中断功能与 Freesca 不兼容 1e 的 HC05、HC08 或 HCS08 家族。

8.4.3 等待和停止模式

分别通过执行 WAIT 或 STOP 指令进入等待和停止模式。在这些模式下，CPU 的时钟被关闭以节省电力，CPU 活动被暂停。CPU 保持这种低功耗状态，直到中断或重置事件唤醒它。请参考[重置和中断](#)关于等待和停止对其他设备外围设备的影响的章节。

8.4.4 主动后台模式

活动后台模式是指 CPU 停止执行用户程序指令并等待来自后台调试系统的串行命令的条件。参考[开发支持](#)有关活动后台模式的详细信息的章节。

算术左移伪指令也可用，因为它的操作与逻辑左移相同。

8.5 简要说明表

指令集摘要命名法

这里列出的术语用于说明说明表 8-1 穿过表 8-2。

运营商

()	=	括号内显示的寄存器或内存位置的内容
←=		加载（阅读：“获取”）
↔	=	交换与
&	=	布尔和
	=	布尔或
⊕=		布尔独家-OR
冒号:	=	把...联系起来
+	=	补充

CPU 寄存器

罗马字母的第一累加器

个字母 =

CCR	=	条件代码寄存器
电脑	=	程序计数器
PCH	=	程序计数器，高阶（最重要）六位
PCL	=	程序计数器，低阶（至少重要）八位
SPC	=	影子程序计数器
SPCH	=	影子程序计数器，高阶（最重要）六位
SPCL	=	影子程序计数器，低阶（最重要）八位

记忆和寻址

罗马字母的第十三个字母 = 内存位置或绝对数据，取决于寻址模式关系 = 相对偏移量，即存储在与分支指令对应的机器代码最后一个字节中的两个补码英语字母中的第二十四字母 = 伪索引寄存器，内存位置 000F 美元

， X 或 D[X] = 内存位置 \$000E 指向伪索引寄存器定义的内存位置（位置 \$000F）

条件代码寄存器（CCR）位

罗马字母表第 26 个字

母 = 零指标 C = 携带/借用

CCR 活动符号

— = 位不受影响
0 = 位被迫为 0

- 1 = 有点被迫到 1
- | = 根据操作结果设置或清除位
- 第二十个罗马字母 = 操作后未定义

机器编码符号

- 女儿 = 低阶八位直接地址\$0000-\$00FF (高字节假设为 00 美元)
- 二 = 一个字节的即时数据
- Hh = 高阶 6 位 14 位扩展地址, 前缀为 2 位 0
- LI = 14 位扩展地址的低阶字节
- Rr = 相对偏移

来源形式

源中的所有内容都形成列, 除了等体字符的表达, 是字面信息, 必须完全如图所示出现在汇编源文件中。最初的 3 到 5 个字母的助记符总是字面表达式。所有逗号、磅符号 (#)、括号和加号 (+) 都是字面字符。

第一 — 任何在 0-7 范围内计算为单个整数的标签或表达式。

十
四
个
英
文
字
母

英 — 任何在 0-\$F 范围内求值为单个十六进制整数的标签或表达式。

语
字
母
中
的
第
二
十
四
个
字
母

Opr8i — 任何评估为 8 位即时值的标签或表达式。

- Opr4a** — 任何评估为微小地址（4 位值）的标签或表达式。该指令将此 4 位值视为 16K 字节地址空间（0000-000F）中地址的低阶四位。这个 4 位值嵌入了操作码中的低阶四位。
 - Opr5a** — 任何评估为短地址（5 位值）的标签或表达式。该指令将此 5 位值视为 16K 字节地址空间（0000-001F）中地址的低阶五位。这个 5 位值嵌入在操作码的低阶 5 位中。
 - Opr8a** — 任何评估为 8 位值的标签或表达式。该指令将此 8 位值视为 16K 字节地址空间中地址的低阶 8 位（\$0000-\$00FF）。
 - Opr16a** — 任何计算为 14 位值的标签或表达式。在 RS08 核心上，上两个位总是 0。该指令将此值视为 16K 字节地址空间中的地址。
- 关系— 任何引用-128 以内的地址的标签或表达式+当前指令对象代码最后一个字节后下一个地址的 127 个位置。汇编器将计算 8 位有符号偏移量，并将其包含在此指令的对象代码中。

地址模式

- INH = 固有（无操作数）
- IMD = 立即直接（在 MOV 指令中）
- IMM = 立即的
- 女儿 = 直接到直接（在 MOV 指令中）
- 迪尔 = 直的
- SRT = 游击手位置
- TNY = 极小的
- 扩展 = 延长了的
- REL = 8 位相对偏移

表 8-1. 指令集摘要（第 1 页，共 6 页）

来源表格	描述	操作	影响 CCR		地址形式	操作码	运算元	周期
			罗马字母表第 2 6 个字母	字母 C				

BEQ 关系	分支, 如果相等	电脑 ← (PC) + 0002 美元 + 关系, 如果 (Z) = 1	—	—	REL	37	Rr	3
--------	----------	--------------------------------------	---	---	-----	----	----	---

1. 这是由普通 RS08 指令集支持的伪指令。
2. 此指令与 HC08 和 HCS08 不同, 因为 RS08 不会自动增加索引寄存器。

表 8-1. 指令集摘要 (第 2 页, 共 6 页)

来源表格	描述	操作	影响 CCR		住址形式	操作码	运算元	周期
			罗马字母表第 2 6 个字母	字母 C				
BGND	背景	进入后台调试模式	—	—	INH	转下页		5+
BHS 关系 ⁽¹⁾	分支, 如果更高或相同 (与 BCC 相同)	电脑 ← (PC) + 0002 美元 + 关系, 如果 (C) = 0	—	—	REL	34	Rr	3
BLO 关系 ⁽¹⁾	分支如果较低 (与 BCS 相同)	电脑 ← (PC) + 0002 美元 + 关系, 如果 (C) = 1	—	—	REL	35	Rr	3
BNE 关系	分支, 如果不相等	电脑 ← (PC) + 0002 美元 + 关系, 如果 (Z) = 0	—	—	REL	36	Rr	3
胸罩关系	总是分支	电脑 ← (PC) + 0002 美元 + 关系	—	—	REL	30	Rr	3
BRN 关系 ⁽¹⁾	分支永远不会	电脑 ← (PC) + 0002 美元	—	—	REL	30	00	3

BRCLR 第十四个英文 字母, Opr8a, 关系				DIR (b0)	01	Dd rr	5
				DIR (b1)	03	dd rr	5
				DIR (b2)	05	dd rr	5
				DIR (b3)	07	dd rr	5
				DIR (b4)	09	dd rr	5
				DIR (b5)	0B	dd rr	5
				DIR (b6)	0D	dd rr	5
				DIR (b7)	0F	dd rr rr	5
				九 (b0)	01	0E rr	5
				九 (b1)	03	0E rr	5
				九 (b2)	05	0E rr	5
				九 (b3)	07	0E rr	5
				九 (b4)	09	0E rr	5
				九 (b5)	0B	0E rr	5
				九 (b6)	0D	0E rr	5
				九 (b7)	0F	0E rr	5
				九 (b0)	01	0F rr	5
				九 (b1)	03	0F rr	5
				九 (b2)	05	0F rr	5
				九 (b3)	07	0F rr	5
				九 (b4)	09	0F rr	5
				九 (b5)	0B	0F rr	5
				九 (b6)	0D	0F rr	5
				九 (b7)	0F	0F rr	5
				DIR (b0)	01	0F rr	5
				DIR (b1)	03	0F rr	5
				DIR (b2)	05	0F rr	5
				DIR (b3)	07	0F rr	5
				DIR (b4)	09	0F rr	5
				DIR (b5)	0B	0F rr	5
				DIR (b6)	0D	0F rr	5
				DIR (b7)	0F	0F rr	5

1. 这是由普通 RS08 指令集支持的伪指令。
2. 此指令与 HC08 和 HCS08 不同，因为 RS08 不会自动增加索引寄存器。

表 8-1. 指令集摘要 (第 3 页, 共 6 页)

来源表格	描述	操作	影响 CCR		地址形式	操作码	运算元	周期
			罗马字母表第 2 6 个字母	字母 C				

BRSET 第十四个英文字母, Opr8a, 关系				DIR (b0)	00	Dd rr	5
				DIR (b1)	02	dd rr	5
				DIR (b2)	04	dd rr	5
				DIR (b3)	06	dd rr	5
				DIR (b4)	08	dd rr	5
				DIR (b5)	0A	dd rr	5
				DIR (b6)	0C	dd rr	5
				DIR (b7)	0E	dd rr rr	5
BRSET 第十四个英文字母, D[X], 关系				九 (b0)	02	0E rr	5
				九 (b1)	04	0E rr	5
				九 (b2)	06	0E rr	5
				九 (b3)	08	0E rr	5
				九 (b4)	0A	0E rr	5
				九 (b5)	0C	0E rr	5
				九 (b6)	0E	0E rr	5
				九 (b7)	02	0F rr	5
				DIR (b0)	04	0F rr	5
				DIR (b1)	06	0F rr	5
				DIR (b2)	08	0F rr	5
				DIR (b3)	0A	0F rr	5
				DIR (b4)	0C	0F rr	5
				DIR (b5)	0E	0F rr	5
				DIR (b6)			
				DIR (b7)			
BRSET 第十四个英文字母, X, 关系	分支如果位第十四个英文字母在内存集中	电脑← (PC) + 0003 美元 + 关系, 如果 (Mn) = 1	—				
BSET 第十四个英文字母, Opr8a				DIR (b0)	10	Dd	5
				DIR (b1)	12	dd	5
				DIR (b2)	14	dd	5
				DIR (b3)	16	dd	5
				DIR (b4)	18	dd	5
				DIR (b5)	1A	dd	5
				DIR (b6)	1C	dd	5
				DIR (b7)	1E	dd	5
BSET 第十四个英文字母, D[X]				九 (b0)	12	0E	5
				九 (b1)	14	0E	5
				九 (b2)	16	0E	5
				九 (b3)	18	0E	5
				九 (b4)	1A	0E	5
				九 (b5)	1C	0E	5
				九 (b6)	1E	0E	5
				九 (b7)	10	0F	5
				九 (b7)	12	0F	5
				DIR (b0)	14	0F	5
				DIR (b1)	16	0F	5
				DIR (b2)	18	0F	5
				DIR (b3)	1A	0F	5
				DIR (b4)	1C	0F	5
				DIR (b5)	1E	0F	5
				DIR (b6)			
				DIR (b7)			
BSET 第十四个英文字母, X	设置位第十四个英文字母在记忆中	明尼苏达州←1	—				

1. 这是由普通 RS08 指令集支持的伪指令。

2. 此指令与 HC08 和 HCS08 不同，因为 RS08 不会自动增加索引寄存器。

表 8-1. 指令集摘要 (第 4 页, 共 6 页)

来源表格	描述	操作	影响 CCR				
			位 5	位 4	位 3	位 2	
BSR 关系	分支子程序	$PC \leftarrow (PC) + 2$ 将 PC 推送到阴影 PC $PC \leftarrow (PC) + 关系$	—	REL	公元	Rr 3	
CBEQA #Opr8i, rel CBEQOpr 8a, rel CBEQ, 英语字母 中的第二 十四个字 母, Rel (1), (2) CBEQ 英 语字母中 的第二十 四个字母, Rel (1)	比较和分支, 如果相等	$PC \leftarrow (PC) + 0003 美元 + 关系$, 如果 (A) - (M) = 00 美元 $PC \leftarrow (PC) + 0003 美元 + 关系$, 如果 (A) - (M) = 00 美元 $PC \leftarrow (PC) + 0003 美元 + 关系$, 如果 (A) - (X) = 00 美元	—	IMM 迪尔 九 迪尔	41 31 31 31	li rr Dd rr 0E rr 0F rr	4 5 5 5
CLC	透明携带钻头	字母 C-0	—	INH	38	1	
CLROpr8 a CLROpr5 a CLR, X (1) CLRA CLRX (1)	无辜	$罗马字母的第十三个字母 \leftarrow 00 美元$ $罗马字母的第一个字母 \leftarrow 00 美元$ $英语字母中的第二十四字母 \leftarrow 00 美元$	1	迪尔 SRT 九 INH INH	3F 8 英语字母中的第二十四字母	女儿 3 2 2 1 2	

									字母 9 英语字母中的第二 十四个字母 8E 4 楼 8 楼			
CMP #Opr8i CMPOpr8 a CMP, X (1) CMP X (1)	将蓄能器与内存进行比较		(A) - (M) (A) - (X)						IMM 迪尔九 INH	A1 B1 B1 B1	二 D d 0 E OF	2 3 3 3
昏迷	伴随物 (一个人的补充)	-	罗马字母的第一个字母← (A)						INH	43		1
DBNZOpr 8a, rel DBNZ, X, rel (1) DBNZ 关 系 DBNZX rel (1)	如果不是零, 则减少和分支	罗马字母的第一个字母← (A) - 01 美元或 M← (M) - 01 美元 电脑← (PC) + 0003 美元 + 关系如果 (结果) ≠ 0 用于 DBNZ 直接 电脑← (PC) + 0002 美元 + 关系如果 (结果) ≠ 0 为 DBNZ 英语字母中的第二十四字母← (X) - 01 美元 电脑← (PC) + 0003 美元 + 关系如果 (结果) ≠ 0						迪尔九 INH INH	3B 3B 4B 3B	Dd rr 0E rr Rr 0F rr	7 7 4 7	
十二月 Opr8a 十二月 Opr4a 12 月, X (1) 德卡 12 月 X 日	减少		罗马字母的第十三个字母← (M) - 01 美元 罗马字母的第一个字母← (A) - 01 美元 英语字母中的第二十四字母← (X) - 01 美元						迪尔 TNY 九 INH 迪尔	3A 5 英语字母中的第二十四个	女 儿	5 4 4 1 4

公司 <i>Opr8a</i> 公司 <i>Opr4a</i> 公司, X ⁽¹⁾ 印加人 INCX ⁽¹⁾	增加	EOR # <i>opr8i</i> EOR <i>opr8a</i> EOR ,X ⁽¹⁾ EOR X	Exclusive OR Memory with Accumulator	$A \leftarrow (A \oplus M)$ $A \leftarrow (A \oplus X)$	↕	字母 5E 4A 5 楼	迪尔 TNY 九 INH INH	3C 2 英语 字母 中的 第二 十 四 个 字 母 2E 4C 2 楼	3 3 5 4 4 1 4	3 3 5 4 4 1 4
		\oplus \oplus		罗马字母的第十三个字母← (M) + 01 美元 罗马字母的第一个字母← (A) + 01 美元 英语字母中的第二四个字母← (X) + 01 美元						
JMP <i>Opr16a</i>	跳	电脑←有效地址			—	扩展	公元 前	Hh II	4	
JSR <i>Opr16a</i>	跳转到 子程序	电脑← (PC) +3 将 PC 推送到阴影 PC 电脑←有效地址			—	扩展	神学 学士	Hh II	4	
LDA # <i>Opr8i</i> LDA <i>Opr8a</i> LDA <i>Opr5a</i> LDA, X ⁽¹⁾	负载蓄 能器从 记忆	罗马字母的第一个字母← (M)			—	IMM 迪尔 SRT 九	A6 B6 字母 C 英语 字母 中的 第二	二 女 儿	2 3 3 3	

移动 <i>Opr8a</i> , <i>opr8a</i> MOV # <i>Opr8i</i> , <i>opr8a</i> MOV D[X], <i>Opr8a</i> 移动 <i>Opr8a</i> , D[X] MOV # <i>Opr8i</i> , D[X]	搬家	(M) 目的地 ← (M) 源	;	—	女儿 IMD IX/DIR DIR/IX IMM/IX	4E 3E 4E 4E 3E	Dd dd li dd 0E dd dd 0E li 0E	5 4 5 5 4
NOP	没有操作	祷告时间	—	—	INH	交流电		1
ORA # <i>Opr8i</i> ORA <i>Opr8a</i> ORA, X ⁽¹⁾ ORA X	包容性或累加器和记忆	罗马字母的第一个字母← (A) (M) A← (A) (X)	;	—	IMM 迪尔 九 迪尔	嗜酒者 互诚协 会 钷 钷 钷	二 Dd 0E 0F	2 3 3 3
罗拉	通过携带向左旋转		;	;	INH	49		1
罗拉	通过携带右旋转		;	;	INH	46		1
RTS	从子程序返回	从影子 PC 中拉出 PC	—	—	INH	从事...	职业	3
SBC # <i>Opr8i</i> SBC <i>Opr8a</i> SBC, X ⁽¹⁾ SBC X	用携带减去	罗马字母的第一个字母←(A) – (M) – (C) 罗马字母的第一个字母← (A) – (X) – (C)	;	;	IMM 迪尔 九 迪尔	A2 B2 B2 B2	二 Dd 0E 0F	2 3 3 3
秒	设置携带钻头	字母 C←1	—	1	INH	39		1
SHA	将 Shadow PC High 与 A 交换	罗马字母的第一个字母⇌ SPCH	—	—	INH	45		1
SLA	将 Shadow PC Low 与 A 交换	罗马字母的第一个字母⇌ SPCL	—	—	INH	42		1
STA <i>Opr8a</i> STA <i>Opr5a</i> STA, X ⁽¹⁾ STA X	存储蓄能器在记忆	罗马字母的第十三个字母← (A)	;	—	迪尔 SRT 九 SRT	B7 E 英语 字母中 的第二 十四个 字母 F 英语字 母中的 第二十 四个字 母 EE EF	女儿	3 2 2 2
STX <i>Opr8a</i> ⁽¹⁾	商店索引寄存器记忆	罗马字母的第十三个字母← (X)	;	—	迪尔	4E	0F dd	5

阻止	将 MCU 置于停止模式		—	—	INH	AE		2+
----	--------------	--	---	---	-----	----	--	----

1. 这是由普通 RS08 指令集支持的伪指令。
2. 此指令与 HC08 和 HCS08 不同，因为 RS08 不会自动增加索引寄存器。

表 8-1. 指令集摘要 (第 6 页, 共 6 页)

来源表格	描述	操作	影响 CCR		地址形式	操作码	运算元	周期
			罗马字母表第 26 个字母	字母 C				
订阅#Opr8i 潜艇 Opr8a 潜艇 Opr4a 子, X ⁽¹⁾ 子 X	减去	罗马字母的第一个字母← (A) — (M) 罗马字母的第一个字母← (A) — (X)	;	;	IMM 迪尔 TNY 九 迪尔	A0 B0 7 英语 字母中的 第二 十四个 字母 7E 7 楼	二 女儿	2 3 3 3 3
税 ⁽¹⁾	将 A 转移到 X	英语字母中的第二十四字母← (A)	;	—	INH	EF		2
TST opr8a ⁽¹⁾ TSTA ⁽¹⁾ TST, X ⁽¹⁾ TSTX ⁽¹⁾	零测试	(M) — 00 美元 (A) — 00 美元 (X) — 00 美元	;	—	女儿 INH 九 INH	4E 嗜酒者 互诚协 会 4E 4E	Dd dd 00 0E 0E 0F 0F	5 2 5 5
TXA ⁽¹⁾	将 X 转移到 A	罗马字母的第一个字母← (X)	;	—	INH	囊性纤 维化		3
等待	将 MCU 置于等待模式		—	—	INH	AF		2+

1. 这是由普通 RS08 指令集支持的伪指令。
2. 此指令与 HC08 和 HCS08 不同，因为 RS08 不会自动增加索引寄存器。

表 8-2。操作码地图

高中 低	迪尔	迪尔	TNY	DIR/REL	INH	TNY	TNY	TNY	SRT	SRT	IMM/INH	DIR/EXT	SRT	SRT	SRT	SRT
0	0	1	2	3	4	5	6	7	8	9	罗马字母 的第一个 字母	字母 b	字母 C	D	E	第六个罗 马字母
1	BRSET0 ⁵ ₃ 迪尔 ²	BSET0 ⁵ ₁ 迪尔	公司 ⁴ TNY	胸罩 ³ REL ²		十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	潜艇 ² IMM ²	潜艇 ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
2	BRCLR0 ⁵ ₃ 迪尔 ²	BCLR0 ⁵ ₁ 迪尔	公司 ⁴ TNY	CBEQ ⁵ 迪尔 ³	CBEQA ⁴ IMM ³	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	CMP ² IMM ²	CMP ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
3	BRSET1 ⁵ ₃ 迪尔 ²	BSET1 ⁵ ₁ 迪尔	公司 ⁴ TNY		SLA ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	SBC ² IMM ²	SBC ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
4	BRCLR1 ⁵ ₃ 迪尔 ²	BCLR1 ⁵ ₁ 迪尔	公司 ⁴ TNY		昏迷 ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹			LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
5	BRSET2 ⁵ ₃ 迪尔 ²	BSET2 ⁵ ₁ 迪尔	公司 ⁴ TNY	BCC ³ REL ²	LSRA ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	和 ² IMM ²	和 ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
6	BRCLR2 ⁵ ₃ 迪尔 ²	BCLR2 ⁵ ₁ 迪尔	公司 ⁴ TNY	BCS ³ REL ²	SHA ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹			LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
7	BRSET3 ⁵ ₃ 迪尔 ²	BSET3 ⁵ ₁ 迪尔	公司 ⁴ TNY	BNE ³ REL ²	罗拉 ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	LDA ² IMM ²	LDA ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
8	BRCLR3 ⁵ ₃ 迪尔 ²	BCLR3 ⁵ ₁ 迪尔	公司 ⁴ TNY	BEQ ³ REL ²		十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹		STA ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
9	BRSET4 ⁵ ₃ 迪尔 ²	BSET4 ⁵ ₁ 迪尔	公司 ⁴ TNY	CLC ¹ INH ¹	LSLA ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	EOR ² IMM ²	EOR ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
罗马字母 的第一个 字母	BRCLR4 ⁵ ₃ 迪尔 ²	BCLR4 ⁵ ₁ 迪尔	公司 ⁴ TNY	秒 ¹ INH ¹	罗拉 ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	ADC ² IMM ²	ADC ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
字母 b	BRSET5 ⁵ ₃ 迪尔 ²	BSET5 ⁵ ₁ 迪尔	公司 ⁴ TNY	十二月 ⁵ 迪尔 ²	德卡 ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	ORA ² IMM ²	ORA ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
字母 C	BRCLR5 ⁵ ₃ 迪尔 ²	BCLR5 ⁵ ₁ 迪尔	公司 ⁴ TNY	DBNZ ⁶ 迪尔 ³	DBNZA ⁴ INH ²	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	补充 ² IMM ²	补充 ³ 迪尔 ²	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
D	BRSET6 ⁵ ₃ 迪尔 ²	BSET6 ⁵ ₁ 迪尔	公司 ⁴ TNY	公司 ⁵ 迪尔 ²	印加人 ¹ INH ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	NOP ¹ INH ³	JMP ⁴ 扩展 ³	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
E	BRCLR6 ⁵ ₃ 迪尔 ²	BCLR6 ⁵ ₁ 迪尔	公司 ⁴ TNY			十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	BSR ³ REL ³	JSR ⁴ 扩展 ³	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹
第六	BRSET7 ⁵ ₃ 迪尔 ²	BSET7 ⁵ ₁ 迪尔	公司 ⁴ TNY	移动 ⁴ IMD ³	移动 ⁵ 女儿 ¹	十二月 ⁴ TNY ¹	补充 ³ TNY ¹	潜艇 ³ TNY ¹	CLR ² SRT ¹	CLR ² SRT ¹	阻止 ²⁺ INH ¹	RTS ³ INH ¹	LDA ³ SRT ¹	LDA ³ SRT ¹	STA ² SRT ¹	STA ² SRT ¹

	灰色框被解码为非法指令										十六进制中操作码的低字节				0	3 潜艇 2 迪尔
个 罗 马 字 母	5 BRCLR7 3 迪尔2	5 BCLR7 2 迪尔	4 公司 TNY	3 CLR 2 迪尔1	1 CLRA INH	4 十二月 TNY	3 补充 TNY	3 潜艇 TNY	2 CLR SRT	2 CLR SRT	2+ 等待 INH	5+ BGND INH	3 LDA SRT	3 LDA SRT	2 STA SRT	2 STA SRT
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
INH	内在的		REL 亲戚													
IMM	立即的		SRT 游击手位置													
迪尔 扩展 女儿	直的 延长了的 直接-直接		TNY 极小的 IMD 即时直接		十六进制中 Opcode 的高字节 字母 b											
															RS08 周期 操作码助记符 字节数/ 寻址模式	

第 9 章

模拟比较器 (RS08ACMPV1)

9.1 简单介绍

模拟比较器模块 (ACMP) 提供了一个电路, 用于比较两个模拟输入电压或将一个模拟输入电压与内部带隔参考电压进行比较。比较器电路可以在整个电源电压范围内运行 (轨道到轨道操作)。

图 9-1 显示 MC9RS08KA8 方框图, 并突出显示 ACMP。

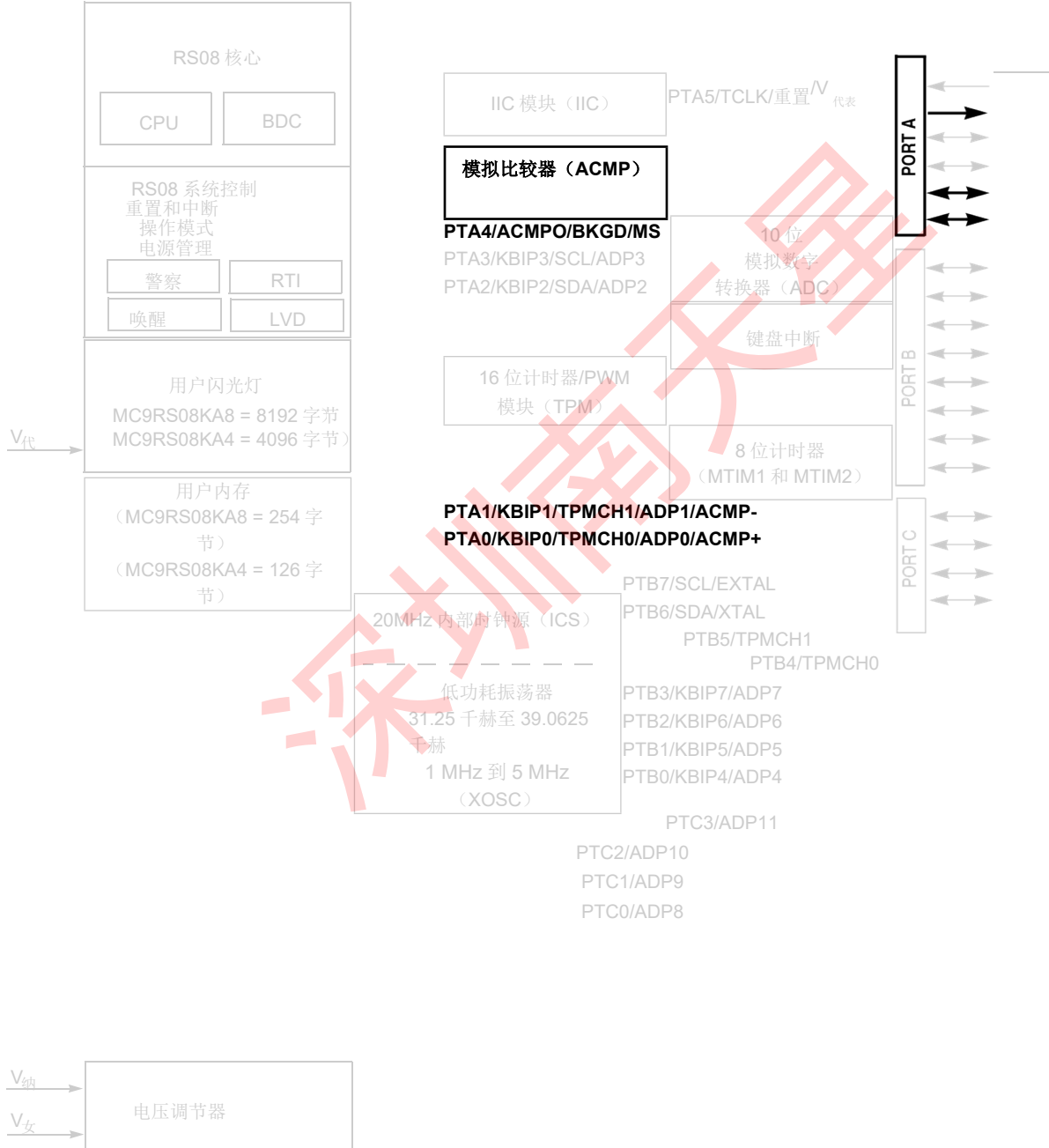


图 9-1. MC9RS08KA8 系列方框图突出显示 ACMP 方块和引脚

模拟比较器 (RS08ACMPV1)

9.1.1 特点

ACMP 具有以下功能:

- 完整的铁路到铁路供应运营
- 少于 40 mV 的输入偏移量
- 少于 15 mV 的滞后
- 在上升边缘、下降边缘或比较器输出的上升或下降边缘上可选择中断
- 与固定内部带隔参考电压进行比较的选项
- 允许比较器输出在引脚上可见的选项, ACMPO
- 在停止模式下保持运行

9.1.2 操作模式

本节定义了等待、停止和后台调试模式下的 ACMP 操作。

9.1.2.1 在等待模式下操作

如果在执行 WAIT 指令之前启用, ACMP 将继续在等待模式下运行。

因此, 如果启用了 ACMP 中断, 则可以使用 ACMP 将 MCU 从等待模式中恢复出来 (ACIE = 1)。为了尽可能低的电流消耗, 如果在等待模式下不需要作为中断源, 则必须由软件禁用 ACMP。

9.1.2.2 在停止模式下操作

如果启用, ACMP 将继续以停止模式运行, 并且比较操作仍然处于活动状态。如果启用了 ACOPE, 比较器输出与正常操作模式一样, 比较器输出被放置在外围引脚上。MCU 被带出了 stop 当发生比较事件并启用 ACIE 时; 相应地设置 ACF 标志。

如果通过重置退出停止, ACMP 将进入重置状态。

9.1.2.3 在活动后台模式下操作

当 MCU 处于活动后台模式时, ACMP 将继续正常运行。

9.1.3 方框图

模拟比较器模块的框图显示在图 9-2。

模拟比较器 (RS08ACMPV1)

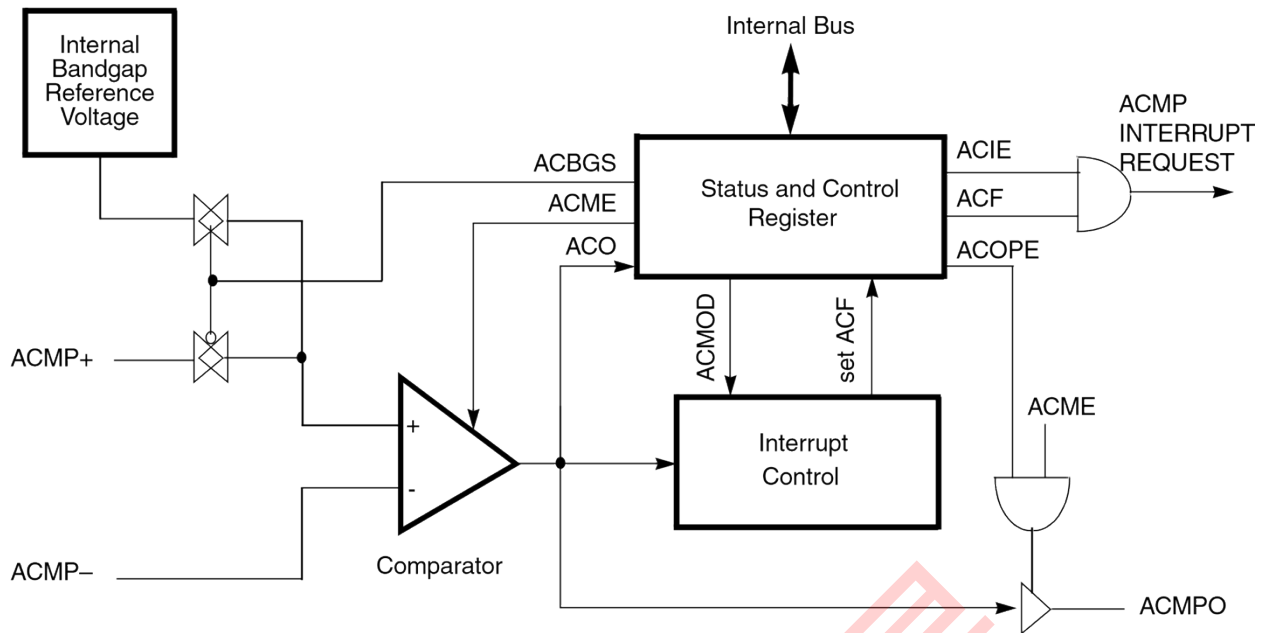


图 9-2. 模拟比较器 (ACMP) 块图

9.2 外部信号描述

ACMP 有两个模拟输入引脚，ACMP+和 ACMP-，以及一个数字输出引脚 ACMPO。每个输入引脚都可以接受在 MCU 的整个工作电压范围内变化的输入电压。

如图所示图 9-2，ACMP-引脚连接到比较器的反转输入，如果 ACBGS=0，ACMP+引脚连接到比较器的非反转输入。如图所示图 9-2，ACMPO 引脚可以启用来驱动外部引脚。

ACMP 的信号属性显示在表 9-1。

表 9-1. 信号属性

信号	函数	I/O
ACMP-	反转模拟输入到 ACMP (负输入)	我
ACMP+	ACMP 的非反转模拟输入 (正面输入)	我
ACMPO	ACMP 的数字输出	字母 O

9.3 注册定义

ACMP 包括一个寄存器：

- 8 位状态和控制寄存器

有关所有 ACMP 寄存器的绝对地址分配，请参阅本数据表内存章节中的直接页面寄存器摘要。

模拟比较器 (RS08ACMPV1)

9.3.1 ACMP 状态和控制登记册 (ACMPSC)

ACMPSC 包含用于启用和配置 ACMP 的状态标志和控制位。

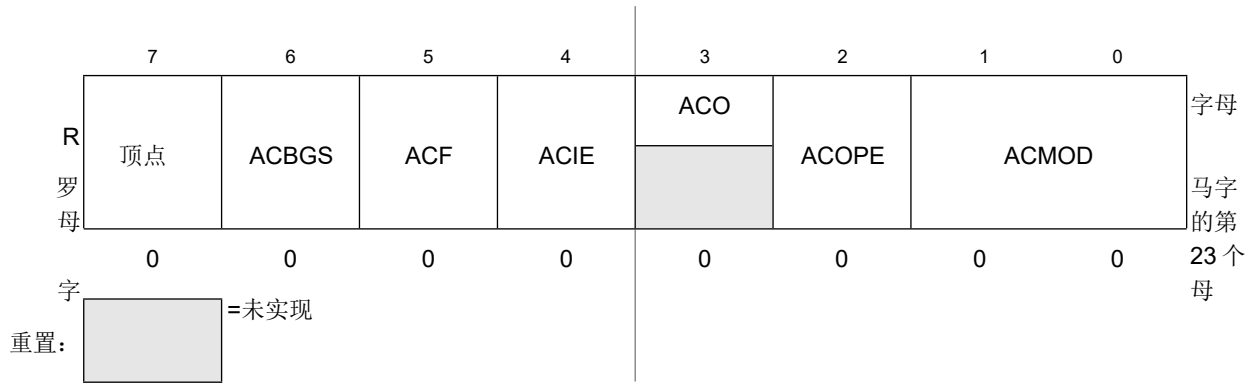


图 9-3. ACMP 状态和控制登记册 (ACMPSC)

表 9-2. ACMPSC 字段描述

字段	描述
7 ACME	模拟比较器模块启用 — ACME 启用 ACMP 模块。0 ACMP 未启用。 1 ACMP 已启用。
6 ACBGS	模拟比较器带盖选择 — ACBGS 用于在内部带隙参考电压或 ACMP+ 引脚之间进行选择，作为模拟比较器的非反转输入。 0 外部引脚 ACMP+ 被选为比较器的非反转输入。 1 内部带隙参考电压被选为比较器的非反转输入。
5 ACF	模拟比较器标志 — 当比较事件发生时，会设置 ACF。比较事件由 ACMOD 定义。ACF 通过给 ACF 写一个来清除。 0 比较事件没有发生。 1 比较事件已经发生。
4 ACIE	模拟比较器中断启用 — ACIE 启用了 ACMP 的中断。设置 ACIE 时，设置 ACF 时将断言中断。 0 中断禁用。 1 启用中断。
3 ACO	模拟比较器输出 — 读取 ACO 将返回模拟比较器输出的当前值。ACO 重置为 0，当 ACMP 被禁用时，将读为 0 (ACME = 0)。
2 ACOPE	模拟比较器输出引脚启用 — ACOPE 用于将比较器输出放置在外部引脚 ACMPO 上。ACOPE 仅在 ACMP 处于活动状态时控制引脚 (ACME=1)。 0 模拟比较器输出在 ACMPO 上不可用。 1 模拟比较器输出在 ACMPO 上被驱动。

1:0 ACMOD	<p>模拟比较器模式— ACMOD 选择设置 ACF 的比较事件类型。</p> <p>00 编码 0 — 比较器输出下降边缘。01 编码 1-比较器输出上升边缘。</p> <p>10 编码 2-比较器输出下降边缘。</p> <p>11 编码 3——比较器输出上升或下降边缘。</p>
--------------	------------------------------------------------------------------------------------------------------------------------------------------------------

9.4 功能描述

模拟比较器可用于比较应用于 ACMP+和 ACMP-的两个模拟输入电压；或者可用于比较应用于 ACMP-的模拟输入电压与内部带隔参考

深圳南天星

电压。ACBGS 用于在带间隙参考电压或 ACMP+ 引脚之间进行选择，作为模拟比较器非反转输入的输入。

当非反转输入大于反转输入时，比较器输出高，当非反转输入小于反转输入时，比较器输出低。ACMOD 用于选择将导致设置 ACF 的条件。ACF 可以在比较器输出的上升边缘，比较器输出的下降边缘，或上升或下降边缘（切换）。比较器输出可以直接通过 ACO 读取。比较器输出也可以使用 A 驱动到 ACMPO 引脚上 COPE。

笔记

比较器输入是高阻抗模拟引脚，对噪音敏感。诺伊西 $V_{\text{女儿}}$ 和/或与模拟输入相邻的引脚切换可能会导致比较器偏移/滞后性能超过指定值。最大源阻抗限制在 MC9RS08KA8 系列中指定的值 *数据表*。为了实现最大性能，建议设备进入等待/停止模式进行 ACMP 测量，并且必须避免相邻的引脚切换。

深圳南天星

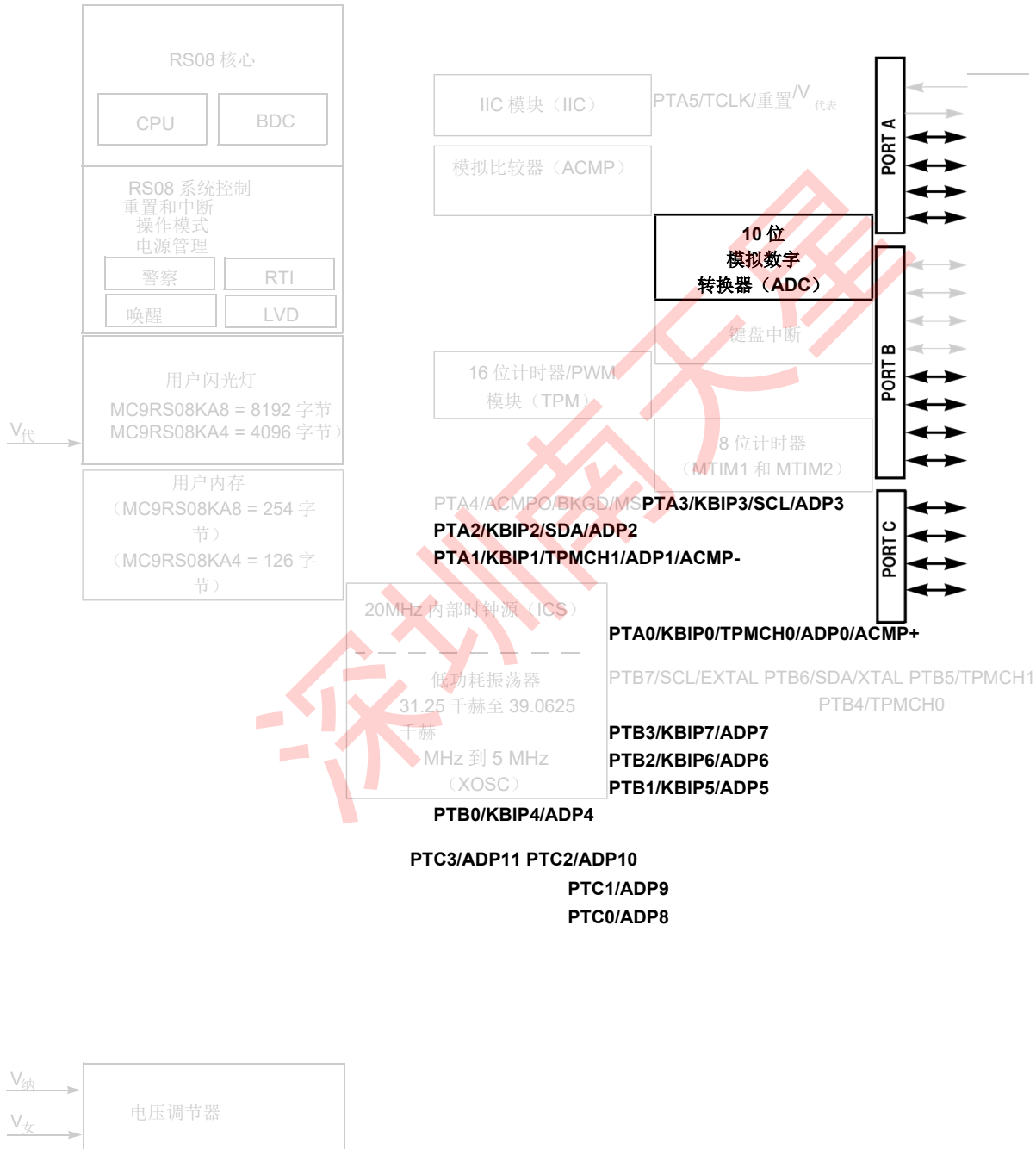
第 10 章

10 位模数转换器 (RS08ADC10V1)

10.1 简单介绍

10 位模数转换器 (ADC) 是连续近似 ADC, 专为在芯片上集成微控制器系统内运行而设计。

图 10-1 显示 MC9RS08KA8 系列, 突出显示 ADC 模块和引脚。



MC9RS08KA8 Series Reference Manual, Rev. 4

图 10-1. MC9RS08KA8 系列块图突出显示 ADC 块和引脚

第 10 章 10 位模数转换器 (RS08ADC10V1)

10.1.1 模块配置

本节提供在 MC9RS08KA8 系列设备上配置 ADC 的设备特定信息。

10.1.1.1 模拟电源和电压参考连接

V_{DDAD} 和 $V_{REFHADC}$ 的源在内部连接到 $V_{\text{女儿}}$ 别针。 V_{SSAD} 和 $V_{REFLADC}$ 的源在内部连接到 $V_{\text{纳粹党卫军}}$ 别针。

10.1.1.2 备用时钟

ADC 可以使用 MCU 总线时钟、总线时钟除以二或模块内的本地异步时钟 (ADACK) 进行转换。MC9RS08KA8 系列 MCU 设备的备用时钟 ALTCK 输入没有实现。

10.1.1.3 硬件触发器

ADC 硬件触发器 ADHWT 从实时中断 (RTI) 计数器输出。ICSERCLK 或 RTI 块内的标称 1 kHz 时钟源可以时钟 RTI 计数器。

输入时钟频率和 RTIS 位决定了 RTI 周期。RTI 计数器是一个自由运行的计数器，以 RTIS 位确定的 RTI 速率生成溢出。当启用 ADC 硬件触发器时，在 RTI 计数器溢出时启动转换。

RTI 可以配置为在 MCU 中导致硬件触发器运行、等待和停止。

笔记

要获得快速的 RTI 硬件触发器，RTI 时钟源必须是高频外部时钟源。

10.1.1.4 模拟引脚启用

MC9RS08KA8 系列设备上的 ADC 仅包含两个模拟引脚启用寄存器，APCTL1 和 APCTL2。

MC9RS08KA8 系列设备的 ADC 通道分配如下表所示。保留的通道转换为未知值。连接到 I/O 引脚的通道具有相关的引脚控制位，如图所示。

表 10-1. ADC 通道分配

ADCH	沟渠	输入的信息	引脚控制	ADCH	沟渠	输入的信息	引脚控制
00000	AD0	PTA0//ADP0	ADPC0				
00001	AD1	PTA1//ADP1	ADPC1				
00010	AD2	PTA2//ADP2	ADPC2				

00011	AD3	PTA3/ADP3	ADPC3
00100	公元 4	PTB0/ADP4	ADPC4
00101	AD5	PTB1/ADP5	ADPC5

第 10 章 10 位模数转换器 (RS08ADC10V1)

表 10-1. ADC 通道分配 (续)

ADCH	沟渠	输入的信息	引脚控制
00110	AD6	PTB2/ADP6	ADPC6
00111	公元 7	PTB3/ADP7	ADPC7
01000	公元 8	PTC0/ADP8	ADPC8
01001	AD9	PTC1/ADP9	ADPC9
01010	AD10	PTC2/ADP10	ADPC10
01011	AD11	PTC3/ADP11	ADPC11

ADCH	沟渠	输入的信息	引脚控制
11100	矜持的	不适用的	不适用的
11101	VREFH	V 女儿	不适用的
11110	VREFL	V 纳粹党卫军	不适用的
11111	模块已禁用	祷告时间	不适用的

10.1.1.5 低功耗模式操作

ADC 可以在停止模式下运行，但需要在 SPMSC1 中设置 LVDSE 和 LVDE。

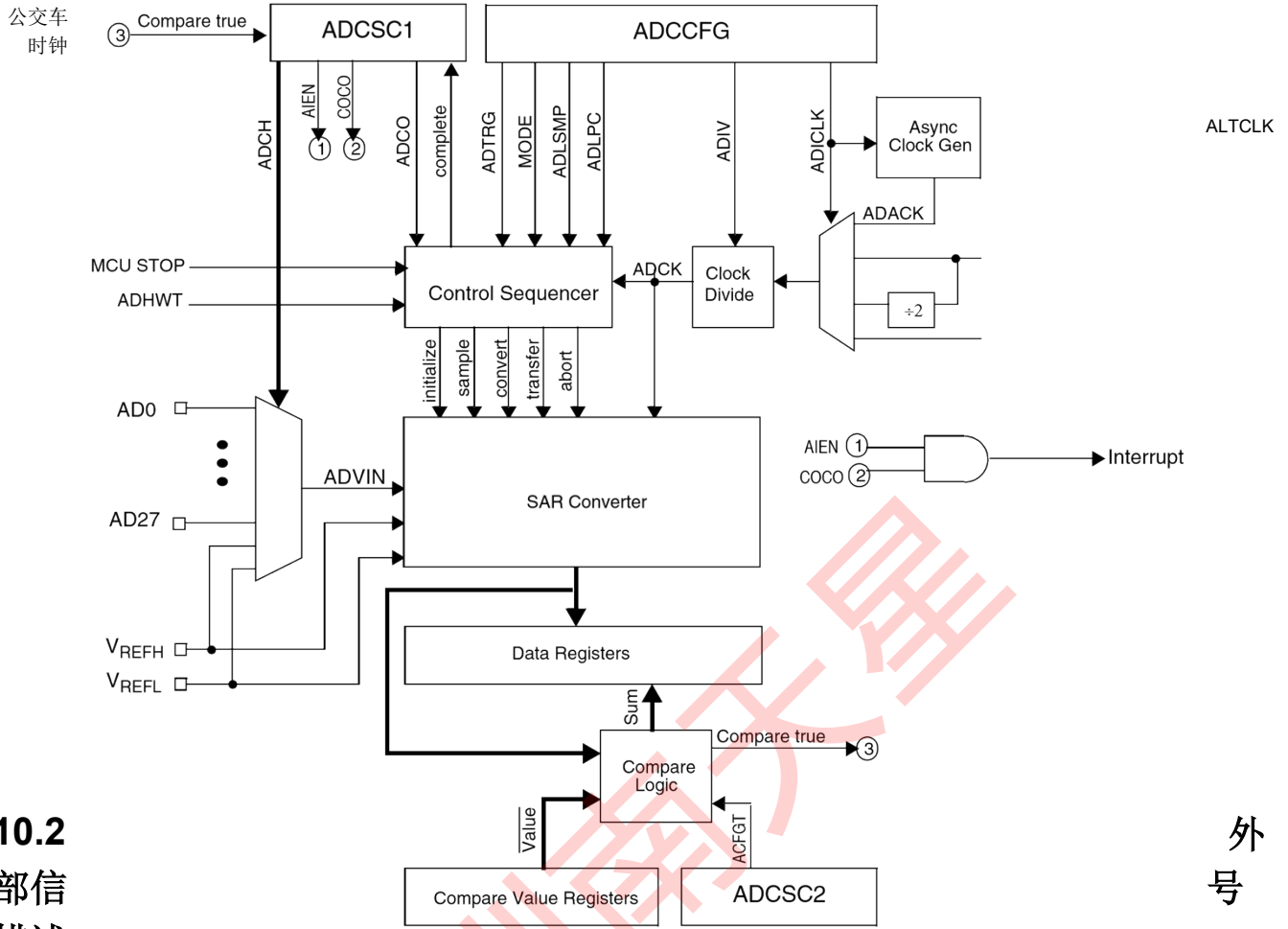
10.1.2 特点

ADC 模块的功能包括：

- 具有 10 位分辨率的线性连续近似算法。
- 多达 28 个模拟输入。
- 以 10 位或 8 位右对齐格式格式化的输出。
- 单次或连续转换（单次转换后自动返回闲置）。
- 可配置的采样时间和转换速度/功率。
- 转换完成标志和中断。
- 输入时钟最多可从四个来源中选择。
- 在等待或停止模式下操作，以降低噪音操作。
- 用于低噪声操作的异步时钟源。
- 可选的异步硬件转换触发器。
- 自动比较小于或大于或等于可编程值的中断。

10.1.3 方框图

图 10-2 提供 ADC 模块的框图。



10.2 部信 描述

外号

Figure 10-2. ADC Block Diagram

ADC 模块支持多达 28 个单独的模拟输入。它还需要四个供应/参考/接地连接。

表 10-2. 信号属性

名字	函数
AD27-AD0	模拟通道输入
VREFH	高参考电压
VREFL	低参考电压
VDDAD	模拟电源
VSSAD	模拟地面

10.2.1 模拟功率 (V_{DDAD})

ADC 模拟部分使用 V_{DDAD} 作为其电源连接。在某些软件包中, V_{DDAD} 在内部连接到 V_{DD} 。如果外部可用, 请连接 V_{DDAD} 引脚到与 V_{DD} 相同的电压电位。可能需要外部过滤来确保清洁 V_{DDAD} 为了取得好成绩。

10.2.2 模拟地面 (V_{SSAD})

ADC 模拟部分使用 V_{SSAD} 作为它的地面连接。在某些软件包中, V_{SSAD} 在内部连接到 V_{SS} 。如果外部可用, 请连接 V_{SSAD} 引脚到与 V_{SS} 相同的电压电位。

10.2.3 电压参考高 (V_{REFH})

V_{REFH} 是转换器的高参考电压。在某些软件包中, V_{REFH} 在内部连接到 V_{DDAD} 。如果外部可用, V_{REFH} 可以连接到与 V_{DDAD} 相同的电位, 或者可以由最小 V_{DDAD} 之间的外部源驱动。规格和 V_{DDAD} 潜力 (V_{REFH} 永远不能超过 V_{DDAD})。

10.2.4 电压参考低 (V_{REFL})

V_{REFL} 是转换器的低参考电压。在某些软件包中, V_{REFL} 在内部连接到 V_{SSAD} 。如果外部可用, 请连接 V_{REFL} 引脚到与 V_{SSAD} 相同的电压电位。

10.2.5 模拟通道输入 (ADx)

ADC 模块支持多达 28 个单独的模拟输入。通过 $ADCH$ 通道选择位选择一个输入进行转换。

10.3 注册定义

这些内存映射寄存器控制和监控 ADC 的操作:

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

10.3.1 状态和控制登记册 1 (ADCSC1)

本节介绍 ADC 状态和控制寄存器 1 (ADCSC1) 的功能。编写 ADCSC1 会中止当前转换并启动新的转换 (如果 ADCH 位等于所有 1 以外的值)。

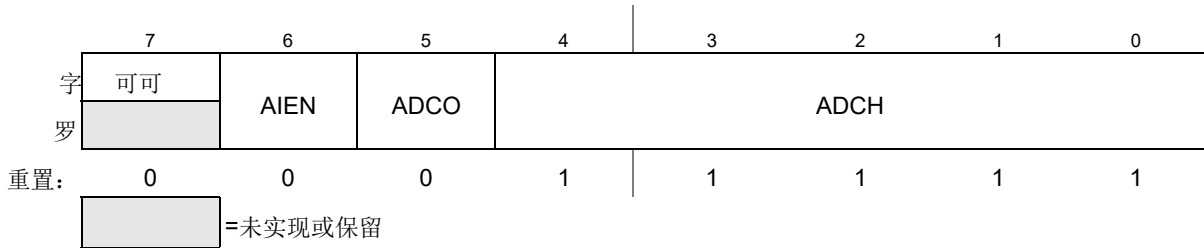


图 10-3. 状态和控制登记册 (ADCSC1)

表 10-3. ADCSC1 注册字段描述

字段	描述
7 可可	转换完整标志—COCO 标志是一个只读位，在禁用比较功能时，每次完成转换时都会设置 (ACFE = 0)。当启用比较功能 (ACFE = 1) 时，仅当比较结果为真时，才会在完成转换时设置 COCO 标志。每当写入 ADCSC1 或读取 ADCRL 时，此位都会被清除。 0 转换未完成 1 转换完成
6 AIEN	中断启用— AIEN 启用转换完全中断。当 COCO 在 AIEN 高时被设置时，会断言中断。 0 转换完全中断被禁用 1 启用转换完全中断
5 ADCO	连续转换启用— ADCO 用于实现连续转换。 0 触发操作后的一次转换 ¹ 1 选择在触发操作后启动的连续转换。
4:0 ADCH	输入通道选择—ADCH 位形成一个 5 位字段，该字段选择其中一个输入通道。输入通道详见图 10-4。当通道选择位都设置为 1 时，连续近似转换器子系统将关闭。此功能允许显式禁用 ADC，并将输入通道与所有源隔离。以这种方式终止连续转换可以防止执行额外的单次转换。当未启用连续转换时，无需将通道选择位设置为所有 1 来将 ADC 置于低功耗状态，因为当转换完成时，模块会自动进入低功耗状态。

¹ 看见表 10-4 如何指定硬件或软件触发器类型 (ADTRG)。

图 10-4. 输入通道选择

ADCH	输入选择	ADCH	输入选择
00000	AD0	10000	公元 16
00001	AD1	10001	公元 17

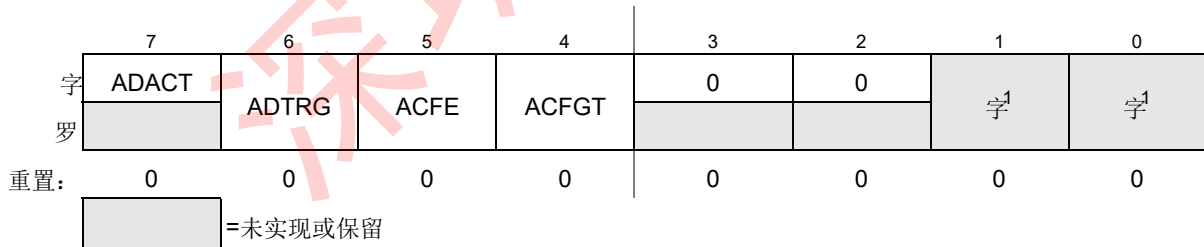
00010	AD2	10010	AD18
00011	AD3	10011	公元 19 年
00100	公元 4	10100	AD20
00101	AD5	10101	公元 21 年
00110	AD6	10110	AD22
00111	公元 7	10111	AD23

图 10-4。输入通道选择 (续)

ADCH	输入选择	ADCH	输入选择
01000	公元 8	11000	公元 24
01001	AD9	11001	AD25
01010	AD10	11010	公元 26
01011	AD11	11011	AD27
01100	AD12	11100	矜持的
01101	AD13	11101	VREFH
01110	公元 14 年	11110	VREFL
01111	AD15	11111	模块已禁用

10.3.2 状态和控制登记册 2 (ADCSC2)

ADCSC2 寄存器用于控制 ADC 模块的比较功能、转换触发器和转换活动。



¹位 1 和 0 是保留位，必须始终写入 0。

图 10-5。状态和控制登记册 2 (ADCSC2)

表 10-4。ADCSC2 注册字段描述

字段	描述
----	----

7 ADACT	转换活跃 — ADACT 表示转换正在进行中。ADACT 在转换启动时设置，并在转换完成或中止时清除。0 转换未进行中 1 正在进行转换
6 ADTRG	转换触发器选择 — ADTRG 选择用于启动转换的触发器类型。 0 选择软件触发器。(在写入 ADCSC1 后启动转换)。 1 选择硬件触发器。(在断言 ADHWT 输入后启动转换)。
5 ACFE	比较功能启用 — ACFE 用于启用比较功能。0 比较功能已禁用 1 启用比较功能
4 ACFGT	比较大于启用的功能 — ACFGT 将比较功能配置为在被监控的输入转换时触发。 0 当输入小于比较水平时触发的比较 1 当输入大于或等于比较水平时触发的比较

10.3.3 数据结果高寄存器 (ADCRH)

ADCRH 包含 10 位转换结果的上两个位。当配置为 8 位转换时，ADR8 和 ADR9 都等于零。每次转换完成时都会更新 ADCRH，除非启用了自动比较且未满足比较条件。在 10 位模式下，读取 ADCRH 可防止 ADC 将后续转换结果传输到结果寄存器，直到读取 ADCRL。如果直到下一次转换完成后才读取 ADCRL，那么中间 conversion 结果将丢失。在 8 位模式下，没有与 ADCRL 互锁。如果 MODE 位被更改，ADCRH 中的任何数据都将失效。

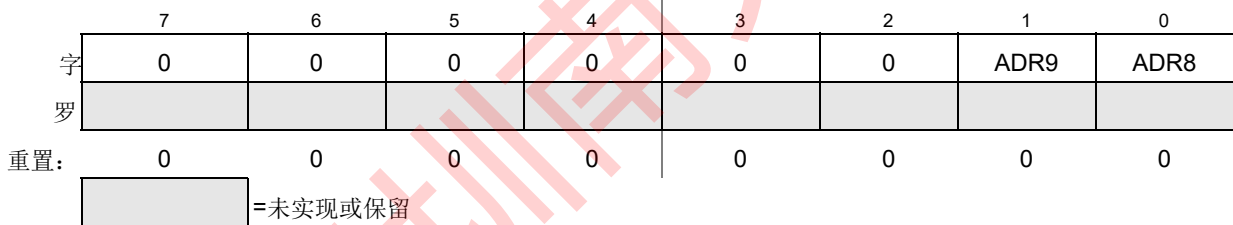


图 10-6. 数据结果高寄存器 (ADCRH)

10.3.4 数据结果低寄存器 (ADCRL)

ADCRL 包含 10 位转换结果的下 8 位，以及 8 位转换的所有 8 位。此寄存器每次转换完成时都会更新，除非启用了自动比较且未满足比较条件。在 10 位模式，读取 ADCRH 阻止 ADC 将后续转换结果传输到结果寄存器，直到读取 ADCRL。如果直到下一次转换完成后才读取 ADCRL，则中间转换结果将丢失。在 8 位模式，没有与 ADCRH 联锁。当 MODE 位被更改时，ADCRL 中的数据将变得无效。

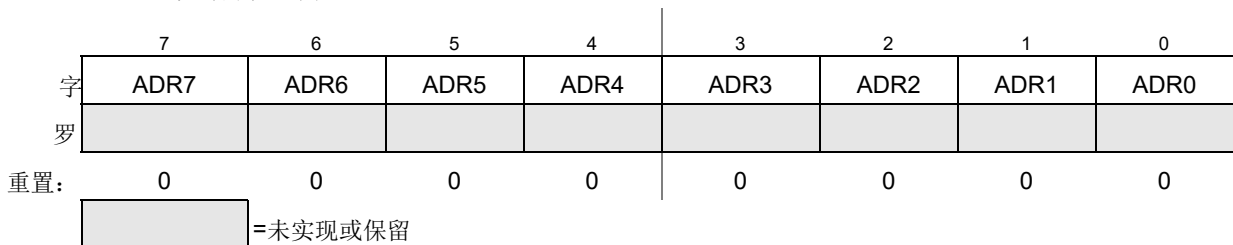


图 10-7。数据结果低寄存器 (ADCRL)

10.3.5 比较价值高寄存器 (ADCCVH)

该寄存器保存 10 位比较值的上两个位。当启用比较功能时，在 10 位模式下转换后，将这些位与结果的上两个位进行比较。在 8 位操作中，在比较期间不使用 ADCCVH。

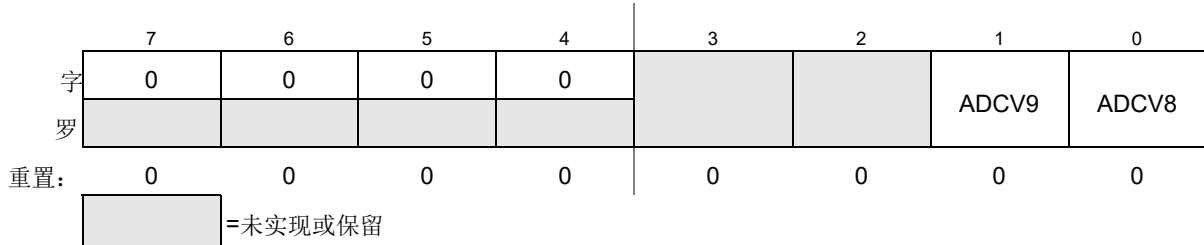


图 10-8。比较价值高寄存器 (ADCCVH)

10.3.6 比较值低寄存器 (ADCCVL)

此寄存器持有 10 位比较值的下 8 位，或 8 位比较值的所有 8 位。在 10 位或 8 位模式下转换后，将 ADCV7: ADCV0 与结果的下 8 位进行比较。

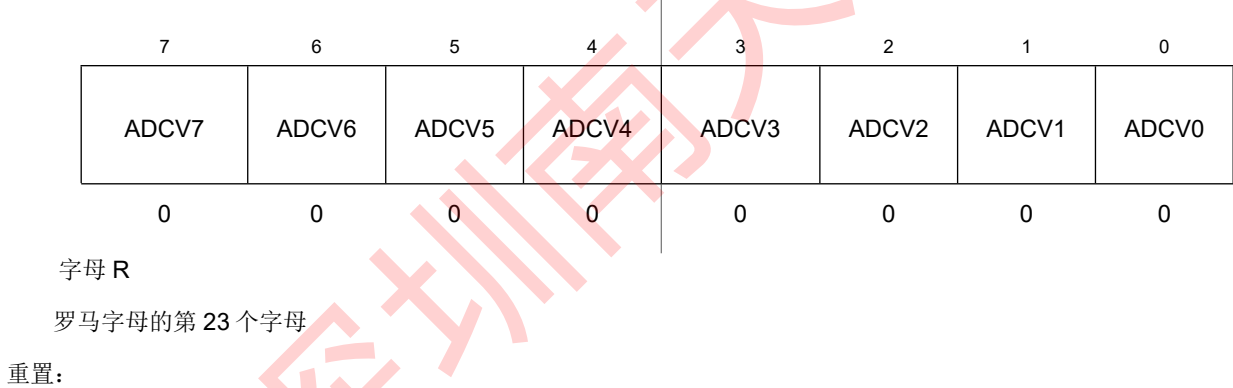
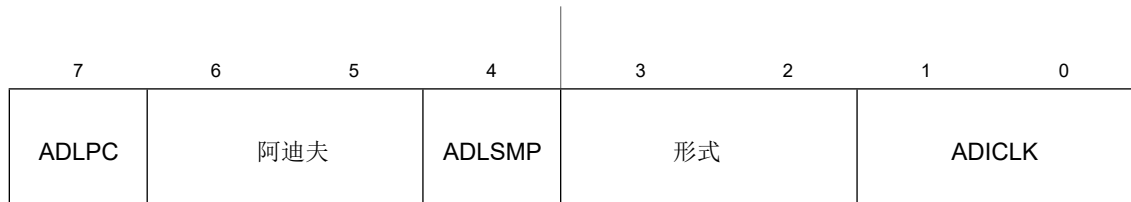
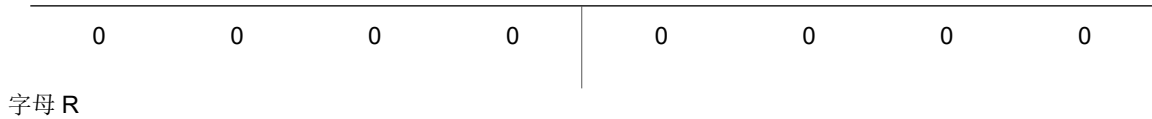


图 10-9。比较价值低寄存器 (ADCCVL)

10.3.7 配置寄存器 (ADCCFG)

ADCCFG 用于选择操作模式、时钟源、时钟分频，并配置低功耗或长采样时间。





字母 R

罗马字母的第 23 个字母

重置:

图 10-10. 配置寄存器 (ADCCFG)

表 10-5. ADCCFG 注册字段描述

字段	描述
7 ADLPC	低功耗配置 — ADLPC 控制连续近似转换器的速度和功率配置。当不需要更高的采样率时, 这可以优化功耗。 0 高速配置 1 低功耗配置: {FC31}功耗以牺牲最大时钟速度为代价。
6:5 ADIV	时钟分割选择 —ADIV 选择 ADC 用于生成内部时钟 ADCK 的除比。表 10-6 显示可用的时钟配置。

表 10-5. ADCCFG 寄存器字段描述 (续)

字段	描述
4 ADLSMP	长采样时间配置 —ADLSMP 在长采样期和短采样期之间进行选择。这调整了采样周期, 以便准确采样更高的阻抗输入, 或最大限度地提高低阻抗输入的转换速度。更长的采样时间也可以用来降低如果不需要高转换率, 则启用连续转换时的所有功耗。 0 采样时间短 1 采样时间长
3:2 形式	转换模式选择 —模式位用于在 10 位或 8 位操作之间进行选择。看见表 10-7。
1:0 ADICLK	输入时钟选择 — ADICLK 位选择输入时钟源以生成内部时钟 ADCK。看见表 10-8。

表 10-6. 时钟分割选择

阿迪夫	除数比	时钟速率
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 10-7. 转换模式

形式	模式描述
00	8 位转换 (N=8)
01	矜持的

10	10 位转换 (N=10)
11	矜持的

表 10-8. 输入时钟选择

ADICLK	选定的时钟源
00	公交车时钟
01	公交车时钟除以 2
10	备用时钟 (ALTCLK)
11	异步时钟 (ADACK)

10.3.8 引脚控制 1 寄存器 (APCTL1)

引脚控制寄存器禁用了用作模拟输入的 MCU 引脚的 I/O 端口控制。APCTL1 控制与 ADC 模块通道 0-7 关联的引脚。



字母 R

罗马字母的第 23 个字母

重置:

图 10-11. 引脚控制 1 寄存器 (APCTL1)

表 10-9. APCTL1 寄存器字段描述

字段	描述
7 ADPC7	ADC 引脚控制 7 — ADPC7 用于控制与通道 AD7 关联的引脚。 0 启用 AD7 引脚 I/O 控制 1 AD7 引脚 I/O 控制被禁用
6 ADPC6	ADC 引脚控制 6 —ADPC6 用于控制与通道 AD6 关联的引脚。 0 启用 AD6 引脚 I/O 控制 1 AD6 引脚 I/O 控制已禁用
5 ADPC5	ADC 引脚控制 5 — ADPC5 用于控制与通道 AD5 关联的引脚。 0 启用 AD5 引脚 I/O 控制 1 AD5 引脚 I/O 控制禁用
4 ADPC4	ADC 引脚控制 4 — ADPC4 用于控制与通道 AD4 关联的引脚。 0 启用 AD4 引脚 I/O 控制 1 AD4 引脚 I/O 控制已禁用

3 ADPC3	ADC 引脚控制 3 —ADPC3 用于控制与通道 AD3 关联的引脚。 0 启用 AD3 引脚 I/O 控制 1 AD3 引脚 I/O 控制已禁用
2 ADPC2	ADC 引脚控制 2 — ADPC2 用于控制与通道 AD2 关联的引脚。 0 启用 AD2 引脚 I/O 控制 1 AD2 引脚 I/O 控制已禁用
1 ADPC1	ADC 引脚控制 1 — ADPC1 用于控制与通道 AD1 关联的引脚。启用 0 AD1 引脚 I/O 控制 1 AD1 引脚 I/O 控制被禁用
0 ADPC0	ADC 引脚控制 0 — ADPC0 用于控制与通道 AD0 关联的引脚。 0 启用 AD0 引脚 I/O 控制 1 AD0 引脚 I/O 控制已禁用

10.3.9 引脚控制 2 寄存器 (APCTL2)

APCTL2 用于控制 ADC 模块的通道 8-15。

7	6	5	4	3	2	1	0
ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0	0	0	0	0	0	0	0

字母 R

罗马字母的第 23 个字母

重置:

图 10-12. 引脚控制 2 寄存器 (APCTL2)

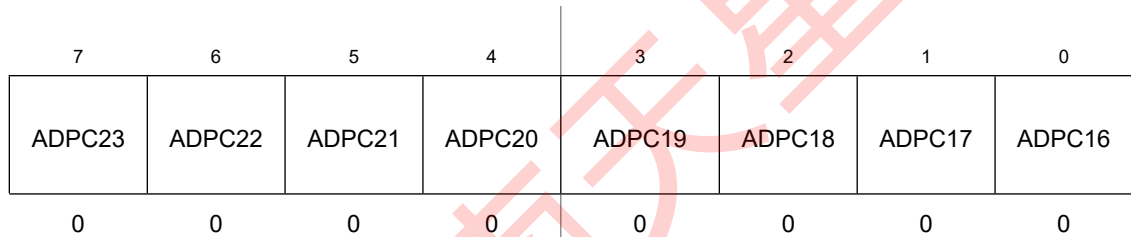
表 10-10. APCTL2 寄存器字段描述

字段	描述
7 ADPC15	ADC 引脚控制 15 — ADPC15 用于控制与通道 AD15 关联的引脚。 0 启用 AD15 引脚 I/O 控制 1 AD15 引脚 I/O 控制已禁用
6 ADPC14	ADC 引脚控制 14 —ADPC14 用于控制与通道 AD14 关联的引脚。 0 启用 AD14 引脚 I/O 控制 1 AD14 引脚 I/O 控制被禁用
5 ADPC13	ADC 引脚控制 13 — ADPC13 用于控制与通道 AD13 关联的引脚。 0 启用 AD13 引脚 I/O 控制 1 AD13 引脚 I/O 控制已禁用
4 ADPC12	ADC 引脚控制 12 —ADPC12 用于控制与通道 AD12 关联的引脚。 0 启用 AD12 引脚 I/O 控制 1 AD12 引脚 I/O 控制被禁用

3 ADPC11	ADC 引脚控制 11 — ADPC11 用于控制与通道 AD11 关联的引脚。 0 启用 AD11 引脚 I/O 控制 1 AD11 引脚 I/O 控制被禁用
2 ADPC10	ADC 引脚控制 10 — ADPC10 用于控制与通道 AD10 关联的引脚。 0 启用 AD10 引脚 I/O 控制 1 AD10 引脚 I/O 控制已禁用
1 ADPC9	ADC 引脚控制 9 — ADPC9 用于控制与通道 AD9 关联的引脚。 0 启用 AD9 引脚 I/O 控制 1 AD9 引脚 I/O 控制被禁用
0 ADPC8	ADC 引脚控制 8 —ADPC8 用于控制与通道 AD8 关联的引脚。启用 0 AD8 引脚 I/O 控制 1 AD8 引脚 I/O 控制禁用

10.3.10 引脚控制 3 寄存器 (APCTL3)

APCTL3 用于控制 ADC 模块的 16-23 通道。



字母 R

罗马字母的第 23 个字母

重置:

图 10-13。引脚控制 3 寄存器 (APCTL3)

表 10-11. APCTL3 寄存器字段描述

字段	描述
7 ADPC23	ADC 引脚控制 23 — ADPC23 用于控制与通道 AD23 关联的引脚。 0 启用 AD23 引脚 I/O 控制 1 AD23 引脚 I/O 控制禁用
6 ADPC22	ADC 引脚控制 22 —ADPC22 用于控制与通道 AD22 关联的引脚。 0 启用 AD22 引脚 I/O 控制 1 AD22 引脚 I/O 控制已禁用
5 ADPC21	ADC 引脚控制 21 — ADPC21 用于控制与通道 AD21 关联的引脚。 0 启用 AD21 引脚 I/O 控制 1 AD21 引脚 I/O 控制禁用
4 ADPC20	ADC 引脚控制 20 — ADPC20 用于控制与通道 AD20 相关的引脚。 0 启用 AD20 引脚 I/O 控制 1 AD20 引脚 I/O 控制禁用
3 ADPC19	ADC 引脚控制 19 —ADPC19 用于控制与通道 AD19 关联的引脚。 0 启用 AD19 引脚 I/O 控制 1 AD19 引脚 I/O 控制禁用
2 ADPC18	ADC 引脚控制 18 —ADPC18 用于控制与通道 AD18 关联的引脚。 0 启用 AD18 引脚 I/O 控制 1 AD18 引脚 I/O 控制被禁用
1 ADPC17	ADC 引脚控制 17 —ADPC17 用于控制与通道 AD17 关联的引脚。 0 启用 AD17 引脚 I/O 控制 1 AD17 引脚 I/O 控制禁用
0 ADPC16	ADC 引脚控制 16 — ADPC16 用于控制与通道 AD16 关联的引脚。 0 启用 AD16 引脚 I/O 控制 1 AD16 引脚 I/O 控制禁用

10.4 功能描述

在重置期间或当 ADCH 位都很高时，ADC 模块被禁用。当转换已完成且尚未启动另一个转换时，该模块处于空闲状态。闲置时，模块处于最低功率状态。

ADC 可以在任何软件可选通道上执行模数转换。选定的通道电压通过连续的近似算法转换为 11 位数字结果。在 8 位模式下，选定的通道电压是通过连续的近似算法转换为 9 位数字结果。

转换完成后，结果被放置在数据寄存器（ADCRH 和 ADCRL）中。在 10 位模式下，结果四舍五入到 10 位，并放置在 ADCRH 和 ADCRL 中。在 8 位模式下，结果四舍五入到 8 位并放置在 ADCRL 中。转换 c 然后设置 complete 标志（COCO），如果启用了转换完全中断，则生成中断（AIEN = 1）。

ADC 模块可以自动将转换结果与其比较寄存器的内容进行比较。设置 ACFE 位以启用比较功能，并与任何转换模式和配置一起操作。

10.4.1 时钟选择和分割控制

可以选择四个时钟源之一作为 ADC 模块的时钟源。然后将该时钟源除以可配置的值，以生成转换器（ADCK）的输入时钟。时钟是通过 ADICLK 位从以下来源之一选择的。

- 总线时钟，等于软件执行的频率。这是重置后的默认选择。
- 公交车时钟除以 2。对于更高的总线时钟速率，这允许将总线时钟的最大除以 16。
- ALTCLK，为本 MCU 定义（见模块部分简介）。
- 异步时钟（ADACK）-此时钟由 ADC 模块中的时钟源生成。当选择时钟源时，当 MCU 处于等待或停止模式时，该时钟保持活动状态，并允许在这些模式下进行转换以降低噪声操作。

所选时钟的频率必须在为 ADCK 指定的范围内。如果可用时钟太慢，ADC 将无法按照规格运行。如果可用的时钟太快，则必须将时钟划分为适当的频率。此分频器由 ADIV 位指定，可以除以 1、2、4 或 8。

10.4.2 输入选择和引脚控制

引脚控制寄存器（APCTL3、APCTL2 和 APCTL1）禁用用作模拟输入的引脚的 I/O 端口控制。当设置引脚控制寄存器位时，将强制满足以下条件 MCU 引脚：

- 输出缓冲区被迫处于高阻抗状态。
- 输入缓冲区被禁用。I/O 端口的读取为任何禁用输入缓冲区的引脚返回零。
- 引体向上被禁用。

10.4.3 硬件触发器

当设置 ADTRG 位时，ADC 模块启用 ADHWT，这是一个可选的异步硬件转换触发器。并非所有 MCU 都提供此源。有关特定于此 MCU 的 ADHWT 源的信息，请参阅模块简介。

当 ADHWT 源可用且硬件触发器启用（ADTRG=1）时，转换在 ADHWT 的上升边缘启动。如果发生上升边缘时正在进行转换，则忽略上升边缘。在连续转换配置中，只有观察到启动连续转换的初始上升边缘。硬件触发功能与任何转换模式和配置一起运行。

10.4.4 转换控制

转换可以在 10 位模式或 8 位模式下执行，由 MODE 位决定。转换可以通过软件或硬件触发器启动。此外，ADC 模块可以配置为低功耗运行，长采样时间，持续 Uous 转换，并将转换结果自动比较到软件确定的比较值。

10.4.4.1 启动转换

启动转换：

- 如果选择软件触发操作，则写入 ADCSC1（ADCH 位不是全部 1s）。
- 如果选择了硬件触发操作，则发生硬件触发（ADHWT）事件。
- 启用连续转换时，将结果传输到数据寄存器。

如果启用了连续转换，则在当前转换完成后自动启动新的转换。在软件触发操作中，ADCSC1 写入后开始连续转换，并一直持续到中止。在硬件中触发操作，连续转换在硬件触发事件后开始，并持续到中止。

10.4.4.2 完成转换

当结果传输到数据结果寄存器、ADCRH 和 ADCRL 时，转换完成。COCO 的设置。如果设置 COCO 时 AIEN 很高，则会生成中断。

如果以前的数据在 10 位模式下正在读取（ADCRH 寄存器已被读取；ADCRL 寄存器尚未读取），则阻塞机制可防止新结果覆盖 ADCRH 和 ADCRL 中的先前数据。当阻止处于活动状态时，数据传输被阻止，COCO 未设置，新结果丢失。在启用比较功能且比较条件为假的单次转换的情况下，阻塞没有效果，ADC 操作终止。在所有其他情况下，当一个数据传输被阻止，无论 ADCO 的状态如何，另一个转换都会启动（启用单次或连续转换）。

如果启用单个转换，阻塞机制可能会导致多次丢弃转换和超额功耗。为了避免这种情况，在转换完成之前，在启动单次转换后，不得读取数据寄存器。

10.4.4.3 中止转换

在以下情况中，任何正在进行的转换都会中止：

- 发生对 ADCSC1 的写入（如果 ADCH 不是全部 1s，则当前转换被中止，并启动新的转换）。
- 发生对 ADCSC2、ADCCFG、ADCCVH 或 ADCCVL 的写入。这表明操作模式发生了变化，因此当前转换无效。
- MCU 已重置。
- MCU 在未启用 ADACK 的情况下进入停止模式。

当转换中止时，ADCRH 和 ADCRL 的内容，数据寄存器不会被更改，而是继续是上次成功转换完成后传输的值。如果重置中止转换，ADCRH 和 ADCRL 将返回到重置状态。

10.4.4.4 电源控制

ADC 模块保持空闲状态，直到启动转换。如果选择 ADACK 作为转换时钟源，则也启用了 ADACK 时钟生成器。

设置 ADLPC 可以降低功耗。导致 f 的最大值更低 f_{ADCK} （见电气规格）。

10.4.4.5 总转换时间

总转换时间取决于采样时间（由 ADLSMP 确定）、MCU 总线频率、转换模式（8 位或 10 位）和转换时钟的频率（第六个罗马字母 f_{ADCK} ）。模块激活后，输入的采样开始。ADLSMP 用于在短采样时间和长采样时间之间进行选择。采样完成后，转换器与输入通道隔离，并执行连续的近似算法 O 确定模拟信号的数字值。转换算法完成后，转换结果会传输到 ADCRH 和 ADCRL。

- 如果总线频率小于 f_{ADCK} 启用短采样时，无法保证频率、连续转换的精确采样时间（ADLSMP=0）。
- 如果总线频率小于 f 的 $1/11 f_{ADCK}$ 当启用长采样时，无法保证连续转换的精确采样时间（ADLSMP=1）。

不同条件的最大总转换时间汇总于表 10-12。

表 10-12. 总转换时间与控制条件

转换类型	ADICLK	ADLSMP	最大总转换时间
单位或第一个连续 8 位	0x, 10	0	20 个 ADCK 周期+5 个总线时钟周期
单位或第一个连续 10 位	0x, 10	0	23 个 ADCK 周期+5 个总线时钟周期
单位或第一个连续 8 位	0x, 10	1	40 个 ADCK 周期+5 个总线时钟周期
单位或第一个连续 10 位	0x, 10	1	43 个 ADCK 周期+5 个总线时钟周期
单位或第一个连续 8 位	11	0	5MS + 20 ADCK + 5 总线时钟周期
单位或第一个连续 10 位	11	0	5MS + 23 ADCK + 5 总线时钟周期
单位或第一个连续 8 位	11	1	5MS + 40 ADCK + 5 总线时钟周期
单位或第一个连续 10 位	11	1	5MS + 43 ADCK + 5 总线时钟周期
后续连续 8 位; 第六个罗马字母公共汽车 \geq 第六个罗马字母 ADCK	X 染色体	0	17 个 ADCK 周期
后续连续 10 位; 第六个罗马字母公共汽车 \geq 第六个罗马字母 ADCK	X 染色体	0	20 个 ADCK 周期
后续连续 8 位; 第六个罗马字母公共汽车 \geq 第六个罗马字母 ADCK/11	X 染色体	1	37 个 ADCK 周期
后续连续 10 位; 第六个罗马字母公共汽车 \geq 第六个罗马字母 ADCK/11	X 染色体	1	40 个 ADCK 周期

选择的时钟源和除法比决定了最大总转换时间。时钟源可由 ADICK 位选择，除法比由 ADIV 位指定。

例如，您可以使用以下等式转换时间在 10 位模式下进行单次转换：

$$\text{转换时间} = \frac{23 \text{ ADCK cyc}}{3.5\text{MS } 8 \text{ MHz}/1} + \frac{5 \text{ 公交车 cyc}}{8 \text{ 兆赫}} =$$

$$\text{总线周期数} = 3.5\text{MS} \times 8 \text{ MHz} = 28 \text{ 个周期}$$

在哪里冒号：

- 形式：10 位
- 输入时钟源：公交车时钟
- 输入时钟比：除以 1
- 总线频率：8 兆赫
- 转换时间：3.5M 罗马字母的第十九个

笔记

ADCK 频率必须在 f_{ADCK} 最小值和 f_{ADCK} 最大限度地满足 ADC 规格。

10.4.5 自动比较功能

比较功能可以配置为检查上限或下限。输入采样和转换后，结果被添加到两者的比较值（ADCCVH 和 ADCCVL）的补集中。当与上限（ACFGT = 1）进行比较时，如果结果大于或等于比较值，设置 COCO。当与下限（ACFGT = 0）进行比较时，如果结果小于比较值，则设置 COCO。通过添加转换结果和两者的完全而产生的值比较值的 nt 被转移到 ADCRH 和 ADCRL。

在启用比较功能且比较条件不为真的转换后，未设置 COCO，也没有将数据传输到结果寄存器。如果启用了 ADC 中断（AIEN = 1），则在设置 COCO 时生成 ADC 中断。

笔记

当 MCU 处于等待或停止模式时，比较功能可用于监控通道上的电压。
当满足比较条件时，ADC 中断会唤醒 MCU。

10.4.6 MCU 等待模式操作

WAIT 指令将 MCU 置于低功耗待机模式。恢复速度非常快，因为时钟源保持活动状态。如果 MCU 进入等待模式时正在进行转换，则会一直持续到完成。转换可以 init 当 MCU 通过硬件触发器或启用连续转换时处于等待模式时。

在等待模式下，总线时钟、总线时钟除以二和 ADACK 可以作为转换时钟源。使用 ALTCLK 作为等待的转换时钟源取决于该 MCU 的 ALTCLK 的定义。查阅模块简介特定于此的 ALTCLK 信息 MCU。

如果启用了 ADC 中断，转换完整事件将设置 COCO 并生成 ADC 中断，以将 MCU 从等待模式唤醒（AIEN = 1）。

10.4.7 MCU 停止模式操作

STOP 指令使 MCU 处于低功耗待机模式，在此期间，MCU 上的大多数或所有时钟源都被禁用。

10.4.7.1 禁用 ADACK 的停止模式

如果异步时钟 ADACK 不是转换时钟，执行 STOP 指令将中止当前转换，并将 ADC 置于空闲状态。ADCRH 和 ADCRL 的内容不受停止模式的影响。退出停止模式后，软需要软件或硬件触发器才能恢复转换。

10.4.7.2 启用 ADACK 的停止模式

如果 ADACK 是转换时钟，则 ADC 在停止模式下继续运行。为了保证 ADC 运行，MCU 的电压调节器必须在停止模式下保持活动状态。有关此 MCU 的配置信息，请参阅模块简介。

如果 MCU 进入停止模式时正在进行转换，它将持续到完成。当 MCU 处于停止模式时，可以通过硬件触发器或启用连续转换启动转换。

如果启用了 ADC 中断，转换完整事件将设置 COCO 并生成 ADC 中断，以将 MCU 从停止模式唤醒（AIEN = 1）。

笔记

ADC 模块将系统从低功率停止中唤醒，并使 MCU 开始消耗运行电平电流，而不会产生系统电平中断。为了防止这种情况，您的应用程序必须确保数据传输阻止机制（讨论在 [第 10.4.4.2 节“完成转换”](#)）在进入停止和继续 ADC 转换时被清除。

10.5 初始化信息

本节给出了如何初始化和配置 ADC 模块的基本指导。在其他选项中，您可以为以下内容配置该模块：

- 8 位或 10 位分辨率
- 单次或连续转换
- 轮询或中断方法

参考表 10-6, 表 10-7, 和表 10-8 有关以下示例中使用的信息。

笔记

由前面的 0x 指定的十六进制值, 由前面的%指定的二进制值, 以及十进制值没有前面的字符。

10.5.1 ADC 模块初始化示例

在 ADC 模块完成转换之前, 必须执行初始化。

10.5.1.1 初始化序列

典型的初始化序列如下:

1. 更新配置寄存器 (ADCCFG) 以选择输入时钟源和用于生成内部时钟 ADCK 的除法比。也用于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) 以选择转换触发器 (硬件或软件), 并比较功能选项 (如果启用)。
3. 更新状态和控制寄存器 1 (ADCSC1) 以选择连续或仅一次的转换, 并启用或禁用转换完整中断。此外, 使用此寄存器选择要执行转换的输入通道。

10.5.1.2 伪代码示例

在本例中, ADC 模块配置了中断, 可以在输入通道 1 上以较长的采样时间在低功耗下执行单个 10 位转换, 其中内部 ADCK 时钟从总线时钟除以 1 派生。

ADCCFG = 0x98 (%10011000)

位 7	ADLPC	1	配置低功耗 (降低最大时钟速度)
位 6:5	阿迪夫	00	将 ADCK 设置为输入时钟 ÷ 1
位 4	ADLSMP	1	配置较长的采样时间
位 3:2	形式	10	在 10 位转换时设置模式
位 1:0	ADICLK	00	选择总线时钟作为输入时钟源

ADCSC2 = 0x00 (%00000000)

位 7	ADACT	0	指示转换是否正在进行中
位 6	ADTRG	0	选择软件触发器
比特 5	ACFE	0	禁用比较功能
位 4	ACFGT	0	本例中未使用
位 3:2		00	未实现或保留, 总是读为零
位 1:0		00	保留给 Freescale 内部使用; 始终写零

ADCSC1 = 0x41 (%01000001)

位 7	可可	0	指示转换何时完成
位 6	AIEN	1	启用转换完全中断
比特 5	ADCO	0	仅指定一个转换（禁用连续转换）
位 4:0	ADCH	00001	选择输入通道 1 作为 ADC 输入通道

ADCRH/L = 0xxx

持有转换结果。在低字节（ADCRL）之前读取高字节（ADCRH），以便转换数据不能被下一次转换的数据覆盖。

ADCCVH/L = 0xxx

启用比较功能时保持比较值

APCTL1=0x02

AD1 引脚 I/O 控制被禁用。所有其他 AD 引脚仍然是通用 I/O 引脚

APCTL2=0x00

所有其他 AD 引脚仍然是通用 I/O 引脚

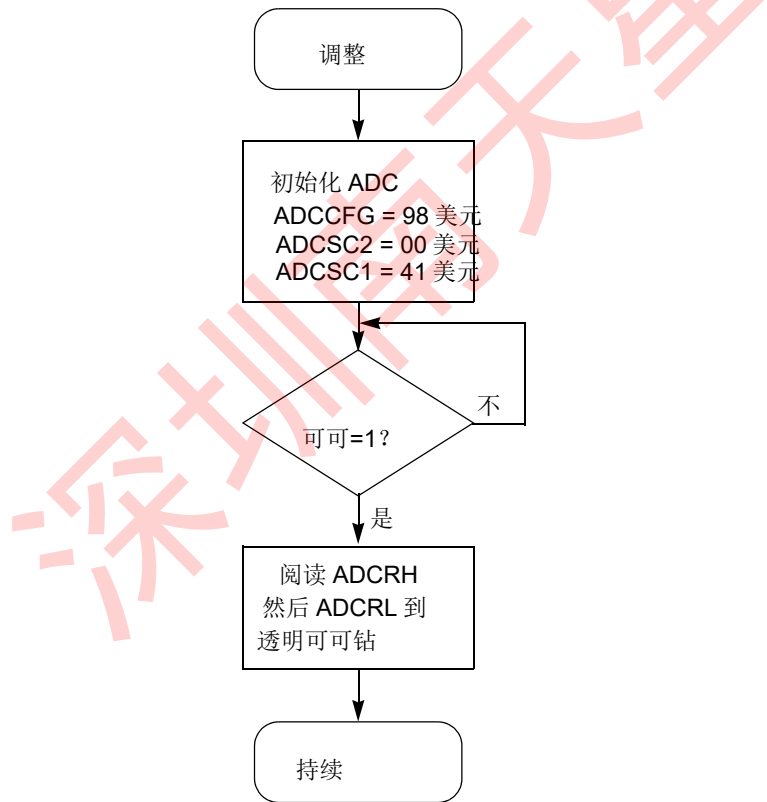


图 10-14。初始化流程图示例

10.6 应用信息

本节包含在应用程序中使用 ADC 模块的信息。ADC 被设计为集成到微控制器中，用于需要 A/D 转换器的嵌入式控制应用。

10.6.1 外部引脚和路由

以下各节讨论了与 ADC 模块相关的外部引脚以及如何使用它们以获得最佳效果。

10.6.1.1 模拟供应引脚

ADC 模块具有模拟电源和接地电源 (V_{DDAD} 和 V_{SSAD}) 在某些设备上作为单独的引脚可用。在其他设备上, V_{SSAD} 与 MCU 数字 V 在同一个大头针上共享;在其他人身, 两者都是 V_{SSAD} 和 V_{DDAD} 与 MCU 数字供应引脚共享。在这些情况下, 模拟电源的单独垫与相应的数字电源粘合在相同的引脚上, 因此在电源之间保持一定程度的隔离。

当在单独的引脚上可用时, V_{DDAD} 和 V_{SSAD} 必须连接到与其相应的 MCU 数字电源相同的电压电位 (V_{DD} 和 V_{SS}) 并且必须小心布线, 以最大限度地提高噪音抗性, 并尽可能靠近包装的旁路电容器。

在模拟和数字电源使用单独电源的情况下, 这些电源之间的接地连接必须在 V_{SSAD} 别针。通常, 如果可能的话, 这必须是这些供应之间的唯一地面连接。 V_{SSAD} 别针是一个很好的单点地面位置。

10.6.1.2 模拟参考引脚

除了模拟电源外, ADC 模块还有两个参考电压输入的连接。高参考是 V_{REFH} , 可以与 V_{DDAD} 在同一个引脚上共享。低参考是 V_{REFL} , 可以与 V_{SSAD} 在某个设备上。

当在单独的引脚上可用时, V_{REFH} 可以连接到与 V_{DDAD} 相同的电位, 或者可以由最小 V_{DDAD} 之间的外部源驱动 V_{DDAD} 规格和 V_{DDAD} 潜力 (V_{REFH} 永远不能超过 V_{DDAD})。当在单独的引脚上可用时, V_{REFL} 必须连接到与 V_{SSAD} 相同的电压电位。两个 V_{REFH} 和 V_{REFL} 必须小心布线, 以获得最大的抗噪能力, 并尽可能靠近包装的旁路电容器。

在每个连续的近似步骤中, 向电容器阵列供电所需的电流尖峰形式的交流电流通过 V_{REFH} 和 V_{REFL} 循环。满足当前需求的最佳外部组件是 0.1M 具有良好高频特性的 F 电容器。这个电容器在 V_{REFH} 和 V_{REFL} 之间连接并且必须尽可能靠近包装销。不建议使用路径中的电阻, 因为电流会导致电压下降, 这可能会导致转换错误。此路径中的电感必须最小 (仅限寄生)。

10.6.1.3 模拟输入引脚

外部模拟输入通常与 MCU 设备上的数字 I/O 引脚共享。通过在其中一个引脚控制寄存器中设置适当的控制位来禁用引脚 I/O 控制。可以在没有相关引脚控制寄存器位集的情况下对输入进行转换。当使用引脚作为模拟输入时，建议始终设置引脚控制寄存器位。这避免了争用问题，因为输出缓冲区将处于高阻抗状态，并且上拉是禁用的 E_d 。此外，当输入不在任何 V 时，输入缓冲区会吸引直流电流 I_{in} 或 V_{in} 。必须为用作模拟输入的所有引脚设置引脚控制寄存器位，才能实现最低工作电流。

经验数据显示，在存在噪声或源阻抗高时，模拟输入上的电容器可以提高性能。使用 0.01M 具有良好高频特性的 F 电容器就足够了。并非所有情况下都需要这些电容器，但使用时，它们必须尽可能靠近包装引脚，并参考 V_{SSA} 。

为了正确转换，输入电压必须在 V_{REFH} 和 V_{REFL} 之间。如果输入等于或超过 V_{REFH} ，转换器电路将信号转换为 \$3FF（全尺度 10 位表示）或 \$FF（全尺度 8 位表示）。如果输入等于或小于 V_{REFL} ，转换器电路将其转换为 000 美元。 V_{REFH} 和 V_{REFL} 之间的输入电压是直线线性转换。将有一个与 V 相关的简短电流 I_{REFL} 当采样电容器充电时。当 $ADLSMP$ 低时，输入采样为 $ADCK$ 源的 3.5 个周期，当 $ADLSMP$ 高时为 23.5 个周期。

为了最大限度地减少电流注入造成的精度损失，在转换过程中，与模拟输入引脚相邻的引脚不得转换。

10.6.2 错误来源

A/D 转换有几个错误来源。以下各节将讨论这些内容。

10.6.2.1 采样错误

为了进行适当的转换，输入的采样时间必须足够长，才能达到适当的准确性。给定最大输入电阻约为 $7k\Omega$ 输入电容约为 $5.5 pF$ ，采样到 $1/4$ 以内 LSB（在 10 位分辨率下）可以在最小采样窗口（3.5 周期在 8 MHz 最大 $ADCK$ 频率）内实现，前提是外部模拟源的电阻（ R_{in} ）低于 $5k\Omega$ 。

通过设置 $ADLSMP$ （将样品窗口增加到 23.5 个周期）或降低 $ADCK$ 频率以增加样品时间，可以实现更高的源电阻或更高精度的采样。

10.6.2.2 引脚泄漏错误

I/O 引脚的泄漏可能导致转换错误，如果外部模拟源电阻（ R_{in} ）很高。如果应用程序不能容忍此错误，请保留 R_{in} 低于 $V_{DDAD}/(2^{第十四个英文字母} \times I_{leak})$ 小于 $1/4$ LSB 泄漏错误（8 位模式下的 $N = 8$ 或 10 位模式下的 10）。

10.6.2.3 噪声诱导的错误

样品或转换过程中的系统噪音会影响转换精度。仅当满足以下条件时，ADC 精度数字才得到保证：

- 有一个 0.1M 来自 V 的低 ESR 电容器 V_{REFH} 到 V_{REFL} 。
- 有一个 0.1M 来自 V 的低 ESR 电容器 V_{DDAD} 到 V_{SSAD} 。
- 如果从初级供应中使用电感隔离，则额外的 1MF 电容器从 V 放置 V_{DDAD} 到 V_{SSAD} 。
- V_{SSAD} （和 V_{REFL} ，如果已连接）已连接到 V 纳粹党卫军在地面平面的一个安静点。
- 在启动（硬件触发转换）或启动（硬件或软件触发转换）ADC 转换后立即以等待或停止模式操作 MCU。
 - 对于软件触发的转换，请立即使用 WAIT 指令或 STOP 指令写入 ADCSC1。
 - 对于停止模式操作，请选择 ADACK 作为时钟源。停止操作减少 $V_{\text{女儿}}$ 噪音，但由于停止恢复而增加了有效转换时间。
- 在转换过程中，MCU 上没有 I/O 切换、输入或输出。

在某些情况下，外部系统活动导致辐射或传导噪声排放或过度 $V_{\text{女儿}}$ 噪声耦合到 ADC 中。在这些情况下，或者当 MCU 无法等待或停止或 I/O 活动无法停止时，请尝试以下建议的操作来减少噪声对精度的影响：

- 放置一个 0.01MF 电容器 ($C_{\text{砷}}$) 在选定的输入通道上到 V_{REFL} 或 V_{SSAD} （这改善了噪音问题，但可能会根据外部模拟源电阻影响采样率）。
- 通过连续多次转换模拟输入并除以结果之和来平均结果。需要四个样本来消除 1 的影响 LSB，一次性错误。
- 通过关闭异步时钟（ADACK）和平均来减少同步噪声的影响。与 ADCK 同步的噪音无法平均。

10.6.2.4 代码宽度和量化错误

ADC 将理想的直线传输函数量化为 1024 步（在 10 位模式下）。理想情况下，每个步骤都有相同的高度（1 个代码）和宽度。宽度被定义为过渡点到一个代码和下一个代码之间的增量。理想的代码 N 位转换器的宽度（在这种情况下，N 可以是 8 或 10），定义为 1LSB，是：

$$1\text{LSB} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2 \text{ 第十四个英文字母} \quad \text{Eqn. 10-1}$$

由于结果的数字化，存在固有的量化错误。对于 8 位或 10 位转换，当电压位于直线传输函数由实际传输函数精确表示的点之间的中点时，代码会过渡。因此，量化误差将是 $\pm 1/2\text{LSB}$ 在 8 位或

10 位模式下。因此，第一个（000 美元）转换的代码宽度只有 $1/2\text{LSB}$ 最后一个（FF 或 3FF）的代码宽度为 1.5LSB 。

10.6.2.5 线性误差

ADC 也可能表现出几种形式的非线性。已尽一切努力减少这些错误，但应用程序必须意识到它们，因为它们会影响整体准确性。这些错误是：

- 零度误差 (E_{ZS})（有时称为偏移）—第一次转换的实际代码宽度与理想代码宽度之间的差异 ($1/2\text{LSB}$)。注意，如果第一次转换是 001 美元，那么实际的 001 美元代码宽度与其理想之间的差异 (1LSB) 被使用。
- 全面误差 (E_{FS})—上次转换的实际代码宽度与理想代码宽度 (1.5LSB)。注意，如果最后一次转换是 3FE，那么实际 3FE 代码宽度与其理想之间的差异 (1LSB) 被使用。
- 差分非线性 (DNL)—所有转换的实际代码宽度和理想代码宽度之间的最坏情况差异。
- 积分非线性 (INL)—DNL 实现的运行和的最高值（绝对值）。更简单地说，这是所有代码的实际过渡电压与给定代码及其相应的理想过渡电压的最坏情况差异。
- 总未调整误差 (TUE)—实际传递函数和理想直线传递函数之间的差异，因此包括所有形式的误差。

10.6.2.6 代码抖动、非单声道和缺少代码

模拟数字转换器容易受到三种特殊形式的错误的影响；

代码抖动——是指在某些点，当重复采样时，给定的输入电压转换为两个值之一。理想情况下，当输入电压比过渡电压小得多时，转换器会产生较低的代码（反之亦然）。然而，即使是极少量的系统噪声也可能导致转换器对过渡电压周围的输入电压范围不确定（在两个代码之间）。这个范围通常在 $\pm 1/2$ 左右 LSB 并随着噪音的增加而增加。反复对输入进行采样并平均结果。此外，讨论的技术第 10.6.2.3 节减少这个错误。

非单调性——定义为除代码抖动外，转换器转换为较低的代码以获得更高的输入电压。

缺少代码——是从未为任何输入值转换的值。

在 8 位或 10 位模式下，ADC 是单调的，没有丢失的代码。

第 11 章

内部时钟源（RS08ICSOSCV1）

11.1 简单介绍

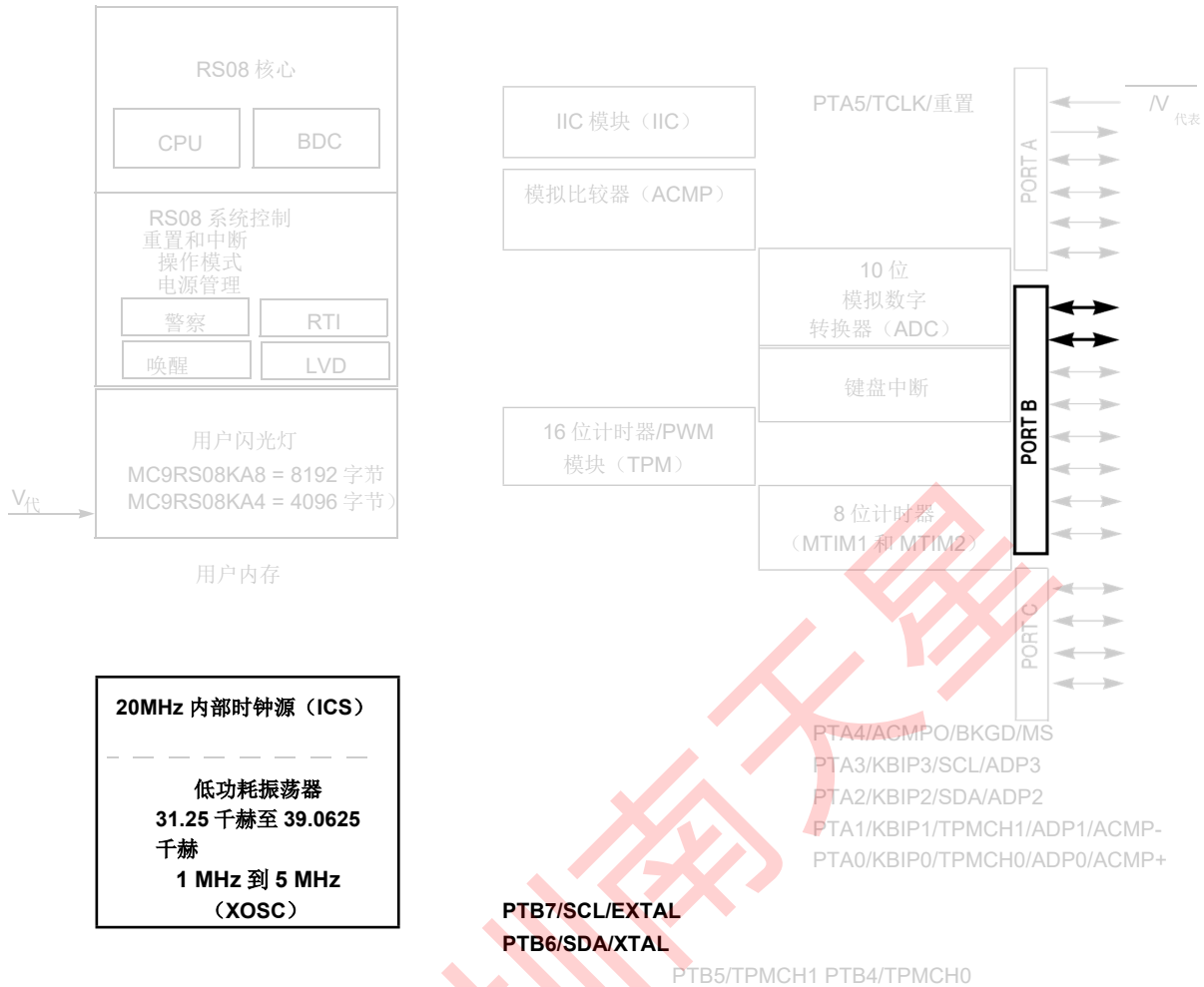
内部时钟源（ICS）模块为 MCU 提供时钟源选择。该模块包含一个频率锁定环路（FLL），作为可由内部参考时钟或外部参考时钟控制的时钟源。该模块可以提供 FLL clock、内部参考时钟或外部参考时钟作为 MCU 系统时钟（ICSOUT）的来源。

无论选择哪个时钟源，ICSOUT 都会通过总线时钟分配器（BDIV），这允许导出较低的最终输出时钟频率。ICSOUT 是总线频率的两倍。

图 11-1 显示 MC9RS08KA8 系列方框图，并突出显示 ICS。



第 11 章内部时钟源 (RS08ICSOSCV1)



PTB3/KBIP7/ADP7
 PTB2/KBIP6/ADP6
 PTB1/KBIP5/ADP5
 PTB0/KBIP4/ADP4

PTC3/ADP11

(MC9RS08KA8 = 254 字节)
 (MC9RS08KA4 = 126 字节) PTC2/ADP10

PTC1/ADP9
 PTC0/ADP8



图 11-1。MC9RS08KA8 系列方框图突出显示 ICS 方块和引脚

深圳南天星

11.1.1 特点

ICS 模块的主要特点是：

- 频率锁定环路（FLL）可修剪，以保证准确性
 - 使用内部 32 kHz 参考的 0.2%分辨率
 - 使用内部 32 kHz 参考，电压和温度偏差为 2%
- 高达 5 MHz 的内部或外部参考时钟可用于控制 FLL
 - 提供 3 位选择参考分频器
- 内部参考时钟有 9 个修剪位可用
- 可以选择内部或外部参考时钟作为 MCU 的时钟源
- 无论选择哪个时钟作为源，都可以向下划分——为时钟分配器提供 2 位选择
 - 允许的分隔线是：1、2、4、8
- 提供低功率振荡器作为外部参考时钟的控制信号
 - HGO, RANGE, EREFS, ERCLKEN, EREFSTEN
- 在重置中自动选择 FLL 接合的内部模式

11.1.2 操作模式

ICS 有七种操作模式：FEI、FEE、FBI、FBILP、FBE、FBELP 和停止。

11.1.2.1 FLL Engaged Internal (FEI)

在 FLL 参与的内部模式（默认模式）中，ICS 提供从 FLL 派生的时钟，该时钟由内部参考时钟控制。

11.1.2.2 FLL 参与外部（费用）

在 FLL 接合的外部模式下，ICS 提供来自 FLL 的时钟，该时钟由外部参考时钟控制。

11.1.2.3 FLL 绕过国际 L（联邦调查局）

在 FLL 绕过内部模式下，FLL 由内部参考时钟启用和控制，但被绕过。ICS 提供从内部参考时钟派生的时钟。

11.1.2.4 FLL 绕过国际 L 低功耗（FBILP）

在 FLL 旁路内部低功耗模式下，FLL 被禁用和旁路，ICS 提供从内部参考时钟派生的时钟。

11.1.2.5 FLL 绕过外部 I (FBE)

在 FLL 旁路外部模式下，FLL 由外部参考时钟启用和控制，但被绕过。ICS 提供从外部参考时钟派生的时钟。外部参考时钟可以提供外部晶体/谐振器由 ICS 控制的 OSC，或者它可以是另一个外部时钟源。

11.1.2.6 FLL 绕过外部 L 低功耗 (FBELP)

在 FLL 旁路外部低功耗模式下，FLL 被禁用和旁路，ICS 提供来自外部参考时钟的时钟。外部参考时钟可以由 ICS 控制的 OSC 提供的外部晶体/谐振器，o 它可以是另一个外部时钟源。

11.1.2.7 停止 (停止)

在停止模式下，FLL 被禁用，可以选择启用或禁用内部或外部参考时钟。ICS 不提供 MCU 时钟源。

11.1.3 方框图

图 11-2 是 ICS 方框图。

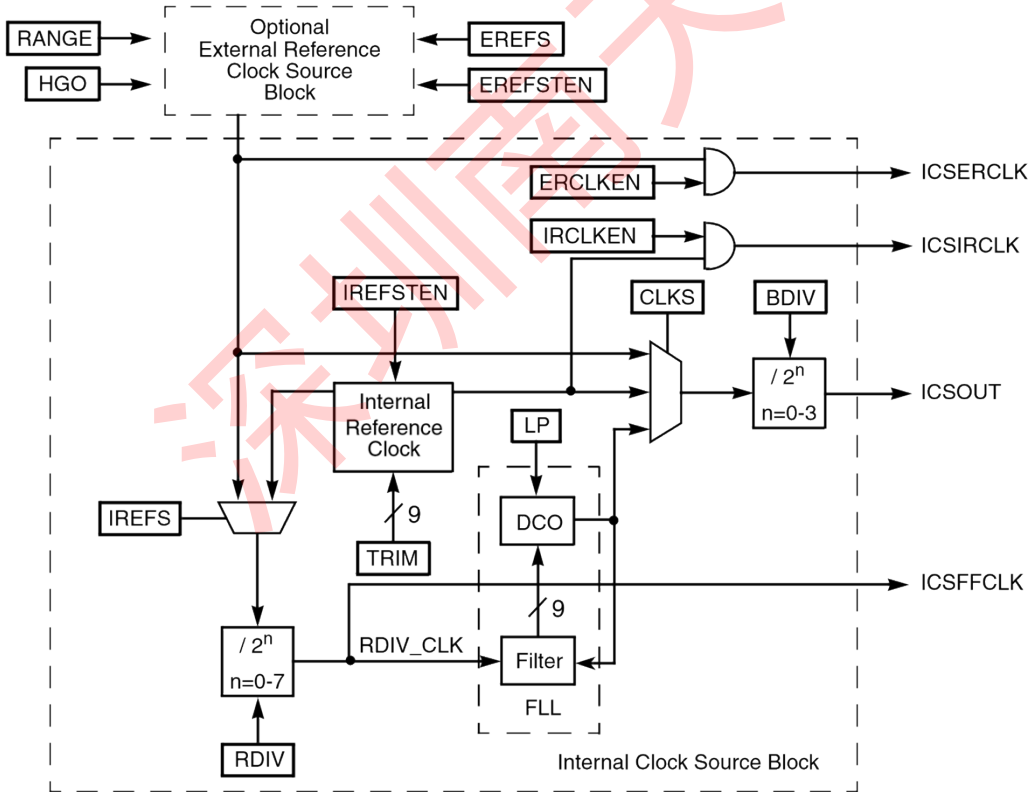


图 11-2. 内部时钟源 (ICS) 框图

11.2 外部信号描述

没有连接芯片的 ICS 信号。

11.3 注册定义

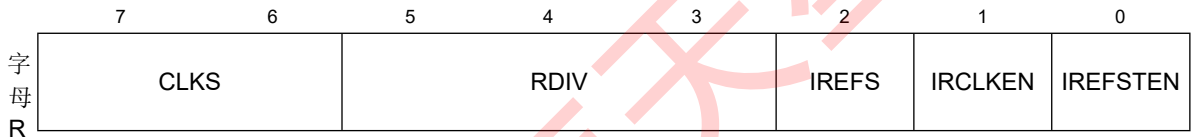
图 11-1 是 ICS 寄存器的摘要。

表 11-1。ICS 注册摘要

名字		7	6	5	4	3	2	1	0
ICSC1	字母 R	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
	罗马字母的第 23 个字母								
ICSC2	字母 R	BDIV	范围	HGO	LP	EREFS	埃尔克尔肯	埃里夫斯滕	
	罗马字母的第 23 个字母								
ICSTRM	字母 R	修剪							
	罗马字母的第								

	23 个 字 母							
ICSSC	字 母 R	0	0	0	0	CLKST	OSCINIT	FTRIM
	罗 马 字 母 的 第 23 个 字 母							

11.3.1 ICS 控制寄存器 1 (ICSC1)



罗马字母的第 23 个字母

重置: 0 0 0 0 0 1 0 0

图 11-3. ICS 控制寄存器 1 (ICSC1)

表 11-2. ICS 控制寄存器 1 字段描述

字段	描述
7:6 CLKS	时钟源选择 —选择控制总线频率的时钟源。实际总线频率取决于 BDIV 位的值。00 选择了 FLL 的输出 01 选择了内部参考时钟 10 选择了外部参考时钟 11 保留，默认为 00

5:3 RDIV	参考分隔线 —选择要除以 IREFS 位选择的 FLL 参考时钟的金额。 产生的频率必须在 31.25 kHz 至 39.0625 kHz 之间。 000 编码 0 — 将参考时钟除以 1 (重置默认值) 001 编码 1 — 将参考时钟除以 2 010 编码 2 — 将参考时钟除以 4 011 编码 3 — 将参考时钟除以 8 100 编码 4—将参考时钟除以 16 101 编码 5—将参考时钟除以 32 110 编码 6—将参考时钟除以 64 111 编码 7—将参考时钟除以 128
2 IREFS	内部参考选择 — IREFS 位为 FLL 选择参考时钟源。 0 外部参考时钟被选中 1 选择了内部参考时钟
1 IRCLKEN	内部参考时钟启用 —IRCLKEN 位使内部参考时钟能够用作 ICSIRCLK。 0 ICSIRCLK 不活跃 1 ICSIRCLK 活跃
0 IREFSTEN	内部参考停止启用 —当 ICS 进入停止模式时, IREFSTEN 位控制内部参考时钟是否保持启用状态。0 内部参考时钟在停止时被禁用 1 如果设置了 IRCLKEN, 或者 ICS 在进入停止之前处于 FEI、FBI 或 FBILP 模式, 则内部参考时钟保持启用状态

11.3.2 ICS 控制寄存器 2 (ICSC2)



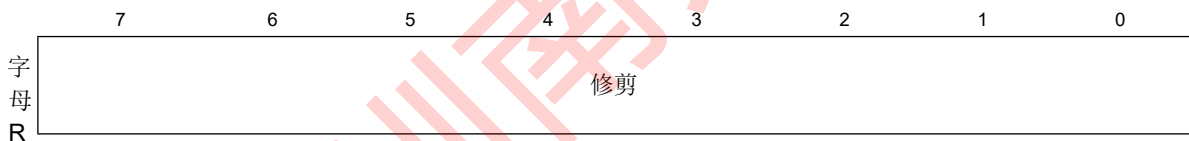
图 11-4. ICS 控制寄存器 2 (ICSC2)

表 11-3. ICS 控制寄存器 2 字段描述

字段	描述
7:6 BDIV	总线分频器 —选择按 CLKS 位选择的时钟源划分的量。这控制了总线频率。 00 编码 0—将选定的时钟除以 1 01 编码 1—将选定的时钟除以 2 (重置默认值) 10 编码 2—将选定的时钟除以 4 11 编码 3—将选定的时钟除以 8

5 范围	频率范围选择 —选择外部振荡器的频率范围。 0 为外部振荡器选择的低频范围 1 为外部振荡器选择的高频范围
4 HGO	高增益振荡器选择 —HGO 位控制外部振荡器的操作模式。 0 为低功耗操作配置外部振荡器 1 配置外部振荡器以进行高增益操作
3 LP	低功耗选择 —LP 位在 FLL 旁路模式下是否禁用 FLL。 0 FLL 在旁路模式下没有被禁用 1 FLL 在旁路模式下被禁用
2 EREFS	外部参考选择 — EREFS 位选择外部参考时钟的源。 0 请求外部时钟源 1 请求振荡器
1 ERCLKEN	外部参考启用 — ERCLKEN 位启用外部参考时钟作为 IC SERCLK 使用。 0 IC SERCLK 不活跃 1 IC SERCLK 活跃
0 埃里夫斯滕	外部引用停止启用 —当 ICS 进入停止模式时， EREFSTEN 位控制外部参考时钟是否仍然启用。0 外部参考时钟在停止时被禁用 1 如果设置了 ERCLKEN， 或者 ICS 在进入停止前处于 FEE、 FBE 或 FBELP 模式， 则外部参考时钟保持启用停止状态

11.3.3 ICS 修剪寄存器 (ICSTRM)



罗马字母的第 23 个字母

POR: 1 0 0 0 0 0 0 0
 重置: 第二十一 第二十一 第二十一 第二十一 第二十一 第二十一 第二十一 第二十一
 个罗马字 母 第二十一 个罗马字 母 第二十一 个罗马字 母 第二十一 个罗马字 母 第二十一 个罗马字 母 第二十一 个罗马字 母

图 11-5. ICS 修剪寄存器 (ICSTRM)

表 11-4. ICS 修剪寄存器字段描述

字段	描述
7:0 修剪	ICS 修剪设置 —TRIM 位通过控制内部参考时钟周期来控制内部参考时钟频率。位的效果是二进制加权的（即位 1 的调整是位 0 的两倍）。增加 TRIM 中的二进制值将增加周期，减少价值将减少周期。 ICSSC 中提供额外的精细修剪钻头作为 FTRIM 钻头。

11.3.4 ICS 状态和控制 (ICSSC)

	7	6	5	4	3	2	1	0
字母 R	0	0	0	0	CLKST		OSCINIT	FTRIM
罗								
马字母的第 23 个字母								
POR:	0	0	0	0	0	0	0	0
重置:	0	0	0	0	0	0	0	U

图 11-6. ICS 状态和控制寄存器 (ICSSC)

表 11-5. ICS 状态和控制寄存器字段描述

字段	描述
7:4	保留，必须清除。
3-2 CLKST	<p>时钟模式状态— CLKST 位表示当前时钟模式。由于时钟域之间的内部同步，CLKST 位在写入 CLKS 位后不会立即更新。</p> <p>00 选择 FLL 的输出。</p> <p>01 跳过 FLL，选择内部参考时钟。</p> <p>10 绕过 FLL，选择外部参考时钟。</p> <p>11 保留。</p>
1 OSCINIT	<p>OSC 初始化—如果外部参考时钟由 ERCLKEN 选择，或 ICS 处于 FEE、FBE 或 FBELP 模式，并且如果设置了 EREFS，则该位在外部分振荡器时钟的初始化周期完成后设置。只有当 ERCLKEN 或 EREFS 被清除时，这个位才会被清除。</p>
0 FTRIM	<p>ICS 精细修剪—FTRIM 位控制内部参考时钟频率的最小调整。设置 FTRIM 将增加期限，清算 FTRIM 将尽可能减少期限。</p>

11.4 功能描述

11.4.1 操作模式

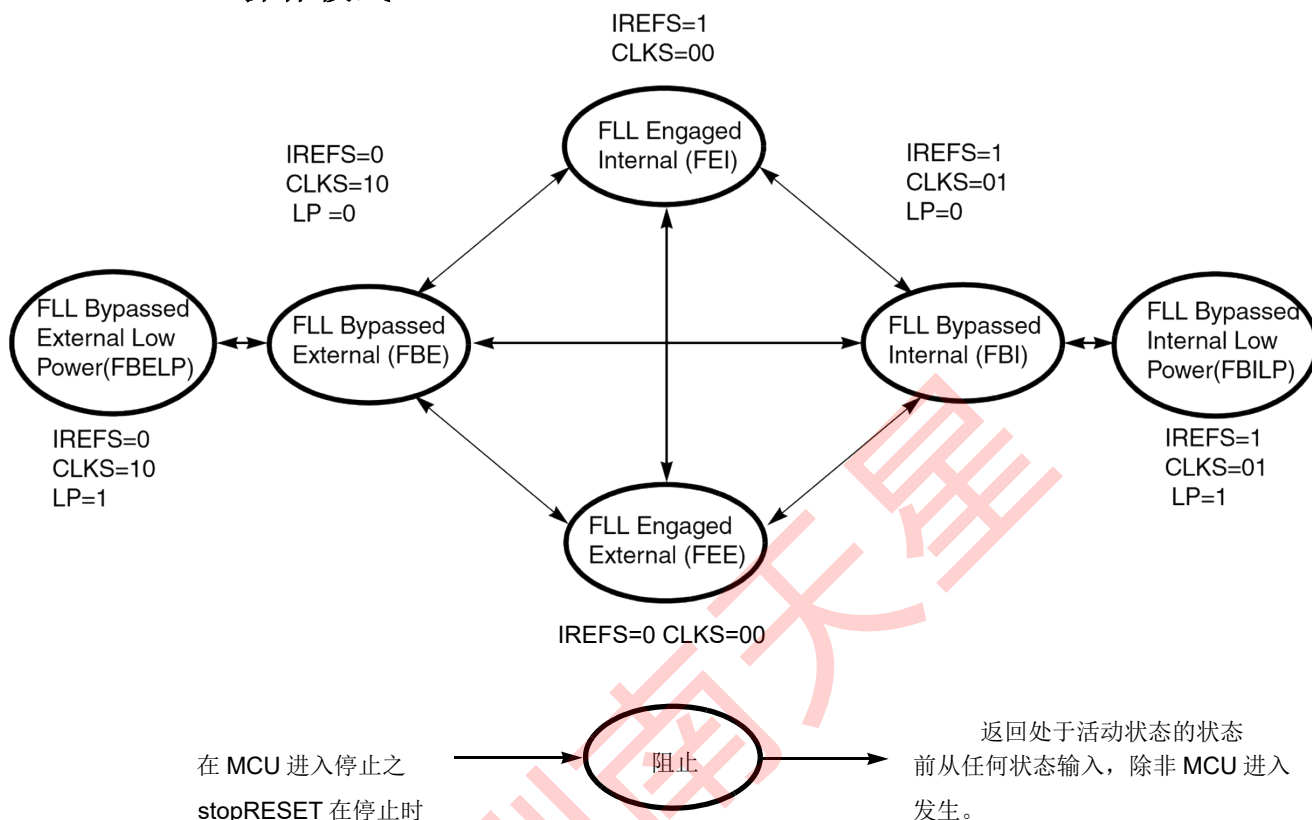


图 11-7. 时钟切换模式

ICS 的七种状态显示为状态图，描述如下。箭头表示各州之间允许的移动。

11.4.1.1 FLL 参与内部 (FEI)

FLL 参与内部 (FEI) 是默认操作模式，在出现以下所有情况时输入：

- CLKS 位写入 00 • IREFS 位写入 1
- RDIV 位被写入将参考时钟划分为 31.25 kHz 至 39.0625 kHz 的范围。

在 FLL 接合的内部模式下，ICSOUT 时钟来自 FLL 时钟，FLL 时钟由内部参考时钟控制。FLL 回路将把频率锁定到 RDIV 位选择的滤波器频率的 512 倍。内部参考时钟已启用。

11.4.1.2 FLL 参与外部 (FEE)

当出现以下所有情况时，将进入 FLL 参与外部 (FEE) 模式：

- CLKS 位写入 00
- IREFS 位写入 0
- RDIV 位被写入以将参考时钟划分为 31.25 kHz 至 39.0625 kHz 的范围

在 FLL 参与外部模式下，ICSOUT 时钟来自外部参考时钟控制的 FLL 时钟。FLL 回路将把频率锁定到 RDIV 位选择的滤波器频率的 512 倍。外部参考时钟已启用。

11.4.1.3 FLL 绕过内部 (FBI)

当发生以下所有情况时，将进入 FLL 绕过内部 (FBI) 模式：• CLKS 位写入 01

- IREFS 位写入 1。
- LP 位写入 0

在 FLL 绕过内部模式下，ICSOUT 时钟来自内部参考时钟。FLL 时钟由内部参考时钟控制，FLL 回路将锁定 FLL 频率为滤波器频率的 512 倍，由 RDIV 位选择。内部参考时钟已启用。

11.4.1.4 FLL 绕过内部低功耗 (FBILP)

当出现以下所有情况时，将进入 FLL 绕过内部低功耗 (FBILP) 模式：• CLKS 位写入 01

- IREFS 位写入 1。
- LP 位写入 1

在 FLL 绕过内部低功耗模式下，ICSOUT 时钟来自内部参考时钟，FLL 被禁用。内部参考时钟已启用。

11.4.1.5 FLL 绕过外部 (FBE)

当出现以下所有情况时，将进入 FLL 绕过外部 (FBE) 模式：

- CLKS 位写入 10。
- IREFS 位写入 0。
- LP 位写为 0。

在 FLL 绕过外部模式下，ICSOUT 时钟来自外部参考时钟。FLL 时钟由外部参考时钟控制，FLL 回路将把 FLL 频率锁定为滤波器频率的 512 倍，由 RDIV 位选择。外部参考时钟已启用。

11.4.1.6 FLL 绕过外部低功耗 (FBELP)

当出现以下所有情况时，将进入 FLL 旁路外部低功耗 (FBELP) 模式：

- CLKS 位写入 10。
- IREFS 位写入 0。
- LP 位写入 1。

在 FLL 绕过外部低功耗模式下，ICSOUT 时钟来自外部参考时钟，并禁用 FLL。外部参考时钟已启用。

11.4.1.7 阻止

每当 MCU 进入停止状态时，就会进入停止模式。在此模式下，除以下情况外，所有 ICS 时钟信号都是静态的：

当出现以下两种情况时，ICSIRCLK 将在停止模式下处于活动状态：

- IRCLKEN 位写入 1
- IREFSTEN 位写入 1

当出现以下两种情况时，ICSERCLK 将在停止模式下处于活动状态：

- ERCLKEN 位写入 1
- EREFSTEN 位写入 1

11.4.2 模式切换

当在 FLL 闕开合内部 (FEI) 和 FLL 闕开合外部 (FEE) 模式之间切换时，IREFS 位可以随时更改，但 RDIV 位必须同时更改，以便产生的频率保持在 31.25 kHz 至 39.0625 kHz 的范围内。在 IREFS 值发生变化后，FLL 将在产生的分割参考频率的几个完整周期后再次开始锁定。

CLKS 位也可以随时更改，但必须同时更改 RDIV 位，以便产生的频率保持在 31.25 kHz 至 39.0625 kHz 的范围内。直到新时钟的几个完整周期后，才会实际切换到新选择的时钟。如果新选择的时钟不可用，上一个时钟将保持选择状态。

11.4.3 总线分频器

BDIV 位可以随时更改，实际切换到新频率将立即发生。

11.4.4 低功耗位使用

提供低功耗位 (LP) 允许禁用 FLL，从而在不使用时节省电力。然而，在某些应用程序中，在切换到 FLL 接合模式之前，启用 FLL 并允许其锁定以获得最大精度可能是可取的。通过将 LP 位写入 0 来做到这一点。

11.4.5 内部参考时钟

当 IRCLKEN 设置时，内部参考时钟信号将显示为 ICSIRCLK，可以用作额外的时钟源。ICSIRCLK 频率可以通过修剪内部参考时钟的周期来重新定位。这可以通过 w 完成将新值添加到 ICSTRM 寄存器中的 TRIM 位。写入更大的值会减慢 ICSIRCLK 频率，而向 ICSTRM 寄存器写入较小的值会加快 ICSIRCLK 频率。TRIM 位将影响 ICSOUT 频率如果 ICS 处于 FLL 接合内部 (FEI)、FLL 绕过内部 (FBI) 或 FLL 绕过内部低功耗 (FBILP) 模式。TRIM 和 FTRIM 值不会受到重置的影响。

在 ICSIRCLK 被修剪之前，编程低参考分频器 (RDIV) 因素可能会导致 ICSOUT 频率超过最大芯片级频率，并违反芯片级时钟时序规范 (请参阅[设备概述](#)章节)。

如果设置了 IREFSTEN，并将 IRCLKEN 位写入 1，则 ICSIRCLK 将在停止模式下继续运行，以便在退出停止时提供快速恢复。

所有 MCU 设备都出厂编程，在保留的内存位置上有一个修剪值。在重置初始化期间，此值可以复制到 ICSTRM 寄存器。出厂装饰值不包括 FTRIM 钻头。为了获得更精细的精度，用户可以 tr 我是应用程序中的内部振荡器，并相应地设置 FTRIM 位。

11.4.6 可选的外部参考时钟

ICS 模块可以在所有模式下支持频率在 31.25 kHz 到 5 MHz 之间的外部参考时钟。当设置 ERCLKEN 时，外部参考时钟信号将显示为 ICSECLK，可以用作额外的时钟源。什么 $E_n IREFS = 1$ ，外部参考时钟不会被 FLL 使用，只会用作 ICSECLK。在这些模式下，频率可以等于芯片级定时规格支持的最大频率 (请参阅[设备概述](#)章节)。

如果设置了 EREFSTEN，并且将 ERCLKEN 位写入 1，则 ICSECLK 将在停止模式下继续运行，以便在退出停止时提供快速恢复。

11.4.7 固定频率时钟

ICS 将分割的 FLL 参考时钟呈现为 ICSFFCLK，用作外围模块的附加时钟源。ICS 提供输出信号 (ICSFFE)，指示 ICS 何时提供四倍或大于四倍的 ICSOUT 频率 e 分割 FLL 参考时钟 (ICSFFCLK)。在 FLL 参与模式 (FEI 和 FEEE) 中，这总是正确的，ICSFFE 总是很高。在 ICS 旁路模式下，ICSFFE 将断言以下 BDIV 和 RDIV 值的组合：

- $BDIV = 00$ (除以 1)， $RDIV \geq 010$

- $BDIV = 01$ (除以 2), $RDIV \geq 011$
- $BDIV = 10$ (除以 4), $RDIV \geq 100$
- $BDIV = 11$ (除以 8), $RDIV \geq 101$

深圳市南天星

第 12 章 集成电路（RS08IICV2）

12.1 简单介绍

集成电路（IIC）提供多个设备之间的通信。该接口可以运行高达 100 kbps，具有最大的总线负载和定时。该设备可以以更高的负载速率运行，最高可达时钟/20，并减少总线负载。最大总线电容为 400 pF 限制了通信长度和连接设备的数量。

笔记

SDA 和 SCL 不得驱动在 V 以上电压。这些引脚是伪开漏，包含对 V 的保护二极管。

12.1.1 模块配置

IIC 模块引脚、SDA 和 SCL 可以使用 SOPT1 中的 IICPS 在软件控制下重新定位（表 12-1）。SOPT 中的 IICPS 选择与 IIC 操作相关的通用 I/O 端口。

表 12-1. IIC 位置选项

SOPT 中的 IICPS	SDA 的端口引脚	SCL 的端口引脚
0（默认）	PTA2	PTA3
1	PTB6	PTB7

图 12-1 显示 MC9RS08KA8 方框图，其中突出显示了 IIC 方块和引脚。

第 12 章 集成电路（RS08IICV2）



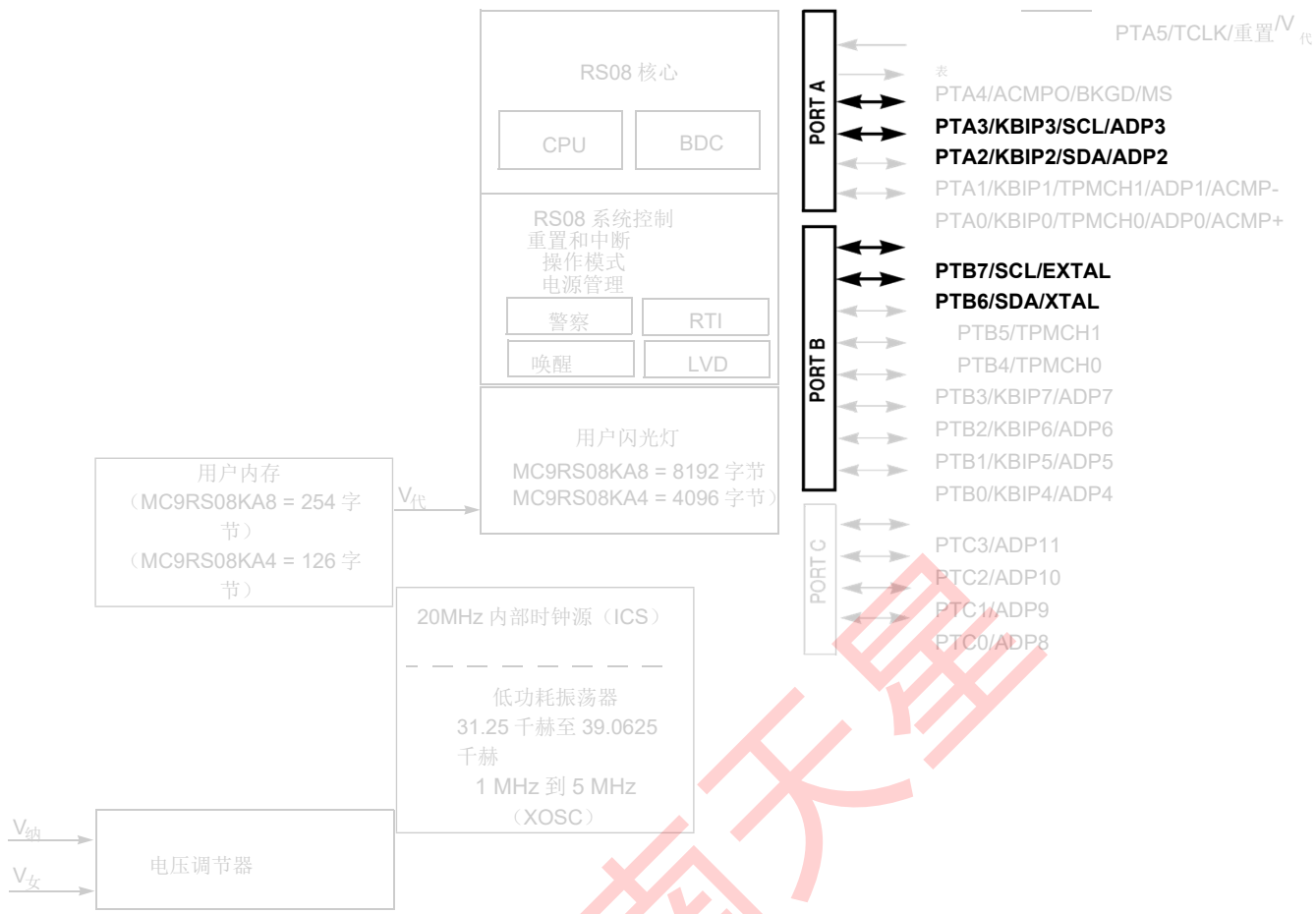


图 12-1. MC9RS08KA8 系列方块图突出显示 IIC 方块和引脚

12.1.2 特点

IIC 包括以下显著特征:

- 与 IIC 总线标准兼容
- 多主操作
- 可为 64 个不同串行时钟频率之一编程的软件
- 软件可选确认位
- 中断驱动的逐字节数据传输
- 仲裁丢失中断, 自动模式从主从切换
- 呼叫地址识别中断
- START 和 STOP 信号生成/检测
- 重复的 START 信号生成
- 确认位生成/检测
- 总线繁忙检测
- 通用呼叫识别
- 10 位地址扩展

12.1.3 操作模式

这里简要描述了各种 MCU 模式下的 IIC。

- **运行模式**—这是基本的操作模式。要在此模式下节省电力, 请禁用该模块。
- **等待模式**—当 MCU 处于等待模式时, 该模块将继续运行, 并可以提供唤醒中断。
- **停止模式**—IIC 在停止模式下处于非活动状态, 以降低功耗。STOP 指令不影响 IIC 注册状态。

12.1.4 方框图

图 12-2 是 IIC 的框图。

集成电路 (RS08IICV2)

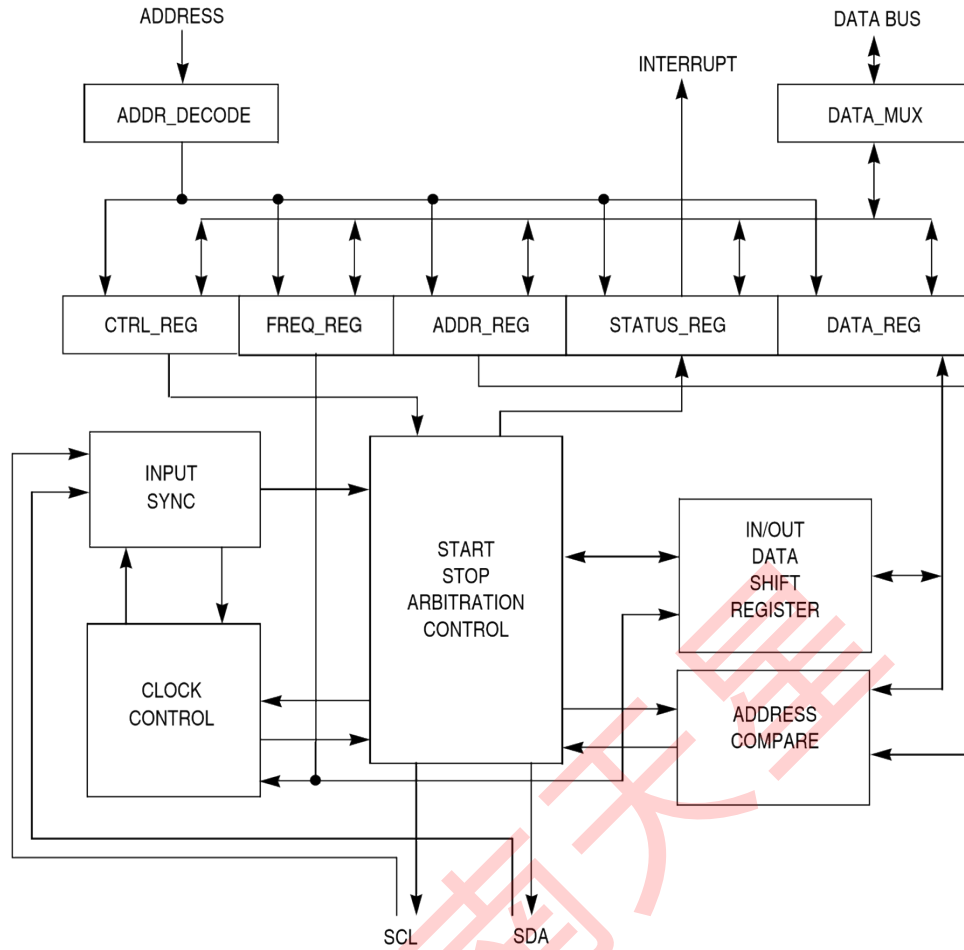


图 12-2。IIC 功能块图

12.2 外部信号描述

本节描述每个用户可访问的引脚信号。

12.2.1 SCL — 串行时钟线

双向 SCL 是 IIC 系统的串行时钟线。

12.2.2 SDA — 串行数据线

双向 SDA 是 IIC 系统的串行数据线。

12.3 注册定义

本节包括按地址顺序排列的 IIC 寄存器描述。

请参阅直接页面寄存器摘要**记忆**本数据表的章节涉及所有 IIC 寄存器的绝对地址分配。本节仅通过其名称来指寄存器和控制位。Freescale 提供的等式或头文件用于将这些名称转换为适当的绝对地址。

12.3.1 IIC 地址登记册 (IICA)

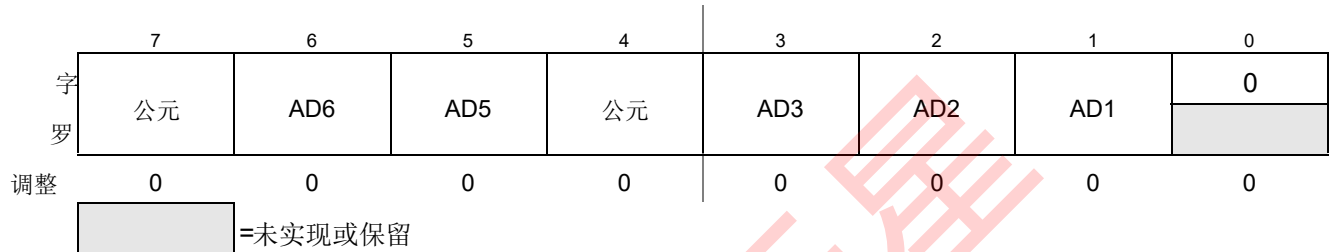
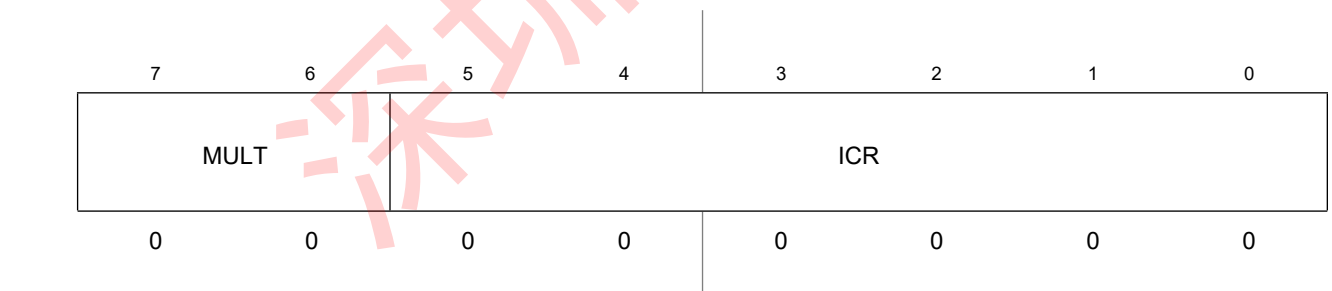


图 12-3. IIC 地址登记册 (IICA)

表 12-2. IICA 字段描述

字段	描述
7: 1 广告[7:1]	奴隶地址— AD[7:1]字段包含 IIC 模块使用的从属地址。此字段用于 7 位地址方案和 10 位地址方案的下 7 位。

12.3.2 IIC 分频器寄存器 (IICF)



字母 R

罗马字母的第 23 个字母

调整

图 12-4. IIC 分频器寄存器 (IICF)

MULT	ICR	保留时间 (MS)		
		SDA	SCL 开始	SCL 停止
0x2	0x00	3.500	4.750	5.125
0x1	0x07	2.500	4.250	5.125
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.125	4.000	5.250

表 12-3. IICF

0x0	0x18	1.125	3.000	5.500
-----	------	-------	-------	-------

字段描述

字段	描述
7:6 MULT	<p>IIC 乘数因子— MULT 位定义了乘数因子 mul。该因子与 SCL 分频器一起用于生成 IIC baud 率。下面提供了由 MULT 位定义的乘数因子 mul。</p> <p>00 Mul = 01 01 Mul = 02 10 Mul = 04 11 矜持的</p>
5:0 ICR	<p>IIC 时钟速率— ICR 位用于预缩放总线时钟以进行比特率选择。这些位和 MULT 位用于确定 IIC baud 率、SDA 保持时间、SCL 开始保持时间和 SCL 停止保持时间。表 12-4 提供 SCL 分隔符，并为 ICR 的相应值保留值。</p> <p>SCL 分频器乘以乘数因子 mul 用于生成 IIC 包率。</p> <p style="text-align: center;">IIC baud 速率=总线速度 (Hz) / (mul×SCL 分频器) Eqn. 12-1</p> <p>SDA 保留时间是从 SDA (IIC 数据) 下降边缘到 SDA (IIC 数据) 变化的延迟。</p> <p style="text-align: center;">SDA 保留时间=总线周期×穆×SDA 持有值 Eqn. 12-2</p> <p>SCL 启动保持时间是从 SDA 下降边缘 (IIC 数据) 而 SCL 高 (启动条件) 到 SCL (IIC 时钟) 下降边缘的延迟。</p> <p style="text-align: center;">SCL 开始保持时间=总线周期×穆×SCL 开始保持值 Eqn. 12-3</p> <p>SCL 停止保持时间是从 SCL 的上升边缘 (IIC 时钟) 到 SDA SDA (IIC 数据) 的上升边缘的延迟，而 SCL 是高 (停止条件)。</p> <p style="text-align: center;">SCL 停止保留时间 = 总线周期×穆×SCL 止损保留值 Eqn. 12-4</p>

例如，如果总线速度为 8 MHz，下表显示了具有不同 ICR 和 MULT 选择的可能保持时间值，以实现 100 kbps 的 IIC 包特率。

表 12-4. IIC 分频器和保持值

ICR (十六制)	SCL 分隔物	SDA 持有值	SCL 保持 (开始) 价值	SDA 保持 (停止) 价值	ICR (十六制)	SCL 分隔物	SDA 抓价值	SCL 保持 (开始) 价值	SCL 保持 (停止) 价值
00	20	7	6	11	20	160	17	78	81
01	22	7	7	12	21	192	17	94	97
02	24	8	8	13	22	224	33	110	113

03	26	8	9	14
04	28	9	10	15
05	30	9	11	16
06	34	10	13	18
07	40	10	16	21
08	28	7	10	15
09	32	7	12	17
0A	36	9	14	19
0B	40	9	16	21
0C	44	11	18	23
0D	48	11	20	25
0E	56	13	24	29
0F	68	13	30	35
10	48	9	18	25
11	56	9	22	29
12	64	13	26	33
13	72	13	30	37
14	80	17	34	41
15	88	17	38	45
16	104	21	46	53
17	128	21	58	65
18	80	9	38	41
19	96	9	46	49
1A	112	17	54	57
1B	128	17	62	65
1C	144	25	70	73
1D	160	25	78	81
1E	192	33	94	97

23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2楼	960	129	478	481
30	640	65	318	321
31	768	65	382	385
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
三维	2560	385	1278	1281
3E	3072	513	1534	1537

1 楼	240	33	118	121
-----	-----	----	-----	-----

3F	3840	513	1918	1921
----	------	-----	------	------

12.3.3 IIC 控制登记册 (IICC1)

	7	6	5	4	3	2	1	0
字 罗	IICEN	IICIE	MST	德	TXAK	0	0	0
						RSTA		
调整	0	0	0	0	0	0	0	0
	=未实现或保留							

图 12-5. IIC 控制登记册 (IICC1)

表 12-5. IICC1 字段描述

字段	描述
7 IICEN	IIC 启用 —IICEN 位决定是否启用 IIC 模块。0 IIC 未启用。 1 IIC 已启用。
6 IICIE	IIC 中断启用 —IICIE 位决定是否请求 IIC 中断。0 IIC 中断请求未启用。 1 启用 IIC 中断请求。
5 MST	主模式选择 —当总线上生成 START 信号并选择主模式时，MST 位从 0 更改为 1。当这个位从 1 变为 0 时，会生成 STOP 信号，操作模式从主变为从。 0 从属模式。 1 主模式。
4 TX	传输模式选择 —TX 位选择主从传输的方向。在主模式下，必须根据所需的传输类型设置此位。因此，对于地址周期，这个位总是很高。当作为从属处理时，这个位必须设置为由软件根据状态寄存器中的 SRW 位。0 接收。 1 传输。
3 TXAK	传输确认启用 —此位指定主接收器和从接收器的数据确认周期期间驱动到 SDA 的值。 0 收到一个数据字节后，将向总线发送确认信号。 1 没有发送确认信号响应。
2 RSTA	重复开始 —将 1 写入此位将生成重复的 START 条件，前提是它是当前主服务器。这一点将永远被解读为低。试图在错误的时间重复将导致仲裁损失。

12.3.4 IIC 状态登记册 (IICS)

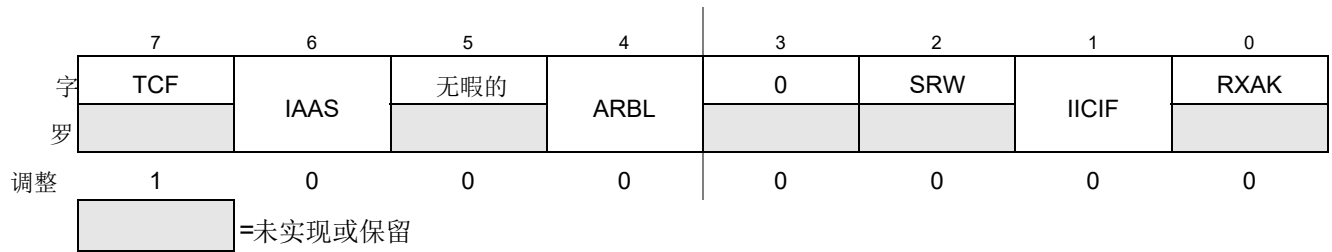


图 12-6. IIC 状态登记册 (IICS)

表 12-6. IICS 字段描述

字段	描述
7 TCF	转移完整标志 —此位设置在字节传输完成时。请注意，此位仅在传输到 IIC 模块或从 IIC 模块传输后立即有效。TCF 位通过在接收模式下读取 IICD 寄存器或在传输模式下写入 IICD 来清除。 0 转移正在进行中。 1 转移完成。
6 IAAS	作为奴隶称呼 —当调用地址与编程的从地址匹配时，或者当设置了 GCAEN 位并收到常规调用时，将设置 IAAS 位。编写 IICC 寄存器会清除这一点。 0 未解决。 1 被称为奴隶。
5 忙	公交车很忙 —BUSY 位指示总线的状态，无论从模式还是主模式。当检测到 START 信号时设置 BUSY 位，当检测到 STOP 信号时清除。 0 公交车闲置。 1 公交车很忙。
4 ARBL	仲裁丢失 —当仲裁程序丢失时，此位由硬件设置。ARBL 位必须由软件清除，向其写入 1。 0 标准总线运行。 1 仲裁的损失。
2 SRW	奴隶读/写 —当设置为从属时，SRW 位表示发送到主服务器的调用地址的 R/W 命令位的值。 0 奴隶接受，主人给奴隶写信。 1 奴隶传输，从奴隶那里主阅读。
1 IICIF	IIC 中断标志 —在中断待处理时设置 IICIF 位。这个位必须由软件清除，在中断例程中写入 1。以下事件之一可以设置 IICIF 位： <ul style="list-style-type: none"> • 一个字节传输完成 • 将从属地址与呼叫地址匹配 • 仲裁失败了 0 没有中断待处理。 1 中断待处理。

0 RXAK	接受确认 —当 RXAK 位低时，它表示在总线上完成一个字节的传输后已收到确认信号。如果 RXAK 位高，则表示没有检测到确认信号。 0 收到确认。 1 没有收到确认。
-----------	---------------------------------------------------------------------------------------------------

12.3.5 IIC 数据 I/O 寄存器 (IICD)

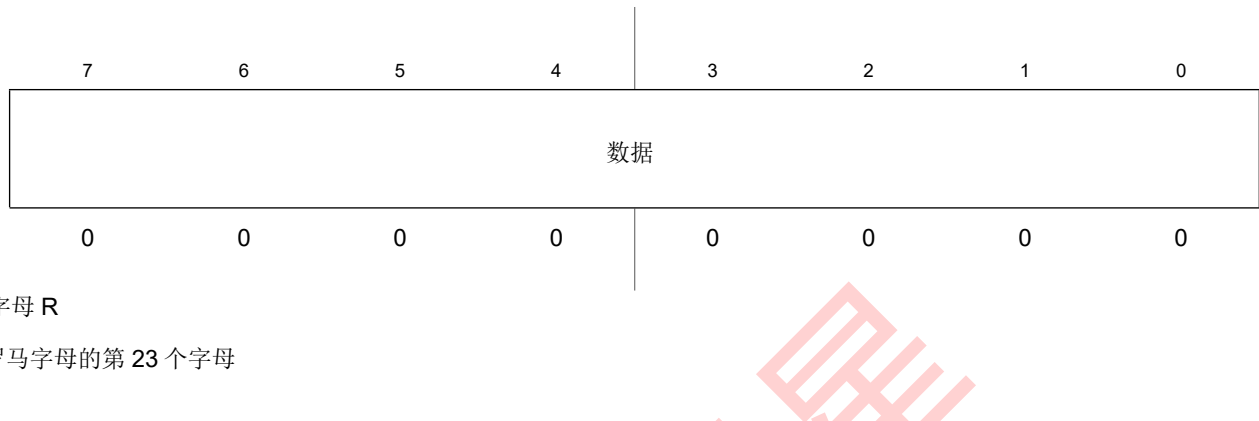


图 12-7. IIC 数据 I/O 寄存器 (IICD)

表 12-7. IICD 字段描述

字段	描述
7:0 数据	数据 —在主传输模式下，当数据写入 IICD 时，将启动数据传输。最重要的位首先发送。在主接收模式下，读取此寄存器会开始接收下一个字节的数据。

笔记

当退出主接收模式时，在读取 IICD 寄存器之前必须切换 IIC 模式，以防止无意中启动主接收数据传输。

在从属模式下，相同的功能在发生地址匹配后可用。

请注意，IICC 中的 TX 位必须正确反映主模式和从模式中所需的传输方向，才能开始传输。例如，如果 IIC 配置为主传输，但需要主接收，则读取 IICD 不会启动接收。

当 IIC 配置为主接收或从接收模式时，读取 IICD 将返回收到的最后一个字节。IICD 不能反映 IIC 总线上传输的每个字节，软件也不能验证一个字节是否已写入。通过回读来正确读取 IICD。

在主传输模式下，MST 断言后写入 IICD 的第一个数据字节用于地址传输，并且必须包括与所需连接的调用地址（以 7 位到 1 为单位）

—
R/W 位（在位置位 0 中）。

12.3.6 IIC 控制寄存器 2 (IICC2)

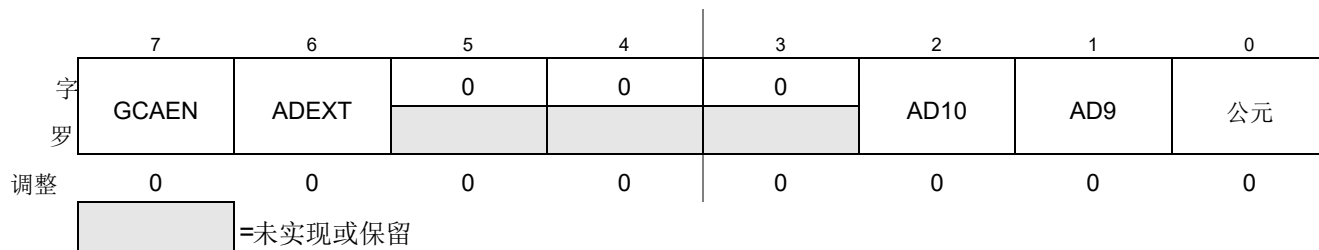


图 12-8. IIC 控制寄存器 (IICC2)

表 12-8. IICC2 字段描述

字段	描述
7 GCAEN	启用常规呼叫地址— GCAEN 位启用或禁用通用呼叫地址。0 通用呼叫地址被禁用 1 通用呼叫地址已启用。
6 ADEXT	地址扩展— ADEXT 位控制用于从属地址的位数。 0 7 位地址方案 1 10 位地址方案
2:0 AD[10:8]	奴隶地址—AD[10:8]字段包含 10 位地址方案中从属地址的上三位。此字段仅在设置 ADEXT 位时有效。

12.4 功能描述

本节提供了 IIC 模块的完整功能描述。

12.4.1 IIC 协议

IIC 总线系统使用串行数据线（SDA）和串行时钟线（SCL）进行数据传输。连接到它的所有设备都必须有开放式漏极或开放式集电极输出。使用外部上拉电阻在两条线上行使逻辑和功能。这些电阻的值取决于系统。

通常，标准通信由四个部分组成：

- 启动信号
- 从属地址传输
- 数据传输
- 停车信号灯

STOP 信号不得与 CPU STOP 指令混淆。IIC 总线系统通信在以下各节中进行了简要描述，并在图 12-9。

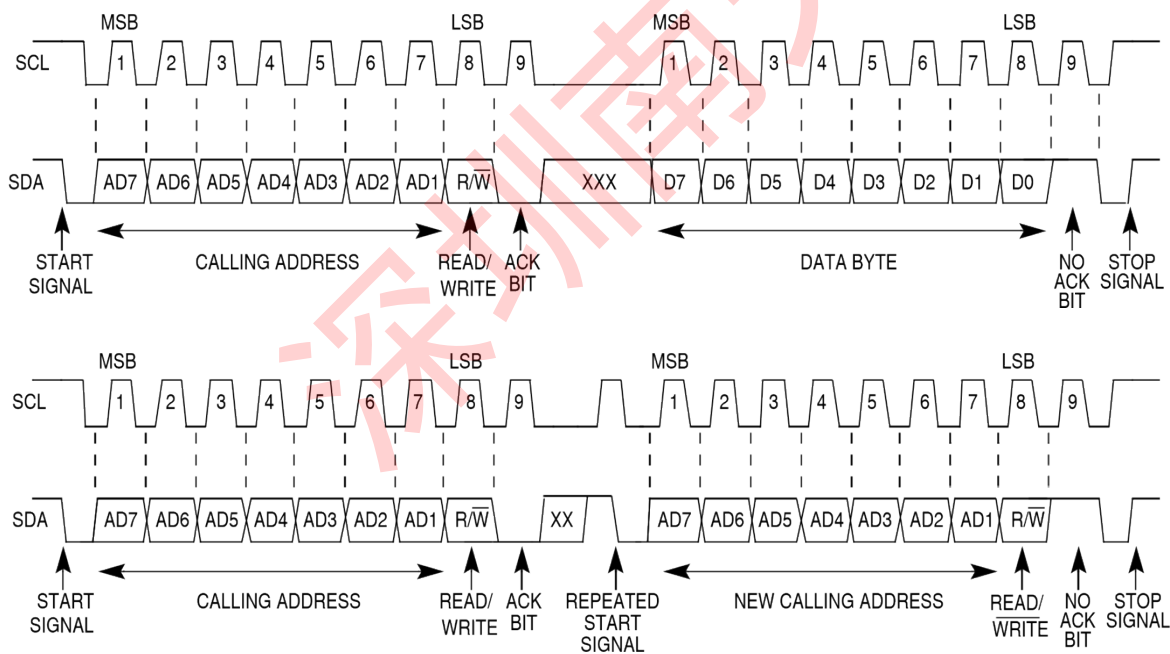


图 12-9。IIC 总线传输信号

12.4.1.1 启动信号

当总线空闲时；即没有主设备接合总线（SCL 和 SDA 线路都处于逻辑高点），主设备可以通过发送 START 信号来启动通信。如图所示图 12-9，START 信号被定义为 SDA 的高到低过渡，而 SCL 是高的。此信号表示新数据传输的开始（每个数据传输可能包含几个字节的数据），并将所有从属从器带出空闲状态。

12.4.1.2 从属地址传输

START 信号后立即传输的第一个数据字节是传输的从属地址

主人。这是一个七位调用地址，后跟一个 R/W 位。R/W 位告诉从属服务器所需的数据传输方向。

1 = 读取传输，从属服务器将数据传输到主服务器。

0 = 写入传输，主服务器将数据传输到从服务器。

只有具有与主服务器传输的调用地址匹配的从属服务器才会通过发送回确认位来响应。这是通过在第九个时钟将 SDA 拉低来完成的（见图 12-9）。

系统中没有两个奴隶可以拥有相同的地址。如果 IIC 模块是主模块，它不能传输等于其自身从属地址的地址。IIC 不能同时成为主人和奴隶。然而，如果仲裁在 a 期间丢失 address 循环，IIC 将恢复到从属模式，即使它由另一个主服务器处理，也能正常运行。

12.4.1.3 数据传输

在成功实现从属寻址之前，数据传输可以逐个方向逐字节进行

由调用主发送的 R/W 位指定。

地址周期之后的所有传输都被称为数据传输，即使它们携带从设备的子地址信息

每个数据字节长度为 8 位。只有在 SCL 低时才能更改数据，并且必须在 SCL 高时保持稳定，如图所示图 12-9。每个数据位在 SCL 上都有一个时钟脉冲，首先传输 MSB。每个数据字节后跟第 9 个（确认）位，该位来自接收设备的信号。通过第九个时钟将 SDA 拉低来发出确认信号。总之，一个完整的数据传输需要九个时钟脉冲。

如果从属接收者在第 9 位时间不承认主，则 SDA 行必须由从属留在高处。主服务器将失败的确认解释为失败的数据传输。

如果主接收器在数据字节传输后不承认从属发射器，从属将此解释为数据传输的结束，并释放 SDA 线路。

无论哪种情况，数据传输都会中止，主服务器会做以下两件事之一：

- 通过生成停止信号来放弃总线。
- 通过生成重复的 START 信号开始新的调用。

12.4.1.4 停车信号灯

主服务器可以通过生成 STOP 信号来释放总线来终止通信。然而，主服务器可以生成 START 信号，然后是调用命令，而无需先生成 STOP 信号。这被称为重复启动。停止信号是 de 罚款为 SDA 的低到高过渡，而 SCL 处于逻辑 1（见图 12-9）。

即使从服务器生成了确认，主服务器也可以生成 STOP，此时从服务器必须释放总线。

12.4.1.5 重复启动信号

如图所示图 12-9，重复的 START 信号是在没有首先生成 STOP 信号来终止通信的情况下生成的 START 信号。这由主使用，在不释放总线的情况下，以不同模式（传输/接收模式）与另一个从属或同一从属进行通信。

12.4.1.6 仲裁程序

IIC 总线是一个真正的多主总线，允许在上面连接多个主总线。如果两个或多个主体试图同时控制总线，时钟同步程序决定总线时钟，其中低周期等于最长的时钟低周期和高周期等于大师中最短的时钟。竞争主的相对优先级由数据仲裁程序决定，总线主发送逻辑 1 而另一个 ma，则总线主服务器将失去仲裁 Ster 传输逻辑 0。失败的主人立即切换到从属接收模式，并停止驱动 SDA 输出。在这种情况下，从主模式到从模式的过渡不会产生 STOP 条件。同时，硬件设置了一个状态位，以指示仲裁的损失。

12.4.1.7 时钟同步

由于线和逻辑是在 SCL 线路上执行的，因此 SCL 线路上的高到低过渡会影响总线上连接的所有设备。设备开始计算其低周期，在设备的时钟变低后，它将 SCL 线保持在低位，直到达到时钟高状态。然而，如果另一个设备时钟仍在其低周期内，则此设备时钟中从低到高的变化可能不会改变 SCL 线的状态。因此，同步时钟 SCL 被设备保持在低位最长的低期。在此期间，低周期较短的设备进入高等待状态（请参阅图 12-10）。当所有相关设备都计算掉其低周期时，同步时钟 SCL 线被释放并拉高。然后，设备时钟和 SCL 线的状态之间没有区别，所有设备都开始计算其高周期。第一个完成高周期的设备再次将 SCL 线拉低。

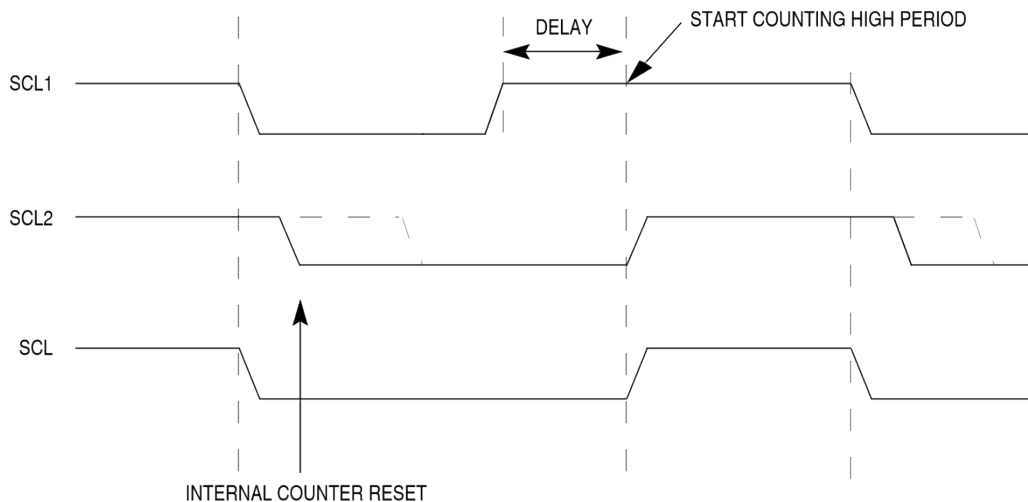


图 12-10。IIC 时钟同步

12.4.1.8 握手

时钟同步机制可以用作数据传输中的握手。完成一个字节传输（9 位）后，从设备可能会将 SCL 保持在低位。在这种情况下，它会停止总线时钟，并强制主时钟进入等待状态，直到从服务器释放 SCL 线路。

12.4.1.9 时钟拉伸

从属服务器可以使用时钟同步机制来减慢传输的比特率。在主服务器将 SCL 低开低后，从站可以在所需时间内将 SCL 开低，然后释放它。如果奴隶 SCL 低周期大于主 SCL 低周期，然后生成的 SCL 总线信号低周期被拉伸。

12.4.2 10 位地址

对于 10 位寻址，0x11110 用于第一个地址字节的前 5 位。在包含 10 位寻址的传输中，可以进行各种读/写格式的组合。

12.4.2.1 主发射器向从接收器寻址

转移方向没有改变（见表 12-9）。当 10 位地址遵循 START 条件时，每个从服务器将从服务器地址（11110XX）的第一个字节的前七个位与自己的

地址并测试第八位（R/W 方向位）是否为 0。多个设备可能会找到匹配项并生成确认（A1）。每个找到匹配项的从属将比较从属地址的第二个字节的八位 s 有自己的地址，但只有一个奴隶会找到匹配项并生成确认（A2）。匹配的从属地将由主服务器寻址，直到它收到 STOP 条件（P）或重复的 START 条件（Sr），然后是不同的从属广告连衣裙。

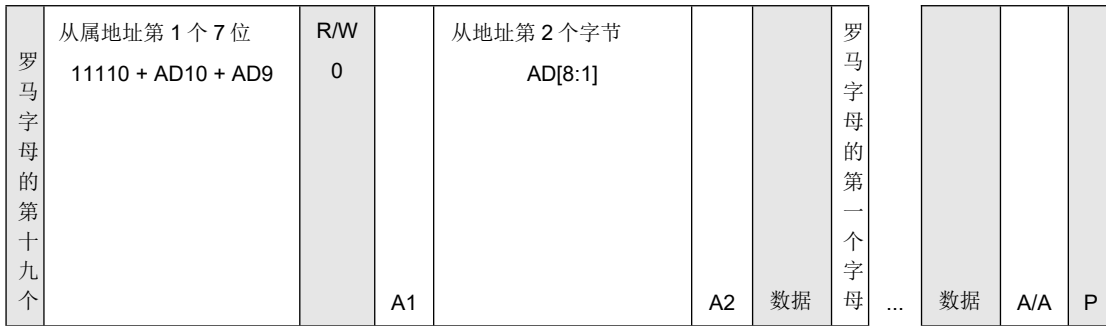


表 12-9。具有 10 位地址的主发射器地址从接收器

在主发射器发送 10 位地址的第一个字节后，从接收器将看到 IIC 中断。用户软件必须确保对于此中断，IICD 的内容被忽略，而不是被视为有效数据。

12.4.2.2 主接收器处理从发送器

传输方向在第二个 R/W 位后发生变化（见表 12-10）。直到并包括确认位 A2，该过程与主发射器寻址从属接收器所描述的过程相同。在重复的 START 条件（Sr）之后，匹配的从服务器会记住它之前已经解决过。然后这个奴隶 c 见鬼，从地址的第一个字节的前七位是否如下

Sr 与 START 条件（S）后相同，并测试第八位（R/W）是否为 1。如果有匹配，从属认为它已被处理为发射器，并生成确认 A3。奴隶发射器仍然是地址 sed，直到它收到 STOP 条件（P）或重复的 START 条件（Sr），然后是不同的从属地址。

在重复的 START 条件（Sr）后，所有其他从设备也将比较前七位

带有自己地址的从属地址的第一个字节，并测试第八位（R/W）位。然而，没有一个

它们将被寻址，因为 R/W = 1（对于 10 位设备），或 11110XX 从属地址（对于 7 位设备）不匹配。

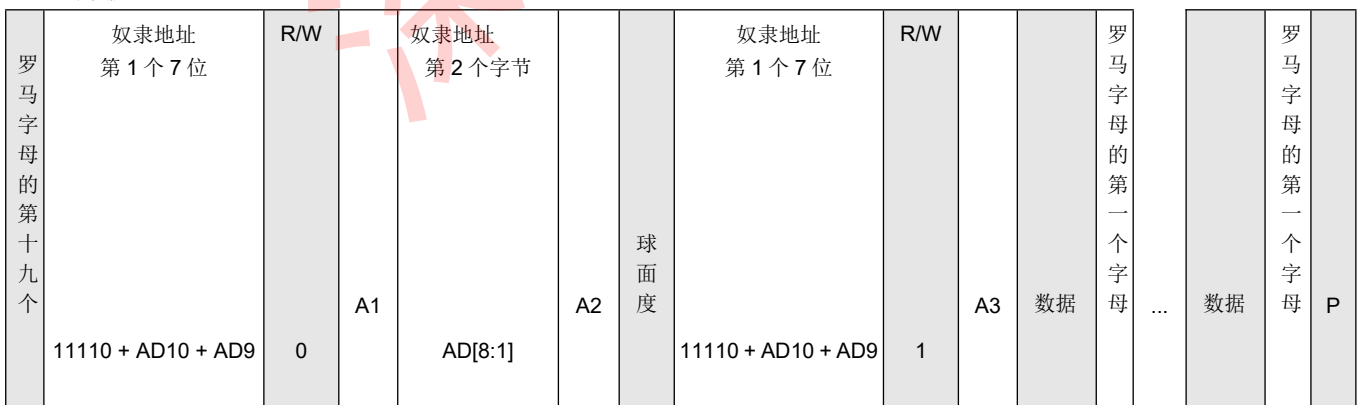


表 12-10. 主接收器用 10 位地址向从属发射器寻址

在主接收器发送 10 位地址的第一个字节后，从发送器将看到一个 IIC 中断。用户软件必须确保对于此中断，IICD 的内容被忽略，而不是被视为有效数据。

12.4.3 一般呼叫地址

一般调用可以用 7 位地址或 10 位地址请求。如果设置了 GCAEN 位，IIC 将匹配通用呼叫地址以及它自己的从属地址。当 IIC 响应一般调用时，它充当从接收器，并在地址周期后设置 IAAS 位。软件必须在第一个字节传输后读取 IICD 寄存器，以确定地址匹配是自己的从属地址还是一般调用。如果值为“00”，则匹配是一般调用。如果 GCAEN 位是 clear，IIC 通过不发出确认来忽略从普通呼叫地址提供的任何数据。

12.5 重置

IIC 在重置后被禁用。IIC 不能导致 MCU 重置。

12.6 中断

IIC 生成单个中断。

当任何事件发生时，会产生来自 IIC 的中断表 12-11 发生，前提是设置了 IICIE 位。中断由位 IICIF（IIC 状态寄存器）驱动，并被位 IICIE（IIC 控制寄存器）屏蔽。软件必须通过在中断例程中写入 1 来清除 IICIF 位。User 可以通过读取状态寄存器来确定中断类型。

表 12-11. 中断摘要

中断源	地位	旗	本地启用
完成 1 字节传输	TCF	IICIF	IICIE
匹配收到的呼叫地址	IAAS	IICIF	IICIE
仲裁丢失	ARBL	IICIF	IICIE

12.6.1 字节传输中断

TCF（传输完整标志）位设置在第 9 个时钟的下降边缘，以指示字节传输的完成。

12.6.2 地址检测中断

当调用地址与编程的从属地址（IIC 地址寄存器）匹配时，或者当设置 GCAEN 位并收到常规调用时，将设置状态寄存器中的 IAAS 位。只要设置了 IICIE，CPU 就会中断。CPU 必须检查 SRW 位并相应地设置其 Tx 模式。

12.6.3 仲裁丢失中断

IIC 是一个真正的多主总线，允许在上面连接多个主总线。如果两个或多个主体试图同时控制总线，则竞争主体的相对优先级由数据仲裁程序决定。第二当 C 模块丢失数据仲裁过程并且设置了状态寄存器中的 ARBL 位时，会断言此中断。

在以下情况下，仲裁会丢失：

- 当主服务器在地址或数据传输周期中驱动高时，SDA 采样为低。
- 当主服务器在数据接收周期的确认位中驱动高时，SDA 采样为低。
- 当公交车繁忙时，会尝试启动循环。
- 在从属模式下请求重复的 START 循环。
- 当主服务器没有请求时，检测到停止状态。

这个位必须由软件通过写入 1 来清除。

深圳南天星

12.7 初始化/应用程序信息

模块初始化（从）

1. 写：IICC2
 - 启用或禁用常规调用
 - 选择 10 位或 7 位寻址模式
2. 写：IICA
 - 设置从属地址
3. 写：IICC1
 - 启用 IIC 和中断
4. 初始化 RAM 变量（IICEN = 1 和 IICIE = 1）以传输数据
5. 初始化用于实现例程的 RAM 变量图 12-12

模块初始化（主）

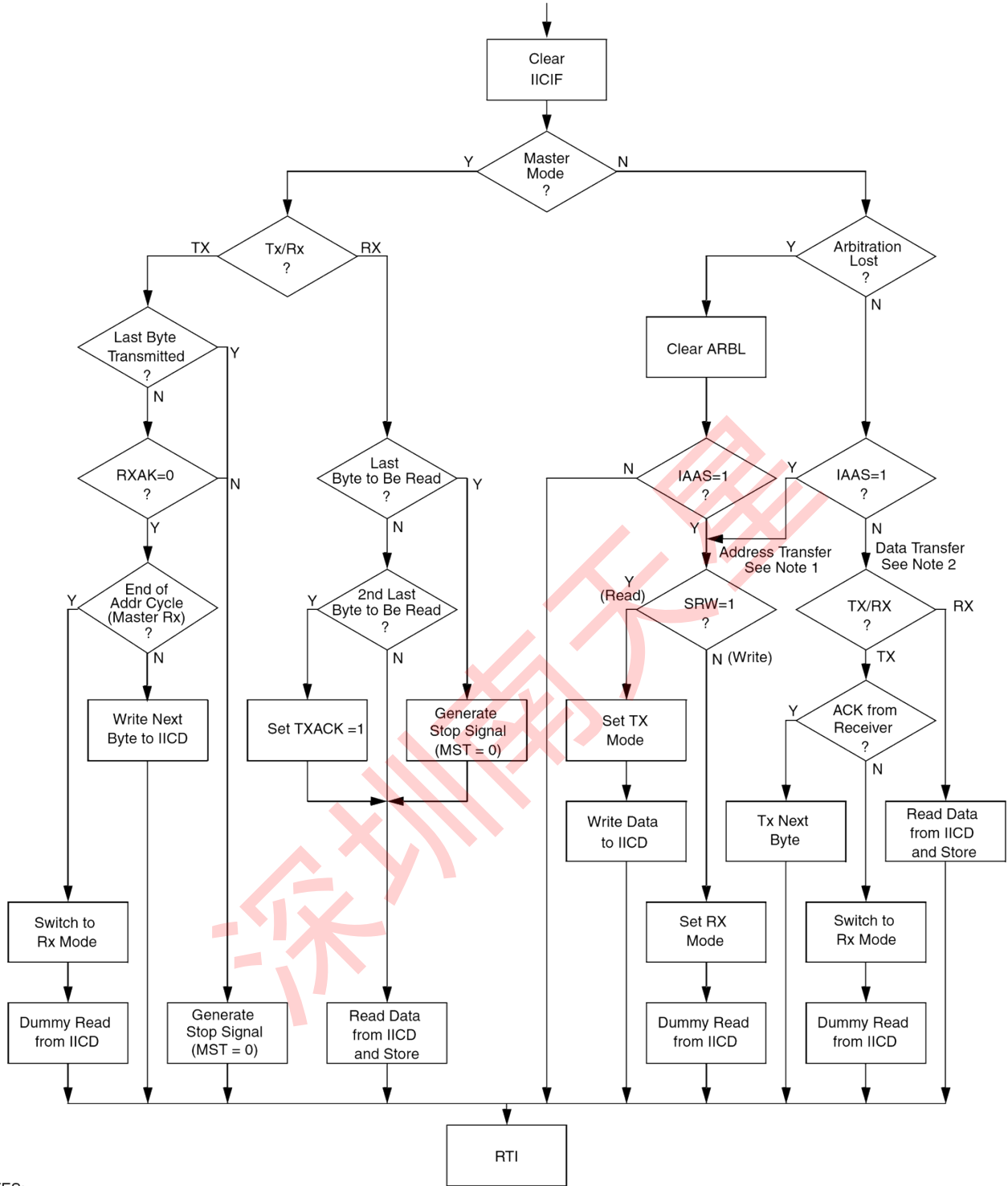
1. 写：IICF
 - 设置 IIC 包率（本章提供的示例）
2. 写：IICC1
 - 启用 IIC 和中断
3. 初始化 RAM 变量（IICEN = 1 和 IICIE = 1）以传输数据
4. 初始化用于实现例程的 RAM 变量图 12-12
5. 写：IICC1
 - 启用 TX

6. 写: IICC1

注册模型

IICA	广告[7:1]						0
模块作为从属处理时将响应的地址（在从属模式下）							
IICF	MULT			ICR			
波特率 = BUSCLK / (2 x MULT x (SCL DIVIDER))							
IICC1							
IICS	IICEN	IICIE	MST	德克萨斯州	TXAK	RSTA	0 0
模块配置							
IICD	TCF	IAAS	无暇的	ARBL	0	SRW	IICIF RXAK
模块状态标志							
IICC2	数据						地址配置
数据寄存器：写入传输 IIC 数据读取 IIC 数据							
GCAEN	ADEXT	0	0	0	AD10	AD9	公元 8

图 12-11。IIC 模块快速启动



NOTES:

- 1 如果启用了普通呼叫，则必须进行检査，以确定接收的地址是否为普通呼叫地址 (0x00)。如果接收的地址是普通呼叫地址，那么普通呼叫必须由用户软件处理。
- 2 当 10 位寻址用于寻址从服务器时，从服务器将在扩展地址的第一个字节后看到一个中断。用户软件必须确保对于此中断，IICD 的内容被忽略，而不是被视为有效的数据传输。

图 12-12. 典型的 IIC 中断例程

第 13 章 模块化计时器（RS08MTIMV1）

13.1 简单介绍

MTIM 是一个简单的 8 位计时器，具有几个软件可选的时钟源和可编程中断。对于包含多个 MTIM 的 MCU，MTIM 统称为 MTIM。例如，具有两个 MTIM 的 MCU 的 MTIM_x 是指 MTIM1 和 MTIM2。对于只包含一个 MTIM 的 MCU，它被称为 MTIM1。

MTIM 的核心组件是 8 位计数器，可以作为自由运行计数器或模计数器运行。可以启用计时器溢出中断，为基于时间的软件循环生成周期性中断。

MTIM 在 MC9RS08KA8 系列中有三个可选的参考时钟。它们是来自 ICS 模块的 TCLK 时钟、ICSFFCLK 时钟和总线时钟。

图 13-1 显示 MC9RS08KA8 系列方框图，并突出显示 MTIM。



第 13 章 模量计时器 (RS08MTIMV1)

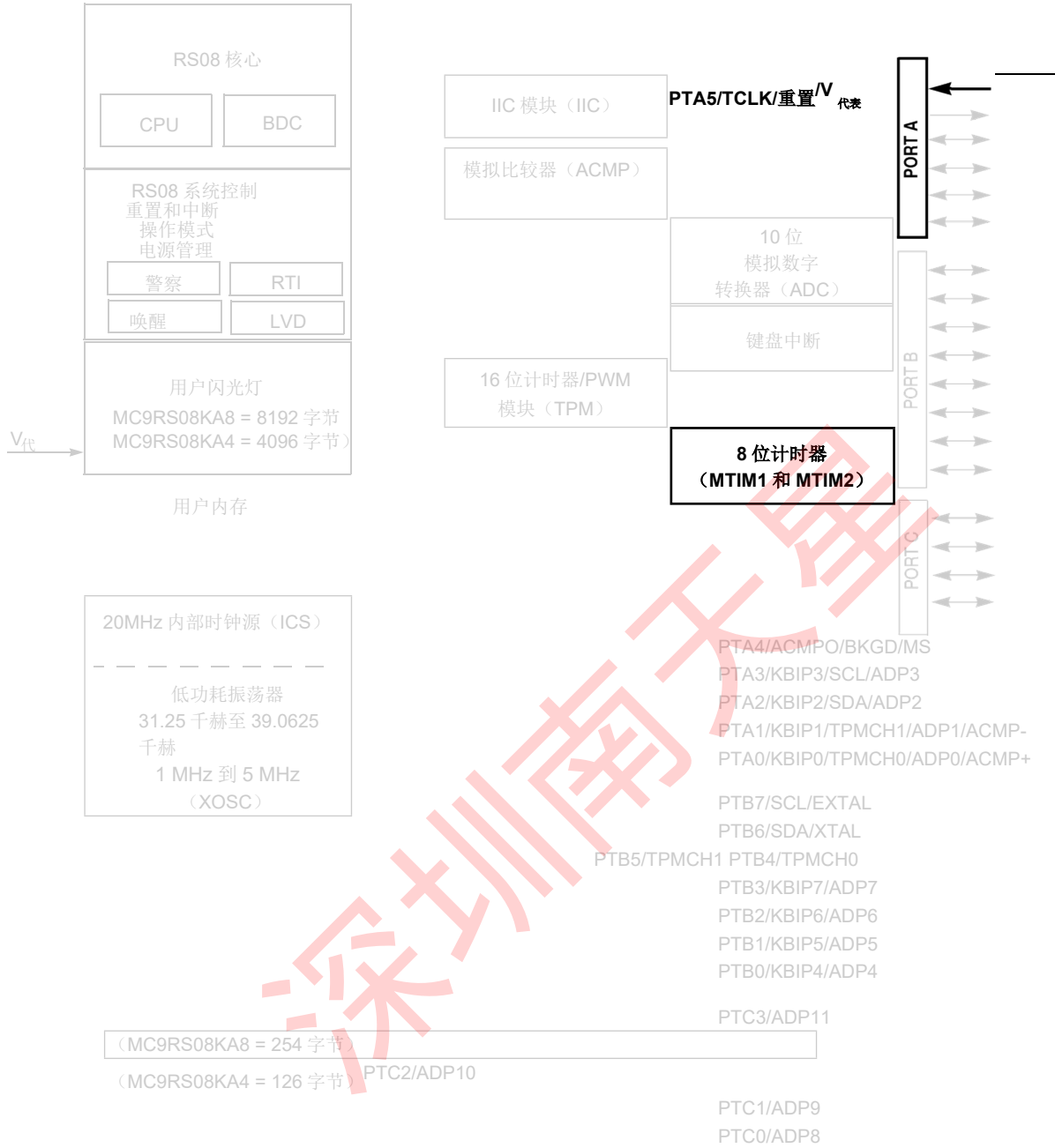




图 13-1. MC9RS08KA8 系列块图突出显示 MTIM 块和销

13.1.1 特点

计时器系统功能包括：

- 8 位上行计数器
 - 自由运行或 8 位模块限制
 - 溢出时软件可控中断
 - 计数器重置位 (TRST)
 - 计数器停止位 (TSTP)
- 四个软件可选时钟源，用于输入到预缩放器：
 - 系统总线时钟 — 上升边缘
 - 固定频率时钟 (XCLK) — 上升边缘
 - TCLK 引脚上的外部时钟源 — 上升边缘 — TCLK 引脚上的外部时钟源 — 下降边缘
- 九个可选的时钟预刻度值：
 - 时钟源除以 1、2、4、8、16、32、64、128 或 256

13.1.2 操作模式

本节定义了 MTIM 在停止、等待和后台调试模式下的操作。

13.1.2.1 在等待模式下操作

如果在执行 WAIT 指令之前启用，MTIM 将继续在等待模式下运行。因此，如果启用了计时器溢出中断，MTIM 可用于使 MCU 退出等待模式。为了尽可能低的电流消耗，MTIM 必须是如果在等待模式下不需要作为中断源，则由软件禁用。

13.1.2.2 在停止模式下操作

在执行 STOP 指令之前，无论设置如何，MTIM 在所有停止模式下都被禁用。因此，MTIM 不能用作停止模式的唤醒源。

如果停止通过重置退出，MTIM 将进入重置状态。如果停止通过中断退出，MTIM 将从进入停止时的状态继续。如果计数器在进入停止时处于活动状态，则计数将从当前值恢复。

13.1.2.3 在活动后台模式下操作

MTIM 暂停所有计数，直到 MCU 恢复到正常用户操作模式。只要没有发生 MTIM 重置，就从暂停值中计算简历（TRST 写入 1 或任何值写入 MTIMMOD 寄存器）。

深圳市南天星

13.1.3 方框图

显示了模量计时器模块的框图图 13-2。

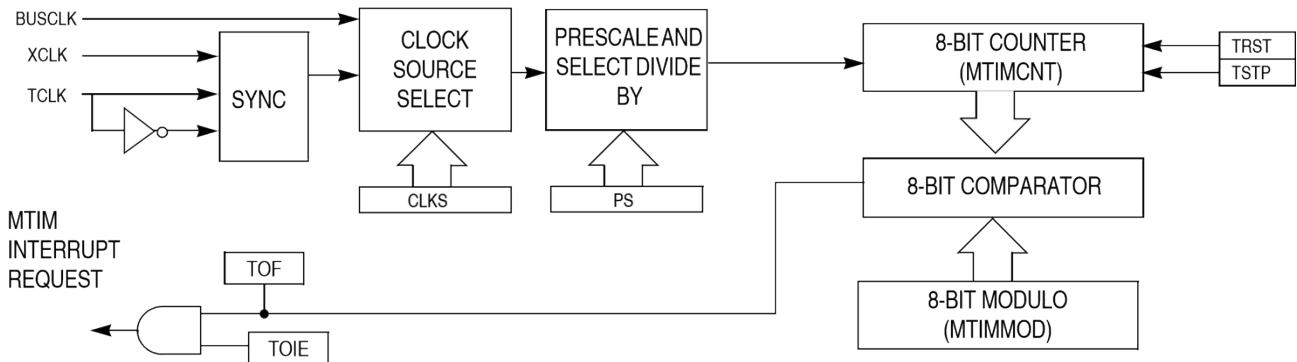


图 13-2. 模定时器 (MTIM) 方框图

13.2 外部信号描述

MTIM 包括一个外部信号，TCLK，用于在选择 MTIM 时钟源时输入外部时钟。TCLK 的信号特性显示在表 13-1。

表 13-1. 信号属性

信号	函数	I/O
TCLK	外部时钟源输入 MTIM	我

TCLK 输入必须由总线时钟同步。此外，必须适应占空比和时钟抖动的变化。因此，TCLK 信号必须限制在总线频率的四分之一。

TCLK 引脚可以与通用端口引脚混合。看到[引脚和连接](#)该函数的引脚位置和优先级的章节。

13.3 注册定义

每个 MTIM 包括四个寄存器，总结在表 13-2 冒号：

- 8 位状态和控制寄存器
- 8 位时钟配置寄存器
- 8 位计数器寄存器
- 8 位模寄存器

有关所有 MTIM 寄存器的绝对地址分配，请参阅本数据表内存部分的直接页面寄存器摘要。本节仅通过其名称来指寄存器和控制位。

表 13-2。MTIM 注册摘要

名字		7	6	5	4	3	2	1	0
MTIMSC	字母 R	TOF	托伊	0	TSTP	0	0	0	0
	罗马字母的第 23 个字母			TRST					
MTIMCLK	字母 R	0	0	记 CLKS					
	罗马字母的第 23 个字母								
MTIMCNT	字母 R	钻 NT							
	罗马字母的第 23 个字母								
MTIMMOD	字母 R	摩登							
	罗马								

	字 母 的 第 23 个 字 母	
--	---------------------------------------	--

13.3.1 MTIM 状态和控制登记册 (MTIMSC)

MTIMSC 包含溢出状态标志和控制位，用于配置中断启用、重置计数器和停止计数器。

		7	6	5	4	3	2	1	0
		TOF	托伊	0	TSTP	0	0	0	0
				TRST					
		0	0	0	1	0	0	0	0

字母 R

罗马字母的第 23 个字母

重置:

图 13-3. MTIM 状态和控制登记册 (MTIMSC)

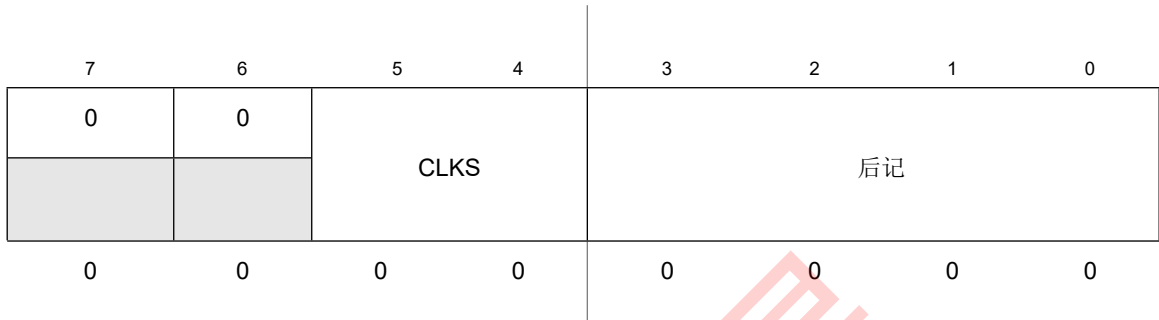
表 13-3. MTIMSC 现场描述

字段	描述
7 TOF	<p>MTIM 溢出标志—当 MTIM 计数器寄存器在达到 MTIM 模寄存器中的值后溢出到 00 美元时，将设置此只读位。在设置 TOF 时读取 MTIMSC 寄存器来清除 TOF，然后将 0 写入 TOF。当写入 TRST 时，TOF 也会被清除到 1 或当任何值写入 MTIMMOD 寄存器时。</p> <p>0 MTIM 计数器尚未达到 MTIM 模寄存器中的溢出值。</p> <p>1 MTIM 计数器已达到 MTIM 模寄存器中的溢出值。</p>
6 托伊	<p>MTIM 溢出中断启用—此读/写位启用 MTIM 溢出中断。如果设置了 TOIE，则当 TOF = 1 时会生成中断。重置清除 TOIE。如果 TOF = 1，请不要设置 TOIE。首先清除 TOF，然后设置 TOIE。</p> <p>0 TOF 中断被禁用。使用软件轮询。</p> <p>1 TOF 中断已启用。</p>
5 TRST	<p>MTIM 计数器重置—当 1 写入此只写入位时，MTIM 计数器寄存器重置为 00 美元，TOF 被清除。阅读此位总是返回 0。</p> <p>0 没有效果。MTIM 计数器仍处于当前状态。</p> <p>1 MTIM 计数器重置为 00 美元。</p>

4 TSTP	<p>MTIM 计数器停止—设置时，此读/写位以当前值停止 MTIM 计数器。当 TSTP 被清除时，从当前值计数恢复。重置设置 TSTP 以防止 MTIM 计数。</p> <p>0 MTIM 计数器处于活动状态。</p> <p>1 MTIM 计数器已停止。</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------

13.3.2 MTIM 时钟配置寄存器 (MTIMCLK)

MTIMCLK 包含时钟选择位 (CLKS) 和预缩放器选择位 (PS)。



字母 R

罗马字母的第 23 个字母

重置:

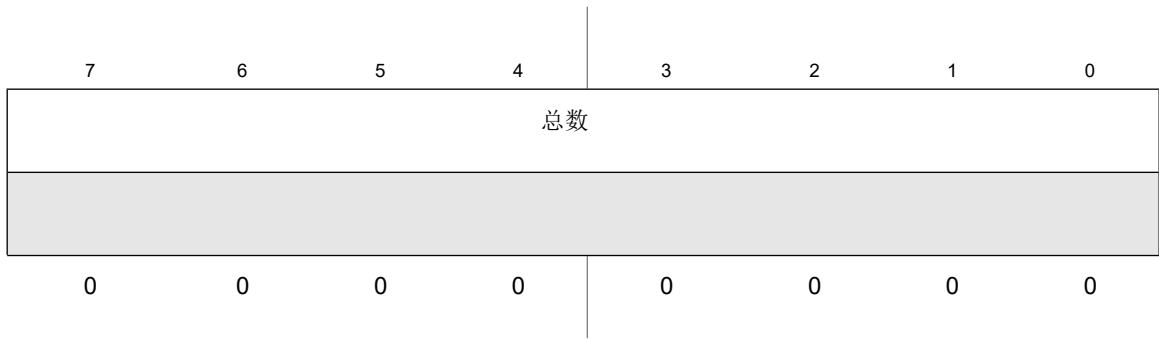
图 13-4. MTIM 时钟配置寄存器 (MTIMCLK)

表 13-4. MTIMCLK 现场描述

字段	描述
5:4 CLKS	<p>时钟源选择—这两个读/写位从四个不同的时钟源中选择一个作为 MTIM 预缩放器的输入。在计数器处于活动状态时更改时钟源不会清除计数器。计数继续使用新的时钟源。重置将 CLKS 清除为 00。</p> <p>00 编码 0—总线时钟 (BUSCLK)。</p> <p>01 编码 1—固定频率时钟 (XCLK)。</p> <p>10 编码 3—外部源 (TCLK 引脚), 下降边缘。</p> <p>11 编码 4—外部源 (TCLK 引脚), 上升边缘。</p>
3:0 后记	<p>时钟源预缩放器—这四个读/写位从 8 位预缩放器的九个输出中选择一个。在计数器处于活动状态时更改预缩放器值不会清除计数器。计数继续使用新的预缩放器值。重置清除 PS 为 0000。</p> <p>0000 编码 0 — MTIM 时钟源 ÷ 1.</p> <p>0001 编码 1 — MTIM 时钟源 ÷ 2.</p> <p>0010 编码 2 — MTIM 时钟源 ÷ 4.</p> <p>0011 编码 3 — MTIM 时钟源 ÷ 8.</p> <p>0100 编码 4 — MTIM 时钟源 ÷ 16.</p> <p>0101 编码 5 — MTIM 时钟源 ÷ 32.</p> <p>0110 编码 6 — MTIM 时钟源 ÷ 64.</p> <p>0111 编码 7 — MTIM 时钟源 ÷ 128.</p> <p>1000 编码 8 — MTIM 时钟源 ÷ 256.</p> <p>所有其他编码默认为 MTIM 时钟源 ÷ 256.</p>

13.3.3 MTIM 计数器寄存器 (MTIMCNT)

MTIMCNT 是 8 位计数器当前 MTIM 计数的只读值。



字母 R

罗马字母的第 23 个字母

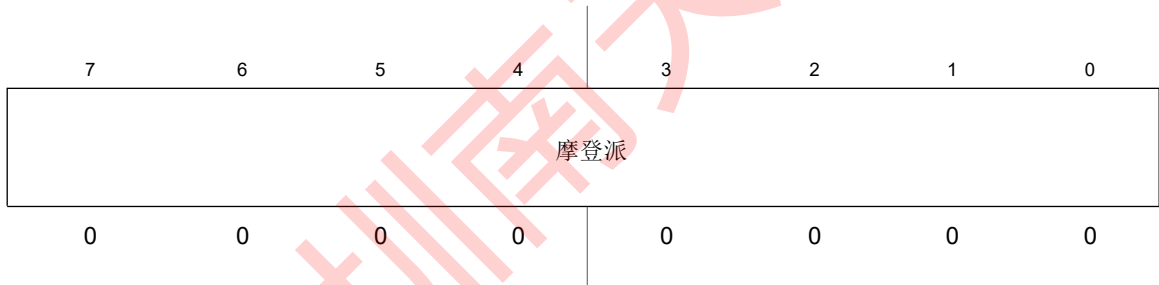
重置:

图 13-5. MTIM 计数器寄存器 (MTIMCNT)

表 13-5. MTIMCNT 字段描述

字段	描述
7:0 总数	MTIM 计数 —这八个只读位包含 8 位计数器的当前值。写入对这个寄存器没有影响。重置将计数清除为 00 美元。

13.3.4 MTIM 模量寄存器 (MTIMMOD)



字母 R

罗马字母的第 23 个字母

重置:

图 13-6. MTIM 模量寄存器 (MTIMMOD)

表 13-6. MTIMMOD 描述

字段	描述
7:0 摩登派	MTIM 模块 —这八个读/写位包含用于重置计数和设置 TOF 的模量值。00 美元的值使 MTIM 处于自由运行模式。写入 MTIMMOD 将计数重置为 00 美元并清除 TOF。重置将模块设置为 00 美元。

13.4 功能描述

MTIM 由一个带有 8 位模量寄存器的主 8 位上计数器、一个时钟源选择器和一个具有九个可选择值的预缩放块组成。该模块还包含软件可选中断逻辑。

MTIM 计数器 (MTIMCNT) 有三种操作模式: 停止、自由运行和模态。重置后, 计数器停止了。如果计数器启动时没有向模量寄存器写入新值, 则计数器将处于自由运行模式。字母 T 当计数器运行时, 除 00 美元以外的值在模块寄存器中时, 计数器处于模模式。

在任何 MCU 重置后, 计数器被停止并重置为 00 美元, 模量设置为 00 美元。总线时钟被选为默认时钟源, 预刻度值除以 1。要以自由运行模式启动 MTIM, 只需写入 MTIM 状态 s 和控制寄存器 (MTIMSC) 并清除 MTIM 停止位 (TSTP)。

四个时钟源是软件可选的: 内部总线时钟、固定频率时钟 (XCLK) 和 TCLK 引脚上的外部时钟, 可选择在上升或下降边缘的增量。使用 MTIMCLK 中的 MTIM 时钟选择位 (CLKS) 选择所需的时钟源。如果在选择新时钟源时计数器处于活动状态 (TSTP = 0), 则计数器将继续使用新时钟源从以前的值计数。

九个预缩放值是软件可选的: 时钟源除以 1、2、4、8、16、32、64、128 或 256。

预缩放器在 MTIMCLK 中选择位 (PS) 选择所需的预缩放值。如果选择新的预缩放器值时计数器处于活动状态 (TSTP = 0), 则计数器将继续使用新的预缩放器值从以前的值计数。

MTIM 模态寄存器 (MTIMMOD) 允许将溢出比较值设置为从 01 美元到 \$FF 的任何值。重置将模量清除为 00 美元, 从而产生一个自由运行的计数器。

当计数器处于活动状态 (TSTP = 0) 时, 计数器以选定的速率递增, 直到计数与模量值匹配。当这些值匹配时, 计数器溢出到 00 美元并继续计数。每当计数器溢出时, 都会设置 MTIM 溢出标志 (TOF)。标志设置从模量值过渡到 00 美元。在计数器处于活动状态时写入 MTIMMOD, 将计数器重置为 00 美元并清除 TOF。

清算 TOF 是一个两步过程。第一步是在设置 TOF 时读取 MTIMSC 寄存器。第二步是向 TOF 写入 0。如果在第一步和第二步之间发生另一个溢出, 清除过程将被重置, TOF 将保持设置在执行第二步之后。这将防止第二次发生被错过。当 1 写入 TRST 或任何值写入 MTIMMOD 寄存器时, TOF 也会被清除。

MTIM 允许在设置 TOF 时生成可选的中断。要启用 MTIM 溢出中断, 请在 MTIMSC 中设置 MTIM 溢出中断启用位 (TOIE)。TOIE 绝不能写入 1, 而 TOF = 1。相反, TOF 必须被清除首先, 然后 TOIE 可以设置为 1。

13.4.1 MTIM 操作示例

本节显示了当计数器从模寄存器中达到匹配值时, MTIM 操作的示例。

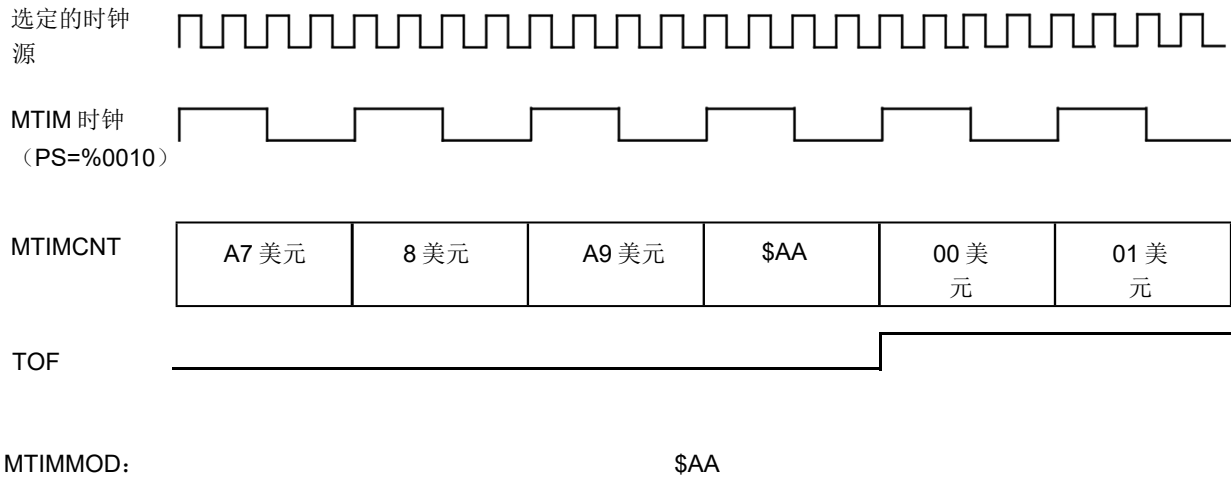


图 13-7. MTIM 计数器溢出示例

以...为例图 13-7，选定的时钟源可以是四种可能选择中的任何一个。预缩放器设置为 PS = %0010 或除以 4。MTIMMOD 寄存器中的模值设置为 \$AA。当计数器 MTIMCNT 达到 \$AA 的模值时，计数器超过欠 00 美元，并继续计算。当计数器值从 \$AA 变为 \$00 时，计时器溢出标志 TOF 会设置。设置 TOF 时会生成 MTIM 溢出中断，如果 TOIE = 1。

第 14 章

16 位定时器/PWM (RS08TPMV2)

14.1 简单介绍

TPM 每个通道使用一个输入/输出 (I/O) 引脚, TPMCH_n, 其中 x 是 TPM 编号 (例如, 1 或 2), n 是通道号 (例如, 0-4)。TPM 与通用 I/O 端口引脚共享其 I/O 引脚 (请参阅第 2 章, “引脚和连接”。)

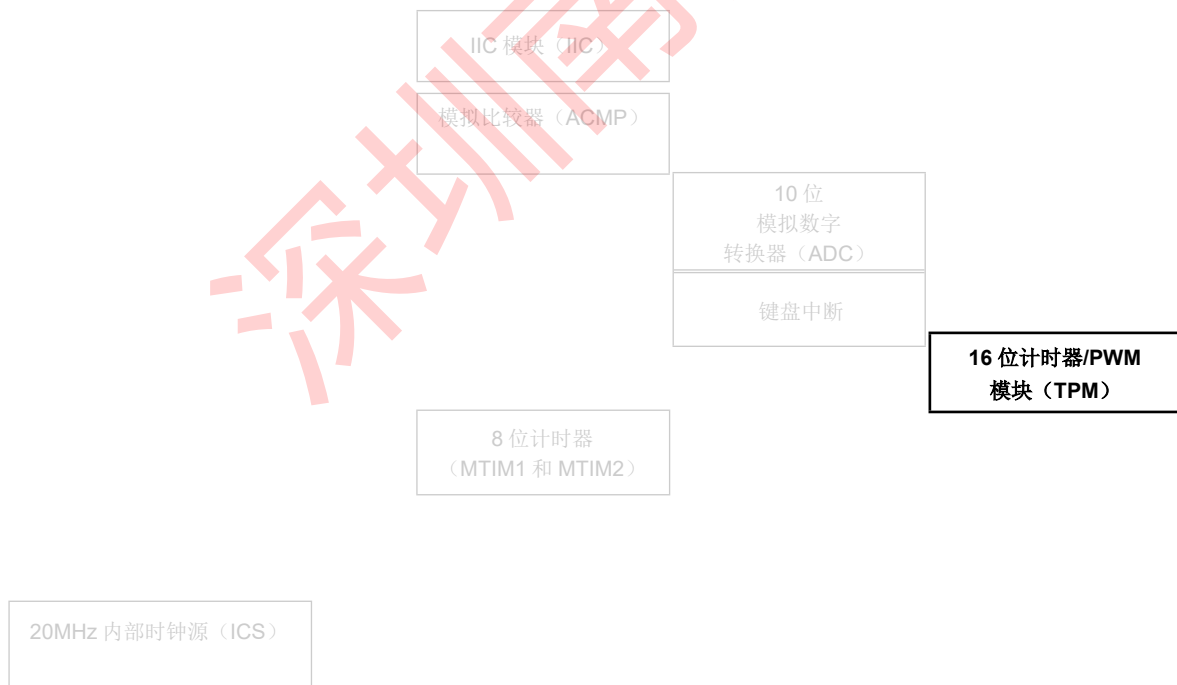
TPM 模块引脚、TPMCH0 和 TPMCH1 可以在软件控制下使用 SPT 中的 TPMCH0PS 重新定位 (表 14-1)。SOPT 中的 TPMCH0PS 和 TPMCH1PS 选择哪些通用 I/O 端口与 TPM 相关联。

表 14-1. TPM 位置选项

SOPT 中的 TPMCH0PS	TPMCH0 的端口引脚	SOPT 中的 TPMCH1PS	TPMCH1 的端口引脚
0 (默认)	PTA0	0 (默认)	PTA1
1	PTB4	1	PTB5

图 14-1 显示了突出显示 TPM 的 MC9RS08KA8 系列方框图。

第 14 章 16 位定时器/PWM (RS08TPMV2)



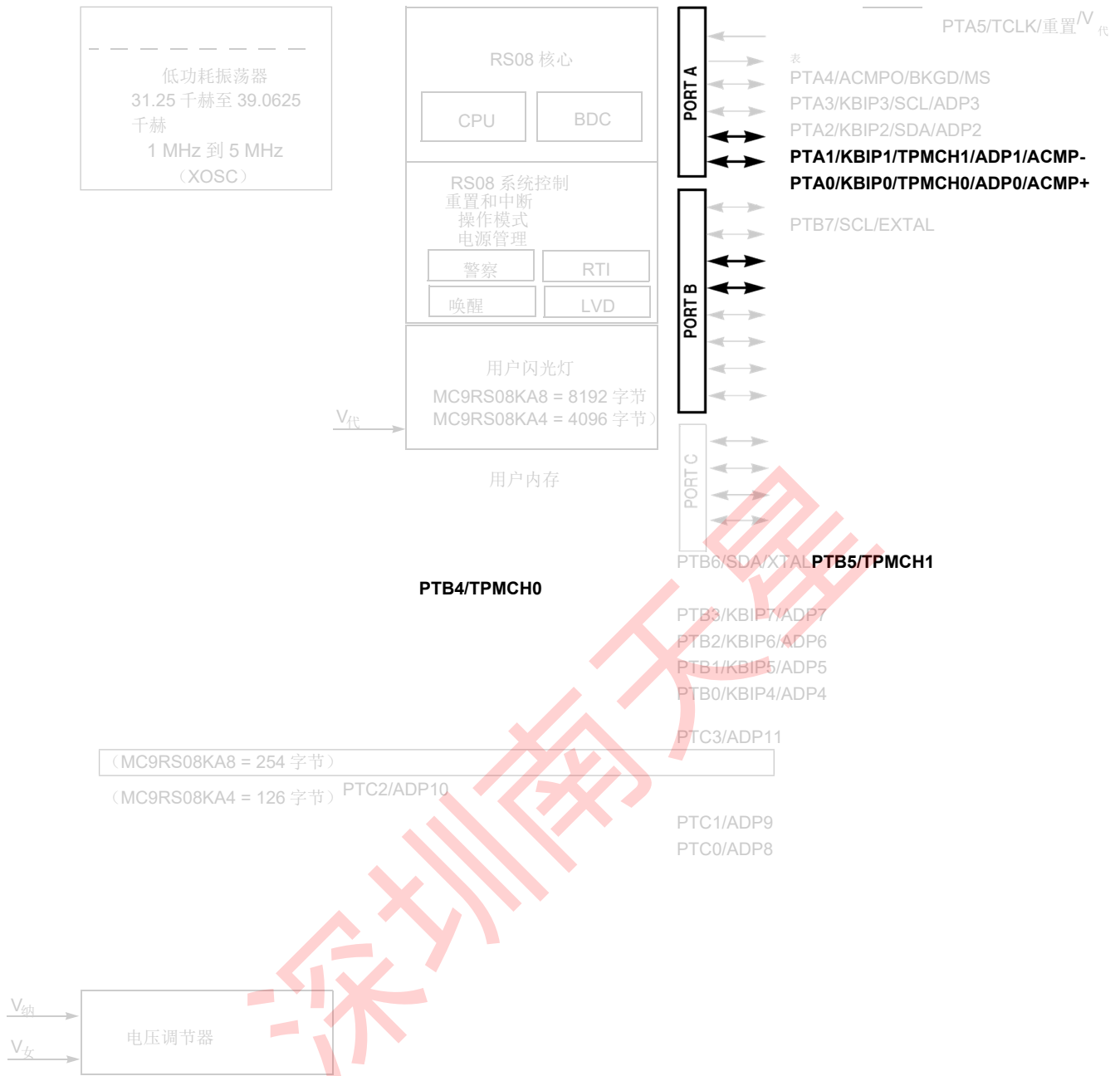


图 14-1。MC9RS08KA8 系列方框图突出显示 TPM 方块和引脚

16 位定时器/PWM (RS08TPMV2)

14.1.1 特点

TPM 具有以下功能:

- 每个 TPM 可以配置为所有通道上的缓冲、中心对齐脉冲宽度调制 (CPWM)
- 时钟源可按 TPM 独立选择 (多个 TPM 设备)
- 可选时钟源 (取决于设备): 总线时钟、固定系统时钟、外部引脚

- 时钟预缩放器水龙头除以 1、2、4、8、16、32、64 或 128
- 16 位自由运行或上/下 (CPWM) 计数操作
- 控制计数器范围的 16 位模量寄存器
- 计时器系统启用
- 每个通道一个中断，外加每个 TPM 模块的终端计数中断 (多个 TPM 设备)
- 频道功能：
 - 每个通道可以是输入捕获、输出比较或缓冲边缘对齐 PWM
 - 上升边缘、下降边缘或任何边缘输入捕获触发器
 - 设置、清除或切换输出比较操作
 - PWM 输出上可选的极性

14.1.2 方框图

图 14-2 显示 TPM 的结构。一些 MCU 包括多个 TPM，具有不同数量的通道。

16 位定时器/PWM (RS08TPMV2)

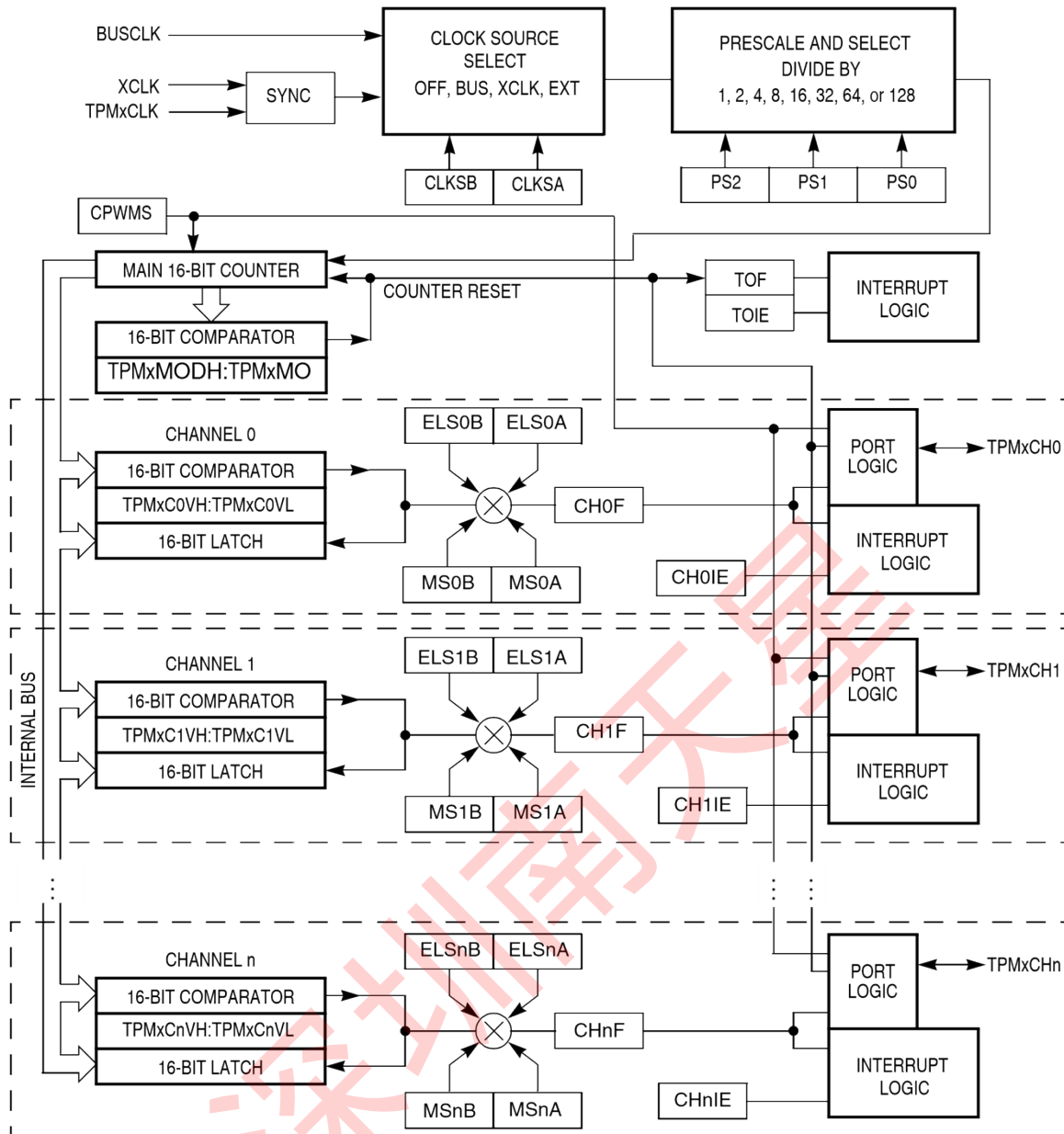


图 14-2. TPM 方框图

TPM 的核心组件是 16 位计数器，当 TPM 配置为中心对齐 PWM 时，它可以作为自由运行计数器、模计数器或上/下计数器运行。TPM 计数器（在正常上计数模式下运行时）提供输入捕获、输出比较和边缘对齐 PWM 功能的时序参考。计时器计数器模寄存器，TPMxMODH: TPMxMODL，控制计数器的模量值。（值 0x0000 或 0xFFFF 有效地使计数器自由运行。）软件可以随时读取计数器值，而不会影响计数顺序。任何写到 TP 的任何一个字节无论写入的数据值如何，MxCNT 计数器都会重置计数器。

16 位定时器/PWM (RS08TPMV2)

所有 TPM 通道都可以作为输入捕获、输出比较或缓冲边缘对齐 PWM 通道独立编程。

14.2 外部信号描述

当与计时器关联的任何引脚配置为计时器输入时，可以启用被动上拉。重置后，TPM 模块被禁用，所有引脚默认为通用输入，被动引脚被禁用。

14.2.1 外部 TPM 时钟源

当计时器状态和控制寄存器中的控制位 CLKSB:CLKSA 设置为 1:1 时，预缩放器以及 TPM_x 的 16 位计数器由连接到 I/O 引脚的外部时钟源 TPM_xCLK 驱动。两者之间需要一个同步器外部时钟和 TPM 的其余部分。此同步器由总线时钟计时，因此外部源的频率必须小于总线速率时钟频率的一半。这个外部时钟源的频率上限是特定的是总线频率的四分之一，以保守地适应占空比和锁相环（PLL）或频率锁定环（FLL）频率抖动效应。

在某些设备上，外部时钟输入与其中一个 TPM 通道共享。当 TPM 通道作为外部时钟输入共享时，关联的 TPM 通道不能使用引脚。（该通道仍然可以在输出比较模式下用作软件计时器。）此外，如果其中一个 TPM 通道用作外部时钟输入，则相应的 ELSnB:ELSnA 控制位必须设置为 0:0，以便该通道不会尝试使用相同的引脚。

14.2.2 TPM_xCH_n — TPM_x 通道 n I/O 引脚

每个 TPM 通道都与 MCU 上的 I/O 引脚相关联。此引脚的功能取决于通道的配置。在某些情况下，不需要引脚功能，因此引脚恢复到由通用 I/O 控制控制。当计时器 h 作为端口引脚的控制，端口数据和数据方向寄存器不影响相关引脚。看到[引脚和连接](#)章节了解有关共享引脚功能的更多信息。

14.3 注册定义

TPM 包括：

- 8 位状态和控制寄存器（TPM_xSC）
- 16 位计数器（TPM_xCNTH: TPM_xCNTL）
- 16 位模寄存器（TPM_xMODH: TPM_xMODL）

每个计时器通道都有：

- 8 位状态和控制寄存器（TPM_xCnSC）
- 16 位信道值寄存器（TPM_xCnVH: TPM_xCnVL）

请参阅直接页面寄存器摘要[记忆](#)本数据表的章节涉及所有 TPM 寄存器的绝对地址分配。本节仅通过其名称来指寄存器和控制位。罗马字母的第一个字母

表 14-3. TPM 时钟源选择

CLKSB: CLKSA	TPM 时钟源到预缩法器输入
0:0	未选择时钟 (TPMx 已禁用)
0: 1	总线速率时钟 (BUSCLK)
1:0	固定系统时钟 (XCLK)
1:1	外部来源 (TPMxCLK) 1,2

- ¹ 作为外部时钟允许的最大频率是总线频率的四分之一。
- ² 如果外部时钟输入与通道 n 共享, 并被选为 TPM 时钟源, 则相应的 ELSnB:ELSnA 控制位必须设置为 0:0, 这样通道 n 就不会尝试为冲突的功能使用相同的引脚。

表 14-4. 预刻度除数选择

PS2: PS1: PS0	TPM 时钟源划分
0:0:0	1
0:0: 1	2
0:1:0	4
0: 1: 1	8
1:0:0	16
1:0: 1	32
1:1:0	64
1:1: 1	128

14.3.2 计时器计数器寄存器 (TPMxCNTH: TPMxCNTL)

两个只读 TPM 计数器寄存器包含 TPM 计数器中值的高字节和低字节。读取任一字节 (TPMxCNTH 或 TPMxCNTL) 将两个字节的內容锁定到缓冲区中, 在那里它们保持锁定状态, 直到读取另一个字节。这允许以任一顺序进行连贯的 16 位读取。一致性机制通过 MCU 重置自动重新启动, 向 TPMxCNTH 或 TPMxCNTL 写入任何值, 或向计时器状态/控制寄存器 (TPMxSC) 写入任何值。

重置清除

TPM 计数器寄存器。

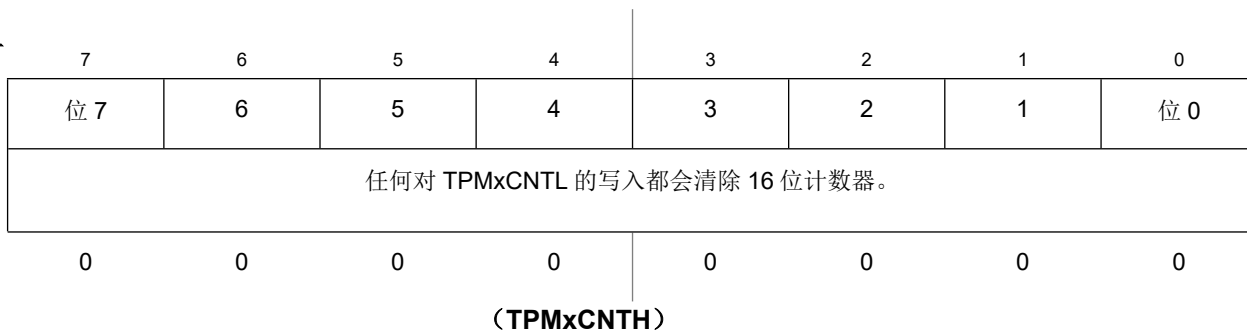
7	6	5	4	3	2	1	0
比特 15	14	13	12	11	10	9	比特 8
任何写入 TPMxCNTH 都会清除 16 位计数器。							
0	0	0	0	0	0	0	0

字母 R

罗马字母的第 23 个字母

调整

图 14-4. 计时器计数器高



字母 R

罗马字母的第 23 个字母

调整

图 14-5. 计时器计数器低 (TPMxCNTL)

当后台模式处于活动状态时，计时器计数器和一致性机制被冻结，这样即使在后台模式处于活动状态时读取计数器的一个或两个字节，缓冲门锁也会保持在后台模式处于活动状态。

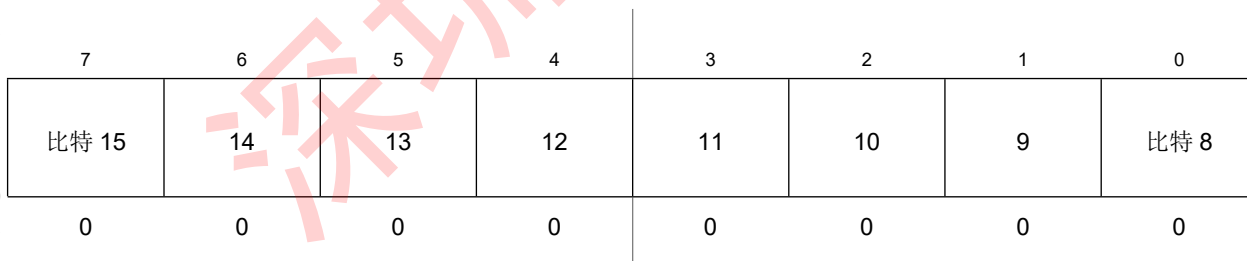
14.3.3 计时器计数器模块寄存器 (TPMxMODH: TPMxMODL)

读/写 TPM 模寄存器包含 TPM 计数器的模值。TPM 计数器达到模量值后，TPM 计数器在下一个时钟 (CPWMS = 0) 从 0x0000 恢复计数或开始倒计时 (CPWMS = 1)，溢出标志 (TOF) 被设置。写入 TPMxMODH 或 TPMxMODL 会抑制 TOF 和溢出中断，直到其他字节被写入。重置将 TPM

计数器模寄存器设置为

0x0000，这会

导致自由运行的计时器计数器（模禁用）。



字母 R

罗马字母的第 23 个字母

调整

图 14-6. 定时器计数器模块寄存器高 (TPMxMODH)

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

字母 R

罗马字母的第 23 个字母

调整

图 14-7. 定时器计数器模块寄存器低 (TPMxMODL)

最好等待溢出中断，以便在新的溢出之前写入模寄存器的两个字节。另一种方法是在写入 TPM 模寄存器之前重置 TPM 计数器，以避免混淆 wh 将发生第一个计数器溢出。

14.3.4 计时器通道 n 状态和控制寄存器 (TPMxCnSC)

TPMxCnSC 包含通道中断状态标志和控制位，用于配置中断启用、通道配置和引脚功能。

7	6	5	4	3	2	1	0
CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
0	0	0	0	0	0	0	0

字 罗

调整

Ed

未实现或保留

图 14-8. 计时器通道 n 状态和控制寄存器 (TPMxCnSC)

表 14-5. TPMxCnSC 寄存器字段描述

字段	描述
7 CHnF	<p>频道 n 标志—当通道 n 配置为输入捕获时，当通道 n 引脚上出现活动边缘时，将设置此标志位。当通道 n 是输出比较或边缘对齐 PWM 通道时，当 TPM 计数器寄存器中的值与 va 匹配时，CHnF 被设置 lue 在 TPM 通道 n 值寄存器中。此标志很少与中心对齐的 PWM 一起使用，因为每次计数器与通道值寄存器匹配时都会设置它，该寄存器对应于有效占空周期的两个边缘。</p> <p>设置 CHnF 并启用中断时，会请求相应的中断 (CHnIE = 1)。通过在 CHnF 设置时读取 TPMxCnSC 来清除 CHnF，然后将 0 写入 CHnF。如果在清除序列完成之前发生另一个中断请求，序列被重置，因此 CHnF 将在早期 CHnF 的清除序列完成后保持设置。这样做是为了不能通过清除之前的 CHnF 丢失 CHnF 中断请求。重置清除 CHnF。将 1 写入 CHnF 无效。</p> <p>0 通道 n 上没有发生输入捕获或输出比较事件</p> <p>1 输入捕获或输出比较事件发生在通道 n 上</p>

6 CHnIE	通道 n 中断启用 —此读/写位允许从通道 n 中断。重置清除 CHnIE。 0 通道 n 中断请求被禁用（使用软件轮询） 1 启用通道 n 中断请求
5 MSnB	TPM 频道 n 的模式选择 B —当 CPWMS = 0 时，MSnB = 1 为边缘对齐 PWM 模式配置 TPM 通道 n。有关通道模式和设置控件的摘要，请参阅表 14-6。
4 MSnA	模式为 TPM 通道 n 选择 A —当 CPWMS = 0 和 MSnB = 0 时，MSnA 将 TPM 通道 n 配置为输入捕获模式或输出比较模式。参考表 14-6 获取通道模式和设置控件的摘要。
3:2 ELSn[B:A]	边缘/水平选择位 —取决于设置的计时器通道的操作模式 CPWMS:MSnB:MSnA 并显示在表 14-6，这些位选择触发输入捕获事件的输入边缘的极性，选择响应输出比较匹配的水平，或选择 PWM 输出的极性。 将 ELSnB:ELSnA 设置为 0:0 会将相关计时器引脚配置为与任何计时器通道功能无关的通用 I/O 引脚。此功能通常用于暂时禁用输入捕获通道或使计时器引脚作为生成器可用当关联的计时器通道设置为不需要使用引脚的软件计时器时，al-purpose I/O 引脚。

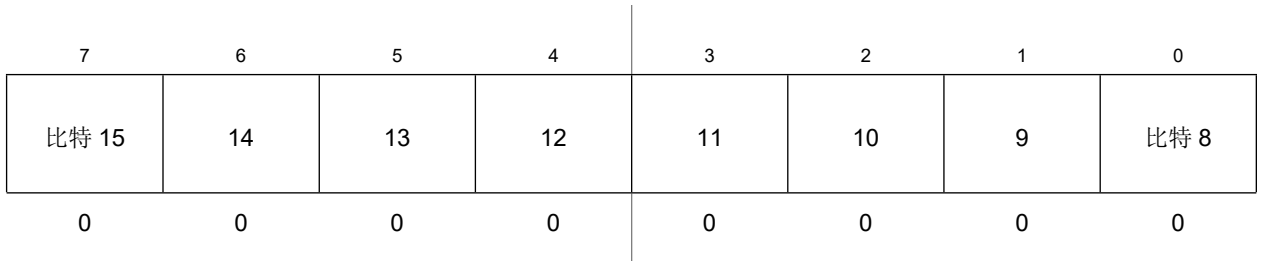
表 14-6. 模式、边缘和级别选择

CPWMS	MSnB: MSnA	ELSnB: ELSnA	形式	布局
英语字母中的第二十四 个字母	X 染色体	00		引脚不用于 TPM 通道；用作 TPM 的外部时钟或恢复到通用 I/O
0	00	01	输入捕获	仅在上升边缘捕获
		10		仅在坠落边缘捕获
		11		捕获上升或下降边缘
	01	输出比较	00	仅限软件比较
01	在比较时切换输出			
10	比较时的清晰输出			
11	在比较时设置输出			
1 倍	1 倍	10	边缘对齐 PWM	高真实脉冲（比较时输出清晰）
		X1		低真实脉冲（在比较时设置输出）
1	X 染色体	10	中心对齐 PWM	高真实脉冲（比较时的清晰输出）
		X1		低真实脉冲（在比较时设置输出）

如果在更改为输入捕获模式之前，关联的端口引脚在至少两个总线时钟周期内不稳定，则可能会意外获得边缘触发器的指示。通常，程序会在更改通道配置后清除状态标志 Uration 位和启用通道中断或使用状态标志之前，以避免任何意外行为。

14.3.5 计时器通道值寄存器 (TPMxCnVH: TPMxCnVL)

这些读/写寄存器包含输入捕获函数捕获的 TPM 计数器值或输出比较或 PWM 函数的输出比较值。通过重置清除通道值寄存器。

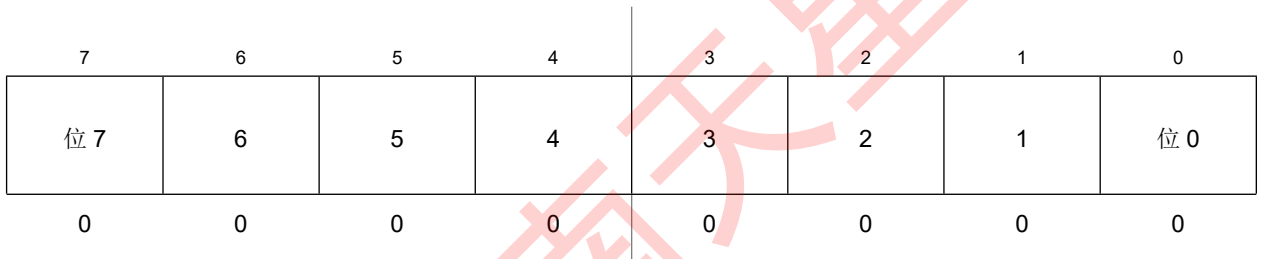


字母 R

罗马字母的第 23 个字母

调整

图 14-9。计时器 通道值寄存器高 (TPMxCnVH)



字母 R

罗马字母的第 23 个字母

调整

图 14-10。计时器 信道价值寄存器低 (TPMxCnVL)

在输入捕获模式下，读取任一字节 (TPMxCnVH 或 TPMxCnVL) 会将两个字节的內容锁定到缓冲区中，在那里它们保持锁定状态，直到读取另一个字节。当写入 TPMxCnSC 寄存器时，这种锁定机制也会重置 (成为未锁定)。

在输出比较或 PWM 模式下，写入任一字节 (TPMxCnVH 或 TPMxCnVL) 都会将值锁定到缓冲区中。当两个字节都写入时，它们作为相干的 16 位值传输到计时器通道值寄存器中。这个锁机械师 m 可以通过写入 TPMxCnSC 寄存器进行手动重置。

这种锁定机制允许以任一顺序进行连贯的 16 位写入，这对各种编译器实现都很友好。

14.4 功能描述

所有 TPM 函数都与一个 16 位主计数器相关联，该计数器允许灵活选择时钟源和预缩放除数。16 位模寄存器也与 TPM 中的主 16 位计数器相关联。每个 TPM 频道都是可选的带有 MCU 引脚和可屏蔽的中断功能。

TPM 具有中心对齐的 PWM 功能，由 TPMxSC 中的 CPWMS 控制位控制。当 CPWMS 设置为 1 时，计时器计数器 TPMxCNT 变为上/下计数器，相关 TPM 中的所有通道都充当中心对齐的 PWM 通道。当 CPWMS = 0，每个通道可以独立配置为在输入捕获、输出比较或缓冲边缘对齐 PWM 模式下运行。

以下部分描述了主要的 16 位计数器和每个计时器操作模式（输入捕获、输出比较、边缘对齐 PWM 和中心对齐 PWM）。由于引脚操作和中断活动的细节取决于操作模式，t 相关模式部分涵盖了这些主题。

14.4.1 柜台

所有计时器函数都基于主 16 位计数器（TPMxCNTH: TPMxCNTL）。本节讨论时钟源的选择、向上计数与向上/向下计数、计数结束溢出和手动计数器重置。

在任何 MCU 重置后，CLKSB:CLKSA = 0:0，因此没有选择时钟源，TPM 处于非活动状态。通常，CLKSB:CLKSA 将设置为 0:1，因此总线时钟驱动计时器计数器。TPM 的时钟源可以选择关闭，总线时钟（BUSCLK），固定系统时钟（XCLK）或外部输入。外部时钟选项允许的最大频率是总线速率的四分之一。参考第 14.3.1 节“计时器状态和控制寄存器（TPMxSC）”和表 14-3 有关时钟源选择的更多信息。

当微控制器处于活动后台模式时，TPM 会暂时暂停所有计数，直到微控制器恢复到正常的用户操作模式。在停止模式下，所有 TPM 时钟都停止；因此，在时钟恢复之前，TPM 被有效禁用。在等待模式下，TPM 继续正常运行。

主 16 位计数器有两种计数模式。当选择中心对齐 PWM（CPWMS = 1）时，计数器以向上/向下计数模式运行。否则，计数器作为一个简单的上计数器运行。作为上计数器，主要的 16 位计数器从 0 开始计数 X0000 通过其终端计数，然后继续 0x0000。终端计数为 0xFFFF 或 TPMxMODH:TPMxMODL 中的模量值。

当指定中心对齐 PWM 操作时，计数器从 0x0000 到其终端计数向上计数，然后向下计数到 0x0000，然后返回向上计数。0x0000 和终端计数值（TPMxMODH:TPMxMODL 中的值）都是 n 正常长度计数（一个计时器时钟周期长）。

中断标志和启用与主 16 位计数器相关联。计时器溢出标志（TOF）是软件可访问的指示，表明计时器计数器已溢出。启用信号在软件轮询（TOIE = 0）之间进行选择，其中没有硬生成重新中断，或中断驱动操作（TOIE = 1），其中每当 TOF 标志为 1 时，自动生成静态硬件中断。

导致 TOF 设置的条件取决于计数模式（向上或向上/向下）。在上计数模式下，主 16 位计数器从 0x0000 到 0xFFFF 计数，并在下一个计数时钟上溢出到 0x0000。TOF 设置为从 0xFFFF 到 0x0000 的过渡。当设置模量限制时，TOF 在从模量寄存器中设置的值过渡到 0x0000 时设置。当主 16 位计数器在向上/向下计数模式下运行时，当计数器发生可怕的变化时，会设置 TOF 标志从模量寄存器中设置的值和下一个较低的计数值过渡时。这对应于 PWM 周期的结束。（0x0000 计数值对应于周期的中心。）

由于 HCS08 MCU 是一个 8 位架构，因此在计时器计数器中内置了一个一致性机制，用于读取操作。每当读取计数器的任一字节（TPMxCNTH 或 TPMxCNTL）时，两个字节都被捕获到缓冲区中，所以当另一个字节是 r 时 Ead，该值将表示读取第一个字节时计数的另一个字节。计数器继续正常计数，但在读取旧计数的两个字节之前，无法从任一字节读取新值。

主计时器计数器可以随时手动重置，方法是将任何值写入计时器计数 TPMxCNTH 或 TPMxCNTL 的字节。以这种方式重置计数器也会重置一致性机制，以防在重置计数器之前只读取一个字节的计数器。

14.4.2 通道模式选择

如果 CPWMS = 0（未指定中心对齐 PWM 操作），通道 n 状态和控制寄存器中的 MSnB 和 MSnA 控制位决定相应通道的基本操作模式。选择包括输入捕获、输出 comp 是，以及缓冲边缘对齐的 PWM。

14.4.2.1 输入捕获模式

通过输入捕获功能，TPM 可以捕获外部事件发生的时间。当活动边缘出现在输入捕获通道的引脚上时，TPM 将 TPM 计数器的内容锁定到通道值寄存器（TPMxCnVH: TPMxCnVL）。上升边缘、下降边缘或任何边缘都可以选择作为触发输入捕获的活动边缘。

当读取 16 位捕获寄存器的任何一个字节时，两个字节都被锁定在缓冲区中，以支持连贯的 16 位访问，无论顺序如何。可以通过写入通道状态/控制寄存器（TPMxCnSC）来手动重置一致性序列。

输入捕获事件设置一个标志位（CHnF），该位可以选择生成 CPU 中断请求。

14.4.2.2 输出比较模式

通过输出比较功能，TPM 可以生成具有可编程位置、极性、持续时间和频率的定时脉冲。当计数器达到输出比较通道的通道值寄存器中的值时，TPM 可以设置、清除或切换通道引脚。

在输出比较模式下，只有在写入 16 位寄存器的两个 8 位字节后，值才会传输到相应的计时器通道值寄存器。此一致性序列可以通过写入通道状态/控制寄存器（TPMxCnSC）手动重置。

输出比较事件设置一个标志位（CHnF），该标志位可以选择生成 CPU 中断请求。

14.4.2.3 边缘对齐 PWM 模式

这种类型的 PWM 输出使用计时器计数器的正常向上计数模式 (CPWMS = 0)，当同一 TPM 中的其他通道配置为输入捕获或输出比较功能时可以使用。这个 PWM 信号的周期由 s 决定在模量寄存器 (TPMxMODH: TPMxMODL) 中。占空比由计时器通道值寄存器 (TPMxCnVH: TPMxCnVL) 中的设置决定。此 PWM 信号的极性由 ELSnA 控制位中的设置决定。占空比案例 0% 和 100% 是可能的。

图 14-11 显示，TPM 信道寄存器中的输出比较值决定了 PWM 信号的脉冲宽度 (占空比)。模量溢出和输出比较之间的时间是脉冲宽度。如果 ELSnA = 0，计数器溢出迫使 PWM 符号高和输出比较迫使 PWM 信号低。如果 ELSnA = 1，计数器溢出迫使 PWM 信号低，输出比较迫使 PWM 信号高。

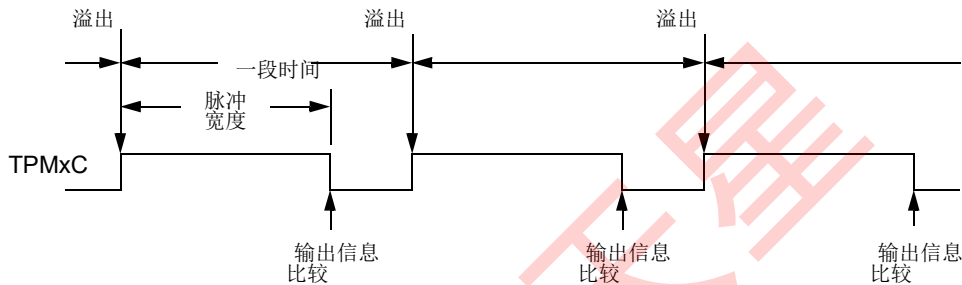


图 14-11. PWM 周期和脉冲宽度 (ELSnA = 0)

当信道值寄存器设置为 0x0000 时，占空比为 0%。通过将定时器通道值寄存器 (TPMxCnVH: TPMxCnVL) 设置为大于模量设置的值，可以实现 100% 的占空比。这意味着 setti 模量 ng 必须小于 0xFFFF 才能获得 100% 的占空比。

由于 HCS08 是一个 8 位 MCU 系列，因此计时器通道寄存器中的设置被缓冲，以确保连贯的 16 位更新，并避免意外的 PWM 脉冲宽度。写入寄存器、TPMxCnVH 或 TPMxCnVL，写入缓冲寄存器。钢边缘 PWM 模式，只有在写入 16 位寄存器的 8 位字节，并且 TPMxCNTH:TPMxCNTL 计数器中的值为 0x0000 后，值才会传输到相应的计时器信道寄存器。(新的工作周期不会生效，直到下一个完整时期。)

14.4.3 中心对齐 PWM 模式

这种类型的 PWM 输出使用计时器计数器的向上/向下计数模式 (CPWMS = 1)。

TPMxCnVH:TPMxCnVL 中的输出比较值决定了 PWM 信号的脉冲宽度 (占空比)，周期由 TPMxMODH:TPMxMODL 中的值决定。

TPMxMODH: TPMxMODL 必须保持在 0x0001 至 0x7FFF 的范围内，因为超出此范围的值可能会产生模棱两可的结果。ELSnA 将确定 CPWM 输出的极性。

$$\text{脉冲宽度} = 2 \times (\text{TPMxCnVH} - \text{TPMxCnVL}) \quad \text{Eqn. 14-1}$$

$$\begin{aligned} \text{周期} &= 2 \times (\text{TPMxMODH} - \text{TPMxMODL}); \\ \text{对于 TPMxMODH: TPMxMODL} &= 0x0001 - 0x7FFF \end{aligned} \quad \text{Eqn. 14-2}$$

如果信道值寄存器 $TPMxCnVH:TPMxCnVL$ 为零或负（设置 15 位），则占空比将为 0%。如果 $TPMxCnVH:TPMxCnVL$ 为正值（位 15 清晰）且大于（非零）模量设置，则占空比将为 100%，因为占空比永远不会发生。这意味着模量寄存器设置的可用周期范围为 $0x0001$ 至 $0x7FFE$ （如果不需要生成 100% 占空比，则为 $0x7FFF$ ）。这不是一个显著的限制，因为由此产生的时期比正常应用所需的时间长得多。

$TPMxMODH:TPMxMODL = 0x0000$ 是一种特殊情况，不得与中心对齐 PWM 模式一起使用。当 $CPWMS = 0$ 时，这种情况对应于从 $0x0000$ 到 $0xFFFF$ 自由运行的计数器，但当 $CPWMS = 1$ 时，计数器需要与模量的有效匹配在 $0x0000$ 以外的地方注册，以便将向上计数的方向从向上计数更改为向下计数。

图 14-12 显示 TPM 信道寄存器中的输出比较值（乘以 2），这决定了 CPWM 信号的脉冲宽度（占空比）。如果 $ELSnA = 0$ ，则计数时的比较匹配迫使 CPWM 输出信号降低，而比较匹配时倒计时迫使输出高。计数器计数，直到达到 $TPMxMODH:TPMxMODL$ 中的模数设置，然后倒计时，直到达到零。这设置的周期等于两倍 $TPMxMODH:TPMxMODL$ 。

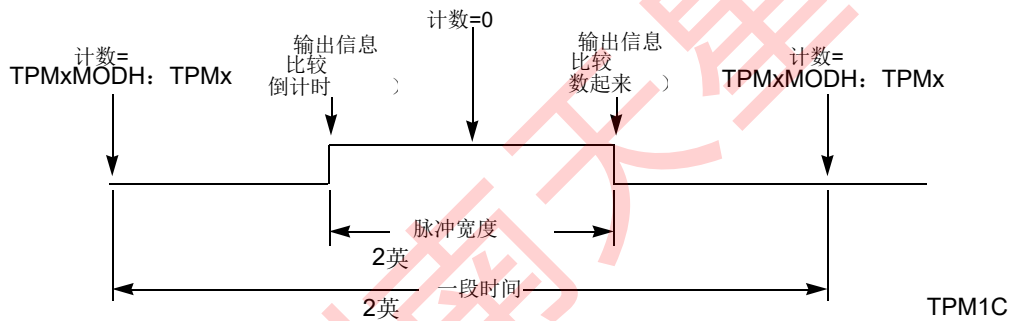


图 14-12. CPWM 周期和脉冲宽度 ($ELSnA = 0$)

中心对齐的 PWM 输出通常比边缘对齐的 PWM 产生更少的噪音，因为在同一系统时钟边缘排列的 I/O 引脚转换更少。某些类型的电机驱动器也需要这种类型的 PWM。

由于 HCS08 是一个 8 位 MCU 系列，因此计时器通道寄存器中的设置被缓冲，以确保连贯的 16 位更新，并避免意外的 PWM 脉冲宽度。写给任何寄存器， $TPMxMODH$ 、 $TPMxMODL$ 、 $TPMxCnVH$ 和 $TPMxCnVL$ 实际上写入缓冲寄存器。只有在写入 16 位寄存器的 8 位字节和计时器计数器溢出（反向）后，值才会传输到相应的计时器通道寄存器在模量寄存器中的终端计数结束时，从向上计数到向下计数的方向。此 $TPMxCNT$ 溢出要求仅适用于 PWM 通道，不适用于输出比较。

或者，当 $TPMxCnTH:TPMxCnTL = TPMxMODH:TPMxMODL$ 时，TPM 可以生成 TOF 在这个计数结束时打断。用户可以选择重新加载任意数量的 PWM 缓冲区，它们将在新周期开始时同时更新。

写入 $TPMxSC$ 会取消写入 $TPMxMODH$ 和/或 $TPMxMODL$ 的任何值，并重置模寄存器的一致性机制。写入 $TPMxCnSC$ 会取消写入到信道值寄存器的任何值，并重置 $TPMxCnVH:TPMxCnVL$ 的相对性机制。

14.5 TPM 中断

TPM 为主计数器溢出生成可选中断，并为每个通道生成中断。通道中断的含义取决于每个通道的操作模式。如果通道配置为输入捕获，中断标志每次识别选定的输入捕获边缘时都设置。如果通道配置为输出比较或 PWM 模式，则每次主计时器计数器与 16 位通道值寄存器中的值匹配时，都会设置中断标志。看到[重置、中断和系统配置](#)绝对中断向量地址、优先级和本地中断掩码控制位的章节。

对于 TPM 中的每个中断源，在识别计时器溢出、通道输入捕获或输出比较事件等中断条件时设置一个标志位。软件可以读取（投票）此标志，以验证操作是否已发生，或关联的启用位（TOIE 或 CHnIE）可以设置为启用硬件中断生成。当设置中断启用位时，每当关联的中断标志等于 1 时，将生成静态中断。这是用户软件的责任在从中断服务例程返回之前，执行一系列步骤来清除中断标志。

14.5.1 清除计时器中断标志

TPM 中断标志由一个两步进程清除，该进程包括在设置（1）时读取标志位，然后向位写入 0。如果在这两个步骤之间检测到新事件，序列将重置，中断标志将保持设置第二步，避免错过新事件的可能性。

14.5.2 计时器溢出中断描述

导致 TOF 设置的条件取决于计数模式（向上或向上/向下）。在上计数模式下，16 位计时器计数器从 0x0000 到 0xFFFF 计数，并在下一个计数时钟上溢出到 0x0000。TOF 设置为从 0xFFFF 到 0x0000 的过渡。当设置模量限制时，TOF 在从模量寄存器中设置的值过渡到 0x0000 时设置。当计数器在向上/向下计数模式下运行时，当计数器改变方向时，会设置 TOF 标志从模量寄存器中设置的值和下一个较低的计数值过渡。这对应于到 PWM 周期结束。（0x0000 计数值对应于周期的中心。）

14.5.3 频道事件中断描述

通道中断的含义取决于通道的当前模式（输入捕获、输出比较、边缘对齐 PWM 或中心对齐 PWM）。

当通道配置为输入捕获通道时，ELSnB:ELSnA 控制位选择上升边缘、下降边缘、任何边缘或无边缘（关闭）作为触发输入捕获事件的边缘。当检测到所选边缘时，将设置中断标志。该标志由中描述的 2 步序列清除[第 14.5.1 节“清除计时器中断标志。”](#)

当通道配置为输出比较通道时，每次主计时器计数器与通道值寄存器中的 16 位值匹配时，都会设置中断标志。该标志由中描述的 2 步序列清除[第 14.5.1 节“清除计时器中断标志。”](#)

14.5.4 PWM 结束工作周期事件

对于为 PWM 操作配置的通道，有两种可能性：

- 当通道配置为边缘对齐 PWM 时，当计时器计数器与标志着有效占空周期结束的通道值寄存器匹配时，将设置通道标志。
- 当通道配置为中心对齐 PWM 时，计时器计数在每个 PWM 周期中与通道值寄存器匹配两次。在这个 CPWM 案例中，通道标志设置在主动占空比的开始和结束时，即计时器计数器与通道值寄存器匹配。

该标志由中描述的 2 步序列清除第 14.5.1 节“清除计时器中断标志。”

深圳南天星

第 15 章 开发支持

15.1 简单介绍

RS08 系列的开发支持系统包括 RS08 后台调试控制器（BDC）。

BDC 为目标 MCU 提供了一个单线调试接口。该界面为编程片上 FLASH 和其他非易失性存储器提供了一种方便的方法。此外，BDC 是开发的主要调试接口，并允许非 intrusive 访问内存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。

在 RS08 系列中，外部引脚上没有地址和数据总线信号。调试通过单线后台调试接口输入目标 MCU 的命令完成，包括在不使用重置引脚的情况下重置设备。

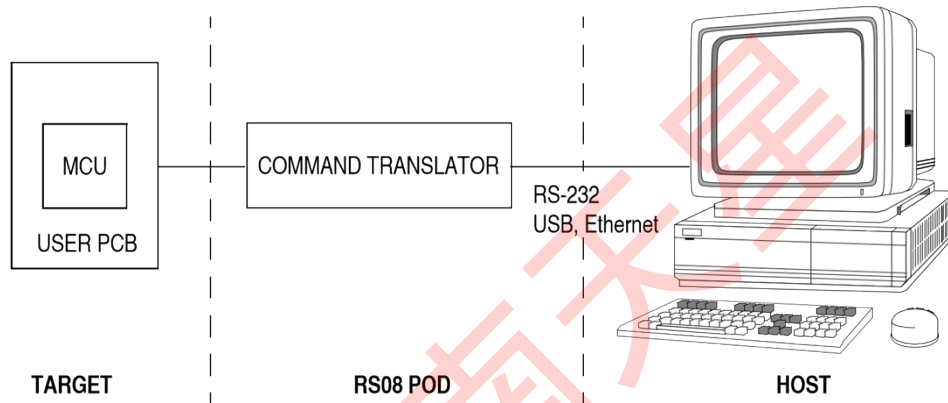


图 15-1。将 MCU 连接到主机进行调试

15.2 特点

RS08 后台调试控制器（BDC）的功能包括：

- 使用单个引脚进行后台调试串行通信
- 非侵入用户内存资源；BDC 寄存器不位于内存映射中
- SYNC 命令来确定目标通信速率
- 非侵入性命令允许在 CPU 运行用户代码时访问内存资源，而不会停止应用程序
- 用于 CPU 寄存器访问的主动后台模式命令
- GO 和 TRACE1 命令
- BACKGROUND 命令可以从等待或停止模式唤醒 CPU

第 15 章 发展支持

- BDC_RESET 命令允许主机在不使用重置引脚的情况下重置 MCU

- 一个内置在 BDC 的硬件地址断点
- 如果启用 BDM 在 CPU 处于停止模式时允许调试，则 RS08 时钟源在停止模式下运行
- COP 监督机构在活动后台模式下暂停

15.3 RS08 后台调试控制器 (BDC)

RS08 系列中的所有 MCU 都包含一个单线背景调试接口，该接口支持片上非易失性内存的电路内编程和复杂的调试功能。与早期 8 位 MCU 上的调试接口不同，这个调试系统提供 S 用于最大限度地减少对正常应用程序资源的干扰。它不使用内存映射中的任何用户内存或位置。它只需要使用仅输出的 BKGD 引脚。此引脚将与简单的仅用户输出功能共享（通常是端口，比较器输出等），可以在正常用户模式下轻松调试。

RS08 BDM 命令分为两组：

- 活动后台模式命令要求目标 MCU 处于活动后台模式（用户程序未运行）。BACKGROUND 命令导致目标 MCU 进入活动后台模式。活动后台模式命令允许 CPU 寄存器 s 被读取或写入，并允许用户一次跟踪一个（TRACE1）用户指令，或从活动后台模式转到用户程序。
- 即使在用户程序运行时，也可以随时执行非侵入性命令。非侵入性命令允许用户在后台调试控制器（BDC）内读取或写入 MCU 内存位置或访问状态和控制寄存器。

通常，一个相对简单的接口 pod 用于将命令从主机转换为自定义串行接口到单线后台调试系统的命令。根据开发工具供应商的不同，此接口 pod 可能会使用标准 RS-232 串行端口、并行打印机端口或其他类型的通信，如以太网或通用串行总线（USB），用于在主机 PC 和 pod 之间进行通信。

图 15-2 显示 RS08 BDM pod 连接的标准标头。pod 是一种小型接口设备，将个人电脑等主机连接到目标 RS08 系统。BKGD 和 GND 是与目标 MCU 通信所需的最低连接。连接器中包含伪开漏 RESET 信号，以允许主机强制或监控（如果 RESET 作为输出可用）目标系统重置的直接硬件方法。

RS08 BDM 吊舱提供 V_{代表} 当需要电路内编程时，电压到 RS08 MCU。

V_{代表} 来自 pod 的连接与 RESET 共享，如图所示图 15-2。对于 V_{代表} 要求请参阅电气附录中的 FLASH 规范。

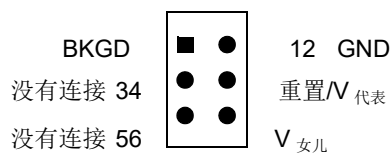


图 15-2. 标准 RS08 BDM 工具连接器

后台调试控制器（BDC）串行通信使用 M68HC12 微控制器系列中首次引入的自定义串行协议。该协议要求主机知道通信时钟速率，该速率由目标 BDC 时钟速率决定。如果主机试图与具有未知 BDC 时钟速率的目标 MCU 通信，则可能会向目标 MCU 发送 SYNC 命令，以请求定时同步响应信号，主机可以从中确定正确的通信在速度上。

对于 RS08 MCU，BDC 时钟与 MCU 总线时钟的频率相同。有关通信协议的详细说明，请参阅第 15.3.2 节“通信细节。”

15.3.1 BKGD 引脚描述

BKGD 是单线背景调试接口引脚。BKGD 是一个伪开排水销，包含片上拉销，因此它不需要外部上拉电阻。与典型的开漏引脚不同，外部电阻电容器（RC）时间常数 τ 这个引脚受外部电容的影响，在信号上升时间中几乎没有作用。自定义协议提供了简短的、主动驱动的加速脉冲，以在此引脚上强制快速上升时间，而不会冒有害的驱动级别冲突的风险。字母 REfer 到第 15.3.2 节“通信细节”，了解更多详情。

该引脚的主要功能是后台调试命令和数据的双向串行通信。在重置期间，此引脚在以活动后台模式启动和运行应用程序的正常用户模式之间进行选择。这个别针也是我们 ed 请求定时同步响应脉冲，以允许主机开发工具确定目标 BDC 时钟频率。

通过控制 BKGD 引脚并强制重置 MCU（发布 BDC_RESET 命令，或通过开机重置（POR）），主机可以强制目标系统重置为活动后台模式，而不是启动用户应用程序。这很有用控制目标 MCU，其 FLASH 程序内存尚未使用用户应用程序编程。

当没有调试器 pod 连接到 6 针 BDM 接口连接器时，BKGD 上的内部上拉决定了正常操作模式。

在某些 RS08 设备上，BKGD 引脚可能与替代的仅输出功能共享。要支持 BDM 调试，用户必须禁用此替代功能。替代功能的调试必须在不使用 BDM 的正常用户模式下完成。

15.3.2 沟通细节

BDC 串行接口要求主机在 BKGD 引脚上生成下降边缘，以指示每个位时间的开始。无论数据是传输还是接收，主机都会提供这种下降的边缘。

第 15 章 发展支持

BDC 串行通信协议要求主机知道目标 BDC 时钟速度。命令和数据以每位 16 个 BDC 时钟周期发送最显著位优先（MSB 优先）。如果在主机的下降边缘之间发生 512 BDC 时钟周期，接口将计时。此超时发生时正在进行的任何 BDC 命令都会中止，而不会影响目标 MCU 系统的内存或操作模式。

图 15-3 显示将逻辑 1 或 0 传输到目标 MCU 的 BKGD 引脚的外部主机。主机与目标异步，因此从主机生成的下降边缘到目标感知位时间开始的地方有 0 比 1 的周期延迟。字母 T 在目标 BDC 时

钟周期之后，目标在 BKGD 引脚上感应到位电平。通常，主机在主机到目标传输期间主动驱动伪开排 BKGD 引脚，以加快上升边缘的速度。因为目标不驱动 BKGD p 在主机到目标期间，没有必要将线路视为开放漏电信号。

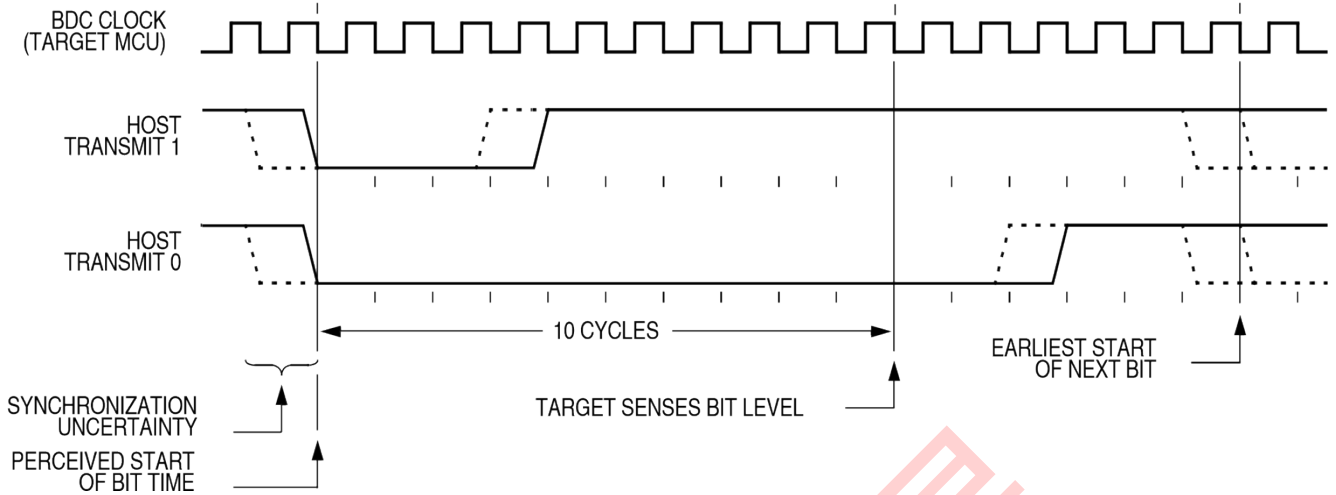


图 15-3. BDC 主机到目标串行位定时

图 15-4 显示从目标 MCU 接收逻辑 1 的主机。由于主机与目标异步，从 BKGD 上主机生成的下降边缘到目标中位时间的感知开始，有一个 0 比 1 的周期延迟。主机持有 BKGD 引脚低到足以让目标识别它（至少两个目标 BDC 周期）。在目标在感知到位时间开始七个周期后，在目标驱动短暂的有源高加速脉冲之前，主机必须释放低驱动器。主机必须抽样位水平在开始位时间后大约 10 个周期。

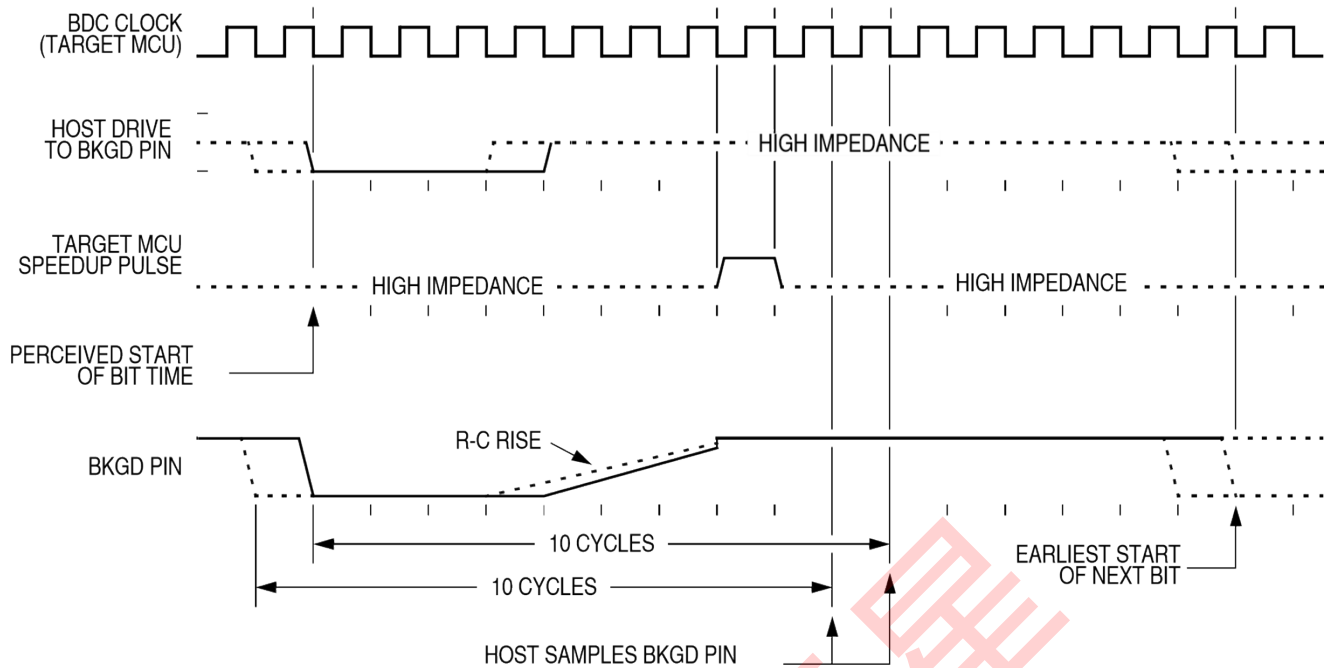


图 15-4。BDC 目标到主机串行位定时（逻辑 1）

图 15-5 显示主机从目标 MCU 接收逻辑 0。由于主机与目标异步，从 BKGD 上主机生成的下降边缘到目标感知的位时间开始，有一个 0 比 1 的周期延迟。主持人发起有点时间，但目标完成了它。因为目标希望主机接收逻辑 0，所以它将 BKGD 引脚低驱动 13 个 BDC 时钟周期，然后短暂地将其高驱动以加速上升边缘。主机对大约 10 cy 的位级进行采样开始位时间后 cles。

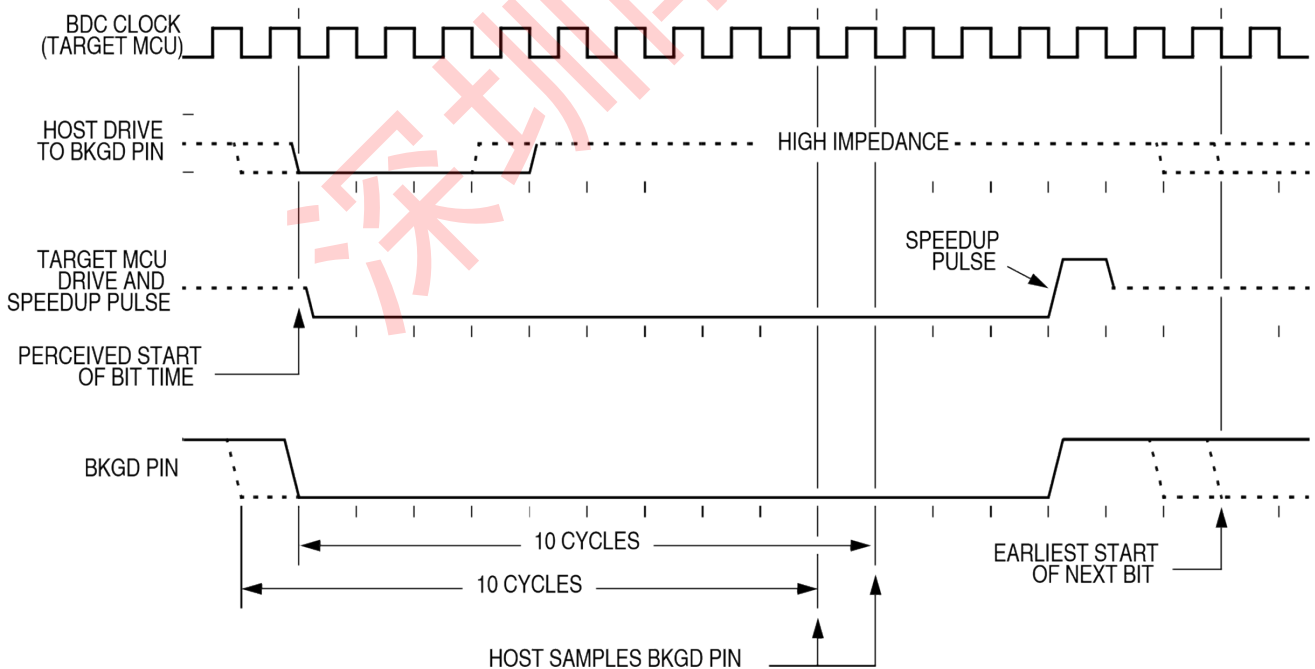


图 15-5。BDM 目标到主机串行位定时（逻辑 0）

15.3.3 SYNC 和串行通信超时

主机通过在 BKGD 引脚上生成下降边缘来启动主机到目标的串行传输。如果 BKGD 保持在低水平超过 128 个目标时钟周期，目标就会明白发出了 SYNC 命令。在这种情况下，目标将保持等待 Ng for a rising edge on BKGD to answer the SYNC request pulse.如果未检测到上升边缘，目标将无限期地等待，没有任何超时限制。当 BKGD 在有效的 SYNC 请求后出现上升边缘时，BDC 将驱动 BKGD 引脚低，正好为 128 个 BDC 周期。

现在考虑一下主机在 128 个周期之前将 BKGD 返回到逻辑 1 的情况。这被解释为有效的位传输，而不是 SYNC 请求。目标将继续等待另一个下降边缘，标志着新钻头的开始。然而，如果一个自上次坠落边缘以来，目标在 512 个时钟周期内没有检测到新的坠落边缘，发生超时，当前命令被丢弃，而不会影响内存或 MCU 的操作模式。这被称为对 BD 的软重置 C。

如果发出了读取命令，但没有在 512 个串行时钟周期内检索到数据，则将发生软重置，导致该命令被忽略。超时发生后，数据无法检索。软重置也用于和 READ_BLOCK 或 WRITE_BLOCK 命令。

以下描述了主机在没有目标超时或将位解释为 SYNC 命令的情况下保证逻辑 1 或 0 位传输的实际位时要求：

- 要发送逻辑 0，BKGD 必须保持在最低水平，至少 12 个 BDC 周期和最多 511 个 BDC 周期，但命令序列的第一个位除外，该命令序列将作为 SYNC 请求被检测。
- 要发送逻辑 1，BKGD 必须保持至少四个 BDC 周期的低位，在第八个周期之前释放，并至少保持高位，直到第 16 个 BDC 周期。
- 后续位必须在上次发送的 512 个 BDC 周期内发生。

15.4 BDC 寄存器和控制位

BDC 包含两个非 CPU 可访问的寄存器：

- BDC 状态和控制寄存器 (BDCSCR) 是一个 8 位寄存器，包含后台调试控制器的控制和状态位。
- BDC 断点寄存器 (BDCBKPT) 持有 16 位断点匹配地址。

这些寄存器使用专用的串行 BDC 命令访问，并且不位于目标 MCU 的内存空间中（因此它们没有地址，用户程序无法访问）。

BDCSCR 中的一些位有写入限制；否则，这些寄存器可以随时读取或写入。例如，当 MCU 处于活动后台模式时，ENBDM 控制位可能无法写入。这防止了模棱两可的条件当 MCU 已经处于活动后台模式时，控制位禁止活动后台模式。此外，状态位 (BDMACT、WS 和 WSF) 是只读状态指示器，永远不能用 WRITE_CONTROL 串行 BDC 命令写入。

15.4.1 BDC 状态和控制登记册 (BDCSCR)

此寄存器可以通过串行 BDC 命令 (READ_STATUS 和 WRITE_CONTROL) 读取或写入, 但用户程序无法访问, 因为它不位于 MCU 的正常内存图中。

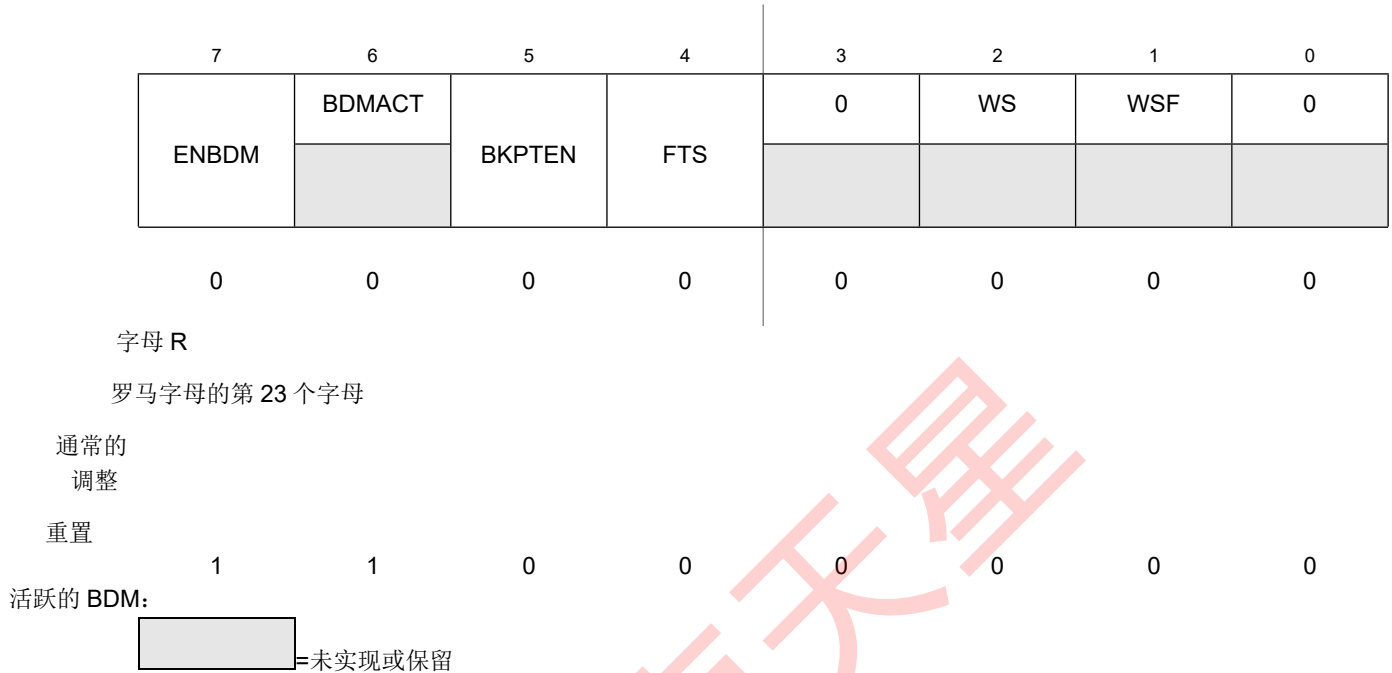


图 15-6. BDC 状态和控制登记册 (BDCSCR)

表 15-1. BDCSCR 注册字段描述

字段	描述
7 ENBDM	启用 BDM (允许活动后台模式) — 通常, 此位由调试主机在调试会话开始后不久或每当调试主机重置目标并保持 1 时写入 1, 直到正常重置清除它。如果应用程序可以进入停止模式, 如果需要调试功能, 则需要设置此位。 0 BDM 无法激活 (仍然允许非侵入性命令)。 1 BDM 可以激活, 以允许活动后台模式命令。
6 BDMACT	后台模式活动状态 — 这是一个只读状态位。 0 BDM 未激活 (用户应用程序正在运行)。 1 BDM 处于活动状态并等待串行命令。

表 15-1. BDCSCR 寄存器字段描述 (续)

字段	描述
5 BKPTEN	BDC 断点启用 — 如果此位清晰, BDC 断点将被禁用, FTS (强制标签选择) 控制位和 BDCBKPT 匹配寄存器将被忽略 0 BDC 断点禁用。 启用 1 个 BDC 断点。

调整

图 15-7. BDC 断点匹配寄存器 (BDCBKPT)

15.5 RS08 BDC 命令

BDC 命令从主机串行发送到目标 MCU 的 BKGD 引脚。所有命令和数据都使用自定义 BDC 通信协议先发送 MSB。活动后台模式命令要求目标 MCU 当前处于活动状态无论目标 MCU 处于活动后台模式还是运行用户应用程序程序，都可以随时发出非侵入性命令。

表 15-2 显示所有 RS08 BDC 命令，其编码结构的速记描述，以及每个命令的含义。

编码结构命名法

以下术语用于表 15-2 描述 BDC 命令的编码结构。

	命令从主机到目标方向的 8 位命令代码开始 (最重要的位优先)
/	= 分隔命令的部分
D	= 延迟 16 到 511 个目标 BDC 时钟周期
软重置	= 从上次主机下降边缘延迟至少 512 个 BDC 时钟周期
AAAA	= 主机到目标方向的 16 位地址 ¹
RD	= 目标到主机方向的八位读取数据
WD	= 八位在主机到目标方向的写入数据
RD16	= 目标到主机方向的 16 位读取数据
WD16	= 主机到目标方向的 16 位写入数据
纳粹党 卫军	= 目标到主机方向 (STATUS) 的 BDCSCR 内容
立方厘 米	= BDCSCR 在主机到目标方向 (CONTROL) 的八位写入数据
RBKP	= 目标到主机方向的 16 位读取数据 (来自 BDCBKPT 断点寄存器)
WBKP	= 主机到目标方向的 16 位写入数据 (用于 BDCBKPT 断点寄存器)

表 15-2. RS08 BDC 命令摘要

命令助记符	活跃背景 模式/ 非侵入性	编码 结构	描述
-------	---------------------	----------	----

¹. The RS08 CPU uses only 14 bits of address and occupies the lower 14 bits of the 16-bit AAAA address field. The values of address bits 15 and 14 in AAAA are truncated and thus do not matter.

同步	非侵入性	不适用的 ¹	请求定时参考脉冲以确定目标 BDC 通信速度
BDC_RESET	任何 CPU 模式	18 ²	请求 MCU 重置
背景	非侵入性	90/天	如果启用，请进入活动后台模式（如果 ENBDM 位等于 0，请忽略）
阅读_状态	非侵入性	E4/SS	从 BDCSCR 读取 BDC 状态
写_控制	非侵入性	C4/CC	在 BDCSCR 中写入 BDC 控件
阅读_字节	非侵入性	E0/AAAA/d/RD	从目标内存中读取一个字节
阅读_字节_WS	非侵入性	E1/AAAA/d/SS/RD	阅读一个字节并报告状态
写_字节	非侵入性	C0/AAAA/WD/d	写入一个字节到目标内存
写_字节_WS	非侵入性	C1/AAAA/WD/d/SS	写一个字节并报告状态
阅读_BKPT	非侵入性	E2/RBKP	读取 BDCBKPT 断点寄存器
写_BKPT	非侵入性	C2/WBKP	编写 BDCBKPT 断点寄存器
围棋	活动后台模式	08/d	从当前 PC 中的地址开始执行用户应用程序
痕迹 1	活动后台模式	10/d	在地址处跟踪一个用户指令在 PC 中，然后返回到活动后台模式
阅读_BLOCK	活动后台模式	80/AAAA/d/RD ³	从 AAAA 开始从目标内存中读取数据块，直到检测到软重置
写_块	活动后台模式	88/AAAA/WD/d ⁴	从 AAAA 开始将数据块写入目标内存，直到检测到软重置
阅读_A	活动后台模式	68/d/RD	读取累加器 (A)

表 15-2。RS08 BDC 指挥摘要 (续)

命令助记符	活跃背景 模式/ 非侵入性	编码 结构	描述
写_A	活动后台模式	48/WD/d	写累加器 (A)
阅读_CCR_PC	活动后台模式	6B/d/RD16 ⁵	读取 CCR 位 z, c 与 14 位程序计数器 (PC) RD16=zc:PC

写_CCR_PC	活动后台模式	4B/WD16/d ⁶	写入 CCR 位 z, c 与 14 位程序计数器 (PC) WD16=z:c: PC
阅读_SPC	活动后台模式	6F/d/RD16 ⁷	阅读 14 位影子程序计数器 (SPC) RD16=0:0:SPC
写_SPC	活动后台模式	4F/WD16/d ⁸	写入 14 位影子程序计数器 (SPC) WD16 = x:x:SPC, “x”显示的两个最重要的 位被目标忽略

¹ SYNC 命令是一个没有命令代码的特殊操作。

² 18 是 TAGGO 的 HCS08 BDC 指挥部。

³ 每个 RD 需要主机读取数据字节和下次读取之间的延迟, 当目标检测到软重置时, 命令结束。

⁴ 每个 WD 需要在主机写入数据字节和下一个字节之间延迟, 当目标检测到软重置时, 命令结束。

⁵ HCS08 BDC 有单独的 READ_CCR 和 READ_PC 命令, RS08 BDC 合并了此命令。

⁶ HCS08 BDC 有单独的 WRITE_CCR 和 WRITE_PC 命令, RS08 BDC 结合了这些命令。

⁷ 6F 是 HCS08 BDC 的 READ_SP (读取堆栈指针)。

⁸ 4F 是 HCS08 BDC 的 WRITE_SP (写入堆栈指针)。

深圳南天星



深圳南天星

如何联系我们:

主页: Freescale.com

网络支持:

Freescale.com/support

本文档中的信息仅用于使系统和软件实现者能够使用

Freescale 产品。本协议不授予任何明示或暗示的版权许可, 以根据本文件中的信息设计或制造任何集成电路。Freescale 保留对此处任何产品进行更改的权利, 恕不另行通知。

Freescale 对其产品是否适合任何特定目的不作任何保证、陈述或保证, Freescale 也不承担因应用或使用任何产品或电路而产生的任何责任, 特别是披露承担任何和所有责任, 包括但不限于间接或附带损害。

Freescale 数据表和/或规格中可能提供的“典型”参数可以而且确实在不同的应用中有所不同, 实际性能可能有所不同随着时间的推移。所有操作参数, 包括“典型值”, 都必须由客户的技术专家针对每个客户应用程序进行验证。

Freescale 不根据其专利权或他人权利传达任何许可。Freescale 销售产品符合标准销售条款和条件, 可在以下地址找到:

freescale.com/SalesTermsandConditions。

Freescale 和 Freescale 徽标是 Freescale Semiconductor, Inc.,

Reg. 的商标。美国帕特。 & Tm。关闭。所有其他产品或服务名称均为其各自所有者的财产。

© 2008-2015 飞思卡尔半导体公司

文件编号 MC9S08KA8RM 修订版
4, 2015 年 5 月

深圳南天星

