



MC9RS08KA2 MC9RS08KA1

数据手册

RS08
Microcontrollers

MC9RS08KA2
Rev.2
12/2006

freescale.com



MC9RS08KA2 特性

8- 位 RS08 中央处理器 (CPU)

- 简化的 S08 指令集，增加了高性能指令
 - LDA、STA 和 CLR 指令 支持短型寻址模式：\$0000 至 \$001F 能被单字节指令访问
 - ADD、SUB、INC 和 DEC 指令 支持小型寻址模式；地址 \$0000-\$000F 能被具有精简指令周期的单字节指令访问
 - SPC 寄存器指令：SHA 和 SLA
- 等待中断指示
- 通过 D[X] 和 X 寄存器变址寻址
- 通过分页区访问整个存储器

存储器

- 片内 Flash EEPROM
 - MC9RS08KA2: 2048 字节
 - MC9RS08KA1: 1024 字节
- 63 字节 RAM

省电模式

- 等待和停止模式
- 在省电模式下，通过实时中断 (RTI)、KBI、ACMP 唤醒

时钟源

- **ICS** — 可调整的 20-MHz 内部时钟源
 - 最大到 10MHz 内部总线工作频率
 - 0.2% 可调精度,在全温度和电压范围 2% 精度

系统保护

- 计算机正常工作 (COP) 复位可以不依赖总线时钟正常工作
- 低电压检测产生复位或唤醒停止状态

外设

- **MTIM** — 8 位模定时器
- **ACMP** — 模拟比较器
 - 轨对轨电压工作
 - 可选择内部固定的待隙参考电压做比较
 - 能在停止模式下工作
- **KBI** — 键盘中断口
 - 6 管脚封装有 3 个 KBI
 - 8 管脚封装有 5 个 KBI

开发支持

- 背景调试系统
- 在线调试允许设置单断点

封装选择

- 6 管脚 DFN 封装
 - 2 个输入 / 输出脚
 - 1 个输入脚
 - 1 个输出脚
- 8 管脚 PDIP 封装
 - 4 个输入 / 输出脚
 - 1 个输入脚
 - 1 个输出脚
- 8 管脚 SOIC 封装
 - 4 个输入 / 输出脚
 - 1 个输入脚
 - 1 个输出脚

MC9RS08KA2 数据手册

包括： MC9RS08KA2
MC9RS08KA1

MC9RS08KA2
Rev. 2
12/2006



修订历史

网上文档的版本是最新的。为了获得最新版的信息，可以查阅：

<http://freescale.com>

下面修订本历史表总结在这个文档中的变化。

修订编号	修订日期	变化描述
1.0	4/2006	最初公布的版本
2	12/2006	增加 MC9RS08KA1

此文档包含新产品开发的信息。飞思卡尔保留改变或停止此产品而不布告的权力。

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

Freescale, and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
© Freescale Semiconductor, Inc., 2007. All rights reserved.

MC9RS08KA2 Data Sheet, Rev. 2.0

目录

第一章	MC9RS08KA2 产品概述	15
第二章	管脚和连接.....	17
第三章	工作模式	21
第四章	存储器.....	25
第五章	复位、中断和系统控制.....	35
第六章	并行输入 / 输出控制	45
第七章	键盘中断 (RS08KBIV1)	51
第八章	中央处理器 (RS08CPUV1)	57
第九章	内部时钟源 (RS08ICSV1).....	73
第十章	模拟比较器 (RS08ACMPV1)	81
第十一章	模定时器 (RS08MTIMV1)	87
第十二章	开发支持	95
附录 A	电气特性	107
附录 B	订单信息和机械图.....	123

目录表

章节	标题	页码
第 1 章		
MC9RS08KA2 产品概述		
1.1	概述	15
1.2	MCU 方框图	15
1.3	系统时钟描述	16
第 2 章		
管脚和连接		
2.1	介绍	17
2.2	芯片管脚分配	17
2.3	推荐的系统连接	18
2.4	管脚详述	18
2.4.1	电源	18
2.4.2	PTA2/KBIP2/TCLK/RESET/V _{PP}	19
2.4.3	PTA3/ACMPO/BKGD/MS	19
2.4.4	通用 I/O 和外设端口	19
第 3 章		
工作模式		
3.1	介绍	21
3.2	特性	21
3.3	运行模式	21
3.4	主动背景模式	21
3.5	等待模式	22
3.6	停止模式	23
3.6.1	在停止模式下主动 BDM 使能	24
3.6.2	在停止模式下 LVD 使能	24
第 4 章		
存储器		
4.1	存储器映象	25
4.2	无效存储器	26
4.3	变址 / 间接寻址	27
4.4	RAM 和寄存器 地址和 位分配	27
4.5	RAM	29
4.6	Flash	29
4.6.1	特点	29
4.6.2	Flash 编程过程	30
4.6.3	Flash 块擦除操作	30

4.6.4	安全	31
4.7	Flash 寄存器和控制位	32
4.7.1	Flash 选项寄存器 (FOPT 和 NVOPT)	32
4.7.2	Flash 控制寄存器 (FLCR)	33
4.8	页选择寄存器 (PAGESEL)	33

第 5 章 复位、中断和常用的系统控制

5.1	介绍	35
5.2	特点	35
5.3	MCU 复位	35
5.4	计算机正常工作 (COP) 看门狗	36
5.5	中断	36
5.6	低电压检测 (LVD) 系统	37
5.6.1	上电复位操作	37
5.6.2	低电压复位操作	37
5.6.3	低电压中断操作	37
5.7	实时中断 (RTI)	37
5.8	复位、中断和系统控制寄存器和控制位	38
5.8.1	系统复位状态寄存器 (SRS)	38
5.8.2	系统选项寄存器 (SOPT)	39
5.8.3	系统设备辨别寄存器 (SDIDH, SDIDL)	40
5.8.4	系统实时中断状态和控制寄存器 (SRTISC)	41
5.8.5	系统电源管理状态和控制寄存器 (SPMSC1)	43
5.8.6	系统中断等待寄存器 (SIP1)	44

第 6 章 并行输入 / 输出控制

6.1	低功耗模式管脚状态	46
6.2	并行输入 / 输出口寄存器	46
6.2.1	端口 A 寄存器	46
6.3	管脚控制寄存器	47
6.3.1	端口 A 管脚控制寄存器	47
6.3.1.1	内部上拉使能	47
6.3.1.2	上拉 / 下拉控制	48
6.3.1.3	输出斜率控制使能	48

第 7 章 键盘中断 (RS08KBIV1)

7.1	介绍	51
7.1.1	特点	51
7.1.2	工作模式	52
7.1.2.1	等待模式	52
7.1.2.2	停止模式	52
7.1.2.3	主动背景模式	52

7.1.3	方框图	52
7.2	外部信号描述	52
7.3	寄存器定义	53
7.3.1	键盘中断状态和控制寄存器 (KBISC)	53
7.3.2	键盘中断管脚使能寄存器 (KBIPE)	54
7.3.3	键盘中断边沿选择寄存器 (KBIES)	54
7.4	功能描述	55
7.4.1	仅边沿触发	55
7.4.2	边沿和电平触发	55
7.4.3	键盘中断上拉 / 下拉设备	55
7.4.4	键盘中断初始化	55

第 8 章 中央处理器单元 (RS08CPUV1)

8.1	介绍	57
8.2	程序设计器模型和 CPU 寄存器	57
8.2.1	累加器 (A)	58
8.2.2	程序计数器 (PC)	59
8.2.3	影子程序计数器 (SPC)	59
8.2.4	条件代码寄存器 (CCR)	59
8.2.5	变址数据寄存器 (D[X])	60
8.2.6	变址寄存器 (X)	60
8.2.7	页选择寄存器 (PAGESEL)	61
8.3	寻址方式	61
8.3.1	隐含寻址方式 (INH)	61
8.3.2	相对寻址方式 (REL)	61
8.3.3	立即寻址方式 (IMM)	62
8.3.4	小型寻址方式 (TNY)	62
8.3.5	短型寻址方式 (SRT)	63
8.3.6	直接寻址方式 (DIR)	63
8.3.7	扩展寻址方式 (EXT)	63
8.3.8	变址寻址方式 (IX, 通过伪指令实现)	63
8.4	特别操作	63
8.4.1	复位顺序	64
8.4.2	中断	64
8.4.3	等待和停止模式	64
8.4.4	主动背景模式	64
8.5	总指令表	65

第 9 章 内部时钟源 (RS08ICSV1)

9.1	介绍	73
9.2	介绍	74
	9.2.1 特点	74
	9.2.2 工作模式	74
	9.2.2.1 FLL 使能内部 (FEI)	74
	9.2.2.2 FLL 旁路内部 (FBI)	74
	9.2.2.3 FLL 旁路内部低功耗 (FBILP)	74
	9.2.2.4 停止 (STOP)	75
	9.2.3 方框图	75
9.3	外部信号描述	75
9.4	寄存器定义	76
	9.4.1 ICS 控制寄存器 1 (ICSC1)	76
	9.4.2 ICS 控制寄存器 2 (ICSC2)	77
	9.4.3 ICS 调整寄存器 (ICSTRM)	77
	9.4.4 ICS 状态和控制 (ICSSC)	78
9.5	功能描述	78
	9.5.1 工作模式	78
	9.5.1.1 FLL 使能内部 (FEI)	79
	9.5.1.2 FLL 旁路内部 (FBI)	79
	9.5.1.3 FLL 旁路内部低功耗 (FBILP)	79
	9.5.1.4 停止	79
	9.5.2 模式切换	79
	9.5.3 总线频率分频器	79
	9.5.4 低功耗位使用	79
	9.5.5 内部参考时钟	80
	9.5.6 固定频率时钟	80

第 10 章 模拟比较器 (RS08ACMPV1)

10.1	介绍	81
	10.1.1 特点	82
	10.1.2 工作模式	82
	10.1.2.1 在等待模式下工作	82
	10.1.2.2 在停止模式下工作	82
	10.1.2.3 在主动背景模式下工作	82
	10.1.3 方框图	82
10.2	外部信号模式	84
10.3	寄存器定义	84
	10.3.1 ACMP 状态和控制寄存器 (ACMPSC)	84
10.4	功能描述	85

第 11 章 模定时器 (RS08MTIMV1)

11.1	介绍	87
11.1.1	特点	88
11.1.2	工作模式	88
11.1.2.1	在等待模式下工作	88
11.1.2.2	在停止模式下工作	88
11.1.2.3	在主动背景模式下工作	88
11.1.3	方框图	89
11.2	外部信号描述	89
11.3	寄存器定义	89
11.3.1	MTIM 状态和控制寄存器 (MTIMSC)	90
11.3.2	MTIM 时钟配置寄存器 (MTIMCLK)	91
11.3.3	MTIM 计数寄存器 (MTIMCNT)	91
11.3.4	MTIM 模寄存器 (MTIMMOD)	92
11.4	功能描述	93
11.4.1	MTIM 工作例子	94

第 12 章 开发支持

12.1	介绍	95
12.2	特点	95
12.3	RS08 背景调试控制器 (BDC)	96
12.3.1	BKGD 管脚描述	97
12.3.2	通讯详述	98
12.3.3	异步 和串行通讯定时溢出	100
12.4	BDC 寄存器和控制位	101
12.4.1	BDC 状态和控制寄存器 (BDCSCR)	101
12.4.2	BDC 断点匹配寄存器	102
12.5	RS08 BDC 命令	103

附录 A 电气描述

A.1	介绍	107
A.2	绝对最大参数	107
A.3	热特性	108
A.4	静电 (ESD) 保护特性	109
A.5	DC 特性	109
A.6	电源电流特性	113
A.7	模拟比较器 (ACMP) 电气特性	115
A.8	内部时钟源特性	115
A.9	AC 特性	116
	A.9.1 控制时间	116
A.10	FLASH 规范	117
A.11	EMC 性能	120
	A.11.1 放射性	120
	A.11.2 导线瞬态磁化性	120

附录 B 订单信息和机械图

B.1	订单信息	123
B.2	机械图	123

Chapter 1 MC9RS08KA2 产品概述

1.1 概述

MC9RS08KA2 系列微控制器 (MCU) 是一款极低成本、低管脚数的芯片，可用在家电、玩具和小设备等。这款芯片是由标准的片内模块构成，包括一个非常小的、高效的 RS08 CPU 内核，63 字节大小的 RAM，2K 字节大小的 FLASH，一个 8 位模定时器，键盘中断和模拟比较器。这款芯片有 6 管脚和 8 管脚两种封装。

1.2 MCU 方框图

Figure 1-1 列出了 MC9RS08KA2 系列 MCU 结构。

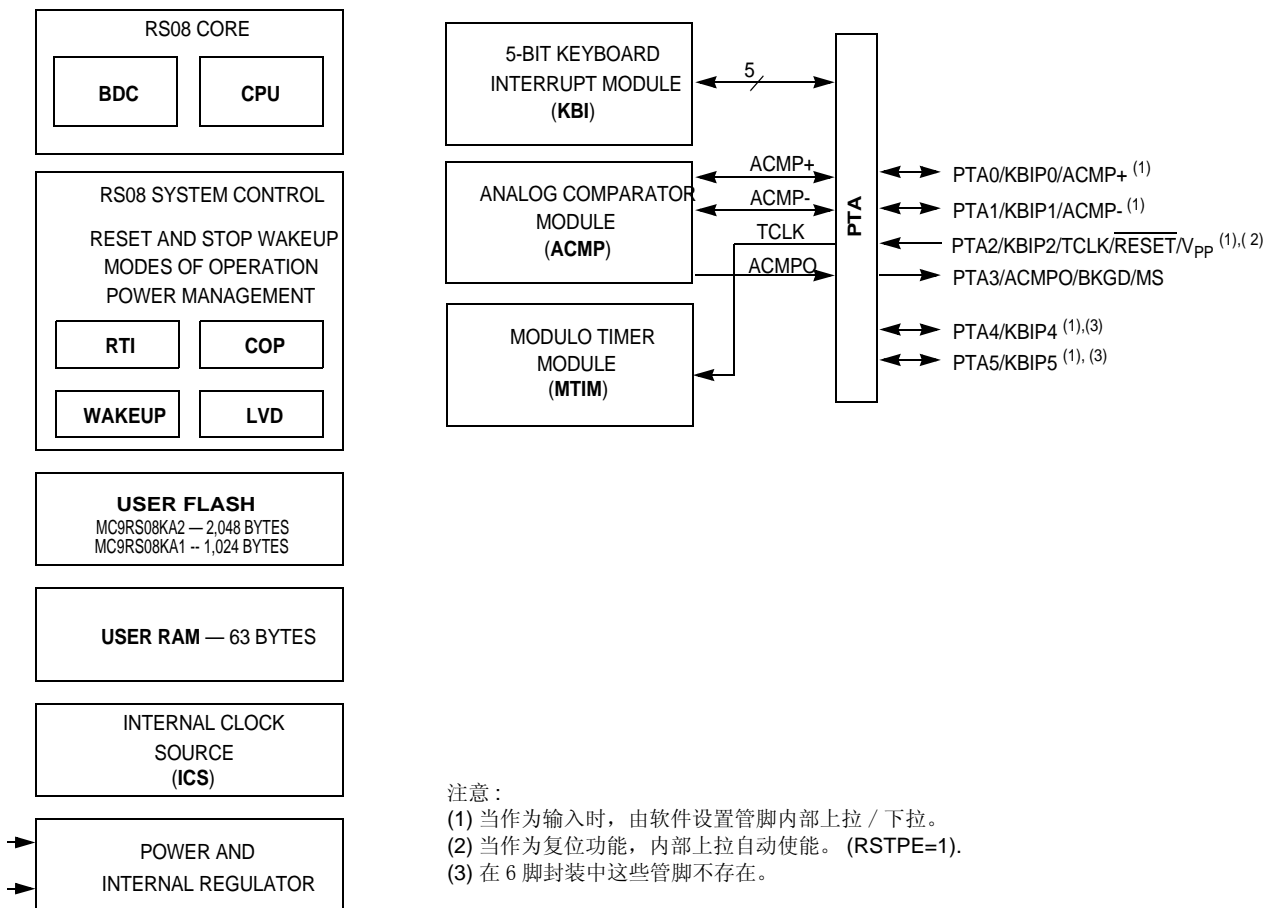


Figure 1-1. MC9RS08KA2 系列方框图

Table 1-1 列出了片内模块的功能版本

Table 1-1. 框图版本

模块	版本
模拟比较器 (ACMP)	1
键盘中断 (KBI)	1
模定时器 (MTIM)	1
内部时钟源 (ICS)	1

1.3 系统时钟描述

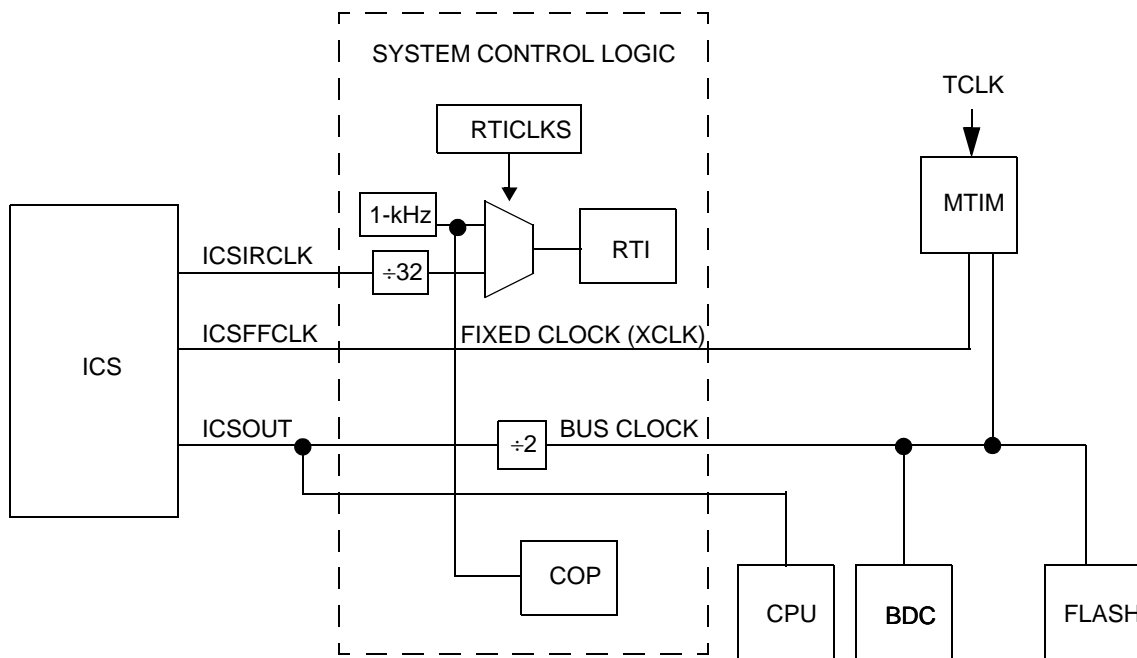


Figure 1-2. 系统时钟分配图

Figure 1-2 列出了此 MCU 简单的时钟连接方框图。总线时钟频率是 ICS 输出频率的一半，用于所有的模块。

Chapter 2 管脚和连接

2.1 介绍

本章描述各封装的管脚信号，包括管脚接线图、信号特性表和信号介绍。

2.2 管脚分配

Figure 2-1 和 Figure 2-3 列出了所有的 MC9RS08KA2 封装的管脚分配情况。

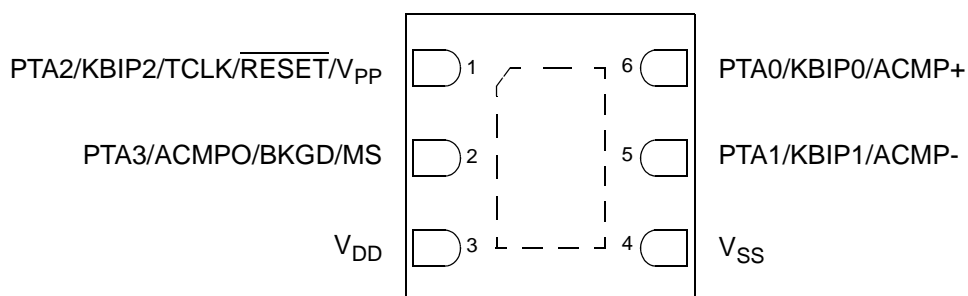


Figure 2-1. MC9RS08KA2 in 6-Pin DFN

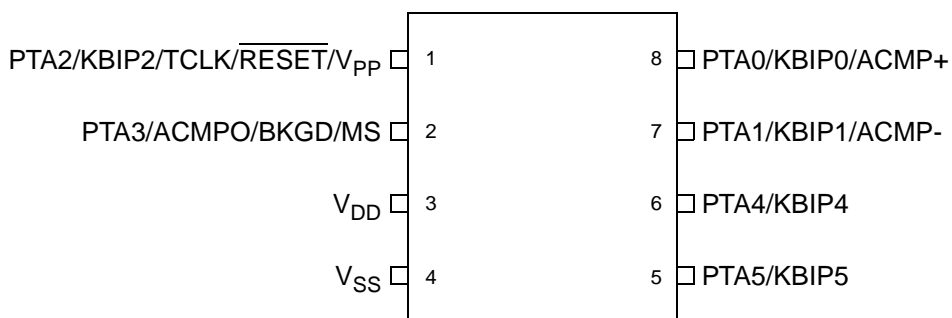


Figure 2-2. MC9RS08KA2 in 8-Pin PDIP

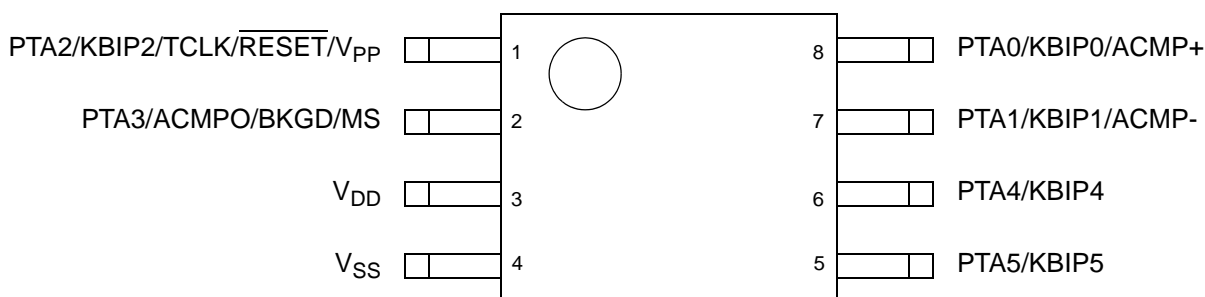


Figure 2-3. MC9RS08KA2 in 8-Pin Narrow Body SOIC

2.3 推荐的系统连接

Figure 2-4 列出了背景调试和 FLASH 编程的参考连接。

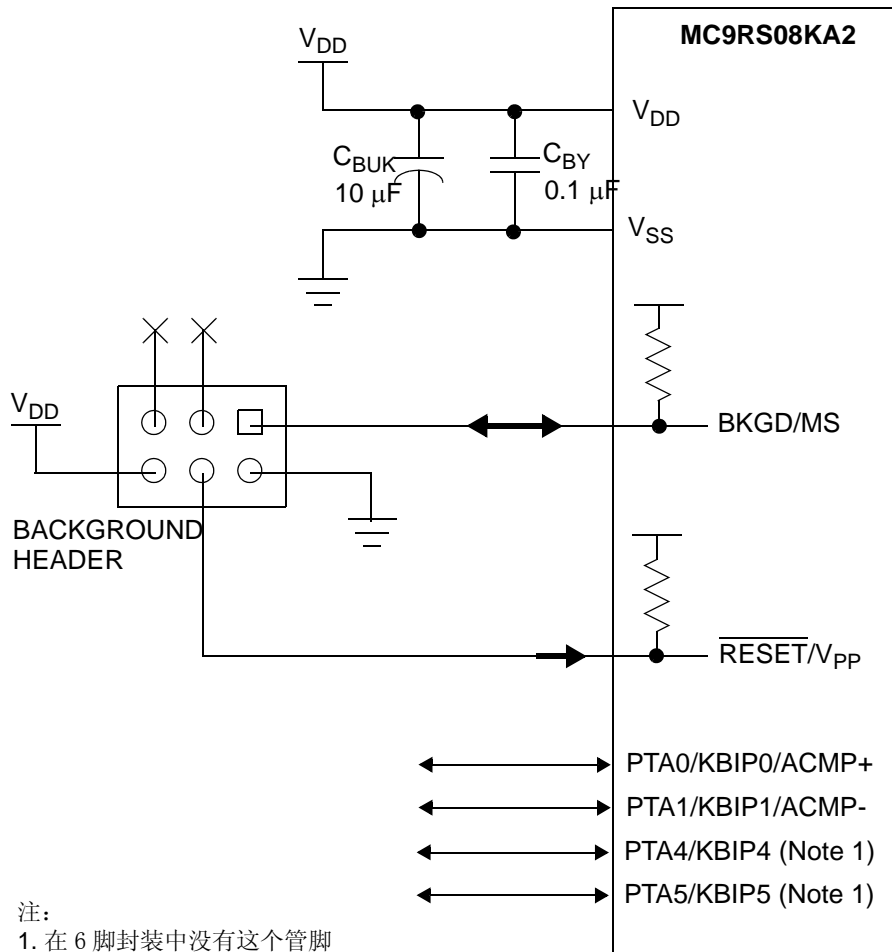


Figure 2-4. 系统连接方框图参考

2.4 管脚详述

本节提供了系统连接的详细描述。

2.4.1 电源

V_{DD} 和 V_{SS} 是 MCU 基本的电源管脚。这个电压源给所有的 I/O 电路和内部参考电压调节器提供电源。内部电压调节器提供一个可调的更低电压源给 CPU 和 MCU 的其它内部电路。

典型地，应用系统有两个独立的电容连接在电源管脚：一个大的电解电容，如一个 10uF 胆电容，给整个系统提供大充电电荷；以及一个旁路电容，如一个 0.1µF 陶瓷电容，放在离 MCU 电源管脚最近的地方，用于抑止高频噪音。

2.4.2 PTA2/KBIP2/TCLK/ $\overline{\text{RESET}}/V_{PP}$

上电复位 (POR) 进入用户模式后，PTA2/KBIP2/TCLK/ $\overline{\text{RESET}}/V_{PP}$ 管脚默认为输入管脚，PTA2。将 SOPT 寄存器的 RSTPE 置位，此管脚作为复位输入脚。配置为复位脚后，此管脚一直为复位脚，

直到下次上电复位。当此管脚被拉低，复位脚以一个外部复位源复位 MCU。当被设置为复位脚 ($RSTPE = 1$)，内部上拉功能自动使能。

当执行 FLASH 编程或擦除时，此管脚需要一个外部 V_{PP} 电压 (典型值 12 V, 参见 Section A.10, “FLASH 特性”)。不管此管脚为何功能， V_{PP} 接线总是连接到内部 FLASH 模块。当没有作为 FLASH 编程或擦除时，为了避免 FLASH 过重负荷，外部 V_{PP} 电压必须被切断，并且高于 V_{DD} 的电压必须避免。

注意

此管脚没有包含一个钳压二极管连接到 V_{DD} ，当没有执行 FLASH 编程或擦除时这个管脚的电压不能高于 V_{DD} 。

2.4.3 PTA3/ACMPO/BKGD/MS

作为只输出的 PTA3 管脚的功能可以设置为背景调试或模拟比较输出。在复位期间，这个管脚功能模式可选择。复位后，这个管脚功能立即作为背景调试模式并且被用于背景调试通讯。在背景调试功能期间，这管脚有一个内部上拉使能。作为一个只输出端口，SOPT 寄存器中的 BKGDPE 位必须被清零。

假如没有任何东西连接到这个管脚，MCU 在复位的上升沿期间进入到正常工作模式。假如有一个调试系统连接到 6 管脚的标准背景调试头，在上电复位期间将 BKGD/MS 一直拉低，强迫 MCU 进入背景调试模式。

BKGD 管脚主要用于背景调试控制器 (BDC) 通讯，此通讯使用目标 MCU 的 BDC 时钟的 16 个时钟周期的通讯协议。目标 MCU 的 BDC 时钟等于总线时钟。因此，大电容连接到 BKGD/MS 管脚将会干扰背景串行通讯。

尽管 BKGD 管脚是个假的开漏极管脚，背景调试通讯协议提供了暂时的、积极驱动的高速脉冲确保快速上升时间。来自电线的小电容和内部上拉设备在 BKGD 管脚的上升和下降沿几乎不起作用。

2.4.4 通用 I/O 和外围端口

剩下的管脚都具有通用 I/O 和芯片外设功能，如定时器和模拟比较器。上电复位后，所有这些管脚立即被设置为高阻态通用输入且内部上拉 / 下拉禁止。

管脚和连接

注意： 为了避免外部漏电流进入悬浮的输入管脚，应用例程中的复位初始化程序应该使能芯片的上拉 / 下拉设备或将管脚设置为输出。

Table 2-1. 管脚共享参数

管脚名称	方向	上拉 / 下拉 ⁽¹⁾	可选的功能 ⁽²⁾	
V _{DD}	—	—	电源	
V _{SS}	—	—	地	
PTA0	I/O	SWC	PTA0 KBIP0 ACMP+	通用输入 / 输出 (GPIO) 键盘中断 (只停止 / 等待唤醒) 模拟比较输入
PTA1	I/O	SWC	PTA1 KBIP1 ACMP-	通用输入 / 输出 (GPIO) 键盘中断 (只停止 / 等待唤醒) 模拟比较输入
PTA2	I	SWC ⁴	PTA2 KBIP2 TCLK RESET V _{PP}	通用输入 键盘中断 (只停止 / 等待唤醒) 模定时器时钟源 复位 V _{PP}
PTA3	I/O ⁽³⁾	— ⁽⁴⁾	PTA3 ACMPO BKGD MS	通用输出 模拟比较器输出 背景调试数据 模式选择
PTA4 ⁽⁵⁾	I/O	SWC	PTA4 KBIP4	通用输入 / 输出 (GPIO) 键盘中断 (只停止 / 等待唤醒)
PTA5 ⁵	I/O	SWC	PTA5 KBIP5	通用输入 / 输出 (GPIO) 键盘中断 (只停止 / 等待唤醒)

1. SWC 是软件控制上拉 / 下拉电阻器，这个寄存器与各自端口有关。
2. 任选功能首先从最低优先级被列出。例如，在 PTA0 管脚 GPIO 是最低优先级功能。ACMP+ 是最高优先级功能。
3. 当配置为 PTA3 功能是只输出。
4. 当 PTA2 或 PTA3 配置为 RESET 或 BKGD/MS，各自的，上拉使能。当接上 V_{PP}，上拉 / 下拉自动禁止。
5. 在 6 管脚封装中这个管脚没有。使能上拉或下拉设备可以确保防止外部漏电流进入输入管脚。

Chapter 3 工作模式

3.1 介绍

本章介绍了 MC9RS08KA2 的工作模式及其功能，也介绍如何进出这些工作模式。

3.2 特点

- 主动背景模式
- 等待模式：
 - CPU 关闭降低功耗
 - 系统时钟继续运行
 - 全电压调节被保持
- 停止模式：
 - 系统时钟被停止；电压调节器备用
 - 所有内部电路保持供电用于快速恢复

3.3 运行模式

这是 MC9RS08KA2 正常工作模式。在复位的上升沿，当 BKGD/MS 管脚是高电平时选择这种模式。这种模式下，CPU 从起始地址为 \$3FFD 的内部存储器开始执行代码。为让 MCU 复位操作后进入正确的用户程序，必须填写一条带操作数的 JMP 指令（操作码 \$BC），该操作数位于 \$3FFE-\$3FFF。这个操作数为用户程序的起始地址，相当于 HC08/S08 系列中复位中断向量。

3.4 主动背景模式

主动背景模式功能通过在 RS08 核中的背景调试控制器 (BDC) 管理。在软件调试期间 BDC 提供分析 MCU 工作的方法。

主动背景模式通过四种方法中的任何一种可进入：

- 在上电复位（POR）期间 BKGD/MS 管脚被拉低或发出了一条背景调试强迫复位 (BDC_RESET) 命令
- 当 BKGD 管脚收到背景调试命令
- 当 BGND 指令被执行
- 当遇到一个 BDC 断点

进入主动背景模式后，CPU 保持挂起状态等待串行背景命令，而不会从用户应用程序执行指令。

背景命令有 2 种类型：

- 无扰命令，被定义为在用户程序运行时能被发出的命令，MCU 在运行状态时能够通过 BKGD 管脚发出。当 MCU 在主动背景模式时无扰命令也可以执行。无扰命令包括：
 - 存储器存取命令
 - 存储器存取与状态命令
 - BACKGROUND 命令

工作模式

- 主动背景命令，仅当 MCU 在主动背景模式时能被执行，包括：
 - 读或写 CPU 寄存器
 - 在某个时间跟踪一条用户程序指令
 - 离开主动背景模式返回到用户应用程序 (GO)

主动背景模式用于 MCU 在进入运行模式之前第一时间将用户应用代码编程到 Flash 程序存储器中。当 MC9RS08KA2 从飞思卡尔半导体工厂运出，Flash 程序存储器通常被擦除，因此在运行模式没有程序可以运行，除非 Flash 程序存储器被编程。在预先编程后，主动背景模式也可以擦除和重新编程 Flash 存储器。

关于主动背景模式附加的信息，请参考此数据手册“开发支持”章节。

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，没有时钟提供给 CPU，MCU 进入低功耗状态。程序计数器 (PC) 停止在 WAIT 被执行的位置。当一个中断需求产生：

1. MCU 退出等待模式恢复工作。
2. PC 被加 1 并且取出下条指令执行。

由于没有向量被取出，用户程序需要查明唤醒 MCU 的相关中断源。

当 MCU 在等待模式，不是所有的背景调试命令都能够被使用，仅 BACKGROUND 命令和存储器存取和状态命令可用。存储器状态存取命令不允许存储器存取，但当 MCU 在停止或等待模式时可以报告一个错误的标志。BACKGROUND 命令可以从等待模式下唤醒 MCU 进入主动背景模式。

Table 3-1 总结了 MCU 在等待模式下的状态。

Table 3-1. 等待模式状态

Mode	CPU	Digital Peripherals	ICS	ACMP	Regulator	I/O Pins	RTI
Wait	Standby	Optionally on	On	Optionally on	On	States held	Optionally on

3.6 停止模式

当系统选型寄存器的 STOPE 位被置位，执行 STOP 指令进入停止模式。在停止模式，所有 CPU 和模块的内部时钟被停止。假如 STOPE 位没有被置位，当 CPU 执行 STOP 指令时，MCU 不会进入停止模式，并且产生一个非法指令强迫复位。

Table 3-2 总结了 MCU 在停止模式下的状态。

Table 3-2. 停止模式状态

Mode	CPU	Digital Peripherals	ICS ⁽¹⁾	ACMP ⁽²⁾	Regulator	I/O Pins	RTI ⁽³⁾
Stop	Standby	Standby	Optionally on	Optionally on	Standby	States held	Optionally on

1. 在停止模式，内部时钟源需要 IREFSTEN = 1 并且 LVDE 和 LVDSE 必须被置位才能允许工作。

2. 假如需要待隙参考，进入停止模式前，SPMSC1 中的 LVDE 和 LVDSE 位必须被置位。

3. 假如在 ICS 模块中的 32-kHz 可调时钟被选择用于作为 RTI 的时钟源，进入停止模式前，在 SPMSC1 中的 LVDE 和 LVDSE 位必须被置位。

进入停止模式后 MCU 内的所有时钟停止。当 IREFSTEN 位被清除内部时钟源默认关闭，电压调整器进入旁路状态。所有内部寄存器和逻辑单元的状态保持不变，如 RAM 内容。I/O 管脚状态被钳住。

任何异步中断或实时中断，都可以退出 STOP 模式。异步中断包括键盘中断、低电压检测中断或模拟比较中断。

假如是通过复位管脚退出停止模式，MCU 将复位且程序从 \$3FFD 地址开始运行。假如是通过异步中断或实时中断退出停止模式，停止指令地址的下条指令被执行。用户程序需要查明唤醒 CPU 的相关中断源。

一个用于实时中断 (RTI) 的独立的时钟源 (≈ 1 kHz) 允许从停止模式唤醒 CPU 而不需要外部器件。当 RTIS = 000, 实时中断功能和 1-kHz 时钟源被禁止。当 1-kHz 时钟源被禁止时功耗更低，但在这种情况下，实时中断不能将 MCU 从停止模式下唤醒。

在内部时钟源模块内可调的 32-kHz 时钟源可以用于实时中断，允许不需要外部器件就可以从停止模式唤醒 MCU。IREFSTEN 位置位，32-kHz 参考时钟使能。为了在停止模式下内部时钟源能运行，进入停止模式前，SPMSC1 寄存器中的 LVDE 和 LVDSE 位必须都置位。

3.6.1 停止模式下主动 BDM 使能

假如 BDCSCR 寄存器中的 ENBDM 位被置位，从运行模式进入背景调试模式是允许的。本数据手册的开发支持章有这个寄存器的描述。当 MCU 进入停止模式，背景调试逻辑的系统时钟保持活动的，因此背景调试通讯依然可能。另外，电压调整器不会进入低功耗旁路状态；它维持全部的内部调节。

工作模式

在停止模式下大部分背景调试命令不能用。存储器状态存取命令不允许存储器存取。但当 MCU 在停止或等待模式时可以报告一个错误的标志。假如 ENBDM 位被置位，BACKGROUND 命令可以从停止模式下唤醒 MCU 进入主动背景模式。进入背景调试模式后，所有的背景命令可用。

Table 3-3 总结了当进入背景模式使能时在停止模式下 MCU 的状态

Table 3-3. BDM 使能停止模式状态

Mode	CPU	Digital Peripherals	ICS	ACMP	Regulator	I/O Pins	RTI
Stop	Standby	Standby	On	Optionally on	On	States held	Optionally on

3.6.2 停止模式下低电压检测使能

当电源下降低于 LVD 电压时，LVD 系统能够产生中断或复位。假如在停止模式下 LVD 使能（在 SPMSC1 寄存器中的 LVDE 和 LVDSE 位被置位），CPU 执行 STOP 指令时，电压调整器保持活动状态。

Table 3-4 总结了当允许 LVD 产生复位时停止模式下 MCU 的状态。

Table 3-4. LVD 使能停止模式状态

Mode	CPU	Digital Peripherals	ICS	ACMP	Regulator	I/O Pins	RTI
Stop	Standby	Standby	Optionally on	Optionally on	On	States held	Optionally on

Chapter 4 存储器

4.1 存储器映象

MCU 的存储器图被划分成以下块：

- 快速存取 RAM 用于小型指令和短型指令 (\$0000-\$000E¹)
- 间接数据存储 D[X] (\$000E)
- 用于 D[X] 的变址寄存器 X (\$000F)
- 经常使用的外设寄存器 (\$0010-\$001E)
- 页寄存器 (\$001F)
- RAM (\$0020-\$004F)
- 分页窗口 (\$00C0-\$00FF)
- 其它外设寄存器 (\$0200-\$023F)
- 非易失性存储器
 - MC9RS08KA2: \$3800-\$3FFF
 - MC9RS08KA1: \$3C00-\$3FFF

1. 当变址寄存器 X 内容是 \$0E 时，\$000E 的物理 RAM 能通过 D[X] 寄存器访问。

存储器

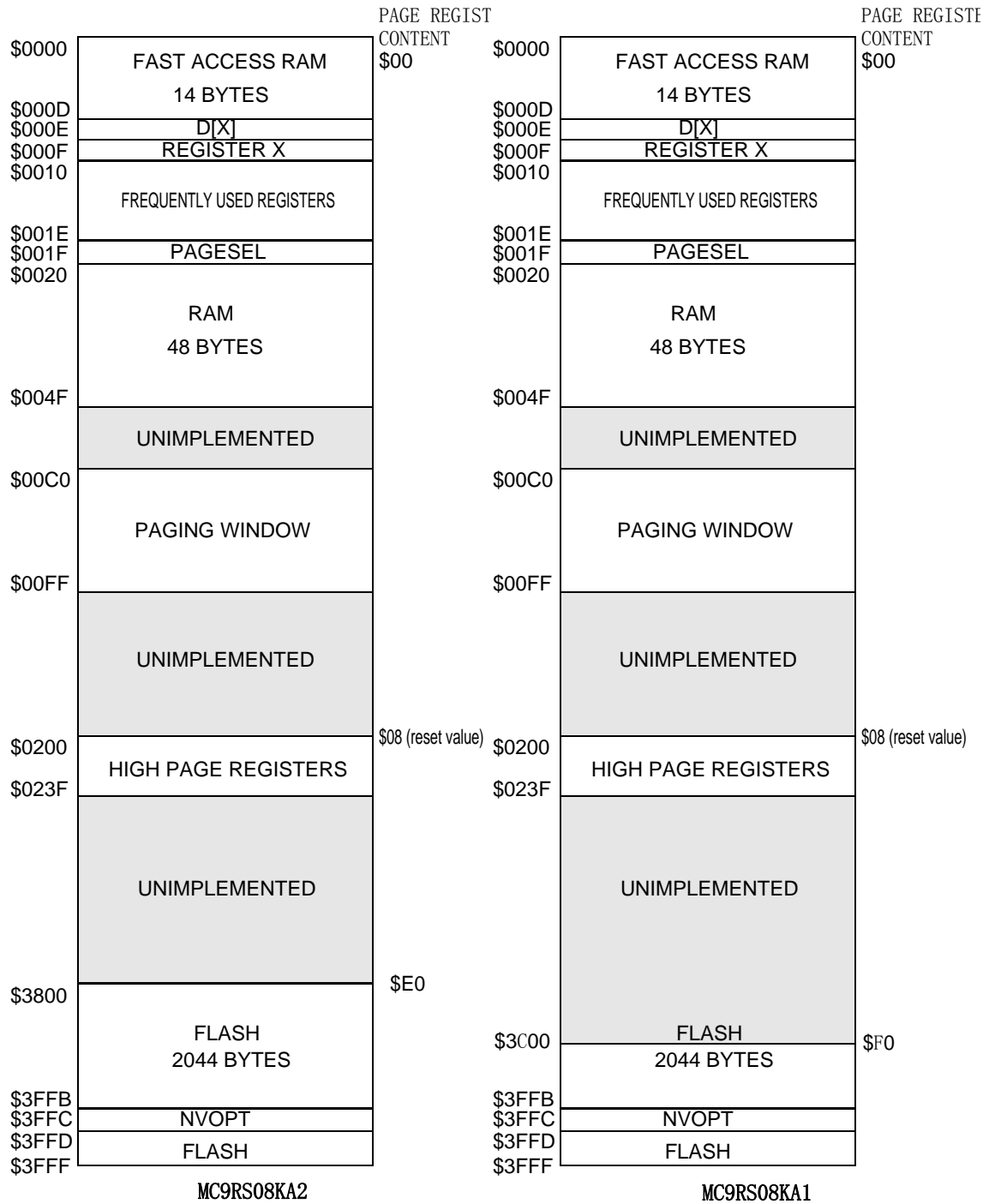


Figure 4-1. MC9RS08K2 系列存储器映象

4.2 无效存储器

在一个无效存储器地址试图存取数据或指令将会导致复位。

4.3 变址 / 间接寻址

寄存器 D[X] 和寄存器 X 一起实现间接数据存取。寄存器 D[X] 被映射地址 \$000E。寄存器 X 位于地址 \$000F。当寄存器 D[X] 被存取时这个 8 位寄存器 X 包含了被使用的地址。复位后寄存器 X 被清零。通过编程寄存器 X，位于第一页 (\$0000-\$00FF) 内的任何位置都能够通过 D[X] 被读 / 写。Figure 4-2 显示了 D[X] 和寄存器 X 的相对关系。例如，当寄存器 X 被写给定值，HC08/S08 语句 *lda ,x* 相当于 RS08 的语句 *lda D[X]*。

\$000E 物理位置在 RAM 区，当寄存器 X 的值是 \$0E 时，通过 D[X] 访问这个地址获得 \$000E RAM 的内容。\$000F 物理位置是寄存器 X 自身。通过 D[X] 读取这个地址获得寄存器 X 的内容；写这个地址修改寄存器 X。

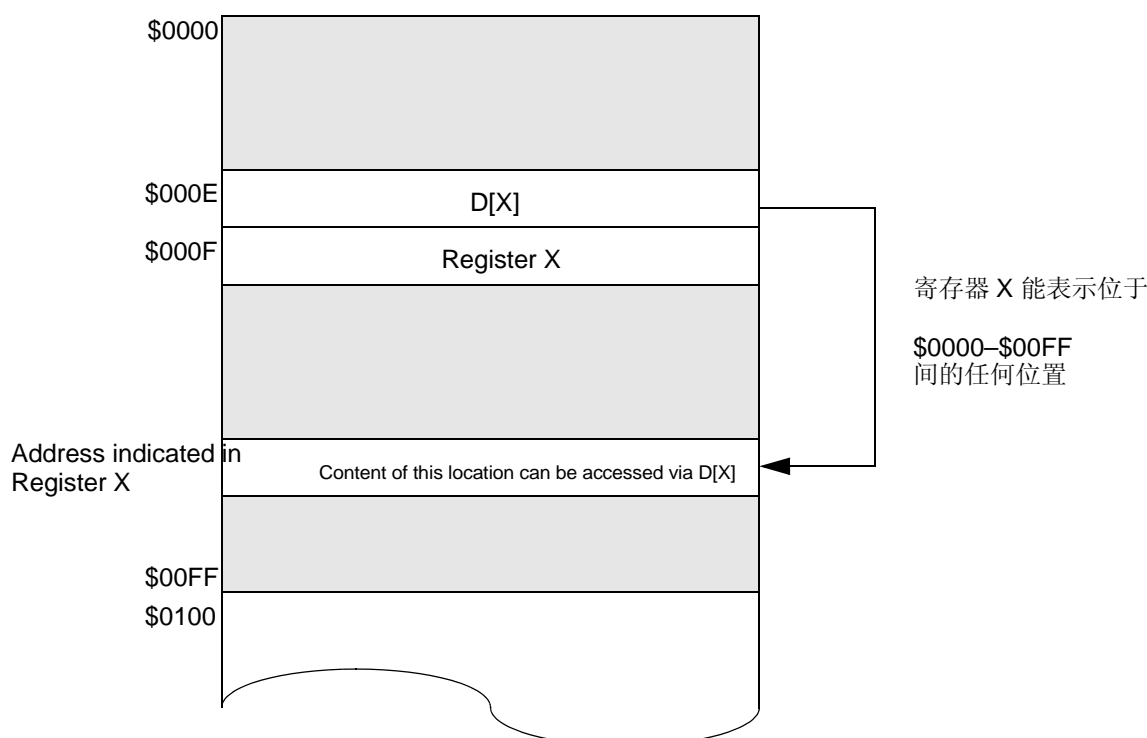


Figure 4-2. 间接寻址寄存器

4.4 RAM 和寄存器地址和位分配

使用小型、短型、直接寻址方式指令能访问直接存取 RAM 空间。对于小型寻址方式指令，操作数与操作码被编码为一个单字节。

存储器

为了完成更快载入、存储和清除工作，频繁使用的寄存器可以使用短型寻址方式指令。对于短型寻址方式指令，操作数与操作码被编码为一个单字节。

Table 4-1. 寄存器总结

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$0000– \$000D		Fast Access RAM							
\$000E	D[X] ⁽¹⁾	Bit 7	6	5	4	3	2	1	Bit 0
\$000F	X	Bit 7	6	5	4	3	2	1	Bit 0
\$0010	PTAD	0	0	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
\$0011	PTADD	0	0	PTADD5	PTADD4	0	0	PTADD1	PTADD0
\$0012	Unimplemented	—	—	—	—	—	—	—	—
\$0013	ACMPSC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD	
\$0014	ICSC1	0	CLKS	0	0	0	0	0	IREFSTEN
\$0015	ICSC2	BDIV		0	0	LP	0	0	0
\$0016	ICSTRM	TRIM							
\$0017	ICSSC	0	0	0	0	0	CLKST	0	FTRIM
\$0018	MTIMSC	TOF	TOIE	TRST	TSTP	0	0	0	0
\$0019	MTIMCLK	0	0	CLKS			PS		
\$001A	MTIMCNT	COUNT							
\$001B	MTIMMOD	MOD							
\$001C	KBISC	0	0	0	0	KBF	KBACK	KBIE	KBIMOD
\$001D	KBIPE	—	—	KBIPE5	KBIPE4	—	KBIPE2	KBIPE1	KBIPE0
\$001E	KBIES	—	—	KBEDG5	KBEDG4	—	KBEDG2	KBEDG1	KBEDG0
\$001F	PAGESEL	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6
\$0020– \$004F		RAM							
\$0050– \$00BF	Unimplemented	—	—	—	—	—	—	—	—
\$00C0– \$00FF		Paging Window							
\$0100– \$01FF	Unimplemented	—	—	—	—	—	—	—	—
\$0200	SRS	POR	PIN	COP	ILOP	ILAD	0	LVD	0
\$0201	SOPT	COPE	COPT	STOPE	0	0	0	BKGDPE	RSTPE
\$0202	SIP1	—	—	—	KBI	ACMP	MTIM	RTI	LVD
\$0203	Unimplemented	—	—	—	—	—	—	—	—
\$0204	Reserved	—	—	—	—	—	—	—	—
\$0205	Unimplemented	—	—	—	—	—	—	—	—
\$0206	SDIDH	REV3	REV2	REV1	REV0	ID			
\$0207	SDIDL	ID							
\$0208	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS		
\$0209	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	BGBE
\$020A	Reserved	—	—	—	—	—	—	—	—
\$020B	Reserved	—	—	—	—	—	—	—	—

□ = Unimplemented or Reserved

Table 4-1. 寄存器总结 (continued)

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$020C– \$020F	Unimplemented	—	—	—	—	—	—	—	—
\$0210	FOPT	0	0	0	0	0	0	0	SECD
\$0211	FLCR	0	0	0	0	HVEN	MASS	0	PGM
\$0212– \$0213	Reserved	—	—	—	—	—	—	—	—
\$0214– \$021F	Unimplemented	—	—	—	—	—	—	—	—
\$0220	PTAPE	0	0	PTAPE5	PTAPE4	0	PTAPE2	PTAPE1	PTAPE0
\$0221	PTAPUD	0	0	PTAPUD5	PTAPUD4	0	PTAPUD2	PTAPUD1	PTAPUD0
\$0222	PTASE	0	0	PTASE5	PTASE4	PTASE3	0	PTASE1	PTASE0
\$0223– \$023F	Unimplemented	—	—	—	—	—	—	—	—
\$3FF8	Reserved	—	—	—	—	—	—	—	—
\$3FF9	Reserved	—	—	—	—	—	—	—	—
\$3FFA ⁽²⁾	Reserved	Reserved for Room Temperature ICS Trim							
\$3FFB ²	Reserved	Reserved							
\$3FFC	NVOPT	0	0	0	0	0	0	0	SECD

— = Unimplemented or Reserved

1. 当变址寄存器 X 内容是 \$0E, 在 \$000E 的物理 RAM 能通过 D[X] 寄存器访问。
2. 假如使用的 MCU 未调整, \$3FFA 和 \$3FFB 可以被应用程序载入。

4.5 RAM

此芯片包含两个区域的静态 RAM。从 \$0000 到 \$000D 的区域可以通过更高效的小型寻址方式指令和短型寻址方式指令直接访问。当寄存器 X 值是 \$0E 时, 可以通过 D[X] 寄存器访问位于 \$000E 的 RAM 值。或当 PAGESEL 寄存器值是 \$00 时通过位于 \$00CE 的分页窗口访问。第二个 RAM 区域从 \$0020 到 \$0040, 它能通过直接寻址方式指令访问。

当 MCU 在低功耗等待和停止模式时 RAM 保持数据。当给 RAM 供电的电压没有下降到低于维持 RAM 所需的最小值时, 任何复位都不会影响 RAM 数据。

4.6 Flash

Flash 存储器主要用于存储程序。在编译成功后, 在线编程允许应用程序下载到 Flash 存储器中。通过单线背景调试接口编程到整个空间是可以的。由于此芯片没有包括在线电压泵, 在编程和擦除工作时需要外部 V_{PP}。

特点

Flash 存储器特点包括:

- 在典型电压和温度下超过 1000 次编程 / 擦除。
- Flash 安全特性

4.6.1 Flash 编程步骤

基于行编程 Flash 存储器。一行由 64 个连续字节组成，起始地址为 \$3X00, \$3X40, \$3X80, 或 \$3XC0。使用下列步骤编程一行 Flash 存储器：

1. 加外部电压 V_{PP} 。
2. PGM 位置位。这用于配置编程的存储器和允许锁住编程的地址和数据。
3. 写任何数据到任何 Flash 位置，经过高页存取窗口，内部的行地址范围被编程（数据写操作之前，PAGESEL 寄存器必须配置正确去映射高页存取窗口到相关的 Flash 行）。
4. 延时 t_{nvs} 。
5. HVEN 位置位。
6. 延时 t_{pgs} 。
7. 写数据到被编程的 Flash 空间。
8. 延时 t_{prog} 。
9. 重复步骤 7 和 8 直到行内所有字节编程完毕。
10. PGM 位清零。
11. 延时 t_{nvh} 。
12. HVEN 位清零。
13. 延时 t_{rcv} 后，存储器又能在读方式下访问。
14. 去除外部电压 V_{PP} 。

这个编程顺序重复到整个存储器，直到所有的数据被烧写。

注意

Flash 存储器不能被 Flash 空间内的代码执行编程或擦除。为了编程或擦除 Flash，必须来自 RAM 或 BDC 的命令才能执行。在擦除或编程过程中，用户代码不能进入等待或停止模式。

这些操作必须在以上规定的步骤执行；其它一些不相干的操作可能在这些步骤之间出现。

4.6.2 Flash 块擦除操作

用下面步骤块擦除整个 Flash 存储器：

1. 加外部电压 V_{PP} 。
2. 在 Flash 控制寄存器 MASS 位置位；
3. 写任何数据到任何 Flash 位置，通过高页存取窗口 \$00C0-\$00FF。（数据写操作之前，PAGESEL 寄存器必须配置正确去映射高页存取窗口任何 Flash 位置）。
4. 延时 t_{nvs} 。
5. HVEN 位置位。
6. 延时 t_{me} 。
7. MASS 位清零。
8. 延时 t_{nvhl} 。

9. HVEN 位清零。
10. 延时 t_{rcv} 后，存储器又能在读模式下访问。
11. 去除外部电压 V_{pp} 。

注意

Flash 存储器不能被 Flash 空间内的代码执行编程或擦除。为了编程或擦除 Flash，必须来自 RAM 或 BDC 的命令才能执行。在擦除或编程过程中，用户代码不能进入等待或停止模式。

这些操作必须在以上规定的步骤执行；其它一些不相干的操作可能在这些步骤之间出现。

4.6.3 安全性

MC9RS08KA2 拥有的电路帮助防止未经授权访问 Flash 存储器的内容。当有安全机制保证时，Flash 可以认为是一个安全的资源。RAM、直接页寄存器和背景调试控制器被认为是不安全的资源。试图通过背景调试接口，或无论 BKGDPE 是否被置位，访问一个安全的存储区是行不通的（读出全是 0）。

基于 FOPT 寄存器的非易失性寄存器位 (SECD) 的状态决定是否安全。在复位期间，非易失性寄存器 NVOPT 被拷贝到高页寄存器 FOPT 寄存器。通过对 NVOPT 编程用户获得安全机制保护，与 Flash 存储器被编程的相同的时间完成。注意被擦除状态 (SECD = 1) 使 MCU 不安全。当 NVOPT 的 SECD 位被清零 (SECD = 0)，单片机下次通过上电复位，内部复位，或外部复位方式被复位，可以确保安全。为了使安全机制失效，块擦除必须通过 BDM 命令执行并且跟随在任何复位后。

独立的背景调试控制器依然可以访问寄存器和 RAM。通过写 Flash 控制寄存器实现 Flash 块擦除是可以的，通过 BDM 命令参考 [Section 4.6.2, “Flash 块擦除操作”](#)。

由下面步骤通过背景调试接口总是能使安全机制失效：

1. 通过背景 BDM 命令或 RAM 装载程序块擦除 Flash。
2. 执行复位单片机将用失效的安全保护引导。

注意

当单片机引导进入正常工作模式，MS 管脚在复位期间为高，SECD 位清零 (SECD = 0)，Flash 安全机制起用。BKGDPE 位复位后为 0，并且所有的 BDM 通讯关闭，背景调试禁止。

存储器

4.7 Flash 寄存器和控制位

Flash 模块有一个非易失性寄存器，NVOPT (\$3FFC), 复位时在 Flash 存储器被拷贝到相关的控制寄存器 FOPT (\$0210)。

4.7.1 Flash 选项寄存器 (FOPT 和 NVOPT)

复位时非易失性单元 NVOPT 的内容在 Flash 内被拷贝到 FOPT。位 7 至 1 不能使用并且读出总是为 0。这个寄存器在任何时间都能被读出，但是写无意义或无效。为了改变这个寄存器的值，通常擦除和再编程 Flash 中的 NVOPT 寄存器后导致一次新的 MCU 复位。

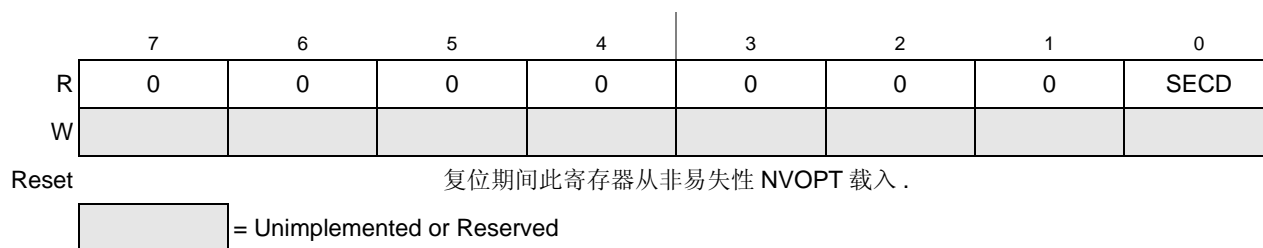


Figure 4-3. Flash 选项寄存器 (FOPT)

Table 4-2. FOPT 域描述

域	描述
0 SECD	安全状态代码 — 此位决定 MCU 的安全状态。MCU 被保护，Flash 存储器内容不能被来自任何不安全源包括背景调试接口访问。参考 Section 4.6.3, “安全性”。 0 安全使能 1 安全失效

4.7.2 Flash 控制寄存器 (FLCR)

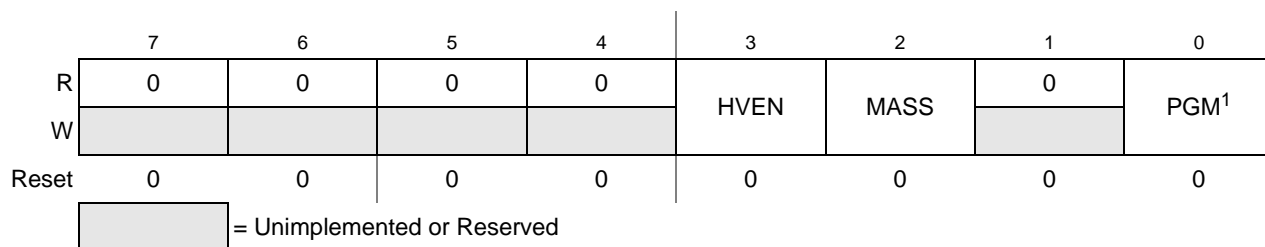


Figure 4-4. Flash 控制寄存器 (FLCR)

Table 4-3. FLCR 域描述

域	描述
3 HVEN	高电压使能 — 此读 / 写位允许高电压加载到 Flash 阵列用于编程和擦除操作。只有当 PGM = 1 或 MASS = 1 和正常编程或擦除顺序被跟随，HVEN 可以被置位。 0 高电压禁止加载到阵列 1 高电压允许加载到阵列
2 MASS	块擦除位 — 这个读 / 写位配置存储器块擦除工作。 0 块擦除没有被选择。 1 选择块擦除。
0 PGM ⁽¹⁾	编程控制位 — 这个读 / 写位配置存储器编程工作。PGM 与 MASS 位互锁因此这两个位不能同时等于 1 或置 1。 0 不选择编程。 1 选择编程。

1. 当 Flash 安全机制有效时，写 PGM 位是无效的。结果，Flash 编程不允许。

4.8 页选择寄存器 (PAGESEL)

这是一个在直接页地址可以分页访问 64 个字节的窗口 (\$00C0-\$00FF)。编程页选择寄存器可以通过直接页访问确定存储器映射中的 64 个字节块。例如，当 PAGESEL 寄存器值位 \$08 时，高页寄存器 (\$0200-\$023F) 能够由直接寻址方式指令通过页窗口 (\$00C0-\$00FF) 访问。

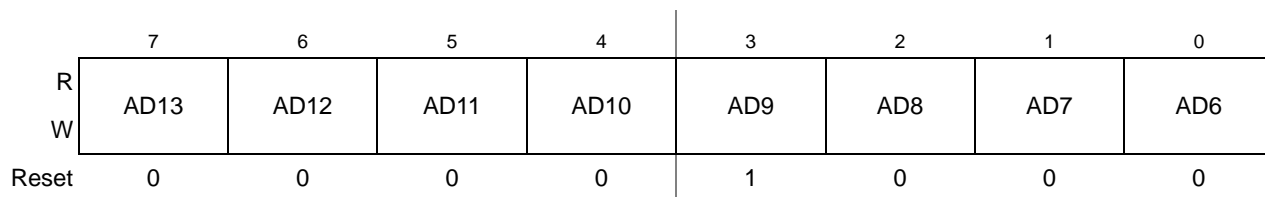


Figure 4-5. 页选择寄存器 (PAGESEL)

Table 4-4. PAGESEL 区描述

Field	Description
7:0 AD[13:6]	页选择器 — 这些位定义地址行位 6 到 13，它们决定通过直接页窗口被访问的存储器块的 64 字节的块界线，见 Figure 4-6 和 Table 4-5。

14- 位存储器地址

Figure 4-6. 存储器块界线选择器

存储器

选择的存储器块地址

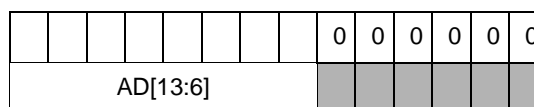


Figure 4-6. 存储器块界线选择器

Table 4-5 列出了通过页窗口 (\$00C0-\$00FF) 访问的存储器块。

Table 4-5. 页窗口对应于 \$00C0-\$00FF

Page	Memory Address
\$00	\$0000-\$003F
\$01	\$0040-\$007F
\$02	\$0080-\$00BF
\$03	\$00C0-\$00FF
\$04	\$0100-\$013F
.	.
.	.
.	.
\$FE	\$3F80-\$3FBF
\$FF	\$3FC0-\$3FFF

注意

物理地址 \$0000-\$000E 是 RAM。物理地址 \$000F 是寄存器 X。D[X] 寄存器仅被映射到地址 \$000E。当 X 寄存器值是 \$0E 或 \$CE，PAGESEL 是 \$00 时，位于 \$000E 的物理 RAM 能够通过 D[X] 寄存器访问。

当 PAGESEL 寄存器是 \$00，页窗口被映射到第一页 (\$00-\$3F)。分页地址 \$00C0-\$00CE 被映射到物理地址 \$0000-\$000E，也就是，RAM。分页地址 \$00CF 被映射到寄存器 X。因此，访问地址 \$CE 的内容返回实际 RAM \$000E 的内容。访问地址 \$000E 返回 D[X] 寄存器内容。

Chapter 5

复位，中断，和通用系统控制

5.1 介绍

本章讨论在 MC9RS08KA2 中的基本复位和中断机制以及不同的复位与中断源。MC9RS08KA2 一些外设模块的中断源在本数据手册的其它章会更详细讨论。本章整合了关于所有中断和复位源的基本信息。一些复位和唤醒源，包括计算机正常工作 (COP) 看门狗和实时中断 (RTI)，在外设系统中没有专门的篇章部分，但也是系统控制逻辑部分。

5.2 特点

复位和中断特点包括：

- 多复位源便于系统配置和可靠工作
- 系统复位状态寄存器 (SRS) 可以显示出最后一次产生复位的复位源
- 系统中断等待寄存器 (SIP1) 可以显示出等待系统中断的状态
 - 模拟比较器中断使能
 - 键盘中断使能
 - 低电压检测中断使能
 - 模定时器中断使能
 - 实时中断中断使能；在停止模式基于独立的 1-KHz 时钟源中断使能；

5.3 MCU 复位

复位的 MCU 通过一个已知初始化条件设置提供一个途径开始执行。复位期间，大多数控制和状态寄存器被强制初始化，并且程序计数器从 \$3FFD 地址开始。为保证正确的复位操作，操作数位于 \$3FFE-\$3FFF 的一条 JMP 指令 (操作码 \$BC) 必须被编程跳转到用户应用软件。此操作数定义的地址为用户程序的开始地址。片上外设模块禁止且 I/O 管脚被初始配置为通用的高阻态输入，上拉 / 下拉功能禁止。

MC9RS08KA2 有七个复位源：

- 外部管脚复位 (PIN) — SOPTC 寄存器的 RSTPE 位被置 1
- 上电复位 (POR)
- 低电压检测 (LVD)
- 计算机正常工作 (COP) 定时器
- 非法操作码检测 (ILOP)
- 非法地址检测 (ILAD)

- 通过 BDC 命令 BDC_RESET 强迫背景调试复位

每一个复位源, 除去背景调试强迫复位外, 在系统复位状态寄存器 (SRS) 中都有一个相关的位。

5.4 计算机正常工作 (COP) 看门狗

假如应用软件未按预期执行, COP 看门狗将会强迫系统复位。为了防止看门狗复位, 应用软件必须周期性喂狗。假如系统程序跑飞, 以致不能在 COP 定时器溢出前将 COP 计数值清零, 系统复位将被产生, 强迫系统回到一个已知的起始点。

任何复位后, SOPT 中的 COPE 位被置位, 允许 COP 看门狗工作 (见 Section 5.8.2, “系统选项寄存器 (SOPT),” 及附加信息)。假如 COP 看门狗在应用软件中不想被使用, 将 COPE 清零即可。SRS 地址单元写入任何值都将 COP 计算器清零。这些写操作不会影响只读 SRS 寄存器的数据。相反地, 写这个地址的行为将被译解为发送一个清零信号到 COP 计数器。

SOPT 寄存器的 COPT 位控制长时间和短时间定时溢出。Table 5-1 概括了 COPT 位的控制功能。COP 看门狗工作时钟源为 1-kHz, 默认为长时间定时溢出 (2^8 cycles)。

Table 5-1. COP 配置选项

COPT	COP 溢出计数 ⁽¹⁾
0	2^5 cycles (32 ms)
1	2^8 cycles (256 ms)

1. 表中列出的值基于 $t_{RTI} \approx 1$ ms。对于这个值的公差, 参见 Section A.9.1, “控制时间,” 中的 t_{RTI} 。

即使应用软件使用 COPE 和 COPT 复位的默认值, 用户在复位初始化期间也应该写这个只能写一次的 SOPT 寄存器用于锁住设置。这样, 假如应用软件跑飞也不会出现意外改变这个寄存器的情况。初次写 SOPT 寄存器将 COP 计数器清零。

在背景调试模式, COP 计数器不计数。

当 MCU 进入停止模式, COP 计数器在进入停止模式前被重新初始化为零。一旦退出停止模式 COP 计数器重新从零开始计数。

5.5 中断

MC9RS08KA2 没有像 HC08 和 HCS08 那样拥有一个具有向量表查询机制的中断控制器。然而, 模块中断源, 如 LVD, KBI, 和 ACMP, 依然可以将 CPU 从等待或停止模式唤醒。用户应用程序的职责是查找相关的模块明确唤醒源。

每个模块唤醒源与相关的中断设置位有关系。假如这个位被禁止, 此中断源被关闭, 这个源不能将 CPU 从等待或停止模式唤醒。然而, 相关的中断标志位依然会被置位, 指出一个外部唤醒事件发生。

系统中断等待寄存器 (SIP1) 显示了系统等待中断的状态。当 SIP1 的只读位被置 1, 表明有一个等待中断需要处理指明的模块。写这个寄存器位是无效的。当来自被指明的模块的所有相关中断标准位被清零, 这个等待中断标志位自动清零。

5.6 低电压检测 (LVD) 系统

为了保护存储器内容和控制 MCU 系统在电源电压变化状态, MC9RS08KA2 拥有一个低电压检测系统。这个系统由一个上电复位和一个具有预先定义下跌电压的 LVD 电路组成。当 SPMSC1 寄存器的 LVDE 置位, LVD 允许工作。进入停止模式后 LVD 禁止工作, 除非 SPMSC1 的 LVDSE 位置 1。假如 LVDSE 和都被置 1, 在停止模式 LVD 工作功耗将更大些。

5.6.1 上电复位 (POR) 工作

当电源刚开始给 MCU 供电时, 或当电源电压降低于 V_{POR} 电压时, POR 电路将产生一个复位条件。由于电源电压上升, LVD 电路将保持 MCU 在复位状态, 直到电源电压已经上升超过 V_{LVD} 电压。随着一个 POR, 在 SRS 寄存器的 POR 位和 LVD 位都被置 1。

5.6.2 LVD 复位工作

通过 LVDRE 位置 1, 依据一个低电压条件, LVD 可以被配置位产生复位。一个 LVD 复位产生后, LVD 系统将保持 MCU 在复位状态, 直到电源电压上升高于 V_{LVD} 。随着 LVD 复位或 POR 后 SRS 寄存器的 LVD 位被置 1。

5.6.3 LVD 中断工作

当一个低电压条件被检测, LVD 电路利用 SPMSC1 寄存器被配置为中断工作 (LVDE 置 1, LVDIE 置 1, 和 LVDRE 清零), SPMSC1 寄存器的 LVDF 位将被置 1 并且一个 LVD 中断请求将产生。

5.7 实时中断 (RTI)

实时中断功能用于产生定期中断。RTI 时钟可以是 1-kHz 内部参考时钟或 ICS 模块的被调整的 32-kHz 内部参考时钟。32-kHz 内部参考时钟被 32 分频产生一个被调整的 1-kHz 时钟用于应用程序需要更精确的实时中断。SRTISC 的 RTICLKS 位用于选择 RTI 时钟源。当 MCU 在运行、等待或停止模式时, 用于 RTI 的 1-kHz 和 32-kHz 时钟源都能使用。

SRTISC 寄存器包含一个只读状态标志位, 一个只写应答位, 和一个用于选择七种唤醒周期中的一种或 RTI 禁止的 3-位控制值 (RTIS)。RTI 由一个逻辑中断允许, RTIE, 用于允许屏蔽实时中断。RTIS 位清零禁止 RTI, 并且不会中断产生。关于这个寄存器的详细介绍见 [Section 5.8.4, “系统实时中断状态和控制寄存器 \(SRTISC\)”](#)。

5.8 复位，中断，及系统控制寄存器和控制位

对于所有寄存器绝对地址分配，参考 [Chapter 4 存储器](#) 直接页寄存器概要。此章节仅涉及到寄存器和控制位的名称。有一个飞思卡尔提供的同等的或页眉文件用于将这些名称转换成合适的绝对地址。

SOPT 寄存器的一些控制位与工作模式有关。虽然这些位在这里提供了简单描述，在 [Chapter 3 工作模式](#) 工作模式” 章节有更详细相关功能阐释。

5.8.1 系统复位状态寄存器 (SRS)

此高页寄存器包含只读状态标志位指出大多数最近复位源。当一个调试工具通过 BDC_RESET 命令强迫复位时，在 SRS 的所有状态位将被清零。写任何值到这个寄存器地址使 COP 看门狗定时器清零且不会影响这个寄存器的内容。这些位的复位状态取决于 MCU 产生复位的原因。

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	ILAD	0	LVD	0
W	写任何值到 SRS 地址清零 COP 看门狗定时器							
POR:	1	0	0	0	0	0	1	0
LVR:	0	0	0	0	0	0	1	0
Any other reset:	0	Note 1	Note 1	Note 1	Note 1	0	0	0

1. 任何这些复位源在复位入口时间时是活动的都将导致相关位被置位；相关源的位在复位入口时间不是活动的将被清零。

Figure 5-1. 系统复位状态 (SRS)

Table 5-2. SRS 域描述

域	描述
7 POR	上电复位 — 通过上电检测逻辑导致复位。因为内部电源电压在那个时候正在上升，低电压复位 (LVR) 状态位也被置位用于表明在内部电源低于 LVR 门限产生复位。 0 POR 没有产生复位 1 POR 产生复位
6 PIN	外部复位管脚 — 一个在外部复位管脚上的有效低电平导致复位。 0 外部复位管脚没有导致复位 1 外部复位管脚导致复位
5 COP	计算机正常工作 (COP) 看门狗 — 通过 COP 看门狗定时器定时溢出导致复位。COPE = 0 时此复位源被锁住。 0 没有 COP 定时溢出导致复位 1 COP 定时溢出导致复位

Table 5-2. SRS 域描述 (continued)

域	描述
4 ILOP	非法操作码 — 试图执行一个无效或非法操作码 导致复位。假如 SOPT 寄存器 STOPE = 0 停止模式被禁止, STOP 指令被认为是非法的。假如 BDCSC 寄存器中 ENBDM = 0 主动背景模式禁止, BGND 指令被认为是非法的。 0 没有非法操作码导致复位 1 非法操作码导致复位
3 ILAD	非法地址 — 试图访问无效存储器地址的数据或指令导致复位 0 没有非法地址导致复位 1 非法地址导致复位
1 LVD	低电压检测 — 假如 LVDRE 位置 1 且电源下跌低于 LVD 跌落电压, 将导致 LVD 复位。 0 没有 LVD 跌落或 POR. 导致复位 1 LVD 跌落或 POR. 导致复位

5.8.2 系统选项寄存器 (SOPT)

此高页寄存器是一个只写一次的寄存器, 因此只有复位后的第一次写是有效的。它能在任何时候被读出。任何后来企图写 SOPT (有意或无意) 被忽略用于防止这些敏感设置被意外改变。SOPT 必须在用户复位初始化程序被写设置期望的控制, 即使这些期望设置与复位默认设置相同。

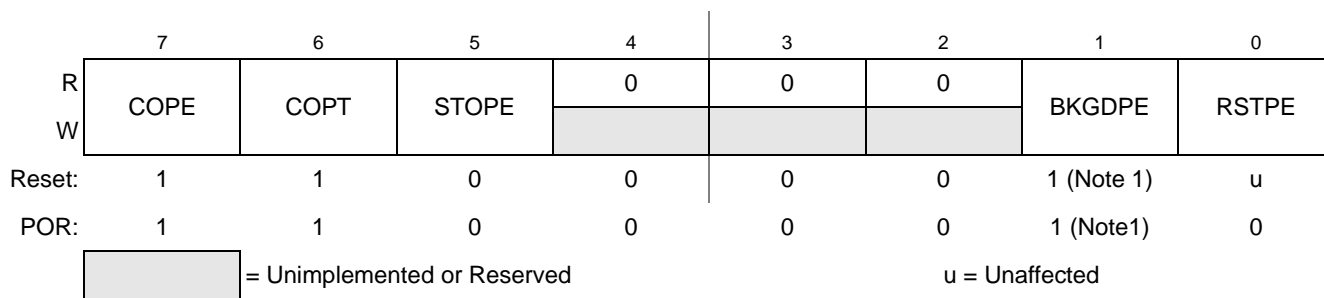


Figure 5-2. 系统选项寄存器 1 (SOPT)

1. 当芯片复位进入正常模式(复位期间MS为高), 假如Flash安全机制禁止 (SECD = 1)BKGDPE被置位; 假如Flash安全机制使能 (SECD = 1)BKGDPE 被清零; 当芯片复位进入主动BDM模式(复位期间MS为低), BKGDPE i总是被置1因此BDM通讯被允许。

Table 5-3. SOPT 寄存器域描述

Field	Description
7 COPE	COP 看门狗使能 — 此仅能写一次的位选择 COP 看门狗是否使用。 0 COP 看门狗定时器禁止 1 COP 看门狗定时器使能 (定时溢出强迫复位)
6 COPT	COP 看门狗定时溢出 — 此仅能写一次的位选择 COP 定时溢出的周期 0 选择短定时溢出周期 1 选择长定时溢出周期
5 STOPE	停止模式使能 — 此仅能写一次的位用于使能停止模式。假如停止模式禁止且用户程序试图执行一条 STOP 指令, 导致一个非法操作码复位 0 停止模式禁止 1 停止模式使能

Table 5-3. SOPT 寄存器域描述

Field	Description
1 BKGDPPE ⁽¹⁾ , (2)	背景调试模式管脚使能 — 此仅能写一次的位当置位时使能 PTA3/ACMPO/BKGD/MS 管脚功能作为 pinBKGD/MS。当清零, 管脚功能作为两种输出可选功能中的一种。任何 MCU 复位后这管脚默认作为 BKGD/MS 功能。 0 PTA3/ACMPO/BKGD/MS 管脚功能作为 PTA3 或 ACMPO。 1 PTA3/ACMPO/BKGD/MS 管脚功能作为 BKGD/MS。
0 RSTPE	RESET 管脚使能 — 当置位时, 此仅能写一次的位使能 PTA2/KBIP2/TCLK/RESET/V _{PP} 管脚功能作为 RESET。当清零, 管脚功能作为两种输入可选功能中的一种。跟随一个 MCU POR, 此管脚是只输入功能。当 RSTPE 被置位, 在 RESET 的内部上拉设备使能。 0 PTA2/KBIP2/TCLK/RESET/V _{PP} 管脚功能作为 PTA2/KBIP2/TCLK/V _{PP} 。 1 PTA2/KBIP2/TCLK/RESET/V _{PP} 管脚功能作为 RESET/V _{PP} 。

- 当芯片复位进入正常模式(复位期间MS为高), 假如Flash安全机制禁止 (SECD = 1)BKGDPPE被置位; 假如Flash安全机制使能 (SECD = 1)BKGDPPE 被清零; 当芯片复位进入主动 BDM 模式(复位期间 MS 为低), BKGDPPE 总是被置 1 因此 BDM 通讯被允许。
- BKGDPPE 仅能从 1 到 0 写一次。被用户软件从 0 到 1 写是不允许的。通过一个 POR 或对于在注释 1 中正常条件复位, BKGDPPE 能被改变成 1。

5.8.3 系统设备识别寄存器 (SDIDH, SDIDL)

此高页只读寄存器包含了用于开发系统识别 RS08 衍生或修正产品的代码, 从而允许开发工具软件配置特定 MCU 的专用存储器块, 寄存器和控制位。

	7	6	5	4	3	2	1	0
R	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
W								
Reset:	0 (Note 1)	0 (Note 1)	0 (Note 1)	0 (Note 1)	1	0	0	0

□ = Unimplemented or Reserved

- 被定义到这些位的修订码反映了当前芯片版本级别。

Figure 5-3. 系统设备识别寄存器 — 高 (SDIDH)

Table 5-4. SDIDH 寄存器域描述

域	描述
7:4 REV[3:0]	修订码 — 地址 SDIDH 的高顺序 4 位被硬编码反映了当前芯片版本级别 (0–F)。
3:0 ID[11:8]	部分识别码 — 在 RS08 系列每个衍生出的都有一个独一无二的识别码。MC9RS08K2 被硬编程值 \$0800。也可见在 Figure 5-4 中的 ID 位。

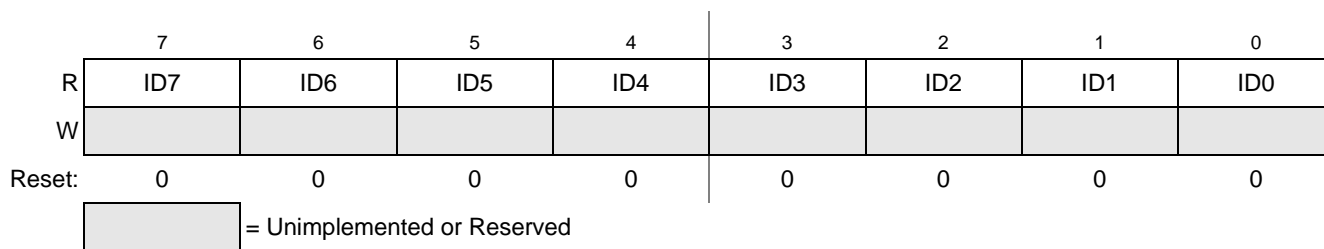


Figure 5-4. 系统设备识别寄存器 — 低 (SDIDL)

Table 5-5. SDIDL Register Field Descriptions

Field	Description
7:0 ID[7:0]	部分识别码 — 在 RS08 系列每个派生出的都有一个独一无二的识别码。MC9RS08K2 被硬编程值 \$0800。也可见在 Figure 5-3 中的 ID 位。

5.8.4 系统实时中断状态和控制寄存器 (SRTISC)

此高页寄存器包含了 RTIde 状态和控制位。

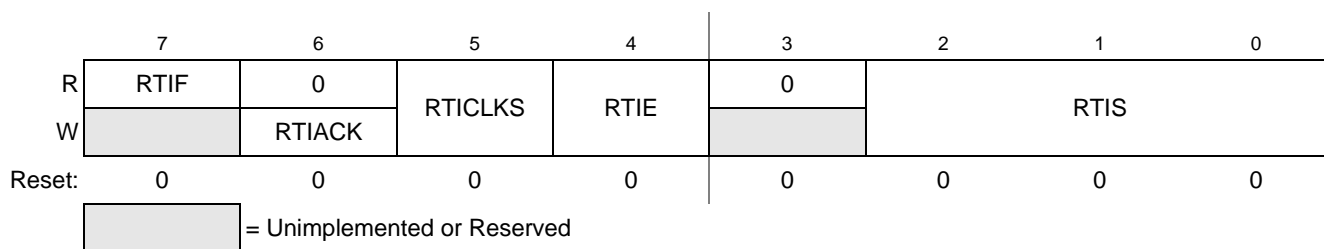


Figure 5-5. 系统 RTI 状态和控制寄存器 (SRTISC)

Table 5-6. SRTISC 寄存器域描述

域	描述
7 RTIF	时钟中断标志位 — 这个只读状态位显示了周期唤醒定时器已经定时溢出。 0 周期唤醒定时器没有定时溢出。 1 周期唤醒定时器定时溢出。
6 RTIACK	实时中断应答 — 这个只写位用于应答实时中断请求 (写 1 清除 RTIF)。写 0 无意义或影响。读总是返回 0。
5 RTICLKS	实时中断时钟选择 — 这读 / 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 1-kHz 振荡器。 1 实时中断请求时钟源是内部可调的 32-kHz 振荡器 (ICS 模块) 和在 RT 逻辑被 32 分频产生一个可调的 1-kHz 时钟源用于 RTI 计数器。
4 RTIE	实时中断使能 — 此读写位使能实时中断。 0 实时中断禁止 1 实时中断允许
2:0 RTIS	实时中断延时选择 — 这些读 / 写位选择 RTI 周期。见 Table 5-7。

Table 5-7. 实时中断周期

RTIS	RTI 定时溢出 ⁽¹⁾
000	Disable RTI
001	8 ms
010	32 ms
011	64 ms
100	128 ms
101	256 ms
110	512 ms
111	1.024 s

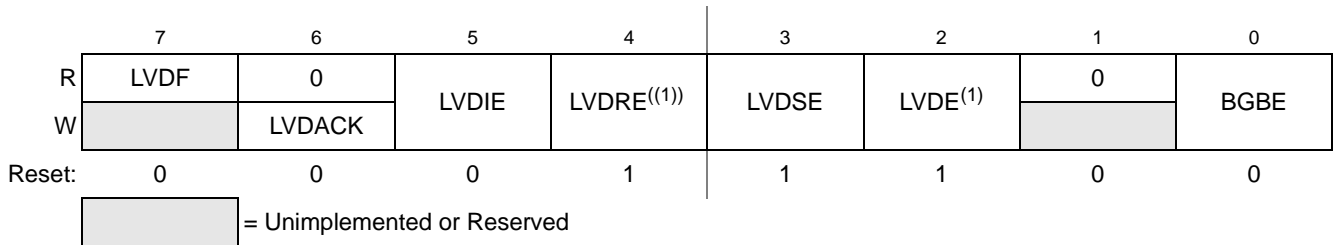
1. 定时溢出值基于 1 ms 周期的 RTI 时钟源。内部 1-kHz 源公差参考 t_{RTI} (Table A-8) 和来自 ICS 的内部 32-kHz (Table A-7)。

注意:

为了停止模式下完全关闭内部 1-kHz 振荡器, RTIS 位必须被选择为 %000 和 RTICLKS 位置 1。

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支持低电压检测功能, 并且允许带隙参考电压用于 ACMP 和 LVD 模块。



1. 复位后此位仅能写一次。额外的写被忽略。

Figure 5-6. 系统电源管理状态和控制 1 寄存器 (SPMSC1) 寄存器

Table 5-8. SPMSC1 寄存器域描述

Field	Description
7 LVDF	低电压检测标志位 — 倘若 LVDE = 1, 此只读状态位指出一个低电压检测事件。
6 LVDACK	低电压检测应答 — T 只写位用于应答低电压检测错误 (写 1 清除 LVDF)。读总是返回 0。
5 LVDIE	低电压检测中断使能 — 此位使能用于 LVDF 硬件中断请求 0 硬件中断禁止 (用于登记). 1 当 LVDF = 1 请求一个硬件中断

Table 5-8. SPMSC1 寄存器域描述 (continued)

Field	Description
4 LVDRE	低电压检测复位使能 — 此只写位使能低电压检测事件产生一个硬件复位 (倘若 LVDE = 1). 0 LVDF 不产生中断复位 1 当 LVDF = 1 强迫 MCU 复位
3 LVDSE	低电压检测停止使能 — 倘若 LVDE = 1, 这读 / 写位确定当 MCU 在停止模式时低电压检测功能是否工作。 0 停止模式期间低电压检测禁止。 1 停止模式期间低电压检测使能。
2 LVDE	低电压检测使能 — 此只写位使能低电压检测逻辑和限制这个寄存器中的其它位的操作。 0 LVD 逻辑禁止 1 LVD 逻辑使能
0 BGBE	待隙缓存使能 — 通过在内部通道中之一的 ACMP 模块, 这位使能一个内部缓存用于待隙电压参考。 0 待隙缓存禁止 1 待隙缓存使能

5.8.6 系统中断等待寄存器 (SIP1)

此高页寄存器包含模块等待中断状态。

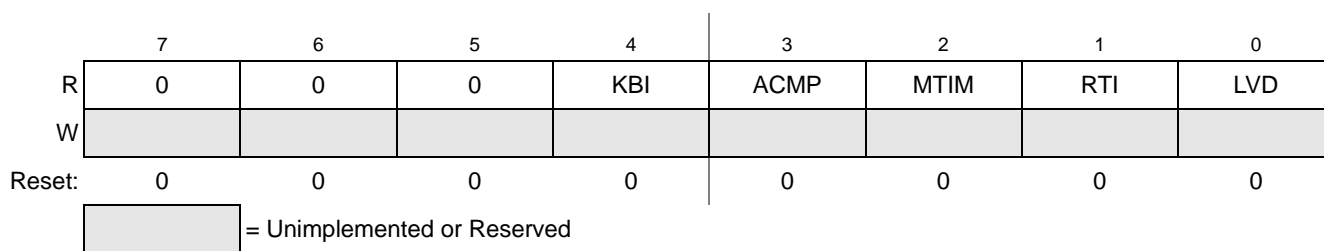


Figure 5-7. 系统中断等待寄存器 (SIP1)

Table 5-9. SIP1 寄存器域描述

域	描述
4 KBI	键盘中断等待 — 此只读位显示有一个来自 KBI 模块的等待中断。清除 KBISC 寄存器中的 KBF 位将此位清零。读也将此位清零。 0 没有等待 KBI 中断; 例如, KBF 标志位和 / 或 KBIE 位被清零。 1 有等待 KBI 中断; 例如, KBF 标志位和 / 或 KBIE 位被置位。
3 ACMP	模拟比较器中断等待 — 此只读位显示有一个来自 ACMP 模块的等待中断。清除 ACMPSC 寄存器中的 ACF 位将此位清零。读也将此位清零。 0 没有等待 ACMP 中断; 例如, ACF 标志位和 / 或 ACIE 位被清零。 1 有等待 ACMP 中断; 例如, ACF 标志位和 / 或 ACIE 位被置位。
2 MTIM	模定时器中断等待 — 此只读位显示有一个来自 MTIM 模块的等待中断。清除 MTIMSC 寄存器中的 TOF 位将此位清零。读也将此位清零。 0 没有等待 MTIM 中断; 例如, TOF 标志位和 / 或 TOIE 位被清零。 1 有等待 MTIM 中断; 例如, TOF 标志位和 / 或 TOIE 位被置位。

Table 5-9. SIP1 寄存器域描述 (continued)

域	描述
1 RTI	<p>实时中断等待 — 此只读位显示有一个来自 RTI 模块的等待中断。清除 SRTISC 寄存器中的 RTIF 位将此位清零。读也将此位清零。</p> <p>0 没有等待 RTI 中断；例如， RTIF 标志位和 / 或 RTIE 位被清零。</p> <p>1 没有等待 RTI 中断；例如， RTIF 标志位和 / 或 RTIE 位被置位。</p>
0 LVD	<p>低电压检测中断等待 — 此只读位显示有一个来自低电压模块的等待中断。清除 SPMSC1 寄存器中的 LVDF 位将此位清零。读也将此位清零。</p> <p>0 没有等待 LVD 中断；例如， LVDF 标志位和 / 或 LVDE 位被清零。</p> <p>1 没有等待 LVD 中断；例如， LVDF 标志位和 / 或 LVDE 位被置位。</p>

Chapter 6 并行输入 / 输出控制

本章节讲述了与并行输入 / 输出和管脚控制相关的软件控制。MC9RS08K2 有一个并行 I/O 端口，此端口在 6 脚封装包括两个 I/O 管脚或在 8 脚封装包括 4 个 I/O 管脚，一个只输出脚，和一个只输入脚。更多关于管脚分配和外部硬件考虑见 [MC9RS08KA2 产品概述](#) 第 2 章管脚与连接”。

Table 2-1 显示所有 I/O 管脚与片内外设功能复用。外设模块有比 I/O 更大的优先权，因此当外设允许，则与此复用的管脚的 I/O 功能禁止。复位后，复用外设功能被禁止因此这些管脚被 I/O 功能控制。所有这些 I/O 被配置为输入 (PTADDn = 0)，相关的上拉 / 下拉设备禁止，只输出管脚除外，它默认为 BKGD/MS 功能。

并行 I/O 读写通过端口数据寄存器完成。方向，输入或输出，被端口数据方向寄存器控制。Figure 6-1 显示了一个独立管脚并行 I/O 端口功能的方框图。

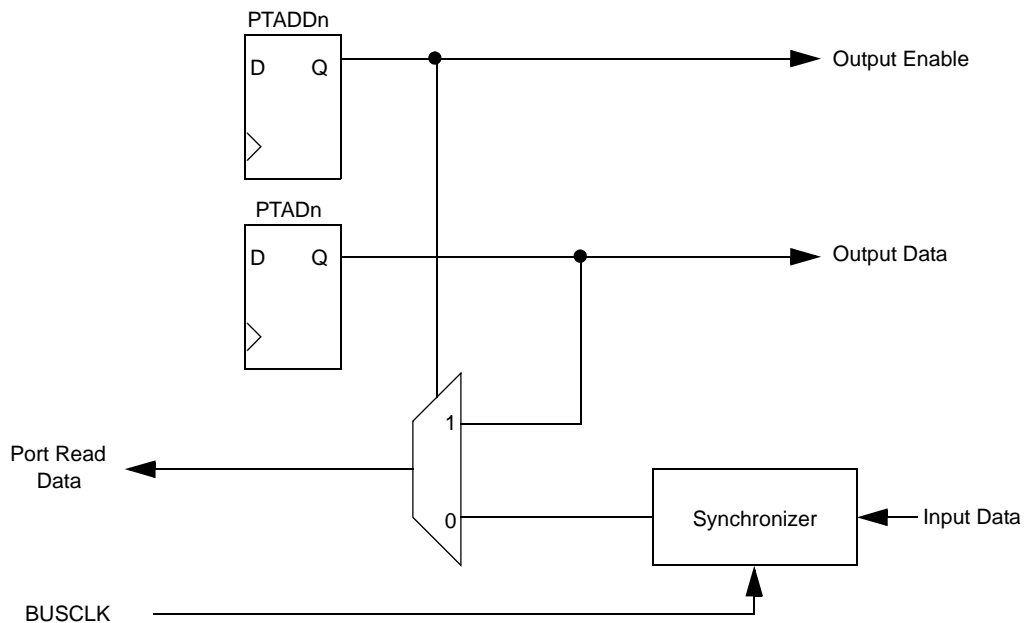


Figure 6-1. 并行 I/O 方框图

数据方向控制位 (PTADDn) 决定相关管脚的输出缓冲区是否允许，也控制端口数据寄存器读来源。相关管脚的输入缓冲区总是允许，除非这个管脚作为模拟功能或是一个只输出管脚。

当一个管脚的复用数字功能允许，输出缓冲区被这些复用功能控制。然而，数据方向寄存器位将继续控制端口数据寄存器的方向。

并行输入 / 输出控制

当一个管脚的复用模拟功能允许，输入和输出缓冲区都禁止。从任何端口数据位读出值为 0，这个位是一个输入 (PTADDn = 0) 且输入缓冲区禁止。通常，一个具有可选择数字功能和模拟功能的管脚，模拟功能具有更高的优先级。因此当数字和模拟功能同时允许时，模拟功能控制此管脚。

在改变端口管脚方向成输出前，写端口数据寄存器是个好的编程习惯。这保证此管脚不会被此端口数据寄存器的老数据驱动。

与并行 I/O 端口相关的是位于高页寄存器空间的独立工作的并行 I/O 寄存器。这些寄存器用于控制管脚上拉 / 下拉和斜率。更详细信息见 Section 6.3, “管脚控制寄存器”。

6.1 在低功耗模式下管脚状态

在等待和停在模式，所有管脚状态被保持，这是因为内部逻辑维持供电。恢复后，所有管脚功能与进入停在模式前一样。

6.2 并口 I/O 寄存器

本节提供了与并行 I/O 端口有关的寄存器的信息。并行 I/O 寄存器位于存储器的地址小于 \$001F，因此短型和直接寻址方式指令可以使用。

所有并行 I/O 绝对地址分配，参考 Chapter 4 存储器”中的表格。本节仅通过名字涉及到寄存器和控制位。飞思卡尔半导体提供等同的或页眉文件将这些名字转换成合适的绝对地址。

6.2.1 端口 A 寄存器

本节描述了端口 A 并行 I/O 功能，通过数据和数据方向寄存器控制。

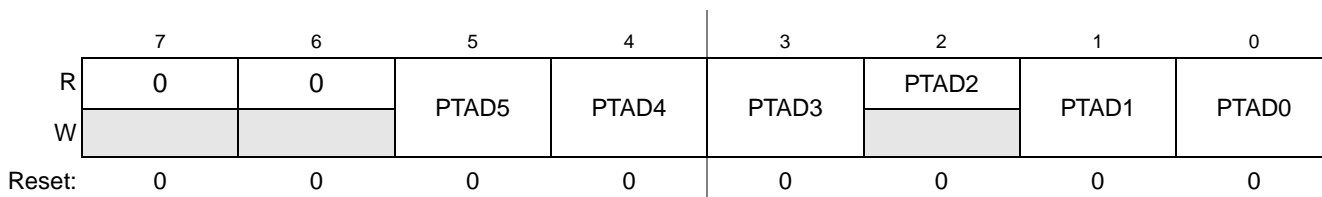


Figure 6-2. 端口 A 数据寄存器 (PTAD)

Table 6-1. PTAD 寄存器区域描述

Field	描述
5:0 PTAD[5:0]	端口 A 数据寄存器位 — 当端口 A 管脚是输入，读管脚返回逻辑电平。当端口 A 被配置为输出，读返回被写入这个寄存器最新的值。写入值被锁住这个寄存器所有的位。由于端口 A 管脚被配置为输出，逻辑电平驱动相关的管脚。 复位后 PTAD 都为 0，但这些 0 电平不用于驱动相关管脚，这是因为复位也配置所有端口管脚为高阻态输入，且上拉 / 下拉禁止。

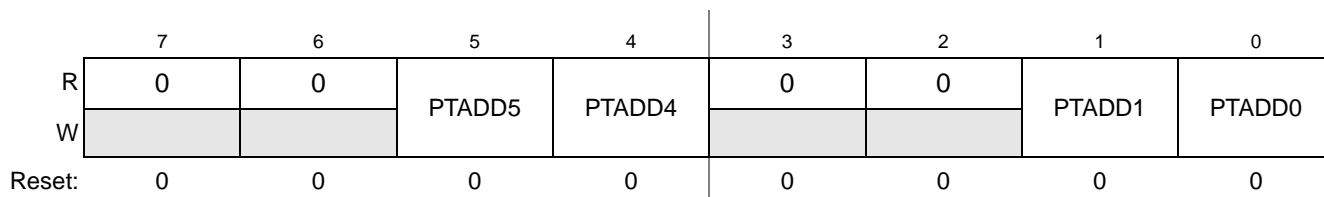


Figure 6-3. 端口 A 数据方向寄存器 (PTADD)

Table 6-2. PTADD 寄存器描述

Field	描述
5:4,1:0 PTADD[5:4,1:0]	端口 A 数据方向位 — 这些读 / 些位控制端口 A 的方向并从 PTAD 读出。 0 输入 (输出驱动器禁止) 且读返回此管脚值 1 端口 A 位输出驱动允许且 PTAD 读返回 PTADn 的内容。

6.3 管脚控制寄存器

此节提供了用于管脚功能的并行 I/O 端口相关的寄存器信息。

关于管脚控制寄存器绝对地址分配，参考 [Chapter 4 存储器](#)。本节仅通过名字涉及到寄存器和控制位。飞思卡尔半导体提供等同的或页眉文件将这些名字转换成合适的绝对地址。

在本节与端口 A 相关管脚被提供的寄存器控制。这些寄存器控制管脚上拉 / 下拉和端口斜率，不依赖于并行 I/O 寄存器。

6.3.0.1 内部上拉装置允许

一个内部上拉装置可以通过设置上拉装置允许寄存器 (PTAPEn) 的相关位允许允许每个端口管脚内部上拉。假如此管脚被并行 I/O 控制逻辑配置为输出或即使相关上拉装置允许寄存器置位但有复用外设输出功能，上拉装置禁止。假如管脚被配置为一个模拟功能，上拉装置也禁止。

并行输入 / 输出控制

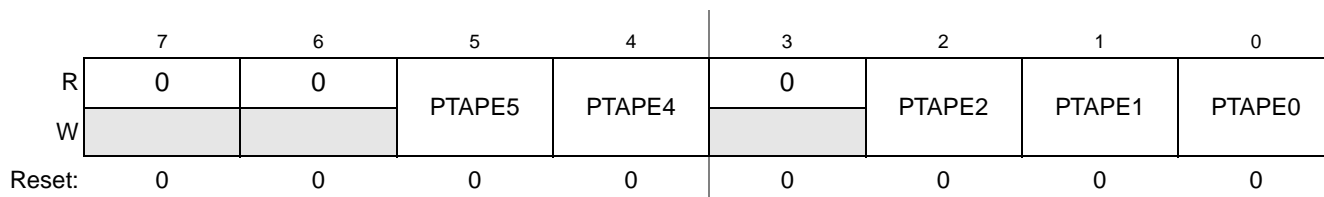


Figure 6-4. 端口 A 寄存器内部上拉装置允许 (PTAPE)

Table 6-3. PTAPE Register Field Descriptions

Field	Description
5:4,2:0 PTAPE[5:4,2:0]	内部拉设备使能用于端口 A 位 — 这些控制位中的每一位决定内部拉设备是否使能相关 PTA 管脚。当端口 A 管脚被配置位输出，这些位无效且内部上拉设备禁止。 0 内部上拉设备禁止用于端口 A 位 n。 1 内部上拉设备允许用于端口 A 位 n。

6.3.0.2 上拉 / 下拉控制

依据相关 PTAPE 位，上拉 / 下拉控制用于选择上拉或下拉装置使能。

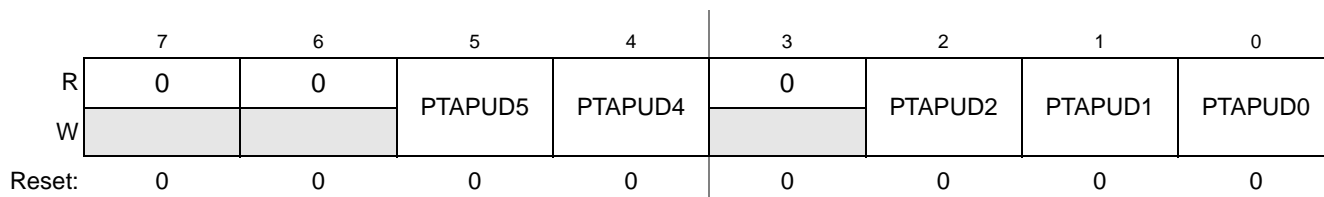


Figure 6-5. 上拉 / 下拉设备控制用于端口 A (PTAPUD)

Table 6-4. PTAPUD 寄存器域描述

域	描述
5:4,2:0 PTAPUD[5:4,2:0]	上拉 / 下拉设备控制用于端口 A 位 — 这些控制位的每一位决定内部上拉或下拉设备是否选择用于相关的 PTA 管脚。此实际的上拉 / 下拉设备仅被使能的相关 PTAPE 位使能。 0 内部上拉设备被选择用于端口 A 位 n。 1 内部下拉设备被选择用于端口 A 位 n。

6.3.0.3 输出斜率控制允许

斜率控制可以被斜率控制寄存器 (PTASEn) 相关位允许用于每个端口管脚。当被允许使用时，斜率控制限制输出电平变化的斜率，可以减小 EMC 发射。当管脚被配置为输入时斜率控制无效。

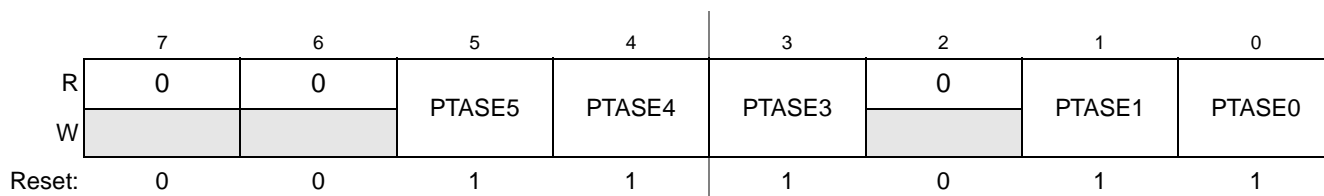


Figure 6-6. S 斜率使能用于端口 A 寄存器 (PTASE)

Table 6-5. PTASE 寄存器域描述

域	描述
5:3;1:0 PTASE[5:3;1:0]	<p>输出斜率使能用于端口 A 位 — 这些控制位中的每一位都决定输出斜率控制是否使能相关的 PTA 管脚。当端口 A 管脚被配置位输入，这些位无效。</p> <p>0 输出斜率控制禁止用于端口 A 位 n。</p> <p>1 输出斜率控制允许用于端口 A 位 n。</p>

Chapter 7 键盘中断 (RS08KBIV1)

7.1 介绍

键盘中断模块提供了独立的允许外部中断源。

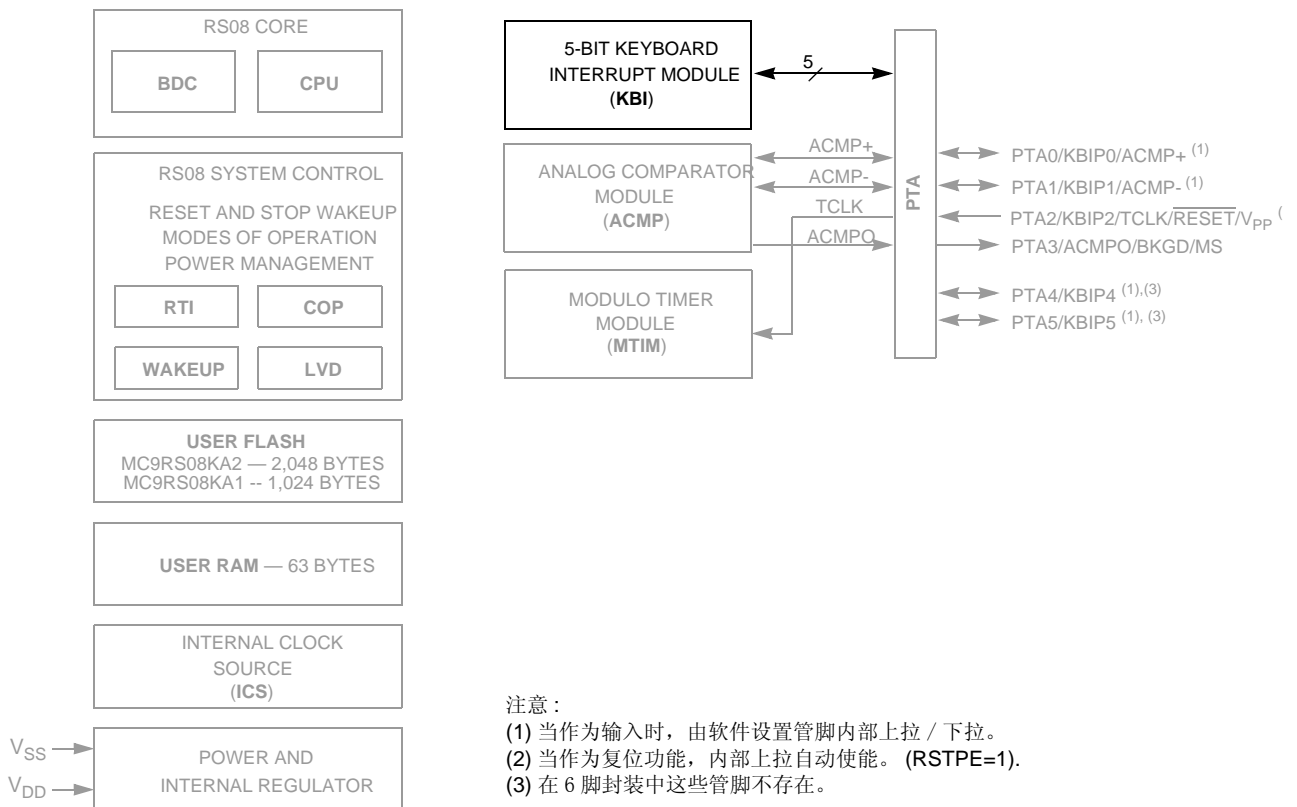


Figure 7-1. MC9RS08KA2 系列标示 KBI 块的框图

7.1.1 特点

KBI 特点包括：

- 每个键盘中断管脚有一个独立的管脚允许位
- 每个键盘中断管脚可被编程作为只下降沿 (或上升沿)，或既是下降沿又是低电平 (或既是上升沿又是高电平) 中断触发
- 一个软件允许键盘中断
- 从低功耗模式退出

键盘中断 (RS08KBIV1)

7.1.2 工作模式

此章定义了等待、停止和背景调试模式下 KBI 工作。

7.1.2.1 等待模式下工作

假如执行 WAIT 指令之前被允许，KBI 在等待模式下可以继续工作。因此，假如 KBI 中断被允许 (KBIE = 1)，一个被允许的 KBI 管脚 (KBPE_n = 1) 可以用于将 MCU 退出等待模式。

7.1.2.2 停止模式下工作

假如执行 STOP 指令前被允许，KBI 可以在停止模式工作。因此，假如 KBI 中断允许 (KBIE = 1)，一个允许的 KBI 管脚 (KBPE_n = 1) 能用于从停止模式唤醒 MCU。

7.1.2.3 在活动背景模式下工作

当单片机工作在活动的背景模式，KBI 将继续正常工作。

7.1.3 方框图

键盘中断模块方框图如 Figure 7-2 所示：

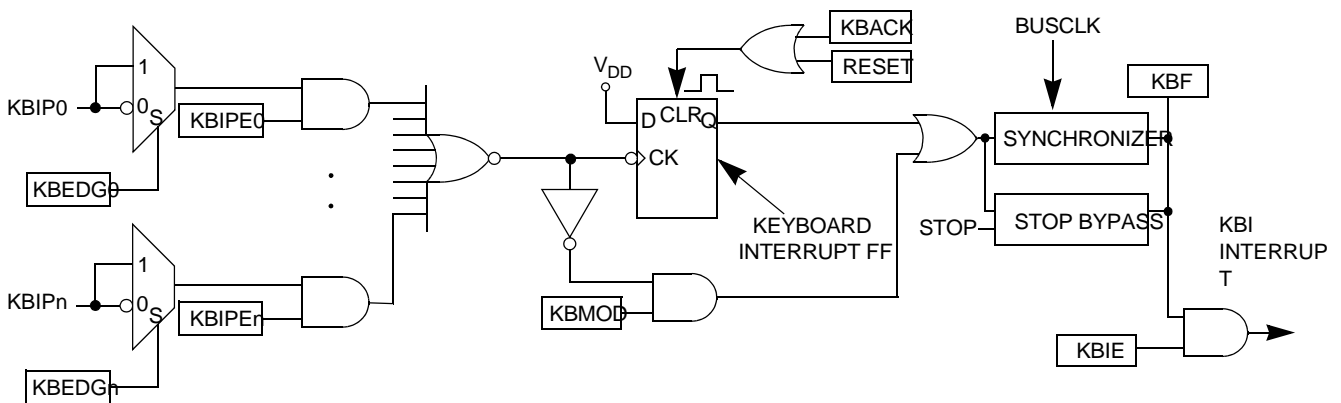


Figure 7-2. 键盘中断 (KBI) 方框图

7.2 外部信号描述

KBI 输入管脚可以用于检查下降沿，或既是下降沿又是低电平中断请求。KBI 输入管脚也可以用于检查上升沿，或既是上升沿有时高电平中断请求。

信号优先级见 Table 7-1。

Table 7-1. 信号优先级

信号	功能	I/O
KBIP _n	键盘中断管脚	I

7.3 寄存器定义

KBI 包括三个寄存器：

- 一个 8 位管脚状态和控制寄存器
- 一个 8 位管脚允许寄存器
- 一个 8 位边沿选择寄存器

参考 Chapter 4 存储器的直接页寄存器概要，包含所有 KBI 寄存器绝对地址分配。本节仅通过它们的名字涉及到寄存器和控制位。

Table 7-2 列出了 KBI 寄存器。

Table 7-2. KBI 寄存器概述

Name		7	6	5	4	3	2	1	0
KBISC	R	0	0	0	0	KBF	0	KBIE	KBMOD
	W						KBACK		
KBIPE	R	0	0	KBIPE5	KBIPE4	0	KBIPE2	KBIPE1	KBIPE0
	W								
KBIES	R	0	0	KBEDG5	KBEDG4	0	KBEDG2	KBEDG1	KBEDG0
	W								

7.3.1 KBI 状态和控制寄存器 (KBISC)

KBISC 包含状态标志和控制位，用于配置 KBI。

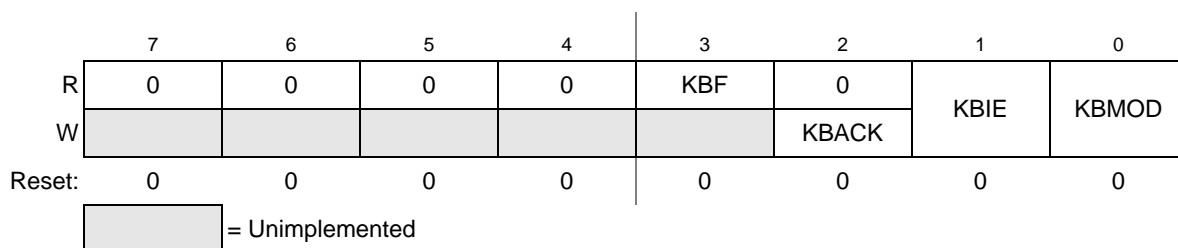


Figure 7-3. KBI 状态和控制寄存器 (KBISC)

键盘中断 (RS08KBIV1)

Table 7-3. KBISC 寄存器域描述

Field	描述
3 KBF	键盘中断标志 — KBF 指出有一个键盘中断被检测到。对 KBF 写无效。 0 没有键盘中断被检测到 1 键盘中断被检测到
2 KBACK	键盘应答 — 写一个 1 到 KBACK 是标志清除机制部分。读 KBACK 总是为 0。
1 KBIE	键盘中断允许 — KBIE 允许键盘中断请求 0 键盘中断请求不允许 1 键盘中断请求允许
0 KBMOD	键盘检测模式 — KBMOD (连同 KBEDG 位) 控制键盘中断管脚检测模式 0 键盘仅检测边沿 1 键盘既检测边沿又检测电平

7.3.2 KBI 管脚允许寄存器 (KBIPE)

KBIPE 包含管脚允许控制位。

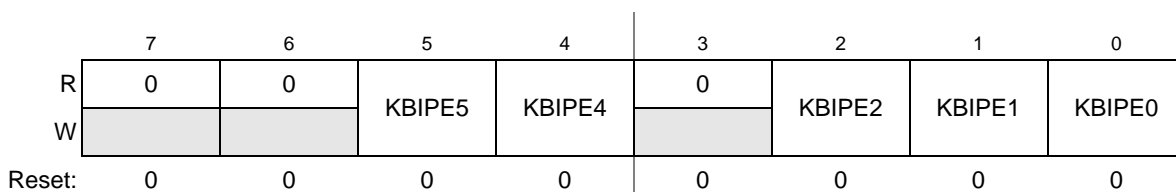


Figure 7-4. KBI 管脚允许寄存器 (KBIPE)

Table 7-4. KBIPE 寄存器域描述

Field	描述
5,4, 2:0 KBIPE _n	键盘管脚允许 — 每个 KBIPE _n 位允许相关的键盘中断管脚 0 相关管脚不允许作为键盘中断 1 相关管脚允许作为键盘中断

7.3.3 KBI 边沿选择寄存器 (KBIES)

KBIES 包含边沿选择控制位。

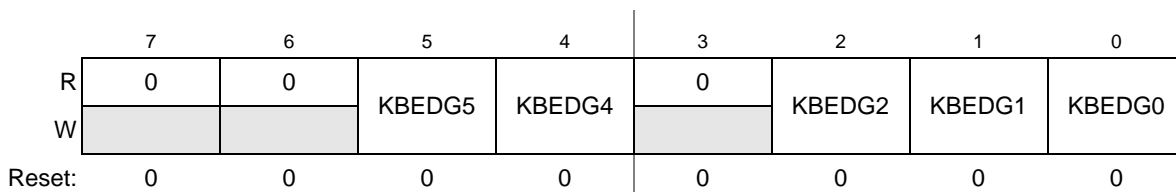


Figure 7-5. KBI 边沿选择寄存器 (KBIES)

Table 7-5. KBIES 寄存器域描述

Field	描述
5,4, 2:0 KBEDGn	键盘边沿选择 — 每个 KBEDGn 位选择下降沿 / 低电平或上升沿 / 高电平相关管脚的功能 0 下降沿 / 低电平 1 上升沿 / 高电平

7.4 功能描述

键盘中断 (KBI) 模块也被用于作为额外的外部中断输入和作为从停止或等待低功耗模式唤醒 MCU 的外部手段。

KBI 模块允许它的管脚担当附加的中断源。在键盘中断管脚允许寄存器 (KBIPEN) 写 KBIPEN 位独立地允许或禁止每个 KBI 管脚。基于键盘中断状态和控制寄存器 (KBISC) 的 KBMOD 位, 每个 KBI 管脚可以被配置位边沿触发或边沿和电平触发。边沿触发可以被软件编程上升沿或下降沿; 电平是高或低。边沿极性或边沿和电平触发的优先级通过键盘中断边沿选择寄存器 (KBIES) 的 KBEDGn 位选择。

异步逻辑用于检测边沿。检测一个边沿之前, 允许键盘输入脚必须是被拉高的逻辑电平。当一个允许的键盘输入信号被检测到在一个总线周期是逻辑 1 (被拉高的电平) 且在下个周期是逻辑 0 (拉低的电平), 一个下降沿被检测到。当输入信号被检测到在一个总线周期是逻辑 0 且下个周期是逻辑 1, 上升沿被检测到。

7.4.1 仅边沿触发

在一个允许 KBI 管脚上的一次有效边沿将 KBISC 中的 KBF 位置 1。假如 KBISC 中的 KBIE 被置 1, 中断请求将被送到 CPU。通过写一个 1 到 KBISC 的 KBACK 位实现将 KBF 清零。

7.4.2 边沿和电平触发

在一个允许 KBI 管脚上的一次有效边沿或电平将 KBISC 中的 KBF 位置 1。假如 KBISC 中的 KBIE 被置 1, 中断请求将被送到 CPU。倘若所有允许键盘输入管脚是被拉为高电平, 通过写一个 1 到 KBISC 的 KBACK 位实现将 KBF 清零。写 1 到 KBACK 试图将 KBF 清零时, 假如允许 KBI 管脚被拉低为低电平, KBF 将保持为 1。

7.4.3 KBI 上拉 / 下拉设备

当一个 KBI 管脚被使能, KBI 管脚不会自动配置内部上拉 / 下拉设备。通过配置相关联的 I/O 端口拉设备允许寄存器和上拉 / 下拉控制寄存器 (PTAPUD) 内部拉设备被使用。

7.4.4 KBI 初始化

当一个键盘中断管脚刚开始被允许使用时, 可能会产生一个错误的键盘中断标志。在键盘初始化时为了避免错误的中断请求产生, 用户应该按照以下步骤:

1. 将 KBISCM 中的 KBIE 为清零屏蔽键盘中断。
2. 假如使用内部上拉 / 下拉设备, 配置相关联的 I/O 端口上拉 / 下拉设备。

键盘中断 (RS08KBIV1)

3. 通过设置 KBIES 中合适的 KBEDG_n 位使能 KBI 优先级。
4. 通过设置 KBIPE 中合适的 KBIPE_n 位使能 KBI 管脚。
5. 写 1 到 KBISC 中的 KBACK 位清除任何错误中断。
6. 将 KBISC 中的 KBIE 位置 1 允许中断。

Chapter 8 中央处理器单元 (RS08CPUV1)

8.1 介绍

本章是 RS08 家族 CPU 的寄存器、寻址模式和指令集等信息介绍。更详细讨论参考 RS08 核参考手册第一卷，飞思卡尔半导体文档单号 RS08RMv1。

RS08 CPU 被开发的目的是极低成本嵌入式应用，用于一个独立处理的设计方法。

RS08 核的主要特点是：

- 改进型的编程模型
- HCS08 指令集的子集，较少扩展指令
- 很小的指令集适用于价格敏感的嵌入式应用
- 用于屏蔽程序计数器处理的新指令，SHA 和 SLA
- 新的短型和小型寻址方式用于代码优化
- 最多 16K 字节存储器空间
- 复位将从 \$3FFD 取出首条指令
- STOP 和 WAIT 指令支持低功耗模式
- 调试和 FLASH 编程支持使用背景调试控制器模块
- 非法地址和代码被检测导致复位

8.2 程序设计器模型和 CPU 寄存器

Figure 8-1 列出了 RS08 CPU 的程序设计器模型。这些寄存器不是位于微处理器的存储器中。它们直接建立在 CPU 逻辑内部。

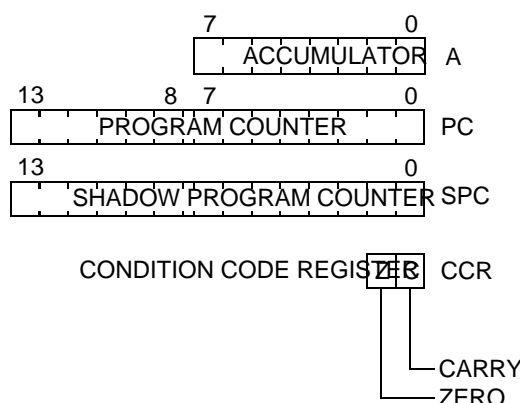


Figure 8-1. CPU 寄存器

除 CPU 寄存器外，在数据读写操作期间，有三个存储器映象寄存器与内核地址产生紧紧联合在一起。它们是变址数据寄存器 (D[X])，变址寄存器 (X) 和页选择寄存器 (PAGESEL)。这些寄存器地址分别位于 \$000E、\$000F 和 \$001F。

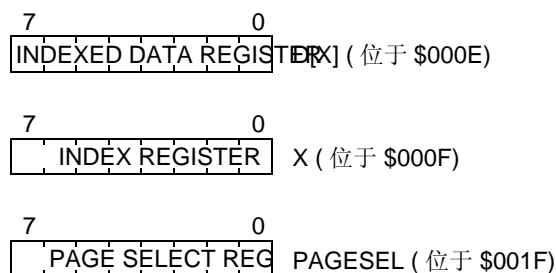


Figure 8-2. Memory Mapped Registers

累加器 (A)

此通用 8 位寄存器是 RS08 MCUs 的基本数据寄存器。通过一条装载累加器 (LDA) 指令数据能从存储器被写入 A。通过一条存储累加器 (STA) 指令在 A 中的数据可以被写入存储器。在指定存储器位置涉及到装载或存储指令时不同的寻址方式变化，允许有很大的灵活性。交换指令允许值在 A 和 SPC 高 (SHA) 之间，A 和 SPC 低 (SLA) 之间交换。

通过 ADD, SUB, RORA, INCA, DECA, AND, ORA, EOR 等指令，A 的值可以实现算术、位移和逻辑操作。在一些指令中，例如 INCA 和 LSLA，A 的值是唯一的输入操作数且结果替换 A 的值。其它一些指令，例如 ADD 和 AND，有两个操作数：A 的值和来自存储器的第二个数值。这些算术或逻辑运算的结果替代 A 的值。

一些指令，例如存储器对存储器移动指令 (MOV)，不能使用累加器。DBNZ 也不用 A，原因是它允许一个循环计数器被执行在存储器中而不是累加器。

复位后，累加器被装载入 \$00。

8.2.1 程序计数器 (PC)

程序计数器是一个 14 位的寄存器，它包含了下一条指令或被取出的操作数的地址。

正常执行时，每次执行一条指令或取出操作数，程序计数器自动增加指向下个时序存储器位置。跳转、分枝和返回操作载入程序计数器的地址不是下个时序存储器位置。这被称为流动改变。

复位期间，程序计数器被载入 \$3FFD 且程序从这个指定位置开始执行。

8.2.2 影子程序计数器 (SPC)

影子程序计数器是一个 14 位寄存器。在子程序调用期间使用 JSR 或 BSR 指令，返回地址将被存入 SPC。在子程序完成前，RTS 指令将恢复程序计数器的内容，此值来自于 SPC。

复位期间，屏蔽程序计数器被载入 \$3FFD。

8.2.3 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 CCR 内容不是直接地被读出的。使用条件分枝指令例如 BCC 和 BEQ 测试 CCR 位。寄存器的这两位直接受 BDC 接口影响。后面段落提供了关于 CCR 位详细信息和如何使用它们。Figure 8-3 表明了 CCR 位和它们位的位置。



Figure 8-3. 条件代码寄存器 (CCR)

复位后状态位 (Z 和 C) 被清零。

这两个状态位显示了算术和其它指令的结果。条件转移指令将转移到一个新的程序位置或转移后依靠 CCR 状态位的值允许程序继续执行下条指令。条件转移指令，例如 BCC、BCS 和 BNE，依靠一个单一的 CCR 位的值产生一个转移。

通常，条件转移立即跟随导致 CCR 位被更新的指令，如下面的顺序：

```

cmp    #5      ; 比较累加器 A 和 5
blo    lower   ; 假如 A 小于 5 则转移
more:  deca    ; 假如 A 大于或等于 5 执行这行指令
lower:

```

在使用的指令没有扰乱 CCR 的影响条件转移的位的时候，其它指令也许在校验和条件转移之间被执行。举例说明，一个校验在子程序或函数中被执行，条件转移直到子程序返回到主程序后才会被执行。这是个参数传输的形成（换句话说，在条件码位内信息被返回到调用的程序）。

Z — 零标志

Z 位被置 1 表明一个操作结果是 \$00。

假如等于 (BEQ) 则转移和假如不等于 (BNE) 则转移是简单的转移，此转移依赖于 Z 位的值。所有装载、存储、移动、算术、逻辑、移位和旋转指令都会导致 Z 位被更新。

C — 进位

一个加法运算后，假如源操作数大于或等于 \$80 或者操作数中的任何一个数大于或等于 \$80 并且结果小于 \$80，C 位被置 1。这相当于无符号溢出。一个存储器操作数作为被减数减去或比较一个 CPU 寄存器的内容，减法操作后，假如存储器中无符号值大于 CPU 寄存器的无符号值 C 位被置 1。这相当于一个无符号借入或下溢。

假如进位清零 (BCC) 则转移和进位置 1 (BCS) 则转移是基于 C 位值独立的转移。C 位也被用于无符号转移 BLO 和 BHS。加、减、位移和旋转指令都会导致 C 位被更新。假如位置 1 (BRSET) 则转移和位清零 (BRCLR) 则转移指令拷贝测试位到 C 位易于有效的串行转并行运算法则。置进位 1 (SEC) 和清零进位 (CLC) 允许进位被直接置 1 或清零。结合位移和旋转指令这是很有用的, 适合于通过状态信息返回到主程序的子程序。

C 位包含在位移和旋转操作中因此这些操作可以很容易地扩展到多字节操作。移位和旋转操作被当作 9 位移位, 包含一个 8 位操作数或 CPU 寄存器和 CCR 的进位。一个逻辑位移后, C 保存被移位出的操作数的第 8 位。假如下次执行旋转指令, C 位被移入操作数, 操作数最后一位移出代替在 CI 中的值。

8.2.4 变址数据寄存器 (D[X])

8 位变址数据寄存器允许用户访问在通过 X 被变址的直接页地址空间的数据。这个寄存器位于存储器映象地址 \$000E。关于 For D[X] 寄存器详细介绍, 参考 Section 8.3.8, “变址寻址方式 (IX, 被伪指令执行)”。

8.2.5 变址寄存器 (X)

8 位变址寄存器允许用户索引或访问在直接页地址空间的任何地址。此寄存器位于存储器映象地址 \$000F。关于 X 寄存器详细介绍, 参考 Section 8.3.8, “变址寻址方式 (IX, 被伪指令执行)”。

8.2.6 页选择寄存器 (PAGESEL)

8 位页选择寄存器允许用户通过一个地址从 \$00C0 到 \$00FF 页窗口访问整个 16K 地址空间的所有存储器地址。此寄存器位于存储器映象地址 \$001F。关于 PAGESEL 寄存器详细信息, 请参考 “the RS08 Core Reference Manual”。

8.3 寻址方式

只要 MCU 读写存储器信息, 一个寻址方式用于确定读写信息存储器的准确地址。本节介绍了几种寻址方式和在不同编程情况下每种如何使用。

每种操作码都会告诉 CPU 通过某种方式执行某一种操作。许多指令, 例如载入累加器 (LDA), 允许几种不同的方法指定存储器位置被载入, 且每种寻址方式的变化需要独立的操作码。所有这些变化使用相同指令助记符号, 且汇编程序依靠操作数域的语法和位置知道使用哪种操作码。在一些案例, 一些特殊的字符用于表明一种特殊的寻址方式 (例如 # [pound] 符号, 表明立即数寻址方式)。在其它一些案例, 操作数的值会告诉汇编程序使用哪种寻址方式。例如, 假如操作数地址从 \$0000 到 \$001F, 汇编程序选择短地址方式代替直接寻址方式。除允许汇编程序基于操作数地址选择寻址方式外, 汇编程序也可以通过在操作数前使用 “>” 或 “<” 前缀强迫为直接或小/短寻址方式。一些指令使用不止一种寻址方式。例如, 移动指令使用第一种寻址方式访问存储器源值和第二种寻址方式访问目的存储器位置。对于这些移动指令, 两种寻址方式在文档里都被列出。所有跳转指令使用相对 (REL) 寻址方式确定跳转的目的地。但 BRCLR、BRSET、CBEQ 和 DBNZ 也必须访问一个存储器操作数。这些指令通过用于存储器操作数的寻址方式被分类。

下面段落包含每个寻址方式如何工作和用于特别的寻址方式的汇编指令的语法。

8.3.1 隐含寻址方式 (INH)

此寻址方式用于当 CPU 内在地知道每件事情它必须完成的指令和在源代码没有寻址信息的时候。通常，CPU 需要的操作数位于 CPU 的内部寄存器，例如 LSLA, CLRA, INCA, SLA, RTS 和其它指令。少数隐含指令，包括空操作 (NOP) 和背景 (BGND)，没有操作数。

8.3.2 相对寻址方式 (REL)

相对寻址方式用于指定跳转指令的偏移地址给程序计数器。典型地，程序员用一个标签或在跳转指令操作数域用表达式指明目的地；汇编器计算位置计数器（在跳转指令后指向下个地址）和通过标签或操作数域表达式表明的地址之间的差别，这个差被称为偏移量且是一个 8 位双符号补数。汇编器存储这个偏移量在跳转指令的结果代码内。

在执行过程中，CPU 评价控制跳转的条件。假如跳转条件是真，CPU 信号扩展偏移量为一个 14 位的值，将偏移量加到当前的 PC 中，使用它作为地址获取下条指令和继续执行，而不是执行跳转指令后面的指令。由于偏移量是一个 8 位双符号补数，目的地必须在跳转指令结果代码地址从 -128 到 +127 地址范围内。

设计一个简单死循环的通用的方法是用一个跳转指令跳转到它自己。此方法有时用于在调试期间结束短的代码段。典型地，为了跳出死循环，使用调试主机（通过背景命令）停止程序，检查寄存器和存储器，或从一个新地址开始执行。此结构不用在正常的应用程序，除非程序检查到一个错误，想强迫看门狗定时器溢出（在死循环中跳转反复执行直到看门狗定时器导致复位）。

8.3.3 立即数寻址方式 (IMM)

在这种寻址方式中，操作数直接位于操作码后面。此寻址方式用于当程序员希望在某个时候程序被写入一个已知的确定的值。# (磅) 符号告诉汇编器使用的操作数作为一个数据而不是一个地址。

立即操作数大小总是 8 位。汇编器会自动切去或扩展需要的操作数与指令所需要的大小相匹配。假如提供一个 16 位操作数，大多数汇编器将产生一个警告。

当使用立即数寻址时使用 # 符号告诉汇编器是程序员的职责。汇编器不会认为它是个错误或停止 # 符号，因为声明结果依然是一个合法的指令（尽管它可能意味与预订的程序有些不同）。

8.3.4 小型寻址方式 (TNY)

TNY 寻址方式仅有能力寻址地址映象中的头 16 个字节，从 \$0000 到 \$000F。此寻址方式适合于 INC、DEC、ADD 和 SUB 指令。通过放置大部分精确计算的数据在这个存储器范围，系统可以被最优化。

由于在操作码中嵌入的是 4 位地址，仅有效的 4 位直到必须包含在指令中；这节省了程序空间和执行时间。在执行过程中，CPU 增加高 10 位到 4 位操作数地址，结合成 14 位地址 (\$000x) 访问预定的操作数。

8.3.5 短型寻址方式 (SRT)

SRT 寻址方式仅有能力寻址地址映象中的头 32 个字节，地址从 \$0000 到 \$001F。此寻址方式适用于 CLR、LDA 和 STA 指令。通过放置大部分精确计算的数据在这个存储器范围，系统可以被优化。

由于 5 位地址被嵌在操作码中，仅需有效的 5 位地址包含在指令中；这节省了程序空间和执行时间。在执行期间，CPU 增加高 9 位到 5 位操作数地址，结合成 14 位地址 (\$000x 或 \$001x) 用于访问预订的操作数。

8.3.6 直接寻址方式 (DIR)

DIR 寻址方式用于访问位于直接地址空间 (\$0000 through \$00FF) 的操作数。

在执行期间，CPU 增加高 6 位到跟随操作码的直接地址操作数的低字节。CPU 使用结合的 14 位地址 (\$00xx) 访问预定的操作数。

8.3.7 扩展寻址方式 (EXT)

在扩展寻址方式，操作数的 14 位地址包含在目标代码中，目标代码位于操作码后两个字节的低 14 位。RS08 MCU 系列中此寻址方式仅被用于 JSR 和 JMP 指令用于跳转到目的地。

8.3.8 变址寻址方式 (IX, 被伪指令执行)

变址寻址方式有时也被称为间接寻址方式，原因是变址寄存器被作为访问预定操作数的参考。

变址寻址方式一个重要的特征是在执行期间，操作数地址被计算是基于在存储器映象 \$000F 位置的 X 变址寄存器的内容，而不是在程序汇编时确定的一个常数地址。这允许编程写可以依靠之前程序指令的结果访问不同的操作数地址位置（而不是访问程序被写时确定的地址位置）。

RS08 家族使用地址位于 \$000F 的寄存器 X 作为一个变址和地址位于 \$000E 的 D[X] 寄存器作为变址数据寄存器支持变址寻址方式。通过编程变址寄存器 X，直接页内的任何位置都可以通过变址数据寄存器 D[X] 读 / 写。

通过使用 D[X] 作为操作数，这些伪指令可以被用于所有支持直接、短型和小型寻址方式的指令。

8.4 特殊操作

大部分 CPU 操作可以通过指令集描述，但是有些特殊的操作必须被考虑，例如上电后 CPU 如何开始执行应用程序的。程序开始运行后，当前正常的指令明确 CPU 下一步做什么。两种意外事件会导致 CPU 临时延缓正常程序执行：

- 复位事件强迫 CPU 工作在应用程序之初的前面，强迫从 \$3FFD 地址开始执行。
- 一个主设备系统可以导致 CPU 转到背景模式而不是继续执行应用程序的下条指令。

8.4.1 复位顺序

程序开始于复位事件之后。产生复位事件的事情可以从一个到另一个变化。然而，大部分普通源是：上电复位，外部 RESET 脚，低电压检测复位，COP 看门狗定时溢出，非法操作码检测，非法地址访问。关于 MCU 如何辨别复位事件和区分内部和外部导致复位的不同之处，更多的信息参考 复位和中断 章。

复位事件强迫 MCU 立即停止它正在做的工作，开始响应复位。在执行中的任何指令被立即异常中断，不会完成任何剩余的时钟周期。一各短的工作次序是判定复位源是内部还是外部并记录导致复位的原因。在这残余时间，复位源保持工作状态，内部时钟停止以节省功耗。在复位事件后期，时钟恢复，CPU 从复位状态退出。在复位后，程序计数器复位指向地址 \$3FFD，一条指令被取出开始执行。

复位后从 \$3FFD 开始执行代码。注意不要强迫 BKDG 管脚在复位后被拉低，这将导致进入背景模式，CPU 将等待从背景通讯接口传来的命令。

8.4.2 中断

RS08 的中断机制没有用于中断正常的指令流程；它用于从等待和停止模式唤醒 RS08。在正常工作模式，中断事件必须被 CPU 记录。中断的特性与飞思卡尔的 HC05, HC08 或 HCS08 系列不兼容。

8.4.3 等待和停止模式

执行一条 WAIT 或 STOP 指令进入等待或停止模式。在这些模式，CPU 的时钟被关闭以节省功耗且 CPU 停止工作。CPU 保持低功耗状态。直到一个中断或复位事件唤醒。关于等待和停止对其它外设的影响请参考 复位和中断章

8.4.4 工作背景模式

工作背景模式涉及到 CPU 已经停止执行用户程序指令并等待背景调试系统的串行命令。工作背景模式更详细信息参考 开发支持章。

8.5 指令表摘要

指令集摘要术语

列出的术语通过 Table 8-2 用于 Table 8-1 中的指令描述。

操作数

- () = 圆括号内显示的是寄存器和存储单元内容
- ← = 被载入 (读: “得到”)
- ↔ = 与... 交换
- & = 布尔与
- | = 布尔或
- ⊕ = 布尔异或
- :
- +

CPU 寄存器

- A = 累加器
- CCR = 条件代码寄存器
- PC = 程序计数器
- PCH = 程序计数器, 更高顺序 (最高有效) 六位
- PCL = 程序计数器, 更低顺序 (最低有效) 八位
- SPC = 影子程序计数器

SPCH = 影子程序计数器, 更高顺序 (最高有效) 六位
 SPCL = 影子程序计数器, 更低顺序 (最低有效) 八位

存储器和寻址方式

M = A 存储器地址或绝对数据, 依靠寻址方式
 rel = 相对偏移量, 它是存储在相对一条分枝指令的机器代码最好字节内的两个补数。
 X = 伪变址寄存器, 存储单元 \$000F
 X or D[X] = 存储单元 \$000E 指向被伪变址寄存器 (位于 \$000F) 定义的存储单元

条件代码寄存器 (CCR) 位

Z = 零标志器
 C = 进位 / 借位

CCR 行为符号

- = 位不受影响
 0 = 位强迫为 0
 1 = 位强迫为 1
 P = 依据操作结果, 位置位或清零
 U = 操作后未定义

机器代码符号

dd = 一个直接地址 \$0000-\$00FF (高字节假定是 \$00) 的低顺序八位
 ii = 一个立即数字节
 hh = 前缀 2 位为 0 的 14 位扩展地址的高顺序 6 位
 ll = 14 位扩展地址的低顺序 8 位
 rr = 相对偏移量

源表格

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

n — 任何标签或表达式用于表示一个在 0-7 范围内的单整型
x — 任何标签或表达式用于表示一个在 \$0-\$F 范围内的简单十六进制的整型。
opr8i — 任何标签或表达式用于表示一个 8 位立即数值。
opr4a — 任何标签或表达式用于表示一个小型地址 (4 位数值)。指令对待这 4 位数值作为
 在 16K 字节地址空间 (\$0000-\$000F) 的低四位。这 4 位数值被放入操作码内的
 低四位。
opr5a — 任何标签或表达式用于表示一个短型地址 (5 位数值)。指令对待这 5 位数值作为
 在 16K 字节地址空间 (\$0000-\$001F) 的低五位。这 5 位数值被放入操作码内的
 低五位。
opr8a — 任何标签或表达式用于表示一个 8 位数值。指令对待这 8 位数值作为在 16K 字节地
 址空间 (\$0000-\$00FF) 的低八位。
opr16a — 任何标签或表达式用于表示一个 14 位数值。在 RS08 内核, 头两位总是为 0。指
 令对待这 16 位数值作为在 16K 字节地址空间。

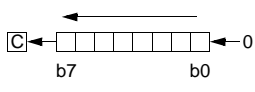
中央处理器单元 (RS08CPUV1)

rel — 在当前指令的目标代码的最后字节之后，任何标签或表达式涉及在 -128 to +127 内来自下个地址的一个地址。汇编器将计算这 8 位有符号偏移量和算入它在这条指令的目标代码内。

寻址方式

- INH = 隐含 (没有操作数)
- IMD = 立即数到直接 (在 MOV 指令内)
- IMM = 立即数
- DD = 直接到直接 (在 MOV 指令内)
- DIR = 直接
- SRT = 短型
- TNY = 小型
- EXT = 扩展
- REL = 8 位相对偏移

Table 8-1. 指令集总结 (Sheet 1 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
ADC #opr8i ADC opr8a ADC ,X <st-super-script>(1) ADC X	Add with Carry	$A \leftarrow (A) + (M) + (C)$ $A \leftarrow (A) + (X) + (C)$	P	P	IMM DIR IX DIR	A9 B9 B9 B9	ii dd 0E 0F	2 3 3 3
ADD #opr8i ADD opr8a ADD opr4a ADD ,X <st-super-script>(1) ADD X	Add without Carry	$A \leftarrow (A) + (M)$	P	P	IMM DIR TNY IX DIR	AB BB 6x 6E 6F	ii dd	2 3 3 3 3
AND #opr8i AND opr8a AND ,X <st-super-script>(1) AND X	Logical AND	$A \leftarrow (A) \& (M)$ $A \leftarrow (A) \& (X)$	P	-	IMM DIR IX DIR	A4 B4 B4 B4	ii dd 0E 0F	2 3 3 3
ASLA<st-super-script>(1)	Arithmetic Shift Left		P	P	INH	48		1
BCC <i>rel</i>	Branch if Carry Bit Clear	$PC \leftarrow (PC) + \$0002 + rel, \text{ if } (C) = 0$	-	-	REL	34	rr	3

1. 这是一条正常 RS08 指令集支持的伪指令。
2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

Table 8-1. 指令集总结 (Sheet 2 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
BCLR <i>n,opr8a</i>	Clear Bit <i>n</i> in Memory	$M_n \leftarrow 0$	-	-	DIR (b0)	11	dd	5
DIR (b1)					13	dd	5	
DIR (b2)					15	dd	5	
DIR (b3)					17	dd	5	
DIR (b4)					19	dd	5	
DIR (b5)					1B	dd	5	
DIR (b6)					1D	dd	5	
DIR (b7)					1F	dd	5	
IX (b0)					11	0E	5	
IX (b1)					13	0E	5	
IX (b2)					15	0E	5	
IX (b3)					17	0E	5	
IX (b4)					19	0E	5	
IX (b5)					1B	0E	5	
IX (b6)					1D	0E	5	
IX (b7)					1F	0E	5	
DIR (b0)					11	0F	5	
DIR (b1)					13	0F	5	
DIR (b2)					15	0F	5	
DIR (b3)					17	0F	5	
DIR (b4)					19	0F	5	
DIR (b5)	1B	0F	5					
DIR (b6)	1D	0F	5					
DIR (b7)	1F	0F	5					
BCS <i>rel</i>	Branch if Carry Bit Set (Same as BLO)	$PC \leftarrow (PC) + \$0002 + rel$, if (C) = 1	-	-	REL	35	rr	3
BEQ <i>rel</i>	Branch if Equal	$PC \leftarrow (PC) + \$0002 + rel$, if (Z) = 1	-	-	REL	37	rr	3
BGND	Background	Enter Background Debug Mode	-	-	INH	BF		5+
BHS <i>rel</i> <st-super-script>(1)	Branch if Higher or Same (Same as BCC)	$PC \leftarrow (PC) + \$0002 + rel$, if (C) = 0	-	-	REL	34	rr	3
BLO <i>rel</i> <st-super-script>(1)	Branch if Lower (Same as BCS)	$PC \leftarrow (PC) + \$0002 + rel$, if (C) = 1	-	-	REL	35	rr	3
BNE <i>rel</i>	Branch if Not Equal	$PC \leftarrow (PC) + \$0002 + rel$, if (Z) = 0	-	-	REL	36	rr	3
BRA <i>rel</i>	Branch Always	$PC \leftarrow (PC) + \$0002 + rel$	-	-	REL	30	rr	3
BRN <i>rel</i> <st-super-script>(1)	Branch Never	$PC \leftarrow (PC) + \$0002$	-	-	REL	30	00	3
BRCLR <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Clear	$PC \leftarrow (PC) + \$0003 + rel$, if (Mn) = 0	-	D	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	5	
IX (b0)					01	0E rr	5	
IX (b1)					03	0E rr	5	
IX (b2)					05	0E rr	5	
IX (b3)					07	0E rr	5	
IX (b4)					09	0E rr	5	
IX (b5)					0B	0E rr	5	
IX (b6)					0D	0E rr	5	
IX (b7)					0F	0E 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. 这是一条正常 RS08 指令集支持的伪指令。

2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

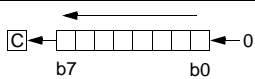
Table 8-1. 指令集总结 (Sheet 3 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
BRSET <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Set	$PC \leftarrow (PC) + \$0003 + rel$, if (Mn) = 1	-	D	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	5	
IX (b0)					00	0E rr	5	
IX (b1)					02	0E rr	5	
IX (b2)					04	0E rr	5	
IX (b3)					06	0E rr	5	
IX (b4)					08	0E rr	5	
IX (b5)					0A	0E rr	5	
IX (b6)					0C	0E rr	5	
IX (b7)					0E	0E rr	5	
DIR (b0)					00	0F rr	5	
DIR (b1)					02	0F rr	5	
DIR (b2)					04	0F rr	5	
DIR (b3)					06	0F rr	5	
DIR (b4)					08	0F rr	5	
DIR (b5)	0A	0F rr	5					
DIR (b6)	0C	0F rr	5					
DIR (b7)	0E	0F rr	5					
BSET <i>n,opr8a</i>	Set Bit <i>n</i> in Memory	$Mn \leftarrow 1$	-	-	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	
IX (b0)					10	0E	5	
IX (b1)					12	0E	5	
IX (b2)					14	0E	5	
IX (b3)					16	0E	5	
IX (b4)					18	0E	5	
IX (b5)					1A	0E	5	
IX (b6)					1C	0E	5	
IX (b7)					1E	0E	5	
DIR (b0)					10	0F	5	
DIR (b1)					12	0F	5	
DIR (b2)					14	0F	5	
DIR (b3)					16	0F	5	
DIR (b4)					18	0F	5	
DIR (b5)	1A	0F	5					
DIR (b6)	1C	0F	5					
DIR (b7)	1E	0F	5					
BSR <i>rel</i>	Branch Subroutine	$PC \leftarrow (PC) + 2$ Push PC to shadow PC $PC \leftarrow (PC) + rel$	-	-	REL	AD	rr	3
CBEQA # <i>opr8i,rel</i> CBEQ <i>opr8a,rel</i> CBEQ <i>X,rel</i> <st-superscript>(1)<st-superscript>(2) CBEQ <i>X,rel</i> <st-superscript>(1)	Compare and Branch if Equal	$PC \leftarrow (PC) + \$0003 + rel$, if (A) - (M) = \$00 $PC \leftarrow (PC) + \$0003 + rel$, if (A) - (M) = \$00 $PC \leftarrow (PC) + \$0003 + rel$, if (A) - (X) = \$00	-	-	IMM DIR IX DIR	41 31 31 31	ii rr dd rr 0E rr 0F rr	4 5 5 5
CLC	Clear Carry Bit	$C \leftarrow 0$	-	0	INH	38		1

1. 这是一条正常 RS08 指令集支持的伪指令。

2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

Table 8-1. 指令集总结 (Sheet 4 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
CLR <i>opr8a</i> CLR <i>opr5a</i> CLR ,X <st-super-script>(1) CLRA CLR X <st-super-script>(1)	Clear	M ← \$00 A ← \$00 X ← \$00	1	-	DIR SRT IX INH INH	3F 8x/9x 8E 4F 8F	dd	3 2 2 1 2
CMP # <i>opr8i</i> CMP <i>opr8a</i> CMP ,X <st-super-script>(1) CMP X <st-super-script>(1)	Compare Accumulator with Memory	(A) - (M) (A) - (X)	P	P	IMM DIR IX INH	A1 B1 B1 B1	ii dd rr OF	2 3 3 3
COMA	Complement (One's Complement)	A ← \overline{A}	P	1	INH	43		1
DBNZ <i>opr8a,rel</i> DBNZ ,X,rel <st-super-script>(1) DBNZA <i>rel</i> DBNZX <i>rel</i> <st-super-script>(1)	Decrement and Branch if Not Zero	A ← (A) - \$01 or M ← (M) - \$01 PC ← (PC) + \$0003 + <i>rel</i> if (result) ≠ 0 for DBNZ direct PC ← (PC) + \$0002 + <i>rel</i> if (result) ≠ 0 for DBNZA X ← (X) - \$01 PC ← (PC) + \$0003 + <i>rel</i> if (result) ≠ 0	-	-	DIR IX INH INH	3B 3B 4B 3B	dd rr 0E rr rr 0F rr	7 7 4 7
DEC <i>opr8a</i> DEC <i>opr4a</i> DEC ,X <st-super-script>(1) DECA DEC X	Decrement	M ← (M) - \$01 A ← (A) - \$01 X ← (X) - \$01	P	-	DIR TNY IX INH DIR	3A 5x 5E 4A 5F	dd	5 4 4 1 4
EOR # <i>opr8i</i> EOR <i>opr8a</i> EOR ,X <st-super-script>(1) EOR X	Exclusive OR Memory with Accumulator	A ← (A ⊕ M) A ← (A ⊕ X)	P	-	IMM DIR IX DIR	A8 B8 B8 B8	ii dd 0E 0F	2 3 3 3
INC <i>opr8a</i> INC <i>opr4a</i> INC ,X <st-super-script>(1) INCA INCX <st-super-script>(1)	Increment	M ← (M) + \$01 A ← (A) + \$01 X ← (X) + \$01	P	-	DIR TNY IX INH INH	3C 2x 2E 4C 2F	dd	5 4 4 1 4
JMP <i>opr16a</i>	Jump	PC ← Effective Address	-	-	EXT	BC	hh ll	4
JSR <i>opr16a</i>	Jump to Subroutine	PC ← (PC) + 3 Push PC to shadow PC PC ← Effective Address	-	-	EXT	BD	hh ll	4
LDA # <i>opr8i</i> LDA <i>opr8a</i> LDA <i>opr5a</i> LDA ,X <st-super-script>(1)	Load Accumulator from Memory	A ← (M)	P	-	IMM DIR SRT IX	A6 B6 Cx/Dx CE	ii dd	2 3 3 3
LDX # <i>opr8i</i> <st-super-script>(1) LDX <i>opr8a</i> <st-super-script>(1) LDX ,X <st-super-script>(1)	Load Index Register from Memory	\$0F ← (M)	P	-	IMD DIR IX	3E 4E 4E	ii 0F dd 0F 0E 0E	4 5 5
LSLA	Logical Shift Left		P	P	INH	48		1

1. 这是一条正常 RS08 指令集支持的伪指令。
2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

Table 8-1. 指令集总结 (Sheet 5 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
LSRA	Logical Shift Right		P	P	INH	44		1
MOV <i>opr8a,opr8a</i> MOV # <i>opr8i,opr8a</i> MOV D[X], <i>opr8a</i> MOV <i>opr8a</i> ,D[X] MOV # <i>opr8i</i> ,D[X]	Move	$(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$	P	-	DD IMD IX/DIR DIR/IX IMM/IX	4E 3E 4E 4E 3E	dd dd ii dd 0E dd dd 0E ii 0E	5 4 5 5 4
NOP	No Operation	None	-	-	INH	AC		1
ORA # <i>opr8i</i> ORA <i>opr8a</i> ORA ,X <st-super-script>(1) ORA X	Inclusive OR Accumulator and Memory	$A \leftarrow (A) (M)$ $A \leftarrow (A) (X)$	P	-	IMM DIR IX DIR	AA BA BA BA	ii dd 0E 0F	2 3 3 3
ROLA	Rotate Left through Carry		P	P	INH	49		1
RORA	Rotate Right through Carry		P	P	INH	46		1
RTS	Return from Subroutine	Pull PC from shadow PC	-	-	INH	BE		3
SBC # <i>opr8i</i> SBC <i>opr8a</i> SBC ,X <st-super-script>(1) SBC X	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$ $A \leftarrow (A) - (X) - (C)$	P	P	IMM DIR IX DIR	A2 B2 B2 B2	ii dd 0E 0F	2 3 3 3
SEC	Set Carry Bit	$C \leftarrow 1$	-	1	INH	39		1
SHA	Swap Shadow PC High with A	$A \leftrightarrow \text{SPCH}$	-	-	INH	45		1
SLA	Swap Shadow PC Low with A	$A \leftrightarrow \text{SPCL}$	-	-	INH	42		1
STA <i>opr8a</i> STA <i>opr5a</i> STA ,X <st-super-script>(1) STA X	Store Accumulator in Memory	$M \leftarrow (A)$	P	-	DIR SRT IX SRT	B7 Ex/ Fx EE EF	dd	3 2 2 2
STX <i>opr8a</i> <st-super-script>(1)	Store Index Register in Memory	$M \leftarrow (X)$	P	-	DIR	4E	0F dd	5
STOP	Put MCU into stop mode		-	-	INH	AE		2+
SUB # <i>opr8i</i> SUB <i>opr8a</i> SUB <i>opr4a</i> SUB ,X <st-super-script>(1) SUB X	Subtract	$A \leftarrow (A) - (M)$ $A \leftarrow (A) - (X)$	P	P	IMM DIR TNY IX DIR	A0 B0 7x 7E 7F	ii dd	2 3 3 3 3
TAX <st-super-script>(1)	Transfer A to X	$X \leftarrow (A)$	P	-	INH	EF		2
TST <i>opr8a</i> <st-super-script>(1) TSTA <st-super-script>(1) TST ,X <st-super-script>(1) TSTX <st-super-script>(1)	Test for Zero	$(M) - \$00$ $(A) - \$00$ $(X) - \$00$	P	-	DD INH IX INH	4E AA 4E 4E	dd dd 00 0E 0E 0F 0F	5 2 5 5

1. 这是一条正常 RS08 指令集支持的伪指令。

2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

Table 8-1. 指令集总结 (Sheet 6 of 6)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
TXA<st-super-script>(1)	Transfer X to A	$A \leftarrow (X)$	D	-	INH	CF		3
WAIT	Put MCU into WAIT mode		-	-	INH	AF		2+

1. 这是一条正常 RS08 指令集支持的伪指令。
2. 在 RS08 内的这条指令不同于 HC08 和 HCS08 的指令，它不能自动增加变址寄存器。

中央处理器单元 (RS08CPUV1)

Table 8-2. Opcode Map

	DIR	DIR	TNY	DIR/REL	INH	TNY	TNY	TNY	SRT	SRT	IMM/INH	DIR/EXT	SRT	SRT	SRT	SRT
HIGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
LOW	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	BRSET0 3 DIR	BSET0 2 DIR	INC 4 TNY	BRA 3 REL		DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	SUB 2 IMM	SUB 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRCLR0 3 DIR	BCLR0 2 DIR	INC 4 TNY	CBEQ 5 DIR	CBEQA 4 IMM	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	CMP 2 IMM	CMP 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRSET1 3 DIR	BSET1 2 DIR	INC 4 TNY		SLA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	SBC 2 IMM	SBC 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRCLR1 3 DIR	BCLR1 2 DIR	INC 4 TNY		COMA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT			LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRSET2 3 DIR	BSET2 2 DIR	INC 4 TNY	BCC 3 REL	LSRA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	AND 2 IMM	AND 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRCLR2 3 DIR	BCLR2 2 DIR	INC 4 TNY	BCS 3 REL	SHA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT			LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRSET3 3 DIR	BSET3 2 DIR	INC 4 TNY	BNE 3 REL	RORA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	LDA 2 IMM	LDA 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRCLR3 3 DIR	BCLR3 2 DIR	INC 4 TNY	BEQ 3 REL		DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT		STA 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRSET4 3 DIR	BSET4 2 DIR	INC 4 TNY	CLC 1 INH	LSLA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	EOR 2 IMM	EOR 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
	BRCLR4 3 DIR	BCLR4 2 DIR	INC 4 TNY	SEC 1 INH	ROLA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	ADC 2 IMM	ADC 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
A	BRSET5 3 DIR	BSET5 2 DIR	INC 4 TNY	DEC 5 DIR	DECA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	ORA 2 IMM	ORA 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
B	BRCLR5 3 DIR	BCLR5 2 DIR	INC 4 TNY	DBNZ 6 DIR	DBNZA 4 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	ADD 2 IMM	ADD 3 DIR	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
C	BRSET6 3 DIR	BSET6 2 DIR	INC 4 TNY	INC 5 DIR	INCA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	NOP 1 INH	JMP 4 EXT	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
D	BRCLR6 3 DIR	BCLR6 2 DIR	INC 4 TNY			DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	BSR 3 REL	JSR 4 EXT	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
E	BRSET7 3 DIR	BSET7 2 DIR	INC 4 TNY	MOV 4 IMD	MOV 5 DD	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	STOP 2+ INH	RTS 3 INH	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT
F	BRCLR7 3 DIR	BCLR7 2 DIR	INC 4 TNY	CLR 3 DIR	CLRA 1 INH	DEC 4 TNY	ADD 3 TNY	SUB 3 TNY	CLR 2 SRT	CLR 2 SRT	WAIT 2+ INH	BGND 5+ INH	LDA 3 SRT	LDA 3 SRT	STA 2 SRT	STA 2 SRT

INH Inherent
IMM Immediate
DIR Direct
EXT Extended
DD Direct-Direct
REL Relative
SRT Short
TNY Tiny
IMD Immediate-Direct

High Byte of Opcode in Hexadecimal

B

Gray box is decoded as illegal instruction

Low Byte of Opcode in Hexadecimal

0

SUB
3 DIR

RS08 Cycles
Opcode Mnemonic
Number of Bytes /
Addressing Mode

Chapter 9 内部时钟源 (RS08ICSV1)

9.1 介绍

内部时钟源 (ICS) 模块给 MCU 提供时钟源选择。模块包含一个锁频环 (FLL) 作为一个时钟源，锁频环能被内部参考时钟控制。模块能提供 FLL 时钟或内部参考时钟作为 MCU 系统时钟源，ICSOUT。

无论是否选择时钟源，ICSOUT 都经过一个总线时钟分频器，它允许得到一个更低终端输出时钟频率。ICSOUT 是总线频率的 2 倍。

Figure 9-1 为标示 ICS 模块的 MC9RS08KA2 系列方框图。

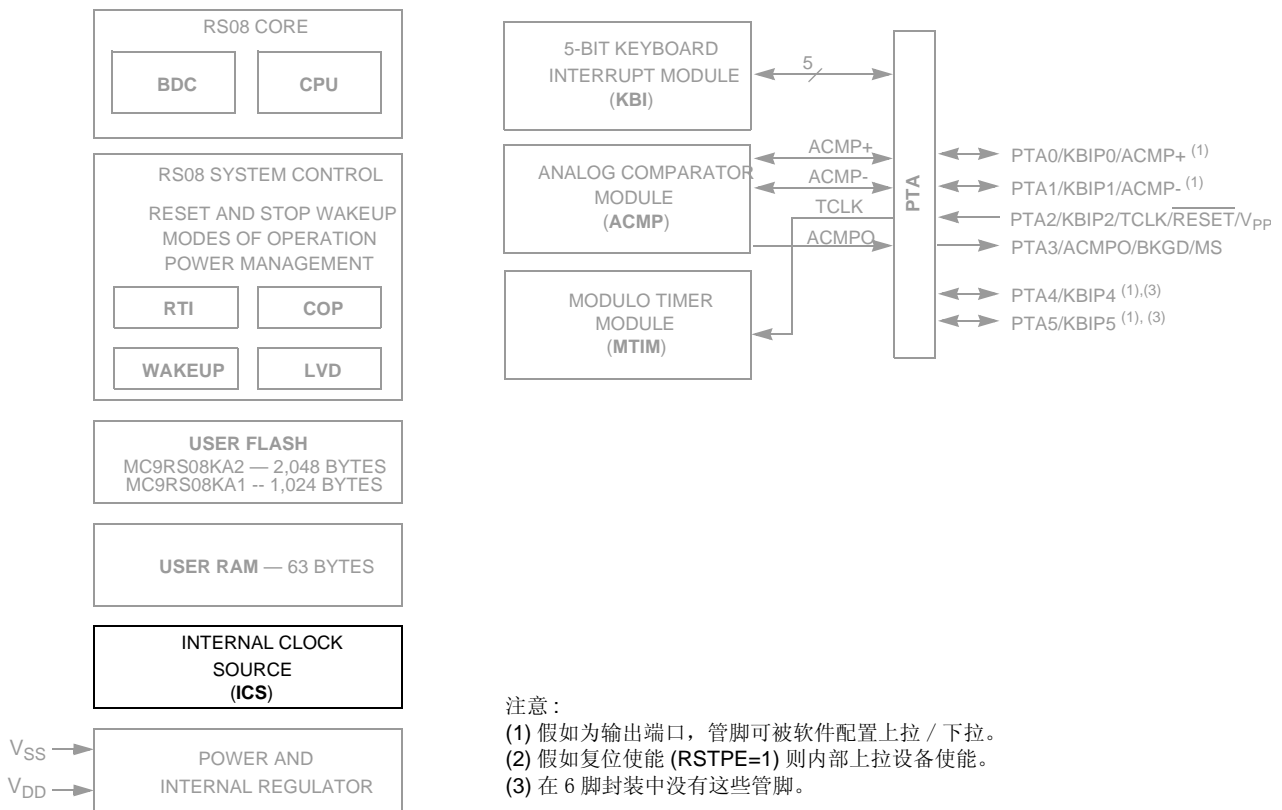


Figure 9-1. MC9RS08KA2 系列高亮 ICS 模块和管脚的框图

9.2 介绍

内部时钟源 (ICS) 模块给 MCU 提供了时钟源选择。对于 MCU，此模块包括一个频率锁频环 (FLL) 作为一个被内部参考时钟控制的时钟源。模块可以提供 FLL 时钟或内部参考时钟作为 MCU 系统时钟源，ICSOUT。

无论哪个时钟源被选用，ICSOUT 经过一个总线时钟分频器 (BDIV)，BDIV 允许得到一个更低的终端输出频率。ICSOUT 是总线时钟频率的 2 倍。

9.2.1 特性

ICS 模块的关键特性：

- 频率锁频环 (FLL) 精确度可调
 - 使用内部 32 kHz 参考时钟可以达到 0.2% 精度
 - 使用内部 32 kHz 参考时钟在全电压和温度范围内 2% 偏差
 - DCO 输出是 512 倍内部参考频率
- 内部参考时钟有 9 位可调整位
- 内部参考时钟可以被选为 MCU 的时钟源
- 无论选用哪个时钟源都可以被分频
 - 提供 2 位可选时钟分频器 (允许被分频伪：1, 2, 4 和 8)
- 复位后自动选用 FLL 使能内部模式

9.2.2 工作模式

ICS 有 4 种工作模式：FEI、FBI、FBILP 和 stop。

9.2.2.1 FLL 使能内部 (FEI)

在 FLL 使能内部模式，它是缺省的模式，ICS 提供一个被内部参考时钟控制的 FLL 分频的时钟

9.2.2.2 FLL 旁路内部 (FBI)

FLL 旁路内部模式，FLL 通过内部参考时钟使能和控制，但是被旁路。ICS 提供一个被内部参考时钟分频的时钟。

9.2.2.3 FLL 旁路内部低功耗 (FBILP)

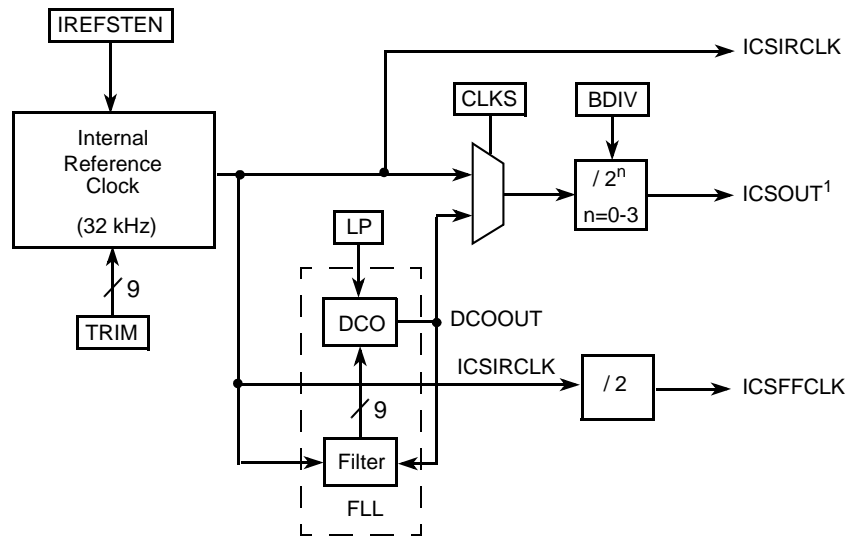
在 FLL 旁路内部低功耗模式，FLL 被禁止和旁路，ICS 提供一个被内部参考时钟分频的时钟。

9.2.2.4 停止 (STOP)

在停止模式，FLL 被禁止且内部参考时钟可选择是否允许或禁止。ICS 不提供 MCU 时钟。

9.2.3 方框图

Figure 9-2 列出了 ICS 方框图。



¹ ICSOUT 是总线频率的 2 倍

Figure 9-2. 内部时钟源 (ICS) 方框图

9.3 外部信号描述

无 ICS 信号连接芯片。

内部时钟源 (RS08ICSV1)

9.4 寄存器定义

Table 9-1 是 ICS 寄存器摘要

Table 9-1. ICS 寄存器摘要

Name		7	6	5	4	3	2	1	0
ICSC1	R	0	CLKS	0	0	0	0	0	IREFSTEN
	W								
ICSC2	R	BDIV		0	0	LP	0	0	0
	W								
ICSTRM	R	TRIM							
	W								
ICSSC	R	0	0	0	0	0	CLKST	0	FTRIM
	W								

9.4.1 ICS 控制寄存器 1 (ICSC1)

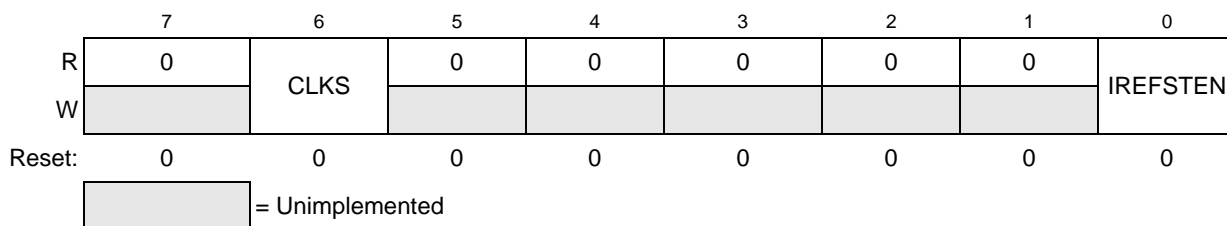


Figure 9-3. ICS 控制寄存器 1 (ICSC1)

Table 9-2. ICSC1 域描述

域	描述
6 CLKS	时钟源选择 — 选择时钟源控制总线频率。实际总线频率依靠 BDIV 位的值。 0 FLL 输出被选择 1 内部参考时钟被选择
0 IREFSTEN	内部参考停止使能 — 当 ICS 进入停止模式，控制内部参考时钟是否保持使能状态 1 在停止模式内部参考时钟保持使能 0 在停止模式内部参考时钟禁止

9.4.2 ICS 控制寄存器 2 (ICSC2)

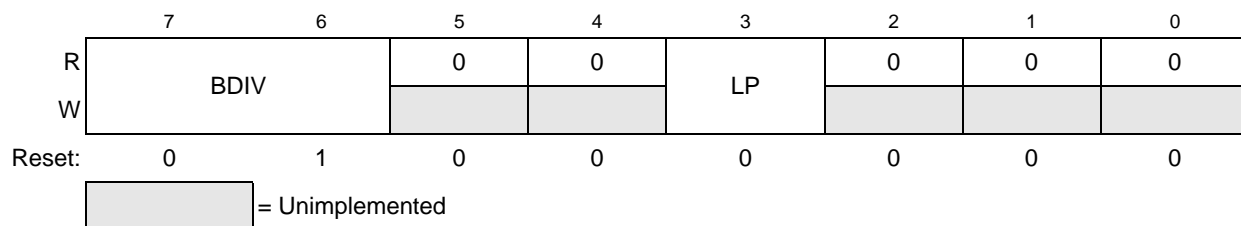


Figure 9-4. ICS 控制寄存器 2 (ICSC2)

Table 9-3. ICSC2 域描述

域	描述
7:6 BDIV	总线频率分频器 — 选择时钟源分频数，时钟源通过 CLKS 位选择。这控制总线频率。 00 编码 0 — 被选时钟被 1 分频 01 编码 1 — 被选时钟被 2 分频 (复位缺省状态) 10 编码 2 — 被选时钟被 4 分频 11 编码 3 — 被选时钟被 8 分频
3 LP	低功耗选择 — 在 FLL 旁路模式，控制 FLL 是否禁止 1 在旁路模式 FLL 被禁止 0 在旁路模式 FLL 不被禁止

9.4.3 ICS 调整寄存器 (ICSTRM)

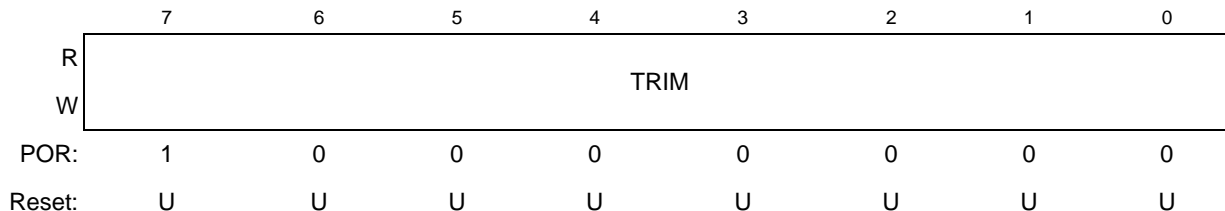


Figure 9-5. ICS 调整寄存器 (ICSTRM)

Table 9-4. ICSTRM 域描述

域	描述
7:0 TRIM	ICS 调整选择 — TRIM 位通过控制内部参考时钟周期控制内部参考时钟频率。这些位的影响是二进位权重 (例如，位 1 将调整是位 0 的 2 倍)。在 TRIM 增加二进位值将增加周期，减小此值将减少周期。 在 ICSSC 寄存器的 FTRIM 是一个附件的极小的调整位。

内部时钟源 (RS08ICSV1)

9.4.4 ICS 状态和控制 (ICSSC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CLKST	0	FTRIM
W								
POR:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	U

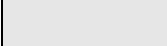
 = Unimplemented

Figure 9-6. ICS 状态和控制寄存器 (ICSSC)

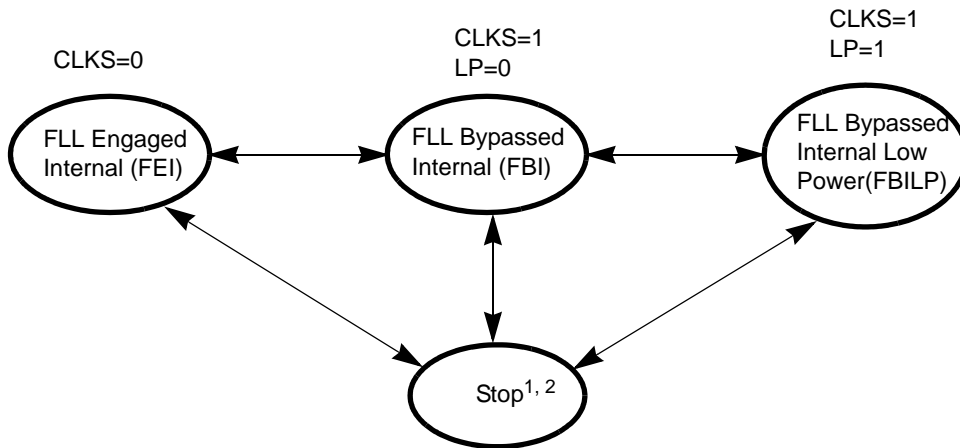
Table 9-5. ICSSC 域描述

域	描述
2 CLKST	时钟模式状态 — CLKST 只读位表明了当前时钟模式。两个时钟域之间对于内部同步写 CLKS 位， CLKST 位不会立即更新。 0 Output of FLL 输出被选择 1 内部参考时钟被选择
0 FTRIM	ICS 微小调整 — FTRIM 位控制内部参考时钟频率最小调整。 FTRIM 置 1 将增加周期， 清零将减小周期。

9.5 功能描述

9.5.1 工作模式

本节 ICS 的状态作为一个状态图表被显示和描述。箭头表明了状态之间的转换。



¹ ICS 当 MCU 进入停止模式 ICS 进入停止状态， FLL 总是被禁止。 MCU 进入停止状态之前 ICS 返回到活动状态， 除非在停止模式产生复位。

² 当 MCU 进入停止模式， 假如 IREFSTEN 被置 1， ICSIRCLK 保持工作状态。

Figure 9-7. 时钟切换模式

9.5.1.1 FLL 使能内部 (FEI)

FLL 使能内部 (FEI) 是任何复位后缺省的工作模式，当 CLKS 被清零进入 FEI 模式。

在 FLL 使能内部模式，ICSOUT 时钟来源于 FLL 时钟，它通过内部参考时钟被控制。The FLL 环将锁住频率到 512 倍滤波频率。

9.5.1.2 FLL 旁路内部 (FBI)

当 CLKS 被置 1 和 LP 位被清 0 进入 FLL 旁路内部 (FBI) 模式。

在 FLL 旁路内部模式，ICSOUT 时钟来源于内部参考时钟。FLL 时钟通过内部参考时钟被控制，且 FLL 环将锁住频率到 512 倍滤波频率。

9.5.1.3 FLL 旁路内部低功耗 (FBILP)

当 CLKS 被置 1 和 LP = 1 进入 FLL 旁路内部低功耗 (FBILP) 模式。

FLL 旁路内部低功耗模式，ICSOUT 时钟来源于内部参考时钟，FLL 被禁止。

9.5.1.4 停止

只有 MCU 停止则进入 ICS 停止模式。在这种模式，所有 ICS 时钟停止，除了 ICSIRCLK。假如 IREFSTEN 被置 1，ICSIRCLK 将保持工作。

当 MCU 从停止模式中断退出，ICS 将回到工作模式，当 MCU 进入停止模式，ICS 工作。假如内部参考在停止模式 (IREFSTEN = 0) 不运行，ICS 将获取一些时间， t_{ir_wu} ，用于内部参考唤醒。假如内部参考在停止模式 (IREFSTEN = 1) 总是运行，进入 FEI 将获取一些时间 t_{fil_wu} ，用于 FLL 返回它早先的精确频率。

9.5.2 模式切换

当从 FBILP 转变到 FEI 或 FBI，或在任何时间调整值被写入，用户应当等待 FLL 获取的时间 $t_{acquire}$ ，FLL 将获得期望得到的频率。

9.5.3 总线频率分频器

BDIV 位能在任何时间被改变，实际切换到新频率将立即得到。

9.5.4 低功耗位使用

低功耗位 (LP) 提供允许 FLL 被禁止并因此保持功耗。然而，在一些应用中，在切换到 FLL 使能模式之前，使能 FLL 和允许它锁在最高精度是可以的。当 LP = 1，FLL 在旁路模式被禁止。

9.5.5 内部参考时钟

ICSIRCLK 频率可以通过调整内部参考时钟周期重新确定。这可以通过写一个新值到 ICSTRM 寄存器的 TRIM 位实现。写一个更大的值将降低 ICSIRCLK 频率，写一个更小的值将增加 ICSIRCLK 频率。假如 ICS 在 FLL 使能内部 (FEI)、FLL 旁路内部 (FBI) 或 FLL 旁路内部低功耗 (FBILP) 模式，TRIM 位将影响 ICSOUT 频率。TRIM 和 FTRIM 值不会应为一个复位而影响。为了使 ICS 在停止模式运行，SPMSC 的 LVDE 和 LVDSE 位必须进入停止模式之前被置 1。

直到 ICSIRCLK 被调整，ICSOUT 频率可能超过最大片内频率和违反片内时钟定时规格 (见芯片概述章)。BDIV 被重新安排一个 2 分频防止总线频率超过最大值。在改变 BDIV 到 1 分频操作之前，用户应该调整设备到一个允许的频率。

9.5.6 固定频率时钟

ICS 提供 ICSFFCLK 输出，当 ICS 在 FEI 模式，它可以用于作为一个附件时钟源到外设例如定时器。在 FBI 或 FBILP 模式，ICSFFCLK 不是一个合法时钟源用于外设。ICSFFCLK 是 ICSRCLK 的二分之一。

Chapter 10 模拟比较器 (RS08ICSV1)

10.1 介绍

模拟比较器模块 (ICS) 提供了用于比较两个模拟输入电压或用于比较一个模拟输入电压与一个内部参考电压的电路。模拟比较器被设计为全电源电压工作范围 (轨对轨工作)。

Figure 9-1 为标示 ICS 的 MC9RS08KA2 系列方框图。

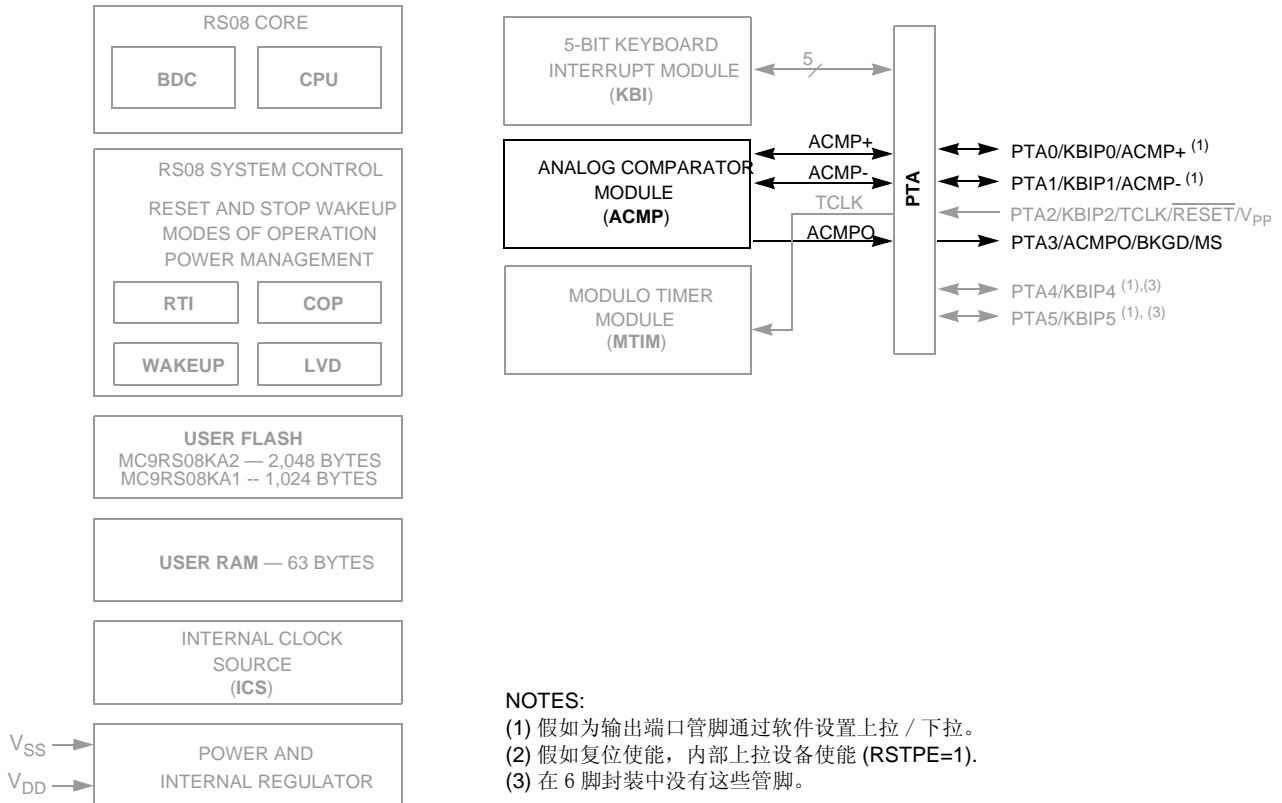


Figure 10-1. MC9RS08KA2 系列方框图中标示 ICS 模块和管脚

模拟比较器 (RS08ICSV1)

10.1.1 特点

ICS 有如下特点:

- 全轨对轨电源范围工作
- 输入偏移量小于 40 mV
- 迟滞小于 15 mV
- 可选择的中断触发条件: 上升沿、下降沿或比较器输出的上升沿或下降沿
- 可选固定的内部待隙参考电压做比较
- 可选允许比较器输出到一个管脚上, ACMPO
- 可工作在停止模式

10.1.2 工作模式

此节详细说明了在等待、停在和背景调试模式下的工作过程。

10.1.2.1 等待模式下工作

假如在执行 WAIT 指令前被使能, 在等待模式 ACMP 将继续工作。因此, 假如 ACMP 中断使能 (ACIE = 1), ACMP 能将 MCU 走出等待模式。为了得到最低的电流功耗, 假如在等待模式不需要作为一个中断源, ACMP 应该被编程禁止。

10.1.2.2 停止模式下工作

假如被使能, ACMP 在停止模式可继续工作, 比较器保持活动状态。假如 ACOPE 被置位, 比较器输出操作在正常工作模式且比较器输出被放置到外部管脚。当比较事件发生和 ACIE 被置位, MCU 退出停止模式; ACF 标志位由此被置位。

假如由于复位而退出停止模式, ACMP 将进入它的复位状态。

10.1.2.3 在工作背景模式下工作

当 MCU 处于工作背景模式, ACMP 将继续正常工作。

10.1.3 方框图

模拟比较器方框图如 [Figure 7-2](#) 所示:

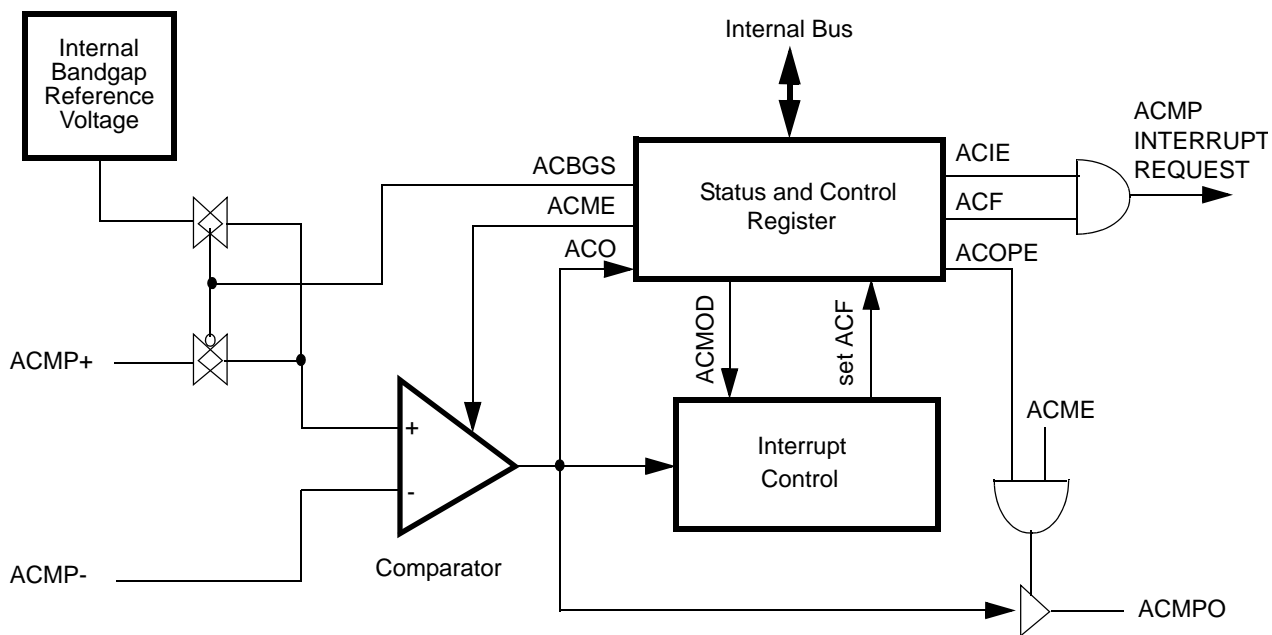


Figure 10-2. 模拟比较器 (ACMP) 方框图

10.2 外部信号描述

ACMP 有两个模拟输入脚，ACMP+ 和 ACMP-，并且有一个数字输出管脚，ACMPO。每个输入管脚能接受一个 MCU 全工作电压范围的输入电压。如 Figure 7-2 所示，ACMP- 管脚连接到比较器的负相输入端；假如 ACBGS=0，ACMP+ 管脚连接到比较器的正相输入端；ACMPO 管脚能被用于驱动一个外部管脚。

ACMP 信号属性如 Table 7-1 所示。

Table 10-1. 信号属性

Signal	Function	I/O
ACMP-	负相模拟输入到 ACMP (负输入)	I
ACMP+	正相模拟输入到 ACMP (正输入)	I
ACMPO	ACMP 的数字输出	O

10.3 寄存器定义

ACMP 包括一个寄存器：

- 一个 8 位状态和控制寄存器

为了了解所有 ACMP 寄存器绝对地址分配，参考此数据手册存储器章直接页寄存器概要。

10.3.1 ACMP 状态和控制寄存器 (ACMPSC)

ACMPSC 包含状标志和控制位用于使能和配置 ACMP。

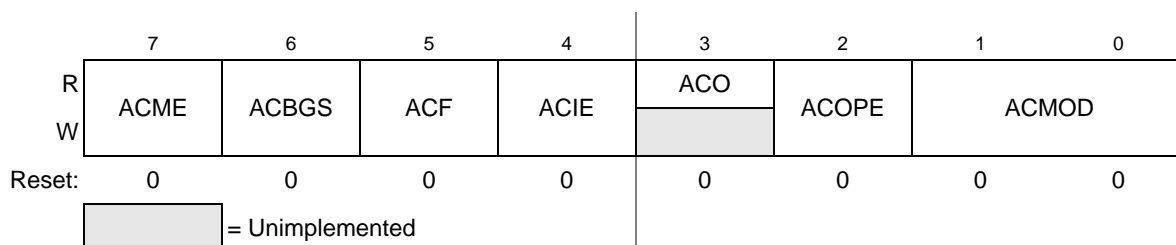


Figure 10-3. ACMP 状态和控制寄存器 (ACMPSC)

Table 10-2. ACMPSC 域描述

域	描述
7 ACME	模拟比较器模块使能 — ACME 使能 ACMP 模块。 0 ACMP 禁止 1 ACMP 使能
6 ACBGS	模拟比较器待隙选择 — ACBGS 用于选择采用内部待隙参考电压还是 ACMP+ 管脚作为模拟比较器正输入 0 外部管脚 ACMP+ 被选择作为模拟器的正输入 1 内部待隙参考电压被选择作为比较器正输入
5 ACF	模拟比较器标志位 — 当一个比较事件发生 ACF 被置 1。比较事件被 ACMOD 定义。通过写 1 到 ACF, ACF 被清零。 0 比较事件没有发生 1 比较事件发生
4 ACIE	模拟比较器中断使能 — ACIE 使能 ACMP 中断。当 ACIE 置 1, 当 ACF 为 1 时, 一个中断将发生。 0 中断禁止 1 中断使能
3 ACO	模拟比较器输出 — 读 ACO 将返回模拟比较器输出的当前值。当 ACMP 禁止 (ACME = 0), 重新为 0 且读将返回 0。
2 ACOPE	模拟比较器输出管脚使能 — ACOPE 使能比较器输出到外部管脚, ACMPO。假如 ACMP 是活动的 (ACME=1), ACOPE 将只控制此管脚。 0 在 ACMPO 管脚, 模拟输出不允许; 1 模拟输出被驱动在 ACMPO。
1:0 ACMOD	模拟比较器模式 — ACMOD 选择导致 ACF 置位的比较事件的类型 00 编码 0 — 比较器输出下降沿 01 编码 1 — 比较器输出上升沿 10 编码 2 — 比较器输出下降沿 11 编码 3 — 比较器输出上升或下降沿

10.4 功能描述

模拟比较器能用于比较应用在 ACMP+ 和 ACMP- 的两个模拟输入电压; 或用于比较一个应用在 ACMP- 的模拟输入电压和一个内部待隙参考电压。ACBGS 用于采用选择待隙参考电压还是 ACMP+ 管脚作为模拟比较器的正相输入。

The 当正相输入大于反相输入时模拟比较器输出为高, 当正相输入小于反相输入时比较器输出为低。ACMOD 用于选择将导致 ACF 被置位的条件。ACF 可以在比较器输出的一个上升沿、下降沿, 或一个上升沿或下降沿 (绑在一起) 被置位。比较器输出能通过 ACO 直接读出。设置 ACOPE, 比较器输出也能被用于驱动 ACMPO 管脚。

模拟比较器 (RS08ICSV1)

注意:

比较器输入是高阻态模拟管脚，对噪音敏感。受噪音干扰的 VDD 与 / 或模拟输入的临近绑定的管脚可能导致比较器偏移 / 迟滞性能超过指定的值。Table 11-7 指明了限制的最大源阻抗。为了实现最佳性能，推荐芯片进入等待 / 停止模式时必须避免 ACMP 测量与临近管脚绑在一起。

Chapter 11

模定时器 (RS08ICSV1)

11.1 介绍

MTIM 是一个简单的 8 位定时器，具有几个软件可选时钟源和一个可编程中断。

MTIM 核心部分是一个 8 位计数器，它能作为一个自由运行的计数器或一个模计数器工作。一个定时中断可以被使能产生周期行中断，用于时间基准的软件循环。

TCLK 输入连接到 MC9RS08KA2 的 PTA2 管脚。XCLK 输入连接到被两分频的 ICSFFCLK 时钟，ICSFFCLK 是来自 ICS 模块的固定频率的内部参考时钟。

Figure 9-1 列出了 MC9RS08KA2 方框图中高亮部分为 ICS 模块。

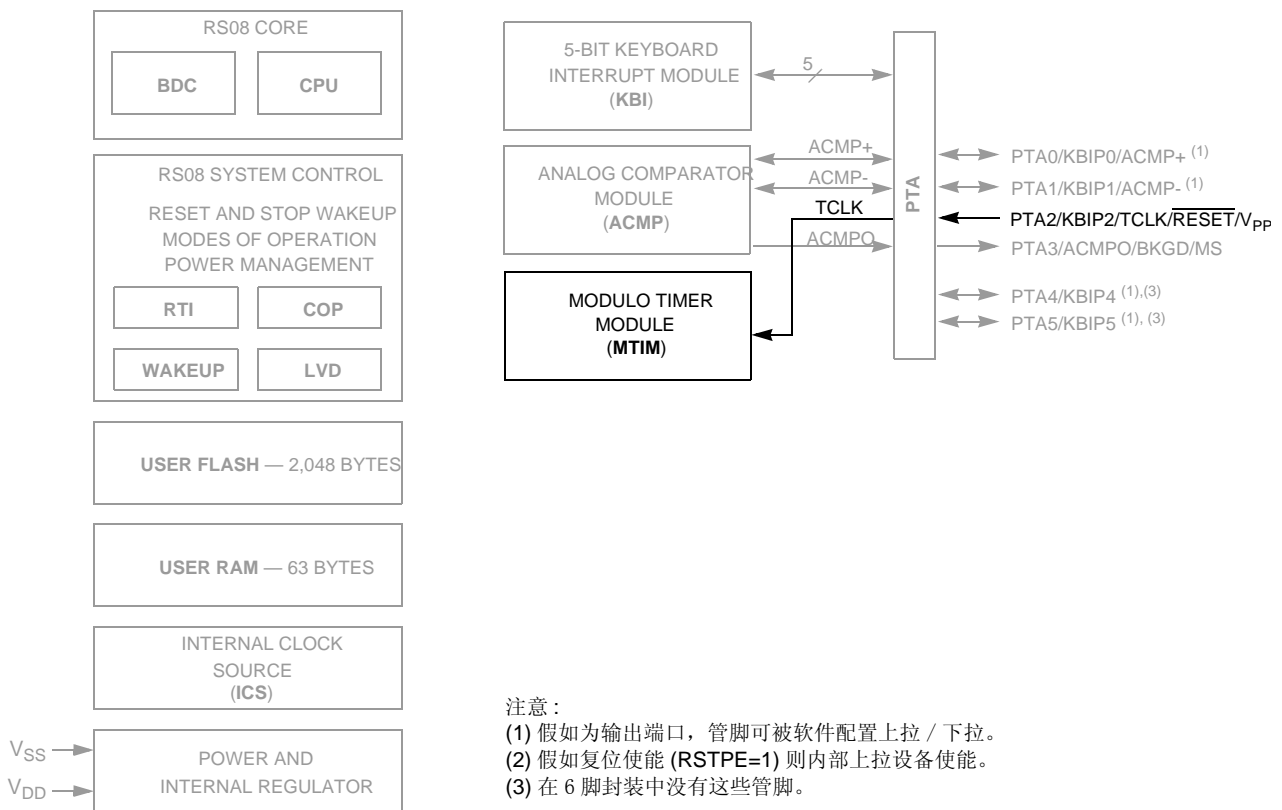


Figure 11-1. MC9RS08KA2 方框图中高亮的 ICS 模块和管脚

11.1.1 特点

定时器系统特点包括:

- 8 为增计数器
 - 自由运行或 8 为模限制
 - 软件可控中断溢出
 - 计数器复位位 (TRST)
 - 计数器停止位 (TSTP)
- 四个可编程选择时钟源用于输入预分频:
 - 系统总线时钟—上升沿
 - 固定频率时钟 (XCLK) —上升沿
 - TCLK 管脚上的外部时钟源—上升沿
 - TCLK 管脚上的外部时钟源—下降沿
- 九个可选择的时钟预分频值:
 - 时钟源被分频为 1, 2, 4, 8, 16, 32, 64, 128, 或 256

11.1.2 工作模式

T 本节详细介绍了在停止、等待和背景调试模式下 MTIM 的工作情况。

11.1.2.1 等待模式下工作

假如执行 WAIT 指令之前被使能, MTIM 在等待模式继续工作。因此, 假如定时器溢出中断使能, MTIM 能用于将 MCU 退出等待模式。为了获得最低电流功耗, 假如在等待模式不需要作为中断源, MTIM 应该被软件禁止。

11.1.2.2 停止模式下工作

不管在执行 STOP 指令之前如何设置, MTIM 在所有停止模式都将禁止。因此, MTIM 不能用于将 MCU 从停止模式唤醒。

假如复位退出停止模式, MTIM 将进入复位时的状态。假如是中断退出停止模式, MTIM 继续保持进入停止模式前的状态。假如计数器进入停止状态时工作, 计数器将从当前的值继续开始。

11.1.2.3 在主动背景模式下工作

MTIM 延迟所有计数直到 MCU 恢复正常用户工作模式。在一个 MTIM 复位没有产生时, 恢复延迟时的值继续计数 (TRST 被写 1 或任何值被写入 MTIMMOD 寄存器)。

11.1.3 方框图

模定时器模块方框图如 Figure 11-2 所示。

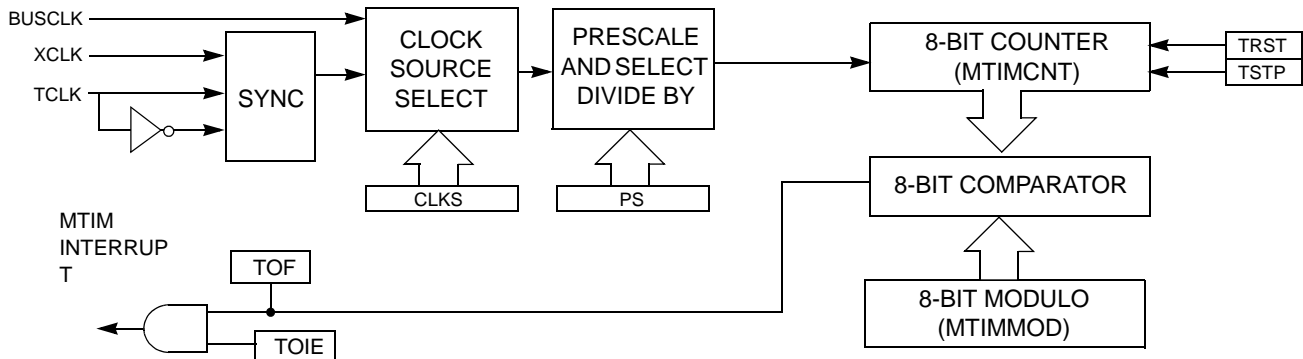


Figure 11-2. 模定时器 (MTIM) 方框图

11.2 外部信号描述

MTIM 包含一个外部信号，TCLK，当选为 MTIM 时钟源时用于输入一个外部时钟。TCLK 信号属性如 Table 7-1 所示。

Table 11-1. 信号属性 Signal

信号	功能	I/O
TCLK	外部时钟源输入到 MTIM	输入

TCLK 输入必须预总线时钟同步。同样，占空比和时钟离散性变化必须适应。因此，TCLK 信号必须限制在总线频率的四分之一。

TCLK 管脚可以预普通端口管脚复用。系管脚位置和功能优先级见 Pins and Connections 章。

11.3 寄存器定义

每个 MTIM 包含四个寄存器，概括在 Table 11-2:

- 一个 8 位状态和控制寄存器
- 一个 8 位时钟配置寄存器
- 一个 8 位计数寄存器
- 一个 8 位模寄存器

对于所有 MTIM 寄存器的绝对地址分配，参考本数据手册的直接页寄存器概要。本节仅通过它们的名字参考寄存器和控制位。

Table 11-2. MTIM 寄存器概要

Name		7	6	5	4	3	2	1	0
MTIMSC	R	TOF	TOIE	0	TSTP	0	0	0	0
	W			TRST					
MTIMCLK	R	0	0	CLKS		PS			
	W								
MTIMCNT	R	COUNT							
	W								
MTIMMOD	R	MOD							
	W								

11.3.1 MTIM 状态和控制寄存器 (MTIMSC)

MTIMSC 包含溢出状态标志位和控制位，用于配置中断使能，重新计数和停止计数。

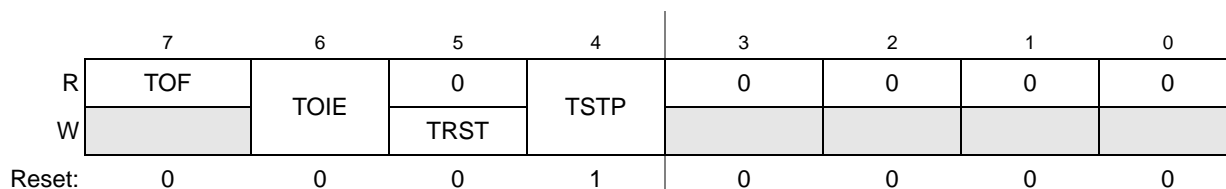


Figure 11-3. MTIM 状态和控制寄存器 (MTIMSC)

Table 11-3. MTIMSC 域描述

域	描述
7 TOF	MTIM 溢出标志位 —当 MTIM 计数寄存器达到 MTIM 模寄存器值溢出回到 \$00，此只读位被置位。在 TOF 位是 1 时读 MTIMSC 寄存器 TOF 清除，然后写 0 到 TOF。当 TRST 被写入 1 或 MTIMMOD 寄存器被写入任何值，TOF 也被清零。 0 在 MTIM 模寄存器计数值没有达到溢出值 1 在 MTIM 模寄存器计数值达到溢出值
6 TOIE	MTIM 溢出中断使能 —此读 / 写位使能 MTIM 溢出中断。假如 TOIE 被置位，当 TOF = 1，产生中断。复位 TOIE 清零。Do not set TOIE if 假如 TOF = 1 不要将 TOIE 置位。先将 TOF 清零，然后将 TOIE 置位。 0 TOF 中断禁止，使用软件巡检。 1 TOF 中断允许
5 TRST	MTIM 计数器复位 —当这个只写位写入 1，MTIM 计数寄存器复位到 \$00 且 TOF 被清零。读此位总是返回 0。 0 无影响。MTIM 计数器保持当前状态 1 MTIM 计数器被复位为 \$00。
4 TSTP	MTIM 计数器停止 —When 置位，此读 / 写位停止 MTIM 计数保持在当前值。当 TSTP 被清零从当前值恢复计数。MCU 复位后将 TSTP 置位防止 MTIM 计数。 0 MTIM 计数器在工作状态 1 MTIM c 计数器处于停止状态

11.3.2 MTIM 时钟配置寄存器 (MTIMCLK)

MTIMCLK 包括时钟选择位 (CLKS) 和预分频选择位 (PS)。

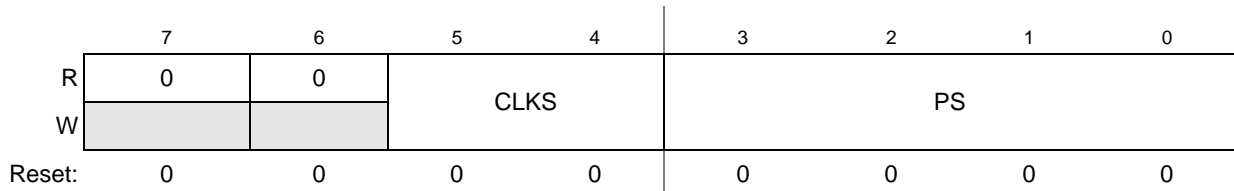


Figure 11-4. MTIM 时钟配置寄存器 (MTIMCLK)

Table 11-4. MTIMCLK 域描述

域	描述
5:4 CLKS	<p>时钟源选择 — 两个读 / 写位选择四个不同时钟源中的一个作为输入到 MTIM 域分频器。在计数器处于工作状态改变时钟源不能将计数器清零。计数继续工作在新的时钟源。复位 CLKS 被清除为 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 管脚), 下降沿</p> <p>11 编码 4 — 外部源 (TCLK 管脚), 上升沿</p>
3:0 PS	<p>时钟源预分频器 — 这四个读 / 写位从 8 位预分频器中选择九个输出当中的一个。在计数器工作状态改变预分频器不会将计数器清零。计数器继续工作在新的预分频值。复位将 PS 清除为 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>所有其它缺省的编码 — MTIM 时钟源 ÷ 256.</p>

11.3.3 MTIM 计数寄存器 (MTIMCNT)

MTIMCNT 是一个 8 位计数器的当前 MTIM 计数的只读值。

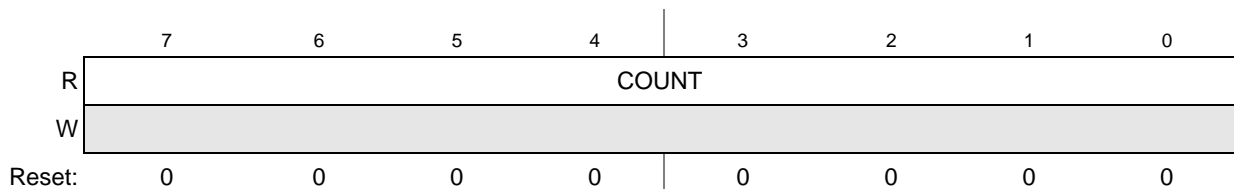


Figure 11-5. MTIM 计数寄存器 (MTIMCNT)

模定时器 (RS08ICSV1)

Table 11-5. MTIMCNT 域描述

域	描述
7:0 COUNT	MTIM 计数 — 这八个只读位包含了 8 位计数器的当前值。写这个寄存器无效。复位清除计数为 \$00。

11.3.4 MTIM 模寄存器 (MTIMMOD)

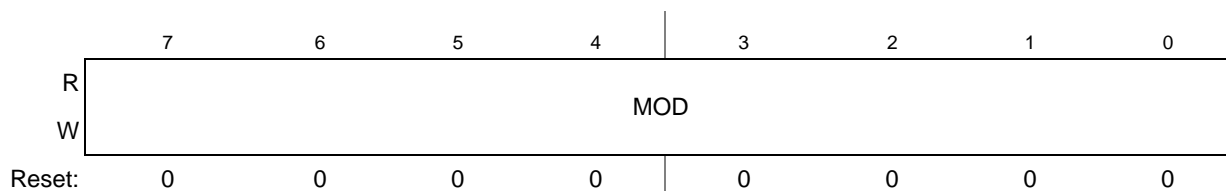


Figure 11-6. MTIM 模寄存器 (MTIMMOD)

Table 11-6. MTIMMOD 描述

域	描述
7:0 MOD	MTIM 模 — 这八个读 / 写位包含模的值用于复位计数和 TOF 置位。一个 \$00 值将 MTIM 置于自由运行模式。写 MTIMMOD 将计数值清除为 \$00 并将 TOF 清零。复位后将模置为 \$00。

11.4 功能描述

MTIM 由一个主要的 8 位增计数器构成，具有一个 8 位模寄存器、一个时钟源选择器和一个具有几个可选值的预分频模块。此模块也包含软件可选中断逻辑。

MTIM 计数器 (MTIMCNT) 由三种工作模式：停止、自由运行和模。复位后计数器停止。假如计数器没有写一个新值到模寄存器开始计数，计数器将处于自由运行模式。当计数器运行时写一个非 \$00 值到模寄存器，计数器处于模运行模式。

任何 MCU 复位，计数器停止且值为 \$00，模寄存器被置为 \$00。缺省情况下选择总线时钟源和预分频值为 1 分频。为了工作在自由运行模式，简单地设置 MTIM 状态和孔子寄存器 (MTIMSC) 和清除 MTIM 停止位 (TSTP)。

四个时钟源可软件选择：内部总线时钟，固定频率时钟 (XCLK)，和一个选择作为上升沿或下降沿增加的在 TCLK 管脚的外部时钟。MTIMCLK 中的 MTIM 时钟选择位 (CLKS) 用于选择想要的时钟源。当一个新时钟源被选择，假如计数器处于工作状态 (TSTP = 0)，计数器将继续根据用新时钟源得到的预分频值计数。

九个预分频值软件可选：时钟源被分频为 1, 2, 4, 8, 16, 32, 64, 128, 或 256。MTIMCLK 中预分频器选择位 (PS) 选择想要的预分频值。当一个新的预分频值被选用，假如计数器正处于工作状态 (TSTP = 0)，计数器将根据使用新预分频值获得的预想的值计数。

MTIM 模寄存器 (MTIMMOD 允许溢出比较值被设置在从 \$01 到 \$FF 之间的任何值。复位清除模值为 \$00，这导致工作在自由允许模式。

当计数器处于活动状态 (TSTP = 0)，计数器在被选择的速率增加直到计数预模值相匹配。当这些值相同时，计数器溢出回到 \$00 且继续计数。只有计数器溢出，MTIM 溢出标志位 (TOF) 就被置位。标志位在从模值到 \$00 转变时置位。在计数器活动期间写 MTIMMOD 将计数器复位到 \$00 和 TOF 清零。

TOF 清零分两步。第一步是在 TOF 置位期间读 MTIMSC 寄存器。第二步是写 0 到 TOF。假如在这两步之间其它溢出产生，清零过程被复位，第二步执行后 TOF 将保持位 1。这将防止第二个溢出产生被错过。当一个 1 被写入 TRST 或当任何值被写入 MTIMMOD 寄存器，TOF 也被清零。

无论 TOF 是否被置位，MTIM 会考虑一个可选的中断产生。为了使能 MTIM 溢出中断，MTIMSC 中的 MTIM 溢出中断使能位 (TOIE) 置 1。在 TOF = 1 期间 TOIE 不能被写 1，相反地，TOF 首先被清零，然后 TOIE 才能被置 1。

模定时器 (RS08ICSV1)

11.4.1 MTIM 工作例子

此节展示了 MTIM 作为计数器工作达到与模寄存器相匹配的值的例子。

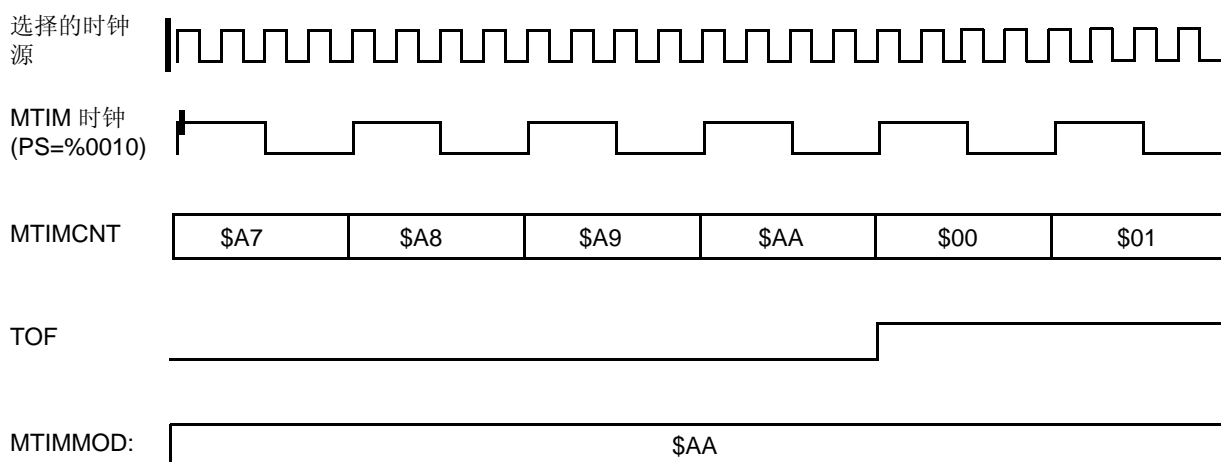


Figure 11-7. MTIM 计数器溢出例子

在 Figure 11-7 例子中，选择的时钟源是四种可选当中的任何一种。预分频器被设置为 PS = %0010 或 4 分频。MTIMMOD 寄存器中的模值被设置为 \$AA。当计数器，MTIMCNT，到达 \$AA 模值时，计数器溢出回到 \$00 且继续计数。定时器溢出标志位，TOF，当计数器值从 \$AA 到 \$00 转变时被置位。假如 TOIE = 1，当 TOF 被置位，产生一个 MTIM 溢出中断。

Chapter 12

开发支持

12.1 介绍

RS08 系列中开发支持系统包括 RS08 背景调试控制器 (BDC)。

BDC 提供一个单线调试接口连接到目标 MCU。此接口为片内 FLASH 和其它非易失性存储器编程提供了一个便捷的方法。同样，BDC 是主要的调试接口，用于开发及和传统调试特点例如允许非入侵式地访问存储器数据 CPU 寄存器修改，断点和单指令跟踪命令。

在 RS08 系列，地址和数据总线信号在外部管脚不开放。调试命令经过单线背景调试接口喂给目标 MCU，包括复位设备不需要用复位管脚。

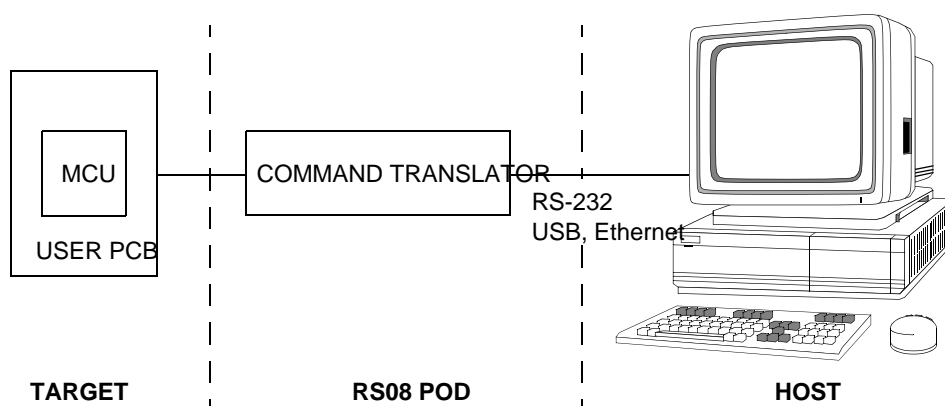


Figure 12-1. 连接 MCU 到主机用于调试

12.2 特点

RS08 背景调试控制器特点包括：

- 使用单线背景调试串行通讯
- 用户存储器资源不能暴力侵入；BDC 寄存器不在存储器映象内
- SYNC 命令决定目标通讯速率
- 在 CPU 运行用户代码非停止时期，非入侵式命令允许访问存储器资源
- 工作背景模式命令用于访问 CPU 寄存器
- GO 和 TRACE1 命令
- BACKGROUND 能将 CPU 从等待或停止模式下唤醒
- BDC_RESET 命令允许主机复位 MCU 不使用复位管脚
- 一个硬件地址断点被嵌入 BDC
- 当 CPU 在停止模式下假如 BDM 使能允许调试，RS08 时钟源运行
- 主动背景模式期间 COP 看门狗暂停

12.3 RS08 背景调试控制器 (BDC)

RS08 系列所有 MCU 包含一个单线背景调试接口支持片内非易失性存储器在线编程和改进的调试性能。与早期 8 位 MCU 调试接口不同，此调试系统提供跟正常应用资源最小限度的冲突。它不用存储器映象中的任何存储器或空间。它需要使用单独的仅输出的 BKGD 管脚。此管脚将与简单用户只输出功能（典型端口，比较器输出，等等）共享，能在正常用户模式中容易调试。

RS08 BDM 命令被分成两部分：

- 主动背景模式命令需要目标 MCU 处于主动背景模式（用户程序不运行）。BACKGROUND 命令导致目标 MCU 进入主动背景模式。主动背景模式命令允许 CPU 寄存器被读或写和允许用户每次跟踪一条 (TRACE1) 用户指令或从主动背景模式转到用户程序。
- 非入侵式命令能在任何时候被执行，甚至在用户程序运行时期。非入侵式命令允许一个用户读或写 MCU 存储器空间或访问在背景调试控制器 (BDC) 内的状态和孔子寄存器。

典型地，一个相对简单的接口设备用于将主机命令转化成单线背景调色系统的客户串行接口命令。依靠开发工具，接口设备可以用于一个标准的 RS-232 串行端口、一个并行打印端口或一些其它典型通讯，如以太网或通用串行口 (USB) 用于主 PC 与接口设备通讯。

Figure 12-2 显示标准 RS08 BDM 设备连接头。此设备是一个小型接口设备用于连接一个主机如个人计算机到一个目标 RS08 系统。BKGD 和 GND 是与一个目标 MCU 通讯的最小连接要求。包含在连接器中的伪开漏型 $\overline{\text{RESET}}$ 信号提供用于主机强迫或监控（假如 $\overline{\text{RESET}}$ 可作为输出）目标系统复位的一个直接的硬件方法。

当在线编程需要时，RS08 BDM 设备提供 V_{PP} 电压给 RS08 MCU。如 Figure 12-2 所示，设备连接出的 V_{PP} 与 $\overline{\text{RESET}}$ 共享。对于 V_{PP} 的需求参见电气附件中的 FLASH 规范。

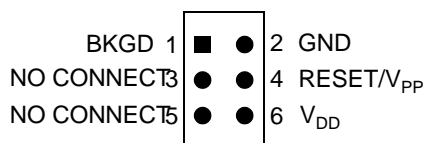


Figure 12-2. 标准 RS08 BDM 工具连接器

背景调试控制器 (BDC) 串行命令使用一个最早在 M68HC12 系列单片机中使用的客户串行协议。此协议需要主机知道通过通讯的时钟速率，此速率由目标 BDC 时钟速率决定。假如主机试图与一个未知 BDC 时钟速率的目标 MCU 通讯，将发送一个 SYNC 命令给目标 MCU 请求一个同步反馈信号，通过这个信号主机能测定正确的通讯速率。

对于 RS08 MCU，BDC 时钟与 MCU 总线时钟相同。更详细的通讯协议描述，参考 [Section 12.3.2, “通讯详述”](#)。

12.3.1 BKGD 管脚描述

BKGD 是一个单线背景调试接口管脚。BKGD 是一个伪开漏极管脚，包含一个片内上拉，因此它无需外部上拉电阻。不同于典型的开漏极管脚，这个受外部影响的管脚的外部电阻电容 (RC) 时间常数，信号上升时间几乎没有作用。客户端协议提供简要的、积极的驱动加速脉冲强迫在管脚上的快速上升时间，确保没有危险的有害的驱动电平冲突。更详细介绍参考 [Section 12.3.2, “通讯详述”](#)。

此管脚的基本功能是背景调试命令和数据的双向串行通讯。在复位期间，此管脚可选择开始工作在主动背景模式或在正常用户模式运行应用程序。此管脚也用于在请求一个时间同步的响应脉冲运行一个主机开发工具确定目标 BDC 的时钟频率。

通过控制 BKGD 管脚和强迫 MCU 复位 (发出一条 BDC_RESET 命令或通过上电复位 (POR))，主机可以强迫目标系统复位进入主动背景模式而不是开始用户应用程序。这对于获得目标 MCU 控制是有用的，MCU 的 FLASH 程序存储器没有被用户应用程序编程。

当调试设备没有与 6 脚 BDM 接口连接器连接时，在 BKGD 的内部上拉确定正常工作模式。

在一些 RS08 单片机，BKGD 管脚具有可选的只输出功能。为了支持 BDM 调试，用户必须禁止这个可选功能。在没有用 BDM 时，此可选功能调试用在正常用户模式。

12.3.2 通讯详述

BDC 串行接口需要主机产生一个下降沿在 BKGD 管脚上显示每一个位时间的开始。无论数据发送还是接收主机都提供这个下降沿。

BDC 串行通讯协议需要主机知道目标 BDC 时钟速度。在 16 个 BDC 时钟周期每位、命令和数据被首先从最高位 (MSB-first) 开始发送。假如 512 个 BDC 时钟周期自主机下降沿间产生，接口超时。当超时产生时 BDC 命令处理被忽略不会影响目标 MCU 系统的存储器或工作模式。

Figure 12-3 展示了一个外部主机发送一个逻辑1或0到目标MCU的BKGD管脚。主机与目标MCU是同步的，因此从主机产生的下降沿到目标感觉到这个位开始时间有一个0到1周期延时。十个目标 BDC 时钟周期后，此对象感觉到在 BKGD 脚的位电平。典型地，在主机到目标传输加速上升沿期间，主机积极驱动伪开漏极 BKGD 管脚。因为在主机到目标期间，目标不驱动 BKGD 管脚，在此期间不需要看作一个开漏极信号。

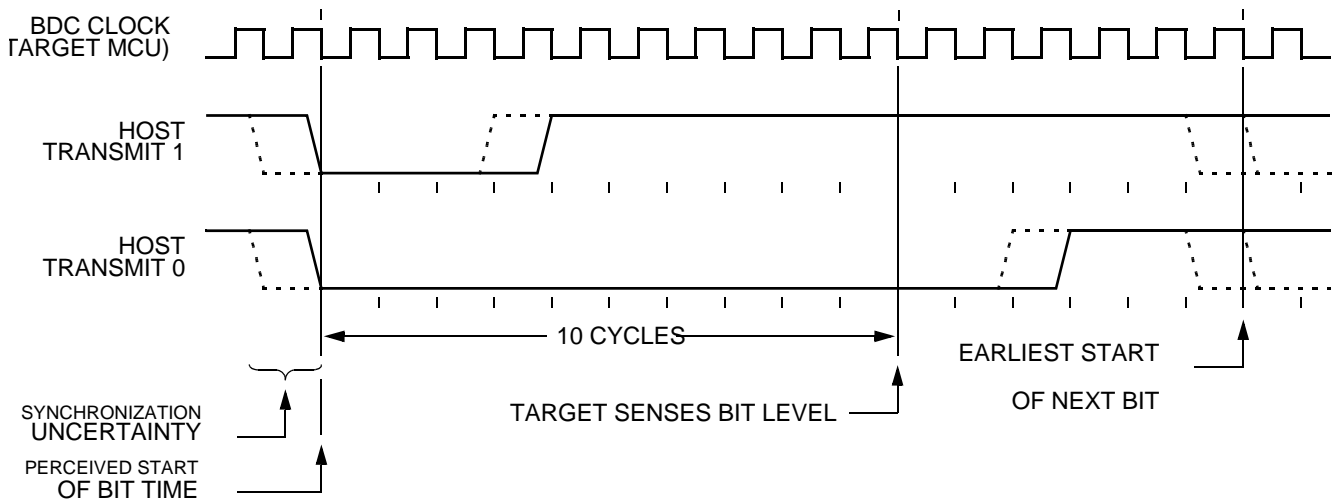


Figure 12-3. BDC 主机—目标串行位控制时间

Figure 12-4. 展示了主机从目标 MCU 接收一个逻辑 1。因为主机与目标是同步的，因此从主机产生的下降沿到目标感觉到这个位开始时间有一个 0 到 1 周期延时。主机钳住 BKGD 管脚低足够长时间让目标认可它 (至少两个 BDC 周期)。在察觉到到位时间开始后，目标驱动积极的高速脉冲七个周期之前主机必须释放低驱动。在位时间开始后，主机应该采样位电平大约 10 个周期。

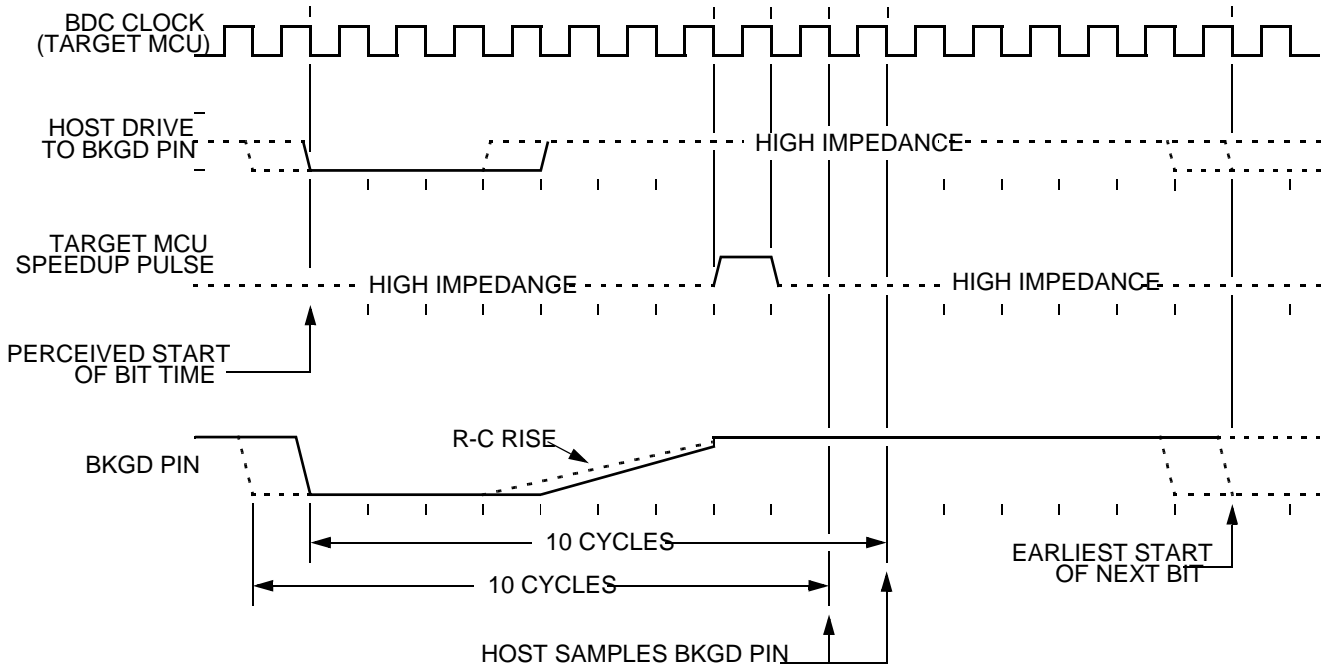


Figure 12-5. BDC 目标—主机串行位控制时间 (逻辑 1)

Figure 12-6 展示了主机从目标 MCU 接收一个逻辑 0。因为主机与目标是同步的，因此从主机在 BKGD 产生的下降沿到目标感觉到这个位开始时间有一个 0 到 1 周期延时。主机初始化位时间但是目标结束它。因为目标想要主机接收一个逻辑 0，它驱动 BKGD 脚位低至 13 个 BDC 时钟周期，然后驱动它为加速上升沿在位时间开始后，主机应该采样位电平大约 10 个周期。

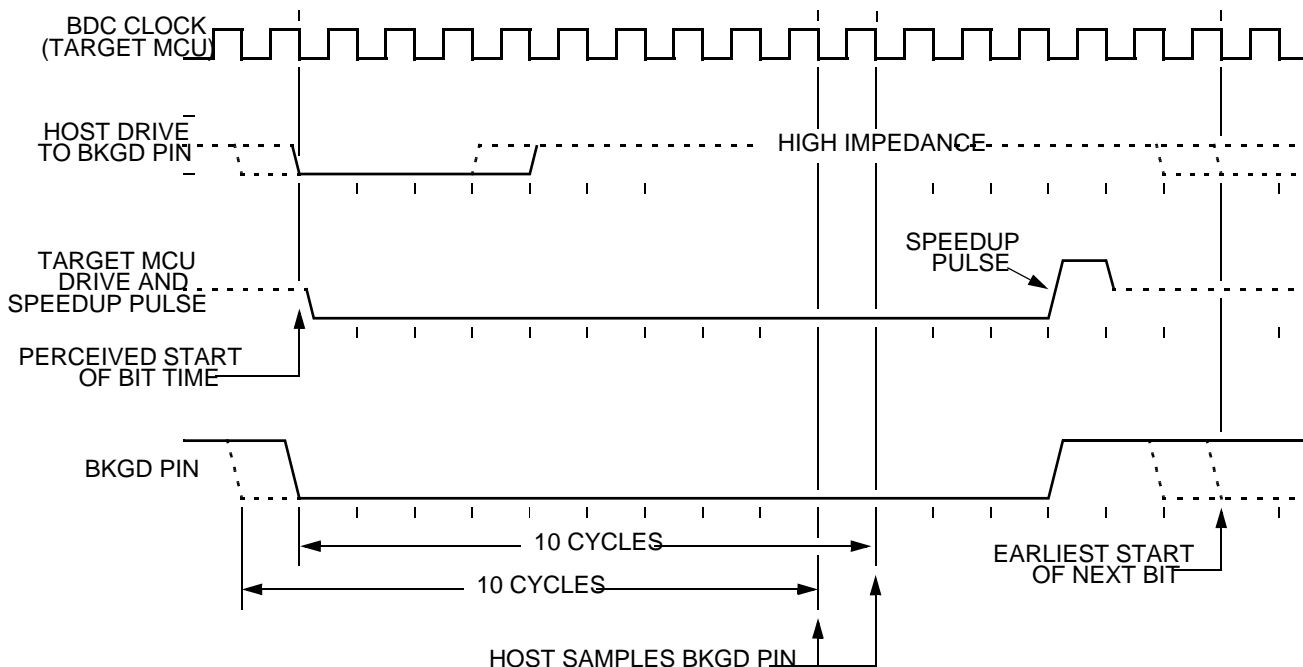


Figure 12-6. BDM 目标—主机串行位控制时间 (逻辑 0)

12.3.3 同步和串行通讯定时溢出

通过产生一个下降沿在 BKGD 管脚，主机发起一个主机到目标串行通讯。假如 BKGD 保持低超过 128 目标时钟周期，目标知道一个 SYNC 命令被发出。这样，目标将保持等待一个上升沿在 BKGD 管脚应答 SYNC 请求的脉冲。假如上升沿没有被检测到，目标将保持等待不会任何时间溢出限制。在一个合法的 SYNC 请求后，当在 BKGD 上的一个上升沿产生，BDC 将驱动 BKGD 管脚为低至 128 个 BDC 周期。

在 128 个周期之前主机返回给 BKGD 逻辑 1。这可以认为是一个合法的位发送，且不是作为一个 SYNC 请求。目标将等待另一个下降沿标记一个新的位开始。假如，无论如何，自从最后的下降沿之后，在 512 时钟周期内一个新的下降沿没有被目标检测到，产生定时溢出和当前的命令被放弃而不会影响 MCU 的存储器或工作模式。这认为给 BDC 的一个软复位。

假如读命令发出但是在 512 连续的时钟周期内数据没有返回，一个软复位将产生，导致命令被忽略。在定时溢出产生后返回的数据无效。软复位也用于结束一条 READ_BLOCK 或 WRITE_BLOCK 命令。

下面描述了实际的位定时要求，用于主机确保逻辑 1 或 0 位传送不会目标定时溢出或认为这个位是一条 SYNC 命令：

- 为了发送一个逻辑 0，BKGD 必须最小在 12 BDC 周期内保持低且直到除了命令串的第一位之外的 511 BDC 周期，命令串将被检测到作为一条 SYNC 命令。
- 为了发送逻辑 1，BKGD 必须被保持低状态至少四个 BDC 周期，在第八个周期被释放，并且保持为高状态至少到第十六个 BDC 周期。
- 后面的位产生，最后的位发送在 512 个 BDC 周期内。

12.4 BDC 寄存器和控制位

BDC 包含两个无需 CPU 可理解的寄存器：

- BDC 状态和控制寄存器 (BDCSCR) 是一个 8 位寄存器，包含控制和状态位用于背景调试控制器。
- BDC 断点寄存器 (BDCBKPT) 掌握一个 16 位断点匹配地址。

这些寄存器可以通过串行 BDC 命令访问，且不位于目标 MCU 的存储器空间内（因此它们没有地址，也不能被用户程序访问）。

在 BDCSCR 内的一些位有写限制；另外的，在任何时间这些寄存器也许可以被读或写。例如，当 MCU 处于主动背景模式的时候，ENBDM 控制位也许不能被写。这防止当 MCU 处于工作背景模式的时候控制位的不明确的条件妨碍背景模式。同样，状态位 (BDMACT, WS, and WSF) 是只读状态显示器且不能写 _ 控制 串行 BDC 命令写。

12.4.1 BDC 状态和控制寄存器 (BDCSCR)

此寄存器能被串行 BDC 命令（读 _ 状态和写 _ 控制）读或写，但是不能被用户程序读或写，这是因为它不位于 MCU 的正常存储器映象中。

	7	6	5	4	3	2	1	0
R		BDMACT			0	WS	WSF	0
W	ENBDM		BKPTEN	FTS				
Normal Reset	0	0	0	0	0	0	0	0
Reset in Active BDM:	1	1	0	0	0	0	0	0

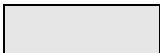
 = Unimplemented or Reserved

Figure 12-7. BDC 状态和控制寄存器 (BDCSCR)

Table 12-1. BDCSCR 寄存器域描述

域	描述
7 ENBDM	使能 BDM (允许主动背景模式) — 典型的, 一个调试开始后或调试主机复位目标, 这位立即被调试主机写 1 并且保持 1 直到一个正常复位清除它。假如应用程序能够进入停止模式, 如果调试需要, 这位必须被置位。 0 BDM 无效 (无侵入命令依然允许)。 1 BDM 有效允许主动背景模式命令。
6 BDMACT	背景模式有效状态 — 这是一个只读状态位。 0 BDM 无效 (用户应用程序正在运行)。 1 BDM 有效, 等待串行命令。
5 BKPTEN	BDC 断点使能 — 假如这位清零, BDC 断点禁止, FTS (force tag select) 控制位和 BDCBKPT 匹配寄存器忽略。 0 BDC 断点禁止 1 BDC 断点使能
4 FTS	强迫 / 标签选择 — 当 FTS = 1, 不管 CPU 地址总线是否匹配 BDCBKPT 匹配寄存器, 一个断点被请求。当 FTS = 0, CPU 地址总线和 BDCBKPT 寄存器匹配导致取出的操作码被标记。假如这个被标记的操作码到达指令列队的终点, CPU 进入主动背景模式而不是执行被标记的操作码。 0 假如 CPU 试图执行那条指令, 在断点地址标记操作码和进入主动背景模式。 1 断点匹配强迫主动背景模式在下条指令边界 (地址不需要一条操作码)。
2 WS	等待或停止状态 — 当目标 CPU 在等待或停止模式, 大部分 BDC 命令不能运行。然而, BACKGROUND 命令能用于强迫目标 CPU 退出等待或停止模式进入主动背景模式, 在这所有 BDC 命令都工作。只有主机强迫目标 MCU 进入主动背景模式, 主机将发出一条读_状态命令去检查在执行其它 BDC 命令前 BDMACT = 1。 0 目标 CPU 正在执行用户应用代码或处于主动背景模式 (当背景变成活动时不处于等待或停止模式)。 1 目标 CP 处于等待或停止模式, 或一条 BACKGROUND 命令用于从等待或停止转变成主动背景摸索。
1 WSF	等待或停止失败状态 — 假如是因为与目标 CPU 执行等待或停止指令的时间相同或之后导致存储器访问命令失败, 这个状态位被置位。通常恢复策略是发出一条 BACKGROUND 命令退出等待或停止模式进入主动背景模式, 重复这条命令导致失败, 将返回到用户程序。(典型地, 主机将恢复 CPU 寄存器和堆栈值且从新执行等待或停止命令)。 0 存储器访问与等待或停止指令没有冲突。 1 存储器访问命令失败, 原因是 CPU 已经进入等待或停止模式。

12.4.2 BDC 断点匹配寄存器

这 16 位寄存器掌握 BDC 内的硬件断点的 14 位地址。在 BDCSCR 内的 BKPTEN 和 FTS 控制位用于使能和配置断点逻辑。串行 BDC 命令 (READ_BKPT 和 WRITE_BKPT) 用于读和写 BDCBKPT 寄存器。在目标 MCU 处于工作背景模式运行用户应用程序之前的时候, 断点被正常设置。然而, 因为 READ_BKPT 和 WRITE_BKPT 是非入侵命令, 它们甚至在用户程序运行时能被执行。关于建立和使用在 BDC 内的硬件断点逻辑的附件说明, 参考“RS08 系列参考手册”。

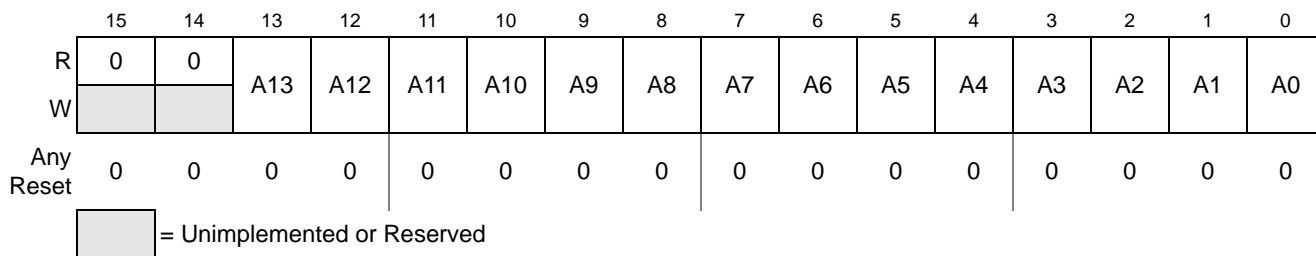


Figure 12-8. BDC 断点匹配寄存器 (BDCBKPT)

12.5 RS08 BDC 命令

BDC 命令从一个主计算机向目标 MCU 的 BKGD 管脚连续发送。所有命令和数据使用客户 BDC 通讯协议首先从 MSB 发送。当非入侵式命令可以在任何时候传送而不管目标 MCU 是否在背景模式或运行一个用户应用程序的时候，主动背景模式命令要求目标 MCU 处于主动背景模式。

Table 12-2 展示了所有 RS08 BDC 命令、它们译码指令的缩写描述和每个命令的意思。

译码指令术语

下面术语被用于 Table 12-2 描述 BDC 命令的译码指令。

	命令首先是主机 - 目标方向的一个 8 位命令代码 (首先最高有效位)
/	= 命令的分开部分
d	= 延迟 16 至 511 目标 BDC 时钟周期
soft-reset	= 来自最后主机下降沿的至少 512 BDC 时钟周期延迟
AAAA	= 主机 - 目标方向的 16 位地址 ¹
RD	= 目标 - 主机方向的读数据的八位
WD	= 主机 - 目标方向的的写地址的八位
RD16	= 目标 - 主机方向的读数据的 16 位
WD16	= 主机 - 目标方向的的写地址的 16 位
SS	= 目标 - 主机方向的 BDCSCR 的内容 (状态)
CC	= 主机 - 目标方向的用于 BDCSCR 的写数据八位 (控制)
RBKP	= 目标 - 主机方向的读数据的 16 位 (来自 BDCBKPT 断点寄存器)
WBKP	= 主机 - 目标方向的写地址的 16 位 (适合于 BDCBKPT 断点寄存器)

1. RS08 CPU 仅用地址 14 位和占用 16 位 AAAA 地址域的低 14 位。在 AAAA 的地址位 15 和 14 的值被删去因此是无用的。

Table 12-2. RS08 BDC 命令总结

命令	主动背景模式 / 无入侵式	译码结构	描述
SYNC	Non-intrusive	n/a ⁽¹⁾	请求一个同步参考脉冲确定目标 BDC 通讯速度
BDC_RESET	Any CPU mode	18 ⁽²⁾	请求一个 MCU 复位
BACKGROUND	Non-intrusive	90/d	假如使能进入主动背景模式 (假如 ENBDM 位等于 0 忽略)
READ_STATUS	Non-intrusive	E4/SS	BDC 从 BDCSCR 读 BDC 状态
WRITE_CONTROL	Non-intrusive	C4/CC	在 BDCSCR 内写 BDC 控制
READ_BYTE	Non-intrusive	E0/AAAA/d/RD	从目标存储器读一个字节
READ_BYTE_WS	Non-intrusive	E1/AAAA/d/SS/RD	读一个直接和报告状态
WRITE_BYTE	Non-intrusive	C0/AAAA/WD/d	写一个字节到目标存储器
WRITE_BYTE_WS	Non-intrusive	C1/AAAA/WD/d/SS	写一个字节和报告状态
READ_BKPT	Non-intrusive	E2/RBKP	读 BDCBKPT 断点寄存器
WRITE_BKPT	Non-intrusive	C2/WBKP	写 BDCBKPT 断点寄存器
GO	Active background mode	08/d	转到执行用户应用程序起始于在 PC 内的正常地址。
TRACE1	Active background mode	10/d	在 PC 内的地址跟踪一条用户指令, 然后返回到主动背景模式
READ_BLOCK	Active background mode	80/AAAA/d/RD ⁽³⁾	从目标存储器起始地址 AAAA 开始读数据块知道一个软件复位被检测到。
WRITE_BLOCK	Active background mode	88/AAAA/WD/d ⁽⁴⁾	从目标存储器起始地址 AAAA 开始写数据块知道一个软件复位被检测到。
READ_A	Active background mode	68/d/RD	读累加器 (A)
WRITE_A	Active background mode	48/WD/d	写累加器 (A)
READ_CCR_PC	Active background mode	6B/d/RD16 ⁽⁵⁾	读 CCR 的位 z、c 于 14 位程序计数器 (PC) 连接, RD16=zc:PC
WRITE_CCR_PC	Active background mode	4B/WD16/d ⁽⁶⁾	写 CCR 的位 z、c 于 14 位程序计数器 (PC) 连接, WD16=zc:PC
READ_SPC	Active background mode	6F/d/RD16 ⁽⁷⁾	读 14 位影子程序计数器 (SPC) RD16=0:0:SPC
WRITE_SPC	Active background mode	4F/WD16/d ⁽⁸⁾	写 14 位影子程序计数器 (SPC) WD16 = x:x:SPC,“x” 显示的两个最高位目标程序忽略不计。

1. SYNC 命令是一个特别的操作它没有命令代码;

2. 18 是 HCS08 BDC 命令用于 TAGGO。

3. 每个 RD 在主机读数据字节和下一个读之间需要一个延时, 当目标检测到一个软复位时命令结束。

4. 每个 WD 在主机读数据字节和下一个读之间需要一个延时，当目标检测到一个软复位时命令结束。
5. HCS08 BDC 有分离的 READ_CCR 和 READ_PC 命令，RS08 BDC 结合了这命令。
6. HCS08 BDC 有分离 WRITE_CCR 和 WRITE_PC 命令，RS08 BDC 结合了这命令。
7. 6F 是适用于 HCS08 BDC 的 READ_SP ()。
8. 4F 是适用于 HCS08 BDC 的 WRITE_SP (写堆栈指针)。

Appendix A 电气特性

A.1 介绍

本章包括电气和时间特性。

A.2 最大绝对额定值

最大绝对额定值是强制的额定值，且在最大绝对值下运行功能是没有保证的。在 Table A-1 中，强制在指定限定值之外工作会影响芯片的可靠性或对芯片产生永久性地伤害。对于功能的工作条件，参考本章的表格。

此芯片包含了对高静态电压或电场的保护电路。然而，它考虑到了正常的防御用于避免任何高于最大额定值电压用到高阻抗电路的应用。假如未知的输入被约束在一个适当的逻辑电压电平（例如， V_{SS} 或 V_{DD} ）或相关的管脚的可编程上拉电阻使能，那么可以增强工作的可靠性。

Table A-1. 最大绝对额定值

参数	符号	值	单位
工作电压	V_{DD}	-0.3 to +5.8	V
流入 V_{DD} 的最大电流	I_{DD}	120	mA
数字输入电压	V_{In}	-0.3 to $V_{DD} + 0.3$	V
瞬间最大电流 单管脚限定 (应用到所有端口管脚) ^{(1), (2), (3)}	I_D	± 25	mA
温度范围	T_{stg}	-55 to 150	°C

1. 输入电流必须限制在指定的值内。为了明确需要的限流电阻的值，对应正 (V_{DD}) 和负 (V_{SS}) 夹钳电压计算阻抗值，取两个阻抗值中更大的。
2. 所有无电压供电的功能管脚被内部夹钳至 V_{SS} 和 V_{DD} ， \overline{RESET}/V_{PP} 管脚除外，它只被内部夹钳至 V_{SS} 。
3. 在瞬态和最大工作电流条件期间电源必须保持调整在工作 V_{DD} 范围内。假如正注入电流 ($V_{In} > V_{DD}$) 大于 I_{DD} ，注入电流也许对于 V_{DD} 溢出，将导致外部电源停止调整。确保外部 V_{DD} 负载将分流大于最大注入电流。当 MCU 没有消耗电能时这将是更大的风险。例如：假如系统时钟停止或假如时钟速率非常低减小总功率功耗。

A.3 热特性

本节提供了工作温度范围、功率消耗和封装热阻的信息。在 I/O 管脚的功率消耗与片内逻辑和电压调整器电路和被 MCU 设计控制的用户电路的功率消耗相比通常很小。为了获得 $P_{I/O}$ 功耗的计算，明确实际管脚电压和 V_{SS} 或 V_{DD} 以及和每个 I/O 管脚的电流乘积的不同之处。除了特别的大管脚电流（大负载），管脚电压和 V_{SS} 或 V_{DD} 的不同之处非常小。

Table A-2. 热特性

参数	符号	值	单位
工作温度范围 (封装)	T_A	T_L to T_H -40 to 85	°C
最大 j 接合处温度	T_{JMAX}	105	°C
热阻(1).(2).(3).(4)			
6-pin DFN	θ_{JA}	1s 2s2p	°C/W
8-pin PDIP		1s 2s2p	
8-pin SOIC		1s 2s2p	
		1s 2s2p	

1. 接合处温度是死区、片内功耗消耗、封装热阻、安装设备 (板) 温度、周边温度、气流、板上其它元件功耗消耗和板热阻的一个功能。
2. 接合到周边自然传送。
3. 1s - 单层板，一层信号层
4. 2s2p - 四层板，2 层信号和 2 层电源层

平均片上结合处温度从下面计算获得：

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

其中：

- T_A = 周边温度， °C
- θ_{JA} = 封装热阻，与周边结合处， °C/W
- $P_D = P_{int} + P_{I/O}$
- $P_{int} = I_{DD} \times V_{DD}$, 瓦 — 片内功率
- $P_{I/O}$ = 在输入和输出管脚上的功耗消耗 — 用户决定

对于大部分应用， $P_{I/O} \ll P_{int}$ 且可以被忽略。 P_D 和 T_J (假如 $P_{I/O}$ 被忽略) 之间近似关系是：

$$P_D = K \div (T_J + 273^{\circ}\text{C}) \quad \text{Eqn. A-2}$$

使用 Equation A-1 和 Equation A-2，得到 K 为：

$$K = P_D \times (T_A + 273^{\circ}\text{C}) + \theta_{JA} \times (P_D)^2 \quad \text{Eqn. A-3}$$

其中 K 是个常数，与特别的部分有关联。从 Equation A-3 中通过对于一个已知的 T_A 测量 P_D (在平衡状态下) 可以确定 K。使用 K 值，对应任何 T_A 值， P_D 和 T_J 值通过方程式 1 和 2 可以获得。

A.4 静电 (ESD) 保护特性

虽然静电放电对这些芯片造成的伤害远小于对早期的 CMOS 电路，但是正常处理规范需要避免暴露在静电放电之前。鉴定试验被执行用来确保这些芯片能抵挡住暴露在合理的静电电平之前不会受到任何永久性伤害。所有 ESD 测试都符合用于汽车电子等级综合电路的 CDF-AEC-Q00 强制测试资格 (<http://www.aecouncil.com/>)。假如通过 ESD 脉冲后，一片芯片被认定有问题，此芯片不再符合规范要求。通过高温的全部的 DC 参数和功能测试确保每片应用芯片符合常温规格说明，除非在芯片说明中有其它特别说明。

Table A-3. ESD 保护特性

参数	符号	值	单位
对于物体模式的 ESD 对象 (MM) MM 电路描述	V_{THMM}	200	V
对于人体模式的 ESD 对象 (HBM) HBM 电路说明	V_{THHBM}	2000	V

A.5 DC 特性

此节包含电源要求、I/O 管脚特性和在不同工作模式下电源电流的信息。

Table A-4. DC 特性
(温度范围 = -40 to 85°C 周围的)

参数	符号	最小值	典型值	最大值	单位
电源电压 (运行、等待和停止模式) $0 < f_{Bus} < 10\text{MHz}$	V_{DD}	1.8		5.5	V
保持最小 RAM 的应用到 V_{DD} 电源电压	V_{RAM}	0.8 ⁽¹⁾		—	V
低电压检测极限值 (V_{DD} falling) (V_{DD} rising)	V_{LVD}	1.80 1.88	1.86 1.94	1.95 2.03	V
上电复位 (POR) 电压	V_{POR}	0.9	1.4	1.7	V
输入高电压 ($V_{DD} > 2.3\text{V}$) (所有数字输入)	V_{IH}	$0.70 \times V_{DD}$		—	V
输入高电压 ($1.8\text{V} \leq V_{DD} \leq 2.3\text{V}$) (所有数字输入)	V_{IH}	$0.85 \times V_{DD}$		—	V

电气特性

Table A-4. DC 特性 (continued)
(温度范围 = -40 to 85°C 周围的)

参数	符号	最小值	典型值	最大值	单位
输入低电压 ($V_{DD} > 2.3\text{ V}$) (所有数字输入)	V_{IL}	—		$0.30 \times V_{DD}$	V
输入低电压 ($1.8\text{ V} \leq V_{DD} \leq 2.3\text{ V}$) (所有数字输入)	V_{IL}	—		$0.30 \times V_{DD}$	V
输入滞后 (所有数字输入)	V_{hys}	$0.06 \times V_{DD}$		—	V
输入漏电流 (每个管脚) $V_{In} = V_{DD}$ 或 V_{SS} , 所有只输入脚	$ I_{In} $	—	0.025	1.0	μA
高阻抗 (关断状态) 漏电流 (每个管脚) $V_{In} = V_{DD}$ 或 V_{SS} , 所有输入 / 输出	$ I_{OZ} $	—	0.025	1.0	μA
内部上拉 / 下拉电阻 ⁽²⁾ (所有端口管脚)	R_{PU}	20	45	65	$\text{k}\Omega$
输出高电压 (端口 A) $I_{OH} = -5\text{ mA}$ ($V_{DD} \geq 4.5\text{ V}$) $I_{OH} = -3\text{ mA}$ ($V_{DD} \geq 3\text{ V}$) $I_{OH} = -2\text{ mA}$ ($V_{DD} \geq 1.8\text{ V}$)	V_{OH}	$V_{DD} - 0.8$		— — —	V
所有端口管脚的最大总 I_{OH}	$ I_{OHT} $	—		40	mA
输出低电压 (端口 A) $I_{OL} = 5\text{ mA}$ ($V_{DD} \geq 4.5\text{ V}$) $I_{OL} = 3\text{ mA}$ ($V_{DD} \geq 3\text{ V}$) $I_{OL} = 2\text{ mA}$ ($V_{DD} \geq 1.8\text{ V}$)	V_{OL}	— — —		0.8 0.8 0.8	V
所有端口管脚的最大总 I_{OL}	I_{OLT}	—		40	mA
dc 注入电流 ^{(3), (4), (5) (6)} $V_{In} < V_{SS}$, $V_{In} > V_{DD}$ 单一管脚限制 总 MCU 限制 I, 包括所有强驱动管脚	$ I_{IC} $	— —		0.2 0.8	mA mA
输入电容 (所有无电源管脚)	C_{In}	—		7	pF

1. 每片芯片此参数被描述但没有被测试。

2. 对于拉电阻测试条件: $V_{In} = V_{SS}$ 对应于上拉和 $V_{In} = V_{DD}$ 对应于下拉。

3. 所有非电源功能管脚内部钳住至 V_{SS} 和 V_{DD} , $\overline{\text{RESET}}/V_{PP}$ 除外, 它金内部钳住至 V_{SS} 。

4. 输入电流必须限制在指定的值内。为了确定所需的限流电阻的值, 对应正 (V_{DD}) 和负 (V_{SS}) 夹钳电压计算阻抗值, 取两个阻抗值中更大的。

5. 在瞬态和最大工作电流条件期间电源必须保持调整在工作 V_{DD} 范围内。假如正注入电流 ($V_{In} > V_{DD}$) 大于 I_{DD} , 注入电流也许对于 V_{DD} 溢出, 将导致外部电源停止调整。确保外部 V_{DD} 负载将分流大于最大注入电流。当 MCU 没有消耗电能时这将是更大的风险。例如: 假如系统时钟停止或假如时钟速率非常低减小总功率功耗。

6. 每片芯片此参数被描述但没有被测试。

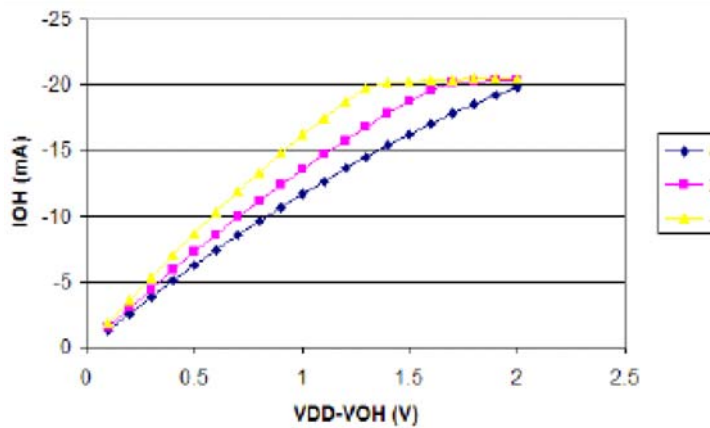


Figure 12-9. Typical IOH vs. VDD-VOH
VDD = 5 V

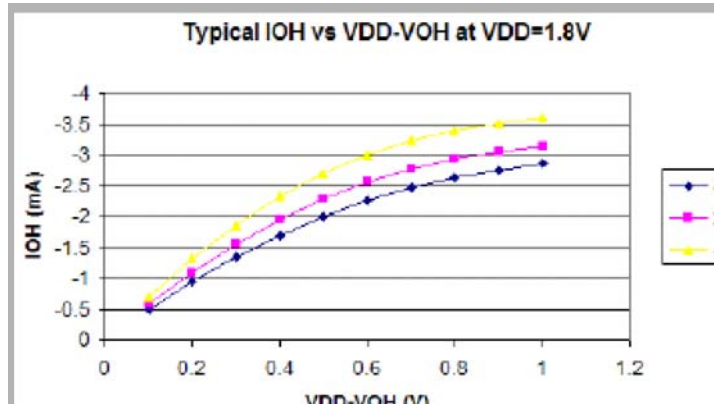


Figure 12-10. Typical IOH vs. VDD-VOH
VDD = 3 V

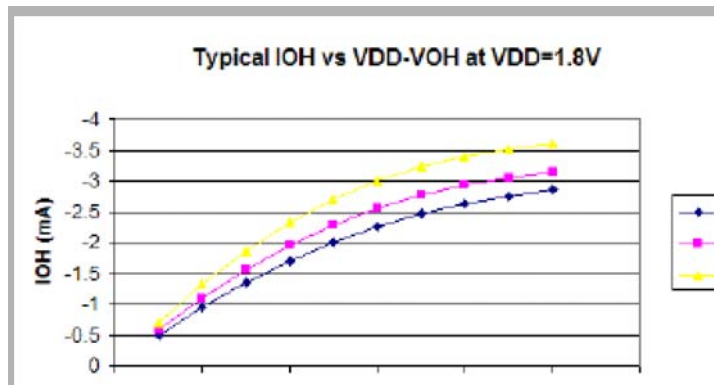


Figure 12-11. Typical IOH vs. VDD-VOH
VDD = 1.8 V

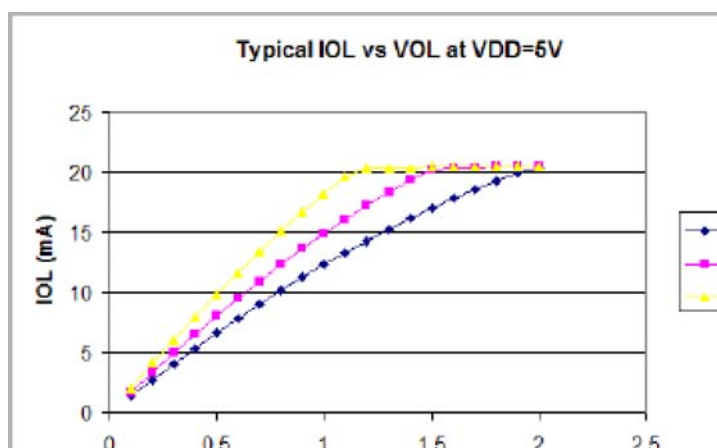


Figure 12-12. Typical I_{OL} vs. V_{OL}
VDD = 5 V

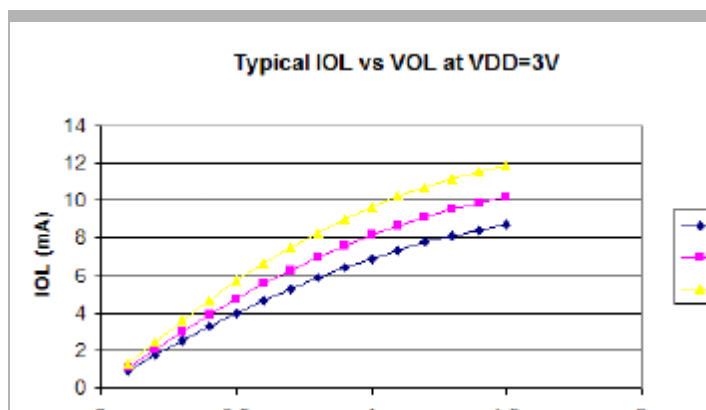


Figure 12-13. Typical I_{OL} vs. V_{OL}
VDD = 3 V

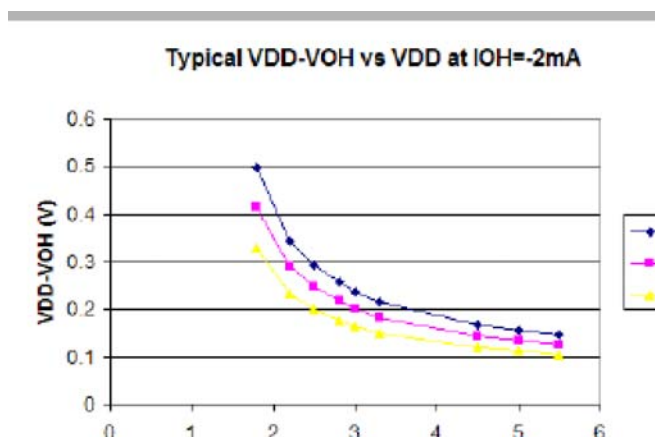


Figure 12-14. Typical $V_{DD}-V_{OH}$ vs. V_{DD} at $I_{OH}=-2mA$

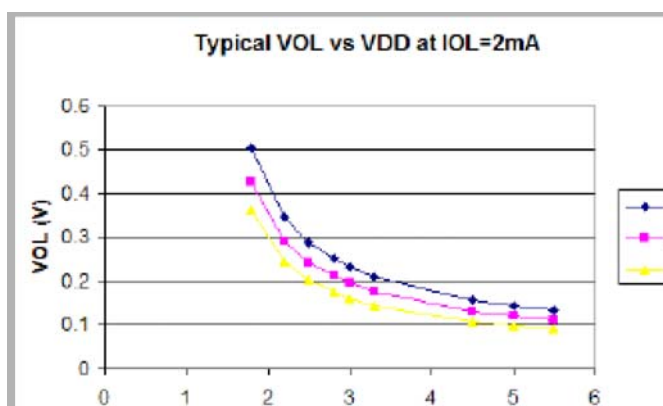


Figure 12-15. Typical V_{OL} vs. V_{DD} at $I_{OL}=2mA$

A.6 电源电流特性

Table A-5. 电源电流特性

参数	符号	V_{DD} (V)	典型值 ⁽¹⁾	最大值 ⁽²⁾	温度 (°C)
在 ($f_{Bus} = 10\text{ MHz}$) 下计算的工作电源电流 ⁽³⁾	RI_{DD10}	5	5.6 mA 5.8 mA	6.5 mA	25 85
		3	4.7 mA 4.8 mA	5.5 mA	25 85
		1.8	2.3 mA 2.4 mA	3 mA	25 85
在 ($f_{Bus} = 1.25\text{ MHz}$) 下计算的工作电源电流 ⁽⁴⁾	RI_{DD1}	5	1 mA 1.1 mA	1.5 mA	25 85
		3	0.9 mA 0.95 mA	1.2 mA	25 85
		1.8	0.6 mA 0.62 mA	0.8 mA	25 85
停止模式下电源电流	SI_{DD}	5	1 μA 3 μA	2 μA 5 μA	25 85
		3	0.9 μA 2.5 μA	2 μA 5 μA	25 85
		1.8	0.7 μA 2 μA	2 μA 4 μA	25 85
停止模式下待隙缓存累加器电流 ($BGBE = 1$)		5	20 μA	30 μA	25
					85
		3	20 μA	30 μA	25
					85
1.8	20 μA	30 μA	25		
			85		

电气特性

Table A-5. 电源电流特性

参数	符号	V _{DD} (V)	典型值 ⁽¹⁾	最大值 ⁽²⁾	温度 (°C)
停止模式下模拟比较器电流 (ACME = 1)		5	15 μA	20 μA	25
					85
		3	15 μA	20 μA	25
					85
		1.8	15 μA	20 μA	25
					85
停止模式下 RTI 电流 使用 1-kHz 时钟源 ⁽⁵⁾		5	300 nA	500nA	25
					85
		3	300 nA	500nA	25
					85
		1.8	300 nA	500nA	25
					85
停止模式下 RTI 电流 使用 32-kHz ICS 内部时钟源参考		5	140 μA	165 μA	25
					85
		3	140 μA	165 μA	25
					85
		1.8	135 μA	160 μA	25
					85
停止模式下 LVI 电流 (LVDE=1 和 LVDSE=1)		5	70 μA	85 μA	25
					85
		3	70 μA	85 μA	25
					85
		1.8	65 μA	80 μA	25
					85

1. 典型值在 25°C 下被测量的。

2. 最大值在正常 V_{DD} 电源误差在 10% 下被测量。这给的值在完成描述之前预先被评估的。

3. 在端口管脚没有接任何直流负载。

4. 在端口管脚没有接任何直流负载。

5. 大部分客户期望得到使用从停止模式自动唤醒代替更高电流的等待模式。等待模式的典型值是在 3V 为 560 μA 和在 2V 为 422 μA，其中 f_{Bus} = 1 MHz。

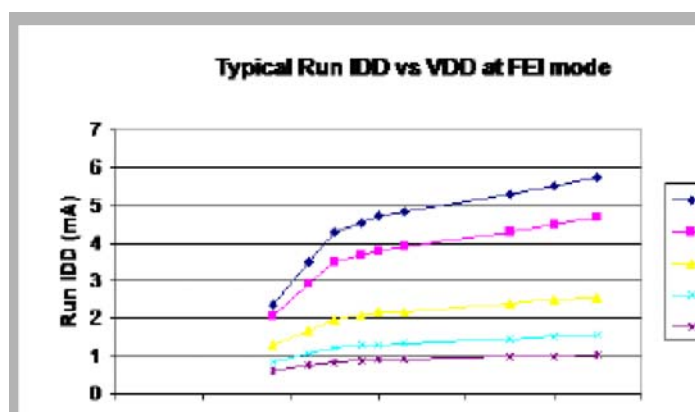


Figure 12-16. Typical Run I_{DD} vs. V_{DD} for FEI mode

A.7 模拟比较器 (ACMP) 电气特性

Table A-6. 模拟比较器电气详述

特性	符号	最小值	典型值	最大值	单位
电源电压	V_{DD}	1.80	—	5.5	V
电源电流 (工作状态)	I_{DDAC}	--	20	35	μA
模拟输入电压	V_{AIN}	$V_{SS} - 0.3$	—	V_{DD}	V
模拟输入偏压 ⁽¹⁾	V_{AIO}	—	20	40	mV
模拟比较器滞后 ¹	V_H	3.0	9.0	15.0	mV
模拟源阻抗	R_{AS}	—	—	10	$k\Omega$
模拟输入漏电流	I_{ALKG}	--	--	1.0	μA
模拟比较器初始化延迟	t_{AINIT}	—	—	1.0	μs
模拟比较器待隙参考电压	V_{BG}	1.208	1.218	1.228	V

1. 这些被描述的数据不是在生产时测试的。芯片进入停止模式时测量值已经获得。

A.8 内部时钟源特性

Table A-7. 内部时钟源详述

特性	符合	最小值	典型值 ⁽¹⁾	最大值	单位
平均内部参考频率—未调整的	f_{int_ut}	25	31.25	41.66	kHz
平均内部参考频率—调整的	f_{int_t}	31.25	31.25	39.0625	kHz
DCO 输出频率范围—未调整的	f_{dco_ut}	12.8	16	21.33	MHz
DCO 输出频率范围—调整的	f_{dco_t}	16	16	20	MHz
在固定电压和温度下调整的 DCO 输出频率精度	$\Delta f_{dco_res_t}$	—	—	± 0.2	$\%f_{dco}$
在过电压和温度下调整的 DCO 输出频率总偏移	Δf_{dco_t}	—	—	± 2	$\%f_{dco}$
FLL 获得时间 ^{(2),(3)}	$t_{acquire}$	—	—	1	ms
停止恢复时间 (FLL 唤醒到预先已获得的频率) IREFSTEN=0 IREFSTEN=1	t_{wakeup}	—	100 86	—	μs

1. 典型值栏中的数据在 3.0 V 和 5.0 V, 25°C 下被描述的, 或是典型的被推荐的值。

电气特性

2. 此参数被描述但不是每片都被测试过。
3. 此详述应用在任何时间: FLL 参考源或参考分频器被改变, 调整值被改变或从 FLL 禁止 (FBILP) 到 FLL 使能 (FEI, FBI) 改变。

A.9 AC 特性

此节描述每个外设系统的 ac 定时特性。

A.9.1 控制时间

Table A-8. 控制时间

参数	符号	最小值	典型值	最大值	单位
总线频率 ($t_{cyc} = 1/f_{Bus}$)	f_{Bus}	dc	—	10	MHz
实时中断内部振荡器周期	t_{RTI}	700	1000	1300	μs
外部 \overline{RESET} 脉冲宽度 ⁽¹⁾	t_{extrst}	150	—	—	ns
KBI 脉冲宽度 ⁽²⁾	t_{KBIPW}	$1.5 t_{cyc}$	—	—	ns
在停止模式下 KBI 脉冲宽度 ¹	t_{KBIPWS}	100	—	—	ns
端口上升和下降时间 (负载 = 50 pF) ⁽³⁾	t_{Rise}, t_{Fall}	—	11	—	ns
斜率控制禁止 (PTxSE = 0)		—	35	—	
斜率控制使能 (PTxSE = 1)		—	—	—	

1. 这是能确保通过管脚输入滤波器电路的最小脉冲。更窄的脉冲也许能或不能被检测到。
2. 这是能确保通过管脚同步电路的最小脉冲宽度。更窄的脉冲也许能或不能被检测到。在停止模式, 同步装置被旁路因此更窄脉冲能被检测到。
3. 定时在 20% V_{DD} 和 80% V_{DD} 电平被展示。温度范围从 $-40^{\circ}C$ 到 $85^{\circ}C$ 。

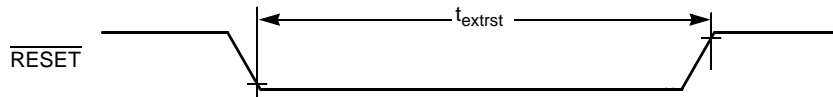


Figure A-1. 复位时间

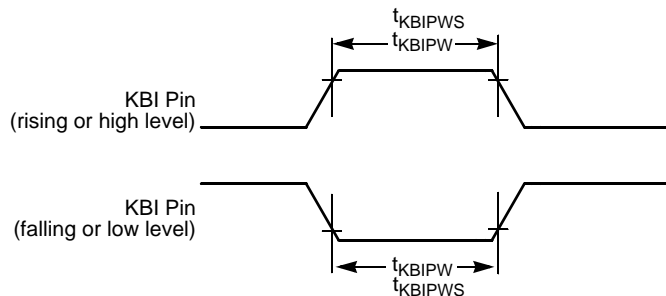


Figure A-2. KBI 脉冲宽度

A.10 FLASH 特性

此节提供了关于 FLASH 存储器的编程 / 擦写时间和编程一擦写强度的详细描述。

关于编程 / 擦写操作的详细信息，参考 [Chapter 4](#), “存储器。”

Table A-9. FLASH 特性

特性	符号	最小值	典型值 I ⁽¹⁾	最大值	单位
编程 / 擦写的电源电压	V _{DD}	2.7	—	5.5	V
编程 / 擦写电压	V _{PP}	11.8	12	12.2	V
V _{PP} 电流					
编程	I _{VPP_prog}	—	—	200	μA
块擦写	I _{VPP_erase}	—	—	100	μA
读操作的电源电压 0 < f _{Bus} < 10 MHz	V _{Read}	1.8	—	5.5	V
字节编程时间	t _{prog}	20	—	40	μs
块擦写时间	t _{me}	500	—	—	ms
累计编程 HV 时间 ⁽²⁾	t _{hv}	—	—	8	ms
总累计编程 HV 时间 (应用在芯片的 t _{me} & t _{hv} 总和)	t _{hv_total}	—	—	2	hours
HVEN 到编程建立时间	t _{pgs}	10	—	—	μs
PGM/MASS 到 HVEN 建立时间	t _{nvs}	5	—	—	μs
对于 PGM, HVEN 保持时间	t _{nvh}	5	—	—	μs
对于 MASS, HVEN 保持时间	t _{nvh1}	100	—	—	μs
V _{PP} 到 PGM/MASS 建立时间	t _{vps}	20	—	—	ns
HVEN 到 V _{PP} 保持时间	t _{vph}	20	—	—	ns
V _{PP} 上升时间 ⁽³⁾	t _{vrs}	200	—	—	ns
恢复时间	t _{rcv}	1	—	—	μs
编程 / 擦写次数 T _L 到 T _H = -40°C 到 +85°C		1000	—	—	周期
数据保持时间	t _{D_ret}	15	100	—	年

1. 典型值在 25°C 下被测试。

2. t_{hv} 是累计高电源编程时间在相同的行且下次擦写之前。在下次擦写之前相同的地址不能被编程两次以上。

3. 快速 V_{PP} 上升时间也许潜在地触发 ESD 保护机构, 它也许产生过流进入这个机构导致永久伤害这个机构。对于 V_{PP} 电源的外部滤波器推荐使用。一个 V_{PP} 滤波举例如 Figure A-3。

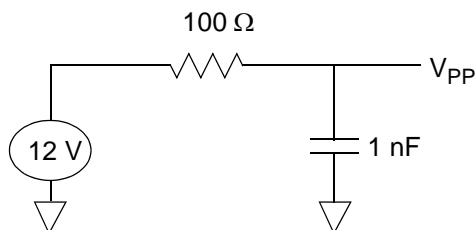
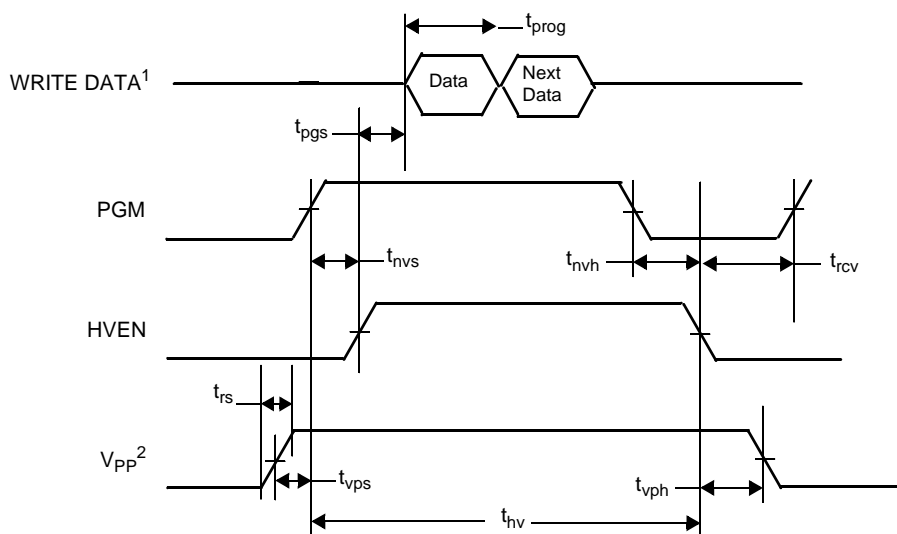


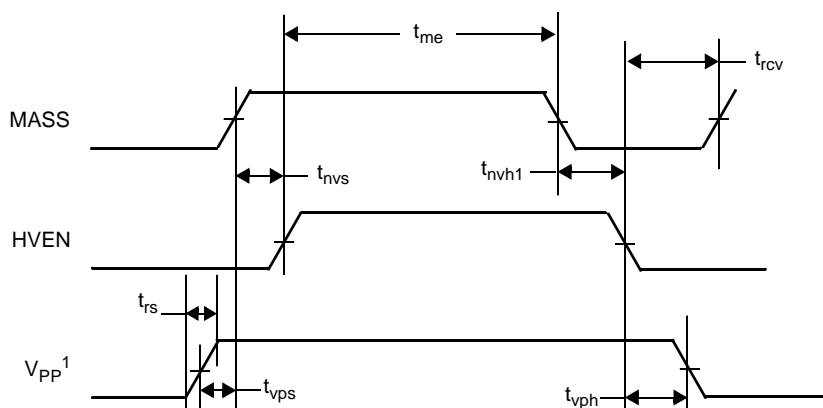
Figure A-3. V_{PP} 滤波举例



¹ N 在单一行假如编程多个字节，下个数据应用，参考 4.6.2, “Flash 编程步骤”。

² 在电压应用或删除到 VPP 管脚之前 VDD 必须是合法的工作电压。

Figure A-4. Flash 编程时间



¹ 在电压应用或删除到 V_{PP} 管脚之前 V_{DD} 必须是合法的工作电压。

Figure A-5. Flash Mass Erase Timing

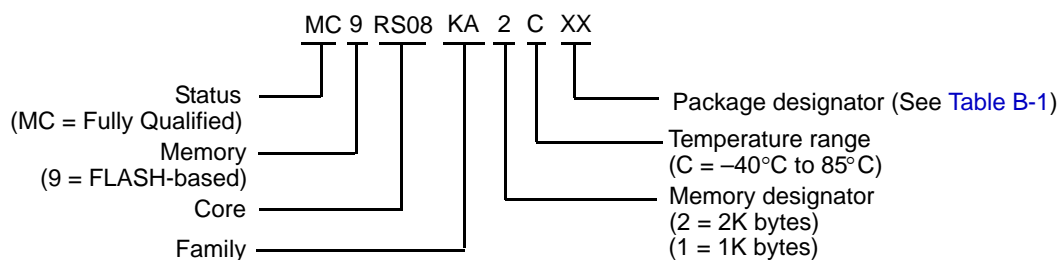
Appendix B 订货信息和机械图

B.1 订货信息

此节包含 MC9RS08KA2 芯片的订货单号。芯片单号系统例子见下面。

Table B-1. 芯片单号系统

芯片单号	存储器		封装		
	FLASH	RAM	典型	命名符	文献号.
MC9RS08KA2 MC9RS08KA1	2K bytes 1K bytes	63 bytes	6 DFN	DB	98ARL01602D
			8 PDIP	PC	98ASB42420B
			8 NB-SOIC	SC	98ASB42564B

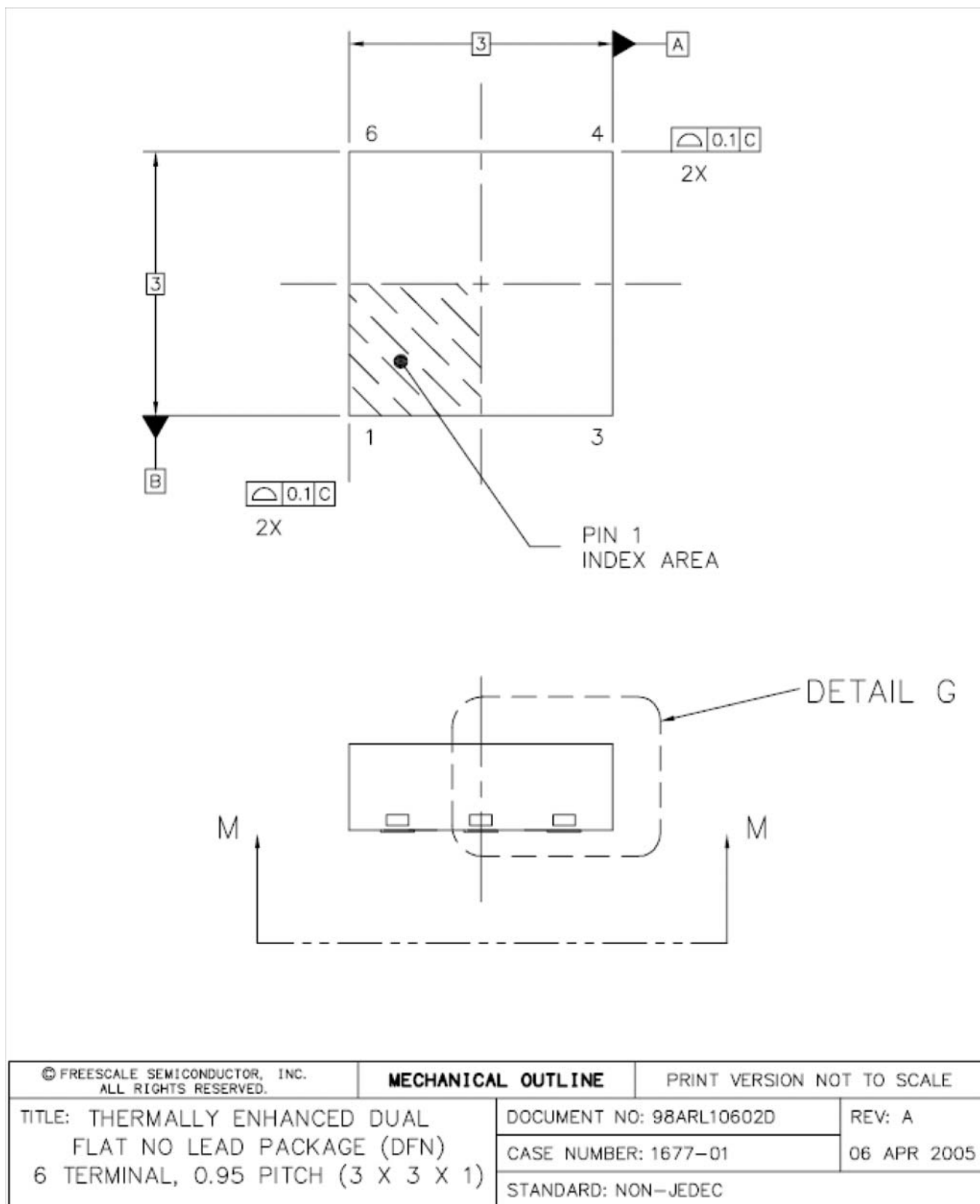


B.2 机械图

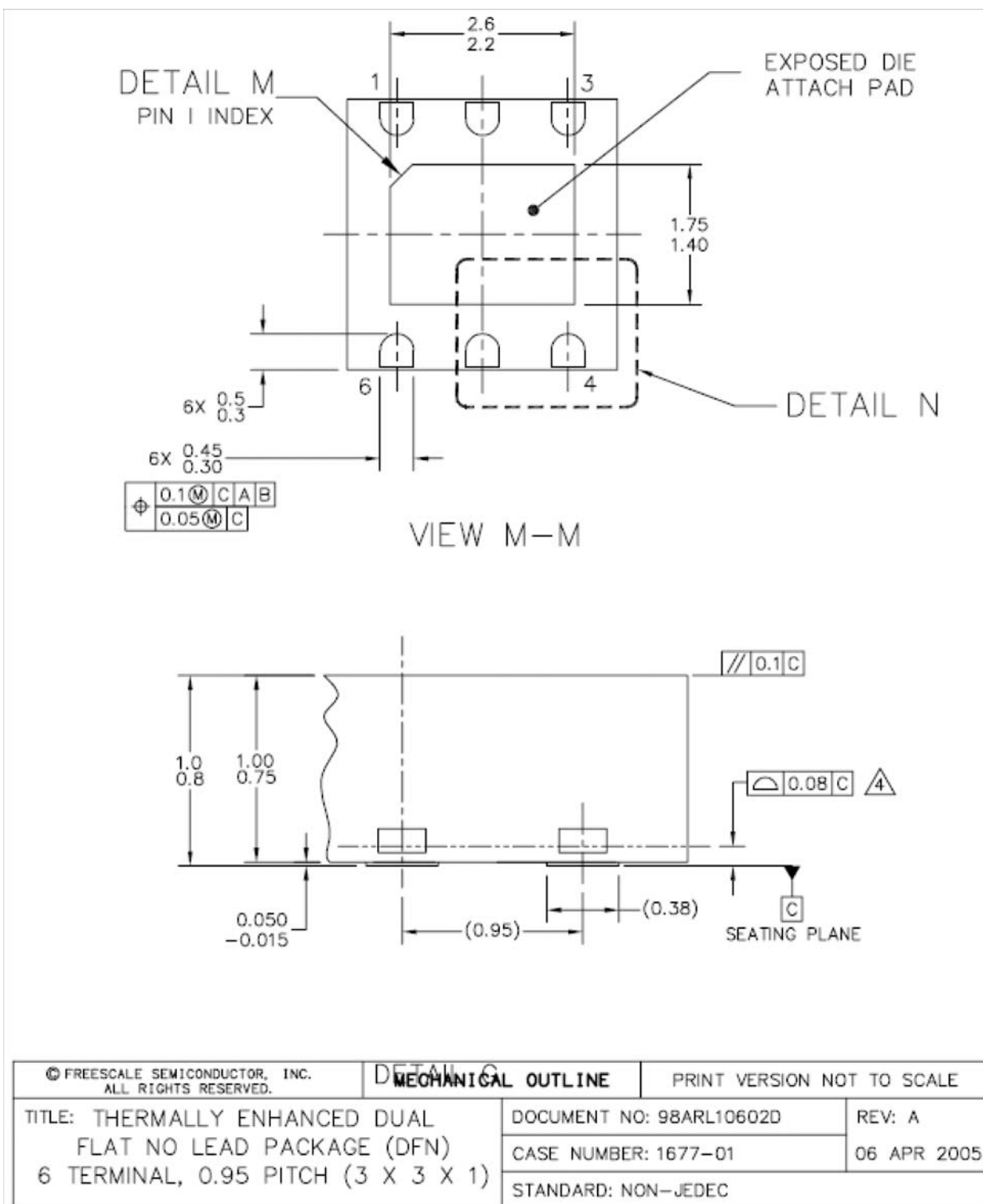
下页保护 MC9RS08KA2 封装的机械规格说明：

- 6-pin DFN (dual flat no-lead)
- 8-pin PDIP (plastic dual in-line pin)
- 8-pin NB-SOIC (narrow body small outline integrated circuit)

订货信息和机械图

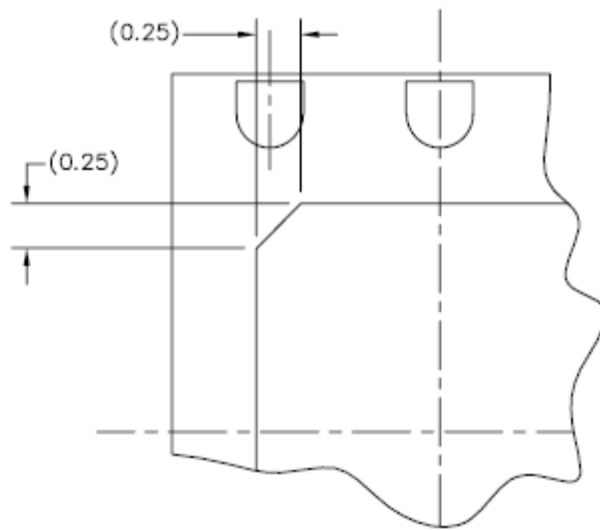


MC9RS08KA2 Data Sheet, Rev. 2.0

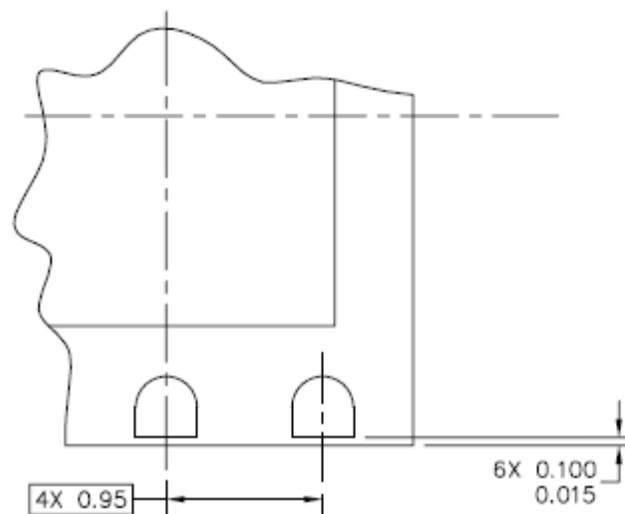


MC9RS08KA2 Data Sheet, Rev. 2.0

订货信息和机械图



DETAIL M
BACKSIDE PIN 1 INDEX




DETAIL N

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED DUAL FLAT NO LEAD PACKAGE (DFN) 6 TERMINAL, 0.95 PITCH (3 X 3 X 1)	DOCUMENT NO: 98ARL10602D	REV: A	
	CASE NUMBER: 1677-01	06 APR 2005	
	STANDARD: NON-JEDEC		

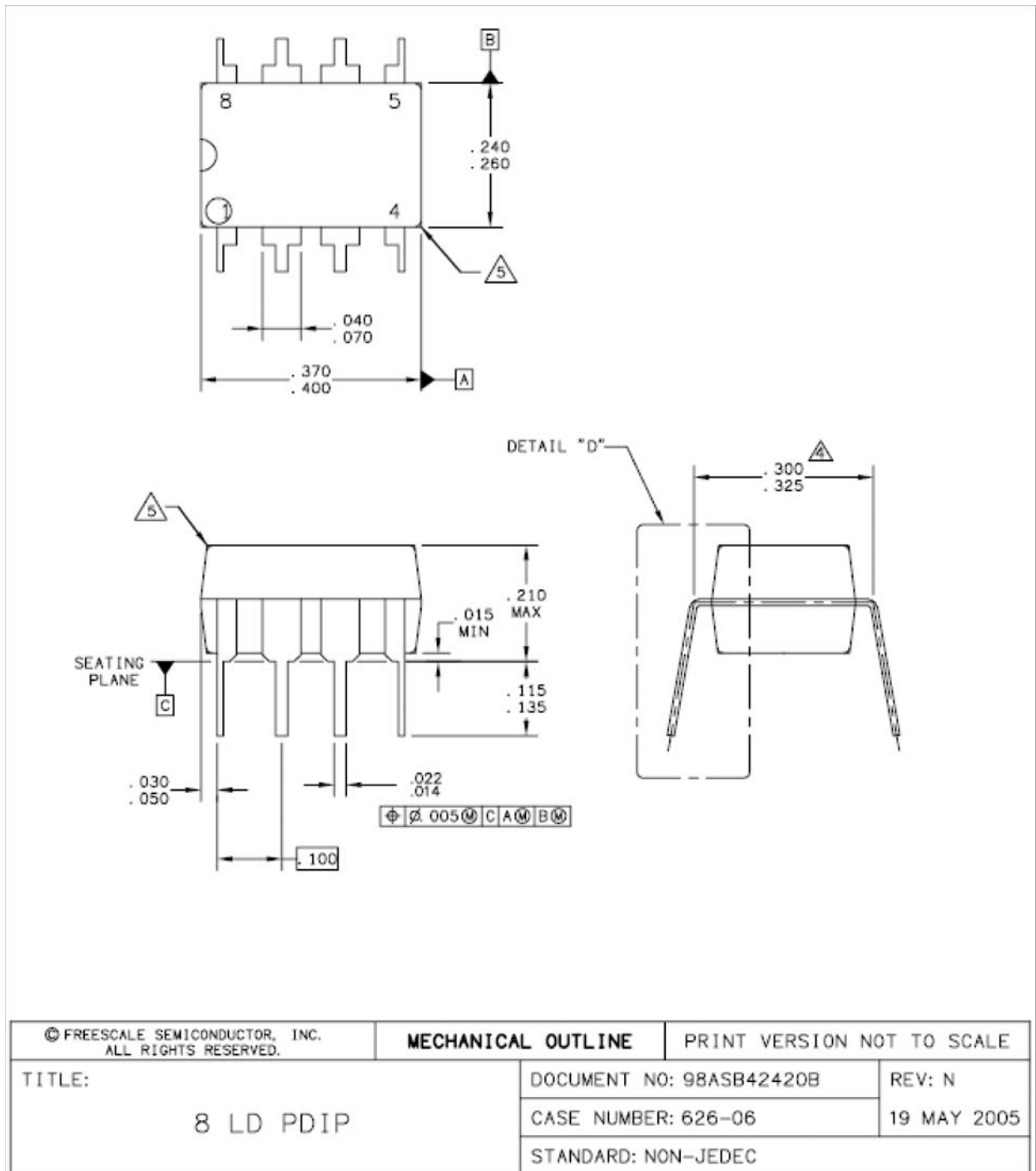
MC9RS08KA2 Data Sheet, Rev. 2.0

NOTES:

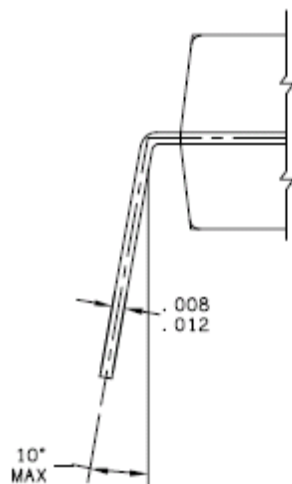
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. THE COMPLETE JEDEC DESIGNATOR FOR THIS PACKAGE IS: HV-PDSO-N.
4.  COPLANARITY APPLIES TO LEADS AND DIE ATTACH PAD.
5. MIN. METAL GAP SHOULD BE 0.2 MM.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE:THERMALLY ENHANCED DUAL FLAT NO LEAD PACKAGE (DFN) 6 TERMINAL, 0.95 PITCH (3X3X1)	DOCUMENT NO: 98ARL10602D	REV: A	
	CASE NUMBER: 1677-01	06 APR 2005	
	STANDARD: NON-JEDEC		

订货信息和机械图



MC9RS08KA2 Data Sheet, Rev. 2.0



DETAIL "D"

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP	DOCUMENT NO: 98ASB42420B	REV: N	
	CASE NUMBER: 626-06	19 MAY 2005	
	STANDARD: NON-JEDEC		

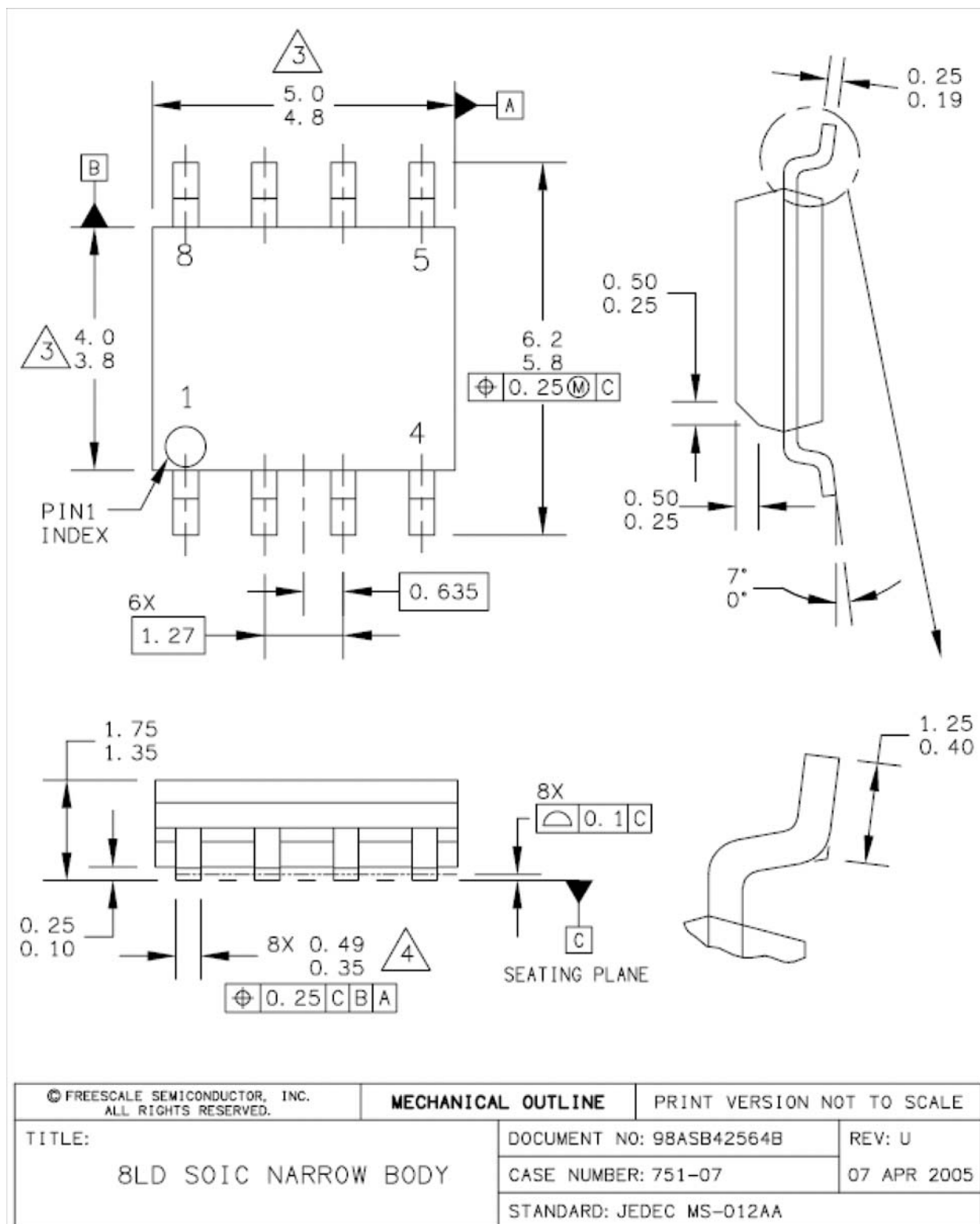
NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M - 1994.
2. ALL DIMENSIONS ARE IN INCHES.
3. 626-03 TO 626-06 OBSOLETE. NEW STANDARD 626-07.
- ⚠ DIMENSION TO CENTER OF LEAD WHEN FORMED PARALLEL.
- ⚠ PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CONERS).

STYLE 1:

PIN	1.	AC IN	5.	GROUND
	2.	DC + IN	6.	OUTPUT
	3.	DC - IN	7.	AUXILIARY
	4.	AC IN	8.	VCC

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP	DOCUMENT NO: 98ASB42420B	REV: N	
	CASE NUMBER: 626-06	19 MAY 2005	
	STANDARD: NON-JEDEC		



MC9RS08KA2 Data Sheet, Rev. 2.0

订货信息和机械图

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
3. DIMENSION DOES NOT INCLUDE MOLD PROTRUSION. MAXIMUM MOLD PROTRUSION 0.15 PER SIDE.
4. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 TOTAL IN EXCESS OF THE DIMENSION AT MAXIMUM MATERIAL CONDITION.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 8LD SOIC NARROW BODY	DOCUMENT NO: 98ASB42564B	REV: U	
	CASE NUMBER: 751-07	07 APR 2005	
	STANDARD: JEDEC MS-012AA		

How to Reach Us:

USA/Europe/Locations not listed:

Freescale Semiconductor Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:

Freescale Semiconductor Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu
Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:

Freescale Semiconductor H.K. Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Learn More:

For more information about Freescale Semiconductor products, please visit <http://www.freescale.com>

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

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

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

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

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2007. All rights reserved.

MC9RS08KA2
Rev. 2
12/2006

