

SC667028

数据手册：技术数据

HCS08
微控制器

SC667028CHS

第 1 版

2008 年 9 月

飞思卡尔半导体

SC667028 产品的特性

8 位 HCS08 中央处理单元 (CPU)

- CPU 的最高频率为 20MHz，工作电压：2.7V-5.5V，温度：-40 °C -85 °C
- HC08 指令子集，增加了 BGND 指令

存储器

- 闪存存在全部操作电压和温度范围内可读 / 写 / 擦除
- RAM
- 使用硬件电路来防止对 RAM 和 Flash 存储器的数据的非法访问

省电模式

- 停止模式
- 使用实时中断 (RTI) 或 ACMP 电源保护模式唤醒

时钟源选项

- 晶振 (XOSC) -- 循环控制晶振，工作范围 31.25KHz-39.0625KHz 或 1MHz-5MHz
- ICS-- 带锁频环 (FLL) 的内时钟源模块 (ICS) 由内部或外部电路控制。精密微调的内部参考可达到 0.2% 的精度和在整个温度和电压范围内 2% 的偏差，支持总线频率最高达 10MHz

系统保护

- 看门狗系统正常操作 (COP) 复位，可选择使用独立的 1KHz 内部时钟源或总线时钟运行。
- 低电压检测复位或中断
- 非法操作符检测复位
- 非法地址检测复位
- Flash 块保护

开发支持

- 单线后台调试接口
- 在线调试时，允许设置单一断点

外部设备

- ADC--12 通道，8 位 AD 转换器，包含 2.5 μ 转换时钟，自动比较功能，能工作在停止模式下，工作电压范围 2.7V-5.5V
- TPM--2 通道，可以作为输入捕捉，输出比较，或带缓冲中央对齐 PWM 通道
- MTIM1 和 MTIM2--2 个 8 位时钟模块
- ACMP--模拟比较器；全工作电压轨对轨操作；能选择固定的内部带隙基准参考电压进行比较；能在工作在停止模式下

输入 / 输出

- 18 个通用输入 / 输出 (I/O) 引脚，其中有一个引脚只能作为输入，有一个只能作为输出。
- 输入时，每个端口都有软件选择的上拉电阻；输出时，每个端口都有软件选择的驱动强度和边沿斜率控制

封装选择

- 20 引脚 SOIC

SC667028 数据手册

SC667028CHS

第 1 版

2008 年 9 月



版本记录

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

<http://freescale.com/>

以下为本手册的修订记录：

版本号	出版日期	更新描述
1	9/11/2008	第一次正式发布

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

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

章节号	标题	页
第 1 章	微控制器概述.....	15
第 2 章	引脚与连接.....	17
第 3 章	操作模式.....	21
第 4 章	存储器.....	25
第 5 章	复位、中断和系统控制.....	35
第 6 章	并行输入 / 输出控制.....	45
第 7 章	中央处理器单元 (RS08CPUV1).....	55
第 8 章	模拟比较器 (RS08ACMPV1).....	69
第 9 章	AD 转换器 (RS08ADCV1).....	77
第 10 章	内部时钟源 (RS08ICSV1).....	99
第 11 章	模定时器 (RS08MTIMV1).....	109
第 12 章	16 位定时器 /PWM(RS08TPMV2).....	119
第 13 章	开发支持.....	139
附录 A	电气特性.....	149
附录 B	订购信息.....	165



手册目录

章节号	标题	页码
第 1 章 微控制器概述		
1.1	概述	15
1.2	MCU 框图	15
1.3	系统时钟分配	16
第 2 章 引脚与连接		
2.1	引言	17
2.2	芯片引脚分配	17
2.3	推荐的系统连接	17
2.4	引脚描述	18
2.4.1	电源引脚	18
2.4.2	PTA5/TCLK/RESET/V _{pp} 引脚	19
2.4.3	PTA4/ACMPO/BKGD/MS 引脚	19
2.4.4	GPIO 及外设端口	19
第 3 章 操作模式		
3.1	引言	21
3.2	特性	21
3.3	运行模式	21
3.4	激活后台模式	21
3.5	停止模式	22
3.5.1	停止模式下激活的 BDM 使能	23
3.5.2	停止模式下 LVD 使能	23
第 4 章 存储器		
4.1	存储器映像	25
4.2	无效的存储器	26
4.3	变址 / 间接寻址	26
4.4	RAM、寄存器地址和位分配	27
4.5	RAM	30
4.6	Flash	30
4.6.1	特性	30
4.6.2	Flash 编程步骤	31
4.6.3	Flash 块擦除操作	31
4.6.4	安全性	32

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、中断和系统控制

5.1	引言	35
5.2	特性	35
5.3	MCU 复位	35
5.4	看门狗 (COP)	36
5.5	中断	36
5.6	低电压检测系统 (LVD)	36
5.6.1	上电复位操作	37
5.6.2	LVD 复位操作	37
5.6.3	LVD 中断操作	37
5.7	实时中断 (RTI)	37
5.8	复位、中断、系统控制寄存器和控制位	37
5.8.1	系统复位状态寄存器 (SRS)	37
5.8.2	系统选项寄存器 (SOPT)	39
5.8.3	系统设备识别寄存器 (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 1 (SPMSC1)	42
5.8.6	系统中断保留寄存器 (SIP1)	42

第 6 章 并行输入 / 输出控制

6.1	低功耗模式下的引脚操作	46
6.2	并口寄存器	46
6.2.1	A 口寄存器	46
6.2.2	B 口寄存器	47
6.2.3	C 口寄存器	47
6.3	引脚控制寄存器	48
6.3.1	A 口引脚控制寄存器	49
6.3.2	B 口引脚控制寄存器	50
6.3.3	C 口引脚控制寄存器	52

第 7 章 中央处理器单元 (RS08CPUV1)

7.1	介绍	55
7.2	程序设计器模型和 CPU 寄存器	55
7.2.1	程序计数器 (PC)	56
7.2.2	影子程序计数器 (SPC)	56
7.2.3	条件代码寄存器 (CCR)	57
7.2.4	变址数据寄存器 (D[X])	58

7.2.5	变址寄存器 (X)	58
7.2.6	页选择寄存器 (PAGESEL)	58
7.3	寻址方式	58
7.3.1	隐含寻址方式 (INH)	58
7.3.2	相对寻址方式 (REL)	58
7.3.3	立即数寻址方式 (IMM)	59
7.3.4	小型寻址方式 (TNY)	59
7.3.5	短型寻址方式 (SRT)	59
7.3.6	直接寻址方式 (DIR)	59
7.3.7	扩展寻址方式 (EXT)	59
7.3.8	变址寻址方式 (IX, 被伪指令执行)	59
7.4	特殊操作	60
7.4.1	复位顺序	60
7.4.2	中断	60
7.4.3	等待和停止模式	60
7.4.4	工作背景模式	60
7.5	指令表摘要	61

第 8 章 模拟比较器 (RS08ACMPV1)

8.1	介绍	69
8.1.1	特性	72
8.1.2	工作模式	72
8.1.3	方框图	72
8.2	外部信号描述	74
8.3	寄存器定义	74
8.3.1	ACMPx 状态和控制寄存器 (ACMPxSC)	74
8.4	功能描述	75

第 9 章 AD 转换器 (RS08ADCV1)

9.1	概述	77
9.1.1	模块配置	77
9.1.2	特性	79
9.1.3	框图	79
9.2	外部信号描述	80
9.2.1	模拟电源 (V_{DDAD})	80
9.2.2	模拟地 (V_{SSAD})	81
9.2.3	参考高电压 (V_{REFH})	81
9.2.4	参考低电压 (V_{REFL})	81
9.2.5	模拟通道输入 (ADx)	81
9.3	寄存器定义	81
9.3.1	状态和控制寄存器 1 (ADCSC1)	81
9.3.2	状态和控制寄存器 2 (ADCSC2)	83
9.3.3	数据高结果寄存器 (ADCRH)	84

9.3.4	数据低结果寄存器 (ADCRL)	84
9.3.5	比较值高寄存器 (ADCCVH)	84
9.3.6	比较值低寄存器 (ADCCVL)	85
9.3.7	配置寄存器 (ADCCFG)	85
9.3.8	引脚控制 1 寄存器 (APCTL1)	86
9.3.9	引脚控制 2 寄存器 (APCTL2)	87
9.3.10	引脚控制 3 寄存器 (APCTL3)	88
9.4	功能描述	89
9.4.1	时钟选择和分频控制	89
9.4.2	输入选择和引脚控制	90
9.4.3	硬件触发	90
9.4.4	转换控制	90
9.4.5	自动比较功能	92
9.4.6	MCU 等待模式操作	92
9.4.7	MCU stop3 模式操作	92
9.4.8	MCU stop1 和 stop2 模式操作	93
9.5	初始化信息	93
9.5.1	ADC 模块初始化举例	93
9.6	应用信息	95
9.6.1	外部引脚和安排	95
9.6.2	错误源	96

第 10 章 内部时钟源 (RS08ICSV1)

10.1	介绍	99
10.1.1	特性	102
10.1.2	工作模式	102
10.1.3	方框图	102
10.2	外部信号描述	103
10.3	寄存器定义	103
10.3.1	ICS 控制寄存器 1 (ICSC1)	103
10.3.2	ICS 控制寄存器 2 (ICSC2)	104
10.3.3	ICS 调整寄存器 (ICSTRM)	105
10.3.4	ICS 状态和控制 (ICSSC)	105
10.4	功能描述	105
10.4.1	工作模式	105
10.4.2	模式切换	107
10.4.3	总线频率分频器	107
10.4.4	低功耗位使用	107
10.4.5	内部参考时钟	107
10.4.6	固定频率时钟	107

第 11 章 模定时器 (RS08MTIMV1)

11.1	介绍	109
------	----	-----

11.1.1 特性	112
11.1.2 工作模式	112
11.1.3 方框图	112
11.2 外部信号描述	113
11.3 寄存器定义	113
11.3.1 MTIM 状态和控制寄存器 (MTIMSC)	114
11.3.2 MTIM 时钟配置寄存器 (MTIMCLK)	114
11.3.3 MTIM 计数寄存器 (MTIMCNT)	115
11.3.4 MTIM 模寄存器 (MTIMMOD)	116
11.4 功能描述	117
11.4.1 MTIM 工作例子	118

第 12 章 16 位定时器 /PWM(RS08TPMV2)

12.1 引言	119
12.1.1 特性	121
12.1.2 框图	121
12.2 外部信号描述	123
12.2.1 外部 TPM 时钟源	123
12.2.2 TPMChn——TPM 通道 n I/O 引脚	123
12.3 寄存器定义	123
12.3.1 定时器状态和控制寄存器 (TPMSC)	123
12.3.2 定时器计数寄存器 (TPMCNTH:TPMCNTL)	125
12.3.3 定时器计数预置寄存器 (TPMMODH:TPMMODL)	125
12.3.4 定时器通道 n 状态和控制寄存器 (TPMCnSC)	126
12.3.5 定时器通道值寄存器 (TPMCnVH:TPMCnVL)	127
12.4 功能描述	128
12.4.1 计数器	128
12.4.2 通道模式选择	129
12.4.3 中央对齐 PWM 模式	130
12.5 TPM 中断	131
12.5.1 清零定时器中断标志	131
12.5.2 定时器溢出中断描述	132
12.5.3 通道事件中断描述	132
12.5.4 PWM 占空比结束事件	132

第 13 章 开发支持

13.1 介绍	139
13.2 特性	139
13.3 RS08 背景调试控制器 (BDC)	140
13.3.1 BKGD 管脚描述	140
13.3.2 通讯详述	141
13.3.3 同步和串行通讯定时溢出	143
13.4 BDC 寄存器和控制位	144

13.4.1 BDC 状态和控制寄存器 (BDCSCR)	144
13.4.2 BDC 断点匹配寄存器	145
13.5 RS08 BDC 命令	146

附录 A 电气特性

A.1 引言	149
A.2 参数分类	149
A.3 最大绝对额定值	149
A.4 热特性	150
A.5 ESD 保护和闭锁抗扰度	151
A.6 DC 特性	151
A.7 电源电流特性	154
A.8 外部振荡器 (XOSC) 特性	156
A.9 AC 特性	157
A.9.1 控制时序	157
A.9.2 TPM/MTIM 模块时序	157
A.10 模拟比较器 (ACMP) 电气	158
A.11 内部时钟源特性	159
A.12 ADC 特性	159
A.13 Flash 规格说明	161
A.14 EMC 性能	163
A.14.1 辐射排放量	163

附录 B 订购信息

B.1 订购信息	165
B.2 机械制图	165

第 1 章 微控制器概述

1.1 概述

SC667028 是一款低成本、引脚数量少、高性能、面向家电及医疗设备的微控制器（MCU）。作为一款通用的微控制器，它由标准的芯片模块组成，包括一个小体积、高性能的 RS08CPU 内核，254 字节 RAM，8K 字节 Flash，两个 8 位时钟模块，12 通道的 8 位 ADC，2 通道的 16 位时钟 /PWM 和模拟比较器。该芯片为 20 引脚 SOIC 封装。

1.2 MCU 框图

图 1-1 显示了 SC667028 MCU 的结构。

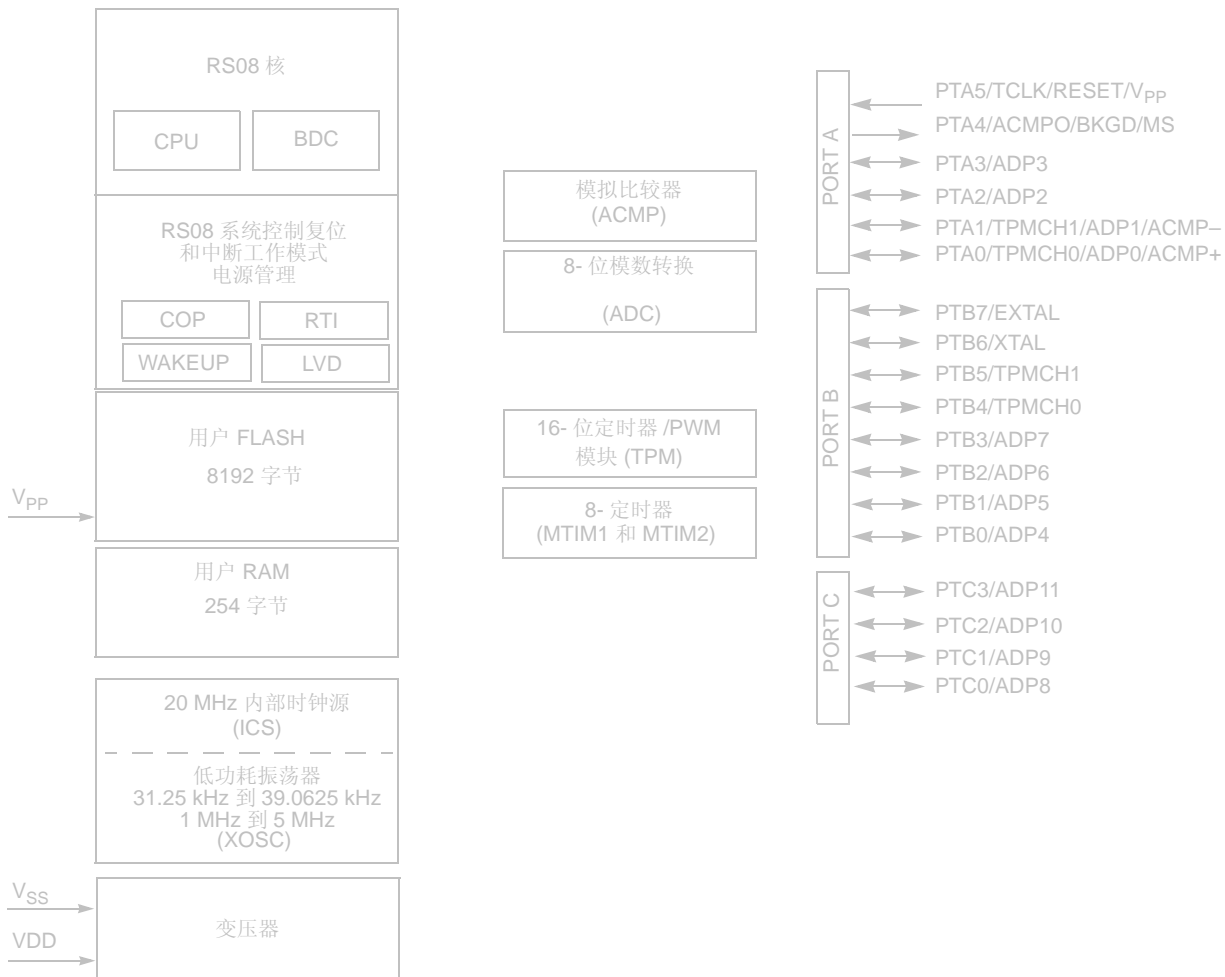


图 1-1. SC667028 MCU 模块结构框图

表 1-1 提供了片上模块的可用版本。

表 1-1. 模块的版本

模块	版本
RS08CPU	1
模拟比较器 (RS08 ACMP)	1
时钟模块 (RS08 MTIM)	1
内部时钟源 (RS08 ICSOSC)	1
模数转换 (RS08 ADC10)	1
16 位定时器 /PWM (RS08 TPM)	2
XOSC	1

1.3 系统时钟分配

图 1-2 显示了 MCU 的简单时钟连接框图。图中总线时钟频率是 ICS 输出频率的 2 分频，并用于所有内部模块。

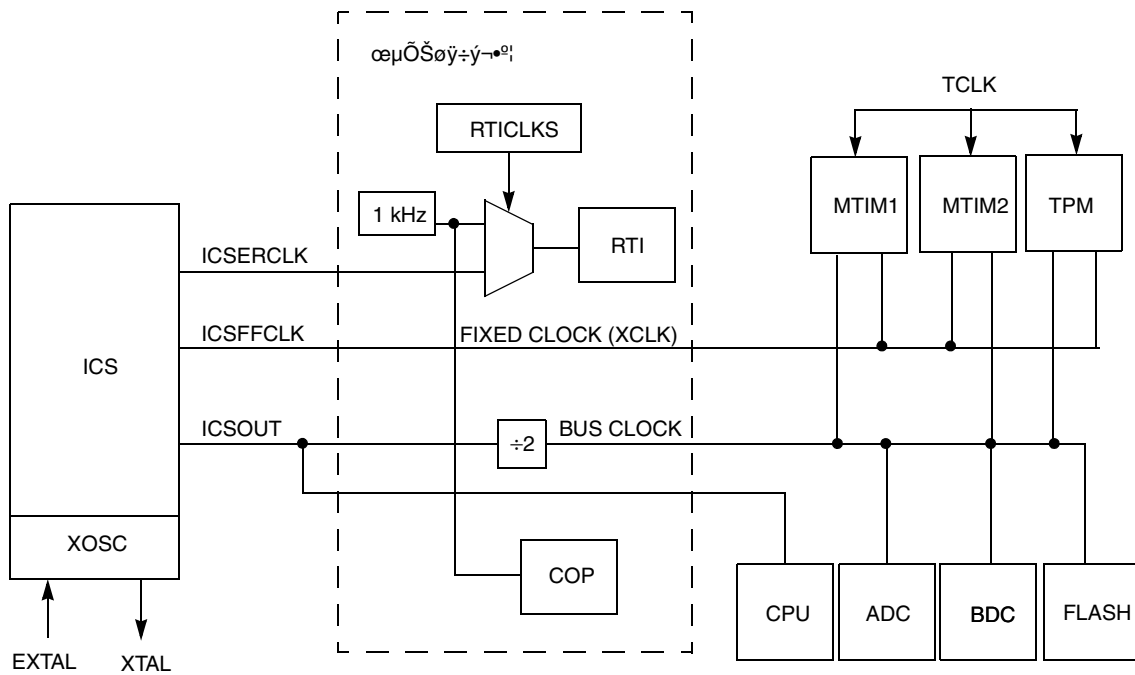


图 1-2. 系统时钟分配图

第 2 章 引脚与连接

2.1 引言

本章描述了连接到封装引脚上的信号。包括引脚输出图、信号特性表和详细的信号功能描述。

2.2 芯片引脚分配

图 2-1 显示了 SC667028 封装的引脚分配。

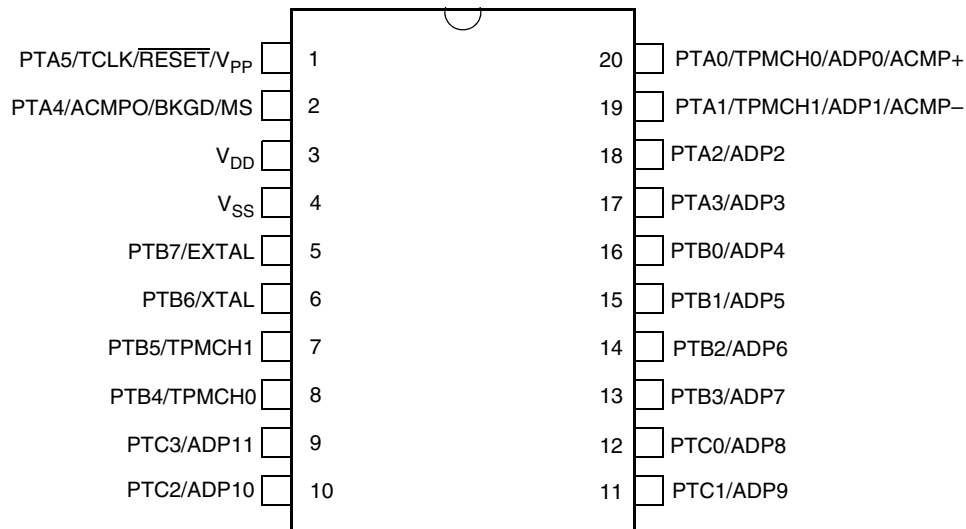
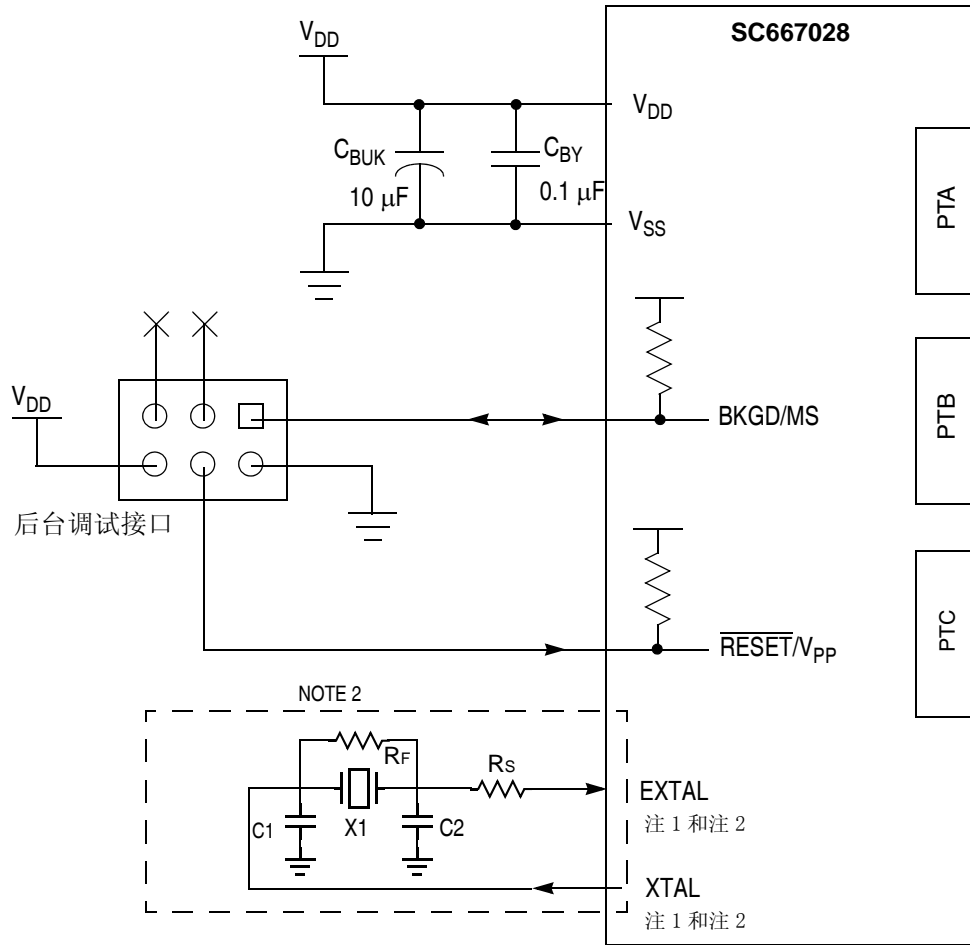


图 2-1. SC667028 20 脚 SOIC 封装图

2.3 推荐的系统连接

图 2-2 显示了系统的后台调试与 FLASH 编程的参考连接。



- PC:
- 1 如果使用内部时钟，不需要外部电路配置。
 - 2 这两个引脚与 PTB6 和 PTB7 复用。

图 2-2. 系统参考连接图

2.4 引脚描述

本节将对系统连接进行详细说明。

2.4.1 电源引脚

V_{DD} 和 V_{SS} 是 MCU 主要电源供应引脚。该电源给所有 I/O 缓冲电路和一个内部电压调节器供电。内部电压调节器提供调节后的低压电源给 CPU 和 MCU 其它内部电路。

一般情况下，实际应用系统中电源引脚接两个电容：一个大容量电解电容器，如 $10\ \mu\text{F}$ 的钽电容，用来为全体系统提供大电量存储，和一个 $0.1\ \mu\text{F}$ 的陶瓷旁路电容，尽可能的靠近 MCU 电源引脚，用以抑制高频噪声。

2.4.2 PTA5/TCLK/RESET/V_{pp} 引脚

上电复位（POR）进入用户模式后，PTA5/TCLK/RESET/V_{pp} 引脚缺省功能为普通输入输出，即 PTA5。设置 SOPT 中的 RSTPE 可以配置该引脚为 RESET 输入引脚。一旦配置为 RESET 引脚，该引脚将保持为 RESET 引脚功能直到下次复位。当 RESET 引脚被拉低时可通过它从外部源复位 MCU。当 RESET 有效时（RSTPE=1），内部上拉电阻自动有效。

当执行 Flash 写入或擦除时，需要一个外部 V_{pp} 电压（典型的为 12 V，可查看附录 A 电气特性）。V_{pp} 始终是连接到内部 Flash 模块，不管该引脚的当前功能如何。在不对 Flash 进行写或擦除工作时，为了避免干扰 Flash，必须撤除外部电压 V_{pp}，同时也必须避免输入电压高于 V_{DD}。

注意

在不进行 Flash 擦写操作时，该引脚不通过钳制二极管连接至 V_{DD} 且不能加高于 V_{DD} 的电压。

2.4.3 PTA4/ACMPO/BKGD/MS 引脚

后台 / 模式选择功能复用 PTA4 引脚，该引脚的另外功能是可以作为输出引脚或模拟比较器的输出。在复位期间，该引脚将作为模式选择引脚。在复位后该引脚立即变为后台引脚被用于后台调试通信。当作为后台 / 模式选择引脚时，有内部上拉电阻。清除 SOPT 寄存器的 BKGDPE 位，该引脚仅作为输出端口使用。

若这个引脚没有连接，则 MCU 在复位的上升沿进入正常的操作模式。如果调试系统被连接到引脚 6 的标准后台调试接口，它可以在复位上升沿时保持 BKGD/MS 为低电平，强制 MCU 进入后台模式。

BKGD 引脚主要被后台调试控制器 (BDC) 使用，它使用私有协议进行通信，该协议使用被调试 MCU 的 16 个 BDC 时钟周期来传送一个位。被调试 MCU 的 BDC 时钟频率使用最大总线时钟频率，因此不要将任何大的电容和 BKGD/MS 引脚相连，否则会干扰 BKGD 与外界的串行通信。

虽然 BKGD 引脚是一个准开漏引脚，但是后台调试通信协议提供了精简的、主动驱动、高速脉冲以确保快速上升沿。内部上拉电阻的寄生小电容对 BKGD 引脚上的上升沿和下降沿几乎不起任何作用。

2.4.4 GPIO 及外设端口

其余引脚与通用 I/O 引脚、片上外设功能模块所复用（比如定时器、模拟比较器）。复位后，所有的引脚被配置为高阻态且没有内部上拉或下拉的通用输入引脚。

注意

为了避免来自浮空输入引脚额外的漏电流，应用程序中的复位初始化例程必须要么使能上拉或下拉电阻，要么改变不常用引脚的方向置为输出使该引脚不再浮空。

表 2-1. 可用引脚

引脚号		<—最低级 优先级 —> 最高级					
20	16	端口引脚	Alt1	Alt2	Alt3		Alt4
1	1	PTA5		TCLK	RESET	V _{PP}	
2	2	PTA4	ACMPO	BKGD	MS		
3	3					V _{DD}	
4	4					V _{SS}	

表 2-1. 可用引脚

引脚号		<—最低级 优先级 —> 最高级					
20	16	端口引脚	Alt1	Alt2	Alt3		Alt4
5	5	PTB7				EXTAL	
6	6	PTB6				XTAL	
7	7	PTB5	TPMCH1 ¹				
8	8	PTB4	TPMCH0 ¹				
9	—	PTC3			ADP11		
10	—	PTC2			ADP10		
11	—	PTC1			ADP9		
12	—	PTC0			ADP8		
13	9	PTB3			ADP7		
14	10	PTB2			ADP6		
15	11	PTB1			ADP5		
16	12	PTB0			ADP4		
17	13	PTA3			ADP3		
18	14	PTA2			ADP2		
19	15	PTA1		TPMCH1 ¹	ADP1	ACMP-	
20	16	PTA0		TPMCH0 ¹	ADP0	ACMP+	

¹ TPM 引脚可被 PTA0 和 PTA1 端口复用

第 3 章 操作模式

3.1 引言

本章将描述 SC667028 的操作模式。同时描述进入各种模式、退出各种模式和在各种模式中的功能。

3.2 特性

- 用于代码调试的后台调试模式
- 停止模式：
 - 系统时钟停止
 - 为快速恢复，所有内部电路不断电

3.3 运行模式

运行模式是 SC667028 在一般情况下的操作模式。在复位上升沿，若 BKGD/MS 为高电平，则进入此模式。在这种模式下，CPU 从内存 \$3FFD 地址处开始执行代码。为实现复位操作，在用户程序中，应使用一个 JMP 指令（操作码为 \$BC）将操作数存放在 \$3FFE-\$3FFF，这个操作数定义了用户应用程序的开始位置。这个用户程序负责执行一个 JMP 指令来重新定位程序计数器到当前用户应用程序的开始位置，而不是像 HC08/S08 系列那样采用中断向量方式。

3.4 激活后台模式

激活后台模式功能由 RS08 内核中的后台调试控制器（BDC）管理。在软件开发期间 BDC 提供各种方式来分析 MCU 操作。

进入后台调试模式的四种方式：

- 上电复位（POR）期间 BKGD/MS 引脚为低电平或发送后台调试强制复位命令
- 当通过 BKGD 引脚接收到后台命令时
- 当 BGND 指令被执行时
- 当进入一个 BDC 断点时

在进入后台模式后，CPU 保持挂起状态等待后台调试命令而不是执行来自用户应用程序的指令。

后台调试命令的两种类型：

- 非插入命令——定义为可在用户程序运行时发出的指令。当 MCU 在运行模式下时，非插入命令可由 BKGD 引脚发出。当 MCU 在后台调试模式下也可执行非插入命令。非插入命令包括：
 - 存储器访问命令
 - 按照状态访问内存命令
 - BACKGROUND 命令
- 激活后台命令——只有当 MCU 在激活后台模式下才可以执行。激活后台调试命令包括以下命令：
 - 读 / 写 CPU 寄存器

- 单步跟踪指令
- 退出激活后台模式，返回用户应用程序 (GO)

激活后台模式通常用于 MCU 第一次在运行模式下运行之前，向 Flash 程序存储器写入用户程序。当 SC667028 出厂时，除非有特别指明，Flash 存储器默认为擦除的，在 Flash 存储器初次被编程之前，由于 Flash 存储器中没有程序，因此在运行模式下没有程序可以运行。激活后台模式也可用于在 Flash 存储器已经写入程序后擦除和重新写入程序。

有关激活后台模式的更多信息可见第 12 章 开发支持。

3.5 停止模式

在系统选项寄存器 (SOPT) 中的 STOPE 位使能时，当执行到停止指令就进入停止模式。在停止模式中，所有的内部时钟和模块被停止。如果 STOPE 位在 CPU 执行停止指令时没有置位，MCU 将不会进入停止模式，并强制产生一个非法指令复位。

表 3-1 总结了 MCU 在停止模式下的行为。

表 3-1. 停止模式下的行为

模式	CPU	数字外设	ICS ¹	ACMP ²	调节器 ³	I/O 引脚	RTI	ADC ⁴
停止	待命	待命	可选	可选	可选	维持状态	可选	可选

¹ 在停止模式下，ICS 要求 IREFSTEN=1，并且 LVDE 和 LVDSE 必须设置成允许操作。

² 如果需要能带隙基准参考电压，在进入停止模式前，SPMSC1 的 LVDE 和 LVDSE 位必须同时被使能。

³ 当 BDM 使能或仅当 SPMSC1 的 LVDE 和 LVDSE 位同时被置位时，调试器处在该模式下工作。

⁴ 当需要异步 ADC 时钟时，SPMSC1 的 LVDE 和 LVDSE 位要同时被置位，否则 ADC 处在待命模式。

当进入停止模式，所有 MCU 时钟被停止。当 IREFSTEN 位被清除且电压调节器进入待命状态时，ICS 被关闭，所有内部寄存器、逻辑寄存器、RAM 内容、I/O 引脚的状态都被保持。

通过设置复位、异步中断 (LVD 中断、KBI 中断、ADC 中断和 ACMP 中断) 或实时中断可退出停止模式。不管在进入停止模式之前 IRQ 是如何设置的，它在停止模式下总是低电平输入有效。

如果是通过设置 **RESET** 引脚来退出停止模式的话，MCU 将重新复位且程序执行的开始定位在 \$3FFD 地址处。如果是通过异步中断或实时中断来退出停止模式，下一条指令将从 STOP 指令执行的位置开始执行，用户程序必须能够探测并响应中断源来唤醒 CPU。

在实时中断中采用一个独立的时钟源 ($\approx 1\text{KHz}$) 就可以从停止模式中唤醒，而不需要额外的元件。当 RTIS=000 时，实时中断的功能是不可用的，当 MCU 处在停止模式，LVD 是不可用的，RTICLKS=1，内部 1KHz 振荡器是不可用的，电源的消耗更小。

在不需要额外的元件的情况下，外部时钟源也可以被使能为产生实时中断 (通过设置 ERCLKEN=1 和 IREFSTEN=1)，唤醒停止模式。

当使 ADC 处在停止模式时，通过设置 LVDE 和 LVDSE 才能使异步 ADC 时钟和 LVD 可用，否则 ADC 在待命状态。

在停止模式下，为了让 XOSC 和外部参考时钟源起作用，必须通过设置 LVDE 和 LVDSE 使 LVD 使能。

3.5.1 停止模式下激活的 BDM 使能

如果 BDCSCR 寄存器里的 ENBDM 标志位被置位，则从运行模式进入激活后台模式被使能。BDCSCR 寄存器将在第 12 章 开发支持介绍。如果当 CPU 执行 S 停止指令时，ENBDM 被置位，这样当 MCU 进入到停止模式后后台调试逻辑的系统时钟仍然在工作，所以后台调试的通信仍然是可以的。此时，电压调节器没有进入低功耗待命状态，而是继续在满负荷工作中。

大部分后台命令在停止模式下是无效的。按照状态访问内存命令不允许存储器存取，但它们会报告一个错误标志，标志 MCU 在停止或等待模式。如果 ENBDM 置位，BACKGROUND 指令可以用于将 MCU 从停止模式中唤醒进入激活后台模式。一旦芯片进入激活后台模式，所有激活后台命令都可用。表 3-2 总结了当 MCU 在进入停止模式时激活后台模块使能的情况下 MCU 的行为。

表 3-2. 停止模式下 BDM 使能时 MCU 行为

模式	CPU	数字外设	ICS	ACMP	电压调节器	I/O 引脚	RTI	ADC
停止	待命	待命	开	可选	开	维持状态	可选	可选

3.5.2 停止模式下 LVD 使能

当供应电压下降到 LVD 电压值以下时，LVD 系统能够产生中断或复位。当 CPU 执行到停止指令时，如果此时已经通过置 LVDE 和 LVDSE 位使得 LVD 有效，则在停止模式下电压调节器仍然保持工作。表 3-3 总结了 MCU 在启用 LVD 的停止模式下的行为。

表 3-3. 启用 LVD 的 STOP 模式下 MCU 的行为

模式	CPU	数字外设	ICS	ACMP	电压调节器	I/O 引脚	RTI	ADC ¹
停止	待命	待命	可选	可选	开	维持状态	可选	可选

¹ 需要设置异步 ADC 时钟为使能。

第 4 章 存储器

4.1 存储器映像

SC667028 的 MCU 的存储器映像分成以下几组：

- 使用小型指令和短指令快速访问 RAM（\$0000-\$000D）
- 间接数据访问 D[X]（\$000E）
- D[X] 的变址寄存器 X（\$000F）
- 频繁访问外设寄存器（\$0010-\$001E，\$0020-\$002F）
- PAGESEL 寄存器（\$001F）
- SC9RS08MZ8 的 RAM（\$0030-\$00BF，\$0100-\$015F）
- 分页窗口（\$00C0-\$00FF）
- 其他外设寄存器（\$0200-\$023F）
- SC9RS08MZ8 的非易失性存储器（\$2000-\$3FFF）

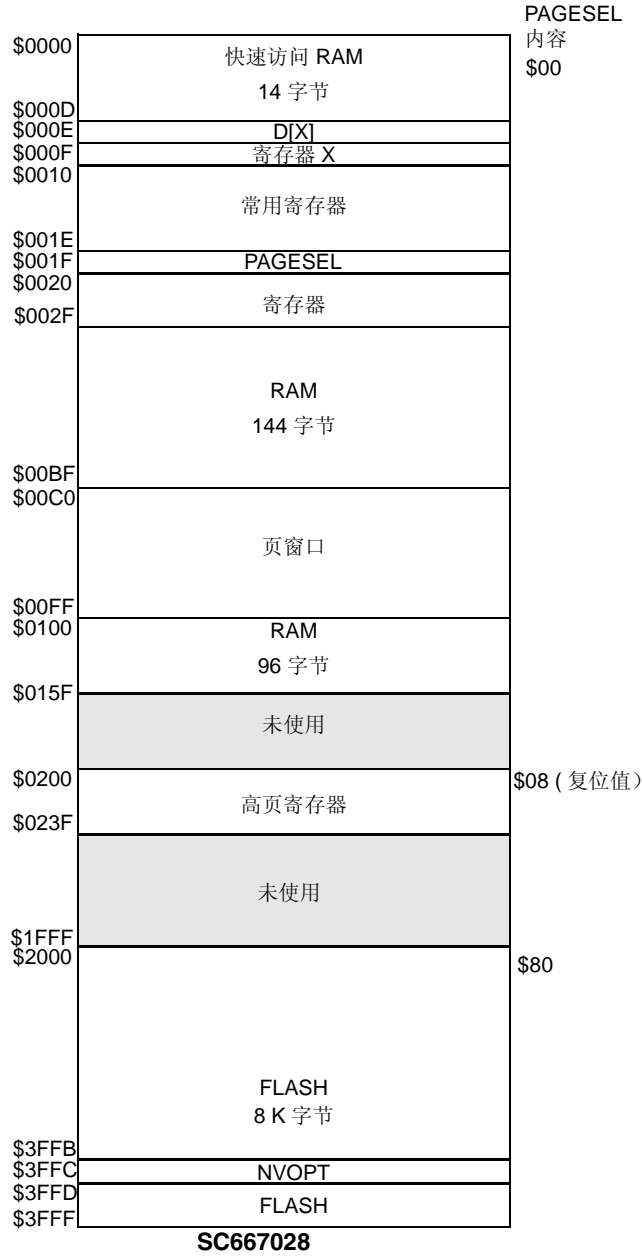


图 4-1. SC667028 的存储器映像

4.2 无效的存储器

试图从无效的存储器地址中去访问数据或执行指令将引起复位。

4.3 变址 / 间接寻址

寄存器 D[X] 和寄存器 X 一起实现对间接数据的存取，寄存器 D[X] 的映像地址为 \$000E，寄存器 X 地址为 \$000F。当寄存器 D[X] 被访问时，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 0。通过编程寄存器 X，位于第一页（\$0000-\$00FF）内的任何地址都能够通过寄存器 D[X] 被读 / 写。图 4-2 显示

了 $D[X]$ 和寄存器 X 之间的关系。例如，当寄存器 X 作为索引值时，HC08/S08 的语法 `lata D[X]`，相当于 RS08 的语句 `lata D[X]`。

物理地址 $\$000E$ 在 RAM 区，当寄存器 X 的值为 $\$0E$ 时，可通过寄存器 $D[X]$ 来获得地址为 $\$000E$ 的 RAM 的内容，写入这个地址可以修改寄存器 X 。

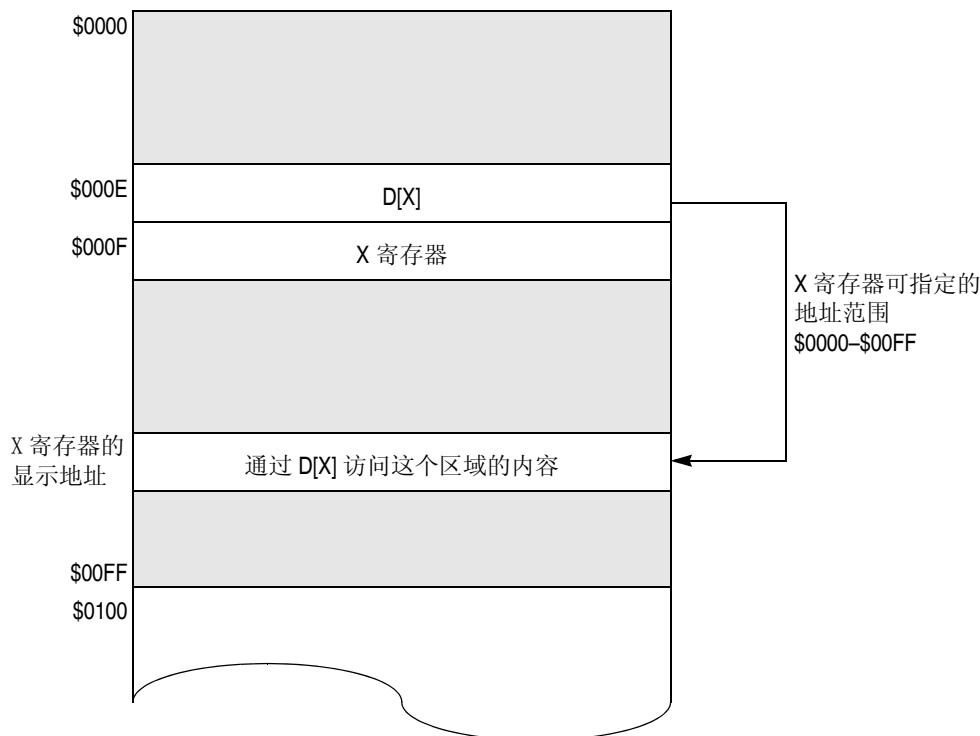


图 4-2. 间接地址寄存器

4.4 RAM、寄存器地址和位分配

通过使用短型寻址和直接寻址模式读写指令可以快速地访问 RAM 区域，使用一个字节的小型寻址模式指令可以将操作数包含在操作码中。

频繁使用寄存器可以使短型寻址模式指令实现快速的装入、存储和清 0 操作。使用一个字节的短寻址模式指令可以将操作数包含在操作码中。

表 4-1. 寄存器 (第 1 页, 共 3 页)

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	ADCSC1	COCO	AIEN	ADCO	ADCH				
\$0011	ADCSC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
\$0012	ADCRH	0	0	0	0	0	0	ADR9	ADR8
\$0013	ADCRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
\$0014	Reserved	—	—	—	—	—	—	—	—
\$0015	TPMC0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0016	TPMC0VH	Bit15	14	13	12	11	10	9	Bit8
\$0017	TPMC0VL	Bit7	6	5	4	3	2	1	Bit0
\$0018	TPMC1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
\$0019	TPMC1VH	Bit15	14	13	12	11	10	9	Bit8
\$001A	TPMC1VL	Bit7	6	5	4	3	2	1	Bit0
\$001B	ACMPSC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD	
\$001C	PTAD	0	0	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
\$001D	PTBD	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
\$001E	PTCD	0	0	0	0	PTCD3	PTCD2	PTCD1	PTCD0
\$001F	PAGESEL	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6
\$0020	MTIM1SC	TOF	TOIE	TRST	TSTP	0	0	0	0
\$0021	MTIM1CLK	0	0	CLKS			PS		
\$0022	MTIM1CNT	COUNT							
\$0023	MTIM1MOD	MOD							
\$0024	MTIM2SC	TOF	TOIE	TRST	TSTP	0	0	0	0
\$0025	MTIM2CLK	0	0	CLKS			PS		
\$0026	MTIM2CNT	COUNT							
\$0027	MTIM2MOD	MOD							
\$0028 – \$002F	Reserved	—	—	—	—	—	—	—	—
\$0030 – \$00BF	—	RAM							
\$00C0 – \$00FF	—	Paging Window							
\$0100 – \$015F	—	RAM							
\$0160 – \$01FF	Unimplemented	—	—	—	—	—	—	—	—
\$0200	SRS	POR	PIN	COP	ILOP	ILAD	0	LVD	0
\$0201	SOPT	COPE	COPT	STOPE	—	TPMCH1P S	TPMCH0P S	BKGDPE	RSTPE
\$0202	SIP1	—	—	ACMP	ADC	TPM	MTIM2	MTIM1	RTI
\$0203	Unimplemented	—	—	—	—	—	—	—	—

表 4-1. 寄存器 (第 2 页, 共 3 页)

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$0204	Reserved	—	—	—	—	—	—	—	—
\$0205	Unimplemented	—	—	—	—	—	—	—	—
\$0206	SDIDH	REV3	REV2	REV1	REV0	ID[11:8]			
\$0207	SDIDL	ID[7:0]							
\$0208	SRTISC	RTIF	RTIACK	RTICLK5	RTIE	0	RTIS		
\$0209	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	BGBE
\$020A – \$020B	Reserved	—	—	—	—	—	—	—	—
\$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	ADCCVH	0	0	0	0	—	—	ADCV9	ADCV8
\$0215	ADCCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
\$0216	ADCCFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
\$0217	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
\$0218	APCTL2	0	0	0	0	ADPC11	ADPC10	ADPC9	ADPC8
\$0219 – \$021F	Unimplemented	—	—	—	—	—	—	—	—
\$0220	PTADD	0	0	0	0	PTADD3	PTADD2	PTADD1	PTADD0
\$0221	PTAPE	0	0	PTAPE5	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0
\$0222	PTAPUD	0	0	PTAPUD5	0	PTAPUD3	PTAPUD2	PTAPUD1	PTAPUD0
\$0223	PTASE	0	0	0	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
\$0224	PTBDD	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
\$0225	PTBPE	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
\$0226	PTBPUD	PTBPUD7	PTBPUD6	PTBPUD5	PTBPUD4	PTBPUD3	PTBPUD2	PTBPUD1	PTBPUD0
\$0227	PTBSE	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
\$0228	PTCDD	0	0	0	0	PTCDD3	PTCDD2	PTCDD1	PTCDD0
\$0229	PTCPE	0	0	0	0	PTCPE3	PTCPE2	PTCPE1	PTCPE0
\$022A	PTCPUD	0	0	0	0	PTCPUD3	PTCPUD2	PTCPUD1	PTCPUD0
\$022B	PTCSE	0	0	0	0	PTCSE3	PTCSE2	PTCSE1	PTCSE0
\$022C – \$022F	Unimplemented	—	—	—	—	—	—	—	—
\$0230	TPMSC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0231	TPMCNTH	Bit15	14	13	12	11	10	9	Bit8
\$0232	TPMCNTL	Bit7	6	5	4	3	2	1	Bit0
\$0233	TPMMODH	Bit15	14	13	12	11	10	9	Bit8
\$0234	TPMMODL	Bit7	6	5	4	3	2	1	Bit0
\$0235	Reserved	—	—	—	—	—	—	—	—
\$0236	Reserved	—	—	—	—	—	—	—	—
\$0237	PTADS	0	0	0	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0

表 4-1. 寄存器 (第 3 页, 共 3 页)

地址	寄存器名称	位 7	6	5	4	3	2	1	位 0
\$0238	PTBDS	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
\$0239	PTCDS	0	0	0	0	PTCDS3	PTCDS2	PTCDS1	PTCDS0
\$023A – \$023B	Unimplemen- ted	—	—	—	—	—	—	—	—
\$023C	ICSC1	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
\$023D	ICSC2	BDIV		RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTE N
\$023E	ICSTRM	TRIM							
\$023F	ICSSC	0	0	0	0	CLKST		OSCINIT	FTRIM
\$3FF8	Reserved	—	—	—	—	—	—	—	—
\$3FF9	Reserved	—	—	—	—	—	—	—	—
\$3FFA ²	Reserved	Reserved for Room Temperature ICS Trim							
\$3FFB ²	Reserved	Reserved							FTRIM
\$3FFC	NVOPT	0	0	0	0	0	0	0	SECD

= 保留或未使用

¹ 当变址寄存器 X 的内容为 \$0E 时, 可通过寄存器 D[X] 来访问物理地址 \$000E。

² 当 MCU 未被调整时, \$3FFA 和 \$3FFB 可以被应用程序使用。

内部时钟源 (ICS) 的工厂校准值将被存储在 \$3FFA 和 \$3FFB, 这个工厂校准值的总线频率为 10MHz。

4.5 RAM

SC667028 包含三个静态 RAM 扇区。位于 \$0000-\$000D 的 RAM 可用更高效的小型寻址方式和短型寻址方式指令来访问, 地址为 \$000E 的 RAM 可以通过寄存器 D[X] (当寄存器 X 为 \$0E) 或通过位于 \$00CE 的分页窗口来访问 (当 PAGESEL 寄存器为 \$00 时)。第二个扇区 RAM 从 \$0030-\$00BF, 这个区域可通过直接寻址方式来访问。第三个扇区 RAM 从 \$0100-\$015F。

当 MCU 处于低压等待和停止模式时, 这时 RAM 中的内容被保持。若提供的电压下降到 RAM 保持的临界电压值以下, 则任何复位都不会影响 RAM 中的数据。

4.6 Flash

Flash 用于存储程序。在线编程允许在产品安装完以后再将程序下载到 flash 中。可以通过单线的后台调试接口对整个 flash 进行写入。因为 flash 不包括片上电荷泵电路, 需要提供外部 V_{pp} 电源来进行编程和擦除操作。

4.6.1 特性

Flash 有以下特性:

- 在典型的电压和温度下, 高达 1000 写入 / 擦除周期
- Flash 安全机制

4.6.2 Flash 编程步骤

Flash 存储器是基于行来编程的，一行由 64 个连续的字节组成，起始地址为 \$2X00，\$2X40，\$2X80 或 \$2XC0，以下是 Flash 存储器写一行的步骤：

1. 加外部 V_{pp} 电源
2. 设置 PGM 位。设置该位，锁存要编程的存储器的地址和数据，开始编程操作。
3. 通过高页存取窗口 \$00C0-\$00FF 可以将任何数据写入 Flash 的任何的任何位置，内部的行地址范围被编程（前提是在数据写入操作的过程中，PAGESEL 寄存器必须正确地配置，去映射高页存取窗口到相关的 Flash 行）
4. 等待时间 t_{nvs}
5. 设置 HVEN 位
6. 等待时间 t_{pgs}
7. 写入被编程的数据到 Flash 指定位置
8. 等待时间 t_{prog}
9. 重复第 7 到 8 步直到行的所有字节被编程
10. 清 PGM 位
11. 等待时间 t_{nvh}
12. 清 HVEN 位
13. 延时时间 t_{rcv} ，又一次在读模式中访问存储器
14. 移除外部 V_{pp} 电源

这个程序序列步骤通过存储器不断地被重复，直到所有数据被编程。

注意

执行 Flash 中的软件代码不能对 Flash 存储器进行编程或擦除。为了能对 Flash 进行编程或擦除，命令必须从 RAM 或 BDC 开始执行，在编程或擦除过程用户程序不能进入等待或停止状态。这些操作必须在以上规定的步骤执行：其他一些不相干的操作可能在这些步骤之间产生。

4.6.3 Flash 块擦除操作

下面是一个完整的 Flash 的块擦除过程：

1. 加外部 V_{pp} 电源
2. 设置 Flash 控制寄存器的 MASS 位
3. 通过高页存取窗口 \$00C0-\$00FF，可以将任何数据写入 Flash 的任何位置（前提是在数据写入操作的过程中，PAGESEL 寄存器必须正确地配置，映射到高页存取窗口的任何 Flash 区域）
4. 等待时间 t_{nvs}
5. 设置 HVEN 位
6. 等待时间 t_{me}
7. 清 MASS 位
8. 等待时间 t_{nvh1}
9. 清 HVEN 位
10. 延时时间 t_{rcv} ，又一次在读模式中访问存储器
11. 移除外部 V_{pp} 电源

注意

在 Flash 中执行软件代码不能对 Flash 存储器进行写入或擦除。为了能对 Flash 进行写入或擦除，命令必须从 RAM 或 BDC 开始执行，在写入或擦除过程用户程序不能进入等待或停止状态。这些操作必须在以上规定的步骤执行：其他一些不相干的操作可能在这些步骤之间产生。

4.6.4 安全性

SC667028 使用硬件电路来防止对 Flash 存储器的数据的非法访问。当安全机制起作用时，Flash 被看作被保护资源。RAM，直接页寄存器，后台调试控制器被看作是不受保护的资源。通过后台调试接口或将 BKGDPPE 置位来访问受保护的存储单元是不允许的（读时总为 0）。

是否使用安全机制是由 FOPT 中的非易失性的寄存器位 SECD 来设置的。在芯片复位时，非易失性的 NVOPT 的内容从 Flash 中复制到工作着的高地址页寄存器 FOPT 中。通过写 NVOPT 来使用 MCU 的安全机制，当该动作与写 Flash 存储器同时进行时也可被执行。擦除的状态 (SECD=1) 使得 MCU 不安全。当在编程时将 NVOPT 的 SECD 清 0，芯片的下一时间将通过 POR 复位、内部复位或外部复位，则 MCU 的安全是保密的。在复位以后通过 BDM 命令来执行块擦除让 MCU 处于非保护的状态。

这个独立的后台调试控制器仍然用于对寄存器和 RAM 的访问，通过 BDM 命令，可以对 Flash 块擦除。写 Flash 控制寄存器的过程遵循 4.6.3 节 Flash 块擦除操作。

安全模式可以通过后台调试接口按照下面的步骤来消除：

1. 通过后台 BDM 命令或 RAM 本地程序可以整体擦除 Flash 存储器。
2. 执行复位，芯片处于非安全状态。

注意

当芯片处于正常操作模式时，在复位期间 MS 引脚为高电平，SECD 为 0，此时 Flash 受安全保护的，BKGDPPE 被复位为 0，所有 BDM 之间的通信是受阻的，后台调试不被允许。

4.7 Flash 寄存器和控制位

Flash 模块有 1 个 Flash 存储器的非易失性寄存器 NVOPT (\$3FFC)，当复位时，内容被复制到相应的控制寄存器 FOPT (\$0210)。

4.7.1 Flash 选项寄存器 (FOPT 和 NVOPT)

复位时，非易失性的 NVOPT 的内容从 Flash 拷贝到 FOPT。第 1 至 7 位在未使用时总是为 0。可以随时读出这个寄存器的值，但写操作是无意义的。改变这个寄存器的值，然后擦除或者写入 Flash 存储器中的 NVOPT 位置，再执行 MCU 复位。

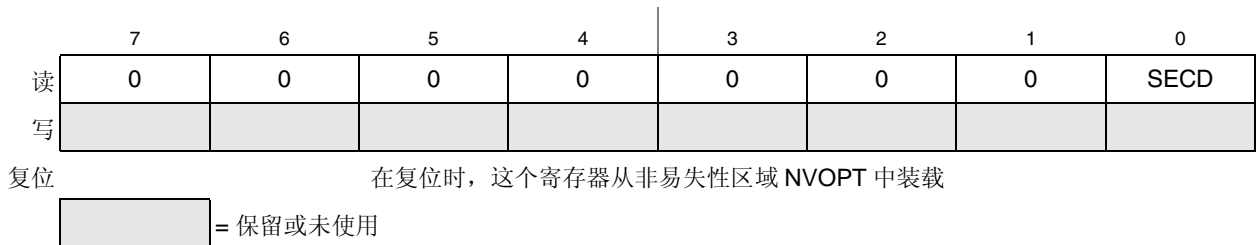


图 4-3. Flash 选项寄存器 (FOPT)

表 4-2. FOPT 字段描述

字段	描述
0 SECD	安全状态码——这位决定 MCU 的安全状态。当 MCU 在安全模式时，Flash 的内容不能被任何来自非法源（包括后台调试接口）的指令访问。安全模式更详细的信息参见 4.6.4 节 安全性。 0 安全保护 1 非安全保护

4.7.2 Flash 控制寄存器 (FLCR)

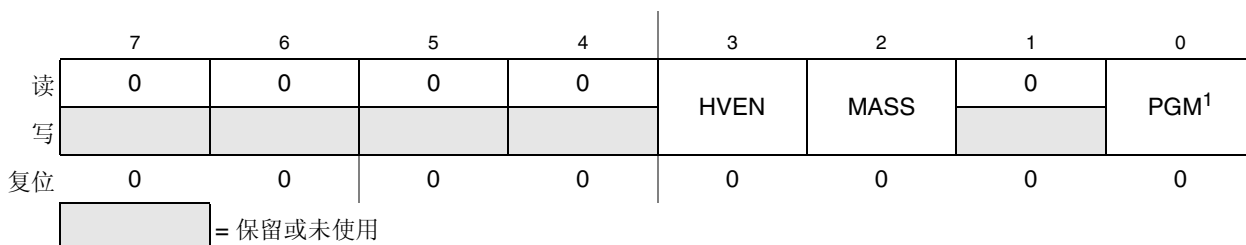


图 4-4. Flash 控制寄存器 (FLCR)

表 4-3. FLCR 字段描述

字段	描述
3 HVEN	使能高压——可读可写位，当将高电平加到 Flash 阵列上可实现编程或擦除操作。当 PGM=1 或 MASS=1 时，HVEN 位可以被设置，然后通过适当的顺序进行编程或擦除。 0 清除 Flash 阵列上的高电平 1 将高电平加到 Flash 阵列
2 MASS	整体擦除控制位——可读可写位，可以将存储器设置为擦除操作 0 不选择擦除操作 1 选择擦除操作
0 PGM ¹	编程（写入）控制位——该位用于设置 Flash 编程操作为编程（写入）操作。PGM1 位与 MASS 位之间存在互锁关系，无法同时被设置为 1 0 不选择写入操作 1 选择写入操作

¹ 当 Flash 处于安全保护时，写 PGM 位是无效的，同时 Flash 编程也是不允许的。

4.8 页选择寄存器 (PAGESEL)

这是一个 64 字节的可供直接页访问的保留空间（\$00C0-\$00FF），编程页选择寄存器决定在存储映像中相关的 64 字节块为直接页存取。例如，当 PAGESEL 在编程时被设置为 \$08，高地址页寄存器（\$0200-\$023F）可以由直接寻址方式指令通过页窗口（\$00C0-\$00FF）访问。

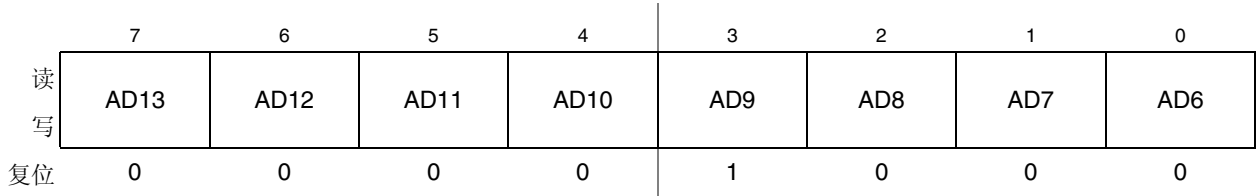


图 4-5. 页选择寄存器 (PAGESEL)

表 4-4. PAGESEL 字段描述

字段	描述
7:0 AD[13:6]	页选择——这些位定义了地址行的第 6 位到 13 位，通过直接页窗口，这些位决定存储器块访问 64 字节块的分界。可查看图 4-6 和表 4-5

图 4-6. 寄存器块边界选择器

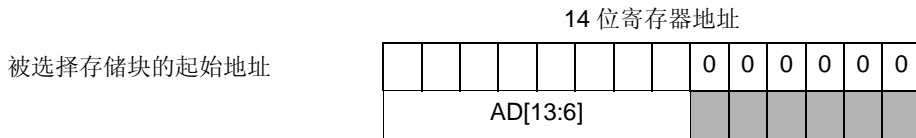


表 4-5 显示了通过页窗口（\$00C0-\$00FF）可被访问的存储器块。

表 4-5. 页窗口（\$00C0-\$00FF）

页	存储器地址
\$00	\$0000-\$003F
\$01	\$0040-\$007F
\$02	\$0080-\$00BF
\$03	\$00C0-\$00FF
\$04	\$0100-\$013F
.....
\$FE	\$3F80-\$3FBF
\$FF	\$3FC0-\$3FFF

注意

RAM 的物理地址为 \$0000-\$000E，X 寄存器地址为 \$000F，D[X] 的映像地址为 \$000E。当 X 值为 \$0E 或 \$CE 且 PAGESEL 为 \$00 时，可通过 D[X] 寄存器来访问 RAM 的 \$000E 地址。当 PAGESEL 为 \$00 时，页窗口被映射到第一页（\$00-\$03F），地址为 \$00C0-\$00CE 的 RAM 页被映射到物理地址 \$0000-\$000E，寄存器 X 的映射地址为 \$00CF，因此访问地址 \$CE 能够得到物理地址为 \$000E 的 RAM 内容，访问 \$000E 返回寄存器 D[X] 的内容。

第 5 章

复位、中断和系统控制

5.1 引言

本章介绍了 SC667028 MCU 基本的复位和中断机制以及各种复位和中断源，一些来自外设模块的中断源在本手册的其他章节会做详细介绍，本章汇总了所有复位和中断源信息以方便参考。有些复位和中断源，如看门狗 (COP) 和实时中断 (RTI) 定时器，没有单独的章节描述的外设的部分，它们仅是系统控制逻辑的一部分。

5.2 特性

复位和中断特性包括：

- 多种复位源用于灵活的系统配置和可靠的操作
- 复位状态寄存器 (SRS) 表明最近发生的复位源
- 系统中断保留寄存器 (SIP1) 表明系统中断的状态
 - 模拟比较器中断使能
 - 时钟模块中断使能
 - 在线读中断使能
 - ADC 中断使能
 - TPM 中断使能

5.3 MCU 复位

复位的 MCU 提供一种从已知的初始条件设置来启动处理的方法。在复位期间，多数控制和状态寄存器被强制为初始值，且程序计数器从 \$3FFD 开始。为正确复位操作，操作数在 \$3FFE-\$3FFF 的 JMP 指令 (操作码 \$BC) 必须被强制写入用户程序，这个操作数定义了用户程序的开始。片上外设模块被关闭，同时 I/O 引脚被初始化配置高阻抗输入，没有上拉或下拉。

SC667028 系列 MCU 包含以下复位源：

- 外部引脚复位 (PIN) — 通过 SOPT 寄存器中的 RSTPE 位使能
- 上电复位 (POR)
- 低电压检测 (LVD)
- 看门狗 (COP) 定时器
- 非法代码检测 (ILOP)
- 非法地址检测 (ILAD)
- 后台调试强制复位

除了后台调试强制复位以外，每种复位源在系统复位状态寄存器 (SRS) 中都有相应位。

5.4 看门狗 (COP)

当应用程序没有按照预期那样执行时，COP 看门狗会强制系统复位。为防止来自 COP 定时器（当它使能时）的系统复位，应用程序必须周期性的复位 COP 计数器。如果应用程序在 COP 定时器溢出前跑飞或者没有复位 COP，就会产生一个系统复位信号使系统回到已知开始点。在任何复位后，SOPT 寄存器中的 COPE 位置位使能 COP 看门狗（详情见 5.8.2 节 系统选项寄存器 (SOPT)）。如果 COP 看门狗在应用程序中未被使用，那么可通过清零 COPE 位来关闭它。通过写任意值到 SRS 的地址可复位 COP 计数器。本次的写入不影响只读寄存器 SRS 中的数据。此时芯片内部电路使得 COP 计数器得到一个复位信号。

SOPT 的 COPT 位控制超时长短。表 5-1 总结了 COPT 位的控制功能。COP 看门狗缺省设置为 1 kHz 时钟操作并为长溢出时间（ 2^8 周期）。

表 5-1. COP 配置选项

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

¹ 本栏中的值基于 $t_{RTI} \approx 1$ ms。t_{RTI} 见附录。

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

在后台调试模式下，COP 计数器计数将不会递增。

当进入停止模式时，COP 计数器会被重新初始化为零。一旦 MCU 退出停止模式 COP 计数器从零开始计数。

5.5 中断

SC667028 没有在 HC08 和 HCS08 器件上使用的向量表查询机制的中断控制器。但是，如 LVD、ADC、RTI 和 ACMP 中断源模块仍然可以用来将 CPU 从等待或停止模式中唤醒，这是由用户应用程序负责轮询模块来唤醒中断源。

每个中断源模块的唤醒要结合相应的中断使能位。如果该位被关闭，中断源被关闭，那么将无法将 CPU 从等待或停止模式中唤醒。但是，相应的中断标志会被设置，并指示出有外部唤醒事件。

系统中断挂起寄存器（SIP1）显示了系统挂起中断的状态。SIP1 中只读位被置位，表明有挂起的中断从被指示的模块请求服务，写该寄存器是无效的。当指示模块中所有相应的中断标志被清除时，挂起中断标志自动被清除。

5.6 低电压检测系统 (LVD)

SC667028 含有一个低电压的保护系统，在电压变更时用来保护存储器的内容和控制 MCU 系统状态。这个系统是由一个上电复位 (POR) 电路和一个预定义电压点的 LVD 电路。当 SPMSC1 的 LVDE 使能，LVD 电路被激活。进入任何停止模式后 LVD 停止工作，除非 SPMSC1 的 LVDSE 位被置位。如果 LVDSE 和 LVDE 都被置位，处于 LVD 被激活的停止模式的功耗将会变大。

5.6.1 上电复位操作

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

5.6.2 LVD 复位操作

LVD 能被设置为在侦测到一个低电压条件时（即 LVDRE 为 1 时）产生一个复位。LVD 复位发生后，LVD 系统将会保持 MCU 在复位状态直到供电电压升高到高于由 V_{LVD} 设定的电压。LVD 复位或上电复位后，SRS 寄存器的 LVD 位将被置位。

5.6.3 LVD 中断操作

当一个低电压条件被检测到，并且 LVD 电路被设置为中断操作（LVDE 置位，LVDIE 置位，LVDRE 清零，SPMSC1 的 LVDF 将会被置位且一个 LVD 中断请求将发生。

5.7 实时中断 (RTI)

实时中断功能可用来产生定期中断。RTI 可被 1KHz 的内部时钟或外部时钟源驱动。SRTISC 中的 RTICLKS 位通常被用来选择 RTI 时钟源。当 MCU 运行在等待或停止模式下时两个时钟源都能被使用。当使用外部时钟时，它必须在正常或等待模式下使能（EREFSTEN = 1），当外部时钟工作在停止模式时，它必须使 EREFSTEN = 1 且 ERCLKEN = 1。

SRTISC 寄存器包括一个只读状态标志，一个只写应答位，和一个 3 位控制值（RTIS）用来选择七个唤醒周期中的一个。该 RTI 含有一个本地中断使能 RTIE，它允许屏蔽实时中断。通过置 RTIS 每一位为零使 RTI 禁止同时没有中断产生。详情见 5.8.4 节 系统实时中断状态和控制寄存器 (SRTISC)。

5.8 复位、中断、系统控制寄存器和控制位

有关直接页寄存器的介绍请参见第 4 章 存储器。这部分只是从概念上来介绍这些的寄存器和控制位。根据 Freescale 提供的通用文件或头文件，可以用来将这些名称转化为相应的绝对地址。

SOPT 寄存器的一些控制位相对应的操作模式，这里只提供简要描述，而相关功能在第 3 章 操作模式里有更加详细的论述。

5.8.1 系统复位状态寄存器（SRS）

这个寄存器包括用于指示最近复位源的只读标志位。当一个调试器通过发送 BDC_RESET 命令时，SRS 的各标志位都会被清除。向该寄存器地址写入任意值可清零 COP 看门狗计数器但不影响寄存器的内容。这些位的复位状态取决于 MCU 复位的原因。

	7	6	5	4	3	2	1	0
读	POR	PIN	COP	ILOP	ILAD	0	LVD	0
写	写任何值到 SRS 地址清零 COP 看门狗定时器							
上电复位:	1	0	0	0	0	0	1	0
低电压复位:	U	0	0	0	0	0	1	0
其他复位:	0	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。

图 5-1. 系统复位状态 (SRS)

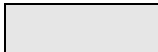
表 5-2. SRS 域描述

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
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。

	7	6	5	4	3	2	1	0
读	COPE	COPT	STOPE		TPMCH1PS	TPMCH0PS	BKGDPE	RSTPE
写								
复位:	1	1	0	0	0	0	1 (Note 1)	u
加电复位:	1	1	0	0	0	0	1 (Note1)	0

 = 保留或未使用

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。

图 5-2. 系统选项寄存器 1 (SOPT)

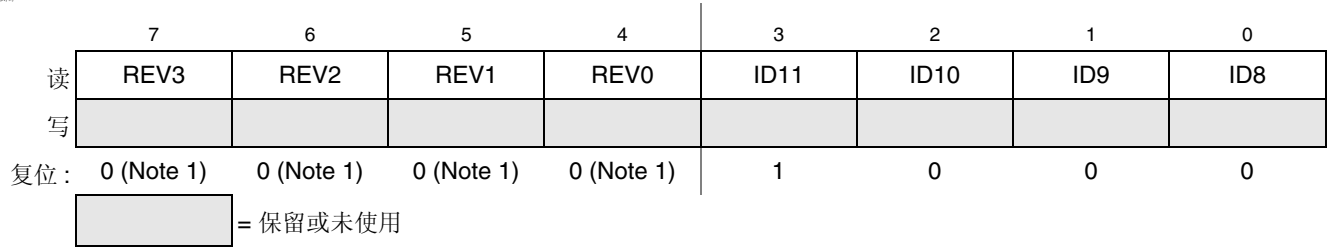
表 5-3. SOPT 寄存器域描述

域	描述
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
3 TPMCH1PS	TPMCH1 引脚选择 -- 这个位选择 TPM 模块的 TPMCH1 引脚 0 PTA1 作为 TPMCH1 引脚 1 PTB5 作为 TPMCH1 引脚
2 TPMCH0PS	TPMCH0 引脚选择 -- 这个位选择 TPM 模块的 TPMCH0 引脚 0 PTA0 作为 TPMCH0 引脚 1 PTB4 作为 TPMCH0 引脚
1 BKGDPE ^{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主机开发系统可以识别 RS08 衍生和修订的码。这允许开发软件认识位于目标 MCU 的专用存储器块, 寄存器和控制位。



- 被硬编码到这些位的修订码反映了当前芯片版本级别。

图 5-3. 系统设备鉴定寄存器高 (SDIDH)

表 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 龔)。
3:0 ID[11:8]	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。SC667028 被硬编程值 \$0800。也可见在图 5-4 中的 ID 位。

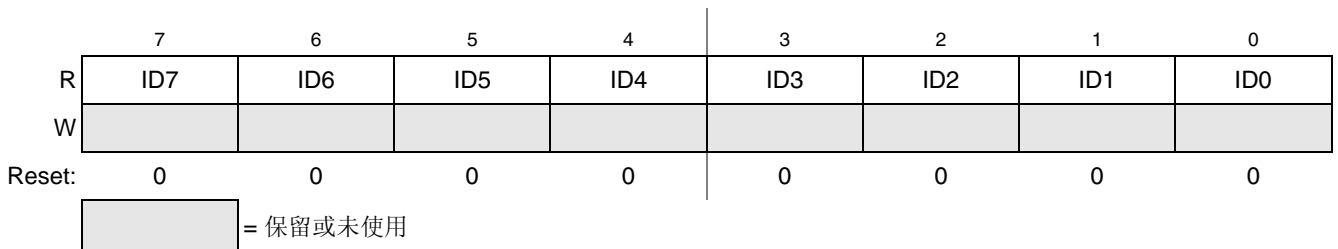


图 5-4. 系统设备识别寄存器 — 低 (SDIDL)

表 5-5. SDIDL 寄存器域描述

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。SC667028 被硬编程值 \$0800。也可见在图 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 RTI 的状态和控制位。

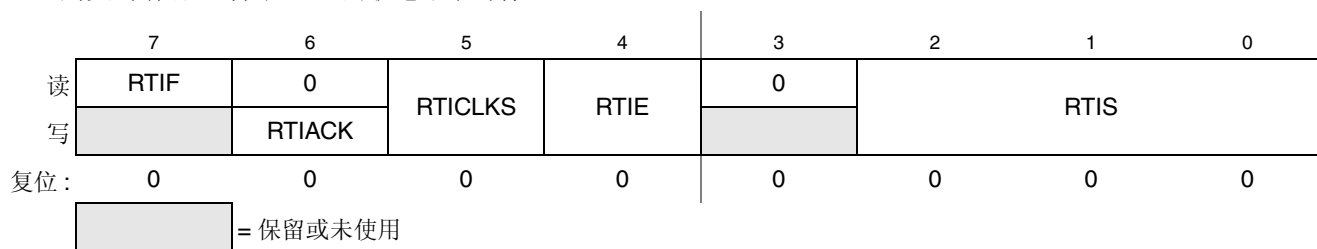


图 5-5. 系统 RTI 状态和控制寄存器 (SRTISC)

表 5-6. SRTISC 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。见表 5-7。

表 5-7. 实时中断周期

RTIS	使用 1KHz 的时钟源 ¹	使用外部时钟
000	RTI 计数器清 0 或不可用	RTI 计数器清 0 或不可用
001	8 ms	$1/f_{\text{extclk}} \times 256$
010	32ms	$1/f_{\text{extclk}} \times 1024$
011	64ms	$1/f_{\text{extclk}} \times 2048$
100	128ms	$1/f_{\text{extclk}} \times 4096$
101	256ms	$1/f_{\text{extclk}} \times 8192$
110	512ms	$1/f_{\text{extclk}} \times 16384$
111	1024ms	$1/f_{\text{extclk}} \times 32768$

¹ 显示在该表中的值基于 $f_{\text{RTI}}=1 \text{ ms}$ 。详情见附录 A 电气特性。

5.8.5 系统电源管理状态和控制寄存器 1 (SPMSC1)

这个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。

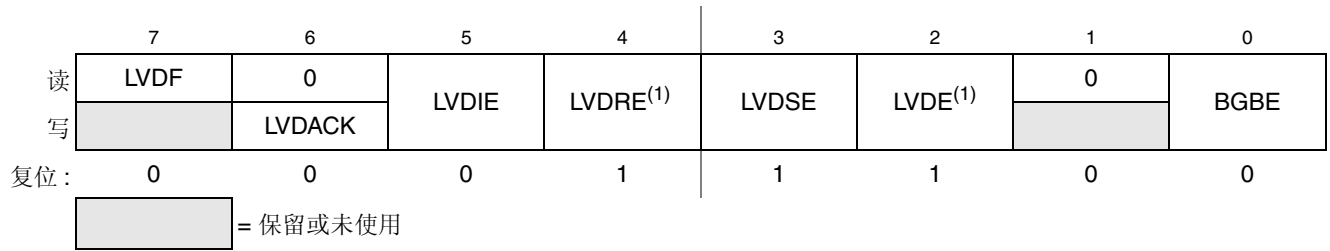


图 5-6. 系统电源管理状态和控制寄存器 1 (SPMSC1)

¹ 复位后此位仅能写一次。额外的写被忽略。

表 5-8. SPMSC1 寄存器字段描述

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

5.8.6 系统中断保留寄存器 (SIP1)

这个高地址页寄存器包含各模块的中断保留位。

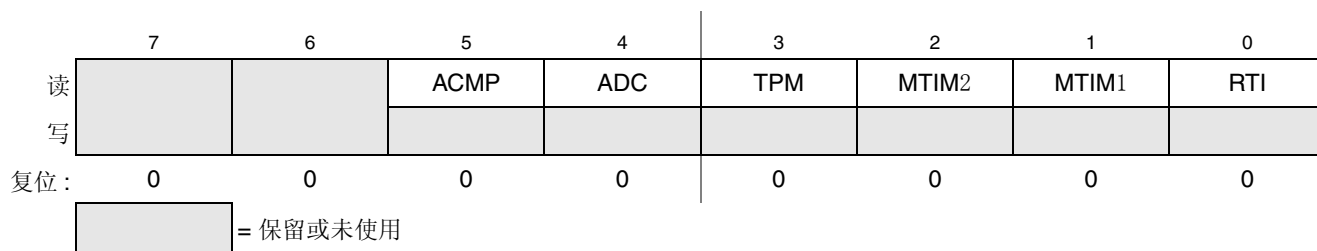


图 5-7. 系统中断保留寄存器 (SIP1)

表 5-9. SIP1 寄存器字段描述

字段	描述
5 ACMP	模拟比较器中断保留位——指示 ACMP 模块的只读保留中断位，当清除 ACMPSC 寄存器的 ACF 标志或复位时，该位被清 0 0 非 ACMP 中断保留位，如 ACF/ACIE 位被清 0 1 ACMP 中断保留位，如 ACF/ACIE 位被置位
4 ADC	ADC 中断保留位——指示 ADC 模块的只读保留中断位，当清除 ADCSC1 寄存器的 COCO 标志或复位时，该位被清 0 0 非 ADC 中断保留位，如 COCO/AIEN 位被清 0 1 ADC 中断保留位，如 COCO/AIEN 位被置位
3 TPM	时钟 /PWM 中断保留位——指示 TPM 模块的只读保留中断位，当清除 TPMSC 寄存器的 TOF 标志或 TPMCnSC 寄存器的 CHnF 或复位时，该位被清 0 0 非时钟 /PWM 中断保留位，如 TOF 标志、TOIE 位、CH0F 标志、CH0IE 位、CH1F 标志或 CH1IE 位被清 0 1 时钟 /PWM 中断保留位，如 TOF 标志、TOIE 位、CH0F 标志、CH0IE 位、CH1F 标志或 CH1IE 位被置位
2 MTIM2	时钟 2 中断保留位——指示 MTIM2 模块的只读保留中断位，当清除 MTIM2SC 寄存器的 TOF 标志或复位时，该位被清 0 0 非 MTIM2 中断保留位，如 TOF/TOIE 位被清 0 1 MTIM2 中断保留位，如 TOF/TOIE 位被置位
1 MTIM1	模定时器 1 中断保留位——指示 MTIM1 模块的只读保留中断位，当清除 MTIM1SC 寄存器的 TOF 标志或复位时，该位被清 0 0 非 MTIM1 中断保留位，如 TOF/TOIE 位被清 0 1 MTIM1 中断保留位，如 TOF/TOIE 位被置位
0 RTI	实时中断保留位——指示 RTI 模块的只读保留中断位，当清除 SRTISC 寄存器的 RTIF 标志或复位时，该位被清 0 0 非 RTI 中断保留位，如 RTIF/RTIE 位被清 0 1 RTI 中断保留位，如 RTIF/RTIE 位被置位

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 RTICKLS 位置 1。

在把引脚方向定义为输出之前向端口数据寄存器写数据。这样确保端口数据寄存器驱动引脚的值不会是旧的数据。

一些和并口相关的寄存器位于高页寄存器空间，并且独立于并口寄存器操作。这些寄存器控制引脚的上拉 / 下拉以及回转速率。参见 6.3 节 引脚控制寄存器

6.1 低功耗模式下的引脚操作

在等待和停止模式，因为内部逻辑保持继续有电，所以所有引脚的状态继续保持。恢复到正常模式后，所有引脚也恢复到进入停止模式之前的状态。

6.2 并口寄存器

本节介绍与并行 I/O 口相关的寄存器。并口数据寄存器位于起始地址为 \$001F 的存储器映像中，这样可以使用短地址和直接地址寻址指令。

涉及到并行 I/O 口绝对地址分配的相关表格，参见第 4 章 存储器。这节涉及到的寄存器和控制位只通过它们的名字引用。飞思卡尔将提供一个头文件将这些名称转换成对应的绝对地址。

6.2.1 A 口寄存器

A 口并行输入 / 输出功能通过本节描述的数据寄存器和数据方向寄存器控制。

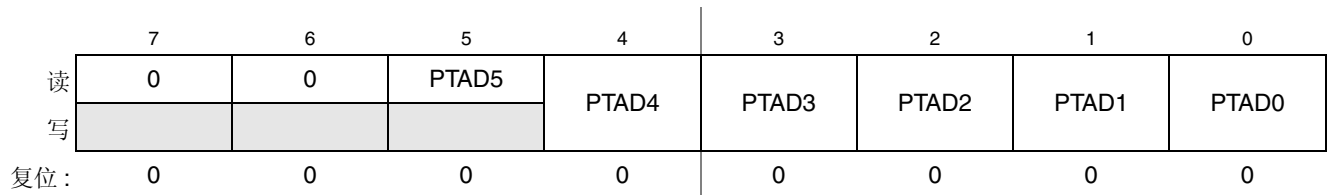


图 6-2. A 口数据寄存器 (PTAD)

表 6-1. PTAD 寄存器域描述

域	描述
5:0 PTAD[5:0]	A 口数据寄存器位——若 A 口为输入引脚，各位值与相应引脚上逻辑电平对应。若 A 口定义为输出，则读取的返回值为写入到寄存器中的最终值。写入的值被锁存到寄存器的所有位中。若引脚定义为输出，则逻辑电平被发送到 MCU 相应的引脚。 复位将 PTAD 所有位强制清 0，但这些 0 不通过相应的引脚驱动输出，因为复位将所有端口引脚配置为高阻态输入，并且禁用上拉 / 下拉。

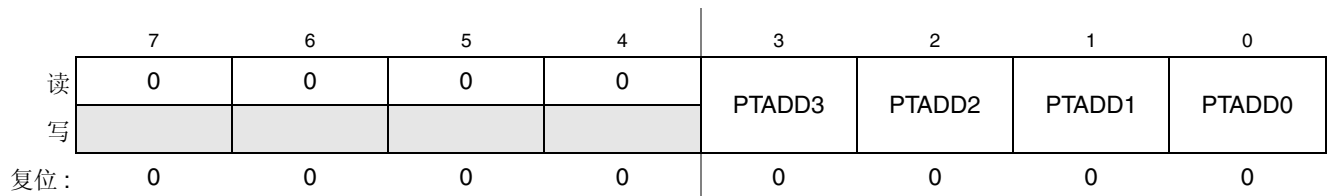


图 6-3. A 口数据方向寄存器 (PTADD)

表 6-2. PTADD 寄存器域描述

域	描述
3:0 PTADD[3:0]	A 口数据方向位——这些读写位控制 A 口引脚方向以及 PTAD 读取的值。 0 输入（输出驱动器禁用），读取引脚上的电平值。 1 A 口位 n 的输出驱动器使能，PTAD 读取 PTADn 中的内容。

6.2.2 B 口寄存器

B 口并行输入 / 输出功能通过本节描述的数据寄存器和数据方向寄存器控制。

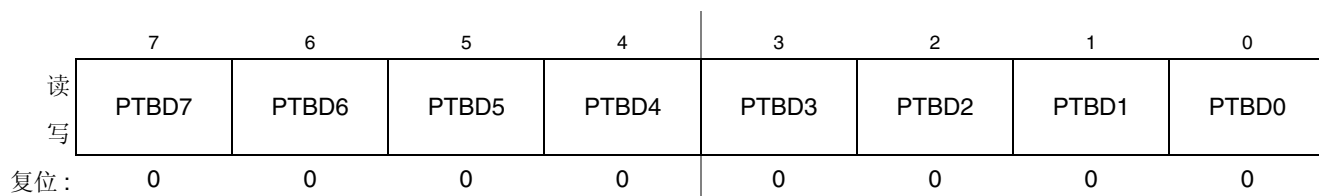


图 6-4. B 口数据寄存器（PTBD）

表 6-3. PTBD 寄存器域描述

域	描述
7:0 PTBD[7:0]	B 口数据寄存器位——若 B 口为输入引脚，各位值与相应引脚上逻辑电平对应。若 B 口定义为输出，则读取的返回值为写入到寄存器中的最终值。写入的值锁存到寄存器的所有位中。若引脚定义为输出，则逻辑电平被发送到 MCU 相应的引脚。 复位将 PTBD 所有位强制清 0，但这些 0 不通过相应的引脚驱动输出，因为复位将所有端口引脚配置为高阻态输入，并且禁用上拉 / 下拉。

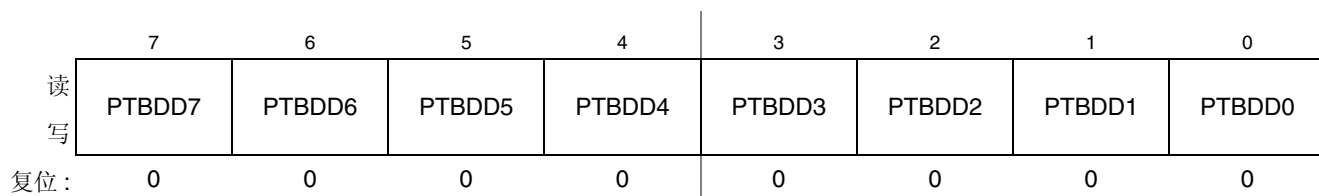


图 6-5. B 口数据方向寄存器（PTBDD）

表 6-4. PTBDD 寄存器域描述

域	描述
7:0 PTBDD[7:0]	B 口数据方向位——这些读写位控制 B 脚方向以及 PTBD 读取的值。 0 输入（输出驱动器禁用），读取引脚上的电平值。 1 B 口位 n 的输出驱动器使能，PTBD 读取 PTBDn 中的内容。

6.2.3 C 口寄存器

C 口并行输入 / 输出功能通过本节描述的数据和数据方向寄存器控制。

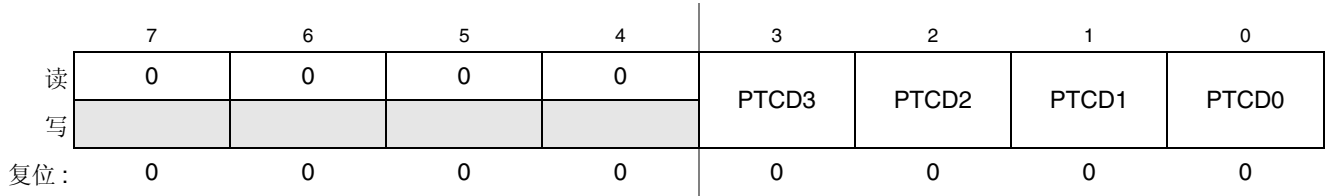


图 6-6. C 口数据寄存器 (PTCD)

表 6-5. PTCD 寄存器域描述

域	描述
3:0 PTCD[3:0]	C 口数据寄存器位——若 C 口为输入引脚，各位值与相应引脚上逻辑电平对应。若 C 口定义为输出，则读取的返回值为写入到寄存器中的最终值。写入的值锁存到寄存器的所有位中。若引脚定义为输出，则逻辑电平被发送到 MCU 相应的引脚。 复位将 PTCD 所有位强制清 0，但这些 0 不通过相应的引脚驱动输出，因为复位将所有端口引脚配置为高阻态输入，并且禁用上拉 / 下拉。

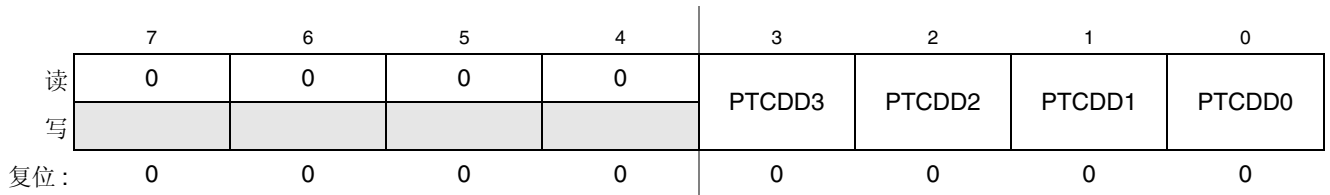


图 6-7. C 口数据方向寄存器 (PTCDD)

表 6-6. PTCDD 寄存器域描述

域	描述
3:0 PTCDD[3:0]	C 口数据方向位——这些读写位控制 C 口引脚方向以及 PTCD 读取的值。 0 输入（输出驱动器禁用），读取引脚上的电平值。 1 C 口位 n 的输出驱动器使能，PTCD 读取 PTCDn 中的内容。

6.3 引脚控制寄存器

这一节提供了与寄存器相关联的，用于引脚控制作用的并行 I/O 口的信息。

相关的表格在第四章，“存储器”。这节从概念上介绍了寄存器和及其控制位。飞思卡尔将提供一个头文件以便将这些名称转换成对应的绝对地址。

注意

通过设置驱动强度选择寄存器 (PTxDSn) 的相应位，对应输出引脚能够被设置成拥有高输出驱动强度。当被设置为高驱动强度时，引脚能够放出和吸纳更大的电流。即使每一个 I/O 引脚都能被选择为高驱动强度，也不能超过 MCU 的总电流源限制。驱动强度选择影响了 I/O 引脚的交流和直流行为。高驱动强度使得一个引脚能够以和低驱动强度相同的开关速度驱动一个更大的负载，从而使得引脚成为一个低负载。因此，使能引脚为高强度驱动将影响 EMC 干扰。

6.3.1 A 口引脚控制寄存器

A 口的相关引脚被这一节中所介绍的寄存器所控制。这些寄存器控制着引脚的上拉和下拉以及的 A 口引脚的边沿斜率，独立于并行 I/O 寄存器。

6.3.1.1 内部上拉下拉器件使能

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

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

图 6-8. A 口寄存器的内部上拉下拉电阻使能（PTAPE）

表 6-7. PTAPE 寄存器域描述

域	描述
5,3:0 PTAPE[5,3:0]	A 口内部上拉下拉电阻的使能位——这些控制位的每一位决定相应的 PTA 引脚的内部上拉下拉电阻是否使能。若 A 口引脚配置为输出，这些位将无效，同时内部上拉下拉电阻也被禁用。 0 A 口位 n 的内部上拉下拉电阻禁用。 1 A 口位 n 的内部上拉下拉电阻使能。

6.3.1.2 上拉 / 下拉控制

上拉 / 下拉控制用于通过设置 PTAPUD 的相应位使能上拉或下拉电阻。

	7	6	5	4	3	2	1	0
读	0	0	PTAPUD5	0	PTAPUD3	PTAPUD2	PTAPUD1	PTAPUD0
写								
复位：	0	0	0	0	0	0	0	0

图 6-9. A 口的上拉 / 下拉电阻控制（PTAPUD）

表 6-8. PTAPUD 寄存器域描述

域	描述
5,3:0 PTAPUD[5,3:0]	A 口的上拉 / 下拉电阻控制位——这些控制位中的每一位都决定了 PTA 的相关引脚是选择内部上拉还是下拉。目前的这些上拉 / 下拉电阻只能够通过设置 PTAPE 中的相关位而使能。 0 内部上拉电阻被 A 口的 n 位选中。 1 内部下拉电阻被 A 口的 n 位选中。

6.3.1.3 输出回转速率控制使能

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

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

图 6-10. A 口寄存器的回转速率使能（PTASE）

表 6-9. PTASE 寄存器域描述

域	描述
4:0 PTASE[4:0]	A 口输出回转速率使能位——这些控制位的每一位决定相应 PTA 引脚输出回转速率控制是否使能。若 A 口引脚配置为输入，这些位将无效。 0 A 口位 n 的输出回转速率控制禁用。 1 A 口位 n 的输出回转速率控制使能。

6.3.1.4 A 口驱动强度选择寄存器（PTADS）

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

图 6-11. 11 A 口输出驱动强度选择（PTADS）

表 6-10. PTADS 寄存器域描述

域	描述
4:0 PTADS[4:0]	A 口输出驱动强度选择位——PTA 的相关引脚选择低还是高输出驱动由这些控制位的每一个对应位决定。 0 A 口位 n 使能低输出驱动。 1 A 口位 n 使能高输出驱动。

6.3.2 B 口引脚控制寄存器

B 口的相关引脚被这一节中所介绍的寄存器所控制。这些寄存器控制着引脚的上拉和下拉以及的 B 口引脚的回转速率，独立于并行 I/O 寄存器。

6.3.2.1 内部上拉下拉电阻使能

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

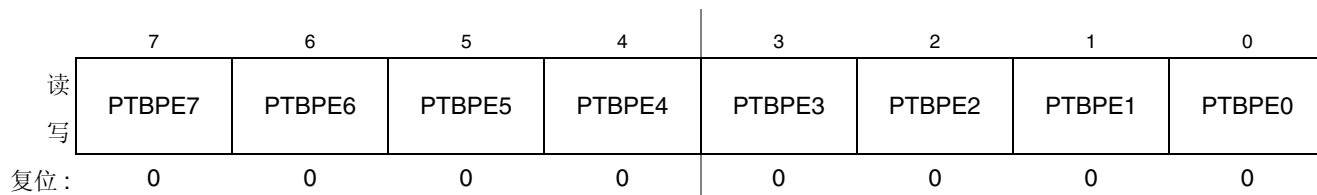


图 6-12. B 口寄存器的内部上拉下拉电阻使能 (PTBPE)

表 6-11. PTBPE 寄存器域描述

域	描述
7:0 PTBPE[7:0]	B 口的内部上拉下拉电阻的使能位——这些控制位的每一位决定相应的 PTB 引脚的内部上拉下拉电阻是否使能。若 B 口引脚配置为输出，这些位将无效，同时内部上拉电阻也被禁止。 0 B 口位 n 的内部上拉下拉电阻禁用。 1 B 口位 n 的内部上拉下拉电阻使能。

6.3.2.2 上拉 / 下拉控制

上拉 / 下拉控制用于通过设置 PTBPE 的相应位使能上拉或下拉电阻。

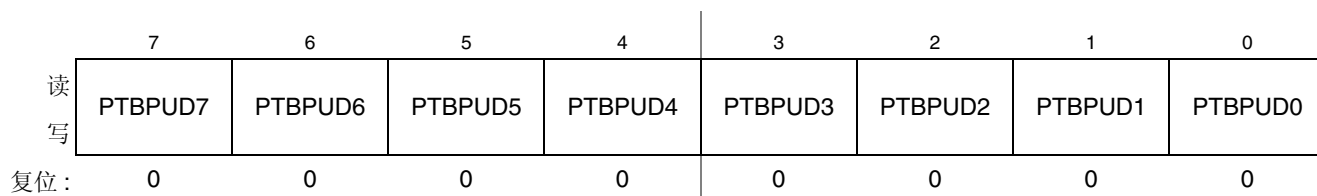


图 6-13. B 口的上拉 / 下拉电阻控制 (PTBPUD)

表 6-12. PTBPUD 寄存器域描述

域	描述
7:0 PTBPUD[7:0]	B 口的上拉 / 下拉电阻控制位——这些控制位中的每一位都决定了 PTB 的相关引脚是选择内部上拉还是下拉。目前的这些上拉 / 下拉电阻只能够通过设置 PTBPE 中的相关位而使能。 0 内部上拉电阻被 B 口的 n 位选中。 1 内部下拉电阻被 B 口的 n 位选中。

6.3.2.3 输出回转速率控制使能

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

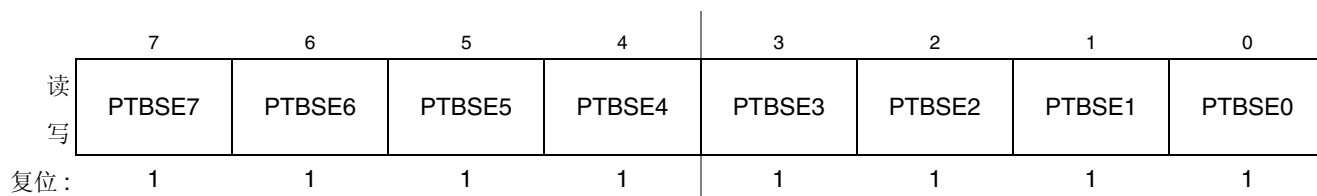


图 6-14. B 口寄存器的回转速率使能 (PTBSE)

表 6-13. PTBSE 寄存器域描述

域	描述
7:0 PTBSE[7:0]	B 口输出回转速率使能位——这些控制位的每一位决定相应 PTB 引脚输出回转速率控制是否使能。若 B 口引脚配置为输入，这些位将无效。 0 B 口位 n 的输出回转速率控制禁用。 1 B 口位 n 的输出回转速率控制使能。

6.3.2.4 B 口驱动强度选择寄存器 (PTBDS)

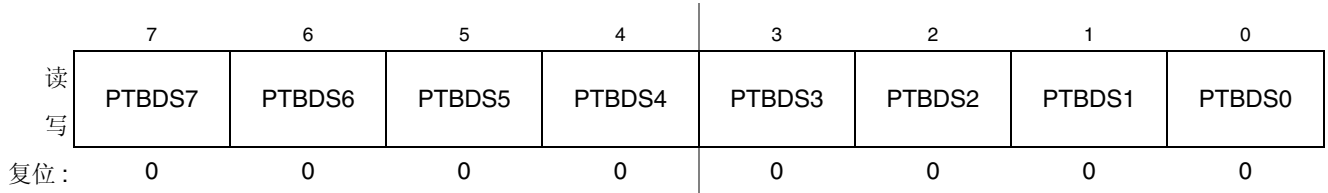


图 6-15. B 口输出驱动强度选择 (PTBDS)

表 6-14. PTBDS 寄存器域描述

域	描述
7:0 PTBDS[7:0]	B 口输出驱动强度选择位——PTB 的相关引脚选择低还是高输出驱动由这些控制位的每一个对应位决定。 0 B 口位 n 使能低输出驱动。 1 B 口位 n 使能高输出驱动。

6.3.3 C 口引脚控制寄存器

C 口的相关引脚被这一节中所介绍的寄存器所控制。这些寄存器控制着引脚的上拉和下拉以及的 C 口引脚的回转速率，独立于并行 I/O 寄存器。

6.3.3.1 内部上拉下拉电阻使能

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

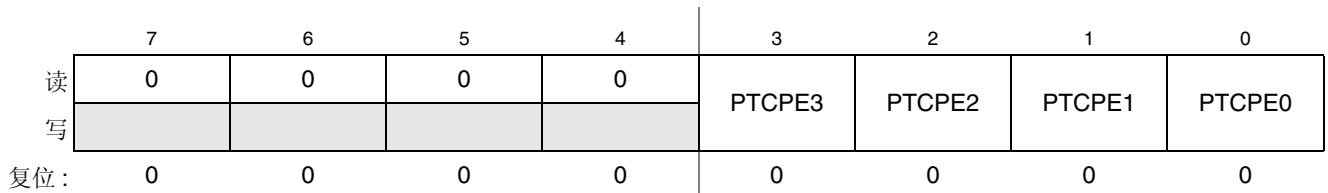


图 6-16. 内部上拉下拉电阻使能 (PTCPE)

表 6-15. PTCPE 寄存器域描述

域	描述
3:0 PTCPE[3:0]	C 口的内部上拉下拉电阻的使能位——这些控制位的每一位决定相应的 PTC 引脚的内部上拉下拉电阻是否使能。若 C 口引脚配置为输出，这些位将无效，同时内部上拉电阻也被禁止。 0 C 口位 n 的内部上拉下拉电阻禁用。 1 C 口位 n 的内部上拉下拉电阻使能。

6.3.3.2 上拉 / 下拉控制

上拉 / 下拉控制用于通过设置 PTCPE 的相应位使能上拉或下拉电阻。

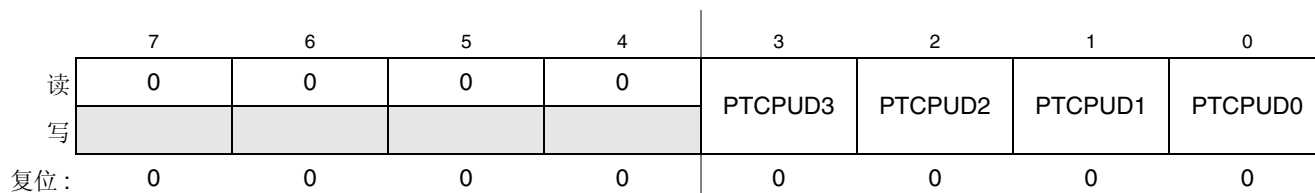


图 6-17. C 口的上拉 / 下拉电阻控制 (PTCPUD)

表 6-16. PTCUD 寄存器域描述

域	描述
3:0 PTCUD[3:0]	C 口的上拉 / 下拉电阻控制位——这些控制位中的每一位都决定了 PTC 的相关引脚是选择内部上拉还是下拉。目前的这些上拉 / 下拉电阻只能够通过设置 PTCPE 中的相关位而使能。 0 内部上拉电阻被 C 口的 n 位选中。 1 内部下拉电阻被 C 口的 n 位选中。

6.3.3.3 输出回转速率控制使能

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

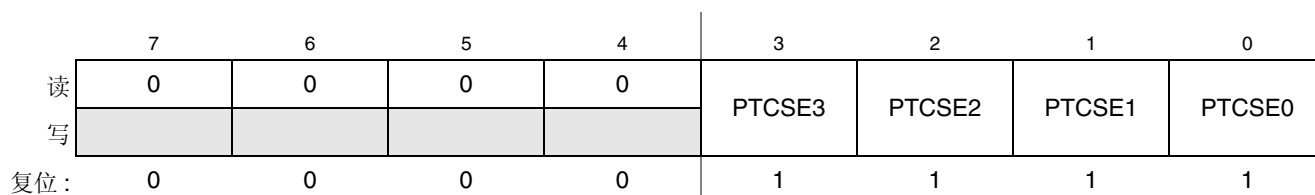


图 6-18. C 口寄存器的回转速率使能 (PTCSE)

表 6-17. PTCSE 寄存器域描述

域	描述
3:0 PTCSE[3:0]	C 口输出回转速率使能位——这些控制位的每一位决定相应 PTC 引脚输出回转速率控制是否使能。若 C 口引脚配置为输入，这些位将无效。 0 C 口位 n 的输出回转速率控制禁用。 1 C 口位 n 的输出回转速率控制使能。

6.3.3.4 C 口驱动强度选择寄存器 (PTCDS)

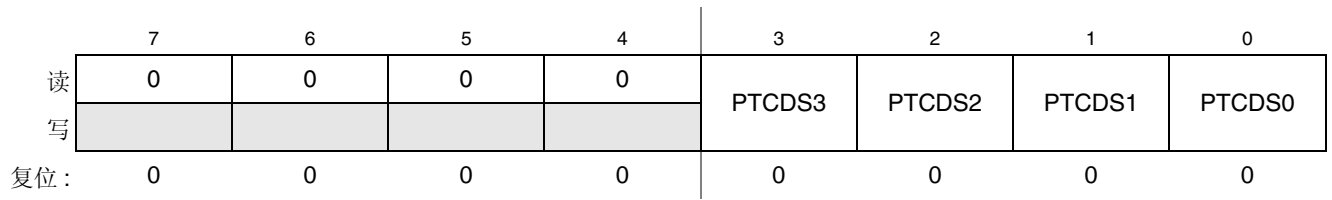


图 6-19. C 口输出驱动强度选择 (PTCDS)

表 6-18. PTCDS 寄存器域描述

域	描述
3:0 PTCDS[3:0]	C 口输出驱动强度选择位——PTC 的相关引脚选择低还是高输出驱动由这些控制位的每一个对应位决定。 0 C 口位 n 使能低输出驱动。 1 C 口位 n 使能高输出驱动。

第 7 章 中央处理器单元 (RS08CPUV1)

7.1 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，允许它在半导体处理技术快速发展步调保持一致。

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大小优化
- 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

7.2 程序设计器模型和 CPU 寄存器

图 7-1 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。

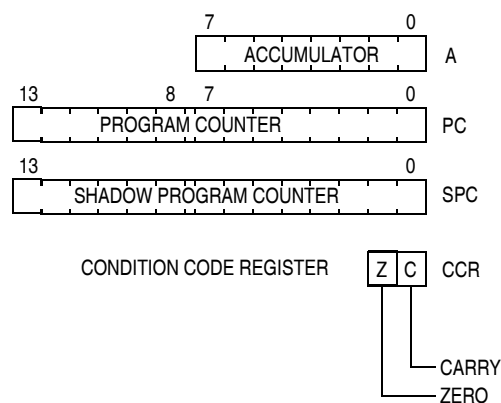


图 7-1. CPU 寄存器

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。

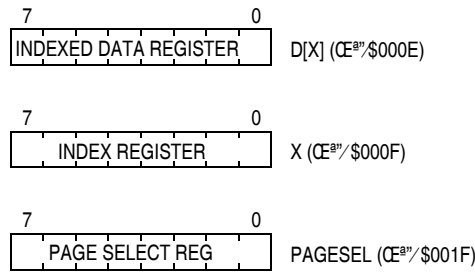


图 7-2. Memory Mapped Registers

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。

7.2.1 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。

7.2.2 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。

7.2.3 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。图 7-3 表明了 CCR 位和它们位的位置。

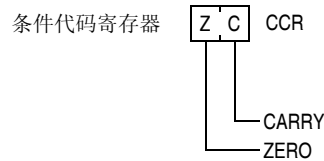


图 7-3. 条件代码寄存器 (CCR)

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。一个逻辑移位后，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。

7.2.4 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 7.3.8 节 变址寻址方式 (IX, 被伪指令执行)。”。

7.2.5 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 7.3.8 节 变址寻址方式 (IX, 被伪指令执行)。”。

7.2.6 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”。

7.3 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。

7.3.1 隐含寻址方式 (INH)

此寻址方式用于当 CPU 内在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。

7.3.2 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）。

7.3.3 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）。

7.3.4 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。

7.3.5 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。

7.3.6 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。

7.3.7 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。

7.3.8 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。

7.4 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。

7.4.1 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。

7.4.2 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。

7.4.3 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

7.4.4 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。

- 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

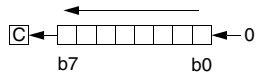
在源表格栏内的每个信息，是字面上的信息，它必须出现在精确显示的汇编源文件。头 3 到 5 个字母助记符总是一个文字表达式。所有逗号，磅符 (#) 和加号 (+) 字符。

- 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 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 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。
- 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

表 7-1. 指令集总结 (Sheet 1 of 5)

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	$A \leftarrow (A) + (M) + (C)$ $A \leftarrow (A) + (X) + (C)$	↕	↕	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	$A \leftarrow (A) + (M)$	↕	↕	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	$A \leftarrow (A) \& (M)$ $A \leftarrow (A) \& (X)$	↕	-	IMM DIR IX DIR	A4 B4 B4 B4	ii dd 0E 0F	2 3 3 3	
ASLA<st-super-script>(1)	Arithmetic Shift Left		↕	↕	INH	48		1	
BCC rel	Branch if Carry Bit Clear	$PC \leftarrow (PC) + \$0002 + rel$, if (C) = 0	-	-	REL	34	rr	3	
BCLR n,opr8a	Clear Bit n in Memory	$M_n \leftarrow 0$	-	-	DIR (b0)	11	dd	5	
BCLR n,D[X]						13	dd	5	
						15	dd	5	
						17	dd	5	
						19	dd	5	
						1B	dd	5	
						1D	0E	5	
						1F	0E	5	
						11	0E	5	
						IX (b0)	13	0E	5
						IX (b1)	15	0E	5
						IX (b2)	17	0E	5
						IX (b3)	19	0F	5
						IX (b4)	1B	0F	5
						IX (b5)	1D	0F	5
	IX (b6)	1F	0F	5					
IX (b7)	11	0F	5						
BCLR n,X	DIR (b0)	1F	0F	5					
	DIR (b1)	11	0F	5					
	DIR (b2)	13	0F	5					
	DIR (b3)	15	0F	5					
	DIR (b4)	17	0F	5					
	DIR (b5)	19	0F	5					
	DIR (b6)	1B	0F	5					
	DIR (b7)	1D	0F	5					
	DIR (b7)	1F	0F	5					
BCS rel	Branch if Carry Bit Set (Same as BLO)	$PC \leftarrow (PC) + \$0002 + rel$, if (C) = 1	-	-	REL	35	rr	3	
BEQ rel	Branch if Equal	$PC \leftarrow (PC) + \$0002 + rel$, if (Z) = 1	-	-	REL	37	rr	3	
BGND	Background	Enter Background Debug Mode	-	-	INH	BF		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。

表 7-1. 指令集总结 (Sheet 2 of 5)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles	
			Z	C					
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	-	↑	DIR (b0)	01	dd rr	5	
BRCLR <i>n,D[X],rel</i>					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		
BRCLR <i>n,X,rel</i>					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	
					BRSET <i>n,opr8a,rel</i>	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						
BRSET <i>n,D[X],rel</i>	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					
	BRSET <i>n,X,rel</i>	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					

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。

表 7-1. 指令集总结 (Sheet 3 of 5)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles			
			Z	C							
BSET <i>n,opr8a</i>	Set Bit <i>n</i> in Memory	$M_n \leftarrow 1$	-	-	DIR (b0)	10	dd	5			
BSET <i>n,D[X]</i>			DIR (b1)	12	dd	5					
			DIR (b2)	14	dd	5					
			DIR (b3)	16	dd	5					
			DIR (b4)	18	dd	5					
			DIR (b5)	1A	dd	5					
			DIR (b6)	1C	dd	5					
			DIR (b7)	1E	dd	5					
BSET <i>n,X</i>			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					
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	41	ii rr	4
						DIR	31	dd rr	5		
IX	31	0E rr				5					
DIR	31	0F rr	5								
CLC	Clear Carry Bit	$C \leftarrow 0$	-	0	INH	38		1			
CLR <i>opr8a</i> CLR <i>opr5a</i> CLR <i>,X</i> <st-superscript>(1) CLRA CLR <i>X</i> <st-superscript>(1)	Clear	$M \leftarrow \$00$ $A \leftarrow \$00$ $X \leftarrow \$00$	1	-	DIR	3F	dd	3			
SRT					SRT	8x / 9x		2			
					IX	8E		2			
					INH	4F		1			
					INH	8F		2			
CMP # <i>opr8i</i> CMP <i>opr8a</i> CMP <i>,X</i> <st-superscript>(1) CMP <i>X</i> <st-superscript>(1)	Compare Accumulator with Memory	$(A) - (M)$ $(A) - (X)$	\updownarrow	\updownarrow	IMM	A1	ii	2			
					DIR	B1	dd	3			
					IX	B1	0E	3			
					INH	B1	0F	3			
COMA	Complement (One's Complement)	$A \leftarrow (\bar{A})$	\updownarrow	1	INH	43		1			
DBNZ <i>opr8a,rel</i> DBNZ <i>,X,rel</i> <st-superscript>(1) DBNZA <i>rel</i> DBNZ <i>rel</i> <st-superscript>(1)	Decrement and Branch if Not Zero	$A \leftarrow (A) - \$01$ or $M \leftarrow (M) - \$01$ $PC \leftarrow (PC) + \$0003 + rel$ if (result) $\neq 0$ for DBNZ direct $PC \leftarrow (PC) + \$0002 + rel$ if (result) $\neq 0$ for DBNZA $X \leftarrow (X) - \$01$ $PC \leftarrow (PC) + \$0003 + rel$ if (result) $\neq 0$	-	-	DIR	3B	dd rr	7			
					IX	3B	0E rr	4			
					INH	4B	rr	7			
					INH	3B	0F rr	7			

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。

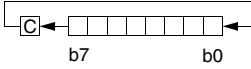
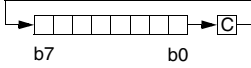
表 7-1. 指令集总结 (Sheet 4 of 5)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
DEC <i>opr8a</i> DEC <i>opr4a</i> DEC ,X <st-super-script>(1) DECA DEC X	Decrement	$M \leftarrow (M) - \$01$ $A \leftarrow (A) - \$01$ $X \leftarrow (X) - \$01$	↑	-	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	$A \leftarrow (A \oplus M)$ $A \leftarrow (A \oplus X)$	↑	-	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	$M \leftarrow (M) + \$01$ $A \leftarrow (A) + \$01$ $X \leftarrow (X) + \$01$	↑	-	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	$A \leftarrow (M)$	↑	-	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	$\$0F \leftarrow (M)$	↑	-	IMD DIR IX	3E 4E 4E	ii 0F dd 0F 0E 0E	4 5 5
LSLA	Logical Shift Left		↑	↑	INH	48		1
LSRA	Logical Shift Right		↑	↑	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	$(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$	↑	-	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	$A \leftarrow (A) (M)$ $A \leftarrow (A) (X)$	↑	-	IMM DIR IX DIR	AA BA BA BA	ii dd 0E 0F	2 3 3 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。

表 7-1. 指令集总结 (Sheet 5 of 5)

Source Form	Description	Operation	Effect on CCR		Address Mode	Opcode	Operand	Cycles
			Z	C				
ROLA	Rotate Left through Carry		↑	↑	INH	49		1
RORA	Rotate Right through Carry		↑	↑	INH	46		1
RTS	Return from Subroutine	Pull PC from shadow PC	–	–	INH	BE		3
SBC #opr8i SBC opr8a SBC ,X <st-super-script>(1) SBC X	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$ $A \leftarrow (A) - (X) - (C)$	↑	↑	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 opr8a STA opr5a STA ,X <st-super-script>(1) STA X	Store Accumulator in Memory	$M \leftarrow (A)$	↑	–	DIR SRT IX SRT	B7 Ex / Fx EE EF	dd	3 2 2 2
STX opr8a <st-super-script>(1)	Store Index Register in Memory	$M \leftarrow (X)$	↑	–	DIR	4E	0F dd	5
STOP	Put MCU into stop mode		–	–	INH	AE		2+
SUB #opr8i SUB opr8a SUB opr4a SUB ,X <st-super-script>(1) SUB X	Subtract	$A \leftarrow (A) - (M)$ $A \leftarrow (A) - (X)$	↑	↑	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	$X \leftarrow (A)$	↑	–	INH	EF		2
TST opr8a <st-super-script>(1) TSTA <st-super-script>(1) TST ,X <st-super-script>(1) TSTX <st-super-script>(1)	Test for Zero	$(M) - \$00$ $(A) - \$00$ $(X) - \$00$	↑	–	DD INH IX INH	4E AA 4E 4E	dd dd 00 0E 0E 0F 0F	5 2 5 5
TXA<st-super-script>(1)	Transfer X to A	$A \leftarrow (X)$	↑	–	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。

表 7-2. 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
0	BRSET0 ⁵ ₃ DIR	BSET0 ⁵ ₂ DIR	INC ⁴ ₁ TNY	BRA ³ ₂ REL	Gray box	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	SUB ² ₂ IMM	SUB ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
1	BRCLR0 ⁵ ₃ DIR	BCLR0 ⁵ ₂ DIR	INC ⁴ ₁ TNY	CBEQ ⁵ ₃ DIR	CBEQA ⁴ ₃ IMM	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	CMP ² ₂ IMM	CMP ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
2	BRSET1 ⁵ ₃ DIR	BSET1 ⁵ ₂ DIR	INC ⁴ ₁ TNY	Gray box	SLA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	SBC ² ₂ IMM	SBC ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
3	BRCLR1 ⁵ ₃ DIR	BCLR1 ⁵ ₂ DIR	INC ⁴ ₁ TNY	Gray box	COMA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	Gray box	Gray box	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
4	BRSET2 ⁵ ₃ DIR	BSET2 ⁵ ₂ DIR	INC ⁴ ₁ TNY	BCC ³ ₂ REL	LSRA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	AND ² ₂ IMM	AND ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
5	BRCLR2 ⁵ ₃ DIR	BCLR2 ⁵ ₂ DIR	INC ⁴ ₁ TNY	BCS ³ ₂ REL	SHA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	Gray box	Gray box	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
6	BRSET3 ⁵ ₃ DIR	BSET3 ⁵ ₂ DIR	INC ⁴ ₁ TNY	BNE ³ ₂ REL	RORA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	LDA ² ₂ IMM	LDA ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
7	BRCLR3 ⁵ ₃ DIR	BCLR3 ⁵ ₂ DIR	INC ⁴ ₁ TNY	BEQ ³ ₂ REL	Gray box	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	Gray box	STA ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
8	BRSET4 ⁵ ₃ DIR	BSET4 ⁵ ₂ DIR	INC ⁴ ₁ TNY	CLC ¹ ₁ INH	LSLA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	EOR ² ₂ IMM	EOR ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
9	BRCLR4 ⁵ ₃ DIR	BCLR4 ⁵ ₂ DIR	INC ⁴ ₁ TNY	SEC ¹ ₁ INH	ROLA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	ADC ² ₂ IMM	ADC ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
A	BRSET5 ⁵ ₃ DIR	BSET5 ⁵ ₂ DIR	INC ⁴ ₁ TNY	DEC ⁵ ₂ DIR	DECA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	ORA ² ₂ IMM	ORA ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
B	BRCLR5 ⁵ ₃ DIR	BCLR5 ⁵ ₂ DIR	INC ⁴ ₁ TNY	DBNZ ⁶ ₃ DIR	DBNZA ⁴ ₂ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	ADD ² ₂ IMM	ADD ³ ₂ DIR	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
C	BRSET6 ⁵ ₃ DIR	BSET6 ⁵ ₂ DIR	INC ⁴ ₁ TNY	INC ⁵ ₂ DIR	INCA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	NOP ¹ ₁ INH	JMP ⁴ ₃ EXT	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
D	BRCLR6 ⁵ ₃ DIR	BCLR6 ⁵ ₂ DIR	INC ⁴ ₁ TNY	Gray box	Gray box	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	BSR ³ ₂ REL	JSR ⁴ ₃ EXT	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
E	BRSET7 ⁵ ₃ DIR	BSET7 ⁵ ₂ DIR	INC ⁴ ₁ TNY	MOV ⁴ ₃ IMD	MOV ⁵ ₃ DD	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	STOP ²⁺ ₁ INH	RTS ³ ₁ INH	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ SRT
F	BRCLR7 ⁵ ₃ DIR	BCLR7 ⁵ ₂ DIR	INC ⁴ ₁ TNY	CLR ³ ₂ DIR	CLRA ¹ ₁ INH	DEC ⁴ ₁ TNY	ADD ³ ₁ TNY	SUB ³ ₁ TNY	CLR ² ₁ SRT	CLR ² ₁ SRT	WAIT ²⁺ ₁ INH	BGND ⁵⁺ ₁ INH	LDA ³ ₁ SRT	LDA ³ ₁ SRT	STA ² ₁ SRT	STA ² ₁ 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 SUB³₂ DIR

RS08 Cycles
 Opcode Mnemonic
 Number of Bytes /
 Addressing Mode

第 8 章

模拟比较器 (RS08ACMPV1)

8.1 介绍

模拟比较器模块 (ACMP) 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)。

图 8-1 显示 SC667028 方框图中增强亮度的 ACMP。

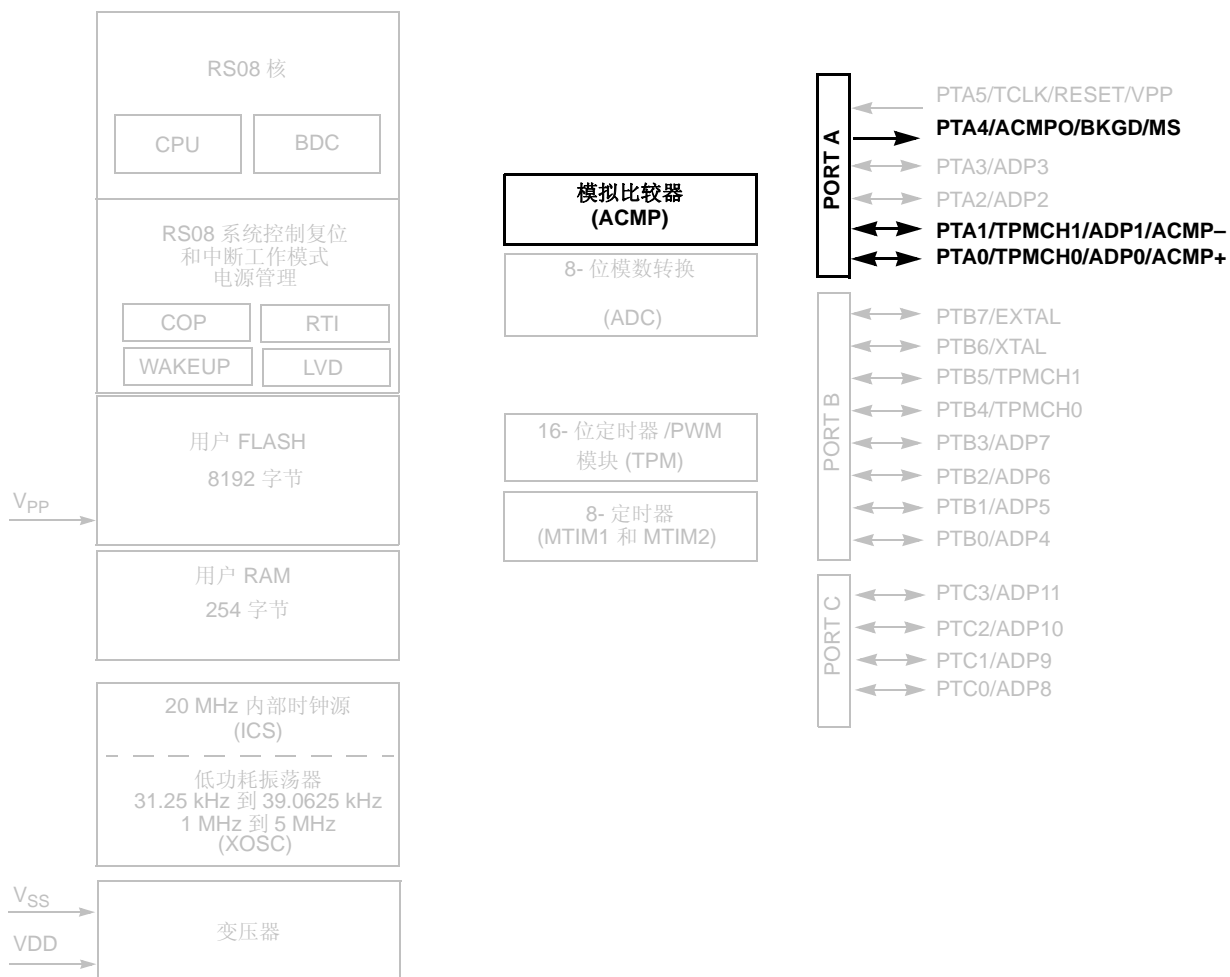


图 8-1. 显亮 ACMP 模块和引脚的 SC667028 模块结构图

8.1.1 特性

ACMP 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可视, ACMPxO
- 可工作在停止模式

8.1.2 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。

8.1.2.1 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。

8.1.2.2 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。

8.1.2.3 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。

8.1.3 方框图

模拟比较器方框图如图 8-2 所示:

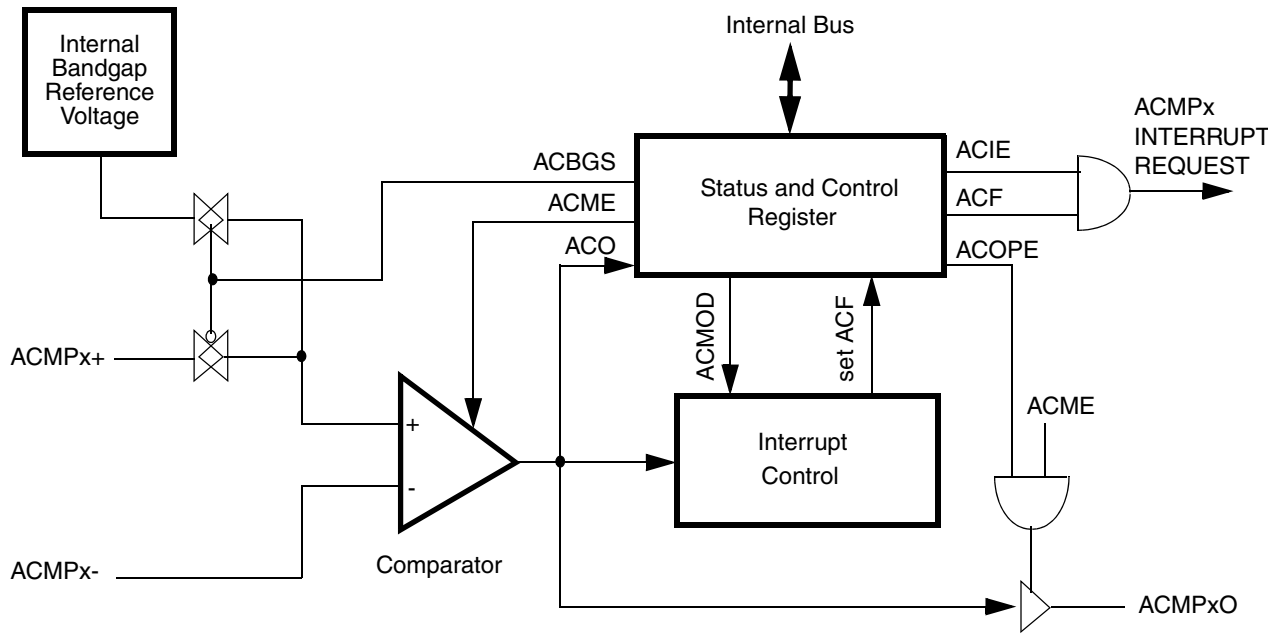


图 8-2. 模拟比较器 (ACMP) 方框图

8.2 外部信号描述

ACMP 有两个模拟输入脚，ACMPx+ 和 ACMPx-，并且有一个数字输出管脚，ACMPxO。每个输入管脚能接受一个 MCU 全工作电压范围的输入电压。如图 8-2 所示，ACMPx- 管脚连接到比较器的负相输入端；假如 ACBGS=0，ACMPx+ 管脚连接到比较器的正相输入端；ACMPxO 管脚能被用于驱动一个外部管脚。

ACMP 信号属性 如表 8-1 所示。

表 8-1. 信号属性

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

8.3 寄存器定义

ACMPx 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。

8.3.1 ACMPx 状态和控制寄存器 (ACMPxSC)

ACMPxSC 包含状态标志和控制位用于使能和配置 ACMP。

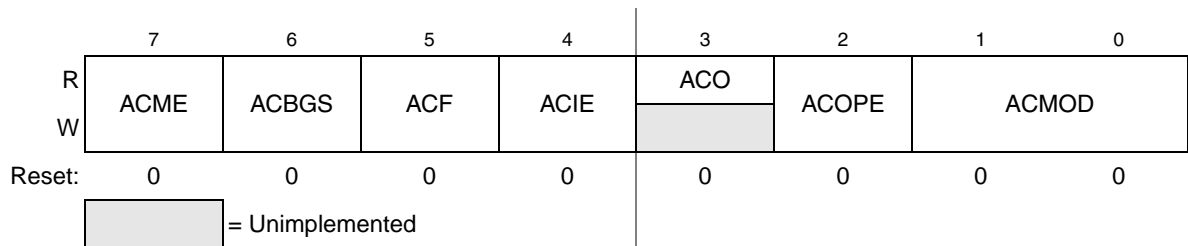


图 8-3. ACMPx 状态和控制寄存器 (ACMPxSC)

表 8-2. ACMPxSC 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 ACMPx+ 管脚作为模拟比较器正输入 0 外部管脚 ACMPx+ 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

表 8-2. ACMPxSC 域描述 (continued)

域	描述
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, ACMPxO。假如 ACMP 是活动的 (ACME=1), ACOPE 将只控制此管脚。 0 在 ACMPxO 管脚, 模拟输出不允许; 1 模拟输出被驱动在 ACMPxO。
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

8.4 功能描述

模拟比较器能用于比较应用在 ACMPx+ 和 ACMPx- 的两个模拟输入电压; 或用于比较一个应用在 ACMPx 械哪 D 馐淙氩繆购鸵桓瞿诘看 恫慰嫉繆埂 ACBGS 用于采用选择待隙参考电压还是 ACMPx+ 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 ACMPxO 管脚。

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 V_{DD} 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。为了实现最佳性能, 推荐芯片进入等待 / 停止模式必须避免 ACMP 测量与临近管脚绑在一起。

第 9 章

AD 转换器 (RS08ADCV1)

9.1 概述

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

注意

SC667028 的 ADC 只提供 8 位，不支持 8 位和 12 位。本章中关于 8 位和 12 位的描述适用于飞思卡尔提供的其他 8 位微控制器。

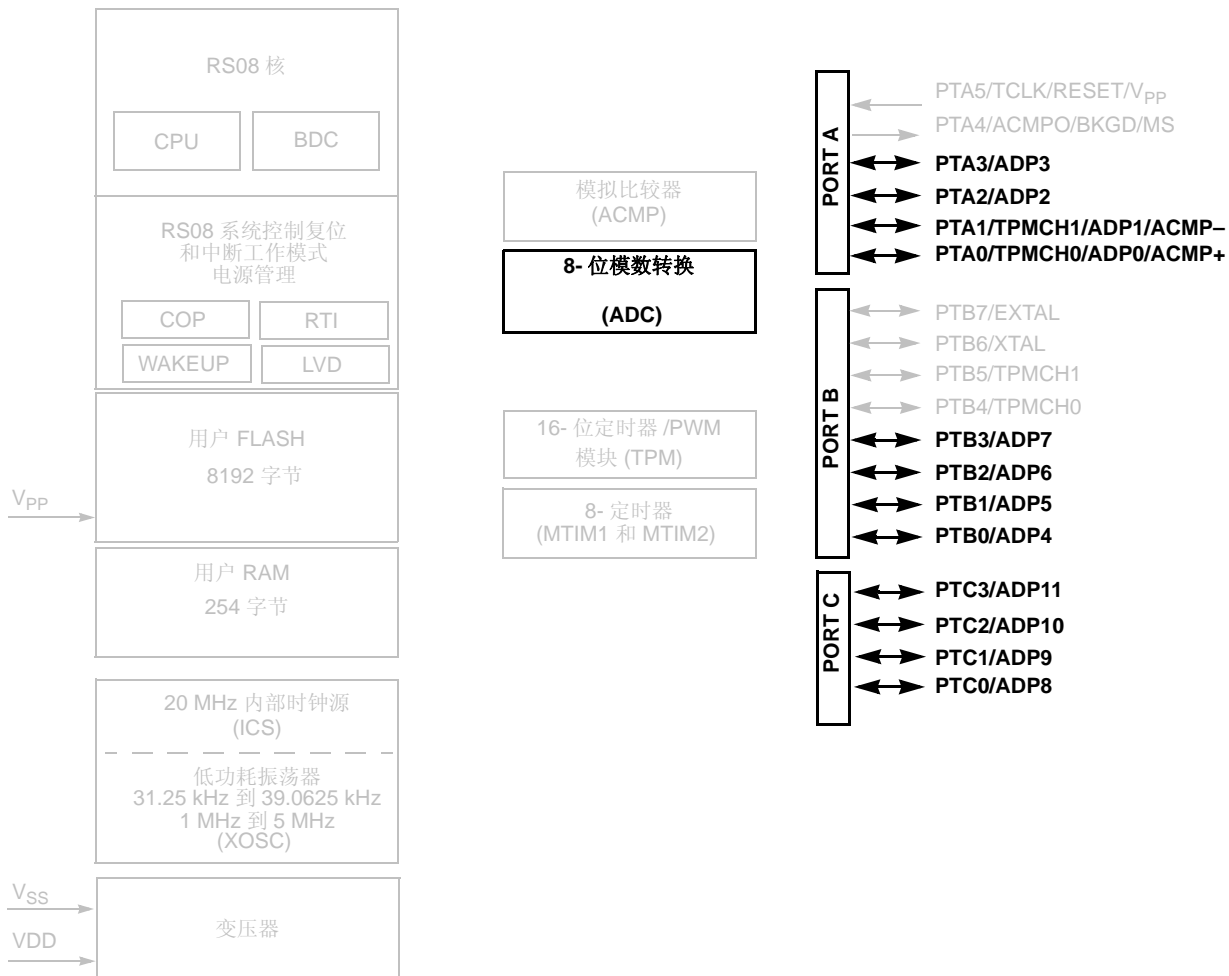


图 9-1. 显亮 ADC 模块和引脚的 SC667028 模块结构图

9.1.1 模块配置

本节为配置 SC667028 的 ADC 模块提供了详细的信息。

9.1.1.1 模拟量和参考电压之间的关系

ADC 模块的 V_{DDAD} 和 V_{REFH} 连接到 V_{DD} 引脚， V_{SSAD} 和 V_{REFL} 连接到 V_{SS} 引脚。

9.1.1.2 可选时钟

进行模数转换时，有 3 种时钟源可以选择：MCU 总线时钟、二分频的总线时钟和 AD 模块内的异步时钟 (ADACK)。可选时钟 (ALTCLK) 在 SC667028 系列 MCU 中没有实现。

9.1.1.3 硬件触发器

ADC 硬件触发器，ADHWT，是实时中断计数器 (RTI) 的输出。RTI 计数器可以使用 IC SERCLK 或 RTI 时钟内标称的 1kHz 时钟源。

输入时钟频率和 RTIS 位决定了 RTI 的周期。RTI 计数器是一个自由运行计数器，按照 RTIS 位决定的 RTI 率产生一个溢出。当允许 ADC 硬件触发器时，若 RTI 计数器溢出，则初始化一个转换。

MCU 处于运行，停止模式时，RTI 可以配置为产生一个硬件触发。

NOTE

为了得到快速的 RTI 硬件触发，RTI 时钟源必须是一个高频率的外部时钟源。

9.1.1.4 模拟引脚使能

SC667028 的 ADC 模块只包含两个模拟引脚使能寄存器：APCTL1 和 APCTL2。

SC667028 的 ADC 模块的通道分配如表 9-1 所示。保留未使用的通道转换为未知的值。连接到一个 I/O 口的通道有一个引脚控制位，显示如下表

表 9-1. ADC 通道分频

ADCH	通道	输入	引脚控制
00000	AD0	PTA0/ADP0	ADPC0
00001	AD1	PTA1/ADP1	ADPC1
00010	AD2	PTA2/ADP2	ADPC2
00011	AD3	PTA3/ADP3	ADPC3
00100	AD4	PTB0/ADP4	N/A
00101	AD5	PTB1/ADP5	N/A
00110	AD6	PTB2/ADP6	N/A
00111	AD7	PTB3/ADP7	N/A
01000	AD8	PTC0/ADP8	ADPC8
01001	AD9	PTC1/ADP9	ADPC9
01010	AD10	PTC2/ADP10	ADPC10
01011	AD11	PTC3/ADP11	ADPC11
01100-11100	保留	N/A	N/A
11101	V_{REFH}	V_{DD}	N/A
11110	V_{REFL}	V_{SS}	N/A
11111	模块被关闭	无	N/A

9.1.1.5 低功耗模式操作

ADC 可以在停止模式下运行，但是需 SPMSC1 中的 LVDSE 和 LVDE 位被置位 SC667028。

9.1.2 特性

ADC 模块特点包括:

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

9.1.3 框图

图 9-2 提供了 ADC 模块的框图。

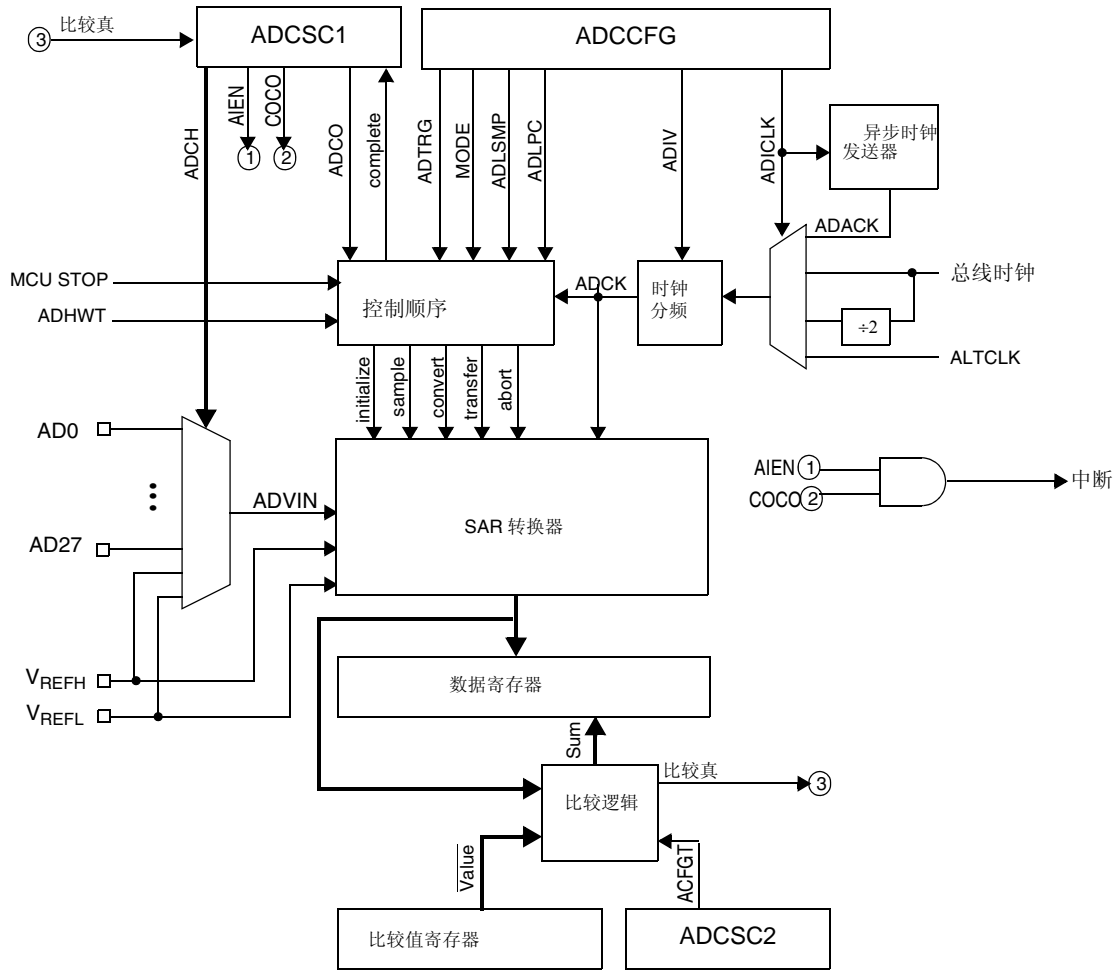


图 9-2.2 ADC 框图

9.2 外部信号描述

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

表 9-2. 信号属性

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

9.2.1 模拟电源 (V_{DDAD})

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

9.2.2 模拟地 (V_{SSAD})

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

9.2.3 参考高电压 (V_{REFH})

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

9.2.4 参考低电压 (V_{REFL})

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

9.2.5 模拟通道输入 (ADx)

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

9.3 寄存器定义

这些内存映像寄存器控制和管理 ADC 的操作:

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

9.3.1 状态和控制寄存器 1 (ADCSC1)

本节描述 ADC 状态和控制寄存器 (ADCSC1) 的功能。写 ADCSC1 可以终止当前的转换并初始化一个新的转换 (如果 ADCH 等于一个不是全 1 的值)。

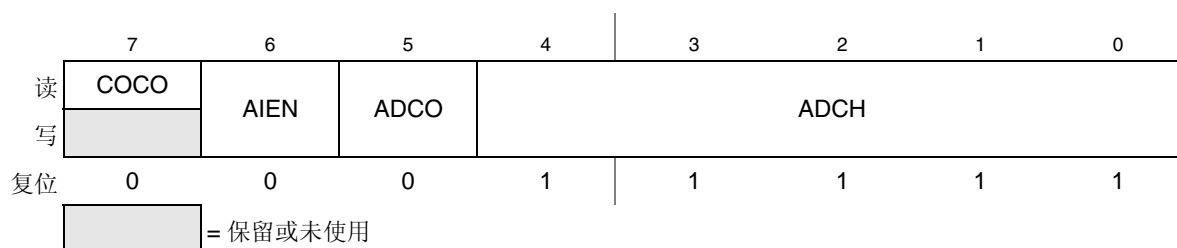


图 9-3. 状态和控制寄存器 (ADCSC1)

表 9-3. ADCSC1 寄存器域描述

域	描述
7 COCO	转换完成标志——COCO 标志是一个只读位。当比较功能禁止 (ACFE=0) 时, 每次转换完成时置位。当比较功能允许 (ACFE=1) 时, 转换完成后, 只要比较结果为真, 则 COCO 置位。只要写 ADCSC1 或读 ADCRL, 该位清零。 0 转换未完成。 1 转换完成。
6 AIEN	中断允许——AIEN 用于使能转换完成中断。当 AIEN 为高, COCO 置位时, 确认一个中断。 0 禁止转换完成中断。 1 允许转换完成中断。
5 ADCO	连续转换使能——ADCO 用于使能连续转换。 0 当选择软件触发中断时, 写 ADCSC1 后开始一个转换。当选择硬件触发中断时, 确认了 ADHWT 后开始一个中断。 1 当选择软件触发中断时, 写 ADCSC1 后初始化连续转换。当选择硬件触发中断时, 连续转换被 ADHWT 事件初始化。
4:0 ADCH	输入通道选择——ADCH 包括 5 位, 用于选择输入通道中的一个。输入通道在表 9-4 中描述。 当通道选择位设置为全 1 时, 逐次逼近转换器子系统关闭。这个特点允许禁止 ADC 和从所有的源中孤立输入通道。

表 9-4. 输入通道选择

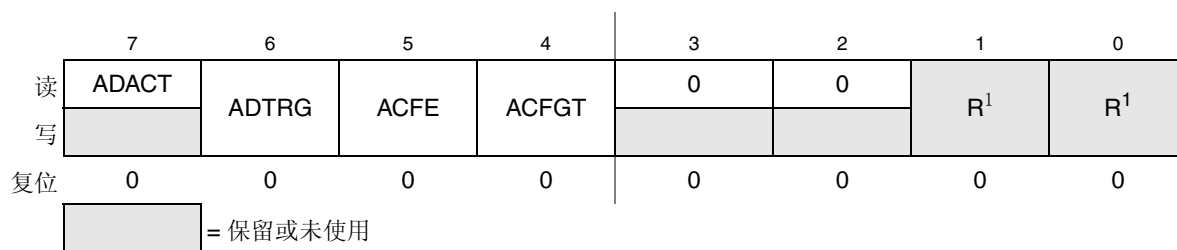
ADCH	输入选择
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7
01000	AD8
01001	AD9
01010	AD10
01011	AD11
01100	AD12
01101	AD13
01110	AD14
01111	AD15
10000	AD16
10001	AD17

表 9-4. 输入通道选择 (continued)

ADCH	输入选择
10010	AD18
10011	AD19
10100	AD20
10101	AD21
10110	AD22
10111	AD23
11000	AD24
11001	AD25
11010	AD26
11011	AD27
11100	保留
11101	V _{REFH}
11110	V _{REFL}
11111	模块被禁止

9.3.2 状态和控制寄存器 2 (ADCSC2)

ADCSC2 寄存器用于控制比较功能，转换触发和 ADC 模块的转换行为。



¹ 位 1 和位 0 为保留位，必须写为 0。

图 9-4. 状态和控制寄存器 (ADCSC2)

表 9-5. ADCSC2 寄存器域描述

域	描述
7 ADACT	转换行为——ADACT 表示转换正在进行中。当初始化转换时，ADACT 置位；当转换完成或终止时，ADACT 清零。 0 转换未进行。 1 转换处理中。
6 ADTRG	转换触发选择——ADTRG 用于选择初始化转换的触发的类型。两种触发类型可选：软件触发和硬件触发。选择软件触发，写 ADCSC1 后初始化一个转换。选择硬件触发，确认了 ADHWT 输入后初始化转换。 0 选择软件触发。 1 需安装硬件触发。

表 9-5. ADCSC2 寄存器域描述

5 ACFE	比较功能使能——ACFE 用于使能比较功能。 0 禁止比较功能。 1 允许比较功能。
4 ACFGT	比较功能更大使能——当监控的输入的转换结果大于或等于比较结果时，ACFGT 用于设置比较功能的触发。当监控的输入的转换结果小于比较结果时，比较功能默认触发。 0 当输入小于比较电平时，比较触发。 1 当输入大于或等于比较电平时，比较触发。

9.3.3 数据高结果寄存器 (ADCRH)

ADCRH 包含 10 位转换结果的高 2 位。当设置为 8 位转换时，ADR8 和 ADR9 都等于 0。每次转换完成，除非自动比较被允许而且不满足比较结果，ADCRH 将被更新。在 10 位数据模式中，读 ADCRH 将暂时禁止下一次转换，直到读取了 ADCRL 中的内容。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRL 的互锁。在这种情况下，MODE 位被改变，ADCRH 中的任何数据都无效。

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


 = 保留或未使用

图 9-5. 数据高结果寄存器 (ADCRH)

9.3.4 数据低结果寄存器 (ADCRL)

ADCRL 包含 10 位转换结果的低 8 位，一个 8 位转换的所有 8 位。每次转换完成，这个寄存器都被更新，除非自动比较被允许而且不满足比较结果。在 10 位数据模式中，读 ADCRH 将暂时禁止下一次转换，直到读取了 ADCRL 中的内容。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRH 的互锁。在这种情况下，MODE 位被改变，ADCRL 中的任何数据都无效。

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


 = 保留或未使用

图 9-6. 数据低结果寄存器 (ADCRL)

9.3.5 比较值高寄存器 (ADCCVH)

该寄存器包含了 10 位比较值的高 2 位。当允许比较功能时，这些位和 10 位模式中的转换结果的高 2 位比较。在 8 位操作中，ADCCVH 在比较过程中不使用。

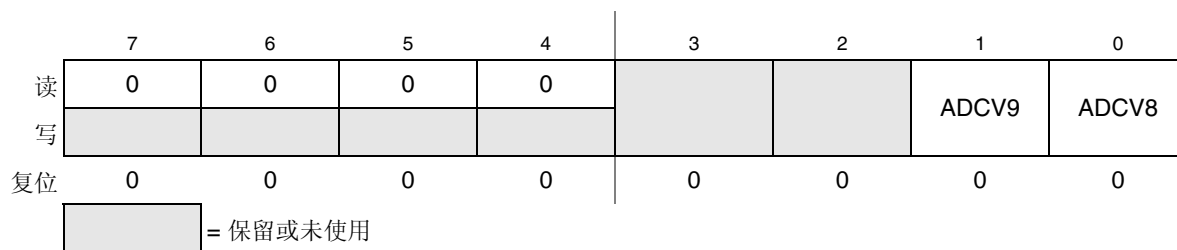


图 9-7. 比较值高寄存器 (ADCCVH)

9.3.6 比较值低寄存器 (ADCCVL)

该寄存器包含了 10 位比较值的低 8 位，或者 8 位比较值的所有 8 位。在 10 位或 8 位模式中，ADC7:ADCV0 转换结果的低 8 位比较。

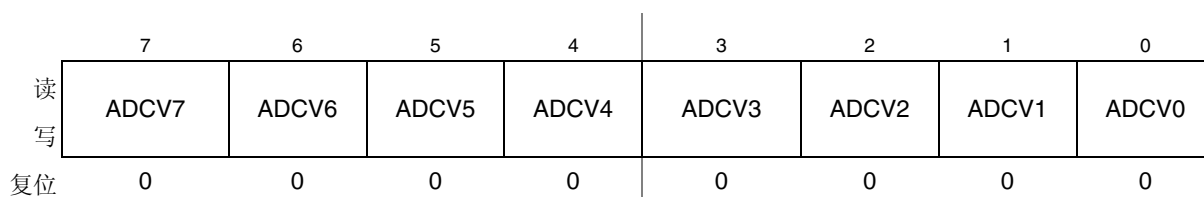


图 9-8. 比较值低寄存器 (ADCCVL)

9.3.7 配置寄存器 (ADCCFG)

ADCCFG 用于选择操作模式，时钟源，时钟分频和低功耗或长采样时间的设置。

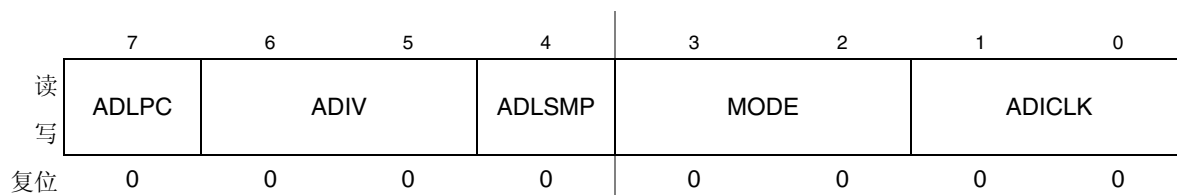


图 9-9. 配置寄存器 (ADCCFG)

表 9-6. ADCCFG 寄存器域描述

域	描述
7 ADLPC	低功耗配置——ADLPC 控制逐次渐进转换器的速度和功耗配置。当不需要更高采样速率时，可以优化功耗。 0 高速配置。 1 低功耗配置：以最大化时钟速率的代价降低功耗。
6:5 ADIV	时钟分频选择——ADIV 选择 ADC 使用的分频因子，产生内部时钟 ADCK。表 9-7 描述了时钟配置。
4 ADLSMP	长采样时间配置——ADLSMP 选择长和短采样时间。这可以调整采样周期，使在更高阻抗的输入下也能得到精确的采样，也可以在低阻抗时最大化转换速度。如果连续采样允许而且不需要高输出斜率，更长的采样时间可以用于更低的总功耗。 0 短采样时间。 1 长采样时间。

表 9-6. ADCCFG 寄存器域描述

3:2 MODE	转换模式选择——MODE 位用于选择 8 位或 10 位操作。参见表 9-8。
1:0 ADICLK	输入时钟选择——ADICLK 选择产生内部时钟 ADCK 的输入时钟源。参见表 9-9。

表 9-7. 时钟分频选择

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

表 9-8. 转换模式

模式	模式描述
00	8 位转换 (N=8)
01	保留
10	10 位转换 (N=10)
11	保留

表 9-9. 输入时钟选择

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

9.3.8 引脚控制 1 寄存器 (APCTL1)

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

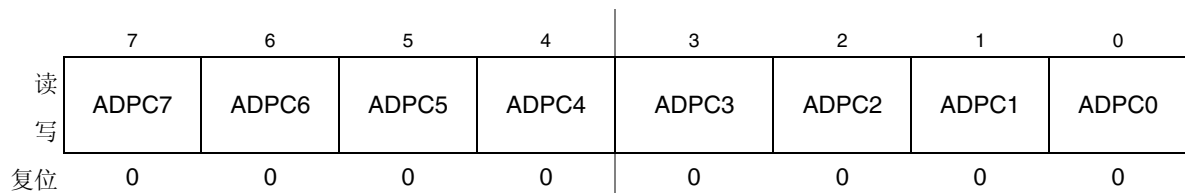


图 9-10. 引脚控制 1 寄存器 (APCTL1)

表 9-10. APCTL1 寄存器域描述

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

9.3.9 引脚控制 2 寄存器 (APCTL2)

APCTL2 用于控制和 ADC 模块的通道 8-15 相关的位。

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

图 9-11. 引脚控制 2 寄存器 (APCTL2)

表 9-11. APCTL2 寄存器域描述

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

9.3.10 引脚控制 3 寄存器 (APCTL3)

APCTL3 用于控制和 ADC 模块的通道 16-23 相关的位。

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

图 9-12. 引脚控制 3 寄存器 (APCTL3)

表 9-12. APCTL3 寄存器域描述

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

9.4 功能描述

当复位或 ADCH 位全高时，禁止 ADC 模块。当转换完成而且另一个转换还未初始化时，该模块空闲。空闲时，模块处于最小功耗状态。

ADC 可以通过软件选择任何一个通道进行模数转换。选择的通道电压可以被逐次渐进算法转换成 11 位数字的结果。在 8 位模式中，选择的通道电压可以被逐次渐进算法转换成 9 位数字的结果。

当转换完成，结果放在数据寄存器中（ADCRH 和 ADCRL）。在 10 位模式中，结果四舍五入成 10 位放在 ADCRH 和 ADCRL 中。在 8 位模式中，结果四舍五入成 8 位放在 ADCRL 中。转换完成标志置 1 并且如果允许转换完成中断（AIEN=1），产生一个中断。

ADC 模块能够自动比较转换结果和比较寄存器的内容。通过置位 ACFE 位，允许比较功能。该功能和任何转换模式和设置协力完成操作。

9.4.1 时钟选择和分频控制

可以选择 4 个时钟源中的一个作为 ADC 模块的时钟源。这个时钟源除以一个设置值就可以产生转换器的输入时钟（ADCK）。时钟选择下面源中的一个，由 ADICLK 位决定。

- 总线时钟，等于软件执行时的频率。这是复位后的默认值。

- 总线时钟 /2。对于更高的时钟，可以允许最大除以 16。
- ALTCLK，由 MCU 定义（参见模块一节的介绍）
- 异步时钟（ADACK）——该时钟由 ADC 模块内部的时钟源产生。当选择这个时钟源时，若 MCU 处于等待或 stop3 模式时，该时钟仍有效，允许在这些模式中以更低的噪音操作来进行转换。

无论选择哪个时钟，它的频率都必须低于规定的 ADCK 频率范围。如果可用的时钟太慢，根据规定 ADC 将不会工作。如果可用的时钟太快，时钟必须分频到适当的频率。分频因子由 ADIV 位决定，可以除以 1, 2, 4, 8。

9.4.2 输入选择和引脚控制

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

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

9.4.3 硬件触发

ADC 模块有一个可选的异步硬件转换触发器，ADHWT，当 ADTRG 位置位时，它被允许。关于该 MCU 的 ADHWT 源的具体细节参考模块介绍。

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

9.4.4 转换控制

可以使用 8 位或 10 位模式转换，由 MODE 位决定。一个软件或硬件触发可以初始化转换。另外，ADC 模块可以设置为低功耗操作，长采样时间，连续采样，自动比较转换值和软件决定的比较值。

9.4.4.1 初始化转换

满足以下条件，即初始化转换：

- 如果选择软件触发操作，在写 ADCSC1 之后（ADCH 不是全 1）。
- 如果选择硬件触发操作，在一个硬件触发（ADHWT）事件之后。
- 当允许连续转换时，在将数据传到数据寄存器之后。

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

9.4.4.2 完成转换

当转换的结果传到数据结果寄存器，ADCRH 和 ADCRL 后，转换完成。通过置位 COCO 表示。如果 AIEN 是高，在 COCO 置位时会产生一个中断。

在 10 位模式中，如果数据正在被读（ADCRH 已经被读但是 ADCRL 还未被读），闭锁机制保护新的数据不会重写在 ADCRH 和 ADCRL 中以前的数据。当锁有效，数据传送被锁，COCO 不能置位，新的数据丢

失。在允许比较功能的单个转换的情况下，并且比较条件为假，锁没有作用，ADC 操作被终止。在其他情况下，当数据传送被锁，除非 ADCO 的状态（允许单个或连续转换），另一个转换被初始化。

如果允许单个转换，闭锁机制可能导致丢弃几个转换并且额外的功耗。为了避免这种情况，在初始化一个单个转换后，数据寄存器直到转换完成才能读。

9.4.4.3 终止转换

下列情况发生时，任何正在出来的转换都会终止：

- 写 ADCSC1 发生（如果 ADCH 不是全 1，当前的转换被取消并开始了新的转换，）。
- 写 ADCSC2、ADCCFG、ADCCVH 或 ADCCVL 发生。这表明转换模式发生改变，因此当前的转换无效。
- MCU 复位。
- MCU 进入停止模式并且禁止 ADACK。

当一个转换终止，数据寄存器（ADCRH 和 ADCRL）的内容不会改变，而是上次转换后完成后的传送的值。在因复位导致的转换终止情况中，ADCRH 和 ADCRL 返回到它们的复位值。

9.4.4.4 电源控制

直到初始化一个转换，ADC 模块都保持空闲。如果 ADACK 被选作转换时钟源，ADACK 时钟产生器也被允许。

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

9.4.4.5 总转换时间

总转换时间依赖于抽样时间（由 ADLSMP 决定），MCU 总线频率，转换模式（8 位或 10 位），转换时钟的频率（ f_{ADCD} ）。模块有效后，输入的采样开始。ADLSMP 用于选择长或短采样时间。当转换完成，转换器和输入通道隔离，用逐次渐进算法将模拟信号转换成数字值。转换算法完成后，转换结果传送到 ADCRH 和 ADCRL。

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

表 9-13 中总结了不同条件下的最大的总转换时间。

表 9-13. 不同控制条件的总转换时间

转换类型	ADICLK	ADLSMP	最大总转换时间
单个或第一个连续转换 8 位	0x, 100	0	20ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 10 位	0x, 100	0	23ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 8 位	0x, 100	1	40ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 10 位	0x, 100	1	40ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 8 位	11	0	5s+20ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	0	5s+23ADCK +5 总线时钟周期

表 9-13. 不同控制条件的总转换时间 (continued)

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

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

$$\text{转换时间} = \frac{23 \text{ ADCK 周期}}{8 \text{ MHz} / 1} + \frac{5 \text{ 总线周期}}{8 \text{ MHz}} = 3.5 \mu\text{s} \quad \text{等式 9-1}$$

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

注意

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

9.4.5 自动比较功能

比较功能可以设置为检测上限或下限。采样和转换输入后，结果和比较值 (ADCCVH 和 ADCCVL) 的补数相加。比较上限时 (ACFGT = 1)，如果结果大于或等于比较值，COCO 置位。比较下限时 (ACFGT=0)，如果结果小于比较值，COCO 置位。转换结果和比较值的补数相加后产生的值传送到 ADCRH 和 ADCRL。

注意

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

9.4.6 MCU 等待模式操作

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

处于等待模式时，总线时钟，总线时钟的一半和 ADACK 可以作为转换时钟源。在等待模式时，ALTCLK 作为转换时钟源使用是由该 MCU 的 ALTCLK 的定义决定的。参考该 MCU 中模块说明中关于 ALTCLK 说明的信息。

9.4.7 MCU stop3 模式操作

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

9.4.7.1 禁止 ADACK 的 stop3 模式

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

9.4.7.2 允许 ADACK 的 stop3 模式

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

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

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

注意

ADC 可能将系统从低功率停止中唤醒，导致 MCU 开始强烈的运行电平电流而没有产生一个系统电平中断。为了避免这种情况，当进入 stop3 模式并继续 ADC 转换时，软件应该确保数据传输闭锁机制（在第 9.4.4.2 节 完成转换）清零。

9.4.8 MCU stop1 和 stop2 模式操作

当 MCU 进入 stop1 或 stop2 模式时，自动禁止 ADC 模块。从 stop1 或 stop2 退出时，所有的模块寄存器存放的是复位值。因此从 stop1 或 stop2 退出时，模块必须被重新使能和重新配置。

9.5 初始化信息

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

注意

十六进制的前缀是 0x，二进制的前缀是 %，十进制没有前缀。

9.5.1 ADC 模块初始化举例

9.5.1.1 初始化顺序

在 ADC 模块执行转换操作前，必须初始化。典型的初始化顺序是：

1. 更新配置寄存器 (ADCCFG) 选择输入时钟源和产生内部时钟 (ADCK) 的分频因子。这个寄存器也可以用来选择采样时间和低功耗配置。
2. 更新状态和控制寄存器 2 (ADCSC2) 选择转换触发器 (硬件或软件) 和比较功能选项 (如果允许)。
3. 更新状态和控制寄存器 1 (ADCSC1) 可以选择转换是否是连续的还是仅一次完成，并且允许或禁止转换完成中断。选择哪路输入通道完成转换也是在这里操作。

9.5.1.2 伪代码举例

在该例中，ADC 模块将提供允许中断产生一个单个 10 位转换，低功耗，输入通道 1 上的长采样时间，内部的 ADCK 时钟将由总线时钟除以 1 得到

AD 转换器 (RS08ADC1)

ADCCFG=0x98 (%10011000)

Bit 7	ADLPC	1	配置为低功耗 (降低最大的时钟速率)
Bit 6:5	ADIV	00	设置 ADCK 为输入时钟 /1
Bit 4	ADLSMP	1	设置位长采样时间
Bit 3:2	MODE	10	设置位 10 位转换模式
Bit 1:0	ADICLK	00	选择总线时钟作为输入时钟

ADCSC2=0x00 (%00000000)

Bit 7	ADACT	0	标志表示转换是否在处理中
Bit 6	ADTRG	0	选择软件触发器
Bit 5	ACFE	0	禁止比较功能
Bit 4	ACFGT	0	在该例中未使用
Bit 3:2		00	未实现或保留, 读为 0
Bit 1:0		00	保留为 Freescale 使用; 写为 0

ADCSC1=0x41 (%01000001)

Bit 7	COCO	0	只读标志当转换完成时置位
Bit 6	AIEN	1	转换完成中断使能
Bit 5	ADCO	0	仅一次转换 (禁止连续转换)
Bit 4:0	ADCH	00001	选择通道 1 作为 ADC 输入通道

ADCRH/L = 0xxx

保存转换结果。在低字节前读高字节, 所以转换数据不会被下一次转换的数据重写。

ADCCVH/L = 0xxx

当允许比较功能时, 保存比较值

APCTL1=0x02

禁止 AD1 引脚 I/O 控制。其他 AD 引脚仍然位通用 I/O 引脚

APCTL2=0x00

其他 AD 引脚仍然时通用 I/O 引脚。

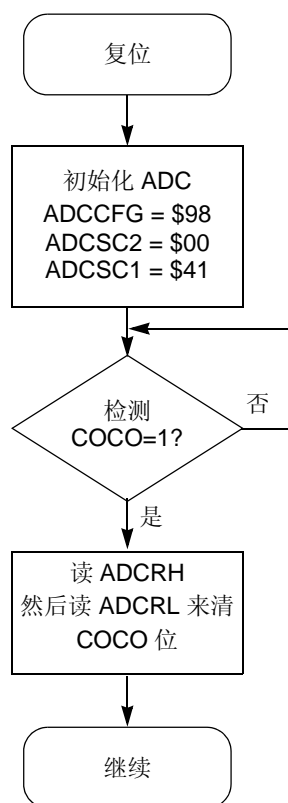


图 9-13. 举例的初始化流程

9.6 应用信息

本节包含了在应用中使用 ADC 模块的信息。ADC 被设计集成了一个微控制器可以使用在需要 A/D 转换器的嵌入式控制应用中。

9.6.1 外部引脚和安排

下面讨论了和 ADC 模块相关的外部引脚和如何最好的使用它们。

9.6.1.1 模拟电源引脚

ADC 模块有电源和地引脚 (V_{DDAD} 和 V_{SSAD})，在一些设备中有独立的引脚。在其他设备， V_{SSAD} 和 MCU V_{SS} 复用相同的引脚。在一些设备中， V_{SSAD} 和 V_{DDAD} 共享数字供电引脚。在这些情况中，有独立模拟供电，和相应的数字电源绑定在相同的引脚。所以这两个电源在一定程度上保持隔离。

当作为独立的引脚时， V_{DDAD} 和 V_{SSAD} 必须和相应的 MCU 数字电源 (V_{DD} 和 V_{SS}) 连接相同的电压，谨慎布线避免干扰，旁路电容离封装尽可能近。

在模拟和数字电源单独供电时的情况，这两个电源的接地连接必须是 V_{SSAD} 引脚。如果可能这应该是这两个电源的唯一接地连接。 V_{SSAD} 接单个的地位置。

9.6.1.2 模拟参考引脚

除了模拟电源，ADC 模块连接了两个参考电压输入。高参考电源是 V_{REFH} ，在有些设备中，可能和 V_{DDAD} 复用相同的引脚。地参考电压是 V_{REFL} ，在有些设备中和 V_{SSAD} 复用相同的引脚。

当作为单独引脚时， V_{REFH} 可能和 V_{DDAD} 连到相同的电压。或可能是由外部得到（介于 V_{DDAD} 最小值和 V_{DDAD} 之间， V_{REFH} 绝对不能大于 V_{DDAD} ）。当作为单独引脚时， V_{REFL} 可能和 V_{DDAD} 连到相同的电压。 V_{REFH} 和 V_{REFL} 必须谨慎布线防止最大干扰度和旁路电容离封装尽可能近。

在每次逐次渐进步骤，电流尖峰组成的交流（AC）电流通过 V_{REFH} 和 V_{REFL} 循环位电容阵列提供电荷。满足这个电流要求的最好的外部元件是 $0.1\mu\text{F}$ 的电容（高频特性）。这个电容连接到 V_{REFH} 和 V_{REFL} 之间，离封装尽可能近。不推荐使用电阻，因为电流导致电压泄露，这可能导致转换错误。该路径上的电磁感应应最小化。

9.6.1.3 模拟输入引脚

外部模拟输入通常和 MCU 设备的 I/O 引脚复用。通过置位引脚控制寄存器的相应位可以禁止引脚 I/O 控制。相应的引脚控制寄存器位没有置位，可以进行转换操作。当引脚作为模拟输入时，推荐置位引脚控制寄存器位。这避免了连接问题，因为输入缓冲区处于高阻抗状态并且禁止上拉电阻。而且，当输入既不是 V_{DD} 也不是 V_{SS} ，输入缓冲吸收直流（DC）电流。置位引脚控制位，所有的引脚作为模拟输入，可以达到最低的操作电路。

试验数据表明，当存在噪音或源阻抗高时，模拟输入上的电容可以提高性能。试验 $0.01\mu\text{F}$ 的电容（有高频特性）完全可以满足。这些电容并不是在所有的情况下都需要，但是它们必须放在离封装尽可能近的地方，作为 V_{SSA} 的参考。

为正确转换，输入电压必须在 V_{REFH} 和 V_{REFL} 之间。如果输入等于或大于 V_{REFH} 转换电路把信号转换成 $\$3\text{FF}$ （共 10 位表示）或 $\$FF$ （共 8 位表示），如果输入等于或小于 V_{REFH} ，转换电路把信号转换成 $\$000$ ，在 V_{REFH} 和 V_{REFL} 之间的输入电压是线性转换。当采样电容正在充电时，将会有有一个和 V_{REFL} 相关的短暂电流。当 ADLSMP 低，将会采样以 ADCK 为源的 3.5 个周期，当 ADLSMP 高，则是 23.5 个周期。

为了减少因电流进入而引起的正确度减小，连接到模拟输入的引脚在转换期间不应该传输。

9.6.2 错误源

A/D 转换中存在几种错误源。它们在该节的后面讨论。

9.6.2.1 采样错误

为正确转换，输入必须被采样足够长时间才能达到合适的精度。如果最大输入电阻 $7\text{ k}\Omega$ 和输入电容 5.5 pF ，外部模拟源（RAS）的电阻小于 $5\text{ k}\Omega$ ，则采样 $1/4\text{LSB}$ （10 位精度）可以在最小的采样窗口完成（3.5 周期， 8MHz 最大 ADCK 频率）。

更高电阻或更高精度的采样可以通过置位 ADLSM （增加采样窗口到 23.5 个周期）或通过减少 ADCK 频率来增加采样时间。

9.6.2.2 引脚漏电流误差

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

9.6.2.3 噪音误差

在采样或转换过程中产生的系统噪音会影响转换的正确性。只要满足下面指定的条件才能保证 ADC 采样的正确性。

- V_{REFH} 和 V_{REFL} 之间有一个 $0.1 \mu\text{F}$ 的低内阻电容。
- V_{DDAD} 和 V_{SSAD} 之间有一个 $0.1 \mu\text{F}$ 的低内阻电容。
- 如果电源使用感应隔离，一个 $1 \mu\text{F}$ 的电容放在 V_{DDAD} 和 V_{SSAD} 之间。
- V_{SSAD} (和 V_{REFL} ，如果连接) 连接到 V_{SS} (连着地平面的点)。
- 初始化 (硬件触发转换) 前或刚初始化 (软件或硬件触发转换) ADC 转换，MCU 处于等待或 stop3 状态。
 - 对于软件触发转换，用 WAIT 或 STOP 指令写 ADCSC1 后。
 - 对于 stop3 模式操作，选择 ADACK 作为时钟源。在 stop3 模式的操作减少 V_{DD} 噪音但是因为停止复原增加了有效转换时间。
- MCU 处于转换时，没有 I/O 选择，输入或输出。

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

- 在选择的输入通道和 V_{REFH} 或 V_{REFL} 之间放置一个 $0.01 \mu\text{F}$ 的电容 (CAS) (这将增加噪音问题但是影响基于外部的模拟源电阻的采样率)。
- 求多次转换模拟值的平均。需要四次采样减少 1LSB 的影响以及一次误差。
- 通过关闭异步时钟 (ADACK) 和求平均，减少同步噪音的影响。和 ADCK 同步的噪音无法达到平均数。

9.6.2.4 编码宽度和量化误差

ADC 可以将输入的线性值量化成 1024 个块 (在 10 位模式)。每块有相同的高度 (1 个代码) 和宽度。宽度定义为 $d\text{leta}$ ，在一个代码到下一个之间。N 位转换器 (N 是 8 或 10) 的理想代码宽度，定义为 1LSB，为：

$$1\text{LSB} = (V_{REFH} - V_{REFL}) / 2^N \quad \text{等式 9-2}$$

数字化结果存在固有的量化误差。对于 8 位或 10 位转换。用两点的中点表示电压，代码被转化。因此在 8 位或 10 位模式中，量化误差 $\pm 1/2\text{LSB}$ 。因此，第一个转化的编码宽度 (\$000) 只有 $1/2\text{LSB}$ 并且最后一个转化的宽度时 1.5LSB 。

9.6.2.5 线性误差

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

- 归零误差 (EVS) (也称作偏移量) ——指第一个转换的实际编码宽度和理想编码宽度的不同 ($1/2\text{LSB}$)。注意如果第一个转换是 \$001，则采用了然后实际的 \$001 编码宽度和理想情况 (1LSB) 的不同。
- 满标误差 (EFS) ——指最后一个转换的实际编码宽度和理想编码宽度的不同 (1.5LSB)。注意如果最后一个转换是 \$3FE，采用了实际的 \$001 编码宽度和理想情况 (1LSB) 的不同。
- 微分非线性 (DNL) ——指所有转换中实际编码宽度和理想编码宽度最大误差。

- 积分非线性 (INL) —— 指 DNL 总和所能达到的最大值。更简单的, 所有编码中, 编码所给的实际的转换电压和和相应的理想电压的最大误差。
- 总非校准误差 (TUE) —— 这个错误定义为实际转换函数和理想线性转换函数的不同, 因此包含所有形式的错误。

9.6.2.6 编码抖动、非单调性和遗编码

ADC 易受三种特殊形式的错误影响。它们是代码抖动, 非单调性, 遗编码。

代码抖动, 在某一点时, 当重复采样时, 一个确定的输入电压转化成两个值中的一个。理想的, 当输入电压比转换电压只小很少时, 转换器产生更低的编码 (反之亦然)。然而, 即使很小的系统噪音也会导致转换器对于转换电压周围一定范围的输入电压不确定 (在两个编码之间)。这个范围通常是 $1/2\text{LSB}$, 并且随噪音而增加。这个误差可以通过重复采样和对结果求均值来减少。另外在 [9.6.2.3 节 噪音误差](#) 节中的技巧可以减少该误差。

非单调性可以定义为, 除了代码抖动, 转换器可能转换一个较高电压位较低编码。遗编码是那些对于任何输入都不会转换的值。

在 8 位或 10 位模式中, ADC 将确保单调并且不遗失编码。

第 10 章

内部时钟源 (RS08ICSV1)

10.1 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。

图 10-1 列出了高亮为 ICS 模块的 SC667028 方框图。

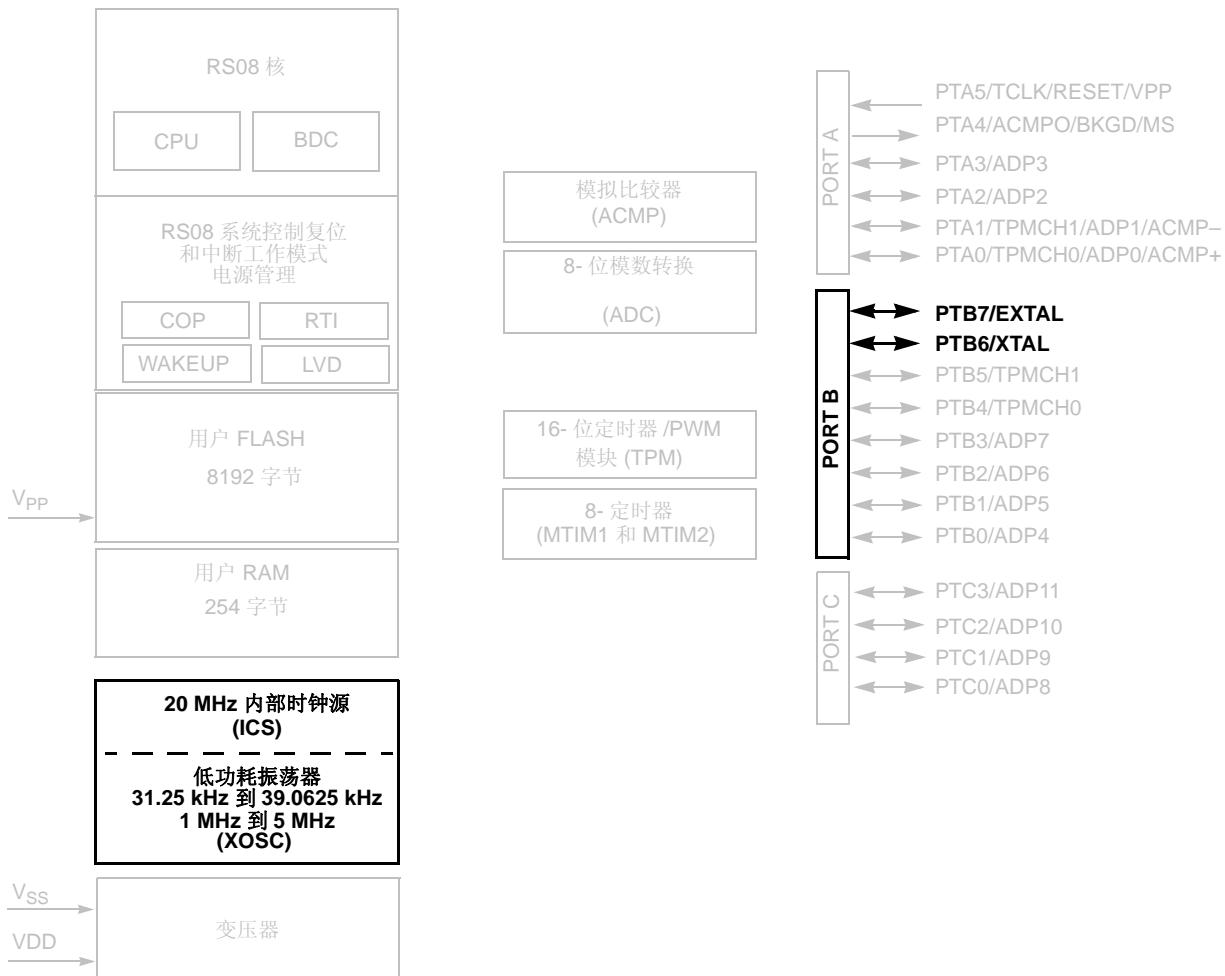


图 10-1. SC667028 方框图高亮的 ICS 模块和管脚

10.1.1 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

10.1.2 工作模式

ICS 有 4 种工作模式: FEI、FBI、FBILP 和 stop.

10.1.2.1 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

10.1.2.2 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。

10.1.2.3 FLL 旁路内部低功耗 (FBILP)

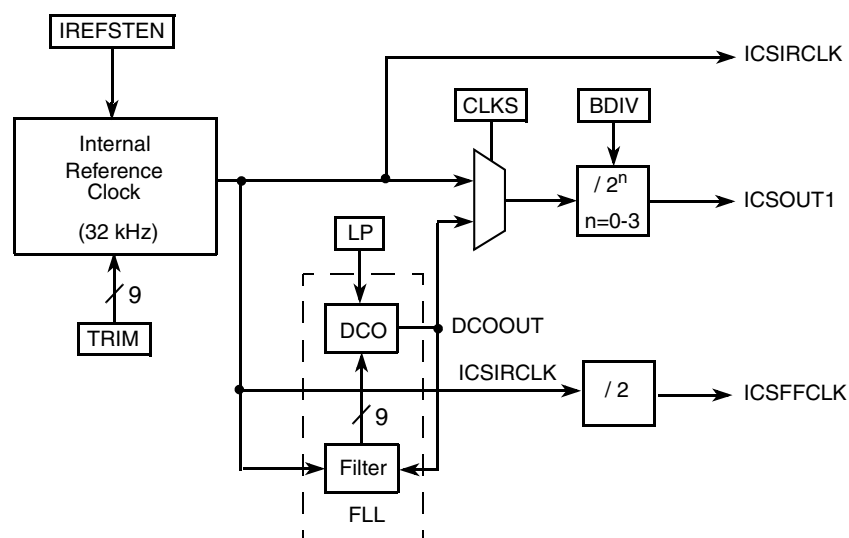
在 FLL 旁路内部低功耗模式, FLL 被禁止和旁路, ICS 提供一个被内部参考时钟分频的时钟。

10.1.2.4 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。

10.1.3 方框图

图 10-2 列出了 ICS 方框图。



1 ICSOUT 是总线频率的 2 倍

图 10-2. 内部时钟源 (ICS) 方框图

10.2 外部信号描述

无 ICS 信号连接芯片。

10.3 寄存器定义

表 10-1 是 ICS 寄存器摘要

表 10-1. 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								

10.3.1 ICS 控制寄存器 1 (ICSC1)

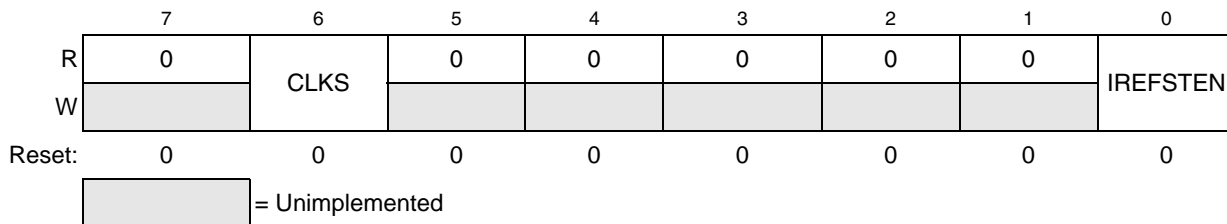


图 10-3. ICS 控制寄存器 1 (ICSC1)

表 10-2. 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

10.3.2 ICS 控制寄存器 2 (ICSC2)

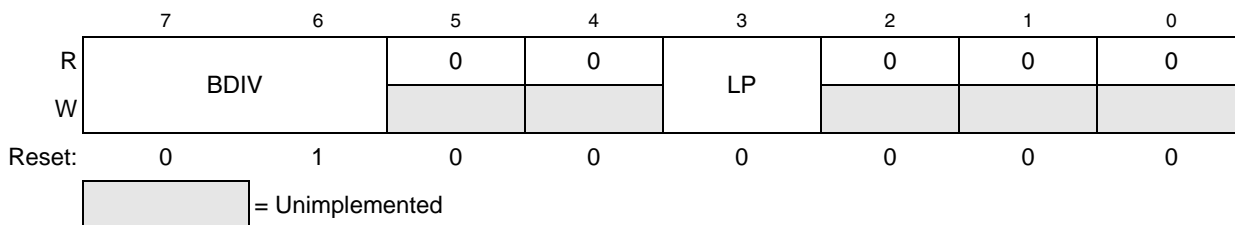


图 10-4. ICS 控制寄存器 2 (ICSC2)

表 10-3. 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

10.3.3 ICS 调整寄存器 (ICSTRM)

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

图 10-5. ICS 调整寄存器 (ICSTRM)

表 10-4. 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。

10.3.4 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

图 10-6. ICS 状态和控制寄存器 (ICSSC)

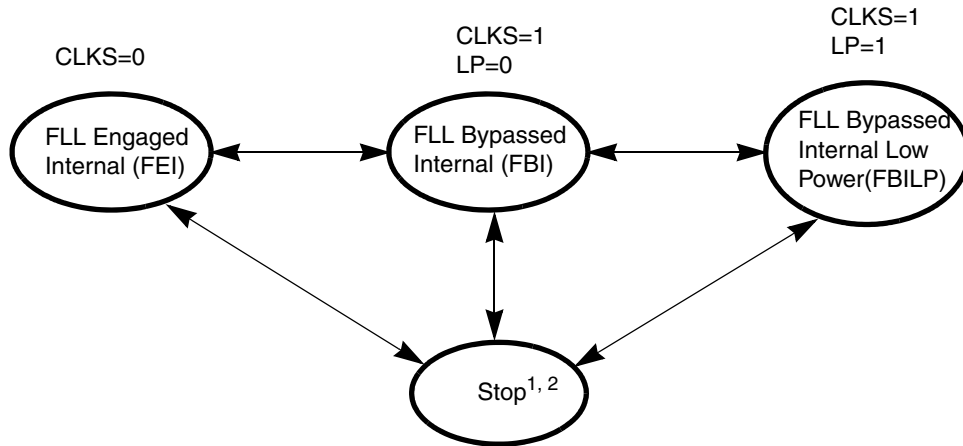
表 10-5. ICSSC 域描述

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。

10.4 功能描述

10.4.1 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。

图 10-7. 时钟切换模式

10.4.1.1 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。

10.4.1.2 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。

10.4.1.3 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。

10.4.1.4 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。

10.4.2 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。

10.4.3 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。

10.4.4 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。

10.4.5 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定位。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。

10.4.6 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。

第 11 章

模定时器 (RS08MTIMV1)

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 SC667028 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。

图 11-1 列出了 SC667028 方框图中高亮部分为 MTIM 模块。

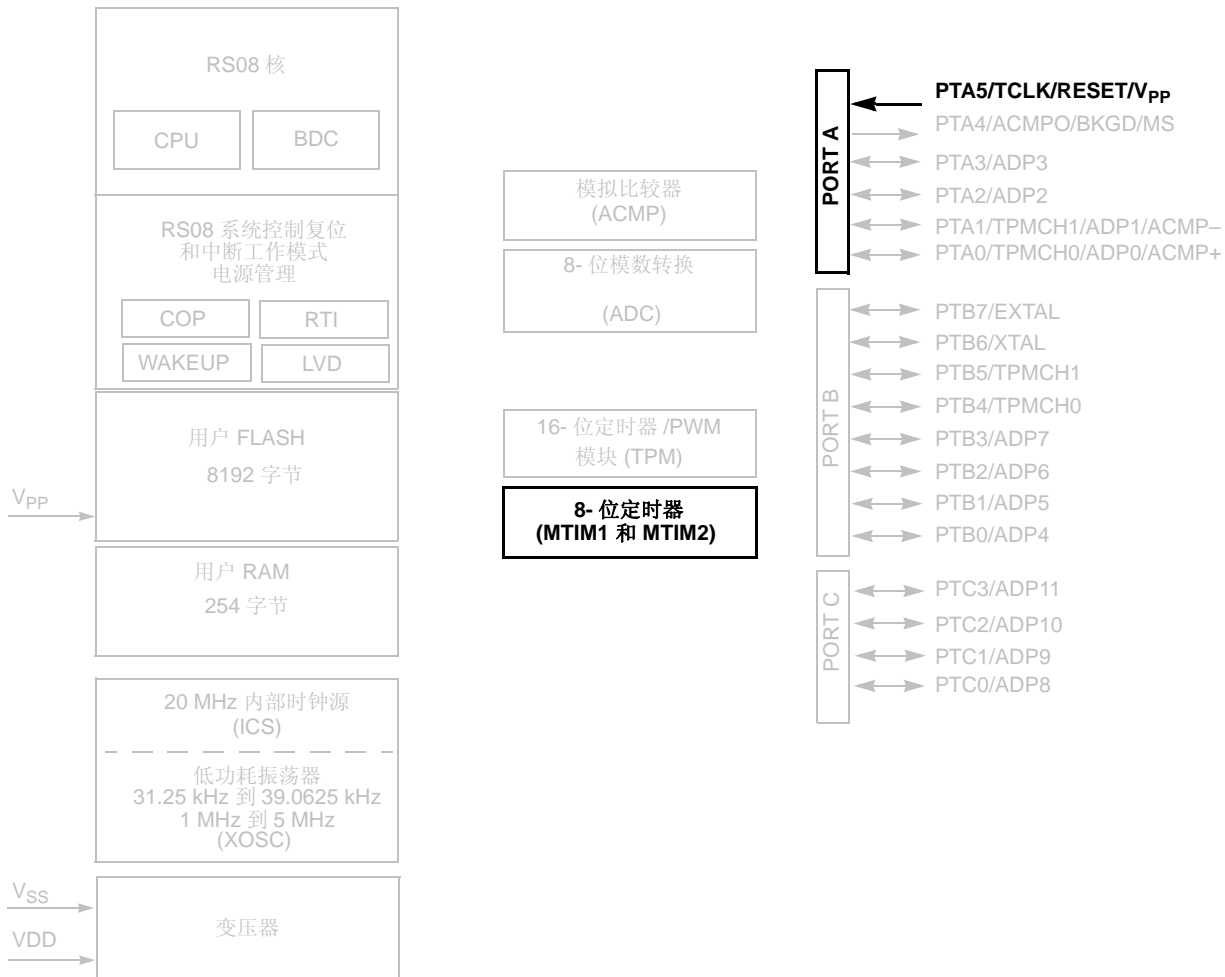


图 11-1. 显亮 MTIM 模块和引脚的 SC667028 模块结构图

11.1.1 特性

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图 11-2 所示。

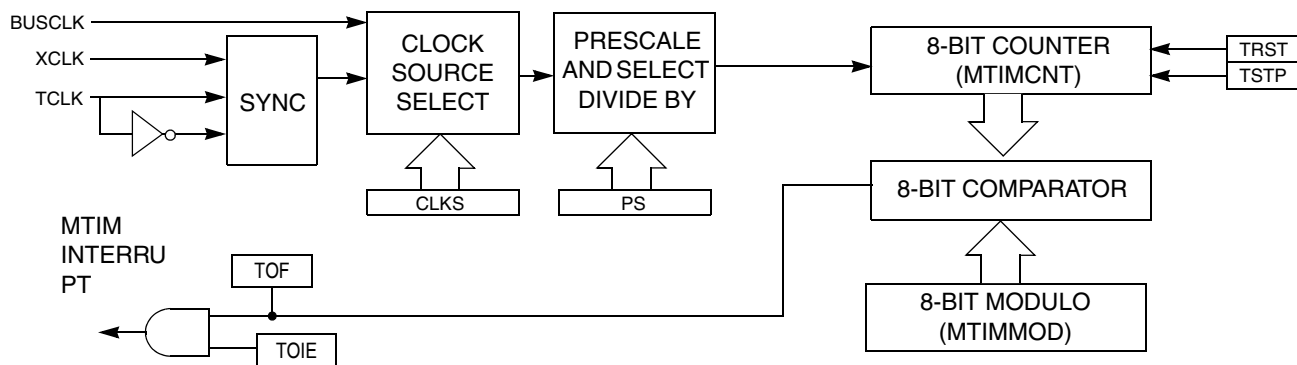


图 11-2. 模定时器 (MTIM) 方框图

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 信号属性如表 11-1 所示。

表 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第 2 章 引脚与连接。

11.3 寄存器定义

每个 MTIM 包含四个寄存器，概括在表 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。

表 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								

表 11-2. MTIM 寄存器概要 (continued)

Name		7	6	5	4	3	2	1	0
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。

	7	6	5	4	3	2	1	0
R	TOF	TOIE	0	TSTP	0	0	0	0
W			TRST					
Reset:	0	0	0	1	0	0	0	0

图 11-3. MTIM 状态和控制寄存器 (MTIMSC)

表 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)。

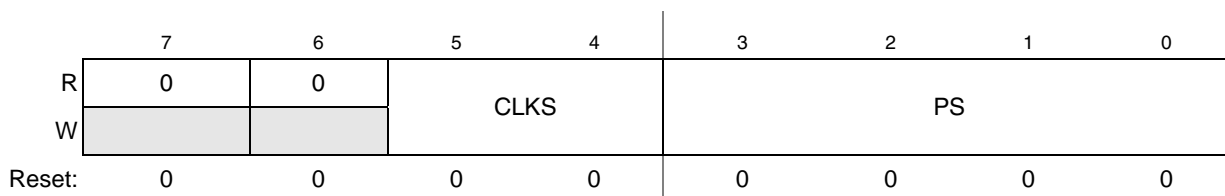


图 11-4. MTIM 时钟配置寄存器 (MTIMCLK)

表 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。

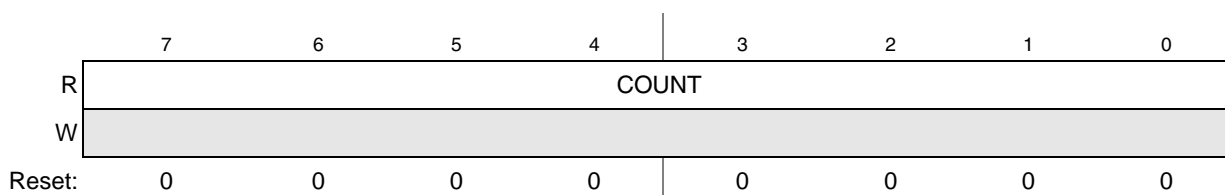


图 11-5. MTIM 计数寄存器 (MTIMCNT)

表 11-5. MTIMCNT 域描述

域	描述
7:0 COUNT	<p>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。</p>

11.3.4 MTIM 模寄存器 (MTIMMOD)

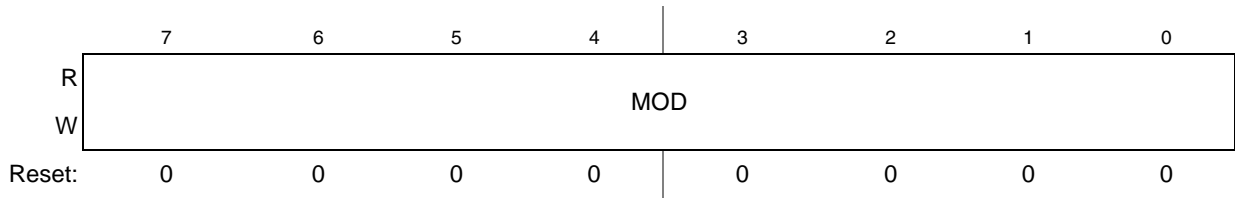


图 11-6. MTIM 模寄存器 (MTIMMOD)

表 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 为模寄存器、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。

The 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。

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。

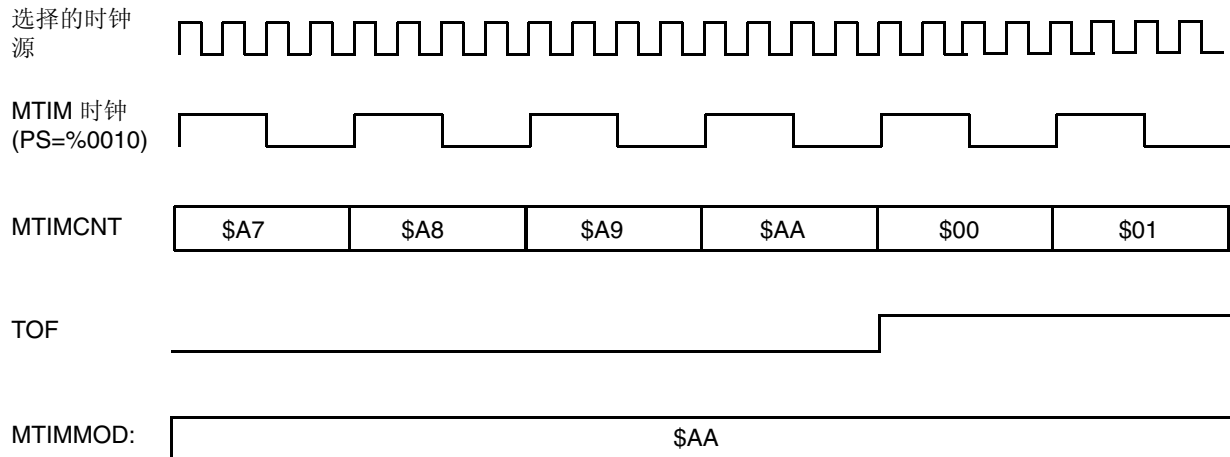


图 11-7. MTIM 计数器溢出例子

在图 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。

第 12 章

16 位定时器 /PWM(RS08TPMV2)

12.1 引言

每路 TPM 使用 1 个输入 / 输出 (I/O) 引脚, TPMCHn。其中, x 表示 TPM 号, 例如 1 或 2; n 表示通道号, 例如 0-4。TPM 与通用 I/O 端口共用 I/O 引脚 (参考第 2 章 引脚与连接)。

TPM 模块的引脚, TPMCH0 和 TPMCH1, 可利用 SOPT 中的 TPMCH0PS (表 12-1) 通过软件控制重新配置。SOPT 中的 TPMCH0PS 和 TPMCH1PS 选择 TPM 使用那个通用 I/O 端口。

表 12-1. TPM 位置选项

SOPT 中的 TPMCH0PS	TPMCH0 引脚	SOPT 中的 TPMCH1PS	TPMCH10 引脚
0 (默认)	PTA0	0 (默认)	PTA1
1	PTB4	1	PTB5

图 12-1 显示了 SC667028 框图, 其中 TPM 模块和引脚高亮。

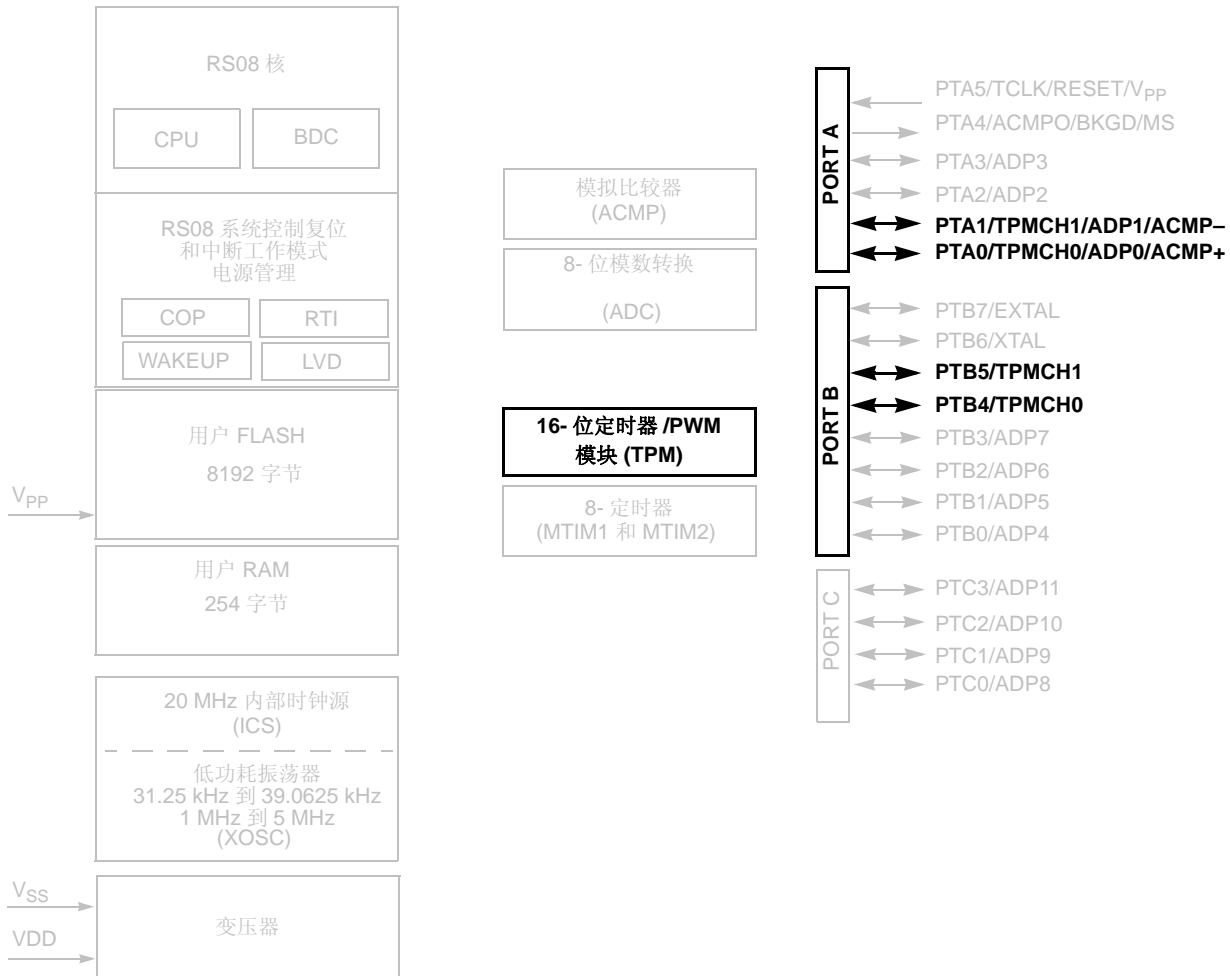


图 12-1. 显亮 TPM 模块和引脚的 SC667028 模块结构图

12.1.1 特性

TPM 模块包括下列特性：

- 每个 TPM 的各个通道均可配置为带缓冲，中央对齐的脉宽调制输出 (CPWM)
- 每个 TPM（多路 TPM）独立选择时钟源。
- 可选的时钟源（依赖于设备）：总线时钟，固定系统时钟，外部引脚。
- 时钟分频因子：1，2，4，8，16，32，64 或 128
- 16 位自由运行或递增递减（CPWM）计数操作。
- 16 位预置寄存器控制计数范围。
- 定时器使能。
- 每个 TPM 模块（多路 TPM）的各个通道都有一个中断，加上一个终端计数中断。
- 通道特性：
 - 每个通道都有输入捕捉，输出比较，带缓冲中央对齐 PWM。
 - 上升沿，下降沿，或任意边沿输入捕捉的触发器。
 - 输出比较动作包括置位，清零，翻转。
 - PWM 输出极性可选

12.1.2 框图

图 12-2 显示了 TPM 的结构。一些 MCU 包含超过一个的 TPM，不同数量的通道。

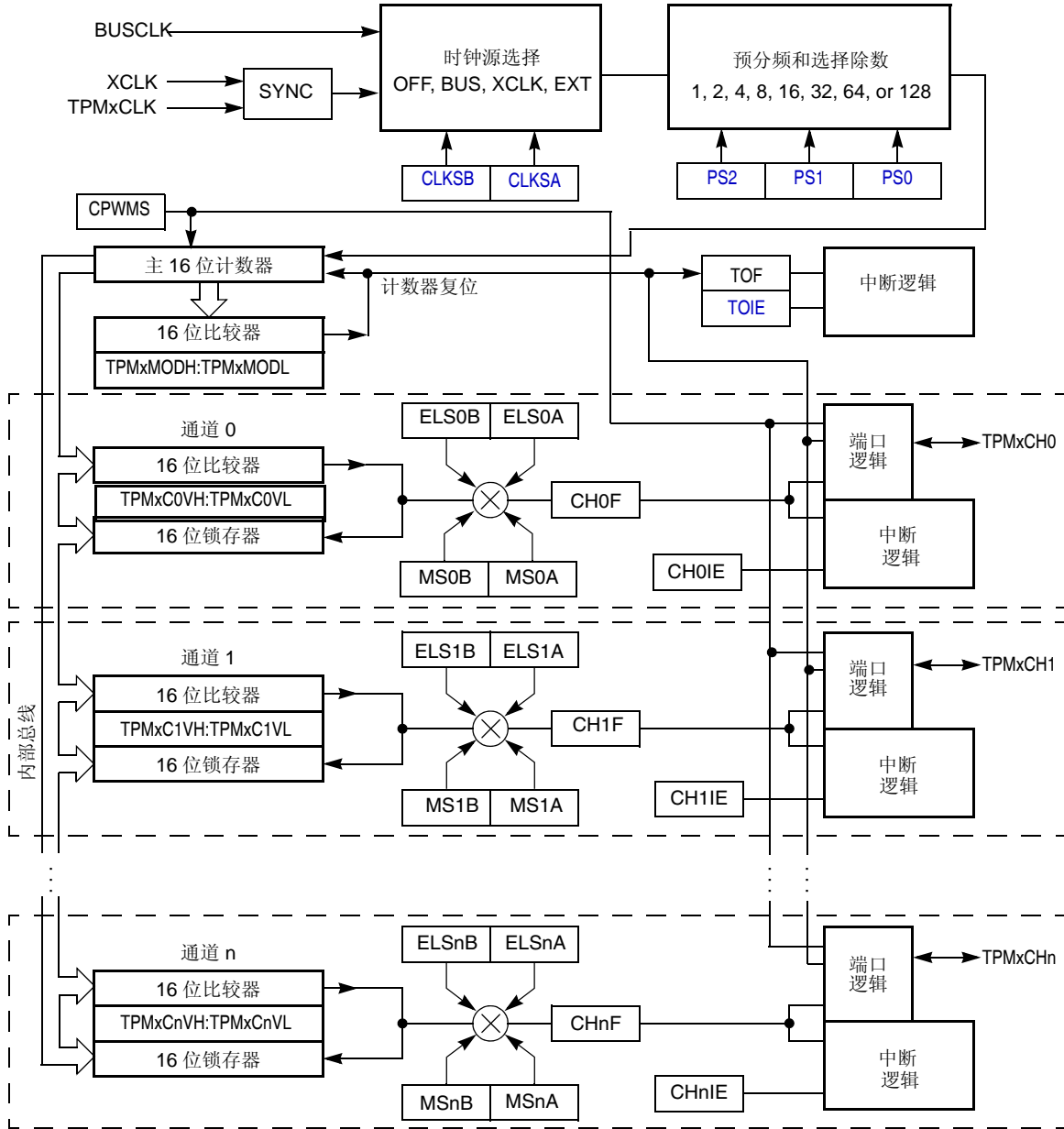


图 12-2. TPM 原理框图

TPM 的中央部件是 16 位计数器，当配置为中央对齐 PWM 时，可以作为一个自由运行计数器，预置计数器，向上 / 向下计数器。TPM 计数器（当处于正常加 1 操作时）提供输入捕捉，输出比较，和边沿对齐 PWM 功能的时序参考。定时器计数器预置寄存器（TPMMODH:TPMMODL）控制计数器的预置值。（0x0000 或 0xFFFF 使计数器自由运行。）软件可以随时读取计数值而不影响计数时序。向 TPMCNT 的任意字节写入任意值，计数器都会复位。

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

12.2 外部信号描述

当和定时器相关的引脚配置为定时器输入时，允许被动上拉。复位后，禁止 TPM 模块，所有的引脚默认为禁止上拉的通用输入。

12.2.1 外部 TPM 时钟源

当定时器状态和控制寄存器中的控制位 CLKSB:CLKSA 设置成 1:1 时，TPM 的预分频和 16 位计数器使用外部时钟源 TPMxCLK 驱动。外部时钟和 TPM 之间需要一个同步器。这个同步器的时钟使用总线时钟，所以外部时钟的频率必须小于总线时钟的一半。这个外部时钟源的最高频率限制规定为总线时钟的四分之一，调节占空比和锁相环或锁频环的频率抖动影响。

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

12.2.2 TPMCHn——TPM 通道 n I/O 引脚

每个 TPM 通道都和 MCU 的 I/O 引脚相关。引脚的功能依赖于通道的配置。在某些情况，不使用引脚特殊功能时，引脚由通用 I/O 控制。当定时器控制端口引脚时，端口数据和方向寄存器都没有影响相应的引脚。更多关于引脚功能复用的信息参见“引脚和连接”一章。

12.3 寄存器定义

TPM 包括：

- 一个 8 位状态和控制寄存器（TPMSC）。
- 一个 16 位计数寄存器（TPMCNTH:TPMCNTL）。
- 一个 16 位预置寄存器（TPMMODH:TPMMODL）。

每个定时器通道包括：

- 一个 8 位状态和控制寄存器（TPMCnSC）。
- 一个 16 位通道值寄存器（TPMCnVH:TPMCnVL）。

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

12.3.1 定时器状态和控制寄存器（TPMSC）

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

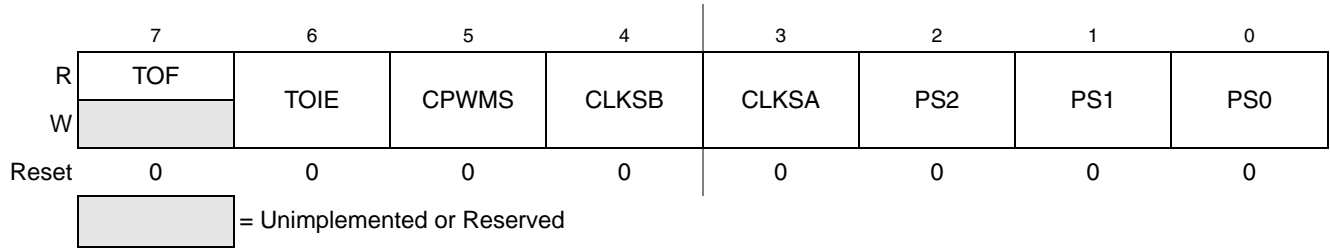


图 12-3. 定时器状态和控制寄存器 (TPMSC)

表 12-2. TPMSC 寄存器域描述

域	描述
7 TOF	定时器溢出标志——当 TPM 计数器达到 TPM 计数器预置寄存器所编程的预置值，TPM 计数器变成 0x0000 时，该位置位。当 TPM 配置为 CPWM，计数器达到预置寄存器的值时，TOF 置位，并转换到下一个更低的计数值。当 TOF 置位时通过读 TPM 状态和控制寄存器清零 TOF，然后向 TOF 写 0。如果在清零序列完成前另一个溢出发生，则序列复位，所以在更早的 TOF 完成清零序列后，TOF 仍然置位。复位会清零 TOF。向 TOF 写 1 没有任何影响。 0 TPM 计数器未达到预置数或溢出。 1 TPM 已经溢出。
6 TOIE	定时器溢出中断使能——该位可读写。允许 TPM 溢出中断。如果置位 TOIE，当 TOF 等于 1 时产生中断。复位清零 TOIE。 0 禁止 TOF 中断（使用软件轮询）。 1 允许 TOF 中断。
5 CPWMS	中央对齐 PWM 选择——该位可读写。选择 CPWM 操作模式。复位清零该位，所以 TPM 以向上计数模式操作输入捕捉，输出比较，边沿对齐 PWM 功能。置位 CPWMS 重新配置 TPM 以向上 / 向下计数模式操作 CPWM 功能。复位清零 CPWMS。 0 所有的 TPM 通道通过各自的通道状态和控制寄存器的 MSnB:MSnA 控制位来选择输入捕捉，输出比较，边沿对齐 PWM 模式。 0 所有的 TPM 通道处于中央对齐 PWM 模式操作。
4:3 CLKS[B:A]	时钟源选择——如图 12-3 所示，这 2 位用于禁止 TPM 系统或选择 3 个时钟源中的一个来驱动计数器预分频。外部源和 XCLK 通过片上同步电路和总线时钟同步。
2:0 PS[2:0]	分频因子选择——这 3 位为 TPM 时钟输入选择表 12-3 所示的 8 个因子中的一个。这个分频在时钟源同步或时钟源选择后确定。无论 TPM 选择什么时钟源驱动 TPM 系统，它都会影响。

表 12-3. TPM 时钟源选择

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

¹ 外部时钟所允许的最大频率是总线频率的四分之一。

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

表 12-4. 分频因子选择

PS2:PS1:PS0	时钟源分频因子
0:0:0	1
0:0:1	2
0:1:0	4
0:1:1	8
1:0:0	16
1:0:1	32
1:1:0	64
1:1:1	128

12.3.2 定时器计数寄存器 (TPMCNTH:TPMCNTL)

两个只读 TPM 计数寄存器包括 TPM 计数器的高字节和低字节。读其中一个字节 (TPMCNTH 或 TPMCNTL) 都将两个字节的內容锁存到一个缓冲区, 一直锁存到读另一个字节。这样以不同顺序读取的 16 位数据仍一致。这种一致性机制可以在 MCU 复位, 写 TPMCNTH 或 TPMCNTL, 写定时器状态 / 控制寄存器 (TPMSC) 后自动重启。

复位清零 TPM 计数寄存器。



图 12-4. 定时器计数寄存器高 (TPMCNTH)

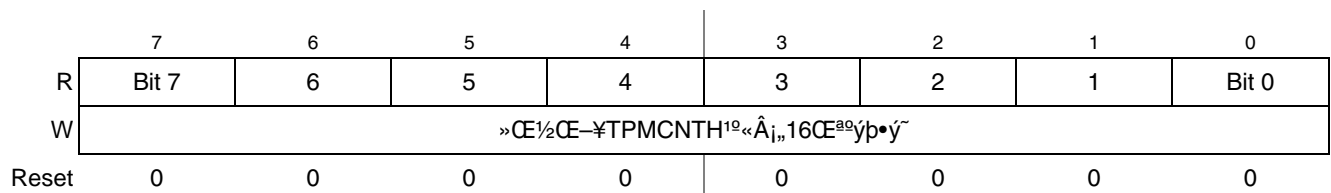


图 12-5. 定时器计数寄存器低 (TPMCNTL)

当后台调试有效时, 定时器计数和一致性机制被冻结, 这样, 当后台模式有效, 即使读计数器的一个或两个字节, 锁存仍然保持它们所处的状态。

12.3.3 定时器计数预置寄存器 (TPMMODH:TPMMODL)

读 / 写 TPM 预置寄存器包含 TPM 计数器的预置值。当 TPM 计数器达到预置值时, TPM 计数器在下一个时钟 (CPWMS=0) 从 0x0000 重新计数或开始向下计数 (CPWMS=1), 并且溢出标志 (TOF) 置位。写 TPMMODH 或 TPMMODL 时禁止 TOF 和溢出中断直到两个都写完。复位设置 TPM 计数器预置寄存器为 0x0000, 将会导致自由运行计数器 (禁止预置)。

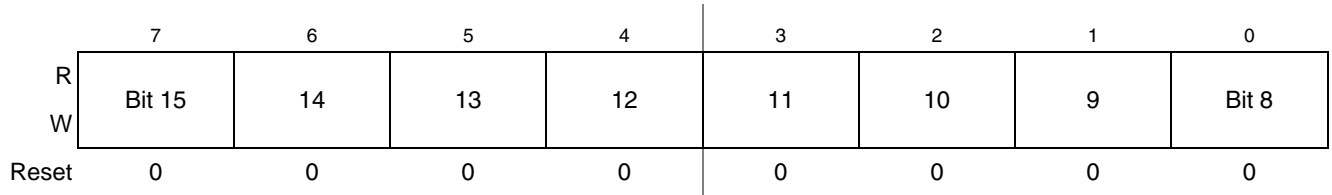


图 12-6. 定时器预置寄存器高位 (TPMMODH)

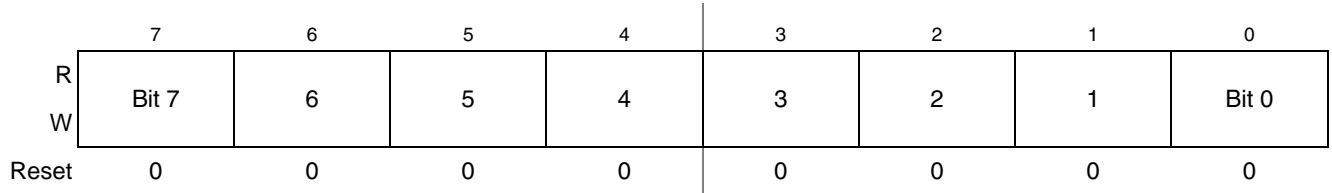


图 12-7. 定时器预置寄存器低位 (TPMMODL)

等待一个溢出中断这样预置寄存器的两字节可以在新的溢出中断前写完。替代方法是写 TPM 预置寄存器前复位 TPM 计数器，以避免第一个计数器溢出发生时发生冲突。

12.3.4 定时器通道 n 状态和控制寄存器 (TPMCnSC)

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

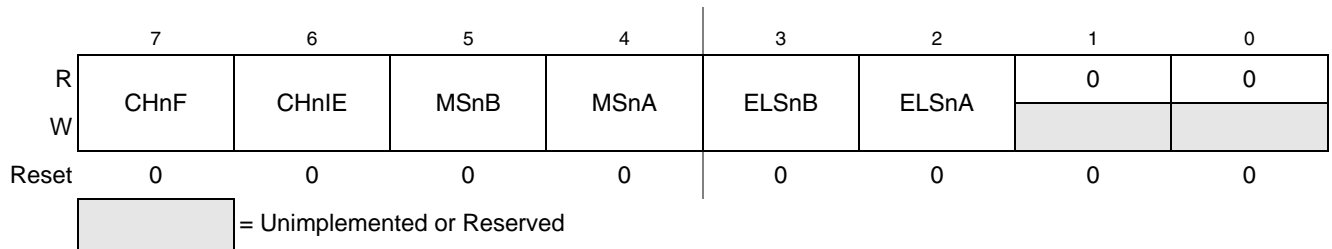


图 12-8. 定时器通道 n 状态和控制寄存器 (TPMCnSC)

表 12-5. TPMCnSC 寄存器域描述

域	描述
7 CHnF	通道 n 标志 — 当通道 n 配置为输入捕捉时，若在通道 n 引脚上产生了有效边沿，则该标志位置位。当通道 n 作为输出比较或边沿对齐 PWM 通道时，若 TPM 计数寄存器的值和 TPM 通道 n 的值寄存器的值相匹配时，则 CHnF 置位。该标志在中央对齐 PWM 中很少使用，因为每次计数值和通道值寄存器相匹配就会置位，这和有效占空比的两个边沿相符。 当 CHnF 置位且允许中断时 (ChnIE=1)，产生相应的中断请求。当 CHnF 置位时读 TPMCnSC 然后写 0 到 CHnF，可以清零 CHnF。如果在清零序列完成前另一个中断请求发生，则序列复位，所以在更早的 CHnF 完成清零序列后，CHnF 仍然置位。这样做的话，清零以前的 CHnF 不会丢失 CHnF 上的中断请求。复位会清零 CHnF。向 CHnF 写 1 没有任何影响。 0 在通道 n 上没有输入捕捉或输出比较事件发生。 1 在通道 n 上发生输入捕捉或输出比较事件。
6 CHnIE	通道 n 中断使能 — 该位可读写。允许通道 n 上的中断。复位清零 CHnIE。 0 禁止通道 n 的中断请求（使用软件轮询）。 1 允许通道 n 的中断请求。
5 MSnB	TPM 通道 n 的模式选择 B — 当 CPWMS=0，MSnB=1，配置 TPM 通道 n 为边沿对齐 PWM 模式，关于通道模式和控制参见表 12-6。

表 12-5. TPMCnSC 寄存器域描述

4 MSnA	TPM 通道 n 的模式选择 A — 当 CPWMS=0, MSnB=0, 配置 TPM 通道 n 为输入捕捉或输出比较模式, 关于通道模式和控制参见表 12-6。
3:2 ELSn[B:A]	边沿 / 电平选择位 — 依赖于定时器通道的工作模式, 如表 12-6 中 CPWMS:MSnB:MSnA 的设置, 这些位选择了触发一个输入捕捉事件所需要的输入边沿的极性, 选择输出比较匹配时的驱动电平, 或选择 PWM 输出的极性。设置 ELSnB:ELSnA 为 0:0 则会配置相应的定时器引脚作为通用 I/O, 和任何定时器通道功能都没有关系。该功能用于临时禁止输入捕捉通道, 或当相应的定时器通道设置为软件定时器而不需要使用引脚时, 使定时器引脚作为通用 I/O 引脚,。

表 12-6. 模式, 边沿和电平选择

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

当改到输入捕捉模式前, 如果相应的端口引脚有至少两个总线时钟周期不稳定, 可能会得到一个所不希望的边沿触发。通常在改变了通道配置位后和允许通道中断或使用状态标志前, 编程清零状态标志以避免意外动作。

12.3.5 定时器通道值寄存器 (TPMCnVH:TPMCnVL)

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

	7	6	5	4	3	2	1	0
R	Bit 15	14	13	12	11	10	9	Bit 8
W								
Reset	0	0	0	0	0	0	0	0

图 12-9. 定时器值寄存器高位 (TPMCnVH)

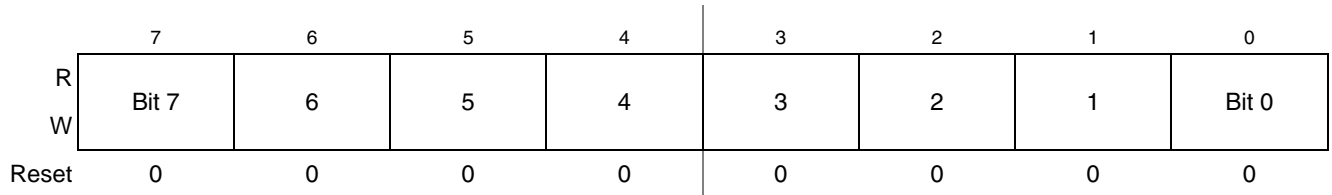


图 12-10. 定时器值寄存器低位 (TPMCnVL)

在输入捕捉模式中，读任何一个字节 (TPMCnVH 或 TPMCnVL) 锁存这两个字节的内容到一个缓冲区，将会锁存直到另一个字节被读。当写 TPMCnSC 寄存器时，该锁存机制也会复位（不锁存）。

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

这种锁存机制使没有同时写的 16 位一致，可以在不同的编译器中实现。

12.4 功能描述

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

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

接下来的几节将描述主 16 位计数器和各种定时器操作模式（输入捕捉，输出比较，边沿对齐 PWM，中央对齐 PWM）。因为引脚操作和中断行为的细节依赖于操作模式，这些内容将在有关模式的章节介绍。

12.4.1 计数器

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

任何 MCU 复位后，CLKSB:CLKSA=0:0，所以没有选择时钟源并且 TPM 无效。通常 CLKSB:CLKSA 设置成 0:1，总线时钟驱动定时器计数器。TPM 的时钟源可以选择关，总线时钟 (BUSCLK)，固定系统时钟 (XCLK)，或外部输入。外部时钟选项所允许的最大频率是总线的四分之一。更多关于时钟源选择的信息参见 12.3.1 节 定时器状态和控制寄存器 (TPMSC) 和表 12-2。

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

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

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

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

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

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

12.4.2 通道模式选择

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

12.4.2.1 输入捕捉模式

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

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

输入捕捉事件置位可选产生 CPU 中断请求的标志位（CHnF）。

12.4.2.2 输出比较模式

对于输出比较功能，TPM 可以产生可编程的定时的脉冲，极性，持续时间和频率。当计数器达到输出比较通道中的通道值寄存器的值时，TPM 置位，清零，或翻转通道引脚。

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

输出比较事件置位可选产生 CPU 中断请求的标志位（CHnF）。

12.4.2.3 边沿对齐 PWM 模式

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

如图 12-11 所示，TPM 通道寄存器的输出比较值决定了 PWM 信号的脉冲宽度（占空比）。预置溢出和输出比较的时间就是脉冲宽度。如果 ELSnA=0，计数器溢出强制 PWM 信号高并且输出比较强制 PWM 信号低。如果 ELSnA=1，计数器溢出强制 PWM 信号低并且输出比较强制 PWM 信号高。

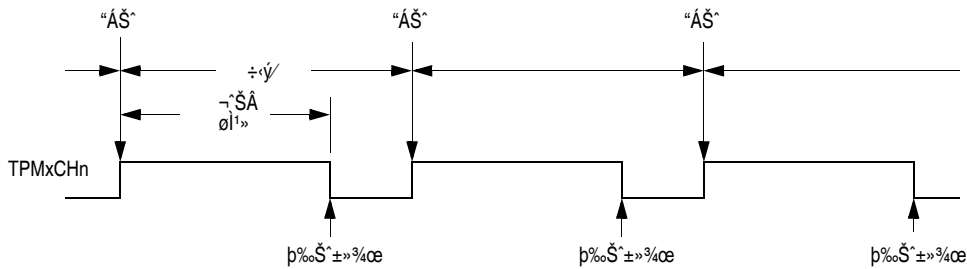


图 12-11. PWM 周期和脉宽 (ELSnA=0)

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

因为 HCS08 是 8 位 MCU 系列，定时器通道寄存器的设置被缓存以确保 16 位更新的一致，避免不希望的 PWM 脉宽。写 TPMCnVH 或 TPMCnVL 中的一个寄存器，写缓冲寄存器。在边沿 PWM 模式中，只有在写了 16 位寄存器的两个 8 位字节并且 TPMCNTH:TPMCNTL 计数器的值时 0x0000，值才被传输到相应的定时器通道寄存器。（新的占空比直到下一个周期才产生作用。）

12.4.3 中央对齐 PWM 模式

这种类型的 PWM 输出使用定时器计数器的向上 / 向下计数模式 (CPWMS=1)。通过设置 TPMCnVH:TPMCnVL 的输出比较值确定 PWM 信号的占空比。设置 TPMMODH:TPMMODL 来确定周期。TPMMODH:TPMMODL 必须保持在 0x0001 到 0x7FFF 的范围，因为超过这个范围将会产生模糊的结果。ELSnA 将确定 CPWM 输出的极性。

$$\text{脉冲宽度} = 2 \times (\text{TPMCnVH:TPMCnVL}) \quad \text{等式 12-1}$$

$$\text{周期} = 2 \times (\text{TPMMODH:TPMMODL}) \quad \text{等式 12-2}$$

其中 TPMMODH:TPMMODL=0x0001 – 0x7FFF

如果通道值寄存器 TPMCnVH:TPMCnVL 是 0 或负数（15 位置位），占空比为 0%。如果 TPMCnVH:TPMCnVL 是一个正数（15 位清零）并且大于（非零）预置设置，占空比为 100%，因为不会产生占空比比较。这隐含了通过预置寄存器设置的可用周期范围是 0x0001 到 0x7FFE（如果产生 100% 的占空比不需要 0x7FFF）。因为产生的周期比正常应用需要的更长，这不是重要的限制。

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

图 12-12 表明了 TPM 通道寄存器的输出比较值（乘以 2），TPM 通道寄存器决定 CPWM 信号的脉宽（占空比）。如果 ELSnA=0，当向上计数强制 CPWM 信号低时比较匹配和当向下计数强制输出高时比较匹

配。计数器向上计数直到达到 TPMODH:TPMODL 设置的预置，然后向下计数直到 0。这样设置周期等于 TPMODH:TPMODL 的两倍。

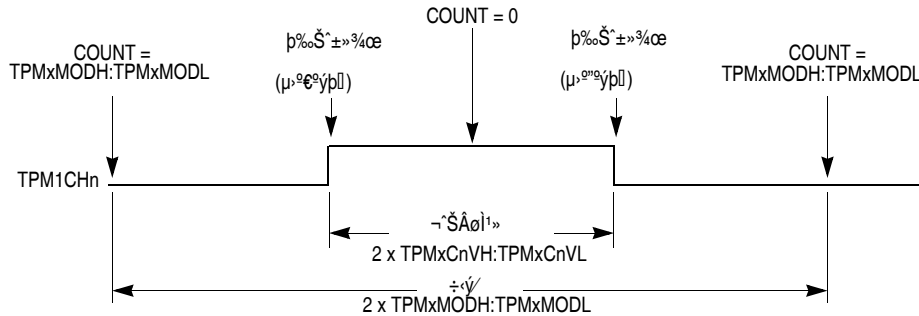


图 12-12. CPWM 周期和脉宽 (ELSnA=0)

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

应为 HCS08 是 8 位 MCU 系列，定时器通道寄存器的设置为缓存，确保 16 位更新一致，并且避免不希望的 PWM 脉宽。写 TPMODH、TPMODL、TPMCnVH 和 TPMCnVL 的任何一个寄存器，实际上都是写缓存寄存器。只有在 16 位寄存器的两个 8 位字节都写完并且定时器计数器溢出（在预置寄存器结束计数后，翻转方向从向上计数到向下计数）之后，值才被传到相应的定时器通道寄存器。TPMCNT 溢出需要仅适用于 PWM 通道，不适用于输出比较。

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

写 TPMSA 将会取消写到 TPMODH 或 / 和 TPMODL 的值，并且会复位预置寄存器的一致性机制。写 TPMCnSC 将会取消写到通道值寄存器的值，并且会复位 TPMCnVH:TPMCnVL 的一致性机制。

12.5 TPM 中断

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

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

12.5.1 清零定时器中断标志

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

12.5.2 定时器溢出中断描述

导致 TOF 置位的条件依赖于计数模式（向上或向上 / 向下）。在向上计数模式中，16 位定时器计数器从 0x0000 到 0xFFFF 计数，并且在下一个计数周期溢出到 0x0000。TOF 在从 0xFFFF 转换到 0x0000 时置位。当定时器操作处于向上 / 向下计数模式时，当计数器方向从预置寄存器设置的值转换到下一个更低计数值时，TOF 标志置位。这对应于 PWM 周期的结束。（0x0000 计数值对应周期的中央。）

12.5.3 通道事件中断描述

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

当通道配置为输入捕捉通道时，ELSnB:ELSnA 控制位选择上升沿，下降沿，任何边沿，或无边沿（关）作为触发输入捕捉事件的边沿。当检测到选择的边沿，中断标志位置位。通过在 12.5.1 节“清零定时器中断标志”所描述的两个步骤可以清零该标志。

当通道配置为输出比较通道时，每次主定时器计数器和 16 位的通道值寄存器相匹配时，中断标志位置位。通过在 12.5.1 节 清零定时器中断标志所描述的两个步骤可以清零该标志。

12.5.4 PWM 占空比结束事件

通道被配置为 PWM 操作，有两个可能性：

- 当通道配置为边沿对齐 PWM 时，当定时器计数器与通道值寄存器匹配时，标志一个有效的占空比的结束，通道标志位置位。
- 当通道配置为中央对齐 PWM 时，每个 PWM 周期定时器计数和通道值寄存器匹配两次。在 CPWM 情况下，在有效占空比的开始和结束通道标志置位，这正是定时器计数器和通道值寄存器相匹配的次数。

通过在 12.5.1 节 清零定时器中断标志所描述的两个步骤可以清零该标志。

第 13 章 开发支持

13.1 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。

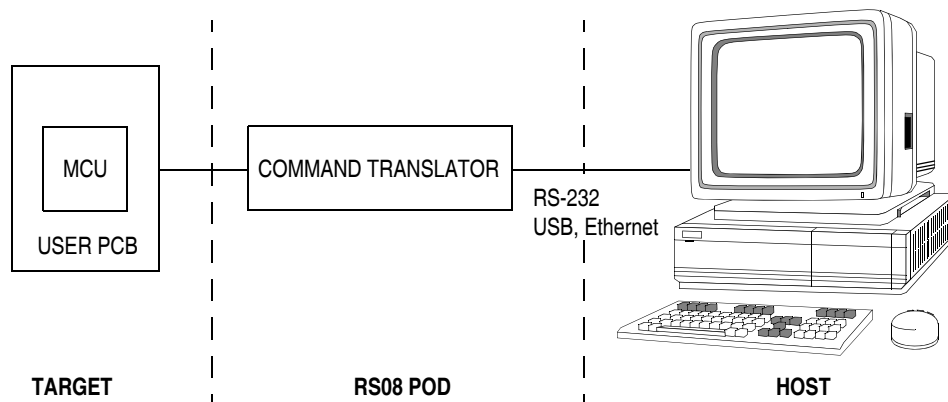


图 13-1. 连接 MCU 到主机用于调试

13.2 特性

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

13.3 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式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。
- 非侵入式命令能在任何时候被执行，甚至在用户程序运行时期。非侵入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。

图 13-2 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 RESET 信号提供用于主机强迫或监控 (假如 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。如图 13-2 所示，设备连接出的 V_{pp} 与 RESET 共享。对于 V_{pp} 的需求参见电气附件中的 FLASH 规范。

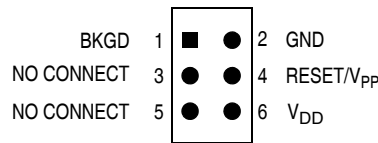


图 13-2. 标准 RS08 BDM 工具连接器

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 13.3.2 节 通讯详述。

13.3.1 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 13.3.2 节 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，用户必须禁止这可选的功能。在没有用 BDM 时，此可选功能调试用在正常用户模式。

13.3.2 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。

图 13-3 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。

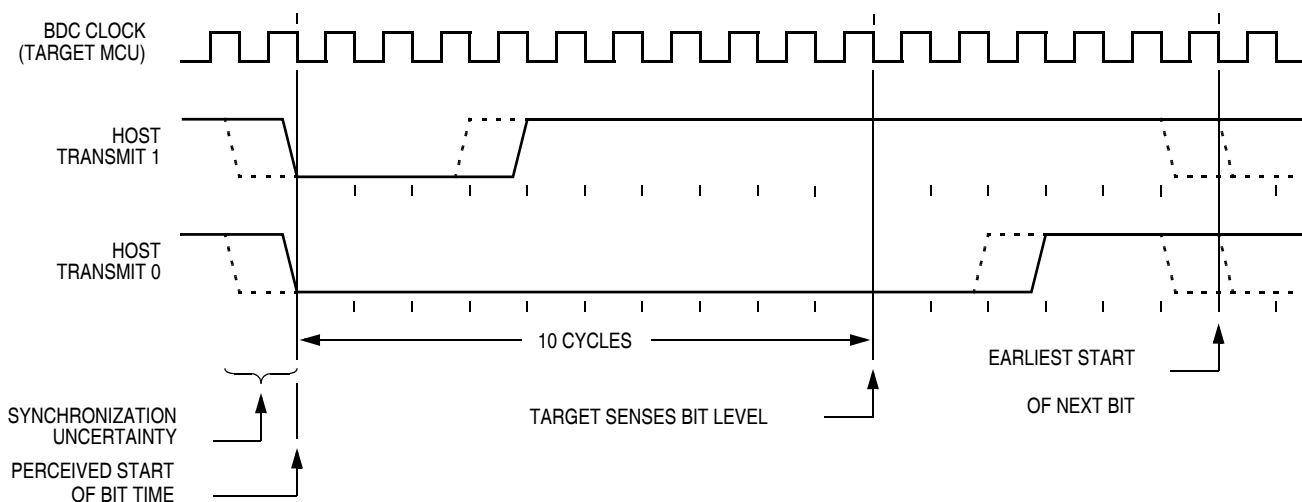


图 13-3. BDC 主机—目标串行位控制时间

图 13-4 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。

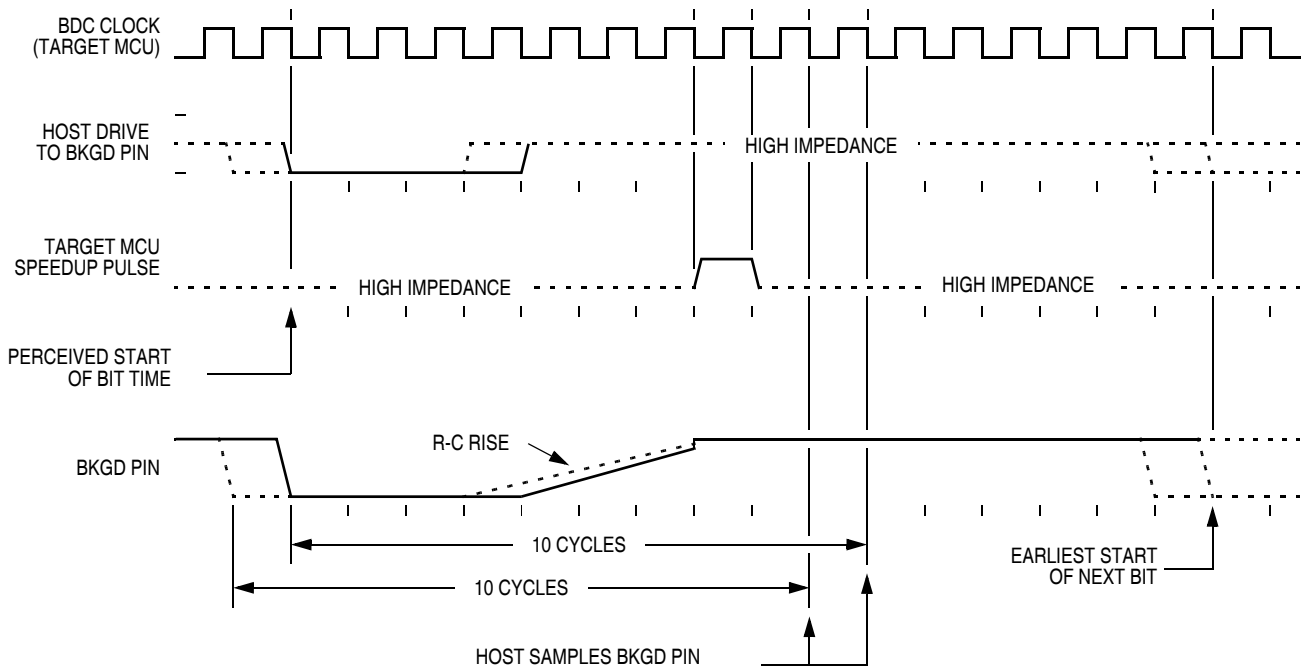


图 13-4. BDC 目标—主机串行位控制时间 (逻辑 1)

图 13-5 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。

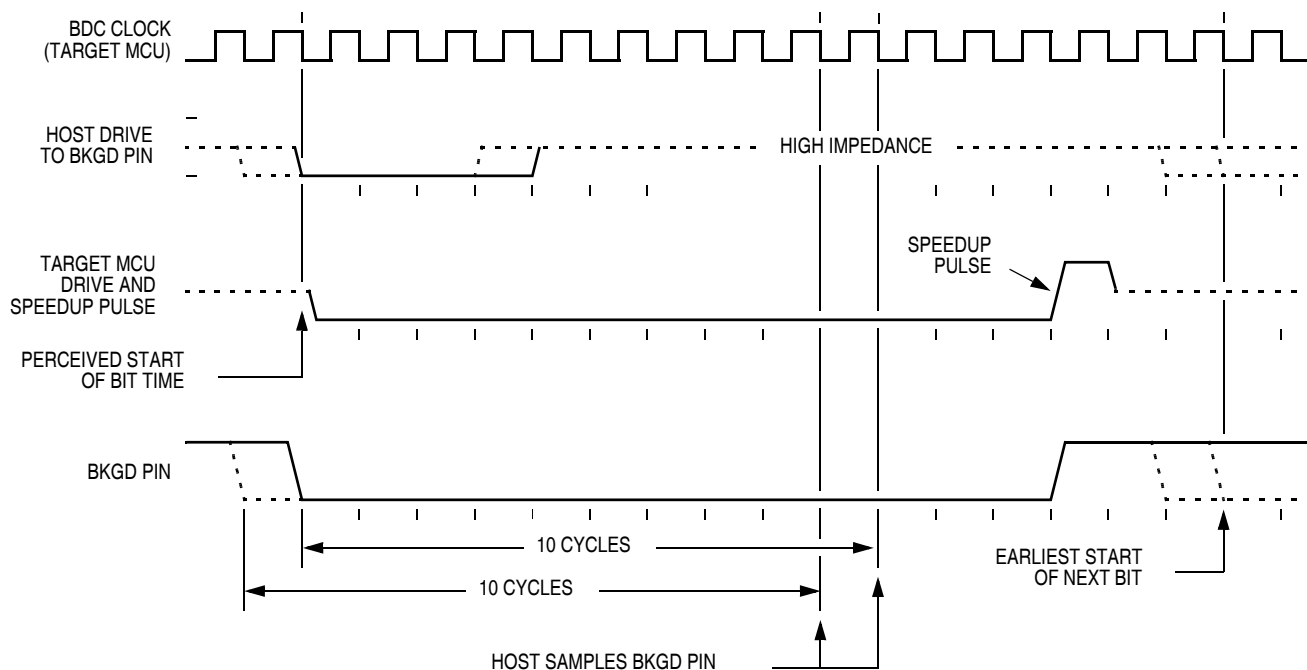


图 13-5. BDM 目标—主机串行位控制时间 (逻辑 0)

13.3.3 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。

13.4 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。

13.4.1 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	ENBDM	BDMACT	BKPTEN	FTS	0	WS	WSF	0
W								
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

图 13-6. BDC 状态和控制寄存器 (BDCSCR)

表 13-1. 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。

13.4.2 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”。

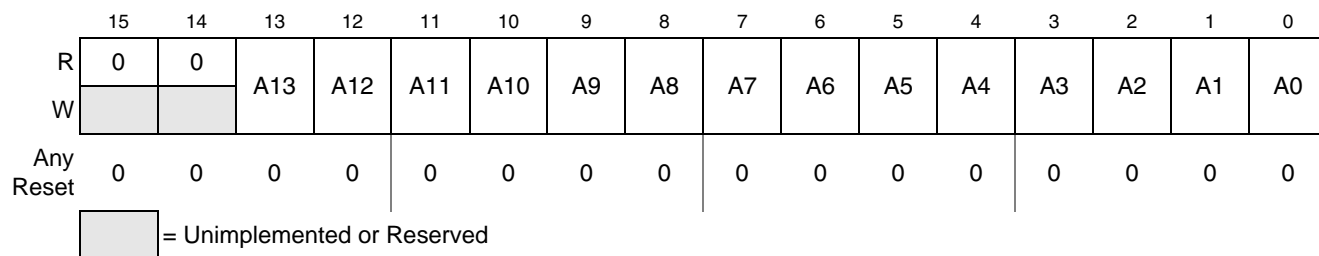


图 13-7. BDC 断点匹配寄存器 (BDCBKPT)

13.5 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。

表 13-2 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

下面术语被用于表 13-2 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。

表 13-2. 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。

第 13 章 开发支持

- 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)。

附录 A

电气特性

A.1 引言

这一章包含 SC667028 微控制器目前的电气和时间特性。

A.2 参数分类

这个附录中显示的电气参数通过各种方式验证。为了让客户更好的理解下列分类被使用的地方，在下表中参数做了适当的标注。

表 A-1. 参数分类

P	参数在每一个设备上的生产测试期间得到保证。
C	参数是通过测量跨过过程变化统计有关样品大小的设计特性取得。
T	除非另有说明，参数是在典型条件下的典型设备的小型样例的设计特性取得。所有的在典型列中显示的数值都在这个范畴以内。
D	参数主要从仿真中派生而来。

注意

在参数表 ‘C’ 栏中显示的分类是它适用的地方。

A.3 最大绝对额定值

最大绝对额定值是强制的额定值，且在最大绝对值下功能工作是没有保证的。在表 A-2 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。关于功能工作条件参见该节的其他表格。

该芯片包含了对高静态电压或电场的保护电路，可以避免因高静态电压或电子磁场产生的损坏。然而，建议采取正常防范以避免任何高于最大额定电压的电压进入该高阻抗电路。如果不再使用的输入被限制于一适合的逻辑电压（例如， V_{SS} 或 V_{DD} ）或者允许和引脚有关的可编程的上拉电阻，操作的可靠性将会加强。

表 A-2. 最大绝对额定值

额定值	标志	值	单位
电源电压	V_{DD}	-0.3 到 5.8	V
V_{DD} 的最大电流	I_{DD}	120	mA
输入电压	V_{in}	-0.3 到 $V_{DD} + 0.3$	V

表 A-2. 最大绝对额定值

单个引脚限制的瞬间最大电流 (应用到所有引脚) ^{1, 2, 3}	I_D	± 25	mA
存储温度范围	T_{stg}	-55 到 +150	°C

- ¹ 输入当前限制于描述值。为了决定需要的限流电阻器的值，计算正 (V_{DD}) 和负 (V_{SS}) 箝位电压的电阻系数，然后使用两个电阻系数中较大的一个。
- ² 除了 \overline{RESET}/V_{PP} 引脚仅在内部箝位到 V_{SS} ，所有功能性非电源引脚在内部箝位至 V_{SS} 和 V_{DD} 。
- ³ 电源必须维持调节 V_{DD} 的瞬间范围和最大电流情况。如果正的注入电流 ($V_{in} > V_{DD}$) 大于 I_{DD} ，注入电流可能对 V_{DD} 溢出导致外部电源停止调整。确保外荷载 V_{DD} 分流大于最大注入电流。当 MCU 不耗电时，这将是更大的风险。例如：如果系统时钟停止，或时钟速率很低减小总功率功耗。

A.4 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 引脚的功率消耗与片内逻辑和电压调整器电路和由用户决定而不被 MCU 设计控制的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，明确实际引脚电压和 V_{SS} 或 V_{DD} 以及和每个 I/O 引脚的电流乘积的不同之处。除了特别的大引脚电流（大负载），引脚电压和 V_{SS} 或 V_{DD} 的不同之处非常小。

表 A-3. 热特性

额定值	标志	值	单位
工作温度范围（已封装）	T_A	T_L 到 T_H -40 到 85	°C
最大连接温度	T_{JMAX}	105	°C
热阻 20 引脚 SOIC	θ_{JA}	96	°C/W

平均片上结合处温度 (T_J) (°C) 从等式 A-1 中得到:

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

其中:

T_A = 周边温度, °C

θ_{JA} = 封装热阻, 与周边结合处, °C /W

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

$P_{int} = I_{DD} \times V_{DD}$, 瓦特——片内功率

$P_{I/O}$ = 输入和输出引脚上的功耗——由用户决定

在大多数应用中, $P_{I/O} \ll P_{int}$, 可以忽略。 P_D 和 T_J (如果忽略 $P_{I/O}$) 的近似关系是:

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

解等式 A-1 和等式 A-2, 得 K:

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

其中 K 是与特殊部分相关的常量。已知 T_A ， K 可以在等式 A-3 中通过测量 PD （在平衡时）确定。对于任何 T_A ，使用该 K 的值，通过解等式 A-1 和等式 A-2 可以得到 P_D 和 T_J 。

A.5 ESD 保护和闭锁抗扰度

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 AEC-Q100 强制测试资格。芯片的强制测试资格，通过人体模型（HBM），机器模型（MM）和电荷设备模型（CDM）完成。

如果通过 ESD 脉冲测试后，芯片被认定有问题，则该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。

表 A-4. ESD 和闭锁测试条件

模型	描述	标志	值	单位
人体	串联电阻	R1	1500	Ω
	存储电容	C	100	pF
	每个引脚的脉冲数	—	3	—
机器	串联电阻	R1	0	Ω
	存储电容	C	200	pF
	每个引脚的脉冲数	—	3	—
闭锁	最小输入电压限制	—	-2.5	V
	最大输入电压限制	—	7.5	V

表 A-5. ESD 和闭锁保护特性

编号	额定值 ¹	标志	最小	最大	单位
1	人体模型（HBM）	V_{HBM}	± 2000	—	V
2	机器模型（MM）	V_{MM}	± 200	—	V
3	电荷设备模型（CDM）	V_{CDM}	± 500	—	V
4	在 $T_A=85^\circ\text{C}$ 时，闭锁电流 (除了引脚 9 PTC3/ADP11 适用与所有引脚)	I_{LAT}	$\pm 100^2$	—	mA
	在 $T_A=85^\circ\text{C}$ 时，闭锁电流 (适用与引脚 9 PTC3/ADP11)	I_{LAT}	$\pm 75^3$	—	mA

¹ 除非另有说明，参数从典型条件的典型设备下的小型样例的设计特性取得。

² 这些引脚满足 JESD78A 第 II 类（第 1.2 节）A 级（第 1.3 节） ± 100 mA 的要求。

³ 这个引脚满足 JESD78A 第 II 类（第 1.2 节）A 级（第 1.3 节） ± 75 mA 的特性。

A.6 DC 特性

这部分包含关于电源要求，I/O 引脚的特性，和各种工作模式下电源电流的信息。

表 A-6. DC 特性（温度范围：-40 到 85 °C 的周围的）

参数	标志	最小	典型	最大	单位
电源电压（运行，等待，停止模式） $0 < f_{\text{BUS}} < 10\text{MHz}$	V_{DD}	2.7	—	5.5	V
连接到 V_{DD} 的 RAM 最小保持电压	V_{RAM}	0.8 ¹	—	—	V
低电压检测阀 V_{DD} 上升 V_{DD} 下降	V_{LVD}	—	1.86 1.94	—	V
RESET((POR) 电压的电源	V_{POR} ¹	0.9	—	1.7	V
输入高电压 ($V_{\text{DD}} > 2.7\text{V}$) (所有的数字输入)	V_{IH}	$0.70 \times V_{\text{DD}}$	—	—	V
输入高电压 ($V_{\text{DD}} > 2.7\text{V}$) (所有的数字输入)	V_{IL}	—	-	$0.3 \times V_{\text{DD}}$	V
输入滞后 (所有的数字输入)	V_{hys} ¹	$0.06 \times V_{\text{DD}}$	-	—	V
输入泄露电流 (每个引脚) $V_{\text{in}} = V_{\text{DD}}$ 或 V_{SS} 仅输入引脚	$ I_{\text{in}} $	—	0.025	—	μA
高阻抗（掉电状态）泄露电流 $V_{\text{in}} = V_{\text{DD}}$ 或 V_{SS} 所有输入 / 输出	$ I_{\text{OZ}} $	—	0.025	—	μA
内部上拉电阻 ² (所有端口引脚)	R_{PU}	—	45	—	$\text{k}\Omega$
内部下拉电阻 ² (除 PTA5 的所有端口引脚)	R_{PD}	—	45	—	$\text{k}\Omega$
PTA5 的内部下拉电阻	—	45	—	95	$\text{k}\Omega$
输出高电压驱动 (PTxDSn = 0) 5 V, $I_{\text{load}} = 2\text{mA}$	V_{OH}	$V_{\text{DD}} - 0.8$	—	—	V
输出高电压驱动 (PTxDSn = 1) 5 V, $I_{\text{load}} = 5\text{mA}$			—	—	
所有端口引脚的最大 I_{OH}	$ I_{\text{OHT}} $	—	—	40	mA
输出低电压驱动 (PTxDSn = 0) 5 V, $I_{\text{load}} = 2\text{mA}$	V_{OL}	—	—	0.8	V
输出低电压驱动 (PTxDSn = 1) 5 V, $I_{\text{load}} = 5\text{mA}$			—	0.8	
所有端口引脚的最大 I_{OL}	I_{OLT}	—	—	40	mA
直流吸纳电流 ^{3, 4, 5, 6} $V_{\text{IN}} < V_{\text{SS}}$, $V_{\text{IN}} > V_{\text{DD}}$ 单一引脚极限 全部 MCU 极限, 包含压力引脚总和		—	—	0.2 0.8	mA mA
输入电容 (所有的非电源引脚)	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: $V_{\text{in}} = V_{\text{SS}}$ 对应于上拉和 $V_{\text{in}} = V_{\text{DD}}$ 对应于下拉。

³ 所有非电源功能引脚内部钳住至 V_{SS} 和 V_{DD} , RESET/ V_{PP} 除外, 它内部钳住至 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}) 钳位电压计算阻抗值, 取两个阻抗值中的更大的。

⁵ 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}) 钳位电压计算阻抗值, 取两个阻抗值中的更大的。

6 每片芯片的参数被描述但没有被测试。

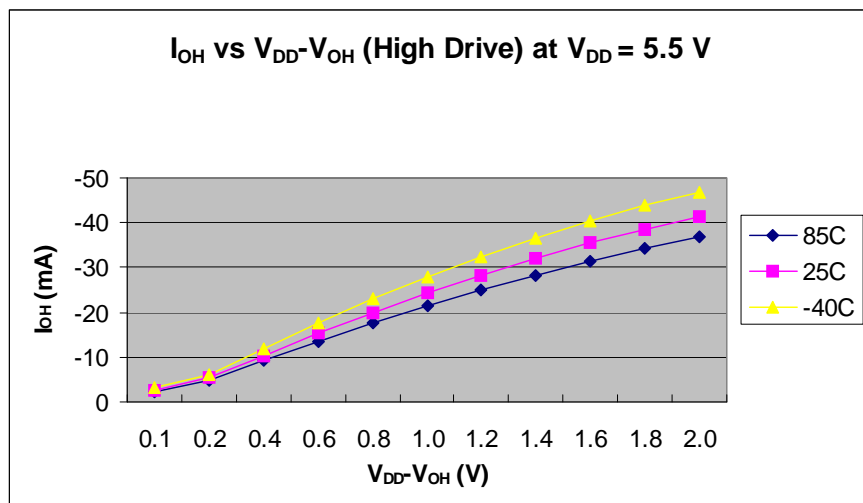


图 A-1. 典型的 I_{OH} 对 $V_{DD}-V_{OH}$ $V_{DD}=5.5$ V (高驱动)

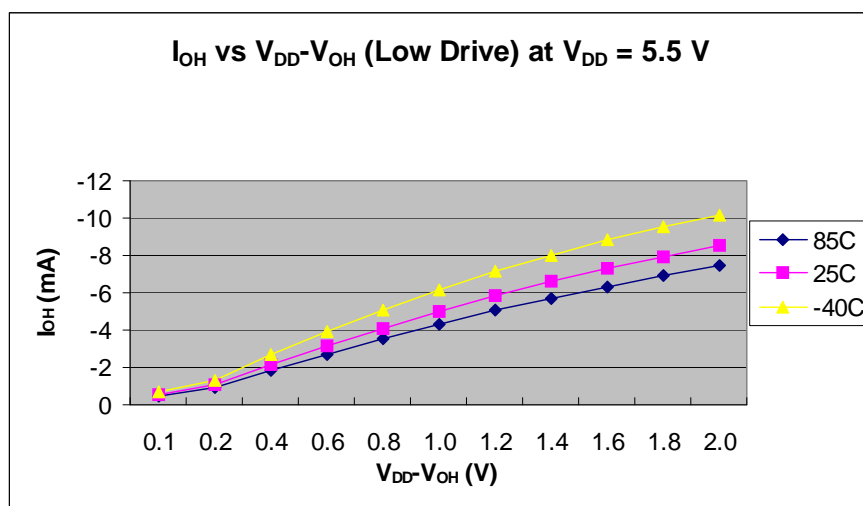


图 A-2. 典型的 I_{OH} 对 $V_{DD}-V_{OH}$ $V_{DD} = 5.5$ V (低驱动)

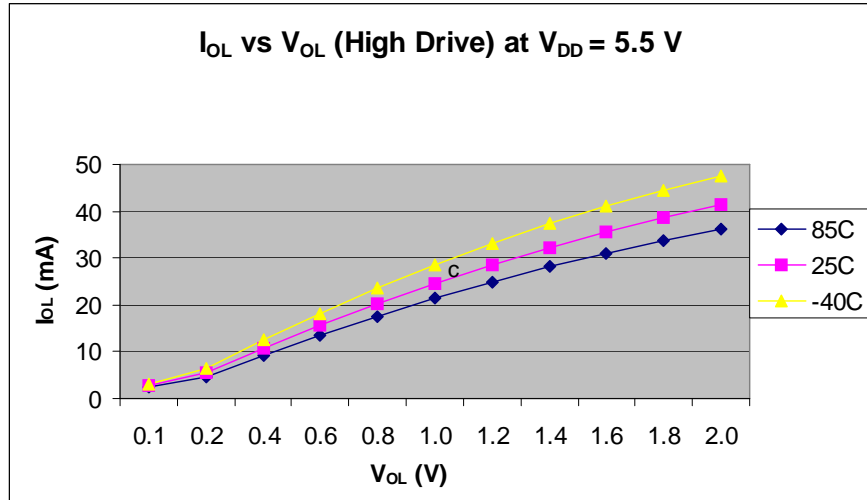


图 A-3. 典型的 I_{OL} 对 V_{DD}-V_{OH} V_{DD} = 5.5 V (高驱动)

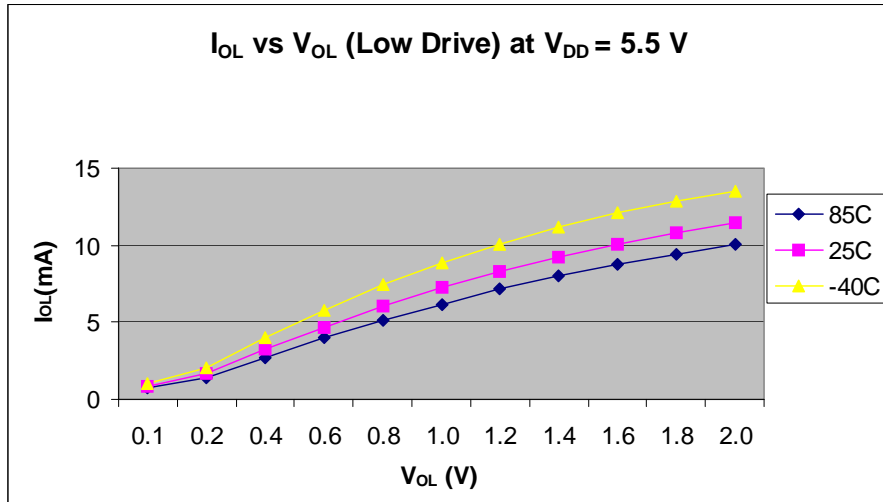


图 A-4. 典型的 I_{OL} 对 V_{DD}-V_{OH} V_{DD} = 5.5 V (低驱动)

A.7 电源电流特性

表 A-7. 电源电流特性

参数	标志	V _{DD} (V)	典型 ¹	温度 (°C)
在 (f _{Bus} = 10MHz) 下计算的工作电源电流 ²	R _{IDD10}	5	2.4mA	25 85
在 (f _{Bus} = 1.25MHz) 下计算的工作电源电流 ³	R _{IDD1}	5	0.42mA	25 85
停止模式的电源电流 ³	S _{IDD}	5	2.4 μA	25 85
ADC 加法电路工作的停止模式 ⁴	—	5	128 μA	25 85

表 A-7. 电源电流特性 (continued)

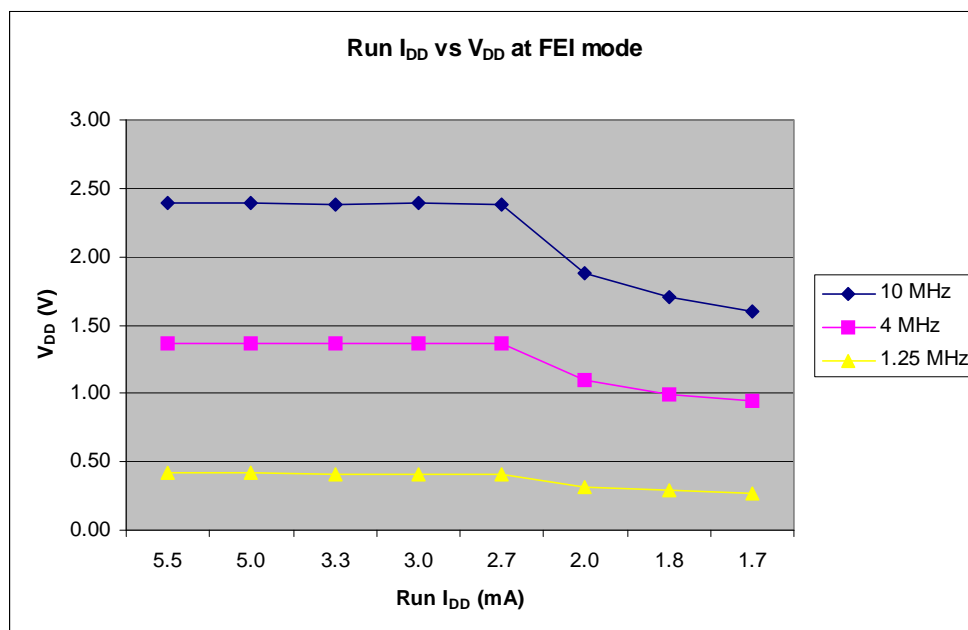
参数	标志	V _{DD} (V)	典型 ¹	温度 (°C)
ACMP 加法电路工作的停止模式 (ACME = 1)	—	5	21μA	25 85
RTI 加法电路工作的停止模式 使用 1kHz 时钟 ⁴	—	5	2.4μA	25 85
RTI 加法电路工作的停止模式 使用 1MHz 外置时钟源参考	—	5	2.1μA	25 85
LVI 加法电路工作的停止模式 (LVDE=1 和 LVDSE=1)	—	5	70μA	25 85

¹ 典型值是在 25 °C 被测量的。

² 在端口引脚上不包含任何 DC 负载。

³ 所需的异步 ADC 的时钟和 LVD 必须启用。

⁴ 大多数消费者希望发现从停止自动唤醒可以用来代替更高电流等待模式。等待模式通常是 1.3 mA, 3 V, 和 1 mA, 2V, f_{BUS} = 1 MHz。

图 A-5. FEI 模式典型运行 I_{DD} 对 V_{DD}

A.8 外部振荡器 (XOSC) 特性

表 A-8. 振荡器的电气特性 (温度范围 =-40 到 85 °C)

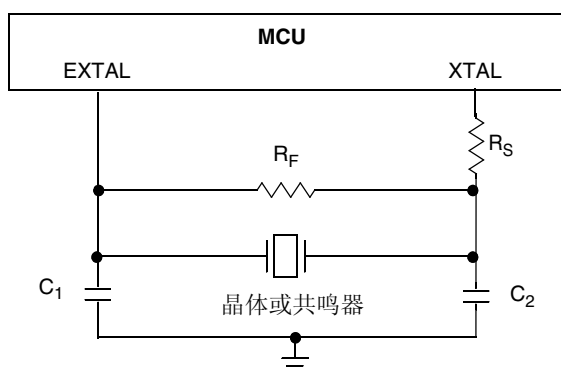
序号	C	额定值	标志	最小	典型 ¹	最大	单位
1	C	振荡器晶体或谐振器 (EREFS=1,ERCLEKN=1) 低范围 (RANGE=0) 高范围 (RANGE=1) FEE 或 FBE 模式 ² 高范围 (RANGE=1, HGO=1) FBELP 模式 高范围 (RANGE=1, HGO=0) FBELP 模式	f_{lo} f_{hi} f_{hi-hgo} f_{hi-lp}	32 1 1 1	—	38.4 5 16 8	kHz MHz MHz MHz
2	D	负载电容	C_1, C_2	见晶体或共鸣器制造商的建议	—	—	—
3	D	反馈电阻 低范围 (32kHz 到 100Hz) 高范围 (1MHz 到 16MHz)	R_F	—	10 1	—	M Ω
4	D	串联电阻 低范围, 低收益 (RANGE=0, HGO=0) 低范围, 高收益 (RANGE=0, HGO=1) 高范围, 低收益 (RANGE=1, HGO=0) 高范围, 高收益 (RANGE=1, HGO=1) $\geq 8\text{MHz}$ 4MHz 1MHz	R_S	—	0 100 0 0 0 0	—	
5	C	晶体开始时间 ³ 低范围, 低收益 (RANGE=0, HGO=0) 低范围, 高收益 (RANGE=0, HGO=1) 高范围, 低收益 (RANGE=1, HGO=0) ⁴ 高范围, 高收益 (RANGE=1, HGO=1) ⁴	$t_{CSTL-LP}$ $t_{CSTL-HGO}$ $t_{CSTH-LP}$ $t_{CSTH-HGO}$	—	200 400 5 —	—	ms
6	D	方波输入脉冲频率 (EREFS=1,ERCLEKN=1) FEE 或 FBE 模式 ² FBELP 模式	f_{extal}	0.3125 0	—	5 40	MHz

¹ 典型数据是在 5.0 V, 25 °C 时, 或是建议值。

² 输入时钟源, 必须除以使用 RDIV 以内的范围, 31.25 kHz 至 39.0625kHz。

³ 每片芯片此参数被描述但没有被测试。适当的 PC 板布局的程序, 必须遵循以实现规格。

⁴ 4 MHz 晶体。



A.9 AC 特性

本节描述了每个外设系统的 AC 时序特性。

A.9.1 控制时序

表 A-9. 控制时序

编号	C	参数	标志	最小	典型	最大	单位
1	D	总线频率 ($t_{cyc} = 1/f_{Bus}$)	f_{Bus}	0	—	10	MHz
2	D	实时中断内部振荡器周期	t_{RTI}	—	1000	—	μs
3	D	外部复位脉宽 ¹	t_{extrst}	150	-	—	ns
4	D	端口上升和下降时间 ($load = 50 pF$) ² 禁止回速率控制 (PTxSE=0) 允许回速率控制 (PTxSE=1)	t_{Rise} , t_{Fall}	- -	11 35	—	ns

¹ 这是能确保通过引脚通过输入滤波电路的最小脉冲宽度。更窄的脉冲可能不被检测到。

² 时序以 V_{DD} 20 % 到 80 % 电平显示。工作温度范围 -40°C 至 85°C 之间。

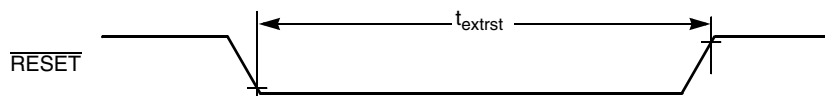


图 A-6. 复位时序

A.9.2 TPM/MTIM 模块时序

同步器电路确定可识别的最短的输入脉冲或可被用来作为对定时器，计数器可选的外部来源最快的时钟。同步器的工作来自于电流总线额定时钟。

表 A-10. TPM 输入时间

序号	C	额定值	标志	最小	最大	单位
1	D	外部时钟频率	f_{TPMext}	DC	$f_{Bus}/4$	MHz
2	D	外部时钟周期	t_{TPMext}	4	-	t_{cyc}
3	D	外部时钟上升时间	t_{clkh}	1.5	-	t_{cyc}
4	D	外部时钟下降时间	t_{clkl}	1.5	-	t_{cyc}
5	D	输入捕捉脉冲宽度	t_{ICPW}	1.5	-	t_{cyc}

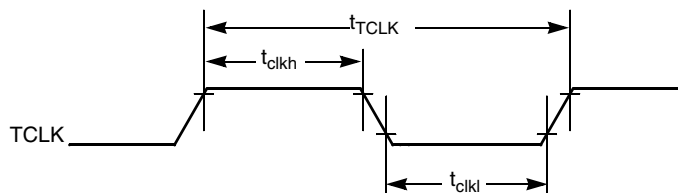


图 A-7. 定时外部脉冲

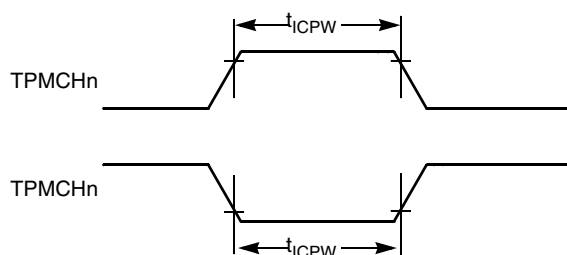


图 A-8. 定时输入捕捉脉冲

A.10 模拟比较器（ACMP）电气

表 A-11. 模拟比较器电气详述

序号	C	特性	标志	最小	典型	最大	单位
1	D	电源电压	V_{DD}	2.7	—	5.5	V
2	P	电源电流（工作状态）	I_{DDAC}	—	20	35	μA
3	D	模拟输入电压 ¹	V_{AIN}	$V_{SS} - 0.3$	—	V_{DD}	V
4	P	模拟输入偏压 ¹	V_{AIO}	—	20	40	mV
5	C	模拟比较器滞后 ¹	V_H	3.0	9.0	15.0	mV
6	C	模拟源阻抗 ¹	R_{AS}	—	—	10	$\text{k}\Omega$
7	P	模拟输入泄漏电流	I_{ALKG}	—	—	1.0	μA
8	C	模拟比较器初始化延迟	t_{AINIT}	—	—	1.0	μs
9	P	模拟比较器的带隙基准参考电压	V_{BG}	1.1	1.208	1.3	V

¹ 这些数据都是描述的并不是生产测试的。

A.11 内部时钟源特性

表 A-12. 内部时钟源详述

编号	C	特性	标志	最小值	类型 ¹	最大值	单位
1	C	内部平均参考频率—未校准的	f_{int_ut}	25	31.25	41.66	KHz
2	P	内部平均参考频率—校准的	f_{int_t}	31.25	39.06	39.625	KHz
3	C	DCO 输出频率范围—未校准的	f_{dco_ut}	12.8	16	21.33	MHz
4	P	DCO 输出频率范围—校准的	f_{dco_t}	16	20	20	MHz
5	C	在固定电压和温度下校准的 DCO 输出频率精度	$\Delta f_{dco_res_t}$	—	—	0.2	% f_{dco}
6	C	在过电压和温度下校准的 DCO 输出频率总偏移	Δf_{dco_t}	—	—	2	% f_{dco}
7	C	FLL 获取时间 ^{2, 3}	$t_{acquire}$	—	—	1	ms
8	C	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	—	μ s

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

² 参数被描述但不是每片都被测试过。

³ 此规范适用于任何时候，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.12 ADC 特性

表 A-13. 8 位 ADC 工作条件

C	特性	条件	标志	最小	典型	最大	单位
D	输入电压	—	V_{ADIN}	V_{SS}	—	V_{DD}	V
C	准确值	$V_{DD} = 2 V$	—	—	8 位	—	—
C	输入电容	—	C_{ADIN}	—	4.5	—	pF
C	输入电阻	—	R_{ANIN}	—	3	—	k Ω
C	MCU 外部模拟源电阻	8 位模型（全部有效 f_{ADCK} ）	R_{AS}	—	—	10	k Ω
D	ADC 转化时钟频率	高速度（ADLPC=0）	f_{ADCK}	0.4	—	8.0	MHz
		低功耗（ADLPC=1）	—	0.4	—	8.0	—

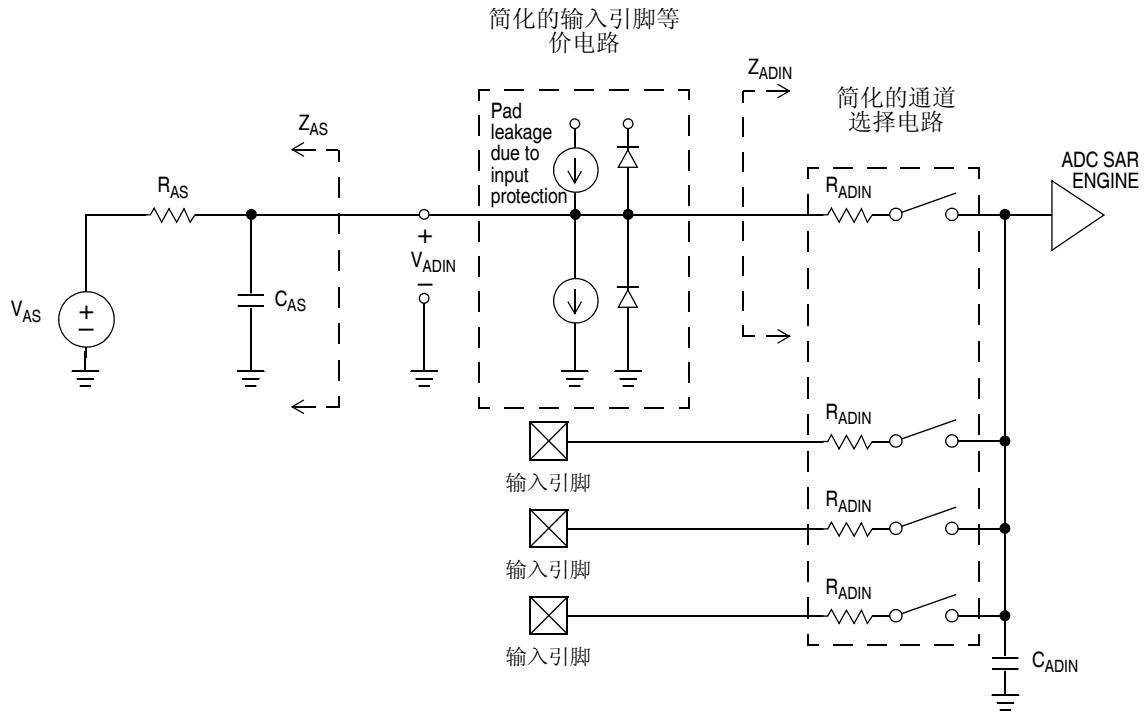


图 A-9. ADC 输入阻抗等价图

表 A-14. 8 位 ADC 特性

特性	条件	C	标志	典型 ¹	单位
电源电流 ADLPC=1 ADLSMP=1 ADCO=1	—	T	I _{DDAD}	133	μA
电源电流 ADLPC=1 ADLSMP=0 ADCO=1	—	T	I _{DDAD}	218	μA
电源电流 ADLPC=0 ADLSMP=1 ADCO=1	—	T	I _{DDAD}	327	μA
电源电流 ADLPC=0 ADLSMP=0 ADCO=1	—	C	I _{DDAD}	0.582	mA
电源电流	停止, 复位, 模块关	T	I _{DDAD}	0.011	μA
ADC 异步时钟源	高速度 (ADLPC=0)	T	f _{ADACK}	3.3	MHz
	低功耗 (ADLPC=1)			2	
转换时间 (包含采样时间)	短样本 (ADLSMP=0)	P	t _{ADC}	20	ADCK 周期
	长样本 (ADLSMP=1)			40	

表 A-14. 8 位 ADC 特性

特性	条件	C	标志	典型 ¹	单位
采样时间	短样本 (ADLSMP=0)	P	t _{ADS}	3.5	ADCK 周期
	长样本 (ADLSMP=1)			23.5	
全部未校准的错误	8 位模式	C	E _{TUE}	± 0.5	LSB ²
微分非线性	8 位模式	P	DNL	± 0.3	LSB ²
	单调性和无失码保证				
积分非线性	8 位模式	C	INL	± 0.5	LSB ²
零尺度误差	8 位模式	P	E _{ZS}	± 0.5	LSB ²
全尺度误差	8 位模式	P	E _{FS}	± 0.5	LSB ²
量化错误	8 位模式	D	E _Q	± 0.5	LSB ²
输入漏错误垫泄漏 ³ R _{AS}	8 位模式	D	E _{IL}	± 0.2	LSB ²

¹ 典型值假定 Temp=25 °C, f_{ADCK}=1.0 MHz 除非在其它状态。典型值仅作参考并不测试。

² 1 LSB = (V_{REFH} - V_{REFL})/2^N

³ 基于对输入端的漏电流。指垫电器

A.13 Flash 规格说明

本节详细描述了 Flash 存储器编程 / 擦写时间及编程 - 擦写强度的详细描述。更多关于编程 / 擦写操作的信息，参见 4.6 节 Flash。

表 A-15. Flash 特性

特性	标志	最小	典型 ¹	最大	单位
编程 / 擦写电源电压	V _{DD}	2.7	-	5.5	V
编程 / 擦写电压	V _{PP}	11.8	12	—	V
V _{PP} 电流 编程 块擦除	I _{VPP_prog} I _{VPP_erase}	—	—	200 100	μA
读操作电源电压 (0 < f _{BUS} < 10MHz)	V _{Read}	2.7	—	5.5	V
字节编程时间	t _{prog}	20	—	40	μs
块擦除时间	t _{me}	500	—	—	ms
累积编程 HV 时间 ²	t _{hv}	—	—	—	ms
总累积编程 HV 时间 (应用于芯片的 t _{me} 与 t _{hv} 总和)	t _{hv_total}	—	—	—	Hours
HVED 到编程建立时间	t _{pgs}	10	—	—	μs
PGM/MASS 到 HVEN 的建立时间	t _{nvS}	5	—	—	μs
对于 HVEN, PGM 的保持时间	t _{nvh}	5	—	—	μs
对于 HVEN, MASS 的保持时间	t _{nvh1}	100	—	—	μs

表 A-15. Flash 特性

V _{PP} 到 PGM/MASS 的建立时间	t _{vps}	20	—	—	ns
HVEN 到 V _{PP} 的保持时间	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 / 擦写强度 T _L 到 T _H =-40 °C 到 85 °C	—	1000	—	—	周期
数据保持 5	t _{D_ret}	15	—	—	年

¹ 典型值是在 25 °C 测量的。

² t_{hv} 是对同一行在下次擦除前累积的高电压编程时间。同一地址在下次擦写前不能编程两次。

³ 快速的 V_{PP} 上升时间可能潜在的触发 ESD 保护结构，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} 滤波举例如表 A-10。

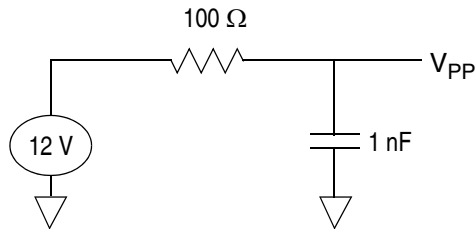
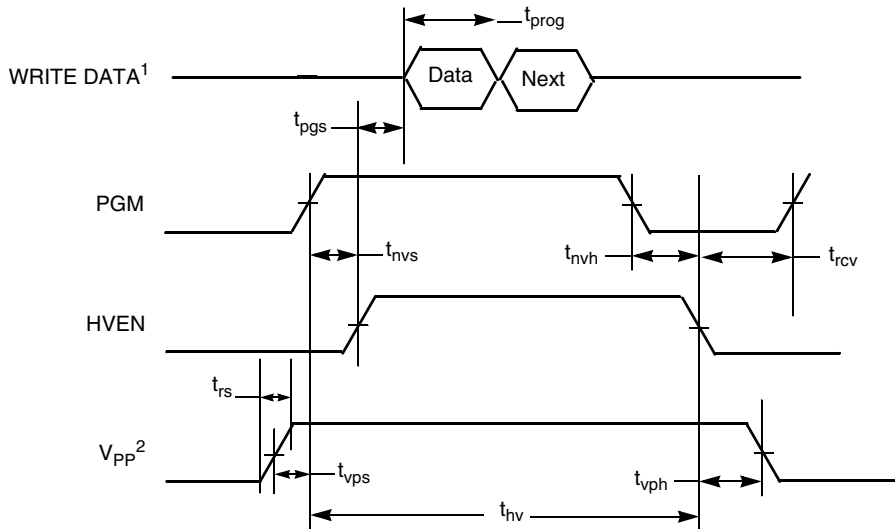


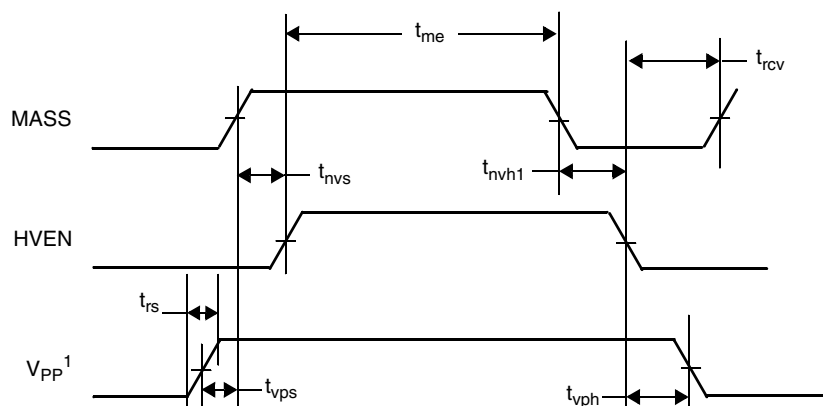
图 A-10. V_{PP} 滤波实例



¹ 如果编程多个字节在同一行，下一个数据可以使用，参考 4.6 节，“Flash”。

² 电压在 V_{PP} 引脚使用或移开之前，V_{DD} 必须是一个有效的工作电压。

图 A-11. Flash 编程时序



¹ 在电压应用或删除到 VPP 引脚之前 VDD 必须是合法的工作电压。

图 A-12. Flash 块擦除时序

A.14 EMC 性能

电磁兼容性的（EMC）性能高度依赖于 MCU 存在的环境。板的设计和布局，电路拓扑选择，外部组件的位置和特性，以及 MCU 的软件运行对 EMC 性能发挥有重要作用。系统设计者应该参考飞思卡尔的应用手册，像 AN2321,AN1050,AN1263,AN2764 以及 AN1259 的专门针对最优 EMC 性能的意见和指导。

A.14.1 辐射排放量

微控制器辐射 RF 的排放量根据 IEC61967-2 和 SAEJ1752/3 标准使用 TEM/GTME 单元方式在 150kHz 到 1GHz 测量。测量是在安装了自定义的 EMC 评估电路板微控制器上执行的同时，运行专门的 EMC 测试软件。来自微控制器的辐射排放量在 TEM 单元两个封装方向（北部和东部）测量。

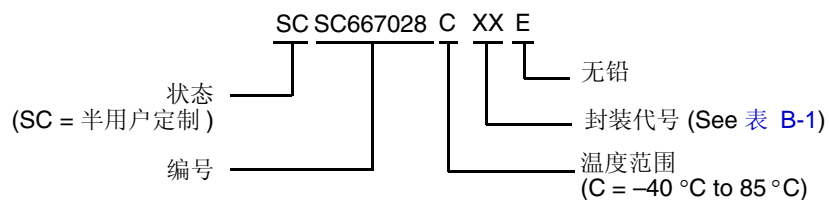
附录 B 订购信息

B.1 订购信息

本节包含了 SC667028 芯片的订货单号。芯片单号系统例子见下面。

表 B-1. 设备编号系统

芯片编号	存储器		封装		
	Flash	RAM	类型	设计者	文档号
SC667028	8 KB	254 B	20 W-SOIC	WJ	98ASB4243B



B.2 机械制图

这以下页面包含 SC667028 封装的机械规格说明。20 引脚 W-SOIC（宽体小外形集成电路）。

如何联系我们:

主页:
www.freescale.com

电子邮件:
support@freescale.com

美国 / 欧洲或未列出的地方:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274 or +1-480-768-2130
support@freescale.com

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

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

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

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

文档号: SC667028CHS
第 1 版
2008 年 9 月

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

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

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

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

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