



Freotech 北京飞锐泰克科技有限公司

MC9RS08KA8

MC9RS08KA4

数 据 手 册

2008.4

Ver1.0

MC9RS08KA8 微控制器简介

8 位 RS08 中央处理器(CPU)

- 20MHz CPU 时钟
- 电源电压 1.8V~5.5V
- 工作温度范围: -40°C~85°C
- 支持 HC08 指令集+BGND 指令

存储器

- 8 KB FLASH 存储器可在全工作电压和温度内擦写
- KA4 为 4 KB FLASH
- 254 字节随机存取存储器(RAM)
- KA4 为 126 字节 RAM
- RAM 和 Flash 具有防止未经授权访问的机制

节电模式

- 等待模式和停止模式
- 实时时钟中断(RTI), KBI 或 ACMP 可唤醒 CPU

时钟源

- 振荡器(XOSC) 一环控制精确振荡器;晶体和陶瓷谐振器的频率范围分别为 31.25KHz~39.0625KHz 和 1MHz~5MHz
- 内部时钟源(ICS) 一包含一个由内部参考或外部参考时钟控制的锁频环(FLL)。内部参考时钟精度可调,分辨率 0.2%;全工作电压和温度范围内仅 2%误差;总线频率高达 10MHz

系统保护机制

- 看门狗复位
- 具有复位和中断功能的低电压检测模块
- 非法操作码错误检测复位
- 非法地址检测复位
- FLASH 块安全保护机制

开发支持

- 单线后台调试接口,支持断点

片上外设模块

- ADC—12 通道,10 位分辨率,2.5 μ s 转换时间;自动比较功能;可在 CPU 停止模式下工作;工作电压 2.7V~5.5V(16 引脚封装型号只有 8 通道可用)
- TPM—2 通道;每一个通道可独立选择为输入捕捉、输出比较或带缓冲的边沿/中心对齐 PWM
- IIC—速率可达 100kbps

- MTIM1, MTIM2—两个 8 位模计数器
- KBI—键盘中断,可设置为上升或下降沿检测;16 引脚和 20 引脚封装的型号有 8 个 KBI 端口
- ACMP—模拟比较器;工作范围为芯片全工作电压;可选择与内部固定能隙参考电压相比较;可在停止模式下工作

输入/输出

- 14/18 个通用输入/输出(I/O)引脚,其中有一个只输入引脚和一个只输出引脚
- 用作输入端时,端口可软件选择上拉;用作输出端时,端口可软件选择驱动能力和压摆率(slew rate)

封装

- 16 引脚 SOIC 或 PDIP
- 20 引脚 SOIC 或 PDIP

目 录

第一章 MC9RS08KA8 概述.....	4
第二章 引脚及连接	6
第三章 工作模式	9
第四章 存储器	11
第五章 复位，中断，系统运行控制.....	15
第六章 并行输入/输出控制	20
第七章 键盘中断(KBI)	23
第八章 CPU	25
第九章 模拟比较器	28
第十章 10 位 ADC	30
第十一章 内部时钟源	44
第十二章 IIC	49
第十三章 模计时器	60
第十四章 16 位计时/PWM.....	63
第十五章 开发工具	72

第一章 MC9RS08KA1/2 概述

1.1 概述

MC9RS08KA8 单片机是超低功耗，小引脚数量的集成电路，广泛应用于家居，玩具等领域。由 RS08 内核，254 字节 RAM，8Kflash，两个 8 位模计时器，12 通道 10 位 ADC，2 通道 16 位计时/PWM，IIC 总线模块，键盘中断，模拟比较器组成。该器件为 16 和 20 引脚封装。

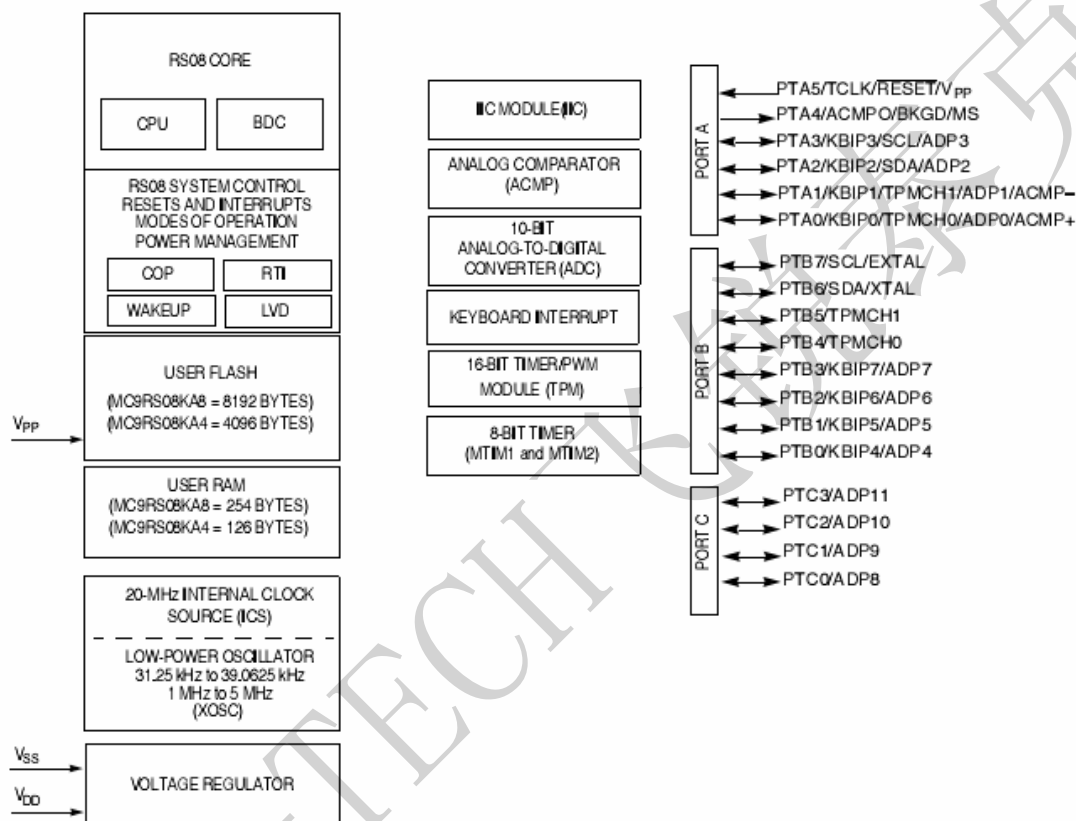


图 1—1MC9RS08KA8 结构图

1.2 系统时钟分布

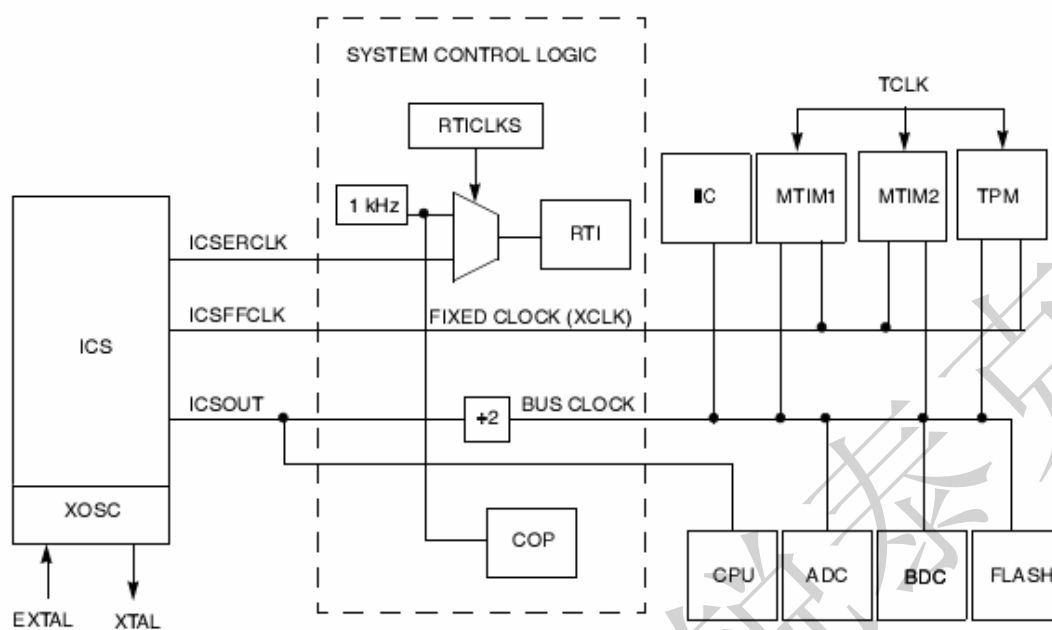


图 1-2 系统时钟分配图

注：总线频率是 ICS 输出频率的 1/2，MCU 内部模块均使用总线频率。

第二章 MC9RS08KA8 引脚特性

2.1 MC9RS08KA8 引脚概述

本章介绍该芯片的引脚电气特性。

2.2 MC9RS08KA8 引脚图

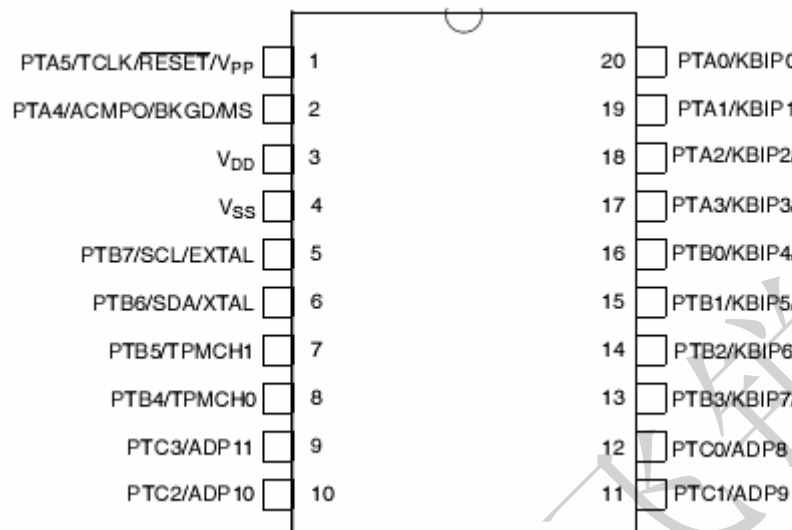


图 2-1 20 引脚 SOIC/PDIP 封装

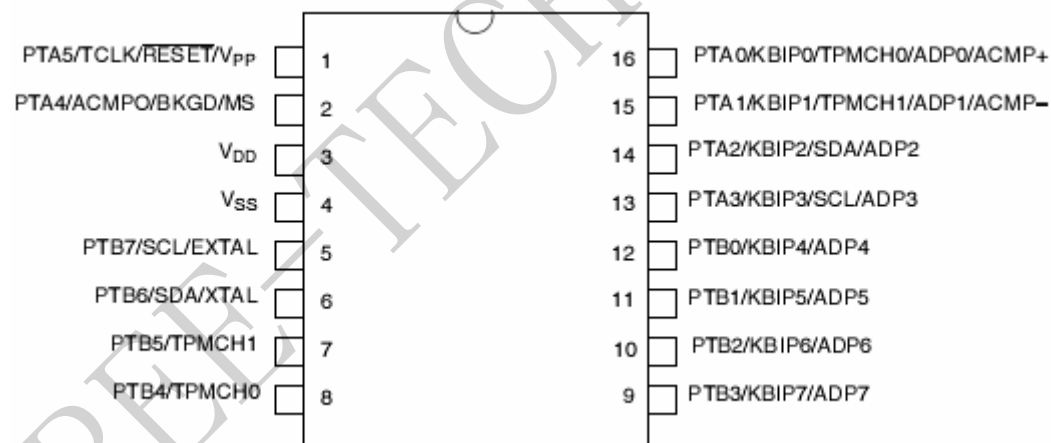


图 2-2 16 引脚 SOIC/PDIP 封装

2.3 芯片应用原理图

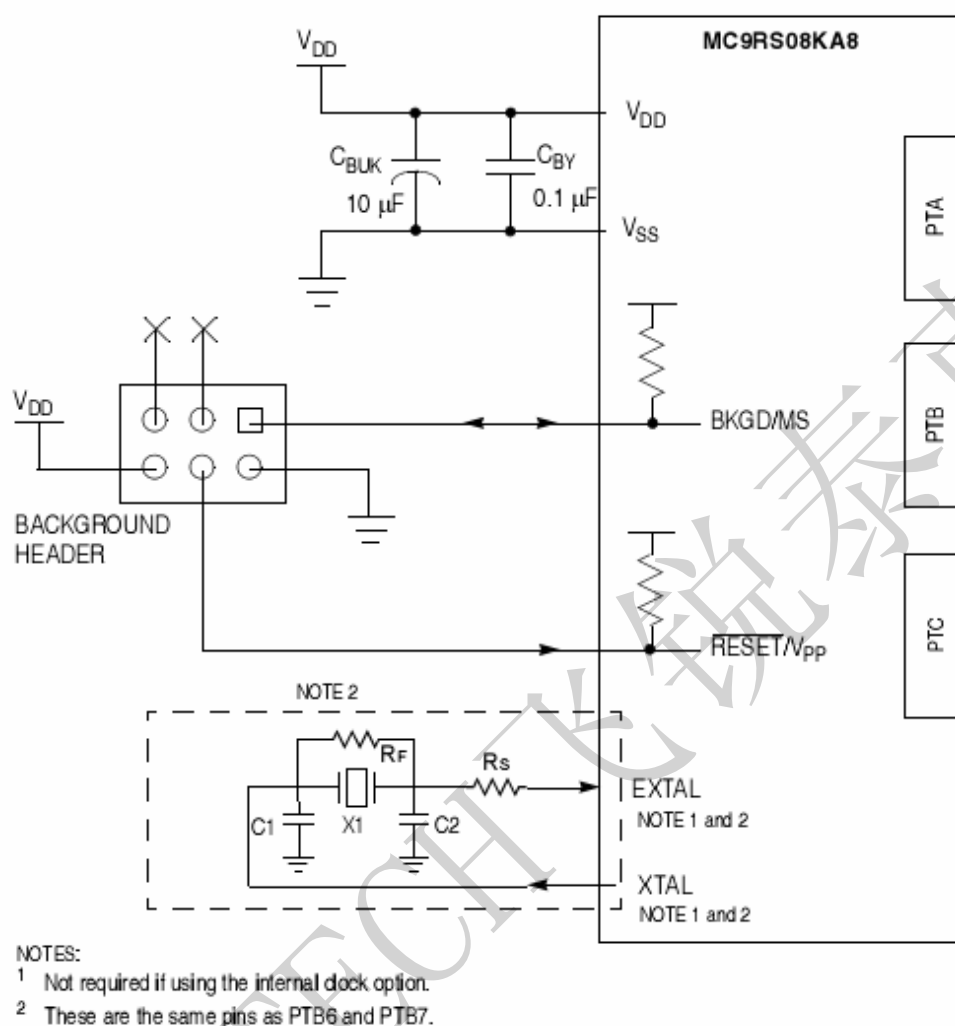


图 2—4 推荐芯片应用原理图

2.4 引脚电气特性

2.4.1 电源

V_{DD}, V_{SS} 是 MCU 的供电引脚，该电源为引脚缓冲电路和内部稳压器提供电源。

内部稳压器为 CPU 和其他模块提供电源。

典型应用中 V_{DD}, V_{SS} 之间放置两个电容，其中一个大电容（10 微法钽电容），在系统断电后为整个系统提供维持电压；另一个 0.1 微法的瓷片电容尽量靠近 V_{DD}, V_{SS} 用来抑制高频噪声。

2.4.2 PTA2/TCLK/RESET(低有效)/VPP 引脚

电源复位(POR)之后，进入用户模式，该引脚默认通用输入引脚。

设置 SOPT 寄存器中的 RSTPE=1，可以将该引脚配置为复位功能引脚且内部上拉自动使能。一旦该引脚设置为复位输入，该引脚始终担当复位输入引脚功能，直到下次上电复位。当作复位输入作用时该引脚可以由外部信号拉低来复位 MCU。

VPP 为芯片编程或擦除时提供电源，不论该引脚配置成何种作用，VPP 始终连接到内部 FLASH 模块。当芯片不处于编程或擦除状态时，应断开外部电压，并且施加到该引脚信号不应超过 V_{DD}，以免对 FLASH 造成误操作。

注：该引脚没有限压二极管保护，故未处于编程或擦除状态下不要施加大于 V_{DD} 的信号。

2. 4. 3PTA4/ACMP0/BKGD/MS

该引脚担当 3 个作用：1) 输出引脚 2) 可选的模拟比较输出 3) 背景调试/模式选择。一旦进入复位状态，该引脚担当模式选择作用。

复位信号变高之后，该引脚立刻担当背景调试通信引脚。在背景调试和模式选择状态下，内部上拉会自动使能。

设置 SOPT 寄存器中的 BKGDPE=0，可使该引脚担当输出引脚且只有输出功能。

在复位信号变高之后，如果没有背景调试信号连接到该引脚，那么 MCU 进入用户工作模式(正常工作模式)。如果系统连接到背景调试工具接口，在复位期间，调试工具将引脚被拉低，强制 MCU 进入背景调试模式。

该引脚在背景调试模式下作为 BDC(背景调试控制器)的数据通信线。每位数据通信时间为 16 个 BDC 时钟周期，而 BDC 时钟频率与系统总线频率相同，故该引脚不应有太大的容性连接，以免影响背景调试通信。

2. 4. 4 通用 IO 和外部端口

其他引脚担当通用 IO 和比较器和定时器的作用。复位之后，这些引脚默认为高阻输入，且内部上拉/下拉未使能。

注意：为了避免外部电流灌入浮空的输入引脚，在用户应用程序的初始化部分中使能引脚的输入上拉或下拉，或者将引脚由输入状态转变为输出状态。

Pin Number		<— Lowest Priority —> Highest				
20	16	Port Pin	Alt 1	Alt 2	Alt 3	Alt 4
1	1	PTA5		TCLK	RESET	V _{PP}
2	2	PTA4	ACMPO	BKGD	MS	
3	3					V _{DD}
4	4					V _{SS}
5	5	PTB7	SCL ¹			EXTAL
6	6	PTB6	SDA ¹			XTAL
7	7	PTB5	TPMCH1 ²			
8	8	PTB4	TPMCH0 ²			
9	—	PTC3			ADP11	
10	—	PTC2			ADP10	
11	—	PTC1			ADP9	
12	—	PTC0			ADP8	
13	9	PTB3	KBIP7		ADP7	
14	10	PTB2	KBIP6		ADP6	
15	11	PTB1	KBIP5		ADP5	
16	12	PTB0	KBIP4		ADP4	
17	13	PTA3	KBIP3	SCL ¹	ADP3	
18	14	PTA2	KBIP2	SDA ¹	ADP2	
19	15	PTA1	KBIP1	TPMCH1 ²	ADP1	ACMP—
20	16	PTA0	KBIP0	TPMCH0 ²	ADP0	ACMP+

¹ IIC pins can be remapped to PTA3 and PTA2

² TPM pins can be remapped to PTA0 and PTA1

表 2—1 引脚参考表

第三章 工作模式

3.1 工作模式概述

本章介绍 MC9RS08KA8 的工作模式。包括进入和退出每种模式的方法，以及每种模式下的工作状态。

3.2 各种工作模式特点

*激活背景调试模式

*等待模式：

—CPU 停止工作

—系统时钟继续运行

—模拟比较器仍工作

*停止模式：

—系统时钟停止；稳压器工作

—所有的内部电路待机状态，以备快速恢复。

3.3 正常运行模式

在复位信号上升沿，BKGD/MS 引脚为高则进入正常工作模式。在该模式下，CPU 执行\$3FFD 开始的程序代码。所以用于一般在\$3FFD 处放 JMP XXXX 指令，跳转到主程序地址。如下表：

3FFD	3FFE	3FFF
BC(JMP 机器码)	跳转地址高字节	跳转地址低字节

3.4 激活背景调试模式

背景调试功能是由 RS08 核内的 BDC 管理的。在软件开发过程中，BDC 提供了一种分析 MCU 操作方法。

有 4 种激活背景调试的方法：

*在 POR 期间或发出背景调试强制复位命令之后，拉 BKGD/MS 引脚为低

*BKGD 引脚接收到 BACKGROUND 命令

*执行 BGND 指令

*遇到 BDC 断点

当进入背景调试模式后，CPU 处于等待接收背景调试命令，而不会执行用户的程序。两种背景调试命令：

*非打断命令：是指可以在用户程序运行时发出，由 BKGD 引脚在用户模式下发出。

MCU 处于背景调试模式下也可以执行非打断指令。非打断指令包括：

—存储器访问指令

—存储器访问状态指令

—BACKGROUND 命令

*主动背景调试指令：MCU 只有处于背景调试模式才能下执行。包括的指令：

—读取修改 CPU 寄存器

—跟踪用户程序一条指令

—退出背景调试模式

更多信息参考开发工具。

3.5 等待模式

等待模式下 MCU 运行情况表:

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

CPU 执行 WAIT 进入等待模式。执行完 WAIT 模式之后, CPU 进入低功耗模式, 它的时钟停止。PC 值保存了 WAIT 指令的位置。

当有中断发生时:

1. MCU 离开等待模式继续执行进入等待模式时的指令
2. PC 加 1, 执行 WAIT 指令后的指令。

用户程序来负责查明唤醒 MCU 的中断源。

当 CPU 处于等待模式时, 不是所有的背景调试指令能使用。只有 BACKGROUND 命令和存储器访问状态命令可用。存储器状态命令不允许访问存储器, 但是它可以报告一个错误指示, 来表示 MCU 处于停止或等待模式。BACKGROUND 命令可以唤醒 CPU, 使它进入背景调试模式。

3.6 停止模式

停止模式下 MCU 运行情况表:

Mode	CPU	Digital Peripherals	ICS ¹	ACMP ²	Regulator ³	I/O Pins	RTI	ADC ⁴
Stop	Standby	Standby	Optionally on	Optionally on	Optionally on	States held	Optionally on	Optionally on

¹ ICS requires IREFSTEN = 1 and LVDE and LVDSE must be set to allow operation in stop.

² If bandgap reference is required, the LVDE and LVDSE bits in the SPMSC1 must both be set before entering stop.

³ When BDM is enabled, the Regulator is on. Or else, only when LVDE and LVDSE bits both in the SPMSC1 to be set, Regulator is in on mode.

⁴ Requires the asynchronous ADC clock, LVDE and LVDSE bits both in the SPMSC1 to be set, otherwise ADC is in standby mode.

当 STOPE 位使能, CPU 执行 STOP 指令后 CPU 进入 STOP 模式。在该模式下, 所有时钟源停止工作。唤醒 STOP 模式的 MCU 方式有复位, 使能的异步中断, 或实时时钟中断。

如果在复位引脚施加低电平唤醒 CPU 方式, MCU 将复位, 执行 \$3FFD 处指令。

如果通过异步中断或实时时钟中断唤醒 CPU 方式, MCU 执行 STOP 后的指令。用户应查询哪种中断源唤醒 CPU。

实时时钟中断: 一个大约 1KHz 的时钟源为实时时钟提供时钟源。当 RTIS=000, 将关闭实时中断和 1KHz 时钟源, 从而降低功耗。外部时钟源也可以为处于 STOP 模式下的实时时钟提供时钟源。

3.6.1 STOP 模式下后台调试使能

在进入 STOP 模式前, 设置 ENBDM=1, 那么进入 STOP 模式后, 用于背景调试的时钟仍运行, 同时稳压器正常运行。

在 STOP 模式, 大多数背景调试命令不能使用。存储器访问状态命令不会访问存储器的内容, 而是报告 MCU 处于 STOP 模式或 WAIT 模式。BACKGROUND 命令可以唤醒处于 STOP 模式下的 CPU, 使 MCU 进入背景调试模式。

3.6.2 STOP 模式下 LVD 模块使能

当供电电压跌落到 LVD 门限电压, 低电压检测模块 (LVD) 产生中断或复位。在进入 STOP 模式前, 设置 LVDE=1, LVDSE=1, 以使 LVD 模块在 STOP 模式下仍处于工作状态。

第四章 存储器

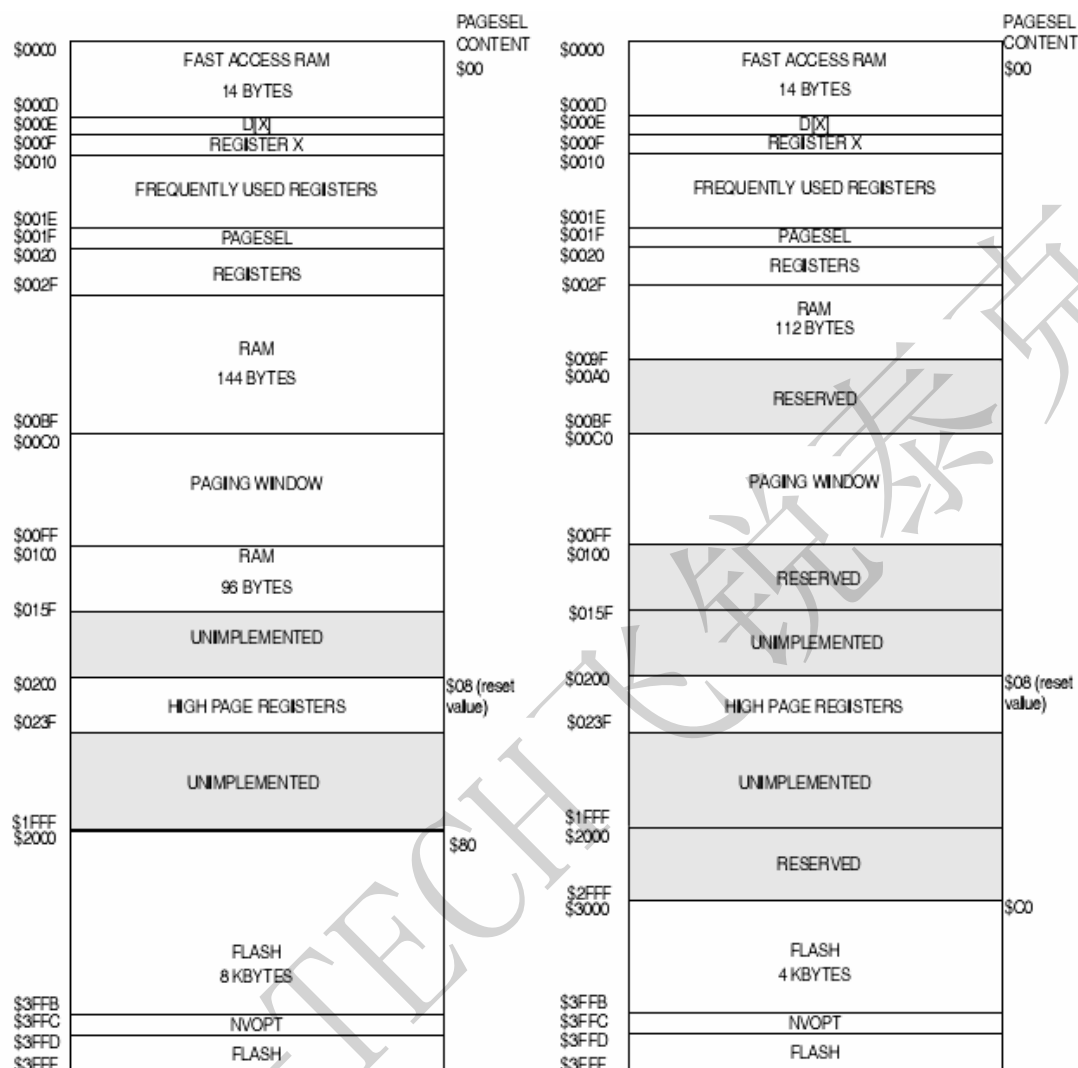


图 4—1 存储器地址映射图

4.1 地址映射

- *简短和短指令快速访问 RAM 区 (\$0000-\$00D)
- *间接数据访问寄存器 D[X] (\$000E)
- *变址寄存器 X (\$000F)
- *常用外围寄存器 (\$0010-\$001E, \$0020-\$002F)
- *PAGESEL 寄存器 (\$001F)
- *RAM (\$0030-\$00BF, \$0100-\$015F)
- *页窗口区 (\$00C0~\$00FF)
- *其他外围寄存器 (\$0200~\$023F)
- *非易失性存储区 (\$3800~\$3FFF): MC9RS08KA4
- *非易失性存储区 (\$2000~\$3FFF): MC9RS08KA8
- *未实现存储器区

4.2 变址/间接寻址模式

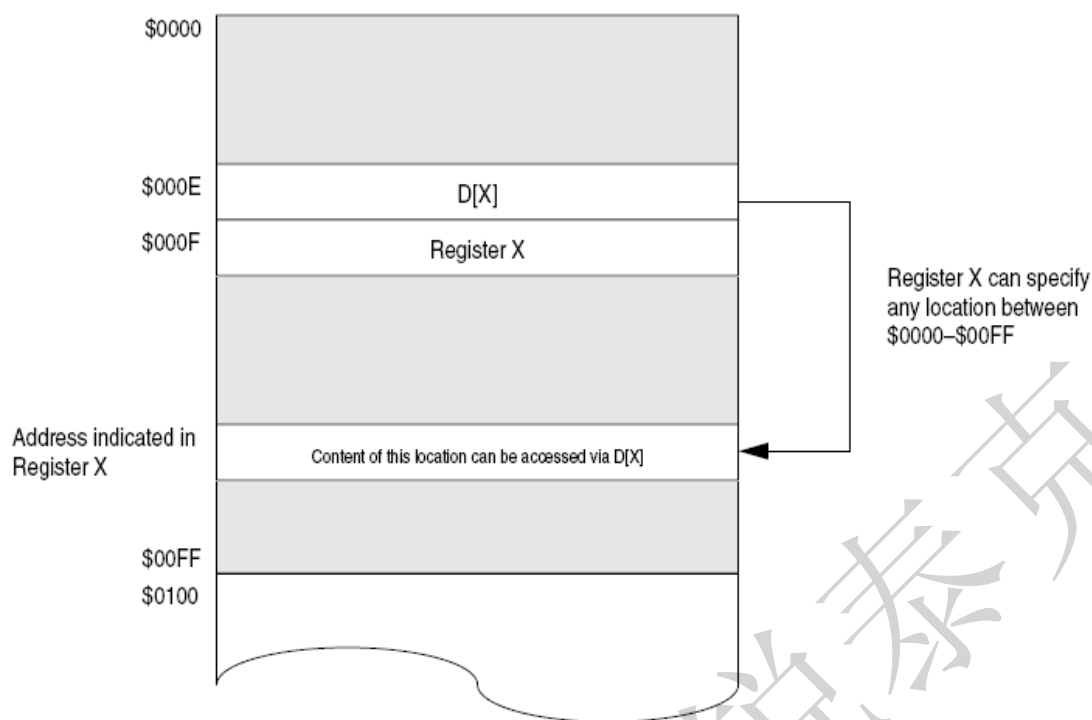


图4—2变址寻址原理图

由上图可知，X寄存器存放要访问的存储区的地址，地址范围为(\$0000~\$00FF)。改变X中的内容，D[X]中出现相应地址中的内容。

4. 3RAM和寄存器及位分配

详情参照英文文档Table4-1.

RAM中的\$0000~\$000D可以被直接访问.

当X寄存器的值为\$0E, 那么D[X]寄存器中的内容就是RAM中地址\$000E中的内容.

RAM中的\$0030~\$00BF可以通过间接寻址模式指令被访问.

当MCU处于低功耗或停止模式, RAM中的数据内部保持.

4. 4FLASH

4. 4. 1FLASH特点

1000次编程/擦除

FLASH安全保护机制

4. 4. 2FLASH页编程过程

1. 施加编程电压VPP。
2. 设置PGM=1。
3. 通过页窗口向FLASH任意位置写入任意数据。
4. 延时等待Tnvs。
5. 设置HVEN=1。
6. 延时等待Tpgs。
7. 向要编程的FLASH地址写入编程数据。
8. 延时等待Tprog。
9. 重复7, 8两步直到写完整页。
10. 清除PGM=0。
11. 延时等待Tnvh。
12. 清除HVEN=0。
13. 延时等待Trcv。
14. 断开VPP。

4. 4. 3FLASH擦除过程

1. 施加编程电压VPP
2. 设置MASS=1
3. 通过页窗口向FLASH任意位置写入任意数据
4. 延时等待Tnvs
5. 设置HVEN=1
6. 延时等待Tme
7. 清除MASS=0
8. 延时等待Tnvhl
9. 清除HVEN=0
10. 延时等待Trcv
11. 断开VPP

4. 5FLASH寄存器和控制位

4. 5. 1FLASH选择寄存器 (FOPT和NVOPT)

复位期间，NVOPT (地址\$3FFC) 中的值复制到FOPT (\$0210) 中。

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

Reset This register is loaded from nonvolatile location NVOPT during reset.

= Unimplemented or Reserved

Figure 4-3. Flash Options Register (FOPT)

SECD=0: 使能安全保护

SECD=1: 取消安全保护

4. 5. 2FLASH控制寄存器 (FLCR)

	7	6	5	4	3	2	1	0
R	0	0	0	0	HVEN	MASS	0	PGM ¹
W								

Reset 0 0 0 0 0 0 0

= Unimplemented or Reserved

Figure 4-4. Flash Control Register (FLCR)

HVEN:

=0 取消施加到FLASH阵列高电压

=1 施加FLASH阵列高电压

MASS:

=0:取消选择块擦除操作

=1:选择块擦除操作

PGM:

=0: 取消选择编程操作

=1: 选择编程操作

4. 6页选寄存器 (PAGESEL)

该寄存器内容与存储区地址对应关系如下图:

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

例如PAGESEL寄存器设置为\$08, 那么高页寄存器\$0200~\$023F被映射到\$00C0~\$00FF, 此时通过间接寻址模式指令进行读写.

第五章 复位，中断，系统运行控制

5.1 本章介绍

本章集中简单的介绍MCU所有的复位和中断源。

5.2 复位，中断特点

多种复位源：使得系统灵活可靠。

系统复位状态寄存器(SRS)：系统复位状态寄存器标识最近一次复位源。

系统未处理中断寄存器(SIP1)：标识系统未处理的中断源。包括：

-模拟比较器中断

-模计数器中断

-键盘中断

-实时时钟中断

-ADC中断

-IIC中断

-TPM中断

5.3 MCU复位

7种复位源：

1. 外部复位引脚(PIN)

2. 上电复位(POR)

3. 低电压检测(LVD)

4. 计算机正确运行计数器(COP)(看门狗)

5. 非法操作码(ILON)

6. 非法操作地址(ILAD)

7. 背景调试命令BDC_RESET

5.4 看门狗COP(WATCHDOG)

COP目的是在MCU未按用户期望运行的情况下强制MCU复位。为了阻止COP复位发生，应用程序必须定期复位COP计数器。

任何复位之后，在SOPT寄存器中的COPE=1，看门狗使能。如果应用程序中不使用看门狗功能，应清COPE=0。向SRS写入任意值，即可复位COP计数器。但写操作并不影响SRS中的内容。通过设置COPT可以选择看门狗的溢出时间。下表所示：

COPT	COP Overflow Count ¹
0	2 ⁵ cycles (32 ms)
1	2 ⁸ cycles (256 ms)

在复位初始化程序中，用户应将设定值写入SOPT寄存器，SOPT寄存器只允许写入一次，因此可以防止以后对SOPT意外修改。

在背景调试模式下，COP计数器不再增加。

一旦MCU进入STOP模式，COP计数器值变为0。而MCU离开STOP模式，COP计数器的值也变为0。

5.5 中断

中断控制器没有中断向量表查询机制。然而，中断源可以唤醒处于STOP或WAIT模式下的MCU，所以应用程序应查询发生了何种中断。

每种中断都有一个相应的使能标志来控制该中断是否能够唤醒处于STOP或WAIT模式的CPU。如果该中断唤醒功能关闭，那么它将不能唤醒处于STOP或WAIT模式

下的CPU，但相应的中断标志仍会置位。

SIP1标志未处理的中断，为只读寄存器。当SIP1的某一位为1时，表示有一个未处理的中断事件。

5.6低电压测试模块(LVD)

MCU上电或供电电压跌落到VPOR以下，POR产生复位。LVD一直使MCU处于复位状态直到供电电压升高到VLVD以上。上电后，SRS寄存器中的POR和LVD位都被置为1。设置LVDR=1可以使LVD产生复位信号。LVD复位发生后，LVD使MCU处于复位状态直到供电电压超过VLVD。

设置LVDE=1, LVDIE=1, LVDRE=0可以使LVD产生LVD中断。

5.7实时时钟中断

实时时钟时钟源可以是1K的内部参考时钟，也可以是32K内部参考时钟的32分频。可以产生中断。

5.8相关寄存器

5.8.1系统复位状态寄存器(SRS)

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	ILAD	0	LVD	0
W	Writing any value to SRS address clears COP watchdog timer.							
POR:	1	0	0	0	0	0	1	0
LVR:	0	0	0	0	0	0	1	0
Any other reset:	0	Note 1	Note 1	Note 1	Note 1	0	0	0

当调试上位机发送BDC_RESET命令给MCU，所以的复位状态标志被清零。

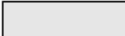
写任何值到该寄存器会清除COP计数器，但不会改变该寄存器的内容。

注：某一个复位源发生，它的相应标志置1，同时清除其他位。各位功能如下：

7 POR	上电复位 0：未发生上电复位 1：上电复位
6 PIN	外部引脚复位 0：外部引脚无复位信号 1：外部引脚复位
5 COP	看门狗复位 0：未发生看门狗复位 1：看门狗复位
4 ILOP	非法操作码复位 0：未发生操作码复位 1：非法操作码复位
3 ILAD	非法地址复位 0：未发生非法地址复位 1：非法地址复位
4 LVD	低电压检测复位 0：未发生低电压复位 1：低电压复位

5.8.2系统选择寄存器SOPT

	7	6	5	4	3	2	1	0
R	COPE	COPT	STOPE	IICPS	TPMCH1PS	TPMCH0PS	BKGDPE	RSTPE
W								
Reset:	1	1	0	0	0	0	1 (Note 1)	u
POR:	1	1	0	0	0	0	1 (Note1)	0

 = Unimplemented or Reserved u = Unaffected

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

该寄存器为复位后第一次写入有效，以后修改将无效。因此在复位初始化过程中，应将该寄存器修改为设定值，以防止以后被修改。


7 COPE	COP看门狗使能 0: 关闭看门狗功能 1: 看门狗使能
6 COPT	看门狗溢出时间选择 0: 看门狗短溢出时间 1: 看门狗长溢出时间
5 STOPE	停止模式下使能 0: 停止模式下看门狗关闭 1: 停止模式下看门狗仍运行
4 IICPS	IIC引脚选择位 0: SDA-PTA2 SCL-PTA3 1: SDA-PTB6 SCL-PTB7
3 TPMCH1PS	TPMCH1引脚选择 0: TPMCH1-PTA1 1: TPMCH1-PTB5
2 TPMCH0PS	TPMCH0引脚选择 0: TPMCH0-PTA0 1: TPMCH0-PTB4
1 BKGDPE	0: PTA4/ACMP0/BKGD/MS引脚作为PTA4或ACMP0 1: PTA4/ACMP0/BKGD/MS作为BKGD/MS
0 RSTPE	RESET(低有效) 0: PTA5/TCLK/RESET(低有效)引脚作为PTA5/TCLK/VPP 1: PTA5/TCLK/RESET(低有效)引脚作为RESET(低有效)/VPP

5.8.3 系统标识寄存器 (SDIDH, SDIDL)

该两个只读寄存器用于表示器件的ID。

5.8.4 系统实时时钟中断状态和控制寄存器

	7	6	5	4	3	2	1	0
R	RTIF	0			0			
W		RTIACK	RTICLKS	RTIE			RTIS	
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

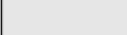
7 RTIF	实时时钟中断标志 0: 实时时钟未超时 1: 实时时钟超时，中断发生
6	实时时钟中断确认位。向该位写1将清除RTIF

RTIACK	
5 RTICKS	实时时钟时钟源选择 0: 内部参考1KHz晶振 1: 内部参考32KHz的32分频
4 RTIE	实时时钟中断使能位 0: 实时时钟中断关闭 1: 实时时钟中断使能
2: 0 RTIS	实时时钟中断超时选择, 见下表。

RTIS	RTI Timeout ¹
000	Disable RTI
001	8 ms
010	32 ms
011	64 ms
100	128 ms
101	256 ms
110	512 ms
111	1.024 s

5.8.5 系统电源状态和控制寄存器1 (SPMSC1)

	7	6	5	4	3	2	1	0
R	LVDF	0	LVDIE	LVDRE ⁽¹⁾	LVDSE	LVDE ⁽¹⁾	0	BGBE
W		LVDACK						
Reset:	0	0	0	1	1	1	0	0


 = Unimplemented or Reserved

7 LVDF	低电压检测中断事件标志, 只读
6 LVDACK	低电压检测确认位, 只写。用于确认低电压检测事件发生。写1到该位, 将清除LVDF
5 LVDIE	低电压检测中断使能位 0: LVDF不产生中断 1: 当LVDF=1时产生中断要求
4 LVDRE	低电压检测复位使能位 0: LVDF不产生硬件复位 1: 当LVDF=1时强制MCU复位
3 LVDSE	低电压检测在STOP模式下运行使能位 0: STOP模式下, LVD停止工作 1: STOP模式下, LVD仍工作
2 LVDE	LVD使能位 0: LVD关闭 1: LVD工作
0	带隙缓冲使能位: 参考带隙电压缓冲器作为ACMP的一路输入

BGNE	0: 带隙缓冲关闭 1: 带隙缓冲使能
------	------------------------

5.8.6 系统未处理中断寄存器(SIP1)

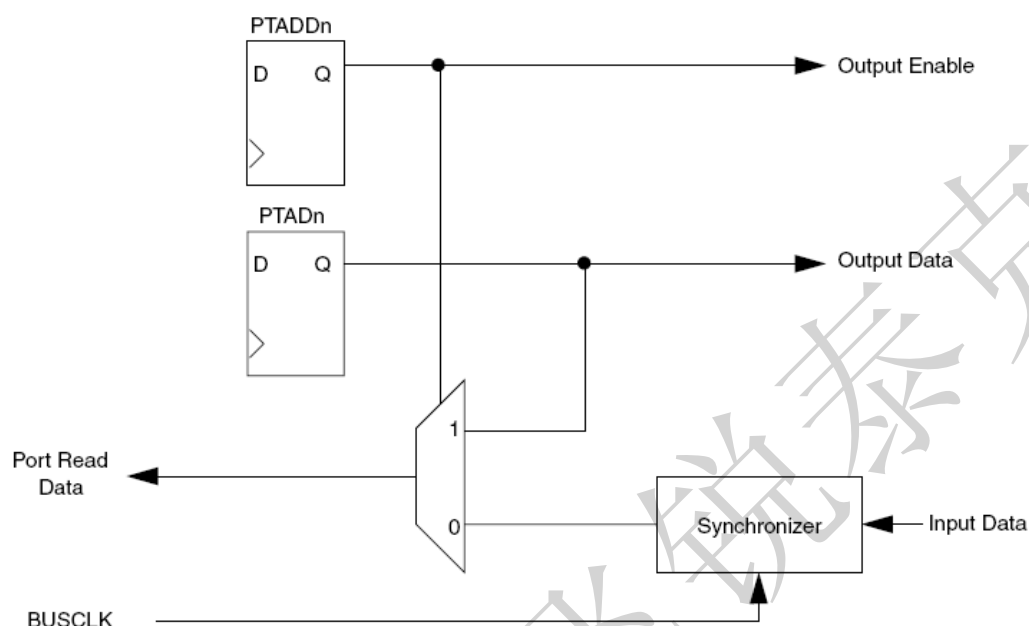
	7	6	5	4	3	2	1	0
R	IIC	KBI	ACMP	ADC	TPM	MTIM2	MTIM1	RTI
W								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

7 IIC	IIC中断未处理标志: 0:无IC中断事件 1:有未处理的IIC中断事件
6 KBI	键盘中断未处理标志: 清除KBISC寄存器中的KBF位操作也将清除该位, 复位将清除该位 0: 无键盘中断事件 1: 有未处理的键盘中断事件
5 ACMP	模拟比较器中断未处理标志: 清除ACMPSC寄存器中的ACF位操作将清除该位, 复位也会清除该位。 0: 无模拟比较器中断事件 1: 有未处理的模拟比较器中断事件
4 ADC	ADC中断未处理标志 0:无ADC中断事件 1:有未处理的ADC中断事件
3 TPM	TPM中断未处理标志 0:无TPM中断事件 1:有未处理的TPM中断事件
2 MTIM2	预定计时器中断未处理标志: 清除MTIM2SC中的TOF位操作将清除该位, 复位也会清除该位 0: 无预定计时器中断事件 1: 有未处理的预定计时器中断事件
1 MTIM1	预定计时器中断未处理标志: 清除MTIM1SC中的TOF位操作将清除该位, 复位也会清除该位 0: 无预定计时器中断事件 1: 有未处理的预定计时器中断事件
0 RTI	实时时钟中断未处理标志: 清除SRTISC中的RTIF位操作将清除该位, 复位也会清除该位 0: 无实时时钟中断事件 1: 有未处理的实时时钟中断事件

第六章 并行输入/输出控制

原理图如下：



6.1 并行输入/输出控制介绍

引脚的输入/输出功能与其他片上外围功能共用。当引脚的片上外围功能使能时，该引脚的输入/输出功能取消。复位后，引脚默认为输入/输出功能。所有 I/O 作为输入，上拉/下拉未使能(除了 PTA4，作为输出，复位后默认为 BKGD/MS 功能)。PTADDn 位决定输出缓冲是否连接到它相关联的引脚，同时也决定端口数据寄存器读取源。输入缓冲一直连接到它相关联的引脚，除非该引脚作为模拟功能或只输出。

当该引脚上的一个复用数字功能使能，输出缓冲由该数字功能控制。但数据方向寄存器仍控制端口数据寄存器读取源。

当该引脚上的一个复用模拟功能使能，输出和输入缓冲均失去作用。当引脚方向控制位为 0，该引脚作为输入并且输入缓冲失去作用，读取该引脚数据位将返回 0。通常，如果引脚数字和模拟功能都使能，模拟功能将控制该引脚。

注意：改变端口从输入状态到输出状态，应先改变端口数据寄存器的值。

在 WAIT 和 STOP 模式，所有引脚状态将维持。一旦恢复，所以引脚将按进入 WAIT 或 STOP 模式前的状态运行。

6.2 I/O 寄存器

6.2.1 端口 A 数据寄存器 PTAD

	7	6	5	4	3	2	1	0
R	0	0	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
W								
Reset:	0	0	0	0	0	0	0	0

端口 A 作输入，读取 PTAD 寄存器将得到引脚上的逻辑电平。

端口 A 作输出，读取 PTAD 寄存器将得到上一次写入该寄存器的值。

6.2.2 端口 A 方向寄存器 (PTADD)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTADD3	PTADD2	PTADD1	PTADD0
W								
Reset:	0	0	0	0	0	0	0	0

0: 作为输入, 可读取引脚上的值。

1: 作为输出。

6.2.3 内部上拉使能寄存器 (PTAPE)

	7	6	5	4	3	2	1	0
R	0	0	PTAPE5	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0
W								
Reset:	0	0	0	0	0	0	0	0

0: 关闭相应引脚的内部上拉

1: 使能相应硬件的内部上拉

6.2.4 上拉/下拉控制寄存器 (PTAPUD)

	7	6	5	4	3	2	1	0
R	0	0	PTAPUD5	0	PTAPUD3	PTAPUD2	PTAPUD1	PTAPUD0
W								
Reset:	0	0	0	0	0	0	0	0

0: 内部上拉使能

1: 内部下拉使能。

6.2.5 输出 slew rate 控制寄存器 (PTASE)

	7	6	5	4	3	2	1	0
R	0	0	0	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
W								
Reset:	0	0	0	1	1	1	1	1

0: 取消输出 slew rate 控制

1: 使能 slew rate 控制。

6.2.6 驱动强度寄存器

	7	6	5	4	3	2	1	0
R	0	0	0	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
W								
Reset:	0	0	0	0	0	0	0	0

0: 低驱动强度

1: 高驱动强度

6.3 端口 B 寄存器

6.3.1 端口 B 数据寄存器

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

6.3.2 端口 B 方向寄存器

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

0:输入

1:输出

6.3.3 端口 B 上拉使能寄存器

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

6.3.4 端口 B 上拉下拉寄存器

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

6.3.5 端口 B 压摆率控制寄存器

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

6.3.6 端口 B 驱动强度寄存器

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

6.4 端口 C 寄存器

6.4.1 端口 C 数据寄存器

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTCD3	PTCD2	PTCD1	PTCD0
W								
Reset:	0	0	0	0	0	0	0	0

6.4.2 端口 C 方向寄存器

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTCDD3	PTCDD2	PTCDD1	PTCDD0
W								
Reset:	0	0	0	0	0	0	0	0

7. 2. 2KBI引脚使能寄存器

	7	6	5	4	3	2	1	0
R	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
W								
Reset:	0	0	0	0	0	0	0	0

0: 相应的引脚不能作为键盘中断引脚

1: 相应的引脚作为键盘中断引脚

7. 2. 3KBI边沿选择寄存器

	7	6	5	4	3	2	1	0
R	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBEDG3	KBEDG2	KBEDG1	KBEDG0
W								
Reset:	0	0	0	0	0	0	0	0

0: 下降沿/低电平

1: 上升沿/高电平

7. 3功能介绍

键盘中断也就是外部中断，可以唤醒处于WAIT或STOP状态的CPU。KBI初始化过程：

1. 屏蔽键盘中断。即清KBICS中的KBIE位为0。
2. 如果使用上拉/下拉，配置好相关联的IO上拉/下拉。
3. 设置KBI极性。即设置KBIES寄存器中的KBEDGn位。
4. KBI使能。即KBIPE寄存器中的KBIPEn位。
5. 向KBISC寄存器中的KBACK位写1，清除错误键盘中断标志。
6. 使能键盘中断。

第八章 CPU

8.1 CPU介绍

RS08 适应于要求超低功耗的应用领域。主要特性如下:

- *指令集为 S08 的子集
- *新增指令 SHA 和 SLA
- *新增简短和短地址寻址模式
- *最大 16K 可寻址空间
- *删除了中断和复位向量机制
- *删除了 RAM 堆栈处理机制
- *单独硬件堆栈用于子程序调用
- *低功耗模式
- *通过内部或外部中断从停止模式唤醒
- *非法地址和操作数将产生复位
- *硬件安全机制用于保护对未授权 NVM 区域的修改
- *通过一个口线即可实现程序调试, NVM 擦除和编程

8.2 CPU寄存器

- 目的寄存器 A:8 位, 复位后为\$00
- 程序计数器PC: 14位, 复位后PC值为\$3FFD
- 影子程序计数器SPC: 14位, 当程序执行JSR或BSR指令(调用子程序)后, SPC保存下返回地址。子程序执行完成后, 执行RTS(子程序返回)指令, SPC中的值复制到PC中。
- 条件代码寄存器 CCR:2 位, 根据该 2 位状态进入分支程序。复位为 0。
Z:结果为 0 标志; C: 进位标志。
- 变址数据寄存器(D[X]):该寄存器在存储区映射地址为\$000E。
- 变址寄存器(X):该寄存器在存储区映射地址为\$000F。
- 页选寄存器(PAGESEL):用于选择存储器的某一页。

8.3 寻址模式

8.3.1 固定寻址模式(INH)

通常当操作数位于 CPU 内部的指令例如 LSLA, CLRA, INCA, SLA, RTS 等或无操作数的指令例如 NOP, BGND 情况下使用该寻址模式。

8.3.2 相对寻址模式(REL)

用于程序跳转。

8.3.3 立即寻址模式(IMM)

通常在操作数前加“#”表征操作码后面的数据为立即数而不是地址。

8.3.4 简短地址寻址模式(TNY)

寻址空间为\$0000~\$000F 使用该模式。该模式使程序更紧凑。

8.3.5 短地址寻址模式(SRT)

寻址空间为\$000~\$001F 使用该模式。该模式使程序更紧凑。

8.3.6 直接寻址模式(DIR)

寻址空间为\$000~\$00FF 使用该模式。

8.3.7 扩展寻址模式(EXT)

寻址空间为\$0000~\$3FFF 使用该模式。

8.3.8 变址(间接)寻址模式(IX)

X 寄存器中内容为地址(\$00~\$FF), D[X]寄存器中的内容为 X 寄存器中指定地址中的内容。

8.4 指令表

操作符定义:

()=表示寄存器或存储器中的内容

<-=加载

↔=交换

&=与操作

|=或操作

⊕=异或操作

:=连接

+=加

CPU 寄存器

A=累加器

CCR=条件代码寄存器

PC=程序计数器

PCH=程序计数器高 6 位

PCL=程序计数器低 8 位

SPC=影子程序计数器

SPCH=影子程序计数器高 6 位

SPCL=影子程序计数器低 8 位

存储器和地址

M=存储器地址或绝对数据, 依赖于寻址模式

Rel=相应偏移量

X=变址寄存器

, X 或 D[X]=X 指定地址中的内容

CCR 位:

Z=结果为 0 标志

C=进位或借位标志

CCR 活动标号:

-=位不受影响

0=位置 0

1=位置 1

U=不确定

机器码符号:

dd:14 位地址的低 8 位

ii:一字节立即数

hh:14 位地址的高 6 位

ll:扩展地址的低字节

rr:相对偏移量

源操作数格式:

n=表示 0~7 的标号或表达式

x=表示\$0~\$F 的标号或表达式

opr8i=表示 8 位立即数的标号或表达式

opr4a=表示地址\$0000~\$000F 的标号或表达式

opr5a=表示地址\$0000~\$001F 的标号或表达式

opr8a=表示地址\$0000~\$00FF 的标号或表达式

opr16a=表示地址\$0000~\$3FFF 的标号或表达式

rel=表示-128~+127 偏移量的标号或表达式

寻址模式

INH=固定寻址模式

IMD=立即数与直接地址寻址模式

IMM=立即数寻址模式

DD=直接地址与直接地址寻址模式

DIR=直接地址寻址模式

SRT=短地址寻址模式

TNY=简短地址寻址模式

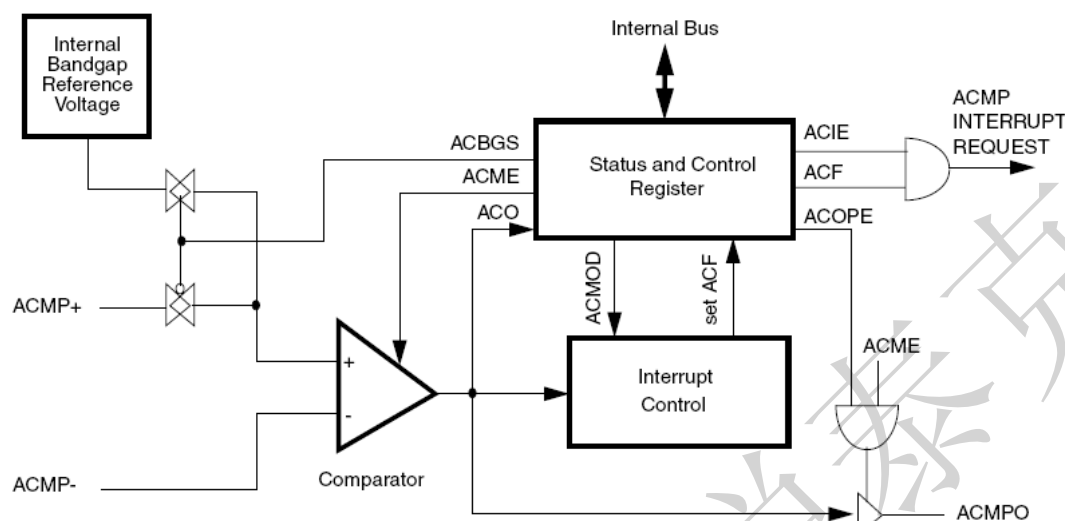
EXT=扩展寻址模式

REL=8 位相对寻址模式

指令表参考英文文档 Table8-1

第九章 模拟比较器(ACMP)

原理图如下:



模拟比较器模块(ACMP)用于比较两个模拟输入或一个模拟输入与内部参考电压的比较。ACMP 为全电压比较器。特点:

- 全电压操作
- 小于 40 毫伏的输入偏置
- 小于 15 毫伏的电压迟滞
- 比较器中断输出: 上升沿, 下降沿或任意
- 可选的内部能隙参考电压
- 比较器输出可以在 ACMP0 引脚呈现
- 在停止模式仍可以工作

ACMP 有两个模拟输入, 一个数字输出。每个模拟输入变化范围为 MCU 的供电电压。

9.2 操作模式

9.2.1 在 WAIT 模式下的操作

ACMP 中断可以唤醒 MCU。为了功耗更下, 在进入 WAIT 模式前应关闭 ACMP 模块。

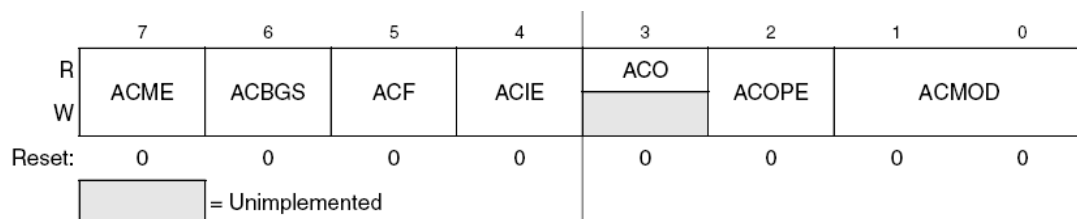
9.2.2 在 STOP 模式下的操作

在 STOP 模式下, ACMP 仍可以运行, 故当比较事件发生时唤醒 STOP 模式下的 MCU。

9.2.3 背景调试模式

当 MCU 处于背景调试模式, ACMP 将继续运行。

9.3ACMP 寄存器



7 ACME	ACMP 模块使能 0: ACMP 模块关闭 1: ACMP 模块使能
6 ACBGS	0: 外部施加到引脚上的信号作为比较器的正输入 1: 内部能隙参考电压作为比较器的正输入
5 ACF	0: 比较事件未发生 1: 比较事件发生
4 ACIE	模拟比较器中断使能 0: 关闭比较器中断 1: 比较器中断使能
3 ACO	模拟比较器输出值
2 ACOPE	0: ACMP 输出不会驱动 ACMPO 引脚 1: ACMP 输出驱动 ACMPO 引脚
1:0 ACMOD	00: 输出下降沿 01: 输出上升沿 10: 输出下降沿 11: 输出上升或下降沿

10.2.4 特点

- 10 位线性逐次逼近算法
- 多达 28 路模拟输入
- 输出格式 10，或 8 位
- 单次或连续转换
- 可配置抽样时间和转换速率
- 转换完成标志和中断
- 输入时钟可选，多达 4 个时钟源
- 可工作于 STOP 和 wait 模式
- 异步时钟源可用于低噪声操作
- 可选的异步硬件转换触发器
- 自动比较中断（当小于，大于或等于预定值）

10.2.5 ADC 框图

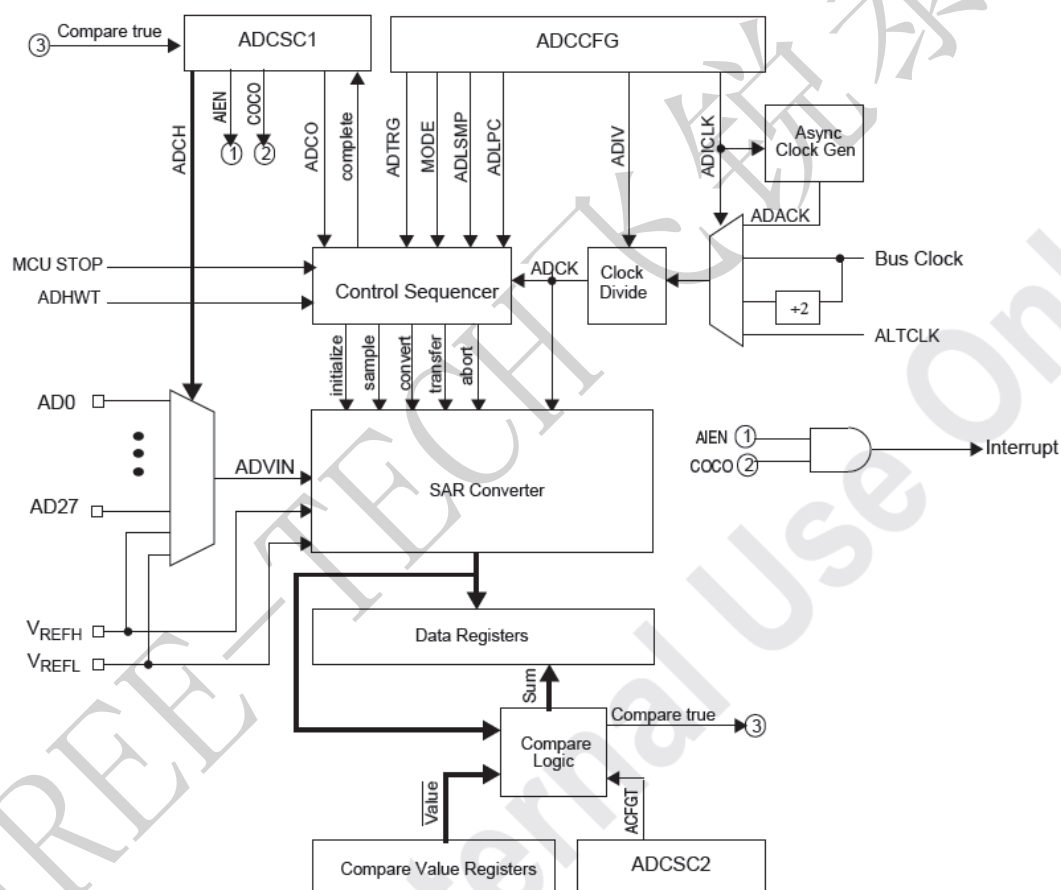


图 10-2 ADC 原理框图

10.3 外部信号描述

Name	Function
AD27–AD0	Analog Channel inputs
V _{REFH}	High reference voltage
V _{REFL}	Low reference voltage
V _{DDAD}	Analog power supply
V _{SSAD}	Analog ground

表 10.1 ADC 外部信号表

10.3.1 模拟电压 (V_{DDAD})

一般 V_{DDAD} 连接到 V_{DD}，加滤波电容。

10.3.2 模拟地 (V_{SSAD})

V_{SSAD} 可连接到 V_S

10.3.3 参考电压高 (V_{REFH})

一般连接到 V_{DDAD}

10.3.4 参考电压低 (V_{REFL})

一般连接到 V_{SSAD}

10.3.5 模拟输入通道 (AD_x)


ADC 模块支持多达 28 个独立的模拟输入。

10.4 寄存器定义

- 状态与控制寄存器，ADC1SC1
- 状态与控制寄存器，ADC1SC2
- 数据结果寄存器，ADC1RH，ADC1RL
- 比较值寄存器，ADC1CVH，ADC1CVL
- 配置寄存器，ADC1CFG
- 引脚使能寄存器，APCTL1,APCTL2,APCTL3

10.4.1 状态和控制寄存器 (ADC1SC1)

	7	6	5	4	3	2	1	0
R	COCO							
W		AIEN	ADCO					
Reset:	0	0	0	1	1	1	1	1

 = Unimplemented or Reserved

7 COCO	转换完成标志一只读，比较功能禁止时 (ACFE=0)，当每次转换完成后，该位置位。当比较功能允许时 (ACFE=1)，只有比较结果为真时，
-----------	---

	COCO 置位。清除该位方法：任意时刻写 ADC1SC1 或读 ADC1RL 0：转换未完成；1：转换完成
6 AIEN	中断允许位 0：禁止转换完成中断；1：允许转换完成中断
5 ADCO	连续转换允许位 0：通过向写 ADC1SC1 启动一次转换（选择的为软件触发）或通过 ADHWT 触发启动一次转换（选择的为硬件触发）； 1：通过向写 ADC1SC1 启动连续转换（选择的为软件触发）或通过 ADHWT 触发启动连续转换（选择的为硬件触发）
4: 0 ADCH	通道选择位一见下表。 当通道选择位均为 1 时，ADC 的转换系统关闭。利用这一特点可直接禁止 ADC 功能。当中止连续转换时将阻止再次转换。 当 ADC 未使能连续转换（即单次转换）那么没有必要将通道选择位都置 1 使 ADC 进入低功耗模式，因为 ADC 模块在完成转换后会自动进入低功耗模式

ADCH	Input Select
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7

ADCH	Input Select
01000	AD8
01001	AD9
01010	AD10
01011	AD11
01100	AD12
01101	AD13
01110	AD14
01111	AD15

ADCH	Input Select
10000	AD16
10001	AD17
10010	AD18
10011	AD19
10100	AD20
10101	AD21
10110	AD22
10111	AD23

ADCH	Input Select
11000	AD24
11001	AD25
11010	AD26
11011	AD27
11100	Reserved
11101	V _{REFH}
11110	V _{REFL}
11111	Module disabled

10.4.2 状态和控制寄存器 2（ADC1SC2）

	7	6	5	4	3	2	1	0
R	ADACT	ADTRG	ACFE	ACFGT	0	0	R ¹	R ¹
W								
Reset:	0	0	0	0	0	0	0	0


= Unimplemented or Reserved

7 ADACT	转换活动标志：当开始转换时该位置位，当转换完成或中止时该位清零 0：转换未进行中；1：转换进行中
------------	---

6 ADTRG	转换触发器选择：软件触发（写 ADC1SC1）或硬件触发 0：软件触发；1：硬件触发
5 ACFE	比较功能允许位 0：禁止比较功能；1：允许比较功能
4 ACFGT	比较功能触发方式选择位 0：当输入小于比较值时比较器触发；1：当输入大于或等于比较值时触发

10.4.3 数据转换结果高（ADCRH）


	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	ADR9	ADR8
W								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

ADC1RH 包含 10 位转换结果的最高两位。当配置为 8 位转换模式时，ADR8 和 ADR9 等于 0。ADC1RH 在每次转换完成后均更新，除非比较功能使能而且比较条件未满足。在 10 位模式下，读取 ADC1RH 将阻止下一次的转换结果转移到结果寄存器中，直到 ADC1RL 被读取。如果下一次转换已经完成，而 ADC1RL 未读取那么这次转换结果丢失。在 8 位模式下，ADC1RL 没有互锁机制。当 MODE 位发生改变，那么在 ADC1RH 中的任何数据将无效。

10.4.4 数据结果寄存器低（ADCRL）


	7	6	5	4	3	2	1	0
R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
W								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

该寄存器功能可参照 ADC1RH。

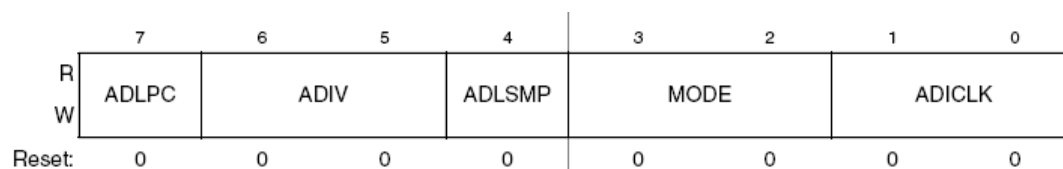
10.4.5 比较值高位寄存器（ADCCVH）

	7	6	5	4	3	2	1	0
R	0	0	0	0			ADCV9	ADCV8
W								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

10.4.6 比较值低位寄存器（ADCCVL）

	7	6	5	4	3	2	1	0
R	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
W								
Reset:	0	0	0	0	0	0	0	0

10.4.7 配置寄存器 (ADCCFG)

7 ADLPC	低功耗配置-控制转换器转换速度和功耗 0: 高速转换; 1: 低速转换
6: 5 ADIV	时钟分频系数-用于选择分频系数从而产生 ADCK, 见下表
4 ADLSMP	抽样间隔配置: 通过调整抽样间隔以允许高阻抗输入可以被精确抽样或低阻抗输入时加快转换速度。当选择的为连续转换模式并且对转换速度要求不高的前提下, 较长的抽样间隔可以降低整体功耗。 0: 短抽样间隔; 1: 长抽样间隔
3: 2 MODE	转换模式选择: 见下表
1: 0 ADICLK	输入时钟选择: 见下表

ADIV	Divide Ratio	Clock Rate
00	1	Input clock
01	2	Input clock ÷ 2
10	4	Input clock ÷ 4
11	8	Input clock ÷ 8

表 10.2 时钟分频选择表

ADICLK	Selected Clock Source
00	Bus clock
01	Bus clock divided by 2
10	Alternate clock (ALTCLK)
11	Asynchronous clock (ADACK)

表 10.3 时钟源选择位

MODE	Mode Description
00	8-bit conversion (N=8)
01	Reserved
10	10-bit conversion (N=10)
11	Reserved

表 10.4 模式选择表

10.4.8 引脚控制寄存器 1(APCTL1)

	7	6	5	4	3	2	1	0
R	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
W								
Reset:	0	0	0	0	0	0	0	0

7	ADPC7	ADC 引脚控制位 7 0: AD7 引脚作为 IO 1: AD7 引脚作为 ADC 的输入
6	ADPC6	ADC 引脚控制位 6 0: AD6 引脚作为 IO 1: AD6 引脚作为 ADC 的输入
5	ADPC5	ADC 引脚控制位 5 0: AD5 引脚作为 IO 1: AD5 引脚作为 ADC 的输入
4	ADPC4	ADC 引脚控制位 4 0: AD4 引脚作为 IO 1: AD4 引脚作为 ADC 的输入
3	ADPC3	ADC 引脚控制位 3 0: AD3 引脚作为 IO 1: AD3 引脚作为 ADC 的输入
2	ADPC2	ADC 引脚控制位 2 0: AD2 引脚作为 IO 1: AD2 引脚作为 ADC 的输入
1	ADPC1	ADC 引脚控制位 1 0: AD1 引脚作为 IO 1: AD1 引脚作为 ADC 的输入
0	ADPC0	ADC 引脚控制位 0 0: AD0 引脚作为 IO 1: AD0 引脚作为 ADC 的输入

10.4.9 引脚控制寄存器 2 (APCTL2)

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

7	ADPC15	ADC 引脚控制位 15 0: AD15 引脚作为 IO 1: AD15 引脚作为 ADC 的输入
6	ADPC14	ADC 引脚控制位 14 0: AD14 引脚作为 IO 1: AD14 引脚作为 ADC 的输入
5	ADPC13	ADC 引脚控制位 13 0: AD13 引脚作为 IO 1: AD13 引脚作为 ADC 的输入
4	ADPC12	ADC 引脚控制位 12 0: AD12 引脚作为 IO 1: AD12 引脚作为 ADC 的输入
3		ADC 引脚控制位 11

ADPC11	0: AD11 引脚作为 IO 1: AD11 引脚作为 ADC 的输入
2 ADPC10	ADC 引脚控制位 10 0: AD10 引脚作为 IO 1: AD10 引脚作为 ADC 的输入
1 ADPC9	ADC 引脚控制位 9 0: AD9 引脚作为 IO 1: AD9 引脚作为 ADC 的输入
0 ADPC8	ADC 引脚控制位 8 0: AD8 引脚作为 IO 1: AD8 引脚作为 ADC 的输入

10.4.10 引脚控制寄存器 3 (APCTL3)

	7	6	5	4	3	2	1	0
R	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
W								
Reset:	0	0	0	0	0	0	0	0

7 ADPC23	ADC 引脚控制位 23 0: AD23 引脚作为 IO 1: AD23 引脚作为 ADC 的输入
6 ADPC22	ADC 引脚控制位 22 0: AD22 引脚作为 IO 1: AD22 引脚作为 ADC 的输入
5 ADPC21	ADC 引脚控制位 21 0: AD21 引脚作为 IO 1: AD21 引脚作为 ADC 的输入
4 ADPC20	ADC 引脚控制位 20 0: AD20 引脚作为 IO 1: AD20 引脚作为 ADC 的输入
3 ADPC19	ADC 引脚控制位 19 0: AD19 引脚作为 IO 1: AD19 引脚作为 ADC 的输入
2 ADPC18	ADC 引脚控制位 18 0: AD18 引脚作为 IO 1: AD18 引脚作为 ADC 的输入
1 ADPC17	ADC 引脚控制位 17 0: AD17 引脚作为 IO 1: AD17 引脚作为 ADC 的输入
0 ADPC16	ADC 引脚控制位 16 0: AD16 引脚作为 IO 1: AD16 引脚作为 ADC 的输入

10.5 功能描述

在复位或 ADCH 各位都为 1 时, ADC 模块关闭。当本次转换完成, 而另一个转换未启动时, 模块处于空闲状态。模块处于空闲状态时为最低功耗。

ADC 模块可以根据软件选择的通道执行模数转换。所选择的通道通过逐次逼近算法将结果转换为 11 位数据。在 8 位模式，会将结果转换为 9 位数据结果。

当转换完成，结果存放在数据寄存器（ADCRH 和 ADCRL）。10 位模式下，10 位结果放置在 ADCRH 和 ADCRL。8 位模式下，结果存放在 ADCRL。转换完成标志 COCO 在转换完成后置位，可以产生中断。

ADC 模块还具有自动将转换结果与比较寄存器值进行比较功能。

10.5.1 时钟选择和分频控制

可以选择的时钟源有四种，时钟源经过分频产生转换时钟—ADCK。时钟源选择由 ADICLK 位控制。

- 1) 总线时钟，也就是软件执行的时钟。复位之后的默认时钟源
- 2) 总线时钟 2 分频
- 3) ALTCLK
- 4) 异步时钟 ADACK。该时钟由 ADC 模块内部产生。选择该时钟源，那么当 MCU 处于等待或 STOP 模式时，ADC 仍工作。

无论选择哪种时钟源，但应符合 ADCK 范围。如果时钟太低，ADC 也不会工作。时钟太快，那么时钟必须分频到合适的值。分频值可以为 1, 2, 4, 8

10.5.2 输入选择和引脚控制

寄存器 APCTL3, APCTL2, APCTL1 控制 ADC 输入通道。当该引脚作为模拟输入引脚时，该引脚的状态为：

- 输出缓冲被强制为其高阻状态。
- 输入缓冲禁止。
- 上拉功能禁止。

10.5.3 硬件触发转换

ADC 模块可采用异步触发—ADHWT，当 ADTRG=1 时。ADC 模块在 ADHWT 的上升沿开始转换。转换过程中，再次出现的上升沿将被忽略。在连续转换模式下，只有第一次启动转换的上升沿被检测。

10.5.4 转换控制

MODE 位决定转换结果为 10 位或 8 位模式。可以软件或硬件启动转换。而且 ADC 模块可以配置为低功耗，长抽样间隔时间，连续转换，自动比较转换结果的功能。

10.5.4.1 启动转换

- 在软件触发操作模式下，写 ADCSC1 寄存器将启动转换。
- 在硬件触发模式下，ADHWT 触发事件
- 在连续转换模式下，当转换结果转移到数据寄存器中。

在连续转换模式中，在完成当前转换完成后，自动启动下一次转换。在软件启动转换模式下，在写 ADCSC1 之后连续转换开始，直到被中止。在硬件触发模式下，硬件触发事件发生后开始连续转换，直到被中止。

10.5.4.2 转换完成

当转换结果转移到数据结果寄存器后，转换完成。此时 COCO 置位，用

于表征转换完成。如果 $AIEN=1$ ，则产生中断。

在读取 $ADCRH$ 和 $ADCRL$ 寄存器过程中，使用互锁阻止机制以避免新的转换结果覆盖先前的数据。当阻止机制发生，那么会阻止转换结果转移到结果寄存器中， $COCO$ 不会被置位，新的转换结果丢失。

在单次转换，比较功能使能，但比较条件未满足时，阻止机制不产生作用， ADC 转换操作被中止。在其他操作模式下，当数据传输互锁机制发生，下一次数据转换仍会启动。

在单次转换模式下，必须等到转换完成后，再去读取数据寄存器中的值。

10.5.4.3 转换中止

当发生下面的操作将中止转换：

- 1) MCU 复位
- 2) MCU 进入 STOP 模式并且 $ADACK=0$
- 3) 写 $ADCSC1$ 寄存器(当前转换被中止,启动一个新的转换,前提 $ADCH$ 不全为 1)
- 4) 写 $ADCSC2, ADCCFG, ADCCVH, ADCCVL$ 。表明操作模式发生变化,当前转换因此无效。

当转换被中止， $ADC1RH$ 和 $ADC1RL$ 中的寄存器的内容不会改变，而是保持为最近上一次成功转换的结果。如果是由于复位中止的转换，那么 $ADC1RH$ 和 $ADC1RL$ 为复位默认值。

10.5.4.4 功耗控制

ADC 模块一般处于空闲状态，直到开始转换。如果 $ADACK$ 被选择为转换时钟， $ADACK$ 时钟发生器使能。

设置 $ADLPC$ 位，可降低 ADC 功耗。

10.5.4.5 总转换时间

总的转换时间依赖于抽样时间（ $ADLSMP$ 位决定），MCU 总线频率，转换模式（8 位或 10 位），和转换时钟（ f_{ADCK} ）。

当模块进入活动状态，开始抽样模拟输入。 $ADLSMP$ 用于选择短的抽样时间和长的抽样时间。当抽样完成，转换器将与输入通道隔离开，执行逐次逼近算法。该算法执行完成后，转换结果转移到 $ADC1RH$ 和 $ADC1RL$ 。

如果总线频率小于 f_{ADCK} 频率时，在连续转换模式下，采用短抽样的话，不能保证精确抽样。当总线频率小于 f_{ADCK} 的 $1/11$ ，在连续转换模式下，采用长抽样时间，也不能保证精确抽样。

Conversion Type	ADICLK	ADLSMP	Max Total Conversion Time
Single or first continuous 8-bit	0x, 10	0	20 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	0x, 10	0	23 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	0x, 10	1	40 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	0x, 10	1	43 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	11	0	5 μ s + 20 ADCK + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	11	0	5 μ s + 23 ADCK + 5 bus clock cycles
Single or first continuous 8-bit	11	1	5 μ s + 40 ADCK + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	11	1	5 μ s + 43 ADCK + 5 bus clock cycles
Subsequent continuous 8-bit; $f_{\text{BUS}} \geq f_{\text{ADCK}}$	xx	0	17 ADCK cycles
Subsequent continuous 10-bit or 12-bit; $f_{\text{BUS}} \geq f_{\text{ADCK}}$	xx	0	20 ADCK cycles
Subsequent continuous 8-bit; $f_{\text{BUS}} \geq f_{\text{ADCK}}/11$	xx	1	37 ADCK cycles
Subsequent continuous 10-bit or 12-bit; $f_{\text{BUS}} \geq f_{\text{ADCK}}/11$	xx	1	40 ADCK cycles

表 10.5 总转换时间和控制条件

最大的转换时间由所选时钟源和分频系数决定。例如，10 位模式，总线频率作为输入时钟源，分频系数为 1，总线频率 8MHz。单次转换时间为：

$$\text{Conversion time} = \frac{23 \text{ ADCK cyc}}{8 \text{ MHz}/1} + \frac{5 \text{ bus cyc}}{8 \text{ MHz}} = 3.5 \mu\text{s}$$

$$\text{Number of bus cycles} = 3.5 \mu\text{s} \times 8 \text{ MHz} = 28 \text{ cycles}$$

10.5.5 自动比较功能

比较功能可用于检查上限或下限。输入被抽样后，然后进行转换，转换结果与 ADCCVH 和 ADCCVL 中的值比较。当 ACFG T=1，转换结果大于或等于比较值，COCO 置位。当 ACFG T=0，转换结果小于比较值，COCO 置位。

一旦转换完成，此时比较功能使能，如果比较条件未满足，那么 COCO 不会置位，没有数据转移到结果寄存器中。

注：比较功能可以用于处于 WAIT 或 STOP 模式下 MCU 监测通道电压。当比较条件满足，ADC 中断可以唤醒 MCU。

10.5.6 WAIT 模式下的操作

MCU 执行 WAIT 指令，MCU 可以进入低功耗模式，在这种模式下时钟源仍运行，故可以快速恢复。当 MCU 进入 WAIT 模式后，转换正进行中，那么它将继续转换直到完成。在 WAIT 模式下，可以通过硬件触发方式启动转换或仍继续连续转换模式。

总线时钟，总线时钟的二分频，ADACK 在 WAIT 模式下仍可作为转换时钟源。如果选择 ALTCLK 作为转换时钟的话，应查看每种 MCU 的 ALTCLK 定义。

转换完成后，COCO 位设置为 1，产生 ADC 中断，可以唤醒 WAIT 模式下的 MCU。

10.5.7 MCU 在 STOP 模式下

10.5.7.1 STOP 模式下 ADACK 关闭

如果 ADACK 未被选择为转换时钟，那么执行 STOP 指令将中断当前转换，ADC 进入空闲状态。ADC1RH 和 ADC1RL 的内容不会被改变。从 STOP 模式唤醒后，软件或硬件触发器用于恢复转换。

10.5.7.2 STOP 模式下，ADACK 运行

如果选择 ADACK 作为转换时钟，那么在 STOP 模式中，ADC 仍继续运行。为确保 ADC 的运行，MCU 稳压器仍处于活动状态。

进入 STOP 模式后，正在运行的转换将继续转换，直到完成。通过硬件触发或连续转换将启动 ADC 转换。

转换完成后，COCO 位置位，如果 AIEN=1，将产生中断，MCU 从 STOP 模式唤醒。

10.6 初始化信息

10.6.1 ADC 模块初始化例子

10.6.1.1 初始化过程

典型初始化过程如下：

1. 更新配置寄存器 ADCCFG 选择时钟源和分频系数产生内部时钟，ADCK。同时该寄存器用于选择抽样时间和低功耗配置。
2. 更新 ADCSC2 选择触发方式和比较器是否使能
3. 更新 ADCSC1 选择单次转换还是连续转换，打开或关闭转换完成中断以及输入通道。

10.6.1.2 实例

该例子中，中断功能使能，单次 10 位转换，低功耗，长抽样时间，输入通道 1，ADCK 时钟由总线时钟驱动，分频系数为 1。

ADCCFG=0X98

BIT7 ADLPC 1 配置为低功耗模式

BIT6:5 ADIV 00 设置 ADCK 为输入时钟的 1 分频

BIT4 ADLSMP 1 配置长抽样时间

BIT3:2 MODE 10 设置为 10 位转换

BIT1:0 ADICLK 00 选择总线时钟作为输入时钟源

ADCSC2=0X00

BIT7 ADACT 0 标识转换进行中

BIT6 ADTRG 0 软件触发方式

BIT5 ACFE 0 关闭比较功能

BIT4 ACFG 0 未使用

BIT3:2 00

BIT1:0 00

ADCSC1=0X41

BIT7 COCO 0

BIT6 AIEN 1 中断使能

BIT5 ADCO 0 单次转换

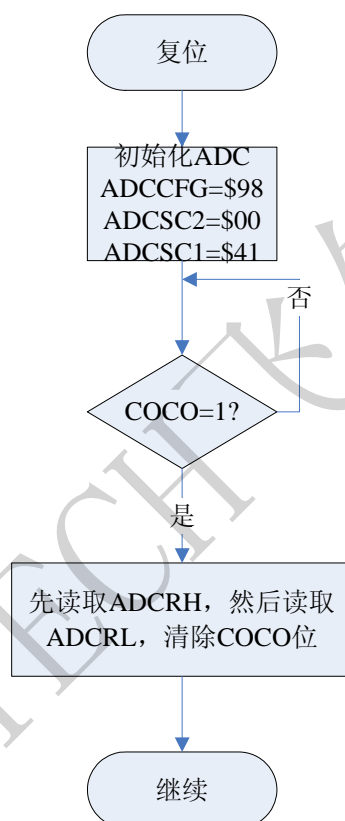
BIT4:0 ADCH 00001 通道 1 作为模拟输入

ADCRH/L=0xxx

ADCCVH/L=0xxx

APCTL1=0X02

APCTL2=0X00



10.7 应用信息

10.7.1 外部引脚

10.7.1.1 模拟电源引脚

大多数情况下, 模拟电源和数字电源独立, 但模拟电源和数字电源应通过一点共地。

10.7.1.2 模拟参考引脚

该两个引脚应加 0.1u 的电容, 并且靠近引脚。

10.7.1.3 模拟输入引脚

外部模拟输入与 I/O 引脚复用。建议引脚作为模拟输入时, 该引脚的控

制寄存器一直置位。尽量靠近输入引脚与 V_{SSA} 之间一个高频 0.01uF 电容，可以提高 ADC 性能。

为了能够正确转换，输入电压必须在 V_{REFH} 和 V_{REFL} 之间。如果输入信号等于或大于 V_{REFH} ，那么数据寄存器将为全 1。如果输入信号等于或小于 V_{REFL} ，数据寄存器将全为 0。

10.7.2 误差源及尽量减小误差的方法

10.7.2.1 抽样错误

为了得到正确的转换结果，输入模拟信号必须被正确的抽样。假如输入最大阻值接近 7k，输入电容接近 5.5pF，如果外部模拟信号源的阻抗在 5k 以下，那么最小的抽样窗口可以满足精度要求。

如果信号源阻抗较大或要求精度高，那么可以采用较长的抽样时间。

10.7.2.2 引脚漏电误差

如果外部模拟信号源内阻很大，那么 I/O 引脚的漏电会产生转换误差。

如果该误差不能被应用所接受，那么应使 R_{AS} 小于 $V_{DDAD} / (2^N * I_{LEAK})$ ，N 等于 8 或 10。

10.7.2.3 噪声引入的误差

在抽样或转换过程中，系统噪声会影响转换的精度。

- V_{REFH} 和 V_{REFL} 之间放置一个 0.1uF 的低等效串连电阻电容
- V_{DDAD} 和 V_{SSAD} 之间放置一个 0.1uF 的低等效串连电阻电容
- 如果 V_{DDAD} 通过电感与主电源隔离，那么 V_{DDAD} 和 V_{SSAD} 之间放置一个 1uF 电容
- 如果 V_{SSAD} 和 V_{REFL} 连接到 V_{SS} 那么应在一个地平面的一个免受干扰的点。
- 在转换期间，没有 I/O 进行开关操作。

10.7.2.4 位宽和量化误差

8 位或 10 位 ADC 有 $\pm 1/2LSB$ 的误差。

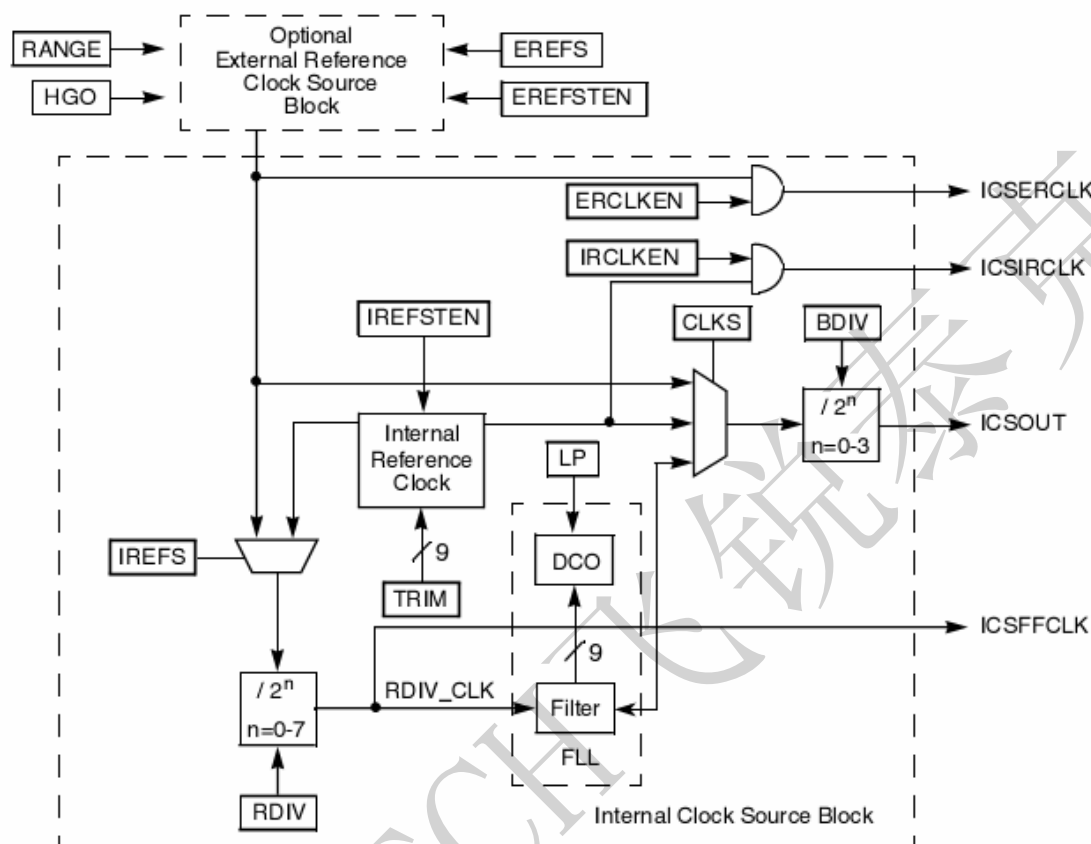
$$1LSB = (V_{REFH} - V_{REFL}) / 2^N$$

10.7.2.5 线性误差

线性误差有几种形式：零点误差；满刻度误差；微分非线性误差；积分非线性误差；总未调误差。

第十一章 内部时钟源(ICS)

原理图如下:



11.1 介绍

内部时钟源（ICS）为MCU提供了几种时钟源选择。模块包括一个内部参考时钟控制的锁频环。因此ICS可以为MCU系统时钟（ICSOUT）提供FLL或内部参考时钟两种选择。无论选择哪种时钟源，ICSOUT为总线频率的2倍。特点：

—FLL可调

使用内部32K赫兹参考源可达0.2%分辨率

使用内部32K赫兹参考源过压过温仅2%偏差

DCO输出是内部参考时钟的512倍

—内部参考时钟有9位调节位可用

—内部参考时钟可选为MCU时钟源

—复位后FLL自动被选为MCU系统时钟

—2位控制对MCU系统时钟分频系数

11.2 工作模式

ICS有7种操作模式：FEI, FEE, FBI, FBILP, FBE, FBELP, STOP

FEI: 内部参考时钟, FLL使能

FEE: 外部参考时钟, FLL使能

FBI: 内部参考时钟, FLL使能, ICS由内部参考时钟驱动

FBILP: 内部参考时钟为MCU系统时钟源, FLL关闭。

FBE: 外部参考时钟, FLL使能, ICS由外部参考时钟驱动

FBELP: 外部参考时钟, FLL关闭, ICS由外部参考时钟驱动

STOP:FLL停止。内部参考时钟可选择为运行或停止。ICS不为MCU提供时钟源。

11. 3ICS寄存器

11. 3. 1控制寄存器1ICSC1

	7	6	5	4	3	2	1	0
R	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
W								
Reset:	0	0	0	0	0	1	0	0
7: 6 CLKS	时钟模式选择：用于选择控制总线时钟源 00:FLL 的输出 01:内部参考时钟源 10:外部参考时钟源 11:预留 CLKS为可以在任何时候被修改。							
5:3 RDIV	对 FLL 参考源分频系数选择 000:不分频 001:2 分频 010:4 分频 011:8 分频 100:16 分频 101:32 分频 110:64 分频 111:128 分频							
IREFS 2	内部参考源选择:该位用于选择 FLL 的时钟参考源 0:选择外部参考时钟源 1:选择内部参考时钟源							
IRCLKEN 1	内部参考时钟使能控制位:该位控制内部参考时钟是否作为 ICSIRCLK 0:ICSIRCLK 关闭 1:ICSIRCLK 使能							
IREFSTEN 0	STOP 模式下内部参考是否关闭 0:STOP 模式下内部参考时钟关闭 1:STOP 模式下内部参考时钟禁止							

11. 3. 2控制寄存器2ICSC2

	7	6	5	4	3	2	1	0
R	BDIV		RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTEN
W								
Reset:	0	1	0	0	0	0	0	0
7:6 BDIV	对 CLKS 控制位所选择的时钟源分频 00:不分频 01:2 分频 10:4 分频 11:8 分频							
5 RANGE	频率范围选择:选择外部晶振范围 0:低频范围 1:高频范围							

4 HGO	外接晶振高增益振荡器选择位 0: 振荡器配置为低功耗模式 1: 振荡器配置为高增益模式
3 LP	FLL低功耗模式控制位 0:FLL旁路模式下,FLL不关闭 1:FLL旁路模式下,FLL关闭
2 EREFS	外部参考时钟源选择 0: 外部时钟源 1: 外接晶振或振荡器组成的振荡器电路
1 ERCLKEN	0:ICSECLK 关闭 1:ICSECLK 使能
0 EREFTEN	0:STOP 模式下外部参考时钟关闭 1:STOP 模式下外部参考时钟禁止

11. 3. 3ICS调整寄存器TRIM

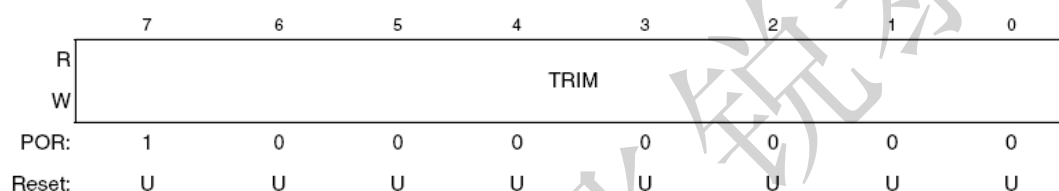
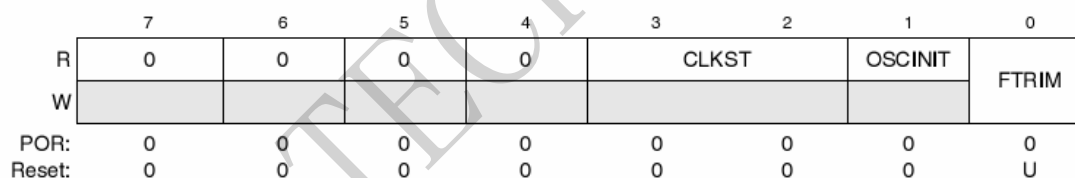


Figure 9-5. ICS Trim Register (ICSTRM)

TRIM: 用来调整内部参考时钟频率。该8位为二进制格式。TRIM值增加, 内部参考时钟频率降低; 反之, 内部参考时钟频率提高。

11. 3. 4ICS状态和控制寄存器ICSSC



CLKST: 表明当前选择的时钟源

0: FLL输出选择为MCU系统时钟

1: 内部参考时钟选择为MCU系统时钟

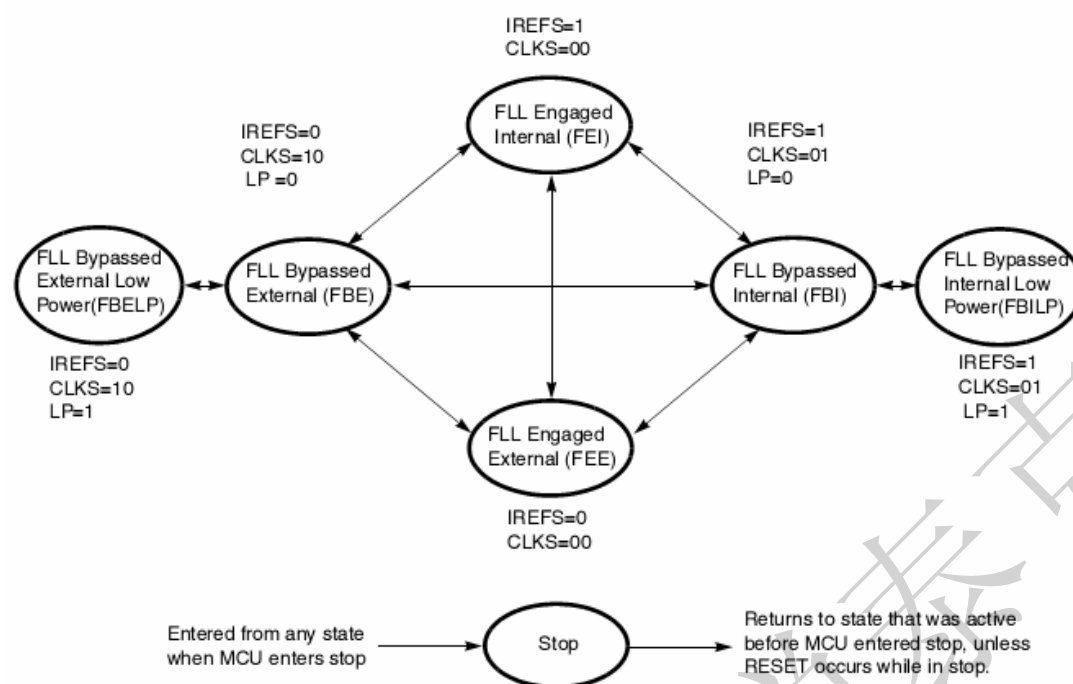
OSCINIT:

OSC初始化状态标志位

FTRIM: 细调内部参考时钟。

该位置1将降低内部参考时钟源频率, 该位清零将提高内部参考时钟频率。这种调整方式每次调整值会尽可能小。

11. 4ICS工作模式之间的切换



11.4.1.1FEI

该模式为复位后的默认方式. ICS 处于该模式的条件为:

CLKS=00

IREFS=1

RDIV 将 FLL 的参考频率分频到 31.25kHz~39.0625kHz.

此模式下 FLL 将参考频率倍频 512 倍.

11.4.1.2FEE

CLKS=00

IREFS=0

RDIV 将 FLL 的参考频率分频到 31.25kHz~39.0625kHz.

该模式 FLL 的参考频率为外部时钟.

11.4.1.3FBI

CLKS=01

IREFS=1

LP=0

该模式下 ICSOUT 由内部参考时钟驱动. FLL 由内部参考源时钟提供参考时钟. FLL 电路工作.

11.4.1.4FBILP

CLKS=01

IREFS=1

LP=1

该模式下 ICSOUT 由内部参考时钟驱动, FLL 电路关闭.

11.4.1.5FBE

CLKS=10

IREFS=0

LP=0

该模式下 ICSOUT 由外部参考时钟驱动, FLL 的参考时钟为外部.

11.4.1.6FBELP

CLKS=10

IREFS=0

LP=1

11.4.1. 7STOP

11.4.2 模式切换

当在 FEI 和 FEE 模式之家切换时, IREFS 位可随时改变, 但 RDIV 位必须同时也要改变.

11.4.3 总线频率分频器

BDIV 位可在任意时刻改变.

11.4.4 低功耗位使用

LP 位可控制 FLL 是否使能.

11.4.5 内部参考时钟(ICSIRCLK)

内部参考时钟是通过改变 TRIM 寄存器或 FTRIM 位的值调整其频率, 调整结果将影响 ICSOUT 的输出. TRIM 和 FTRIM 的值不会受复位的影响. 为使 ICS 在 STOP 模式仍运行, 在进入 STOP 模式前应置 SPMSC1 寄存器中的 LVDE 和 LVDSE 位为 1.

复位后分频系数默认为 2 其目的是为防止调整内部参考时钟后 ICSOUT 超过 MCU 系统时钟的频率要求.

11.4.6 固定时钟频率(ICSFFCLK)

当 ICS 工作于 FEI 模式, ICSFFCLK 为 MCU 的外围模块(例如计数器)提高额外的时钟源, 但在 FBI 或 FBILP 模式下, ICSFFCLK 是对外围模块而言是无效的.

第十二章 IIC

12.1 介绍

IIC 接口为与多个设备进行通信提供了一种方法。通信速率高达 100kbps。

12.1.1 特征

- 与 IIC 标准总线兼容
- 多主机操作
- 软件可编程选择为 64 个不同串口时钟频率
- 软件可选的确认位
- 数据传输可产生中断
- 仲裁失败中断（具有自动由主机转为从机模式）
- 寻址识别中断
- 开始和停止信号产生与检测
- 重新启动信号产生
- 确认位产生与检测
- 总线忙检测
- 全局呼叫识别
- 10 位地址扩展

12.1.2 运行模式

- 正常运行模式—基本的运行模式。
- 等待模式—模块继续运行
- 停止模式—在 STOP 模式下，IIC 不再活动，但 STOP 指令不会影响 IIC 寄存器状态。

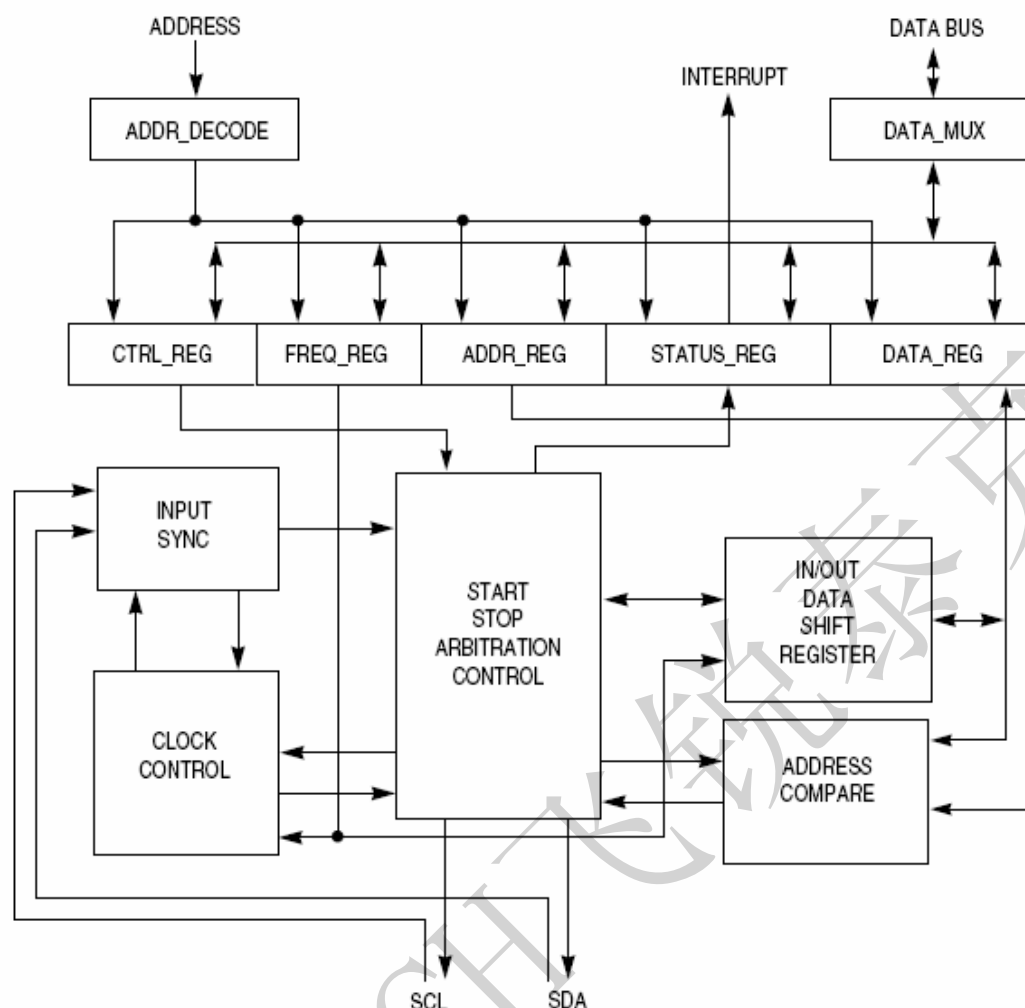


图 12.1 IIC 原理框图

12.2 外部信号描述

12.2.1 SCL—串行时钟线

SCL 为双向的串行时钟线

12.2.2 SDA—串行数据线

SDA 为双向的串行数据线

12.3 寄存器定义

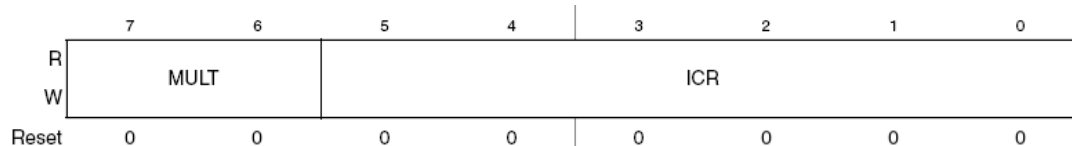
12.3.1 IIC 地址寄存器 (IICA)

	7	6	5	4	3	2	1	0
R	AD7	AD6	AD5	AD4	AD3	AD2	AD1	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

7: 1 ADDR[7:1]	从机地址—该 7 位为 IIC 模块的从机地址
-------------------	-------------------------

12.3.2 IIC 频率分频寄存器 (IICF)




7: 6 MULT	<p>IIC 倍频因子—用于产生波特率</p> <p>00 mul=01</p> <p>01 mul=02</p> <p>10 mul=04</p> <p>11 未用</p>
5: 0 ICR	<p>IIC 时钟波特率</p> <p>IIC 波特率=总线频率/ (mul * SCL divider)</p> <p>SDA 维持时间定义为从 SCL (IIC 时钟) 的下降沿到 SDA 发生改变的延时时间。ICR 用于决定 SDA 的维持时间值。</p> <p>IIC 维持时间=总线周期*SDA 维持时间值。</p> <p>例如:</p> <p>总线频率=8MHZ</p> <p>MULT=01(mul=2)</p> <p>要是 IIC 波特率=100kbps</p> <p>IIC 波特率=总线频率/(mul * SCL 分频系数)</p> <p>100000=8000000/(2 * SCL 分频系数)</p> <p>SCL 分频系数=40</p> <p>此时, 假如 ICR=0X0B</p> <p>SDA 维持时间=总线周期 * SDA 维持时间值</p> <p>SDA 维持时间=1/8000000 * 9 = 1.125us。</p>

ICR (hex)	SCL Divider	SDA Hold Value	SCL Hold (Start) Value	SDA Hold (Stop) Value	ICR (hex)	SCL Divider	SDA Hold Value	SCL Hold (Start) Value	SDA Hold (Stop) Value
00	20	7	6	11	20	160	17	78	81
01	22	7	7	12	21	192	17	94	97
02	24	8	8	13	22	224	33	110	113
03	26	8	9	14	23	256	33	126	129
04	28	9	10	15	24	288	49	142	145
05	30	9	11	16	25	320	49	158	161
06	34	10	13	18	26	384	65	190	193
07	40	10	16	21	27	480	65	238	241
08	28	7	10	15	28	320	33	158	161
09	32	7	12	17	29	384	33	190	193
0A	36	9	14	19	2A	448	65	222	225
0B	40	9	16	21	2B	512	65	254	257
0C	44	11	18	23	2C	576	97	286	289
0D	48	11	20	25	2D	640	97	318	321
0E	56	13	24	29	2E	768	129	382	385
0F	68	13	30	35	2F	960	129	478	481
10	48	9	18	25	30	640	65	318	321
11	56	9	22	29	31	768	65	382	385
12	64	13	26	33	32	896	129	446	449
13	72	13	30	37	33	1024	129	510	513
14	80	17	34	41	34	1152	193	574	577
15	88	17	38	45	35	1280	193	638	641
16	104	21	46	53	36	1536	257	766	769
17	128	21	58	65	37	1920	257	958	961
18	80	9	38	41	38	1280	129	638	641
19	96	9	46	49	39	1536	129	766	769
1A	112	17	54	57	3A	1792	257	894	897
1B	128	17	62	65	3B	2048	257	1022	1025
1C	144	25	70	73	3C	2304	385	1150	1153
1D	160	25	78	81	3D	2560	385	1278	1281
1E	192	33	94	97	3E	3072	513	1534	1537
1F	240	33	118	121	3F	3840	513	1918	1921

表 12.1 IIC 分频与维持值

12.3.3 IIC 控制寄存器 (IICC1)

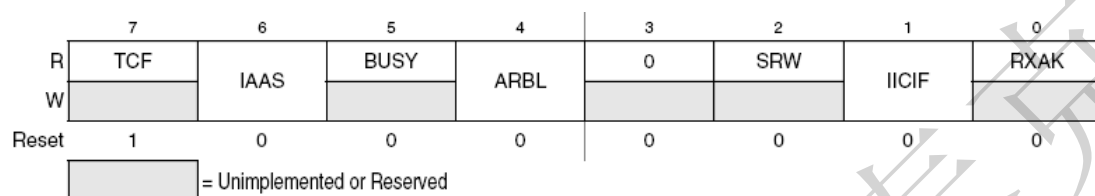
	7	6	5	4	3	2	1	0
R	IICEN	IICIE	MST	TX	TXAK	0	0	0
W						RSTA		
Reset	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

7 IICEN	IIC 模块允许控制位 0: IIC 模块关闭; 1: IIC 模块使能
6 IICIE	IIC 中断允许控制位 0: 禁止 IIC 中断; 1: 允许 IIC 中断
5 MST	主设备模式选择—MST 位由 0 变为 1, 总线上产生一个开始信号, 且作为主设备; 该位由 1 变为 0, 在总线上产生一个停止信号, 且由主设备变为从设备。 0: 从机模式 1: 主机模式
4 TX	发送模式选择—TX 位选择主和从传送方向。在主模式下, 根据传送要求该位必须被置位。因此在寻址期间, 该位应始终为高。当作为从设备时, 软件应根据 SRW 位对 TX 位置位。

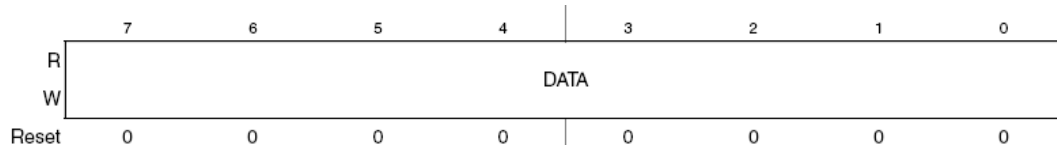
	0: 接收; 1: 发送
3 TXAK	发送确认使能位:该位控制在数据确认期间 SDA 线上的值。 0: 接收完一字节数据后, 确认信号发送到总线 1: 不发送确认信号
2 RSTA	重启动—假设它是当前主设备, 向该位写 1 将产生一个重复开始信号。读该位始终为低。而试图在错误时间发送重复开始信号将产生仲裁失败错误。

12.3.4 IIC 状态寄存器 (IICS)



7 TCF	传输完成标志—当完成一字节传送后该位置位。注意该位只在数据传送期间或紧随其后有效。在接收模式时, 读取 IIC1D 寄存器或在发送模式时, 写 IIC1D 寄存器可清除该位。 0: 正在传送; 1: 传送完成
6 IAAS	作为从设备被寻址—当寻址地址与从设备的可编程地址匹配时, 该位置位。或者 GCAEN=1, 接收到全局寻址。写 IICC 操作将清除该位。 0: 未被寻址到; 1: 作为从设备被寻址到
5 BUSY	总线忙标志—无论主设备模式还是从设备模式, 该位表明总线的状态。当起始信号被检测到, BUSY 位置位, 检测到 STOP 信号该位清除。 0: 总线空闲; 1: 总线忙
4 ARBL	仲裁失败标志—清除该位方法: 向该位写 1。 0: 标准总线操作; 1: 仲裁失败
2 SRW	从设备读/写标志位:当 IIC 模块作为从设备,该位表明 R/W 位的值 0: 从设备接收; 主设备写从设备; 1: 从设备发送; 主设备读取从设备数据
1 IICIF	IIC 中断标志 0: 无中断事件; 1: 有中断; 该位必须通过软件清除; 方法为向该位写 1。以下事件将导致 IICIF 位置位。 — 一字节传送完毕 — 地址匹配 — 仲裁丢失
0 RXAK	接收确认状态位—当 RXAK 位为低, 表明在总线传送完一个字节后收到一个确认信号。 0: 接收到一个确认信号; 1: 未收到确认信号

12.3.5 IIC 数据寄存器 (IICD)



7: 0 DATA	数据一在主设备发送模式，当数据写入到 IICD，数据传送开始；最高位先被发送。在主设备接收模式，读取该寄存器开始接收下一个字节数据
--------------	---

当主机由发送模式转为接收模式前，不对 IICD 进行读取，避免错误的启动一次主机接收数据的传送。

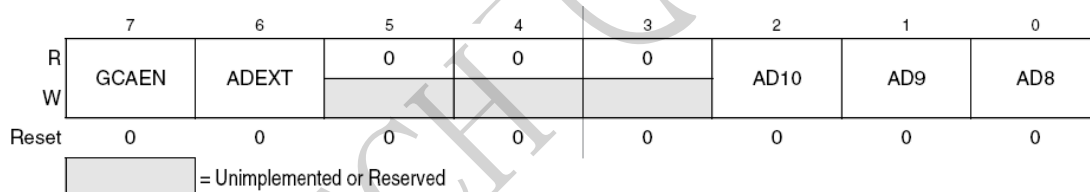
在从机模式，当地址匹配后，IICD 作为数据发送和读取源。

注意在 IICC 寄存器中的 TX 位必须正确反映主设备模式和从设备模式下传送方向。例如 IIC 配置为主设备发送，但想进行主设备接收过程，那么读取 IICD 并不会启动接收。

当 IIC 配置为主设备接收或从设备接收模式，读取 IICD 将返回上一次接收到的字节。IICD 不会反映总线上发送的每一字节，也不会校验已经写入到 IICD 数据的准确性。

在主机发送模式，主机应先将 MST 由 0 变为 1，然后发送 7 位从机地址，R/W 位为 0。

12.3.6 IIC 控制寄存器 2 (IICC2)



7 GCAEN	General call 寻址使能控制位 0: 禁止 1: 允许
6 ADEXT	地址扩展控制位 0: 7 位地址 1: 10 位地址
2:0 AD[10:8]	从机设备的高 3 位。

12.4 功能描述

12.4.1 IIC 协议

IIC 总线使用串行数据线 SDA 和串行时钟线 SCL 进行数据传输。所有设备连接到总线上，漏极开路或集电极开路。使用外部电阻上拉，线与功能。

正常情况下，标准通信由 4 部分组成：

主机发送启动信号

主机发送从设备地址

主机发送数据

停止信号

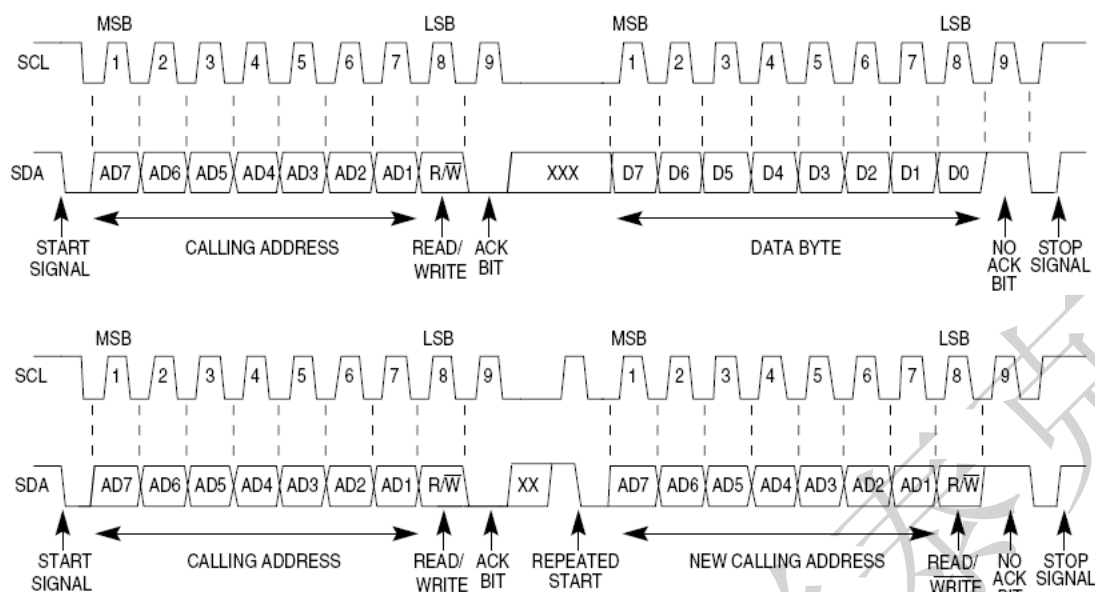


图 12.2 IIC 传送协议

12.4.1.1 启动信号

当总线空闲时，SCL 和 SDA 线都为高电平。主设备通过发送 START 信号启动通信。启动信号定义为：SCL 为高，SDA 由高变为低。该信号通知一个新数据传输开始，唤醒总线上所有其他空闲状态的设备。

12.4.1.2 从设备地址发送

起始信号后的第一个字节是从设备地址。其中高 7 位为地址，第 8 位为 R/W 位。R/W 位告诉从设备数据传送方向。

1=主机读：从设备发送数据到主设备

0=主机写：主设备发送数据到从设备

只有和该地址匹配的从设备作出响应，从设备会发一个确认信号。即从设备在第 9 个时钟将 SDA 拉低。

在系统中，从设备地址是唯一的。如果 IIC 模块为主设备，那么它不能发送一个等于它作为从设备时的地址。IIC 模块不能同时作为主设备和从设备。然而，如果在一个寻址期间内，仲裁失败，那么 IIC 模块将转变为从机模式。那么其他主机如果寻址正确的话，就可以进行通信。

12.4.1.3 数据传输

每个数据长度位 8 位。数据只能在 SCL 为低期间改变，而在高电平期间必须维持。每一位数据对应一个时钟脉冲。先发送最高位。每字节数据后有一个确认位，该位由接收方发出。即在第 9 个时钟周期内，将 SDA 线拉低。所以一个完整字节的传输需要 9 个时钟脉冲。

如果在主设备的第 9 个脉冲时间，接收器没有确认。那么 SDA 线必须由从设备维持高电平。主设备会由于未得到确认而产生中断。

如果在主设备接收时，当接收完一字节数据后，主机未发确认位，那么

从设备认为传送数据完成，释放 SDA 线。

中断数据传输方法：主设备进行一下任意一种操作：

- 通过产生一个停止信号放弃总线。
- 通过产生一个重复的起始信号开始一个新的寻址。

12.4.1.4 停止信号

主设备通过发送停止信号来释放总线。然而主设备可以产生一个开始信号，然后发寻址命令，这种操作称为重复开始。

停止信号的定义：SCL 为 1，SDA 由低变高。

在数据传输过程中，从设备发出确认信号后，主机可以产生停止信号，该停止信号告诉从机释放总线。

12.4.1.5 重新开始信号

重复开始信号定义：一个新的开始信号（前一个开始信号后未发停止信号）。通常用于在不释放总线的情况下主设备与另一个从设备通信或者同一个从设备（不同模式下）。

12.4.1.6 仲裁过程

IIC 是真正的多主总线，因此允许多个主设备连接在总线上。如果两个以上的主设备试图同时控制总线时，一个时钟同步过程决定总线时钟，低电平持续时间等于时钟信号中的最长的低电平时间，而高电平时间等于最短的高电平时间。

多主机竞争的相对优先级由数据仲裁过程确定，在仲裁过程中，如果一个主机发送 1，而其他主机发送 0，那么该主机仲裁失败。仲裁失败的主机立刻转变为从机接收模式，停止驱动 SDA。在这种情况下，主设备转变为从设备不会产生停止信号。同时，它的状态位被硬件置位表明其仲裁失败。

12.4.4.7 时钟同步

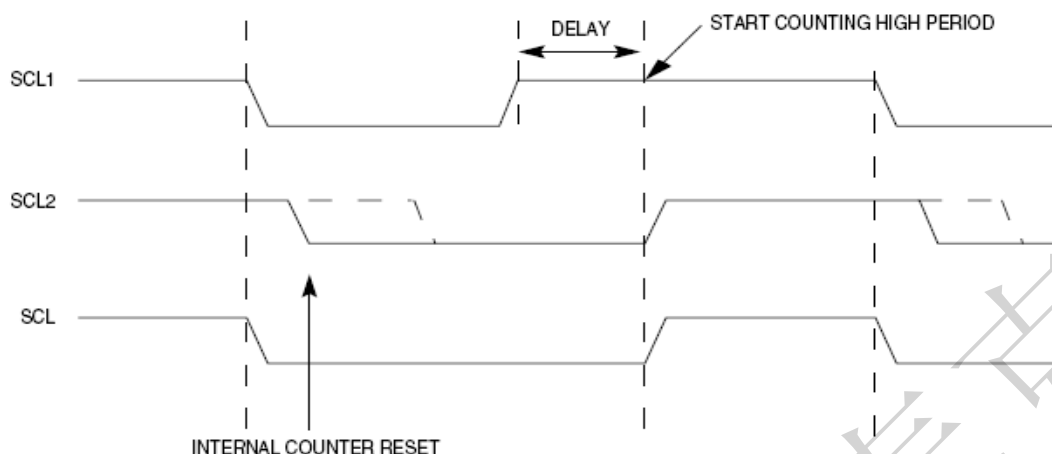


图 12.3 时钟同步图

因为 SCL 线具有线与逻辑功能，SCL 线上一个由高到低转变会影响到总线上所有设备。当某个设备开始计数它们的低电平间隔，此时该设备将使 SCL 维持低电平，直到遇到时钟高电平。然而，该设备时钟由低变高时，其他设备时钟仍处于低电平状态。那么它不能将 SCL 驱动到高电平。因此，同步时钟 SCL 会被低电平最长的设备控制。

时间较短的低电平设备在该期间进入高电平等待状态。当所有的设备都结束它们的低电平时间，那么同步时钟 SCL 线释放，被拉高。同理，第一个完成它高电平的设备将 SCL 线先拉低。

12.4.1.8 握手

时钟同步机制可以用于数据传输的握手。完成一字节传送之后，从设备可能维持 SCL 为低，在这种情况下，它将挂起总线时钟，迫使主机时钟进入等待模式直到从设备释放 SCL 线。

12.4.1.9 时钟扩展

从设备利用时钟同步机制可用于降低传送波特率。在主设备将 SCL 驱动低后，从设备可以驱动 SCL 低并持续一个必需的时间然后释放。如果从设备驱动 SCL 为低的时间大于主设备的，结果 SCL 总线信号的低电平时间加长。

12.4.2 10 位寻址地址

12.4.2.1 主机（发送）寻址从机（接收）

S	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 0	A1	Slave Address 2nd byte AD[8:1]	A2	Data	A	...	Data	A/A	P
---	--	----------	----	-----------------------------------	----	------	---	-----	------	-----	---

如上图所示，可能不止一个从设备向主机发送确认信号 A1。而向主机发送确认信号 A2 的从设备是唯一的。

12.4.2.2 主机（接收）寻址从机（发送）

S	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 0	A1	Slave Address 2nd byte AD[8:1]	A2	Sr	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 1	A3	Data	A	...	Data	A	P
---	---	----------	----	--------------------------------------	----	----	---	----------	----	------	---	-----	------	---	---

上图所示，A2 之前的过程同主机寻址从机一样。在主机发送完 Sr 信号之后，已经被寻址到的从机会核实 Sr 信号之后的 11110+AD10+AD9 和 R/W 位，如果该从机比较 11110+AD10+AD9 与自己的一致，并且 R/W 位为 1。那么从机就处于主动发送状态，一直到它接收到 STOP 信号或 Sr 信号。

12.4.3 General Call（全局）寻址

全局寻址可以为 7 位或 10 位地址。如果 GCAEN=1，那么当 general call 寻址地址匹配 MCU 的从机地址时，IIC 模块会做出响应，并且作为从机（接收），寻址周期之后 IAAS=1。当寻址地址的第一个字节传送后，软件必须读取 IICD 寄存器，并确定地址匹配是自己的从机地址或 general call 地址。如果 IICD 为“00”，那么匹配的为 general call 地址。如果 GCAEN 为被清除，那么 IIC 模块会忽略 general call 寻址，也会忽略任何总线上的数据。

12.5 复位

复位之后，IIC 关闭。IIC 不能产生 MCU 复位。

12.6 中断

Interrupt Source	Status	Flag	Local Enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration Lost	ARBL	IICIF	IICIE

表 12.2 IIC 中断表

12.6.1 字节传送中断

在第 9 个时钟的下降沿，TCF 位置位，表明该字节传送完毕。

12.6.2 地址检测中断

当寻址地址与 IIC 地址配合或 GCAEN=1 并且接收到 general call 地址时，IAAS 位置位。如果 IICIE 使能，那么 CPU 被中断。此时 CPU 必须检查 SRW 位，设置它相应的 Tx 模式。

12.6.3 仲裁失败中断

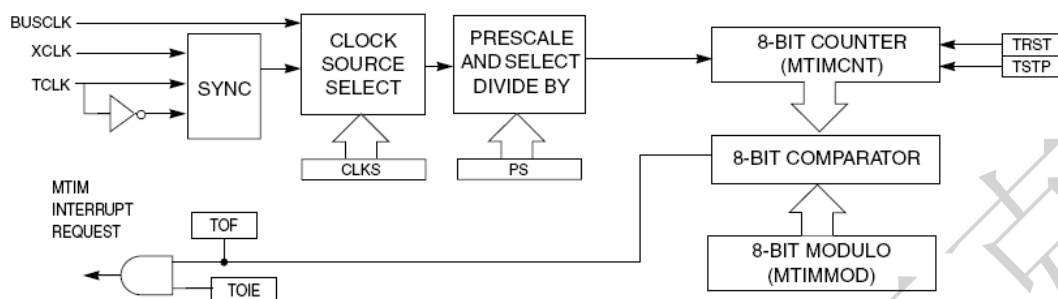
IIC 为真正的多主总线。如果两个或更多主机试图同时控制总线，那么通过一个数据仲裁的过程来解决竞争。下列情况下会发生仲裁失败：

- 在地址或数据发送周期期间，当主设备驱动 SDA 为高，SDA 被抽样为低。
- 在数据接收周期的确认位期间，当主设备驱动 SDA 为高，SDA 被抽样为低。
- 当总线忙时，试图发起信号
- 在从设备模式下，发一个重复开始信号

FREE-TECH 飞锐泰克

第十三章 模计数器(MTIM)

原理图如下:



13.1 介绍

MTIM(modulo timer)是8位计数器,可工作于自由计数或模计数。计数溢出可产生中断。特点如下:

- 自由计数或模计数
- 溢出中断可软件控制
- 计数复位位
- 计数停止位
- 四种软件选择时钟源

系统总线时钟: 上升沿

固定频率时钟 XCLK: 上升沿

外部时钟源 TCLK 引脚: 上升沿

外部时钟 TCLK 引脚: 下降沿

- 9种可选分频系数

1, 2, 4, 8, 16, 32, 64, 128, 256

13.2 工作模式

13.2.1 WAIT 模式下的操作

MTIM的溢出中断可唤醒处于WAIT模式的CPU。为WAIT模式下更小功耗,软件应在进入WAIT模式前关闭MTIM。

13.2.2 STOP 模式下的操作

MTIM在STOP模式停止工作。

如果是复位使CPU脱离STOP模式,MTIM将恢复到复位状态。

如果是中断使CPU脱离STOP模式,MTIM将继续按进入STOP模式前的状态工作。

13.2.3 在背景调试模式下的操作

CPU进入背景调试模式MTIM将挂起;CPU离开背景调试模式,MTIM从挂起状态恢复。

13.3 寄存器

13.3.1 MTIM 状态和控制寄存器 (MTIMSC)

	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

7 TOF	MTIM 溢出标志：只读 当 MTIM 计数寄存器值达到 MTIM 模寄存器的值时，溢出变为\$00，并置 TOP=1。 读取 MTIMSC 寄存器，然后写 0 到 TOF，可以清除 TOF。 TOF 也可以通过向 TRST 写 1 或向 MTIMMOD 写任何值来清除。 0：MTIM 计数未达到 MTIM 模寄存器中的值 1：MTIM 计数达到 MTIM 模寄存器中的值
6 TOIE	溢出中断使能位：可读写 0：不产生溢出中断 1：溢出中断使能
5 TRST	只写位 0：无影响 1：MTIM 计数值变为\$00
4 TSTP	0：允许 MTIM 计数 1：停止 MTIM 计数

13.3.2 MTIM 时钟配置寄存器

	7	6	5	4	3	2	1	0
R	0	0	CLKS		PS			
W								
Reset:	0	0	0	0	0	0	0	0

5:4 CLKS	00:总线时钟 01:固定频率时钟 10:外部时钟，下降沿 11:外部时钟，上升沿
3:0 PS	0000: MTIM/1 0001: MTIM/2 0010: MTIM/4 0011: MTIM/8 0100: MTIM/16 0101: MTIM/32 0110: MTIM/64 0111: MTIM/128

13.3.4 MTIM 计数寄存器

只读，复位为\$00

13.3.5 MTIM 模计数寄存器

可读写，复位为\$00

13.4 功能介绍

MTIM 计数有 3 种操作模式：停止，自由运行，和与模寄存器比较模式。CPU 复位后，MTIM 计数是停止的，计数寄存器值为 0，模寄存器值为 0，总线时钟为时钟源，分频系数为 1，只要清除 MTIMSC 中的 TSTP 位，定时器就开始运行。当没有新值写入模寄存器，计数器将工作于自由运行模式。当有非\$00 值写入模寄存器，计数器将工作于比较计数模式。

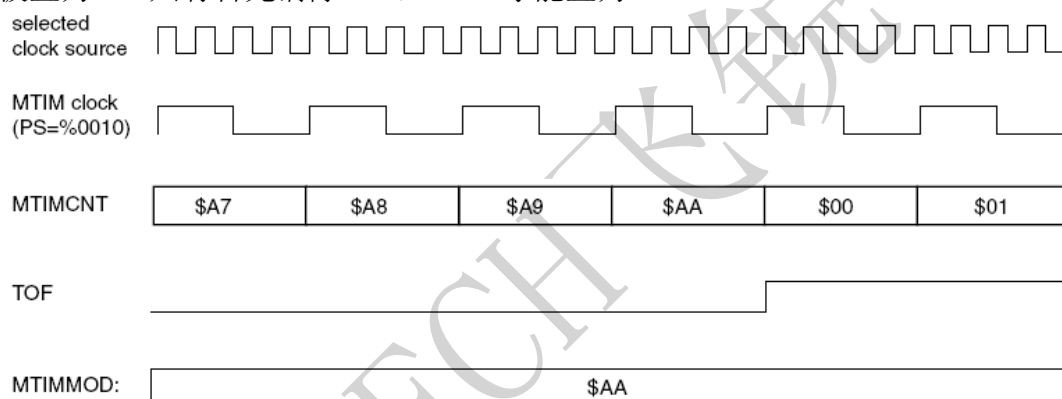
MTIM 处于运行模式下，改变 MTIM 的时钟源或改变分频系数，MTIM 将继续计数。

一旦 TSTP=0，计数器以选择的速率递增。当计数寄存器的值与模寄存器中的值相等时，计数寄存器溢出变为\$00，然后继续计数，TOF 被置 1。在计数器运行状态下，改写 MTIMMOD 将复位计数寄存器中的置，并且清除 TOF。

清除 TOF 需要两步：第一步读取 MTIMSC，第二步写 TOF 为 0。如果在上面两步之间产生另外一个溢出，那么执行完第二步并不会清除 TOF，从而避免遗漏溢出事件。

向 TRST 写 1 或向 MTIMMOD 写入任何值也可以清除 TOF。

为了能产生溢出中断，应置 MTIMSC 中的 TOIE=1。当 TOF=1 时，TOIE 不会被置为 1。只有首先清除 TOF，TOIE 才能置为 1。



第十四章 计数器/脉宽调制

14.1 介绍

TPM 模块的通道 TPMCH0 和 TPMCH1 可通过软件重新分配.

TPMCH0PS in SOPT	Port Pin for TPMCH0
0 (default)	PTA0
1	PTB4

TPMCH1PS in SOPT	Port Pin for TPMCH1
0 (default)	PTA1
1	PTB5

14.1.1 特点

- 每一个 TPM 模块的所有通道可以配置为缓冲中心对齐的 PWM
- 每一个 TPM 模块可以有独立的时钟源
- 计时时钟源可为总线时钟，固定系统时钟或外部输入时钟
- 16-位计数可工作于自由计数或递增/递减计数
- 计时系统使能
- 每一个 TPM 模块每一个通道有一个中断外加一个计数溢出中断
- 通道特点

— 每一个通道可以为输入捕捉，输出比较，或带缓冲的边沿对齐 PWM

— 上升沿，下降沿，或任意边缘触发输入捕捉

— 设置，清除或触发比较输出功能

— 可选择 PWM 输出极性

14.1.2 框图

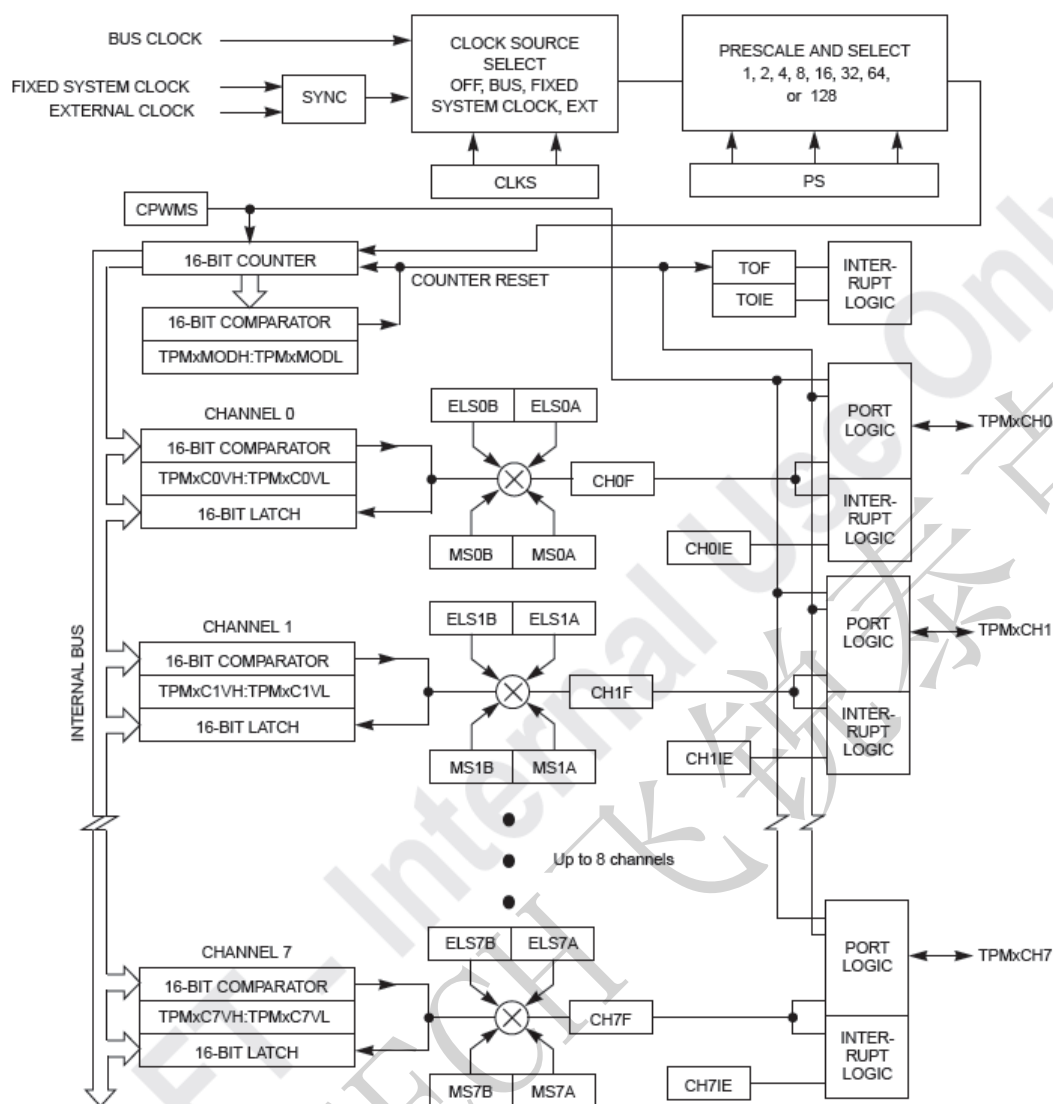


图 14.1 TPM 原理框图

TPM 核心是 16 位计数器，它可自由计数，预置计数值模式，或作为中心对齐 PWM 时的递增/递减计数。当 TPM 计数值为输入捕捉，比较输出和边沿对齐 PWM 提供一个时间参考。预置数寄存器 TPMxMODH:TPMxMODL 用来控制计数器的比较值。当=0x0000 或=0xFFFF 时，计数器为自由计数。软件任何时候均可以读取计数器的值，不会影响其计数时序。对 TPMxCNT 计数器任意字节写入任意值的操作将复位计数器。

14.2 外部信号描述

当与计数器相关的引脚配置为计数输入时，引脚内部上拉使能。复位之后，TPM 模块禁止；所有的引脚默认为通用输入，内部上拉禁止。

14.2.1 外部 TPM 时钟源

TPM 模块中的状态和控制寄存器允许用户程序选择无时钟源（关闭计时器），总线时钟（默认模式），与晶振相关的时钟或外部时钟。外部时钟源是在 **TPM** 模块被同步，而总线时钟为同步者。

当 CLKSB:CLKSA=1:1 时, TPMx 时钟由外部时钟源 TPMxCLK 驱动。在外部时钟和 TPM 之间有一个同步器。该同步器时钟由总线时钟驱动,

因此外部时钟源要小于总线频率的 1/2。外部时钟源的最大频率极限为总线频率的 1/4。

14.2.2 TMPxCHn—TMPx 通道 n 的输入/输出引脚

MCU 每一个通道都有响应的输入/输出引脚，引脚的具体作用由通道配置决定。当计数器控制引脚时，端口数据和数据方向寄存器不再对相应的引脚有控制作用。

14.3 寄存器定义

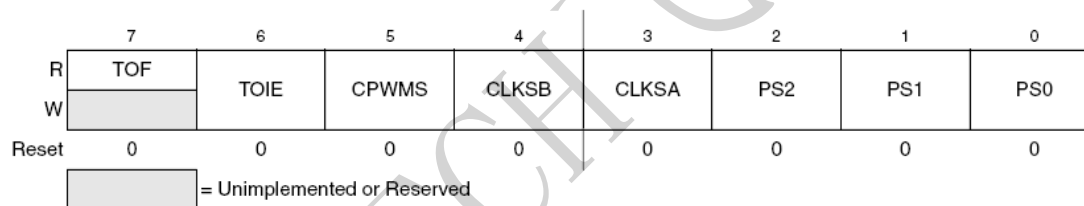
TPM 寄存器：

- 一个 8 位状态和控制寄存器 (TPMxSC)
- 一个 16 位计数器 (TPMxCNTH:TPMxCNTL)
- 一个 16 位模寄存器 (TPMxMODH:TPMxMODL)

每一个计数器通道包括：

- 一个 8 位状态和控制寄存器 (TPMxCnSC)
- 一个 16 位通道计数寄存器 (TPMxCnVH:TPMxCnVL)

14.3.1 计时器 n 状态和控制寄存器 (TPMxSC)



7 TOF	定时器溢出标志—当定时器的计数达到\$FFFF（自由计数模式）或预置计数器的值时，该位置 1，定时器的计数清零。 当 TPM 配置为 CPWM 时，当达到预置计数器的值 x 时，该位置 1，定时器的计数值变为 (x-1)，开始递减计数。 清除 TOF 方法：读 TPM 的状态和控制寄存器，然后向 TOF 位写 0。如果在清除该标志位之前又发生了下一次溢出，那么这一次清零将无法完成。程序必须保证每一次溢出发生后，都要在下次溢出发生前清除该溢出标志位。在使用溢出中断的情况下，要求在中断返回之前必须清除溢出中断标志，并且该中断的处理时间在最不利的情形下也不能超过一次计数溢出时间。 0：未发生溢出；1：溢出。
6 TOIE	定时器溢出中断允许位：可读写 0：不允许溢出中断；1：运行溢出中断
5 CPWMS	中心对齐 PWM 0：由 MSnB：MSnA 位决定所有的 TPMx 通道工作于输入捕捉，比较输出，或边沿对齐 PWM 模式 1：所有的 TPMx 通道工作于中心对齐 PWM 模式
4: 3 CLKS[B:A]	时钟源选择：见下表

2: 0 PS[2:0]	时钟源分频因子选择控制位：见下表
-----------------	------------------

CLKSB:CLKSA	TPM Clock Source to Prescaler Input
0:0	No clock selected (TPMx disabled)
0:1	Bus rate clock (BUSCLK)
1:0	Fixed system clock (XCLK)
1:1	External source (TPMxCLK) ^{1,2}

表 14.1 TPM 时钟源选择

注：1 外部时钟信号最大频率为总线频率的 1/4。

2 当外部时钟引脚与通道复用时，ELSnB:ELSnA 应设置为 0: 0。

PS2:PS1:PS0	TPM Clock Source Divided-By
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

表 14.2 时钟源分频列表

14.3.2 定时器 x 计数寄存器 (TPMxCNTH:TPMxCNTL)

只读；当读取任意一个字节时，会将这两个寄存器的内容闩锁到一个缓冲器中，它们处于闩锁状态，直到另一个字节被读取；从而保证了数据一致性。

	7	6	5	4	3	2	1	0
R	Bit 15	14	13	12	11	10	9	Bit 8
W	Any write to TPMxCNTH clears the 16-bit counter.							
Reset	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	Any write to TPMxCNTL clears the 16-bit counter.							
Reset	0	0	0	0	0	0	0	0

当后台调试器被激活，定时器的计数功能和一致性机制冻结。即使读取计数器的一个或两个寄存器值，缓冲器仍为拴锁状态。

14.3.3 定时器 x 预置数寄存器 (TPMxMODH:TPMxMODL)

当定时器计数值等于预置数时，并且 CPWMS=0，那么计数器寄存器的值恢复为 0x0000。如 CPWMS=1，开始向下计数；同时 TOF 被置为 1。向这两个寄存器之一的写操作将屏蔽 TOF 和中断功能，直到另一个寄存器被写。复位会使这两个寄存器变为 0x0000。定时器运行于自由计数模式。

	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

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

在修改模寄存器值之前，复位 TPM 计数值，从而避免当模计数值改变后，计数器的不正确溢出。

14.3.4 定时器 x 通道 n 状态和控制寄存器 (TPMxCnSC)

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

 = Unimplemented or Reserved

7 CHnF	<p>通道 n 标志：</p> <p>当通道 n 配置为输入捕捉模式时，通道 n 上的有效边沿触发该位置位。</p> <p>当通道 n 配置为比较输出或边沿对齐 PWM 模式时，TPM 计数寄存器中的值等于 TPM 通道 n 的寄存器值时，CHnF 置位。</p> <p>当该通道作为中心对齐 PWM 输出时，该标志位很少使用。</p> <p>如果 CHnIE=1,那么当 CHnF 置位，就会产生相应中断事件。CHnF 不会自动清除，人工清除方法：读取 TPMxCnSC 寄存器，然后向 CHnF 位写 0。</p> <p>如果在清除 CHnF 位过程中，另一个中断被请求，那么清除过程被复位，CHnF 保持置位状态。之所以这样做，是因为为了保证中断事件不丢失。</p> <p>0：在通道 n 上无输入捕捉或输出比较事件发生</p> <p>1：通道 n 发生输入捕捉或输出比较事件</p>
6 CHnIE	<p>通道 n 中断允许位：</p> <p>0：通道 n 中断请求不允许；1：通道 n 中断请求允许</p>
5 MSnB	通道模式选择位，见下表
4 MSnA	通道模式选择位，见下表
3: 2 ELSn[B:A]	边沿/电平选择位，见下表

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00		Pin not used for TPM channel; use as an external clock for the TPM or revert to general-purpose I/O
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

在改变通道配置之后，在使能通道中断之前，清除状态标志，以避免发生不希望的事件。

14.3.5 计时器 x 通道值寄存器 (TPMxCnVH:TPMxCnVL)

	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

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

在输入捕捉模式下，读取任一寄存器值将门锁这两个寄存器的值到一个缓冲器中，这两个值维持门锁状态，直到另一个字节被读取。当 TPMxCnSC 寄存器被改写时，这种门锁机制被复位。在输入捕捉模式下，对通道寄存器的写入值被忽略掉。

在比较输出或 PWM 模式中，写任意一个通道寄存器，将被门锁到一个缓冲器中，只有这两个寄存器都被写，它们才真正传送到通道寄存器中。

14.4 功能描述

所有的 TPM 功能与主要部件 16 位计数器相关联。

14.4.1 计数

复位后，CLKSB:CLKSA=0:0；无时钟源，TPM 未激活。通常，CLKSB:CLKSA=0:1，选择总线时钟驱动定时器计数。每个 TPM 可独立选择时钟源。当选择外部时钟输入时，允许的最大输入时钟为总线频率的四分之一。

MCU 处于后台调试模式时，TPM 临时挂起所有计数，直到 MCU 回到正常操作模式。在 STOP 模式，所有的 TPM 时钟停止；在 WAIT 模式，TPM 继续正常运行。

作为递增计数，16 位计数器从 0x0000 到终点值，之后继续从 0x0000 计起。

终点值为 0xFFFF 或预置值。

当处于中心对齐 PWM 模式时，重复从 0x0000 递增到终点值，然后递减到 0x0000 的过程。

TOF 和 TOIE 配合使用可以实现查询溢出溢出事件或中断。

在递增模式，16 位计数器从 0x0000 计起，当到达 0xFFFF 后，在一个计数时钟，溢出自动变为 0x0000。TOF 在从 0xFFFF 转变到 0x0000 后置 1。当预置数起作用时，TOF 在计数器由预置值变为 0x0000 时置位。

一致性机制：当读取计数器寄存器任意一个计数寄存器时，这两个寄存器的值被锁存到一个缓冲器中，计数器正常计数，此时缓冲器中不会更新，直到都读取一遍。

通过写任意值到计数寄存器之一，将人工复位计数器。

14.4.2 通道模式选择

14.4.2.1 输入捕捉模式

用于捕捉发生在外部引脚事件时间。当一个有效的边沿触发输入捕捉通道，TPM 将 TPM 计数寄存器中的值锁存到通道寄存器中。

通道寄存器遵循一致性原则。

可以产生中断请求。

14.4.2.2 比较输出模式

用于产生定时脉冲。

当计数器计数值等于通道寄存器值时，TPM 置位，降清除或触发通道引脚电平信号。

修改通道寄存器值遵循一致性原则

可以产生中断。

14.4.2.3 边沿 PWM 模式

PWM 输出的一种典型方式为计数器正常递增计数 (CPWMS=0)，同时同一 TPM 模块的其他通道配置为输入捕捉或输出比较功能。PWM 信号的周期由模寄存器 (TPMxMODH:TPMxMODL) 决定。占空比由通道值寄存器 (TPMxCnVH:TPMxCnCL) 决定。而 PWM 的信号极性由 ELSnA 控制位决定。占空比为 0% 或 100% 是有可能的。下图所示：

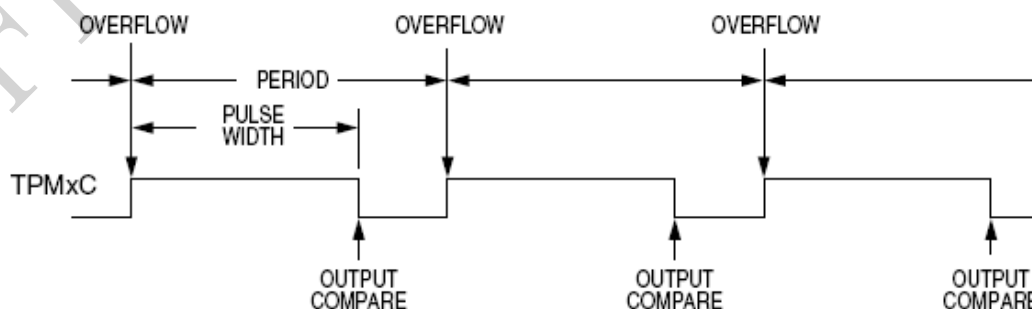


图 14.2 边沿 PWM 周期与占空比 (ELSnA=0)

TPMxMODH:TPMxMODL 用于决定 PWM 周期。

TPMxCnVH:TPMxCnVL 用于决定一个周期内高电平时间，即脉宽时间 Tpulse。

ELSnA 决定信号变化。 $=0$ ：计数溢出引脚输出高电平，比较输出低电平。 $=1$ ：反之。

当通道值寄存器设置为 0X0000，占空比为 0%。

如果通道值寄存器的值等于或大于模寄存器的值，占空比为 100%。

工作过程 (ELSnA=0)：

当用户程序设置好 Tcycle 和 Tpulse 及 ELSnA 后，计数器从 0x0000 开始计数，此时通道引脚为高电平，当计数值等于 Tpulse 时，通道引脚变为低电平，计数值继续递增，当计数值等于 Tcycle 时，溢出，通道引脚变为高电平，重复该过程。

14.4.3 中心对齐 PWM 模式

当 CPWMS=1 时，计数器会递增/递减计数模式。TPMxCnVH:TPMxCnVL 中的值决定 PWM 的脉宽；TPMxMODH:TPMxMODL 决定 PWM 的周期。

TPMxMODH:TPMxMODL 的值设置范围为 0X0001~0X7FFF，超过这个范围产生不确定的结果。ELSnA 决定 CPWM 输出的极性。

$$\text{pulse width} = 2 \times (\text{TPMxCnVH:TPMxCnVL})$$

$$\text{period} = 2 \times (\text{TPMxMODH:TPMxMODL});$$

for TPMxMODH:TPMxMODL = 0x0001–0x7FFF

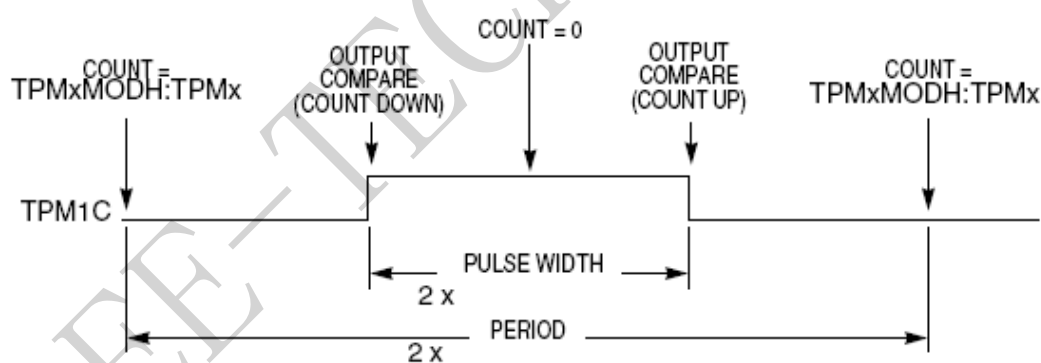


图 14.3 CPWM 周期与占空比

当 TPMxCnVH:TPMxCnVL=0 或最高位=1，占空比为 0%。

当 TPMxCnVH:TPMxCnVL（最高位=0）大于 TPMxMODH:TPMxMODL（非 0），占空比为 100%。

工作过程：

计数器从 0 开始递增，通道引脚为高电平，当计数值等于比较输出计数值时，通道引脚强制为低电平，计数值继续递增；当计数值等于预置数值时，开始递减；当计数值递减到比较输出数值时，通道引脚强制为高电平，计数值继续递减；当计数值递减到零时，开始递增。

由于中心对齐 PWM 输出降低噪声。

PMxCnVH:TPMxCnVL 和 TPMxMODH:TPMxMODL 实际上是写缓冲器。

当 $\text{TPMxCNTH}:\text{TPMxCNTL} = \text{TPMxMODH}:\text{TPMxMODL}$ 时, 计数器溢出。

14.5 TPM中断

TPM在计数溢出或每个通道产生中断。如果通道配置为输入捕捉, 当通道引脚出现有效边沿时触发。当通道配置为比较输出或PWM模式时, 每当计数器匹配通道寄存器中的值触发。

对于TPM中的每一个中断源, 有一个相应的标志位进行标识。该标志位可被软件轮询到或向CPU申请中断。

14.5.1 清除计时器中断标志

分两步。读取该位然后向该位写 0。

14.5.2 定时器溢出中断

定时器自由运行模式时, 当计数器的值由 $0xFFFF$ 变为 $0x0000$ 时, TOF 置位。

定时器运行于预置数模式时, 当计数器的值由预置数变为 $0x0000$ 时, TOF 置位。

定时器运行于递增/递减模式时, 当改变计数方向时, 该位置位。

14.5.3 通道事件中断

通道事件包括输入捕捉, 比较输出, 边沿对齐 PWM 或中心对齐 PWM

14.5.4 PWM 事件

当通道配置为边沿对齐 PWM, 当计数器的值与通道寄存器的值匹配时, 通道标志被置位。

当通道配置为中心对齐 PWM, 计数器的值会与通道寄存器的值匹配两次, 故通道标志在脉冲的起始和结束均置位。

第十五章 开发工具

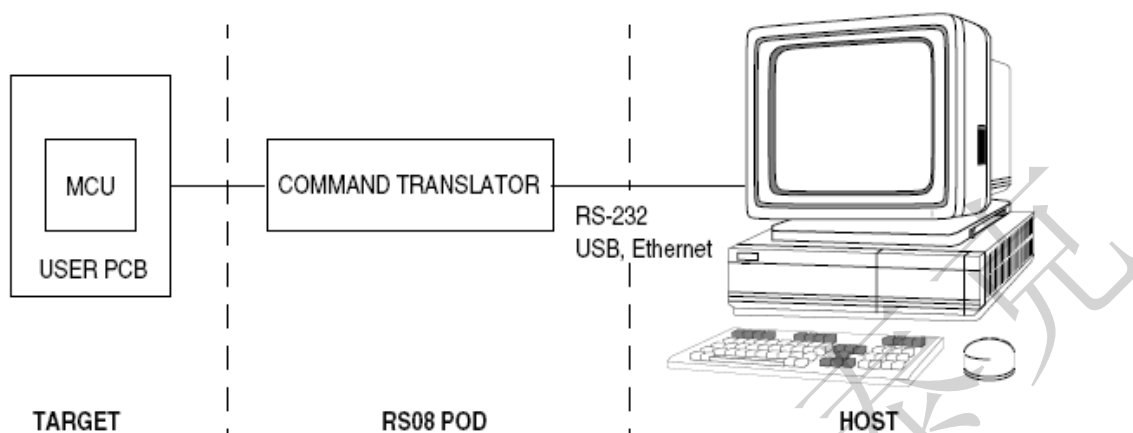


图 15—1 系统调试连接图

HOST：主机：运行 CODEWARRIOR 开发软件。

RS08 POD：协议转换器(调试工具)：完成对 HOST 下发命令的解析转发和 TARGET 上传数据的解析转发。

USER PCB：用户开发板。

背景调试(BDC)原理：当用户在 CODEWARRIOR 环境下开发的应用程序编译通过后，运行调试，主机通过调试工具将目标代码下载到用户开发板的 MCU 中。主机将调试命令通过调试工具转发给开发板上的 MCU，相应的开发板上的 MCU 通过调试工具应答 HOST 的调试命令。

15.1 背景调试特点

- *通过一根双向数据引脚(即 PTA3/ACMP0/BKGD/MS)

- *背景调试命令不会干扰用户存储器。BDC 寄存器不位于存储器映射空间。

- *同步命令使主机与目标 MCU 通信速率相同。

- *CPU 处于运行状态下，非干扰调试命令仍能访问存储器。

- *背景调试命令可访问 CPU 寄存器。

- *连续运行(GO)和单步跟踪(TRACE1)命令。

- *BACKGROUND 命令可以唤醒处于等待或停止模式的 CPU。

- *BDC_RESET 命令运行主机复位 MCU。

- *MCU 处于背景调试模式下，看门狗暂停。

- *当 CPU 处于停止模式下，但运行背景调试，那么 MCU 时钟源在停止模式下会仍运行。

- *有一个硬件地址断点。

15.2 背景调试控制器

背景调试命令分为两类：

- *激活背景调试命令：要求 MCU 处于活动的背景调试模式(用户程序不再运行)。

例如：BACKGROUND, TRACE1, GO 命令。

- *非干扰命令：该命令可以在用户程序运行情况下的任何时刻运行，用于读或写 MCU 的存储器或状态和控制寄存器。

15.3 背景调试接口

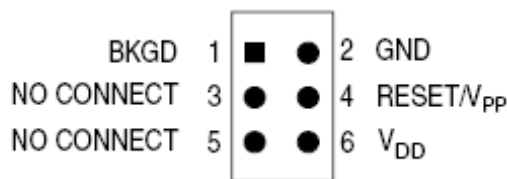


图 12-2 背景调试接口

其中 BKGD 背景调试通信数据线。VDD, GND 电源。RESET 用于通过硬件方法主机强制或查询 MCU 复位。

BKGD 引脚为半开漏引脚，内有上拉，因此不需要外部上拉电阻。与典型开漏引脚不同，外部 RC 影响该引脚的 RC 时间常数对通信数据的上升沿影响很小。复位期间，该引脚可控制 MCU 运行于背景调试模式或正常运行模式。该引脚对 HOST 的 SYNC 命令做应答，从而使主机获知目标 MCU 的通信速率。

通过控制 BKGD 引脚和强制 MCU 复位(主机下发 BDC_RESET 命令或上电复位)，主机可强制 MCU 进入背景调试模式。当 MCU 未连接调试工具时，MCU 运行于正常模式。

有些器件 BKGD 引脚与只输出功能复用。为了支持 BDM 调试，用户必须关闭输出功能。在程序正常模式下，测试该引脚的输出功能。

15.4 背景调试通信

通信过程中，先发送数据最高位，每位耗时 16 个 BDC 时钟(系统总线时钟)。

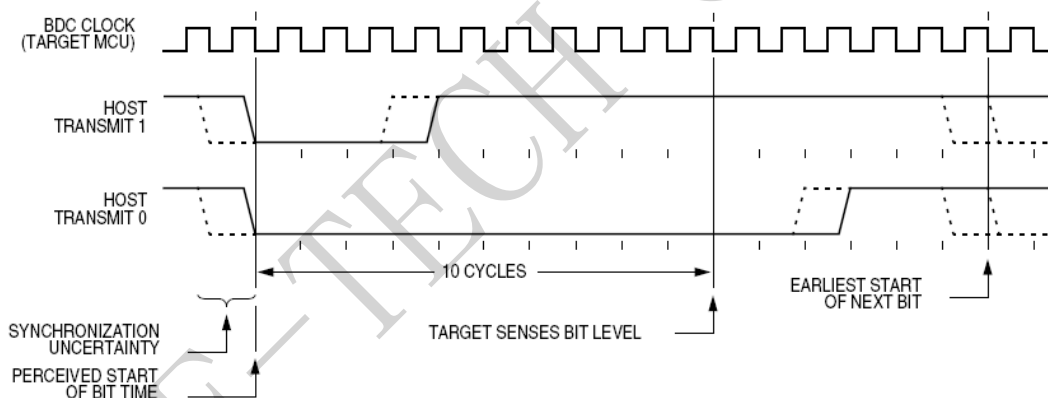


图 15-3 主机发送给目标 MCU 一位数据时序图

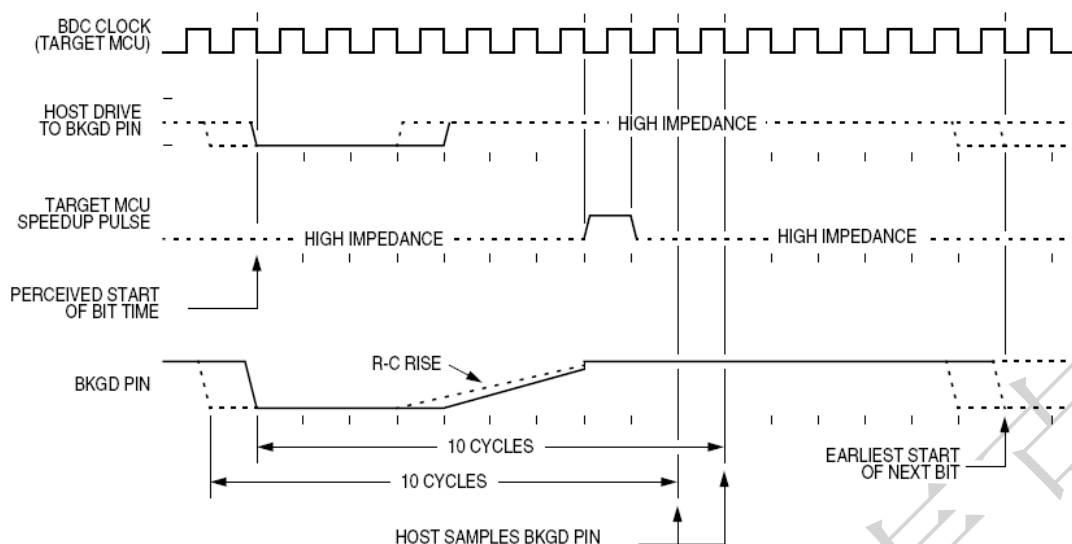


图 15-4 目标 MCU 发送给主机数据位“1”

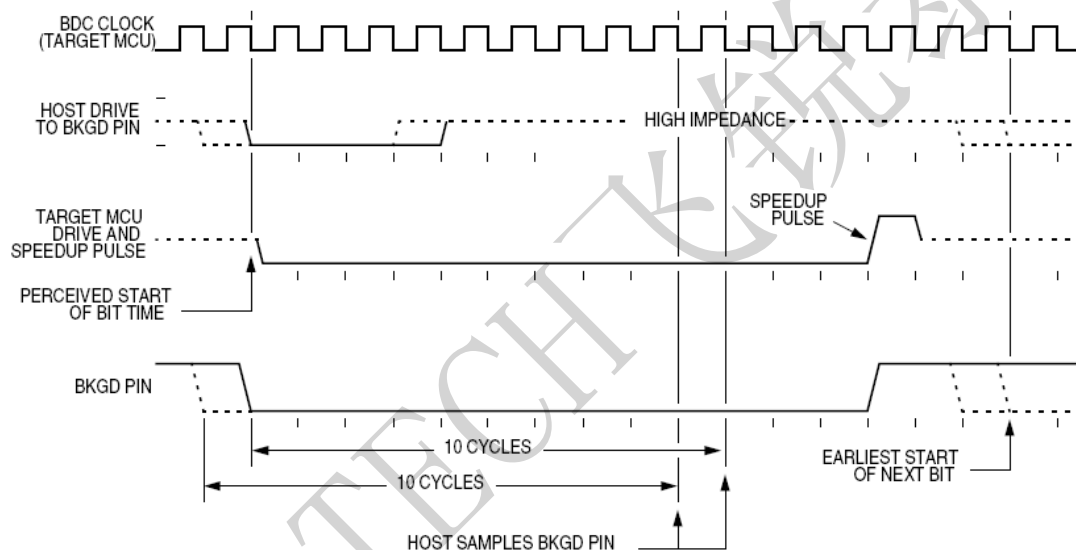


图 15-4 目标 MCU 发送给主机数据位“0”

15.5 同步和串口通信超时

主机通过在 BKGD 引脚产生一个下降沿来发起主机—目标 MCU 的串口通信。如果 BKGD 保持低电平时间超过 128 个目标 MCU 的时钟周期，那么目标 MCU 认为主机发出了 SYNC 命令。这种情况下，目标 MCU 应等待 BKGD 引脚上出现上升沿，然后对 SYNC 命令做应答。如果目标 MCU 未检测在 BKGD 引脚上的上升沿，那么目标 MCU 将一直处于未确定的等待状态，该等待没有超时限制。当主机发出一个有效的 SYNC 命令请求后，BKGD 上有上升沿出现，那么目标 MCU 将驱动 BKGD 引脚为低，持续时间为 128 个时钟周期，作为对主机 SYNC 命令的应答，从而主机获知了目标 MCU 的通信速率。

主机发出 SYNC 命令后，在 128 个时钟周期之前返回高电平，那么目标 MCU 会认为主机发出一个有效的位传送，不认为是 SYNC 请求。目标 MCU 将等待下一个下降沿（即下一个位传输的开始）。这种情况下，目标 MCU 在 512 个周期内未检测到下一个下降沿，会产生超时错误，那么当前的命令不会影响目标 MCU 的存储器或操作模式。

如果主机发出一个读取数据命令，但目标 MCU 在 512 个串口时钟周期内未返回数据，那么主机会发出一个复位命令，取消上次的读数据命令。也就是超时返

回数据无效。软件复位命令也用于 READ_BLOCK 或 WRITE_BLOCK 命令的结束。

主机发送数据位“1”或“0”，且不产生超时错误以及有效的 SYNC 命令时间要求。

发送“0”：BKGD 引脚保持低电平至少 12 个周期，最大 511 个周期。该时间要求适用于 SYNC 命令。

发送“1”：BKGD 引脚必须保持低电平至少 4 个周期，在第八个周期应释放该引脚，然后保持高电平且至少到第 16 个周期为止。


位序列之间不能超过 512 个周期。

15.6 背景调试寄存器和控制位

以下寄存器用户程序是访问不到的。

15.6.1 背景状态和控制寄存器

	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

7 ENBDM	后退调试使能位(通常该位在开始调试或主机复位目标 MCU 后, 由主机将该位改写为“1”。只有目标 MCU 正常复位才能清除。如果在停止模式下仍要求调试, 目标 MCU 应设置该位为“1”) 0: 背景调试不能激活 1: 背景调试允许激活
6 BDMACT	后退调试模式活动状态标志: 只读。 0: 后退调试未激活 1: 后退调试激活等待串口命令
5 BKPTEN	背景调试断点使能位。 0: 背景调试断点不允许 1: 背景调试断点使能
4 FTS	强制/标记选择: FTS=1, 只要 CPU 地址总线与 BDCBKPT 寄存器匹配, 请求一个端点。FTS=0, 当 CPU 地址总线与 BDCBKPT 寄存器匹配, 提取的操作码被标记。如果标记的操作码曾位于指令队列的结尾, CPU 进入背景调试模式。
2 WS	0: 目标 MCU 运行用户应用程序或激活的背景调试模式。 1: 允许 BACKGROUND 命令唤醒处于等待或停止模式的 MCU
1 WSF	0: 访问存储器与 WAIT 或 STOP 命令不冲突 1: CPU 进入 WAIT 或 STOP 模式, 访问存储器错误

15.4.2 背景调试断点匹配寄存器

15.5 背景调试命令

详细内容参考英文文档 RS08 BDC commands 部分。