

XC866

8 位单片微控制器

8bit

Microcontrollers



Never stop thinking

Edition 2005-03

**Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
81669 München, Germany**

**© Infineon Technologies AG 3/25/05.
All Rights Reserved.**

Attention please!

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

XC866

8 位单片微控制器

Microcontrollers



Never stop thinking

XC866

版本信息:

2005 - 03

V 0.3D1

前版本:

期待您的指正

本手册中如有不当、错误及遗漏之处，敬请批评指正，以便我们不断改进用户手册的质量。

目录:

1	介绍	1-1
1.1	特性总结	1-3
1.2	引脚配置	1-5
1.3	引脚定义及功能	1-6
1.4	命名规则	1-12
1.5	保留位, 未定义位及未实现位	1-13
1.6	缩写	1-14
2	处理器结构	2-1
2.1	功能概述	2-2
2.2	CPU 寄存器描述	2-4
2.2.1	堆栈指针 (SP)	2-4
2.2.2	数据指针 (DPTR)	2-4
2.2.3	累加器 (ACC)	2-4
2.2.4	B 寄存器	2-4
2.2.5	程序状态字	2-5
2.2.6	扩展操作寄存器 (EO)	2-6
2.2.7	功率控制寄存器 (PCON)	2-7
2.3	指令时序	2-8
3	存储器结构	3-1
3.1	程序存储器	3-3
3.2	数据存储器	3-3
3.2.1	内部数据存储器	3-3
3.2.2	外部数据存储器	3-3
3.3	特殊功能寄存器	3-4
3.3.1	映射地址扩展	3-4
3.3.2	分页地址扩展	3-6
3.3.3	位寻址	3-10
3.3.4	系统控制寄存器	3-10
3.3.4.1	位保护方案	3-12
3.3.5	XC866 寄存器概览	3-13
3.3.5.1	CPU 寄存器	3-13
3.3.5.2	系统控制寄存器	3-16

3.3.5.3	WDT 寄存器	3-19
3.3.5.4	端口寄存器	3-20
3.3.5.5	ADC 寄存器	3-23
3.3.5.6	定时器 2 寄存器	3-29
3.3.5.7	CCU6 寄存器	3-30
3.3.5.8	SSC 寄存器	3-38
3.3.5.9	OCDS 寄存器	3-40
3.4	Boot ROM 工作模式	3-42
3.4.1	用户模式	3-42
3.4.2	引导程序加载器模式	3-42
3.4.3	OCDS 模式	3-43
4	闪存存储器	4-1
4.1	Flash 存储器映射	4-2
4.2	Flash Bank 分区	4-3
4.3	字线地址	4-5
4.4	工作模式	4-7
4.5	检错和纠错	4-8
4.6	在系统编程	4-9
4.7	在应用编程	4-10
4.7.1	D-Flash 编程	4-11
4.7.2	D-Flash 擦除	4-13
5	中断系统	5-1
5.1	非屏蔽中断	5-1
5.2	可屏蔽中断	5-1
5.2.1	内部中断	5-1
5.2.2	外部中断	5-2
5.2.3	扩展中断	5-2
5.3	中断源和中断向量	5-8
5.4	中断寄存器描述	5-9
5.4.1	中断使能寄存器	5-9
5.4.2	中断请求标志	5-15
5.4.3	中断优先级寄存器	5-21
5.4.4	中断优先级	5-24

5.4.5	中断请求标志	5-26
5.5	中断处理	5-28
5.6	中断响应时间	5-30
6	并行端口	6-1
6.1	基本端口操作	6-2
6.1.1	基本寄存器描述	6-5
6.1.1.1	数据寄存器	6-6
6.1.1.2	方向寄存器	6-6
6.1.1.3	开漏输出控制寄存器	6-7
6.1.1.4	上拉/下拉器件寄存器	6-7
6.1.1.5	其他输入功能	6-9
6.1.1.6	其他输出功能	6-9
6.2	寄存器映射	6-10
6.3	P0 口	6-12
6.3.1	功能	6-12
6.3.2	寄存器描述	6-15
6.4	P1 口	6-18
6.4.1	功能	6-18
6.4.2	寄存器描述	6-21
6.5	P2 口	6-24
6.5.1	功能	6-24
6.5.2	寄存器描述	6-27
6.6	P3 口	6-29
6.6.1	功能	6-29
6.6.2	寄存器描述	6-32
7	电源，复位和时钟管理	7-1
7.1	内嵌电压调节器的电源系统	7-1
7.2	复位控制	7-3
7.2.1	复位类型	7-3
7.2.1.1	上电复位	7-3
7.2.1.2	硬件复位	7-4
7.2.1.3	看门狗定时器复位	7-4
7.2.1.4	掉电唤醒复位	7-4

7.2.1.5	压降复位.....	7-5
7.2.2	模块复位行为.....	7-6
7.2.3	启动方案.....	7-6
7.2.4	寄存器描述.....	7-7
7.3	时钟系统.....	7-9
7.3.1	时钟产生单元.....	7-9
7.3.1.1	功能描述.....	7-10
7.3.2	时钟源控制.....	7-11
7.3.3	时钟管理.....	7-13
7.3.4	寄存器描述.....	7-15
8	省电模式.....	8-1
8.1	功能描述.....	8-2
8.1.1	空闲模式.....	8-2
8.1.2	减速模式.....	8-2
8.1.3	掉电模式.....	8-3
8.1.4	外设时钟管理.....	8-4
8.2	寄存器描述.....	8-5
9	看门狗定时器.....	9-1
9.1	功能描述.....	9-2
9.2	寄存器映射.....	9-5
9.3	寄存器描述.....	9-6
10	串行接口.....	10-1
10.1	UART.....	10-2
10.1.1	UART 模式.....	10-2
10.1.1.1	模式 1, 8 位 UART,波特率可变.....	10-2
10.1.1.2	模式 2, 9 位 UART,波特率固定.....	10-5
10.1.1.3	模式 3, 9 位 UART,波特率可变.....	10-5
10.1.2	多处理器通信.....	10-7
10.1.3	寄存器描述.....	10-7
10.1.4	波特率产生.....	10-9
10.1.4.1	波特率发生器.....	10-10
10.1.5	UART 接口.....	10-13
10.2	局域网互联网 (LIN).....	10-15

10.2.1	LIN 协议	10-15
10.2.2	LIN 报文头传输	10-17
10.2.3	LIN 波特率检测	10-18
10.3	高速同步串行接口	10-20
10.3.1	基本操作	10-21
10.3.1.1	工作模式选择	10-21
10.3.1.2	全双工工作	10-22
10.3.1.3	半双工工作	10-25
10.3.1.4	连续传输	10-26
10.3.1.5	端口控制	10-26
10.3.1.6	波特率产生	10-27
10.3.1.7	检错机制	10-28
10.3.2	中断	10-31
10.3.3	寄存器映射	10-32
10.3.4	寄存器描述	10-33
10.3.4.1	端口输入选择寄存器	10-33
10.3.4.2	控制寄存器	10-34
10.3.4.3	波特率定时器重载寄存器	10-39
10.3.4.4	发送和接收缓冲寄存器	10-40
11	定时器	11-1
11.1	定时器 0 和定时器 1	11-1
11.1.1	定时器基本操作	11-1
11.1.2	定时器工作模式	11-2
11.1.2.1	模式 0	11-3
11.1.2.2	模式 1	11-4
11.1.2.3	模式 2	11-5
11.1.2.4	模式 3	11-6
11.1.3	寄存器映射	11-7
11.1.4	寄存器描述	11-8
11.2	定时器 2	11-13
11.2.1	自动重载模式	11-13
11.2.1.1	禁止递增/递减计数	11-13
11.2.1.2	使能递增/递减计数	11-14

11.2.2	捕获模式	11-16
11.2.3	寄存器映射	11-17
11.2.4	寄存器描述	11-17
12	捕获/比较单元 6	12-1
12.1	功能描述	12-3
12.1.1	定时器 T12	12-3
12.1.1.1	定时器设置	12-4
12.1.1.2	计数规则	12-5
12.1.1.3	切换规则	12-5
12.1.1.4	T12 的比较模式	12-6
12.1.1.5	0%和 100%的占空比	12-7
12.1.1.6	死区时间产生	12-8
12.1.1.7	捕获模式	12-9
12.1.1.8	单次模式	12-10
12.1.1.9	类磁滞控制模式	12-10
12.1.2	定时器 T13	12-12
12.1.2.1	定时器设置	12-12
12.1.2.2	比较模式	12-13
12.1.2.3	单次模式	12-13
12.1.2.4	T13 与 T12 同步	12-13
12.1.3	调制控制	12-15
12.1.4	强制中断处理	12-18
12.1.5	多通道模式	12-19
12.1.6	霍尔传感器模式	12-21
12.1.6.1	霍尔序列采样	12-21
12.1.6.2	无刷直流控制	12-22
12.1.7	中断产生	12-26
12.1.8	端口连接	12-26
12.2	寄存器映射	12-29
12.3	寄存器描述	12-32
12.3.1	系统寄存器	12-35
12.3.1.1	端口输入选择	12-35
12.3.2	定时器 T12 的相关寄存器	12-39

12.3.3	定时器 T13 相关寄存器	12-45
12.3.4	捕获/比较控制寄存器	12-49
12.3.5	调制控制寄存器.....	12-62
12.3.5.1	全局模块控制	12-62
12.3.5.2	多通道控制	12-70
12.3.6	中断控制寄存器.....	12-84
13	模数转换单元.....	13-1
13.1	结构概述	13-2
13.2	时钟方案	13-3
13.2.1	转换时间	13-4
13.3	低功耗模式.....	13-7
13.4	功能描述	13-8
13.4.1	请求源仲裁.....	13-9
13.4.2	转换启动模式	13-10
13.4.3	通道控制	13-10
13.4.4	顺序请求源.....	13-11
13.4.4.1	概述.....	13-11
13.4.4.2	请求源控制	13-12
13.4.5	并列请求源.....	13-13
13.4.5.1	概述.....	13-13
13.4.5.2	请求源控制	13-13
13.4.5.3	外部触发.....	13-14
13.4.5.4	软件控制.....	13-14
13.4.5.5	自动扫描.....	13-14
13.4.6	待读模式	13-15
13.4.7	转换结果的产生.....	13-16
13.4.7.1	概述.....	13-16
13.4.7.2	极限检查.....	13-17
13.4.7.3	数据压缩滤波	13-18
13.4.7.4	FIFO 型结果寄存器.....	13-19
13.4.7.5	结果寄存器的读取方式	13-19
13.4.8	中断	13-21
13.4.8.1	事件中断.....	13-22

13.4.8.2	通道中断.....	13-24
13.4.9	外部触发输入.....	13-26
13.5	ADC 初始化序列.....	13-27
13.6	寄存器映射.....	13-30
13.7	寄存器描述.....	13-33
13.7.1	基本功能寄存器.....	13-33
13.7.2	优先级和仲裁寄存器.....	13-35
13.7.3	外部触发控制寄存器.....	13-37
13.7.4	通道控制寄存器.....	13-38
13.7.5	输入综合控制寄存器.....	13-39
13.7.6	顺序请求源寄存器.....	13-40
13.7.7	并列请求源寄存器.....	13-47
13.7.8	结果寄存器.....	13-51
13.7.9	中断寄存器.....	13-55
14	片上调试支持.....	14-1
14.1	功能描述.....	14-2
14.2	调试.....	14-3
14.2.1	调试事件.....	14-3
14.2.1.1	硬件断点.....	14-4
14.2.1.2	软件断点.....	14-5
14.2.1.3	外部断点.....	14-6
14.2.2	调试动作.....	14-6
14.2.2.1	调用监控器程序.....	14-6
14.2.2.2	激活 MBC 引脚.....	14-6
14.3	寄存器描述.....	14-8
14.3.1	JTAG 标识寄存器.....	14-10
15	索引.....	15-1
15.1	关键词索引.....	15-1
15.2	寄存器索引.....	15-8

1 介绍

XC866 是高性能 8 位微控制器 XC800 的系列产品，基于和标准 8051 处理器兼容的 XC800 核。XC866 高度集成片内器件，例如片内振荡器或嵌入式电压调节器（从而可由 3.3V 或 5.0V 的单电源供电），因此具备了许多增强功能以满足新型应用。此外，嵌入式闪存（Flash）器件为系统开发和批量生产提供了很大的灵活性；兼容 ROM 器件为大批量生产提供了节省成本的空间。

多闪存组（Flash Bank）结构支持在应用编程（IAP），运行某 bank 中的用户程序的同时可编程或擦除另一 bank 中的内容。可通过位于启动 ROM（Boot ROM）中的引导程序加载器（BSL）进行在系统编程（ISP），通过外部主机（例如 PC）对嵌入式 FLASH 方便的编程或擦除。

XC866 的其他一些主要特性包括：用来产生脉宽调制信号、带有电机控制专用模式的捕获/比较单元（CCU6）；功能扩展的 10 位模数转换单元（ADC），具有如自动扫描和结果累加（用于抗混迭滤波或结果平均）等特性；功能扩展的通用异步收发器（UART），支持局域网互联网（LIN）应用，并为许多器件提供 LIN 的低级驱动程序；提供不同的省电模式选择，以满足低功耗应用；用于优化中断处理的智能分页机制，扩展了控制片内外设功能的特殊功能寄存器（SFR）的地址范围。

XC866 的各功能单元如图 1-1 所示。

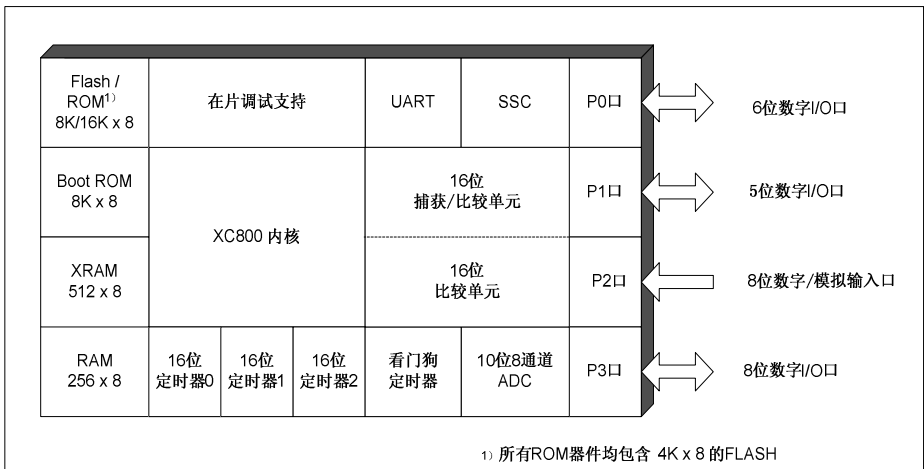


图 1-1 XC866 功能框图

根据配置和程序存储器容量不同，XC866 系列有八种器件，满足不同的应用需求。通常，每种器件包括非易失性 8K × 8 只读程序存储器、易失性 768 × 8 读/写数据存储器、四个端口、三个 16 位定时器、一个 16 位捕获/比较单元、一个 16 位比较定时

器、14 个中断向量（和一个非屏蔽中断 NMI）、四级优先级的中断结构、两个串行接口、多用途的故障保险机制，在片调试支持和一个 10 位模数转换单元。

XC866 的各种器件及差异比较见表 1-1。

表 1-1 器件总结

器件类型	器件名称	FLASH 容量	ROM 容量	支持 LIN
Flash	XC866L-4FR	16K 字节	-	是
	XC866-4FR	16K 字节	-	否
	XC866L-2FR	8K 字节	-	是
	XC866-2FR	8K 字节	-	否
ROM	XC866L-4RR	4K 字节	16K 字节	是
	XC866-4RR	4K 字节	16K 字节	否
	XC866L -2RR	4K 字节	8K 字节	是
	XC866 -2RR	4K 字节	8K 字节	否

除非特殊声明，否则本手册中的“XC866”统指 XC866 系列产品的所有器件。

1.1 特性总结

XC866 的主要特性归纳如下：

- 高性能 XC800 内核
 - 和标准 8051 处理器兼容
 - 2 个时钟的机器周期结构（快速无等待内存访问）
 - 双数据指针
- 片内存储器
 - 8 K 字节 Boot ROM
 - 256 字节 RAM
 - 512 字节 XRAM
 - 8/16K 字节 Flash；或
8/16K 字节 ROM，外加 4K 字节 Flash
- 输入/输出（I/O 口）需 3.3V 或 5.0V 供电；内核需 2.5V 供电（由嵌入式电压调节器产生）
- 上电复位
- 内核压降检测
- 片内振荡器（OSC）和锁相环（PLL）产生时钟
 - PLL 失锁检测
- 省电模式
 - 减速模式
 - 空闲模式
 - 通过 RXD 或 EXINT0 将系统从掉电模式中唤醒
 - 每种外设均由时钟门控制
- 可编程 16 位看门狗定时器（WDT）
- 四个端口
 - 19 个数字 I/O 引脚
 - 8 个数字/模拟输入引脚
- 8 通道，10 位模数转换单元（ADC）
- 三个 16 位定时器
 - 定时器 0 和定时器 1（T0 和 T1）
 - 定时器 2
- 产生脉宽调制信号（PWM）的捕获/比较单元（CCU6）

- 全双工串行接口 (UART)
- 同步串行接口 (SSC)
- 片上调试支持
 - 1K 字节监控器 ROM (8K 字节 Boot ROM 的一部分)
 - 64 字节监控器 RAM
- PG-TSSOP-38 引脚封装
- 温度范围 T_A :
 - SAF (-40 至 85°C)
 - SAK (-40 至 125°C)

XC866 的方框图如图 1-2 所示。

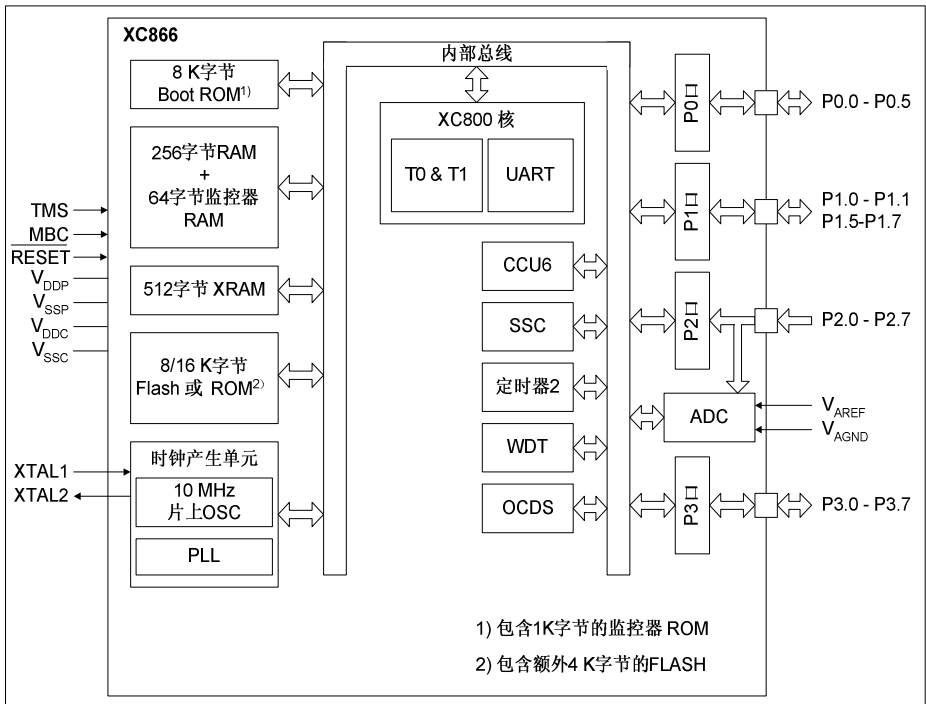


图 1-2 XC866 方框图

1.2 引脚配置

基于 PG-TSSOP-38 封装的 XC866 引脚配置如所示图 1-3 示。

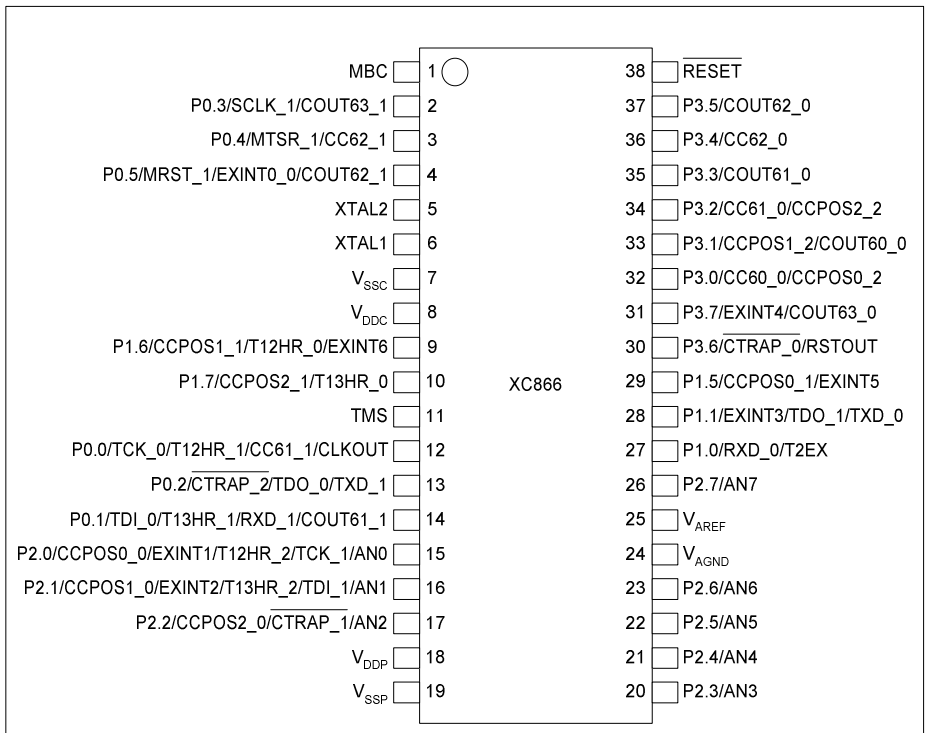


图 1-3 XC866 引脚配置，PG-TSSOP-38 封装（顶视图）

1.3 引脚定义及功能

复位后所有引脚被配置为具有下列特性之一的输入引脚：

- 上拉器件使能 (PU)
- 下拉器件使能 (PD)
- 上拉/下拉器件均被禁止的高阻态 (Hi-Z)

表 1-2 给出 XC866 的引脚功能及缺省状态。

表 1-2 引脚定义和功能

符号	引脚编号	类型	复位状态	功能
P0		I/O		P0 口 P0 口是 6 位通用 I/O 双向口。还可用作 JTAG, CCU6, UART 和 SSC 片内外设的 I/O 口。
P0.0	12		高阻	TCK_0 JTAG 时钟输入 T12HR_1 CCU6 定时器 12 硬件运行输入 CC61_1 捕获/比较通道 1 输入/输出
P0.1	14		高阻	CLKOUT OSC 时钟输出 TDI_0 JTAG 串行数据输入 T13HR_1 CCU6 定时器 13 硬件运行输入 RXD_1 UART 接收输入 COUT61_1 捕获/比较通道 1 输出
P0.2	13		上拉	$\overline{\text{CTRAP}}_2$ CCU6 强制中断输入 TDO_0 JTAG 串行数据输出 TXD_1 UART 发送输出

P0.3	2		高阻	SCK_1	SSC 时钟输入/输出
				COU63_1	捕获/比较通道 3 输出
P0.4	3		高阻	MTSR_1	SSC 主机发送输出/ 从机接收输入
				CC62_1	捕获/比较通道 2 输入/输出
P0.5	4		高阻	MRST_1	SSC 主机接收输入/ 从机发送输出
				EXINT0_0	外部中断输入 0
				COU62_1	捕获/比较通道 2 输出

表 1-2 引脚定义和功能

符号	引脚编号	类型	复位状态	功能
P1		I/O		P1 口 P1 口是 5 位通用 I/O 双向口。还可用作 JTAG, CCU6, UART 和 SSC 片内外设的 I/O 口。
P1.0	27		上拉	RXD_0 UART 接收输入 T2EX 定时器 2 外部触发输入
P1.1	28		上拉	EXINT3 外部中断输入 3 TDO_1 JTAG 串行数据输出 TXD_0 UART 发送输出
P1.5	29		上拉	CCPOS0_1 CCU6 霍尔输入 0 EXINT5 外部中断输入 5
P1.6	9		上拉	CCPOS1_1 CCU6 霍尔输入 1 T12HR_0 CCU6 定时器 12 硬件运行输入 EXINT6 外部中断输入 6
P1.7	10		上拉	CCPOS2_1 CCU6 霍尔输入 2 T13HR_0 CCU6 定时器 13 硬件运行输入 P1.5 和 P1.6 可用作 SSC 的软件片选输出。

表 1-2 引脚定义和功能

符号	引脚编号	类型	复位状态	功能
P2		输入		P2 口 P2 口是 8 位通用单向输入口。还可用作 JTAG 和 CCU6 的数字输入；也可用作 ADC 的模拟输入。
P2.0	15		高阻	CCPOS0_0 CCU6 霍尔输入 0 EXINT1 外部中断输入 1 T12HR_2 CCU6 定时器 12 硬件运行输入 TCK_1 JTAG 时钟输入 AN0 模拟输入 0
P2.1	16		高阻	CCPOS1_0 CCU6 霍尔输入 1 EXINT2 外部中断输入 2 T13HR_2 CCU6 定时器 13 硬件运行输入 TDI_1 JTAG 串行数据输入 AN1 模拟输入 1
P2.2	17		高阻	CCPOS2_0 CCU6 霍尔输入 2 $\overline{\text{CTRAP}}_1$ CCU6 强制中断输入 AN2 模拟输入 2
P2.3	20		高阻	AN3 模拟输入 3
P2.4	21		高阻	AN4 模拟输入 4
P2.5	22		高阻	AN5 模拟输入 5
P2.6	23		高阻	AN6 模拟输入 6
P2.7	26		高阻	AN7 模拟输入 7

表 1-2 引脚定义和功能

符号	引脚编号	类型	复位状态	功能
P3		I/O		P3 口 P3 口是 8 位通用 I/O 双向口。还可用作 CCU6 片内外设的 I/O 口。
P3.0	32		高阻	CC60_0 捕获/比较通道 0 输入/输出
P3.1	33		高阻	COOUT60_0 捕获/比较通道 0 输出
P3.2	34		高阻	CC61_0 捕获/比较通道 1 输入 / 输出
P3.3	35		高阻	COOUT61_0 捕获/比较通道 1 输出
P3.4	36		高阻	CC62_0 捕获/比较通道 2 输入/输出
P3.5	37		高阻	COOUT62_0 捕获/比较通道 2 输出
P3.6	30		下拉	$\overline{\text{CTRAP}}_0$ CCU6 强制中断输入 RSTOUT 内部复位状态的复位输出指示
P3.7	31		高阻	EXINT4 外部中断输入 4 COOUT63_0 捕获/比较通道 3 输出

表 1-2 引脚定义和功能

符号	引脚编号	类型	复位状态	功能
V _{DDP}	18	-	-	I/O 口电源 (3.3/5.0V)
V _{SSP}	19	-	-	I/O 口接地
V _{DDC}	8	-	-	内核电源输出 (2.5V)
V _{SSC}	7	-	-	内核电源接地
V _{AREF}	25	-	-	ADC 参考电压
V _{AGND}	24	-	-	ADC 参考接地
XTAL1	6	输入	高阻	片外 OSC 输入 (片内 OSC 的备份单元, 通常不连接)
XTAL2	5	输出	高阻	片外 OSC 输出 (片内 OSC 的备份单元, 通常不连接)
TMS	11	输入	下拉	测试模式选择
RESET	38	输入	上拉	复位输入
MBC	1	输入	上拉	监控器 & 引导程序加载器控制

1.4 命名规则

本手册使用下面的规则命名 XC866 的组成单元：

- XC866 的功能单元用大写表示。例如：“SSC 可和移位寄存器通信”。
- 低电平有效的引脚，符号上方加横杠表示。例如：“复位输入引脚 $\overline{\text{RESET}}$ 用作硬件复位”
- 寄存器中的位域和位通常表示为“寄存器名称.位域”或“寄存器名称.位”。大多数寄存器名称包含模块名称前缀，用下划线“_”和真正的寄存器名称分开。例如“SSC_CON”中“SSC”是模块名称前缀，“CON”才是真正的寄存器名称。
- 变量出现在大小写混用中，用来表示一组处理单元或寄存器。例如，寄存器“CC6xR”表示具有变量 x ($x=0, 1, 2$) 的多个“CC6xR”寄存器。首次使用寄存器时设置变量边界（例如“ $x=0-2$ ”），并在必要时重复出现。
- 缺省为十进制。十六进制常数用下标“H”表示（如 $C0_H$ ）；二进制常数用下标“B”表示（如 11_B ）。
- 某一寄存器的位域、信号组或引脚组共同命名时，表示为“名称[A:B]”，定义了被命名组从 B 到 A 的范围。单独的位、信号或引脚表示为“名称[C]”，同时给出变量 C 的范围（例如 CFG[2:0] 和 TOS[0]）。
- 单位缩写如下：
 - **MHz** = 兆赫兹
 - **μs** = 微秒
 - **kBaud, kbit** = 每秒 1000 个字符/位
 - **MBaud, Mbit** = 每秒 1, 000, 000 个字符/位
 - **Kbyte** = 1024 个存储器字节
 - **Mbyte** = 1, 048, 576 个存储器字节

通常，前缀 *k* 将单位扩大 1000 倍，前缀 *K* 将单位扩大 1024 倍。因此，Kbyte 将数值扩大 1024 倍；kBaud 将数值扩大 1000 倍。前缀 *M* 将单位扩大 1, 000, 000 倍或 1, 048, 576 倍；前缀 μ 将单位缩小 0.000001 倍。例如 1Kbyte 是 1024 字节；1Mbyte 是 1024×1024 字节；1kBaud/kbit 是每秒 1000 个字符/位；1MBaud/Mbit 是每秒 1, 000, 000 个字符/位，1MHz 是 1, 000, 000Hz。

- 数据格式定义如下：
 - **byte** = 8 位

1.5 保留位，未定义位及未实现位

定义寄存器位域时，使用下列规则定义未定义位和未实现位的功能。另外，位和位域类型用表 1-3 中的缩写来定义。

表 1-3 位功能描述

位功能	描述
未实现位	<p>寄存器位域为“0”表示该位未实现功能，具有下列行为：</p> <ul style="list-style-type: none"> – 读取这些位域返回 0 – 写入这些位域不起作用 <p>这些位域被保留。写入时，软件应始终将其设置为 0，从而和后续产品保持兼容；设置为 1 可能导致不可预测的结果。</p>
未定义位	<p>位域中的某些位组合起来标记为“保留”，表示这些位组合时 XC866 的行为未定义。将寄存器设置为未定义的位组合可能导致不可预测的结果。</p> <p>这样的位组合被保留。写入时，软件必须始终将其设置为位域描述表中提供的合法值。</p>
rw	该位或位域可读写。
r	该位或位域只可读。
w	该位或位域只可写，读取时始终返回 0。
h	该位或位域可由硬件修改（如状态位）。该特性可分别和“rw”或“r”组合为“rwh”和“rh”。

1.6 缩写

表 1-4 列出本手册中所使用的缩写。

表 1-4 缩写

ADC	模数转换单元
ALU	算术/逻辑单元
BSL	引导程序加载器
CCU6	捕获/比较单元 6
CGU	时钟产生单元
CPU	中央处理单元
ECC	纠错码
EVR	嵌入式电压调节器
FIFO	先入先出
GPIO	通用输入/输出
IAP	在应用编程
I/O	输入/输出
ISP	在系统编程
JTAG	联合测试行动单元
LIN	局域互连网络
NMI	非屏蔽中断
OCDS	片上调试支持
PC	程序计数器
POR	上电复位
PLL	锁相环
PSW	程序状态字
PWM	脉宽调制

RAM	随机存取存储器
ROM	只读存储器
SFR	特殊功能寄存器
SPI	串行外设接口
SSC	同步串行接口
UART	通用异步收发器
WDT	看门狗定时器

2 处理器结构

XC866 具有和标准 8051 兼容的高性能 8 位中央处理单元（CPU）。8051 的机器周期由 12 个振荡时钟组成，而 XC866 CPU 的机器周期由 2 个时钟组成，可快速、无等待的访问 ROM 和 RAM。访问 Flash 需要额外一个等待状态（一个机器周期），具体描述见章节 2.3。XC866 的指令集由 45% 的单字节，41% 的双字节和 14% 的三字节指令组成。

XC866 的 CPU 提供的调试特性包括：基本的停止/开始，单步执行，断点支持以及对数据存储器、程序存储器和 SFR 的读写操作。

特性：

- 2 个时钟的机器周期结构（快速无等待内存访问）
- 访问 Flash 支持等待状态插入
- 程序存储器下载选择
- 15 个中断源，4 级优先级的中断控制器
- 双数据指针
- 省电模式
- 专用调试模式和调试信号
- 2 个 16 位定时器（定时器 0 和定时器 1）
- 全双工串行接口（UART）

2.1 功能概述

图 2-1 所示为 CPU 的功能方框图。CPU 由指令译码器，运算单元和程序控制单元组成。每条指令由指令译码器译码，产生的内部信号用来控制 CPU 内部各个模块的功能。这些内部信号影响数据传送和控制算术逻辑运算单元（ALU）的操作。

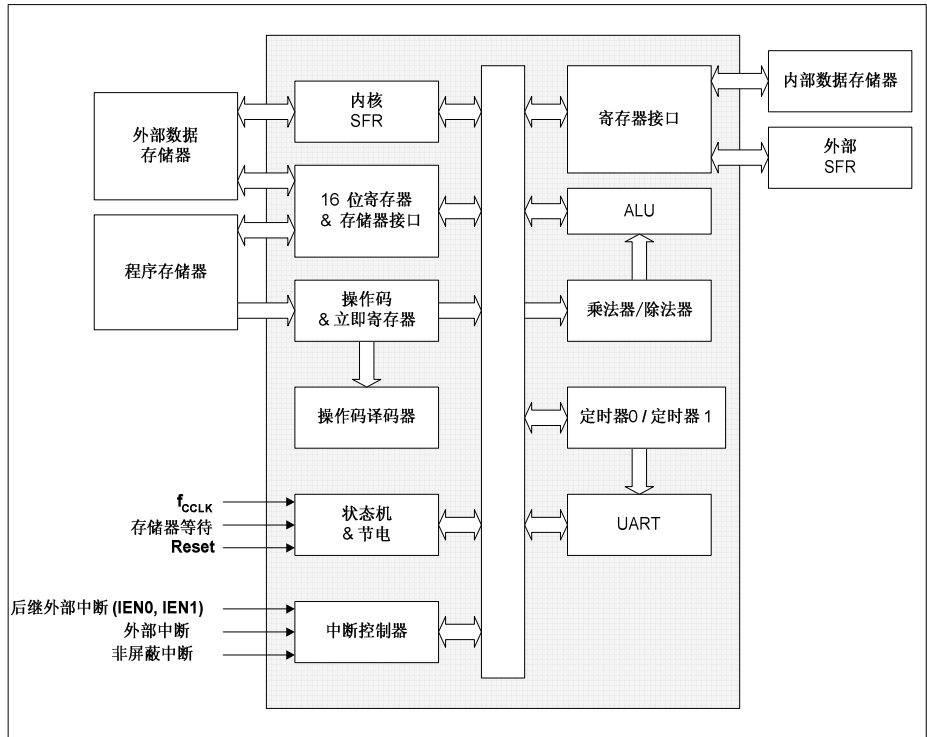


图 2-1 CPU 功能方框图

处理器的运算单元具有强大的数据处理能力，包括 ALU，A 寄存器，B 寄存器和程序状态字寄存器 PSW。

来自一个或两个源地址的 8 位数据字送至 ALU，在译码器的控制下产生 8 位数据结果。ALU 用来完成算术运算和逻辑运算。算术运算包括加，减，乘，除，加 1，减 1，BCD 加法的十进制调整，比较。逻辑运算包括与（AND），或（OR），异或，补码，循环移位（右环移，左环移，或 4 位环移）。ALU 还包括一个布尔处理机，可执行置位，清零，补码，等于 1 跳转，等于 0 跳转，等于 1 跳转并清 0，送入/取自进位位的

位操作。在任何可寻址的位（或该位内容取补码）和进位标志之间，可以进行逻辑与、逻辑或操作，其结果送回到进位标志。

程序控制单元控制程序存储器中指令的执行顺序。16 位程序计数器（PC）中保存下一条将要执行的指令的地址。条件转移逻辑允许处理器响应内部和外部事件，引起程序执行顺序的改变。

2.2 CPU 寄存器描述

CPU 寄存器的地址范围为内部数据存储器的地址段 80_H-FF_H。

2.2.1 堆栈指针 (SP)

堆栈指针寄存器中包含了堆栈指针 SP。在执行 LCALL 和 ACALL 指令时，用 SP 保存 PC 的值；在执行 RET 和 RETI 指令时，从 SP 取回 PC 的值。也可以通过 PUSH 和 POP 指令在堆栈中保存或从堆栈中取回数据内容。压入堆栈前 SP 先自动加 1，出栈后 SP 自动减 1，从而保证了堆栈指针始终指向压入堆栈的最后一个字节，即栈顶。CPU 复位后，SP 总是初始化指向 07_H，所以第一个压入堆栈的数据放在寄存器 bank 0 的地址 08_H 单元中。SP 的读写操作由软件控制。

2.2.2 数据指针 (DPTR)

数据指针(DPTR)存放在寄存器 DPL (数据指针低字节) 和 DPH (数据指针高字节) 中构成 16 位的数据地址，用于外部数据存储器访问 (MOVX A,@DPTR 和 MOVX @DPTR,A)，程序字节传送 (MOVC A,@A+DPTR)，及间接程序调转 (JMP @A+DPTR)。

数据指针支持两个真正的 16 位操作：立即载入(MOV DPTR,# data) 和加 1 (INC DPTR) 操作。

2.2.3 累加器 (ACC)

大多数 ALU 操作中，该寄存器存放其中一个操作数。累加寄存器用 ACC 表示。和累加器相关的指令，其助记符简化为符号“A”。

2.2.4 B 寄存器

乘除法操作时 B 寄存器用来存放第二个操作数，在其他指令中用作暂存器。

2.2.5 程序状态字

程序状态字（PSW）寄存器中存放着反映当前 CPU 状态的状态位。

PSW

程序状态字寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
rw	rwh	rwh	rw	rw	rwh	rwh	rh

符号	位序号	读写类型	功能描述															
P	0	rh	奇偶校验标志 每条指令执行后由硬件置位/清零，表明累加器中“1”的个数为奇数/偶数，如偶校验															
F1	1	rwh	通用标志															
OV	2	rwh	溢出标志 用于算术指令															
RS0 RS1	3 4	rw	寄存器组（Register Bank）选择位 这两位决定（从 4 个 Bank 中）选择哪个 Bank <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>选择 Bank 0，数据地址 00_H - 07_H</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>选择 Bank 1，数据地址 08_H - 0F_H</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>选择 Bank 2，数据地址 10_H - 17_H</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>选择 Bank 3，数据地址 18_H - 1F_H</td> </tr> </tbody> </table>	RS1	RS0	功能	0	0	选择 Bank 0，数据地址 00 _H - 07 _H	0	1	选择 Bank 1，数据地址 08 _H - 0F _H	1	0	选择 Bank 2，数据地址 10 _H - 17 _H	1	1	选择 Bank 3，数据地址 18 _H - 1F _H
RS1	RS0	功能																
0	0	选择 Bank 0，数据地址 00 _H - 07 _H																
0	1	选择 Bank 1，数据地址 08 _H - 0F _H																
1	0	选择 Bank 2，数据地址 10 _H - 17 _H																
1	1	选择 Bank 3，数据地址 18 _H - 1F _H																
F0	5	rwh	通用标志															
AC	6	rwh	辅助进位标志															

符号	位序号	读写类型	功能描述
			用于执行 BCD 操作的指令
CY	7	rwh	进位标志 用于算术指令

2.2.6 扩展操作寄存器 (EO)

指令集还包括一条指令 **MOVC@ (DPTR++) ,A**，允许对程序存储器写操作。初始化 CPU 时用该指令将代码下载到程序存储器中，也用该指令进行软件更新。该指令将累加器中的内容复制到当前数据指针指向的程序存储器地址中，接着数据指针加 1。

该指令的操作码为 **A5_H**，和软件断点指令 **TRAP** 的操作码相同（见表 2-1）。位 **EO.TRAP_EN** 用来选择 **A5_B** 操作码执行的指令。当 **TRAP_EN** 为 0（缺省值）时，**A5_H** 执行 **MOVC** 指令；当 **TRAP_EN** 为 1 时，**A5_H** 执行软件断点指令 **TRAP**，CPU 切换至调试模式执行断点操作。

EO

扩展操作寄存器

复位值: **00_H**

7	6	5	4	3	2	1	0
0		TRAP_EN		0		DPSEL0	
r		rw		r		rw	

符号	位序号	读写类型	功能描述
DPSEL0	0	rw	数据指针选择 0 选择 DPTR0 1 选择 DPTR1
TRAP_EN	4	rw	TRAP 使能位 0 选择 MOVC@ (DPTR++) ,A 1 选择软件断点 TRAP 指令
0	[3:1], [7:5]	r	保留 读操作返回 0；应写入 0

2.2.7 功率控制寄存器 (PCON)


CPU 有两种省电模式：空闲模式和掉电模式。通过控制寄存器 PCON 进入空闲模式。空闲模式下，CPU 的时钟暂停，但定时器，串行口和中断控制器仍以半速时钟工作。掉电模式下，整个 CPU 的时钟全部停止。

PCON

功率控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
SMOD	0			GF1	GF0	0	IDLE
rw	r			rw	rw	r	rw

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
IDLE	0	rw	空闲模式使能位 0 不进入空闲模式 1 进入空闲模式
GF0	2	rw	通用标志位 0
GF1	3	rw	通用标志位 1

2.3 指令时序

对于无等待状态的存储器访问，CPU 的每个机器周期由两个振荡时钟组成，分别由拍 1 (P1) 和拍 2 (P2) 表示，对应两个不同的 CPU 状态。执行指令时，CPU 的状态由机器周期和状态编号 (P1/P2) 共同表示，例如 C2P1 表示第二个机器周期中的第一个时钟周期。可在机器周期的任一拍、或两拍访问存储器；只在拍 2 结束时对 SFR 写入。执行一条指令需要一个、两个或者四个机器周期。通常在一条指令的最后一个机器周期的拍 2 结束时进行寄存器内容的更新和下一个操作码的读取。

可以在任一拍 (P1 或 P2) 开始访问 Flash，指令执行时间延长一个机器周期（插入一个等待状态）。

图 2-2 所示为存取/执行时序的内部状态和节拍。每条指令从 C1P1 开始执行，双字节指令的第二个字节的读取从 C1P1 开始。

图 2-2 (a) 示出两个单字节单周期 (1× 机器周期) 指令的时序图。第一个时序图所示为下一个操作码 (C1P2) 从无等待状态的存储器中读取，指令在一个周期内完成；第二个时序图所示为操作码从 Flash 中读取，插入一个等待状态，相同的指令需要两个机器周期（指令时间延长）完成。

图 2-2 (b) 示出两个双字节单周期 (1× 机器周期) 指令的时序图。第一个时序图所示为第二个字节 (C1P1) 和下一个操作码 (C1P2) 从无等待状态的存储器中读取，指令在一个周期内完成；第二个时序图所示为每次访问 Flash 插入一个等待状态的情况（共插入两个等待状态），相同的指令需要三个机器周期（指令时间延长）完成。

图 2-2 (c) 示出两个单字节双周期 (2× 机器周期) 指令的时序图。第一个时序图所示为下一个操作码 (C2P2) 从无等待状态的存储器中读取，指令在两个周期内完成；第二个时序图所示为操作码从 Flash 中读取，插入一个等待状态，相同的指令需要三个机器周期（指令时间延长）完成。

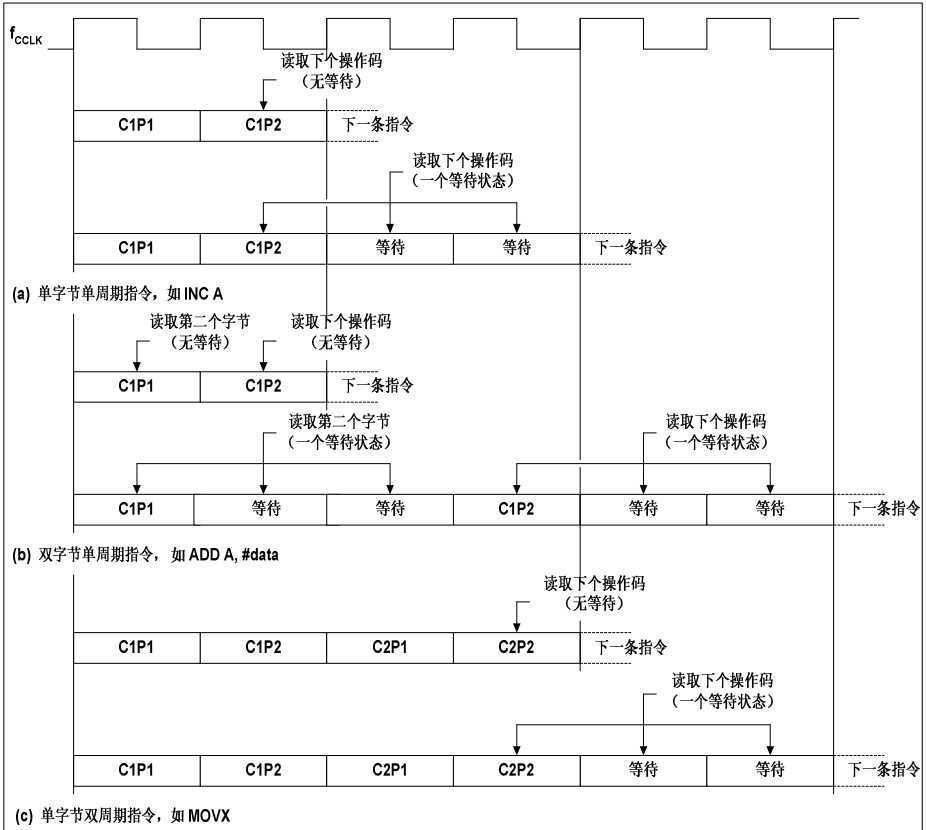


图 2-2 CPU 指令时序

指令长度为 1 个、2 个或 3 个字节，具体见表 2-1 “字节” 列。XC866 执行每条指令所需的时间包括：

- 译码/执行取回的操作码
- 取操作数（指令长度超过一个字节时）
- 预取下条指令的第一个字节（操作码）（XC866 CPU 流水线结构）

注：XC866 CPU 在执行当前指令时预取下条指令的操作码。

表 2-1 中列出每条指令所需的时钟周期数。第一个值对应从无等待状态的快速存储器（如 Boot ROM 和 RAM）中读取操作数和操作码的情况；第二个值对应从插入一个等待状态的慢速存储器（如 Flash）中读取操作数和操作码的情况。表格最后一列列出 8051 的指令时钟数，以便和 XC866 CPU 的性能进行比较。即使每次读取操作数和操作码都要插入一个等待状态，XC866 CPU 执行指令的速度仍然比 8051 处理器快 2 倍（如双字节单周期指令）到 6 倍（如单字节四周期指令）。

表 2-1 CPU 指令周期

助记符	十六进制码	字节数	f _{CLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
算术运算					
ADD A, Rn	28-2F	1	2	4	12
ADD A, dir	25	2	2	6	12
ADD A, @Ri	26-27	1	2	4	12
ADD A, #data	24	2	2	6	12
ADDC A, Rn	38-3F	1	2	4	12
ADDC A, dir	35	2	2	6	12
ADDC A, @Ri	36-37	1	2	4	12
ADDC A, #data	34	2	2	6	12
SUBB A, Rn	98-9F	1	2	4	12
SUBB A, dir	95	2	2	6	12

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CCLK} 时钟数		
			XC866		8051
			无等待状态	1个等待状态	
SUBB A, @Ri	96-97	1	2	4	12
SUBB A, #data	94	2	2	6	12
INC A	04	1	2	4	12
INC Rn	08-0F	1	2	4	12
INC dir	05	2	2	6	12
INC @Ri	06-07	1	2	4	12
DEC A	14	1	2	4	12
DEC Rn	18-1F	1	2	4	12
DEC dir	15	2	2	6	12
DEC @Ri	16-17	1	2	4	12
INC DPTR	A3	1	4	4	24
MUL AB	A4	1	8	8	48
DIV AB	84	1	8	8	48
DA A	D4	1	2	4	12
逻辑运算					
ANL A, Rn	58-5F	1	2	4	12
ANL A, dir	55	2	2	6	12
ANL A, @Ri	56-57	1	2	4	12
ANL A, #data	54	2	2	6	12
ANL dir, A	52	2	2	6	12
ANL dir, #data	53	3	4	10	24

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CCLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
ORL A, Rn	48-4F	1	2	4	12
ORL A, dir	45	2	2	6	12
ORL A, @Ri	46-47	1	2	4	12
ORL A, #data	44	2	2	6	12
ORL dir, A	42	2	2	6	12
ORL dir, #data	43	3	4	10	24
XRL A, Rn	68-6F	1	2	4	12
XRL A, dir	65	2	2	6	12
XRL A, @Ri	66-67	1	2	4	12
XRL A, #data	64	2	2	6	12
XRL dir, A	62	2	2	6	12
XRL dir, #data	63	3	4	10	24
CLR A	E4	1	2	4	12
CPL A	F4	1	2	4	12
SWAP A	C4	1	2	4	12
RL A	23	1	2	4	12
RLC A	33	1	2	4	12
RR A	03	1	2	4	12
RRC A	13	1	2	4	12
数据传送					
MOV A, Rn	E8-EF	1	2	4	12

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CCLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
MOV A, dir	E5	2	2	6	12
MOV A, @Ri	E6-E7	1	2	4	12
MOV A, #data	74	2	2	6	12
MOV Rn, A	F8-FF	1	2	4	12
MOV Rn, dir	A8-AF	2	4	8	24
MOV Rn, #data	78-7F	2	2	6	12
MOV dir, A	F5	2	2	6	12
MOV dir, Rn	88-8F	2	4	8	24
MOV dir, dir	85	3	4	10	24
MOV dir, @Ri	86-87	2	4	8	24
MOV dir, #data	75	3	4	10	24
MOV @Ri, A	F6-F7	1	2	4	12
MOV @Ri, dir	A6-A7	2	4	8	24
MOV @Ri, #data	76-77	2	2	6	12
MOV DPTR, #data	90	3	4	10	24
MOVC A, @A+DPTR	93	1	4	6	24
MOVC A, @A+PC	83	1	4	6	24
MOVX A, @Ri	E2-E3	1	4	6	24
MOVX A, @DPTR	E0	1	4	6	24
MOVX @Ri, A	F2-F3	1	4	6	24
MOVX @DPTR, A	F0	1	4	6	24

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CCLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
PUSH dir	C0	2	4	8	24
POP dir	D0	2	4	8	24
XCH A, Rn	C8-CF	1	2	4	12
XCH A, dir	C5	2	2	6	12
XCH A, @Ri	C6-C7	1	2	4	12
XCHA A, @Ri	D6-D7	1	2	4	12
位操作					
CLR C	C3	1	2	4	12
CLR bit	C2	2	2	6	12
SETB C	D3	1	2	4	12
SETB bit	D2	2	2	6	12
CPL C	B3	1	2	4	12
CPL bit	B2	2	2	6	12
ANL C, bit	82	2	4	8	24
ANL C, /bit	B0	2	4	8	24
ORL C, bit	72	2	4	8	24
ORL C, /bit	A0	2	4	8	24
MOV C, bit	A2	2	2	6	12
MOV bit, C	92	2	4	8	24
控制转移					
ACALL addr 11	11->F1	2	4	8	24

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
LCALL addr 16	12	3	4	10	24
RET	22	1	4	4	24
RETI	32	1	4	4	24
AJMP addr 11	01->E1	2	4	8	24
LJMP addr 16	02	3	4	10	24
SJMP rel	80	2	4	8	24
JC rel	40	2	4	8	24
JNC rel	50	2	4	8	24
JB bit, rel	20	3	4	10	24
JNB bit, rel	30	3	4	10	24
JBC bit, rel	10	3	4	10	24
JMP @A+DPTR	73	1	4	4	24
JZ rel	60	2	4	8	24
JNZ rel	70	2	4	8	24
CJNE A, dir, rel	B5	3	4	10	24
CJNE A, #d, rel	B4	3	4	10	24
CJNE Rn, #d, rel	B8-BF	3	4	10	24
CJNE @Ri, #d, rel	B6-B7	3	4	10	24
DJNZ Rn, rel	D8-DF	2	4	8	24
DJNZ dir, rel	D5	3	4	10	24

表格 2-1 CPU 指令周期 (续上页)

助记符	十六进制码	字节数	f _{CCLK} 时钟数		
			XC866		8051
			无等待状态	1 个等待状态	
其他指令					
NOP	00	1	2	4	12
额外指令					
MOVC @(DPTR++), A	A5	1	4	4	-
TRAP	A5	1	2	tbd	-

3 存储器结构

XC866 的 CPU 可寻址以下五个地址空间：

- 8 K 字节启动 ROM (Boot ROM) 程序存储器
- 256 字节内部 RAM 数据存储器
- 512 字节 XRAM 存储器
(XRAM 可作为程序存储器或外部数据存储器进行读写)
- 128 字节 SFR 区
- 8/16K 字节 Flash 程序存储器 (Flash 器件)；或
8/16K 字节 ROM 程序存储器，外加 4K 字节 Flash (ROM 器件)

图 3-1 所示为 16K 字节 Flash 器件的存储器地址空间分配。对于 8K 字节的 Flash 器件，不包括阴影标出的 P-Flash Bank。

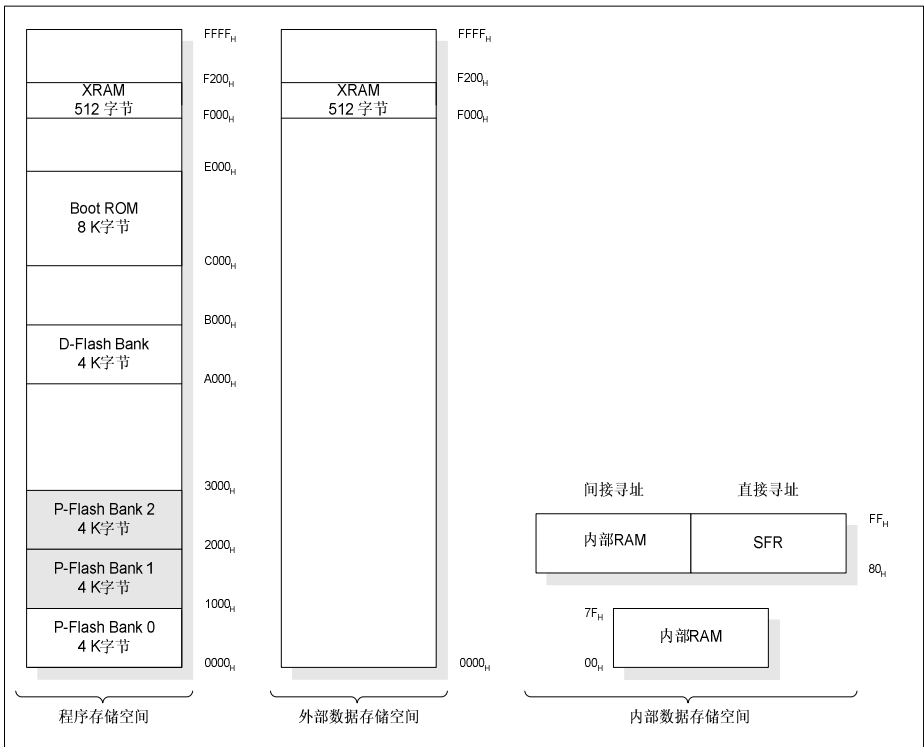


图 3-1 XC866 Flash 器件映射

图 3-2 所示为 16K 字节 ROM 器件的存储器地址空间分配。对于 8K 字节的 ROM 器件，不包括阴影标出的地址区域。

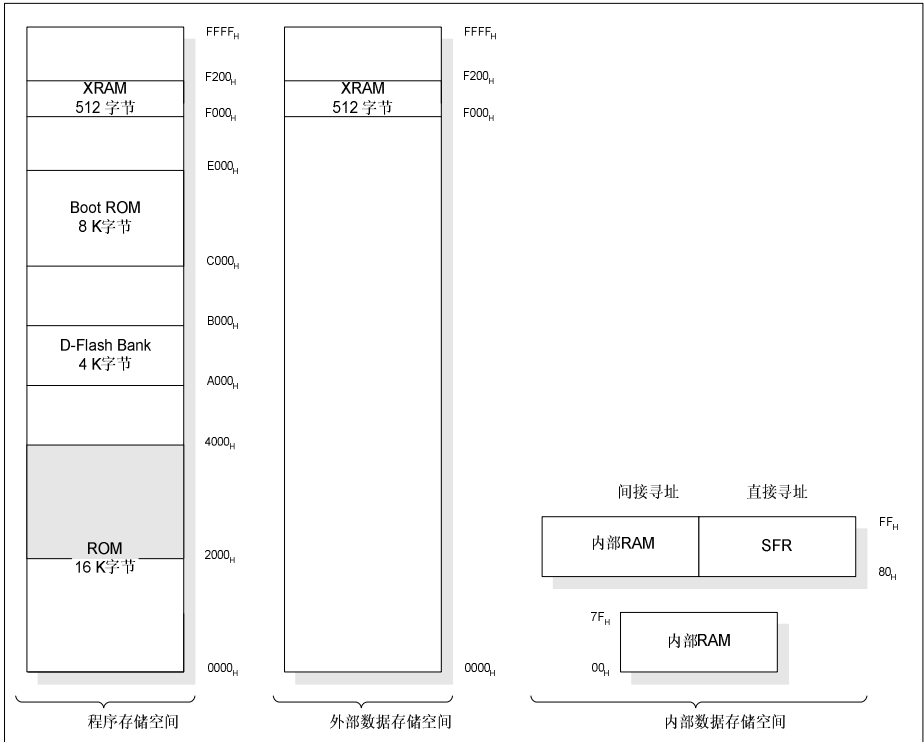


图 3-2 XC866 ROM 器件映射

3.1 程序存储器

使用一个专用接口将 CPU 和程序存储器直接相连（而不用任何引脚连接），CPU 的性能得到了优化。这意味着在每个时钟的上升沿均能存取代码。因此，这里不存在“内部”或“外部”程序存储器的概念，所有的代码均从同一个程序存储器接口存取。

3.2 数据存储器

数据存储器由内部和外部存储器组成。数据存储器的“内部”和“外部”标记用来区分寄存器存储器和使用“MOVX”指令存取数据的 64K 字节数据存储器。该标记并不是指外部数据存储器位于片外。

3.2.1 内部数据存储器

内部数据存储器被划分为物理上分开的两个不同区域：256 字节 RAM 和 128 字节 SFR。RAM 的高 128 字节和 SFR 区共用相同的地址段，通过不同的寻址方式访问。RAM 的低 128 字节可通过直接寻址或寄存器间接寻址方式访问；RAM 的高 128 字节只能通过寄存器间接寻址方式访问；SFR 只能通过直接寻址方式访问。

RAM 的地址段 20_{H} 到 $2F_{\text{H}}$ 的 16 个字节可位寻址；地址段 30_{H} 到 $7F_{\text{H}}$ 可用作暂存寄存器或堆栈。

3.2.2 外部数据存储器

512 字节的 XRAM 被映射为外部数据存储器区和程序存储器区。可用“MOVX”和“MOVC”指令访问。

3.3 特殊功能寄存器

特殊功能寄存器（SFR）占据内部数据存储器的地址段 80_H到 FF_H。除程序计数器之外的所有寄存器均位于该 SFR 区。这些 SFR 包括为 CPU 和片内外设提供接口的指针和寄存器。内部数据存储区中共有 128 个 SFR，小于所需的寄存器总数，从而需要采用地址扩展机制来增加可寻址 SFR 的数目。地址扩展机制包括：

- 映射
- 分页

3.3.1 映射地址扩展

在系统级通过映射进行地址扩展。SFR 区被扩展为两部分：标准（非映射）SFR 区和映射 SFR 区。两个 SFR 区占据相同的地址段 80_H到 FF_H，从而使可寻址的 SFR 个数扩展到 256 个。选择扩展地址区不由 CPU 指令直接控制，而是由位于地址 8F_H上的系统控制寄存器 SYSCON0 中的位 RMAP 来控制。置位 SYSCON0 中的 RMAP，控制访问映射 SFR 区时；对 RMAP 清零，控制访问标准 SFR 区。SFR 区的选择如图 3-3 所示。

SYSCON0

系统控制寄存器 0

复位值: 00_H

7	6	5	4	3	2	1	0
0							RMAP
r							rw

符号	位序号	读写类型	功能描述
RMAP	0	rw	SFR 映射控制 0 访问标准 SFR 区 1 访问映射 SFR 区
0	[7:1]	r	保留 读操作返回 0；应写入 0

只要 RMAP 被置位，即可访问映射 SFR 区。该位不可由硬件自动清零。因此，访问标准/映射寄存器前，必须由软件对位 RMAP 分别清零/置位。

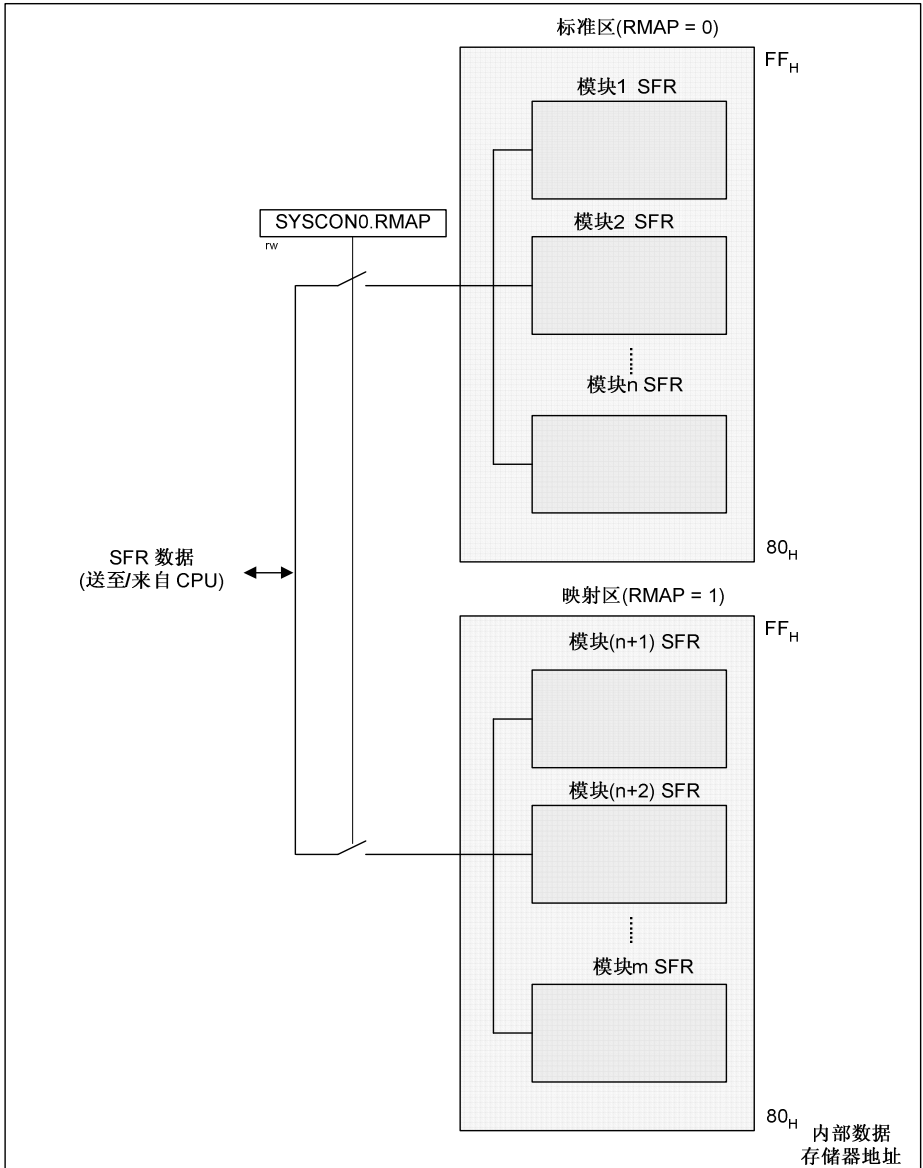


图 3-3 映射地址扩展

3.3.2 分页地址扩展

在模块级通过分页进一步扩展地址。映射地址扩展使 XC866 的 SFR 个数达到 256 个，但即使这样，SFR 的数目仍小于片内外设所需的 SFR 总数。为了满足 SFR 的数目要求，某些外设采用内嵌局部地址扩展机制，增加可寻址 SFR 的数目。选择扩展地址区不由 CPU 指令直接控制，而是由模块分页寄存器 MOD_PAGE 中的位域 PAGE 来控制。因此，在访问目标模块的 SFR 前，必须先设置位域 PAGE。根据具体要求，每个模块中可能包含的页数不同，每页上 SFR 的个数不同。除了正确设置 RMAP 值来选择 SFR 区之外，用户必须确保选择了有效的 PAGE 指向所要的 SFR。页的选择如图 3-4 所示。

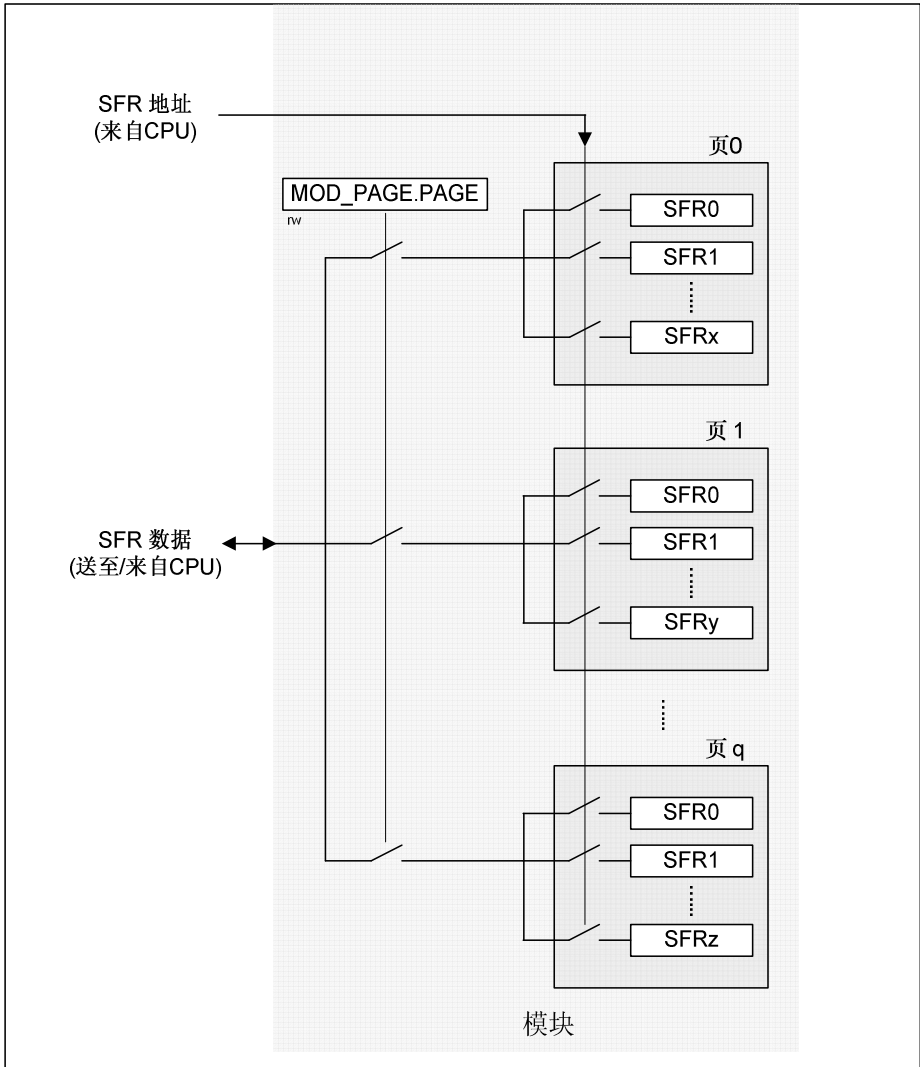


图 3-4 分页地址扩展

要访问位于（和当前页）不同页上的寄存器，必须离开当前页。需重新设置分页寄存器的位域 PAGE，只有这样才能访问到所要的 SFR。

如果在访问分页寄存器和模块寄存器之间开始执行某个中断服务程序，且中断需要访问位于另一页上的寄存器，保存当前页设置，然后设置新页，最后恢复原先页设置。可以用保存域 **STx** ($x = 0 - 3$) 来保存和恢复当前页的设置。同时指出应使用哪些保存域和新页值，单独的写操作即可完成：

- **PAGE** 中的内容在被新值覆盖之前保存在 **STx** 中
(在中断服务程序开始处保存当前页设置，并设置新页编号)；或
- 用 **STx** 中的内容覆盖 **PAGE** 的内容，对写入 **PAGE** 的值不予理睬
(在中断服务程序结束时恢复中断发生之前上一一次的页设置)

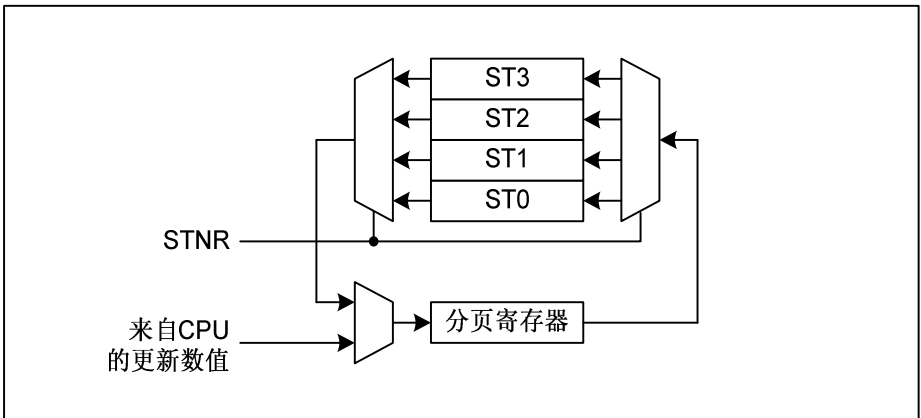


图 3-5 页保存

通过这种分页机制，中断服务程序（或其他程序）无需读出并保存上次使用的页信息即可改变页的设置。仅用写操作使系统更加简单和高速。该机制显著改善了小中断服务程序的性能。

XC866 的以下外设支持局部地址扩展：

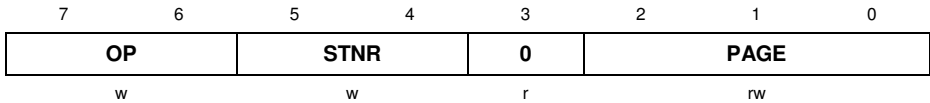
- 并行端口
- 模数转换单元（ADC）
- 捕获比较单元（CCU6）
- 系统控制寄存器

分页寄存器定义如下:

MOD_PAGE

模块 MOD 分页寄存器

复位值: 00H



符号	位序号	读写类型	功能描述
PAGE	[2:0]	rw	<p>分页位</p> <p>写入时, 该值表示新页的值</p> <p>读出时, 该值表示当前有效页的值</p>
STNR	[5:4]	w	<p>保存编号</p> <p>该编号指明在哪个保存位域上执行 OP 定义的操作。</p> <p>若 OP=10_B,</p> <p>PAGE 的内容在被新值覆盖之前保存在 STx 中</p> <p>若 OP=11_B,</p> <p>PAGE 的内容被 STx 覆盖。写入 PAGE 的值不予理睬</p> <p>00 选择 ST0</p> <p>01 选择 ST1</p> <p>10 选择 ST2</p> <p>11 选择 ST3</p>
OP	[7:6]	w	<p>操作</p> <p>0X 手动保存页模式, STNR 的值被忽略, PAGE 被直接写入</p> <p>10 带有自动页保存的新页设置。当前写入 PAGE 中的内容被保存的同时, 上次写入 PAGE 的内容被保存在 STNR 规定的位域 STx 中</p>

符号	位序号	读写类型	功能描述
			11 自动恢复页。对写入 PAGE 的内容不予理睬，PAGE 由 STNR 规定的位域 STx 中的内容覆盖
0	3	r	保留 读操作返回 0；应写入 0

3.3.3 位寻址

以格式 1XXXX000_B（如：80_H，88_H，90_H，...F0_H，F8_H）为地址的 SFR 均为位可寻址寄存器。这些位可寻址寄存器的地址在表 3-1 至表 3-9 中以黑体标出。

3.3.4 系统控制寄存器

系统控制 SFR 用来控制整个系统的功能，例如中断，可变波特率的产生，时钟管理，位保护方案，振荡器（OSC）和锁相环（PLL）控制。这些 SFR 位于标准存储器区（RMAP = 0），组织成两页。SCU_PAGE 寄存器位于地址 B2_H，包含分页值和页控制信息。

SCU_PAGE

系统控制分页寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
OP		STNR		0	PAGE		
w		w		r	rw		

符号	位序号	读写类型	功能描述
PAGE	[2:0]	rw	分页位 写入时，该值表示新页的值 读出时，该值表示当前有效页的值
STNR	[5:4]	w	保存编号 该编号指明在哪个保存位域上执行 OP 定义的操作。 若 OP=10 _B ， PAGE 的内容在被新值覆盖之前保存在 STx

符号	位序号	读写类型	功能描述
			<p>中</p> <p>若 OP=11_B,</p> <p>PAGE 的内容被 STx 覆盖。写入 PAGE 的值不予理睬</p> <p>00 选择 ST0</p> <p>01 选择 ST1</p> <p>10 选择 ST2</p> <p>11 选择 ST3</p>
OP	[7:6]	w	<p>操作</p> <p>0X 手动保存页模式，STNR 的值被忽略，PAGE 被直接写入</p> <p>10 带有自动页保存的新页设置。当前写入 PAGE 中的内容被保存的同时，上次写入 PAGE 的内容被保存在 STNR 规定的位域 STx 中</p> <p>11 自动恢复页。对写入 PAGE 的内容不予理睬，PAGE 由 STNR 规定的位域 STx 中的内容覆盖</p>
0	3	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

3.3.4.1 位保护方案

位保护方案通过 PASSWD 寄存器来阻止软件对选择位直接写入（即被保护位）。当 MODE 为 11_B，位 PASS 中写入 10011_B 会开放所有被保护位的访问权限；位 PASS 中写入 10101_B 会关闭所有被保护位的访问权限。注意如果未写入“关闭访问权限”口令，权限最多开放 32 个 CCLK 时钟周期。如果在 32 个 CCLK 时钟周期结束前再次写入“开放访问权限”口令，将重新计数 32 个 CCLK 周期。被保护位包括 NDIV，WDTEN，PD 和 SD。

PASSWD

口令寄存器

复位值: 07_H

	7	6	5	4	3	2	1	0
PASS						PROTE CT_S	MODE	
wh						rh	rw	

符号	位序号	读写类型	功能描述
MODE	[1:0]	rw	位保护方案控制位 00 保护方案禁止 11 保护方案使能（缺省值） 其它：保护方案使能 这两位不能直接写入。要在 11 _B 和 00 _B 之间改变值，必须在位域 PASS 中写入 11000 _B ，只有这样 MODE[1:0] 的值才能被写入。
PROTECT_S	2	rh	位保护信号状态位 该位表明保护状态。 0 软件能够写入所有被保护位。 1 软件不能写入任何被保护位。
PASS	[7:3]	wh	口令位 位保护方案只能识别三个序列。 11000 _B MODE 写入使能 10011 _B 开放所有被保护位的写权限 10101 _B 关闭所有被保护位的写权限

3.3.5 XC866 寄存器概览

XC866 的 SFR 按照功能单元分组。SFR 的内容（位）总结见 3.3.5.1 至 3.3.5.9。

注：位可寻址 SFR 的地址在表 3-1 至表 3-9 中以黑体标出。

3.3.5.1 CPU 寄存器

CPU 内核 SFR 可从标准存储器区和映射存储器区访问（RMAP = 0 或 1）。

表 3-1 CPU 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0 或 1										
81 _H	SP 复位值：07 _H 堆栈指针寄存器	位域	SP							
		类型	rw							
82 _H	DPL 复位值：00 _H 数据指针寄存器， 低位字节	位域	DPL 7	DPL 6	DPL 5	DPL 4	DPL 3	DPL 2	DPL 1	DPL 0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
83 _H	DPH 复位值：00 _H 数据指针寄存器， 高位字节	位域	DPH 7	DPH 6	DPH 5	DPH 4	DPH 3	DPH 2	DPH 1	DPH 0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
87 _H	PCON 复位值：00 _H 功率控制寄存器	位域	SMD	0			GF1	GF0	0	IDLE
		类型	rw	r			rw	rw	r	rw
88 _H	TCON 复位值：00 _H 定时器控制寄存器	位域	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		类型	rwh	rw	rwh	rw	rwh	rw	rwh	rw
89 _H	TMOD 复位值：00 _H 定时器模式寄存器	位域	GAT E1	0	T1M		GAT E0	0	T0M	
		类型	rw	r	rw		rw	r	rw	
8A _H	TLO 复位值：00 _H 定时器 0 寄存器， 低位字节	位域	VAL							
		类型	rwh							

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
8B _H	TL1 复位值: 00 _H 定时器 1 寄存器, 低位字节	位域	VAL							
		类型	rwh							
8C _H	TH0 复位值: 00 _H 定时器 0 寄存器, 高位字节	位域	VAL							
		类型	rwh							
8D _H	TH1 复位值: 00 _H 定时器 1 寄存器, 高位字节	位域	VAL							
		类型	rwh							
98 _H	SCON 复位值: 00 _H 串行通道控制寄存 器	位域	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
		类型	rw	rw	rw	rw	rw	rwh	rwh	rwh
99 _H	SBUF 复位值: 00 _H 串行数据缓冲寄存 器	位域	VAL							
		类型	rwh							
A2 _H	EO 复位值: 00 _H 扩展操作寄存器	位域	0			TRAP E N	0			DPS E L0
		类型	r			rw	r			rw
A8 _H	IEN0 复位值: 00 _H 中断使能寄存器 0	位域	EA	0	ET2	ES	ET1	EX1	ET0	EX0
		类型	rw	r	rw	rw	rw	rw	rw	rw
B8 _H	IP 复位值: 00 _H 中断优先级寄存器	位域	0		PT2	PS	PT1	PX1	PT0	PX0
		类型	r		rw	rw	rw	rw	rw	rw
B9 _H	IPH 复位值: 00 _H 中断优先级寄存 器, 高位字节	位域	0		PT2 H	PSH	PT1 H	PX1 H	PT0 H	PX0 H
		类型	r		rw	rw	rw	rw	rw	rw
D0 _H	PSW	位域	CY	AC	F0	RS1	RS0	OV	F1	P

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00 _H 程序状态字寄存器	类型	rw	rwh	rwh	rw	rw	rwh	rwh	rh
E0 _H	A 复位值: 00 _H 累加寄存器	位域	ACC 7	ACC 6	ACC 5	ACC 4	ACC 3	ACC 2	ACC 1	ACC 0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
E8 _H	IEN1 复位值: 00 _H 中断使能寄存器 1	位域	ECC IP3	ECC IP2	ECC IP1	ECC IP0	EXM	EX2	ESS C	EAD C
		类型	rw	rw	rw	rw	rw	rw	rw	rw
F0 _H	B 复位值: 00 _H B 寄存器	位域	B7	B6	B5	B4	B3	B2	B1	B0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
F8 _H	IP1 复位值: 00 _H 中断优先级寄存器 1	位域	PCC IP3	PCC IP2	PCC IP1	PCC IP0	PXM	PX2	PSS C	PAD C
		类型	rw	rw	rw	rw	rw	rw	rw	rw
F9 _H	IPH1 复位值: 00 _H 中断优先级寄存器 1, 高位字节	位域	PCC IP3H	PCC IP2H	PCC IP1H	PCC IP0H	PXM H	PX2 H	PSS CH	PAD CH
		类型	rw	rw	rw	rw	rw	rw	rw	rw

3.3.5.2 系统控制寄存器

系统控制 SFR 从标准存储器区访问 (RMAP = 0)。

表 3-2 系统控制寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0 或 1										
8F _H	SYSCON0 复位值: 00 _H 系统控制寄存器 0	位域	0							RMAP
		类型	r							rw
RMAP = 0										
BF _H	SCU_PAGE 复位值: 00 _H 系统控制分页寄存器	位域	OP		STNR		0	PAGE		
		类型	w	w		r	rw			
RMAP = 0, 页 0										
B3 _H	MODPISEL 复位值: 00 _H 外设输入选择寄存器	位域	0		JTAGTDIS	JTAGTCKS	0		EXINT0IS	URRIS
		类型	r		rw	rw	r		rw	rw
B4 _H	IRCON0 复位值: 00 _H 中断请求寄存器 0	位域	0	EXINT6	EXINT5	EXINT4	EXINT3	EXINT2	EXINT1	EXINT0
		类型	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
B5 _H	IRCON1 复位值: 00 _H 中断请求寄存器 1	位域	0			ADC SRC 1	ADC SRC 0	RIR	TIR	EIR
		类型	r			rwh	rwh	rwh	rwh	rwh
B7 _H	EXICON0 复位值: 00 _H 外部中断控制寄存器 0	位域	EXINT3		EXINT2		EXINT1		EXINT0	
		类型	rw		rw		rw		rw	
BA _H	EXICON1	位域	0		EXINT6		EXINT5		EXINT4	

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 外部中断控制寄存器 1	类型	r		rw		rw		rw	
BB _H	NMICON 复位值: 00_H NMI 控制寄存器	位域	0	NMI ECC	NMI VDD P	NMI VDD	NMI OCD S	NMI FLA SHTI MER	NMI PLL	NMI WDT
		类型	r	rw	rw	rw	rw	rw	rw	rw
BC _H	NMISR 复位值: 00_H NMI 状态寄存器	位域	0	FNM IEC C	FNM IVD DP	FNM IVD D	FNM IOC DS	FNM IFLA SHTI MER	FNM IPLL	FNM IWD T
		类型	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
BD _H	BCON 复位值: 00_H 波特率控制寄存器	位域	BGSEL		T2E XIS	BRE N	BRPRE			R
		类型	rw		rw	rw	rw			rw
BE _H	BG 复位值: 00_H 波特率定时器/重载寄存器	位域	BR_VALUE							
		类型	rw							

RMAP = 0, 页 1

B3 _H	ID 复位值: 01_H 标识寄存器	位域	PRODID					VERID			
		类型	r					r			
B4 _H	PMCON0 复位值: 00_H 功率模式控制寄存器 0	位域	0	WDT RST	WK RS	WKS EL	SD	PD	WS		
		类型	r	rwh	rwh	rw	rw	rwh	rw		
B5 _H	PMCON1 复位值: 00_H 功率模式控制寄存器 1	位域	0				T2_ DIS	CCU_ DIS	SSC_ DIS	ADC_ DIS	
		类型	r				rw	rw	rw	rw	
B6 _H	OSC_CON	位域	0			OSC PD	XPD	OSC SS	ORD RES	OSC R	

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0	
	复位值: 08_H OSC 控制寄存器	类型	r			rw	rw	rw	rwh	rw	
B7 _H	PLL_CON 复位值: 20_H PLL 控制寄存器	位域	NDIV				VCO BYP	OSC DIS C	RES LD	LOC K	
		类型	rw				rw	rw	rwh	rh	
BA _H	CMCON 复位值: 00_H 时钟控制寄存器	位域	0				CLKREL				
		类型	r				rw				
BB _H	PASSWD 复位值: 07_H 口令寄存器	位域	PASS					PRO TEC T_S	MODE		
		类型	wh					rh	rw		
BC _H	FEAL 复位值: 00_H FLASH 出错地址 寄存器, 低位字节	位域	ECCERRADDR[7:0]								
		类型	rh								
BD _H	FEAH 复位值: 00_H FLASH 出错地址 寄存器, 高位字节	位域	ECCERRADDR[15:8]								
		类型	rh								

3.3.5.3 WDT 寄存器

WDT SFR 从映射存储区访问 (RMAP = 1)。

表 3-3 WDT 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 1										
BB _H	WDTCON 复位值: 00 _H WDT 控制寄存器	位域	0	WIN BEN	WDT PR	0	WDT EN	WDT RS	WDT IN	
		类型	r	rw	rh	r	rw	rwh	rw	
BC _H	WDTREL 复位值: 00 _H WDT 重载寄存器	位域	WDTREL							
		类型	rw							
BD _H	WDTWINB 复位值: 00 _H 看门狗窗界计数寄存器	位域	WDTWINB							
		类型	rw							
BE _H	WDTL 复位值: 00 _H WDT 寄存器, 低位字节	位域	WDT[7:0]							
		类型	rh							
BF _H	WDTH 复位值: 00 _H WDT 寄存器, 高位字节	位域	WDT[15:8]							
		类型	rh							

3.3.5.4 端口寄存器

端口 SFR 从标准存储器区访问 (RMAP = 0)。

表 3-4 端口寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0										
B2 _H	PORT_PAGE 复位值: 00 _H 端口分页寄存器	位域	OP		STNR		0	PAGE		
		类型	w	w		r		rw		
RMAP = 0, 页 0										
80 _H	P0_DATA 复位值: 00 _H P0 口数据寄存器	位域	0	P5	P4	P3	P2	P1	P0	
		类型	r	rw	rw	rw	rw	rw	rw	rw
86 _H	P0_DIR 复位值: 00 _H P0 口方向寄存器	位域	0	P5	P4	P3	P2	P1	P0	
		类型	r	rw	rw	rw	rw	rw	rw	rw
90 _H	P1_DATA 复位值: 00 _H P1 口数据寄存器	位域	P7	P6	P5	0		P1		P0
		类型	rw	rw	rw	r		rw		rw
91 _H	P1_DIR 复位值: 00 _H P1 口方向寄存器	位域	P7	P6	P5	0		P1		P0
		类型	rw	rw	rw	r		rw		rw
A0 _H	P2_DATA 复位值: 00 _H P2 口数据寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
B0 _H	P3_DATA 复位值: 00 _H P3 口数据寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
B1 _H	P3_DIR 复位值: 00 _H P3 口方向寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
RMAP = 0, 页 1										
80 _H	P0_PUDSEL	位域	0	P5	P4	P3	P2	P1	P0	

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: FF_H P0 口上拉/下拉选择寄存器	类型	r		rw	rw	rw	rw	rw	rw
86 _H	P0_PUDEN 复位值: C4_H P0 口上拉/下拉使能寄存器	位域	0		P5	P4	P3	P2	P1	P0
		类型	r		rw	rw	rw	rw	rw	rw
90 _H	P1_PUDSEL 复位值: FF_H P1 口上拉/下拉选择寄存器	位域	P7	P6	P5	0			P1	P0
		类型	rw	rw	rw	r			rw	rw
91 _H	P1_PUDEN 复位值: FF_H P1 口上拉/下拉使能寄存器	位域	P7	P6	P5	0			P1	P0
		类型	rw	rw	rw	r			rw	rw
A0 _H	P2_PUDSEL 复位值: FF_H P2 口上拉/下拉选择寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
A1 _H	P2_PUDEN 复位值: FF_H P2 口上拉/下拉使能寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
B0 _H	P3_PUDSEL 复位值: BF_H P3 口上拉/下拉选择寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
B1 _H	P3_PUDEN 复位值: 40_H P3 口上拉/下拉使能寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
RMAP = 0, 页 2										
80 _H	P0_ALTSELO 复位值: 00_H P0 口其他功能选择寄存器 0	位域	0		P5	P4	P3	P2	P1	P0
		类型	r		rw	rw	rw	rw	rw	rw

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
86 _H	P0_ALTSEL1 复位值: 00 _H P0 口其他功能选择寄存器 1	位域	0		P5	P4	P3	P2	P1	P0
		类型	r		rw	rw	rw	rw	rw	rw
90 _H	P1_ALTSEL0 复位值: 00 _H P1 口其他功能选择寄存器 0	位域	P7	P6	P5	0			P1	P0
		类型	rw	rw	rw	r			rw	rw
91 _H	P1_ALTSEL1 复位值: 00 _H P1 口其他功能选择寄存器 1	位域	P7	P6	P5	0			P1	P0
		类型	rw	rw	rw	r			rw	rw
B0 _H	P3_ALTSEL0 复位值: 00 _H P3 口其他功能选择寄存器 0	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
B1 _H	P3_ALTSEL1 复位值: 00 _H P3 口其他功能选择寄存器 1	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw

RMAP = 0, 页 3

80 _H	P0_OD 复位值: FF _H P0 口开漏输出控制寄存器	位域	0		P5	P4	P3	P2	P1	P0
		类型	r		rw	rw	rw	rw	rw	rw
90 _H	P1_OD 复位值: FF _H P1 口开漏输出控制寄存器	位域	P7	P6	P5	0			P1	P0
		类型	rw	rw	rw	r			rw	rw
B0 _H	P3_OD 复位值: FF _H P3 口开漏输出控制寄存器	位域	P7	P6	P5	P4	P3	P2	P1	P0
		类型	rw	rw	rw	rw	rw	rw	rw	rw

3.3.5.5 ADC 寄存器

ADC SFR 从标准存储器区访问 (RMAP = 0)。

表 3-5 ADC 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0										
D1 _H	ADC_PAGE 复位值: 00 _H ADC 分页寄存器	位域	OP		STNR		0	PAGE		
		类型	w		w		r	rw		
RMAP = 0, 页 0										
CA _H	ADC_GLOBCTR 复位值: 00 _H 全局控制寄存器	位域	ANO N	DW	CTC			0		
		类型	rw	rw	rw					
CB _H	ADC_GLOBSTR 复位值: 00 _H 全局状态寄存器	位域	0		CHNR			0	SAM PLE	BUS Y
		类型	r		rh			r	rh	rh
CC _H	ADC_PRAR 复位值: 00 _H 优先级和仲裁寄存器	位域	ASE N1	ASE N0	0	ARB M	CSM 1	PRI O1	CSM 0	PRI O0
		类型	rw	rw	r	rw	rw	rw	rw	rw
CD _H	ADC_LCBR 复位值: B7 _H 边界检测寄存器	位域	BOUND1				BOUND0			
		类型	rw				rw			
CE _H	ADC_INPCR0 复位值: 00 _H 输入综合控制寄存器 0	位域	STC							
		类型	rw							
CF _H	ADC_ETRCR 复位值: 00 _H 外部触发控制寄存器	位域	SYN EN1	SYN EN0	ETRSEL1			ETRSEL0		
		类型	rw	rw	rw			rw		
RMAP = 0, 页 1										
CA _H	ADC_CHCTR0	位域	0	LCC			0	RESRSEL		

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 通道 0 控制寄存器	类型	r	rw				r	rw	
CB _H	ADC_CHCTR1 复位值: 00_H 通道 1 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
CC _H	ADC_CHCTR2 复位值: 00_H 通道 2 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
CD _H	ADC_CHCTR3 复位值: 00_H 通道 3 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
CE _H	ADC_CHCTR4 复位值: 00_H 通道 4 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
CF _H	ADC_CHCTR5 复位值: 00_H 通道 5 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
D2 _H	ADC_CHCTR6 复位值: 00_H 通道 6 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	
D3 _H	ADC_CHCTR7 复位值: 00_H 通道 7 控制寄存器	位域	0	LCC				0	RESRSEL	
		类型	r	rw				r	rw	

RMAP = 0, 页 2

CA _H	ADC_RESR0L 复位值: 00_H 结果寄存器 0, 低位字节	位域	RESULT[1:0]	0	VF	DRC	CHNR		
		类型	rh	r	rh	rh	rh		
CB _H	ADC_RESR0H 复位值: 00_H 结果寄存器 0, 高位字节	位域	RESULT[9:2]						
		类型	rh						
CC _H	ADC_RESR1L	位域	RESULT[1:0]	0	VF	DRC	CHNR		

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 结果寄存器 1, 低位字节	类型	rh		r	rh	rh	rh		
CD _H	ADC_RESR1H 复位值: 00_H 结果寄存器 1, 高位字节	位域	RESULT[9:2]							
		类型	rh							
CE _H	ADC_RESR2L 复位值: 00_H 结果寄存器 2, 低位字节	位域	RESULT[1:0]	0	VF	DRC	CHNR			
		类型	rh	r	rh	rh	rh			
CF _H	ADC_RESR2H 复位值: 00_H 结果寄存器 2, 高位字节	位域	RESULT[9:2]							
		类型	rh							
D2 _H	ADC_RESR3L 复位值: 00_H 结果寄存器 3, 低位字节	位域	RESULT[1:0]	0	VF	DRC	CHNR			
		类型	rh	r	rh	rh	rh			
D3 _H	ADC_RESR3H 复位值: 00_H 结果寄存器 3, 高位字节	位域	RESULT[9:2]							
		类型	rh							

RMAP = 0, 页 3

CA _H	ADC_RESRA0L 复位值: 00_H 结果寄存器 0 累加读取, 低位字节	位域	RESULT[2:0]	VF	DRC	CHNR				
		类型	rh	rh	rh	rh				
CB _H	ADC_RESRA0H 复位值: 00_H 结果寄存器 0 累加读取, 高位字节	位域	RESULT[10:3]							
		类型	rh							
CC _H	ADC_RESRA1L 复位值: 00_H 结果寄存器 1 累加读取, 低位字节	位域	RESULT[2:0]	VF	DRC	CHNR				
		类型	rh	rh	rh	rh				

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
CD _H	ADC_RESRA1H 复位值: 00 _H 结果寄存器 1 累加 读取, 高位字节	位域	RESULT[10:3]							
		类型	rh							
CE _H	ADC_RESRA2L 复位值: 00 _H 结果寄存器 2 累加 读取, 低位字节	位域	RESULT[2:0]		VF	DRC	CHNR			
		类型	rh		rh	rh	rh			
CF _H	ADC_RESRA2H 复位值: 00 _H 结果寄存器 2 累加 读取, 高位字节	位域	RESULT[10:3]							
		类型	rh							
D2 _H	ADC_RESRA3L 复位值: 00 _H 结果寄存器 3 累加 读取, 低位字节	位域	RESULT[2:0]		VF	DRC	CHNR			
		类型	rh		rh	rh	rh			
D3 _H	ADC_RESRA3H 复位值: 00 _H 结果寄存器 3 累加 读取, 高位字节	位域	RESULT[10:3]							
		类型	rh							

RMAP = 0, 页 4

CA _H	ADC_RCR0 复位值: 00 _H 结果控制寄存器 0	位域	VFC TR	WFR	FEN	IEN	0		DRC TR
		类型	rw	rw	rw	rw	r		rw
CB _H	ADC_RCR1 复位值: 00 _H 结果控制寄存器 1	位域	VFC TR	WFR	FEN	IEN	0		DRC TR
		类型	rw	rw	rw	rw	r		rw
CC _H	ADC_RCR2 复位值: 00 _H 结果控制寄存器 2	位域	VFC TR	WFR	FEN	IEN	0		DRC TR
		类型	rw	rw	rw	rw	r		rw
CD _H	ADC_RCR3 复位值: 00 _H 结果控制寄存器 3	位域	VFC TR	WFR	FEN	IEN	0		DRC TR
		类型	rw	rw	rw	rw	r		rw

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
CE _H	ADC_VFCR 复位值: 00 _H 有效标志清零寄存器	位域	0				VFC 3	VFC 2	VFC 1	VFC 0
		类型	r				w	w	w	w

RMAP = 0, 页 5

CA _H	ADC_CHINFR 复位值: 00 _H 通道中断标志寄存器	位域	CHI NF7	CHI NF6	CHI NF5	CHI NF4	CHI NF3	CHI NF2	CHI NF1	CHI NF0
		类型	rh	rh	rh	rh	rh	rh	rh	rh
CB _H	ADC_CHINCR 复位值: 00 _H 通道中断清零寄存器	位域	CHI NC7	CHI NC6	CHI NC5	CHI NC4	CHI NC3	CHI NC2	CHI NC1	CHI NC0
		类型	w	w	w	w	w	w	w	w
CC _H	ADC_CHINSR 复位值: 00 _H 通道中断置位寄存器	位域	CHI NS7	CHI NS6	CHI NS5	CHI NS4	CHI NS3	CHI NS2	CHI NS1	CHI NS0
		类型	w	w	w	w	w	w	w	w
CD _H	ADC_CHINPR 复位值: 00 _H 通道中断节点指针寄存器	位域	CHI NP7	CHI NP6	CHI NP5	CHI NP4	CHI NP3	CHI NP2	CHI NP1	CHI NP0
		类型	rw	rw	rw	rw	rw	rw	rw	rw
CE _H	ADC_EVINFR 复位值: 00 _H 事件中标志寄存器	位域	EVI NF7	EVI NF6	EVI NF5	EVI NF4	0		EVI NF1	EVI NF0
		类型	rh	rh	rh	rh	r		rh	rh
CF _H	ADC_EVINCR 复位值: 00 _H 事件中清零标志寄存器	位域	EVI NC7	EVI NC6	EVI NC5	EVI NC4	0		EVI NC1	EVI NC0
		类型	w	w	w	w	r		w	w
D2 _H	ADC_EVINSR 复位值: 00 _H 事件中置位标志寄存器	位域	EVI NS7	EVI NS6	EVI NS5	EVI NS4	0		EVI NS1	EVI NS0
		类型	w	w	w	w	r		w	w
D3 _H	ADC_EVINPR 复位值: 00 _H 事件中节点指针寄存器	位域	EVI NP7	EVI NP6	EVI NP5	EVI NP4	0		EVI NP1	EVI NP0
		类型	rw	rw	rw	rw	r		rw	rw

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0, 页 6										
CA _H	ADC_CRCR1 复位值: 00 _H 转换请求控制寄存器 1	位域	CH7	CH6	CH5	CH4	0			
		类型	rwh	rwh	rwh	rwh	r			
CB _H	ADC_CRPR1 复位值: 00 _H 转换请求挂起寄存器 1	位域	CHP7	CHP6	CHP5	CHP4	0			
		类型	rwh	rwh	rwh	rwh	r			
CC _H	ADC_CRMR1 复位值: 00 _H 转换请求模式寄存器 1	位域	0	LDEV	CLRPND	SCAN	ENSI	ENTR	ENGT	
		类型	r	w	w	rw	rw	rw	rw	
CD _H	ADC_QMR0 复位值: 00 _H 队列模式寄存器 0	位域	CEV	TREV	FLUSH	CLRV	TRMD	ENTR	ENGT	
		类型	w	w	w	w	rw	rw	rw	
CE _H	ADC_QSR0 复位值: 20 _H 队列状态寄存器 0	位域	0		EMPTY	EV	0			
		类型	r		rh	rh	r			
CF _H	ADC_QOR0 复位值: 00 _H 队列 0 寄存器 0	位域	EXTR	ENSI	RF	V	0	REQCHNR		
		类型	rh	rh	rh	rh	r	rh		
D2 _H	ADC_QBUR0 复位值: 00 _H 队列备份寄存器 0	位域	EXTR	ENSI	RF	V	0	REQCHNR		
		类型	rh	rh	rh	rh	r	rh		
D2 _H	ADC_QINR0 复位值: 00 _H 队列输入寄存器 0	位域	EXTR	ENSI	RF	0		REQCHNR		
		类型	w	w	w	r		w		

3.3.5.6 定时器 2 寄存器

定时器 2 SFR 从标准存储器区访问 (RMAP = 0)。

表 3-6 定时器 2 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
C0 _H	T2_T2CON 复位值: 00 _H 定时器 2 控制寄存器	位域	TF2	EXF2	0		EXEN2	TR2	0	CP/RL2
		类型	rwh	rwh	r		rw	rwh	r	rw
C1 _H	T2_T2MOD 复位值: 00 _H 定时器 2 模式寄存器	位域	0		EDGESEL	PREN	T2PRE		DCEN	
		类型	r		rw	rw	rw		rw	
C2 _H	T2_RC2L 复位值: 00 _H 定时器 2 重载/捕获寄存器, 低位字节	位域	RC2[7:0]							
		类型	rwh							
C3 _H	T2_RC2H 复位值: 00 _H 定时器 2 重载/捕获寄存器, 高位字节	位域	RC2[15:8]							
		类型	rwh							
C4 _H	T2_T2L 复位值: 00 _H 定时器 2 寄存器, 低位字节	位域	THL2[7:0]							
		类型	rwh							
C5 _H	T2_T2H 复位值: 00 _H 定时器 2 寄存器, 高位字节	位域	THL2[15:8]							
		类型	rwh							

3.3.5.7 CCU6 寄存器

CCU6 SFR 从标准存储器区访问 (RMAP = 0)。

表 3-7 CCU6 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 0										
A3 _H	CCU6_PAGE 复位值: 00 _H CCU6 分页寄存器	位域	OP		STNR		0		PAGE	
		类型	w		w		r		rw	
RMAP = 0, 页 0										
9A _H	CCU6_CC63SRL 复位值: 00 _H 通道 CC63 捕获/ 比较映射寄存器, 低位字节	位域	CC63SL							
		类型	rw							
9B _H	CCU6_CC63SRH 复位值: 00 _H 通道 CC63 捕获/ 比较映射寄存器, 高位字节	位域	CC63SH							
		类型	rw							
9C _H	CCU6_TCTR4L 复位值: 00 _H 定时器控制寄存器 4, 低位字节	位域	T12 STD	T12 STR	0		DTR ES	T12 RES	T12 RS	T12R R
		类型	w	w	r		w	w	w	w
9D _H	CCU6_TCTR4H 复位值: 00 _H 定时器控制寄存器 4, 高位字节	位域	T13 STD	T13 STR	0			T13 RES	T13 RS	T13R R
		类型	w	w	r			w	w	w
9E _H	CCU6_MCMOUT SL 复位值: 00 _H 多通道模式输出映 射寄存器, 低位字 节	位域	STR MC M	0		MCMPS				
		类型	w	r		rw				
9F _H	CCU6_MCMOUT SH	位域	STR HP	0		CURHS			EXPHS	

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 多通道模式输出映射寄存器, 高位字节	类型	w	r	rw			rw		
A4 _H	CCU6_ISRLL 复位值: 00_H 捕获/比较中断状态复位寄存器, 低位字节	位域	RT1 2PM	RT1 2OM	RCC 62F	RCC 62R	RCC 61F	RCC 61R	RCC 60F	RCC 60R
		类型	w	w	w	w	w	w	w	w
A5 _H	CCU6_ISRHH 复位值: 00_H 捕获/比较中断状态复位寄存器, 高位字节	位域	RST R	RIDL E	RW HE	RCH E	0	RTR PF	RT1 3PM	RT1 3CM
		类型	w	w	w	w	r	w	w	w
A6 _H	CCU6_CMPMODIFL 复位值: 00_H 比较状态修改寄存器, 低位字节	位域	0	MCC 63S	0			MCC 62S	MCC 61S	MCC 60S
		类型	r	w	r			w	w	w
A7 _H	CCU6_CMPMODIFH 复位值: 00_H 比较状态修改寄存器, 高位字节	位域	0	MCC 63R	0			MCC 62R	MCC 61R	MCC 60R
		类型	r	w	r			w	w	w
FA _H	CCU6_CC60SRL 复位值: 00_H 通道 CC60 捕获/比较映射寄存器, 低位字节	位域	CC60SL							
		类型	rwh							
FB _H	CCU6_CC60SRH 复位值: 00_H 通道 CC60 捕获/比较映射寄存器, 高位字节	位域	CC60SH							
		类型	rwh							
FC _H	CCU6_CC61SRL 复位值: 00_H 通道 CC61 捕获/比较映射寄存器, 低位字节	位域	CC61SL							
		类型	rwh							

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
FD _H	CCU6_CC61SRH 复位值: 00 _H 通道 CC61 捕获/ 比较映射寄存器, 高位字节	位域	CC61SH							
		类型	rwh							
FE _H	CCU6_CC62SRL 复位值: 00 _H 通道 CC62 捕获/ 比较映射寄存器, 低位字节	位域	CC62SL							
		类型	rwh							
FF _H	CCU6_CC62SRH 复位值: 00 _H 通道 CC62 捕获/ 比较映射寄存器, 高位字节	位域	CC62SH							
		类型	rwh							

RMAP = 0, 页 1

9A _H	CCU6_CC63RL 复位值: 00 _H 通道 CC63 捕获/ 比较寄存器, 低位 字节	位域	CC63VL							
		类型	rh							
9B _H	CCU6_CC63RH 复位值: 00 _H 通道 CC63 捕获/ 比较寄存器, 高位 字节	位域	CC63VH							
		类型	rh							
9C _H	CCU6_T12PRL 复位值: 00 _H 定时器 T12 周期 寄存器, 低位字节	位域	T12PVL							
		类型	rwh							
9D _H	CCU6_T12PRH 复位值: 00 _H 定时器 T12 周期 寄存器, 高位字节	位域	T12PVH							
		类型	rwh							
9E _H	CCU6_T13PRL 复位值: 00 _H 定时器 T13 周期 寄存器, 低位字节	位域	T13PVL							
		类型	rwh							

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0	
9F _H	CCU6_T13PRH 复位值: 00 _H 定时器 T13 周期寄存器, 高位字节	位域	T13PVH								
		类型	rwh								
A4 _H	CCU6_T12DTCL 复位值: 00 _H 定时器 T12 死区时间控制寄存器, 低位字节	位域	DTM								
		类型	rw								
A5 _H	CCU6_T12DTCH 复位值: 00 _H 定时器 T12 死区时间控制寄存器, 高位字节	位域	0	DTR2	DTR1	DTR0	0	DTE2	DTE1	DTE0	
		类型	r	rh	rh	rh	r	rw	rw	rw	
A6 _H	CCU6_TCTR0L 复位值: 00 _H 定时器控制寄存器 0, 低位字节	位域	CTM	CDIR	STE12	T12R	T12PRE	T12CLK			
		类型	rw	rh	rh	rh	rw	rw			
A7 _H	CCU6_TCTR0H 复位值: 00 _H 定时器控制寄存器 0, 高位字节	位域	0		STE13	T13R	T13PRE	T13CLK			
		类型	r		rh	rh	rw	rw			
FA _H	CCU6_CC60RL 复位值: 00 _H 通道 CC60 捕获/比较寄存器, 低位字节	位域	CC60VL								
		类型	rh								
FB _H	CCU6_CC60RH 复位值: 00 _H 通道 CC60 捕获/比较寄存器, 高位字节	位域	CC60VH								
		类型	rh								
FC _H	CCU6_CC61RL 复位值: 00 _H 通道 CC61 捕获/比较寄存器, 低位字节	位域	CC61VL								
		类型	rh								

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
FD _H	CCU6_CC61RH 复位值: 00 _H 通道 CC61 捕获/比较寄存器, 高位字节	位域	CC61VH							
		类型	rh							
FE _H	CCU6_CC62RL 复位值: 00 _H 通道 CC62 捕获/比较寄存器, 低位字节	位域	CC62VL							
		类型	rh							
FF _H	CCU6_CC62RH 复位值: 00 _H 通道 CC62 捕获/比较寄存器, 高位字节	位域	CC62VH							
		类型	rh							

RMAP = 0, 页 2

9A _H	CCU6_T12MSEL L 复位值: 00 _H T12 捕获/比较模式选择寄存器, 低位字节	位域	MSEL61				MSEL60			
		类型	rw				rw			
9B _H	CCU6_T12MSEL H 复位值: 00 _H T12 捕获/比较模式选择寄存器, 高位字节	位域	DBY P	HSYNC			MSEL62			
		类型	rw	rw			rw			
9C _H	CCU6_IENL 复位值: 00 _H 捕获/比较中断使能指针寄存器, 低位字节	位域	ENT 12P M	ENT 12O M	ENC C62 F	ENC C62 R	ENC C61 F	ENC C61 R	ENC C60 F	ENC C60 R
		类型	rw	rw	rw	rw	rw	rw	rw	rw
9D _H	CCU6_IENH 复位值: 00 _H 捕获/比较中断使能指针寄存器, 高位字节	位域	ENS TR	ENI DLE	EN WHE	ENC HE	0	ENT RPF	ENT 13P M	ENT 13C M
		类型	rw	rw	rw	rw	r	rw	rw	rw

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
9E _H	CCU6_INPL 复位值: 40_H 捕获/比较中断节点指针寄存器, 低位字节	位域	INPCHE		INPCC62		INPCC61		INPCC60	
		类型	rw		rw		rw		rw	
9F _H	CCU6_INPH 复位值: 39_H 捕获/比较中断节点指针寄存器, 高位字节	位域	0		INPT13		INPT12		INPERR	
		类型	r		rw		rw		rw	
A4 _H	CCU6_ISSL 复位值: 00_H 捕获/比较中断状态置位寄存器, 低位字节	位域	ST1 2PM	ST1 2OM	SCC 62F	SCC 62R	SCC 61F	SCC 61R	SCC 60F	SCC 60R
		类型	w	w	w	w	w	w	w	w
A5 _H	CCU6_ISSH 复位值: 00_H 捕获/比较中断状态置位寄存器, 高位字节	位域	SST R	SIDL E	SW HE	SCH E	SW HC	STR PF	ST1 3PM	ST1 3CM
		类型	w	w	w	w	w	w	w	w
A6 _H	CCU6_PSLR 复位值: 00_H 被动态电平寄存器	位域	PSL 63	0	PSL					
		类型	rwh	r	rwh					
A7 _H	CCU6_MCMCTR 复位值: 00_H 多通道模式控制寄存器	位域	0		SWSYN		0	SWSEL		
		类型	r		rw		r	rw		
FA _H	CCU6_TCTR2L 复位值: 00_H 定时器控制寄存器 2, 低位字节	位域	0	T13TED		T13TEC			T13 SSC	T12 SSC
		类型	r	rw		rw			rw	rw
FB _H	CCU6_TCTR2H 复位值: 00_H 定时器控制寄存器 2, 高位字节	位域	0				T13RSEL		T12RSEL	
		类型	r				rw		rw	
FC _H	CCU6_MODCTR L	位域	MC MEN	0	T12MODEN					

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0	
	复位值: 00_H 调制控制寄存器, 低位字节	类型	rw	r	rw						
FD _H	CCU6_MODCTR H 复位值: 00_H 调制控制寄存器, 高位字节	位域	ECT 13O	0	T13MODEN						
		类型	rw	r	rw						
FE _H	CCU6_TRPCTRL 复位值: 00_H 强制中断控制寄存器, 低位字节	位域	0				TRP M2	TRP M1	TRP M0		
		类型	r				rw	rw	rw		
FF _H	CCU6_TRPCTRH 复位值: 00_H 强制中断控制寄存器, 高位字节	位域	TRP PEN	TRP EN1 3	TRPEN						
		类型	rw	rw	rw						

RMAP = 0, 页 3

9A _H	CCU6_MCMOUT L 复位值: 00_H 多通道模式输出寄存器, 低位字节	位域	0	R	MCMCP					
		类型	r	rh	rh					
9B _H	CCU6_MCMOUT H 复位值: 00_H 多通道模式输出寄存器, 高位字节	位域	0		CURH			EXPH		
		类型	r		rh			rh		
9C _H	CCU6_ISL 复位值: 00_H 捕获比较中断状态寄存器, 低位字节	位域	T12 PM	T12 OM	ICC6 2F	ICC6 2R	ICC6 1F	ICC6 1R	ICC6 0F	ICC6 0R
		类型	rh	rh	rh	rh	rh	rh	rh	rh
9D _H	CCU6_ISH 复位值: 00_H 捕获比较中断状态寄存器, 高位字节	位域	STR	IDLE	WH E	CHE	TRP S	TRP F	T13 PM	T13 CM
		类型	rh	rh	rh	rh	rh	rh	rh	rh
9E _H	CCU6_PISEL0L	位域	ISTRP		ISCC62		ISCC61		ISCC60	

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 端口输入选择寄存器 0, 低位字节	类型	rw		rw		rw		rw	
9F _H	CCU6_PISEL0H 复位值: 00_H 端口输入选择寄存器 0, 高位字节	位域	IST12HR		ISPOS2		ISPOS1		ISPOS0	
		类型	rw		rw		rw		rw	
A4 _H	CCU6_PISEL2 复位值: 00_H 端口输入选择寄存器 2	位域	0						IST13HR	
		类型	r						rw	
FA _H	CCU6_T12L 复位值: 00_H 定时器 T12 计数寄存器, 低位字节	位域	T12CVL							
		类型	rwh							
FB _H	CCU6_T12H 复位值: 00_H 定时器 T12 计数寄存器, 高位字节	位域	T12CVH							
		类型	rwh							
FC _H	CCU6_T13L 复位值: 00_H 定时器 T13 计数寄存器, 低位字节	位域	T13CVL							
		类型	rwh							
FD _H	CCU6_T13H 复位值: 00_H 定时器 T13 计数寄存器, 高位字节	位域	T13CVH							
		类型	rwh							
FE _H	CCU6_CMPSTA TL 复位值: 00_H 比较状态寄存器, 低位字节	位域	0	CC6 3ST	CCP OS2	CCP OS1	CCP OS0	CC6 2ST	CC6 1ST	CC6 0ST
		类型	r	rh	rh	rh	rh	rh	rh	rh
FF _H	CCU6_CMPSTA TH 复位值: 00_H 比较状态寄存器, 高位字节	位域	T13I M	COU T63 PS	COU T62 PS	CC6 2PS	COU T61 PS	CC6 1PS	COU T60 PS	CC6 0PS
		类型	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

3.3.5.8 SSC 寄存器

SSC SFR 从标准存储器区访问 (RMAP = 0)。

表 3-8 SSC 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0	
RMAP = 0											
A9 _H	SSC_PISEL 复位值: 00 _H 端口输入选择寄存器	位域	0					CIS	SIS	MIS	
		类型	r					rw	rw	rw	
AA _H	SSC_CONL 复位值: 00 _H 控制寄存器, 低位字节 编程模式	位域	LB	PO	PH	HB	BM				
		类型	rw	rw	rw	rw	rw				
	工作模式	位域	0				BC				
		类型	r				rh				
AB _H	SSC_CONH 复位值: 00 _H 控制寄存器, 高位字节 编程模式	位域	EN	MS	0	AREN	BEN	PEN	REN	TEN	
		类型	rw	rw	r	rw	rw	rw	rw	rw	
	工作模式	位域	EN	MS	0	BSY	BE	PE	RE	TE	
		类型	rw	rw	r	rh	rwh	rwh	rwh	rwh	
AC _H	SSC_TBL 复位值: 00 _H 发送缓冲寄存器, 低位字节	位域	TB_VALUE								
		类型	rw								
AD _H	SSC_RBL	位域	RB_VALUE								

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
	复位值: 00_H 接收缓冲寄存器, 低位字节	类型	rh							
AE _H	SSC_BRL 复位值: 00_H 波特率定时器重载 寄存器, 低位字节	位域	BR_VALUE[7:0]							
		类型	rw							
AF _H	SSC_BRH 复位值: 00_H 波特率定时器重载 寄存器, 高位字节	位域	BR_VALUE[15:8]							
		类型	rw							

3.3.5.9 OCDS 寄存器

OCDS SFR 从映射存储区访问 (RMAP = 1)。

表 3-9 OCDS 寄存器概览

地址	寄存器名	位	7	6	5	4	3	2	1	0
RMAP = 1										
E9 _H	MMCR2 复位值: 00 _H 监控模式控制寄存器 2	位域	EXB C_P	EXB C	MBC ON_P	MBC ON	NME P_P	MM EP	MM ODE	JEN A
		类型	w	rw	w	rwh	w	rwh	rh	rh
F1 _H	MMCR 复位值: 00 _H 监控模式控制寄存器	位域	MEX IT_P	MEX IT	MST EP_P	MST EP	MRA MS_P	MRA M	TRF	RRF
		类型	w	hw	w	rw	w	rwh	rh	rh
F2 _H	MMSR 复位值: 00 _H 监控模式状态寄存器	位域	MBC AM	MBC IN	EXB F	SWB F	HW B3F	HW B2F	HW B1F	HW B0F
		类型	rw	rh	rwh	rwh	rwh	rwh	rwh	rwh
F3 _H	MMBPCR 复位值: 00 _H 断点控制寄存器	位域	SWB C	HWB3C		HWB2C		HW B1C	HWB0C	
		类型	rw	rw		rw		rw	rw	
F4 _H	MMICR 复位值: 00 _H 监控模式中中断控制寄存器	位域	DVE CT	DRE TR	0		MM UIE_P	MM UIE	RRI E_P	RRI E
		类型	rwh	rwh	r		w	rw	w	rw
F5 _H	MMDR 复位值: 00 _H 监控模式数据传输寄存器 接收	位域	MMRR							
		类型	rh							
	发送	位域	MMTR							

存储器结构

地址	寄存器名	位	7	6	5	4	3	2	1	0
		类型	w							
F6 _H	HWBPSR 复位值: 00 _H 硬件断点选择寄存器	位域	0			BPS EL_ P	BPSEL			
		类型	r			w	rw			
F7 _H	HWBPDR 复位值: 00 _H 硬件断点数据寄存器	位域	HWBP _{xxx}							
		类型	rw							

3.4 Boot ROM 工作模式

复位后 CPU 开始工作，始终先执行程序存储器地址段 0000_H-1FFF_H 的 Boot ROM 代码。Boot ROM 启动进程首先会把 Boot ROM 的地址空间切换到 C000_H-DFFF_H，如图 3-6 所示。而原先占据地址段 C000_H-DFFF_H 的程序存储器（Flash 或 ROM）将被映射到 0000_H-1FFF_H。地址空间交换之后，其余的 Boot ROM 启动进程将从 C00X_H继续执行。这包括检查引脚 MBC，TMS 和 P0.0 的锁存值以进入所选的 Boot ROM 工作模式。不同 Boot ROM 工作模式的选择请参照章节 7.2.3。本手册中给出的 XC866 存储器结构对应存储器地址交换之后执行不同工作模式时的情况。

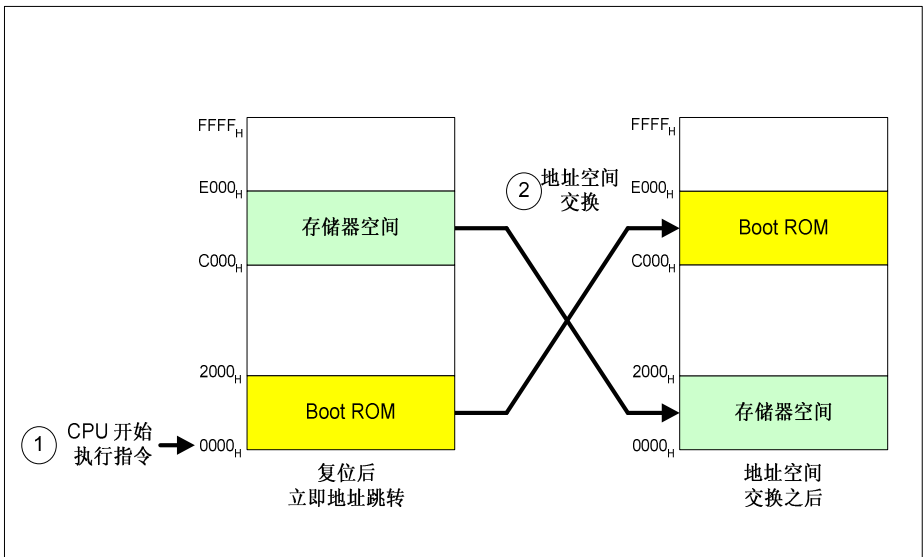


图 3-6 Boot ROM 地址空间交换

3.4.1 用户模式

如果 (MBC, TMS, P0.0) = (1, x, x)，Boot ROM 将跳转到程序存储器的地址 0000_H处执行 Flash 或 ROM 存储器中的用户代码。这是 XC866 的正常工作模式。

3.4.2 引导程序加载器模式

如果 (MBC, TMS, P0.0) = (0, 0, x)，将执行位于 Boot ROM 的引导程序加载器程序，允许对 XRAM 和 Flash 存储器（若片内包括该存储器）编程、擦除和执行。不同的 BSL 工作模式请参照章节 4.6。

3.4.3 OCDS 模式

如果 $(MBC, TMS, P0.0) = (0, 1, 1)$ ，将进入 OCDS 模式进行程序代码的调试。首先初始化 OCDS，接下来跳转到程序存储器的地址 0000_H 处，执行 Flash 或 ROM 存储器中的用户代码，开始进行调试。

在 OCDS 模式下，内部数据存储器的低 64 字节（地址 00_H - $3F_H$ ）可选择映射为 64 字节的监控 RAM 或内部数据 RAM。

4 闪存存储器

闪存（Flash）存储器提供了一种嵌入式、用户可编程的非易失性存储器，能够快速、可靠的存储用户代码和数据。Flash 的操作由嵌入式电压调节器（EVR）提供的 2.5V 电压供电，不需要额外的编程或擦除电压。Flash 存储器的分区特性使每个扇区可被独立擦除。

特性:

- 通过 UART 在系统编程（ISP）
- 在应用编程（IAP）
- 纠错码（ECC）可自动纠正一位错误
- 最小编程宽度为 32 字节
- 最小擦除宽度为 1 个扇区
- 每次读取 1 个字节
- 3×CCLK 周期的读取时间（包括一个等待状态）

4.1 Flash 存储器映射

XC866 系列产品提供了四种 Flash 器件，分别包含 8K 或 16K 字节的嵌入式 Flash。这些 Flash 存储器分别由两个或四个 4K 字节的 Flash bank 构成。每种 Flash 器件由具有不同分区的程序 Flash bank (P-Flash) 和数据 Flash bank (D-Flash) 构成。两种不同大小的 Flash 存储器映射如图 4-1 所示。

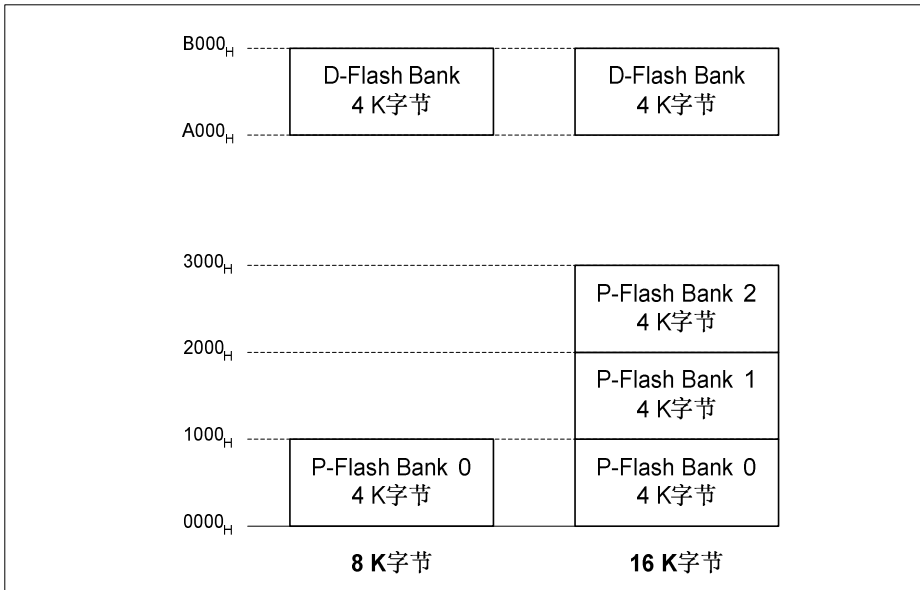


图 4-1 Flash 存储器映射

8K 字节的 Flash 器件，P-Flash bank 0 占据程序存储器的低位地址段，复位和中断向量指向从地址 0000_H 开始的地址空间。16K 字节的 Flash 器件，包括两个额外的 P-Flash bank (1 和 2) 用来存储用户代码：

- P-Flash bank 1 占据地址段 1000_H-1FFF_H
- P-Flash bank 2 占据地址段 2000_H-2FFF_H

XC866 系列产品中的所有器件 (包括 ROM 器件) 均有一个 4K 字节的 D-Flash bank，占据地址段 A000_H-AFFF_H。

4.2 Flash Bank 分区

XC866 的 Flash 器件由两种具有不同分区的 4K 字节的 Flash bank 构成，即程序 Flash bank（P-Flash）和数据 Flash bank（D-Flash），如图 4-2 所示。它们均可用于存储代码和数据。D-Flash 中的标记“D”并非指该 Flash 被映射为数据存储器，也并非指该 Flash 仅能用来存储数据。使用该标记旨在区分不同的 Flash bank 分区。

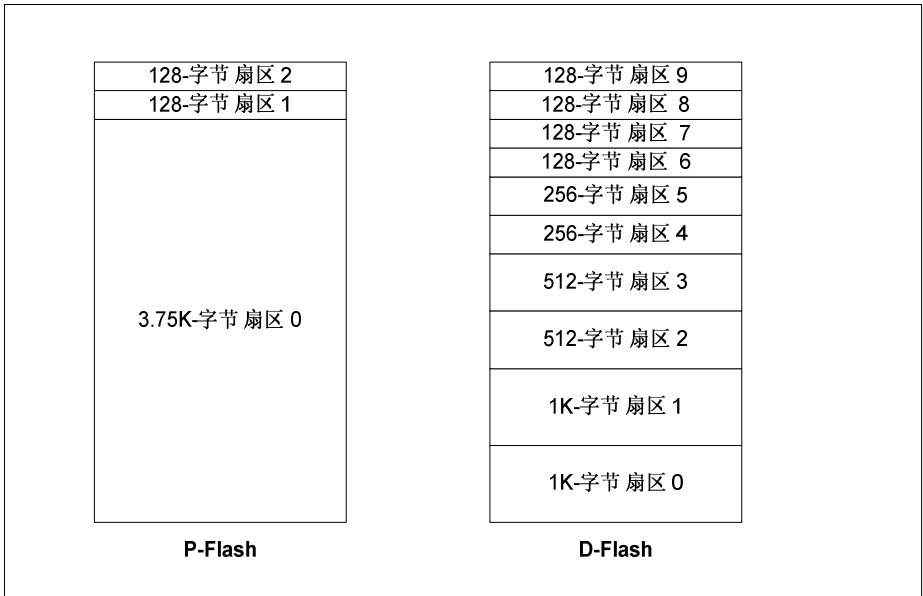


图 4-2 Flash Bank 分区

P-Flash 的扇区划分：

- 一个 3.75K 字节的扇区
- 两个 128 字节的扇区

D-Flash 的扇区划分：

- 两个 1K 字节的扇区
- 两个 512 字节的扇区
- 两个 256 字节的扇区
- 四个 128 字节的扇区

闪存存储器

每个 Flash bank 内部是具有灵活擦除能力的扇区结构。最小擦除宽度始终为一个完整的扇区。各个扇区可被分别擦除、或者同时擦除。和标准可擦除可编程 ROM (EPROM) 相反, Flash 存储器单元擦除后值全为 0。

D-Flash bank 被划分为更多的物理扇区, 从而具有扩展的擦除和重新编程能力。每种扇区的个数均为偶数, 从而具有更大的灵活性, 适用于广泛的应有领域。

举例说明, 用户编程可实现扇区的缓冲存储器机制。每个数据组双份复制保存在大小相同、分离的扇区中, 以保证当真正的数据组被破坏或删除时可使用其备份数据。

用户还可实现电可擦除可编程 ROM (EEROM) 的仿真算法, 把 D-Flash bank 当作环形堆栈存储器使用。最新值的更新始终设定在真正区域的顶部。当达到扇区顶部时, 所有真正的数据 (代表 EEPROM 的数据) 被复制到下一个扇区的底部, 上一个扇区被擦除。这种多方复制 EEPROM 的循环机制, 大大提高了存储器的耐受能力。为了加速数据查询, RAM 中存放指向有效数据组的指针。

4.3 字线地址

图 4-3 给出 P-Flash 和 D-Flash 的字线 (WL) 地址。

P-Flash 2				P-Flash 1				P-Flash 0				D-Flash			
字节 31	字节 2	字节 1	字节 0	字节 31	字节 2	字节 1	字节 0	字节 31	字节 2	字节 1	字节 0	字节 31	字节 2	字节 1	字节 0
2FFF _H	2FE2 _H	2FE1 _H	2FE0 _H	1FFF _H	1FE2 _H	1FE1 _H	1FE0 _H	0FFF _H	0FE2 _H	0FE1 _H	0FE0 _H	AFFF _H	AFE2 _H	AFE1 _H	AFE0 _H
2FF8 _H	2FE7 _H	2FE6 _H	2FE5 _H	1F9F _H	1F82 _H	1F81 _H	1F80 _H	0F9F _H	0F82 _H	0F81 _H	0F80 _H	AF9F _H	AF82 _H	AF81 _H	AF80 _H
2FF7 _H	2FE6 _H	2FE5 _H	2FE4 _H	1F7F _H	1F62 _H	1F61 _H	1F60 _H	0F7F _H	0F62 _H	0F61 _H	0F60 _H	AF7F _H	AF62 _H	AF61 _H	AF60 _H
2FF6 _H	2FE5 _H	2FE4 _H	2FE3 _H	1F1F _H	1F02 _H	1F01 _H	1F00 _H	0F1F _H	0F02 _H	0F01 _H	0F00 _H	AF1F _H	AF02 _H	AF01 _H	AF00 _H
2FF5 _H	2FE4 _H	2FE3 _H	2FE2 _H	1EFF _H	1EE2 _H	1EE1 _H	1EE0 _H	0EFF _H	0EE2 _H	0EE1 _H	0EE0 _H	AEFF _H	AEE2 _H	AEE1 _H	AEE0 _H
2FF4 _H	2FE3 _H	2FE2 _H	2FE1 _H	107F _H	1062 _H	1061 _H	1060 _H	007F _H	0062 _H	0061 _H	0060 _H	AE9F _H	AE82 _H	AE81 _H	AE80 _H
2FF3 _H	2FE2 _H	2FE1 _H	2FE0 _H	105F _H	1042 _H	1041 _H	1040 _H	005F _H	0042 _H	0041 _H	0040 _H	AE7F _H	AE62 _H	AE61 _H	AE60 _H
2FF2 _H	2FE1 _H	2FE0 _H	2FE0 _H	103F _H	1022 _H	1021 _H	1020 _H	103F _H	1022 _H	1021 _H	1020 _H	AE1F _H	AE02 _H	AE01 _H	AE00 _H
2FF1 _H	2FE0 _H	2FE0 _H	2FE0 _H	101F _H	1002 _H	1001 _H	1000 _H	101F _H	1002 _H	1001 _H	1000 _H	ADFF _H	ADE2 _H	ADE1 _H	ADE0 _H
2FF0 _H	2FE0 _H	2FE0 _H	2FE0 _H	007F _H	0062 _H	0061 _H	0060 _H	007F _H	0062 _H	0061 _H	0060 _H	AD1F _H	AD02 _H	AD01 _H	AD00 _H
2FEF _H	2FD2 _H	2FD1 _H	2FD0 _H	005F _H	0042 _H	0041 _H	0040 _H	005F _H	0042 _H	0041 _H	0040 _H	ACFF _H	ACE2 _H	ACE1 _H	ACE0 _H
2FE8 _H	2FD7 _H	2FD6 _H	2FD5 _H	003F _H	0022 _H	0021 _H	0020 _H	003F _H	0022 _H	0021 _H	0020 _H	AC1F _H	AC02 _H	AC01 _H	AC00 _H
2FE7 _H	2FD6 _H	2FD5 _H	2FD4 _H	001F _H	0002 _H	0001 _H	0000 _H	001F _H	0002 _H	0001 _H	0000 _H	ABFF _H	ABE2 _H	ABE1 _H	ABE0 _H
2FE6 _H	2FD5 _H	2FD4 _H	2FD3 _H	000F _H	0002 _H	0001 _H	0000 _H	000F _H	0002 _H	0001 _H	0000 _H	AA3F _H	AA22 _H	AA21 _H	AA20 _H
2FE5 _H	2FD4 _H	2FD3 _H	2FD2 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	AA1F _H	AA02 _H	AA01 _H	AA00 _H
2FE4 _H	2FD3 _H	2FD2 _H	2FD1 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A9FF _H	A9E2 _H	A9E1 _H	A9E0 _H
2FE3 _H	2FD2 _H	2FD1 _H	2FD0 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A83F _H	A822 _H	A821 _H	A820 _H
2FE2 _H	2FD1 _H	2FD0 _H	2FD0 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A81F _H	A802 _H	A801 _H	A800 _H
2FE1 _H	2FD0 _H	2FD0 _H	2FD0 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A7FF _H	A7E2 _H	A7E1 _H	A7E0 _H
2FE0 _H	2FD0 _H	2FD0 _H	2FD0 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A70F _H	A702 _H	A701 _H	A700 _H
2FDFF _H	2FDC _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A45F _H	A442 _H	A441 _H	A440 _H
2FD8 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A43F _H	A422 _H	A421 _H	A420 _H
2FD7 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A41F _H	A402 _H	A401 _H	A400 _H
2FD6 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A3FF _H	A3E2 _H	A3E1 _H	A3E0 _H
2FD5 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A05F _H	A042 _H	A041 _H	A040 _H
2FD4 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A03F _H	A022 _H	A021 _H	A020 _H
2FD3 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	A01F _H	A002 _H	A001 _H	A000 _H
2FD2 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H				
2FD1 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H				
2FD0 _H	2FDB _H	2FDB _H	2FDA _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H	0000 _H				

图 4-3 Flash 字线地址

4.4 工作模式

Flash bank 的工作模式如图 4-5 所示。

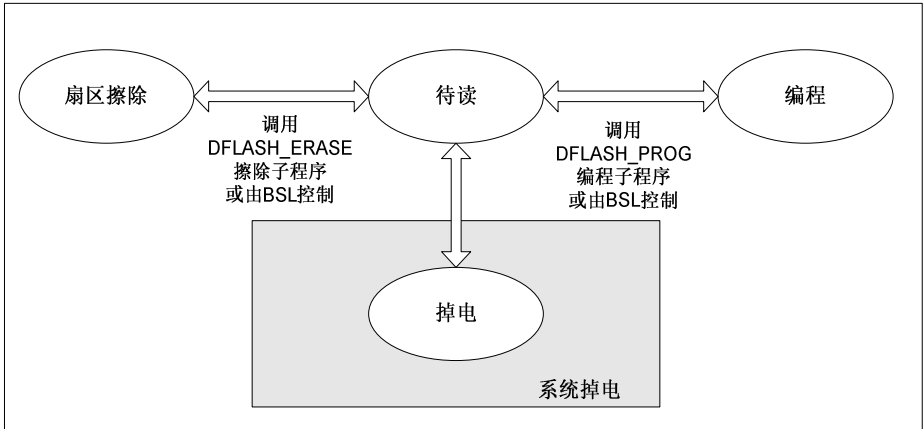


图 4-5 Flash 工作模式

通常，Flash 的工作模式由 BSL 和 D-Flash 编程或擦除子程序控制（见章节 4.7）。

Flash bank 在进入编程或擦除模式之前必须处在待读模式。待读模式下，Flash bank 的 32 字节写缓冲存储器可被写入，存储器单元的内容可由 CPU 读出。编程模式下，32 字节写缓冲存储器中的数据被编程到目标字线的 Flash 单元中。

Flash bank 的工作模式由状态机控制，以保证 Flash 模式的正确转移。这就避免了 Flash 的内容被无意破坏、并且有合理的较低软件开销。状态机还保证了 Flash bank 在编程或擦除过程中被封锁（不可读取）。但在读取一个 Flash bank 中的内容的同时，可对另一个 Flash bank 编程或擦除。

当用户设置 $PMCON0.PD = 1$ 进入系统掉电模式时，Flash bank 将由硬件自动带入掉电模式。系统掉电唤醒后，Flash bank 被带入待读模式，允许 CPU 访问。

4.5 检错和纠错

CPU 送出的 8 位数据被存储到 Flash 存储器中之前，采用纠错码（ECC）编码。读取时，从 Flash 中取出数据并译码，从而动态的检错和纠错。

纠错算法（Hamming 码）能够：

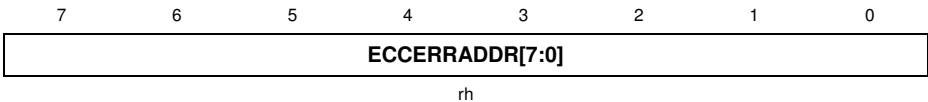
- 检查并纠正所有的 1 位错误
- 检错所有的 2 位错误，但无法纠正

可纠正的 1 位错误（结果有效）和不可纠正的 2 位错误（结果无效）不可区分，两种情况均产生 ECC 非屏蔽中断（NMI）。ECC 出错的 16 位 Flash 地址被保存在系统控制寄存器 FEAL 和 FEAH 中，中断服务程序访问该寄存器，从而确定错误出现在哪个 Flash bank 或 Flash 扇区。

FEAL

Flash 出错地址寄存器, 低位字节

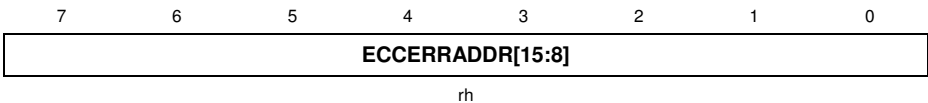
复位值: 00_H



FEAH

Flash 出错地址寄存器, 高位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
ECCERRADDR	FEAH 的[7:0], FEAH 的[7:0]		ECC 出错地址

4.6 在系统编程

引导程序加载器（BSL）支持 Flash 在系统编程（ISP），用户可对装在教育板上的空白微控制器进行编程；对以前已编程的微控制器，无需从教育板上卸下即可直接擦除并重新编程。该特性使嵌入式设计易于使用且功能多样。

微控制器的串行接口（UART）支持 ISP，通过普通的 RS-232 串行电缆和 PC 主机相连。上电复位或硬件复位后，若 MBC 和 TMS 引脚的锁存值均为 0，选择进入 BSL 模式。BSL 程序首先和串行通信端（PC 主机）的传输速率（波特率）自动同步。BSL 和主机之间的通信通过简单的传输协议完成。主机按照规定的块结构向微控制器按块发送信息，BSL 程序返回一个响应或出错字节以响应接收到的数据。用户可编程、擦除或执行 P- Flash 和/或 D- Flash bank 的内容。

已有的工作模式包括：

- 将用户程序从主机传送到 XRAM 和/或 Flash 中
- 执行 XRAM 中的用户程序
- 执行 Flash 中的用户程序
- 从相同或不同的 Flash bank 中擦除 Flash 扇区

4.7 在应用编程

多数应用中，D-Flash 中的数据在程序执行过程中需要修改。系统支持在应用编程，利用 Flash 定时器 NMI 调用一些特殊子程序，用户可以对 Flash 用户程序中的 D-Flash 数据编程或擦除。因此，有必要将 Flash 定时器 NMI 中断服务程序合并到 Flash 用户程序中；Flash 定时器 NMI 中断服务程序要作为 D-Flash 编程和擦除序列的一部分。

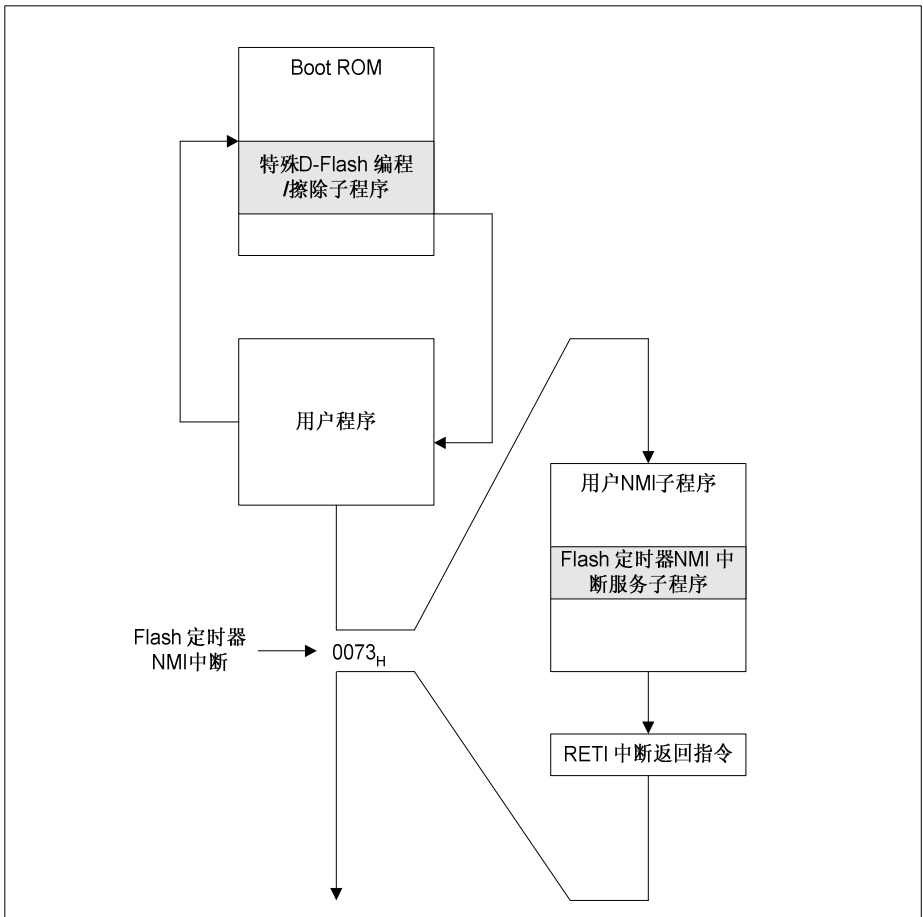


图 4-6 D-Flash 编程/ 擦除流程

4.7.1 D-Flash 编程

用户可调用 Flash 编程子程序把 32 个数据字节编程到一条 D-Flash 字线上。该子程序开始阶段，Flash 定时器 NMI 被使能，在整个编程过程中每次定时器一下溢就转入 Flash 定时器的 NMI 中断服务程序。

调用该子程序前，用户必须保证 32 字节的字线内容已逐一保存在 IRAM 的寄存器 bank 3 从 R0 开始的地址段中。另外，输入 DPTR0 (EO.DPSEL0=0) 必须保存着有效的 D-Flash 字线地址。否则，PSW.CY 将置位，不执行编程。如果调用子程序之前输入数据已有效，系统将继续初始化编程序列（包括把 32 字节的 IRAM 数据传输到 D-Flash 写缓冲存储器中），退出子程序，返回用户程序。用户程序代码从上次中断的地方继续执行，直到下次 Flash 定时器 NMI 被触发，进入 Flash 定时器 NMI 中断服务程序（见图 4-6）。Flash 定时器 NMI 中断服务程序将首先检查 Flash 定时器 NMI 的状态位 (NMISR.FNMIFLASHTIMER=1)，在执行其余的中断服务程序之前确保 NMI 的请求源来自 Flash 定时器。

表 4-1 D-Flash 编程子程序

子程序	DFF6H: DFLASH_PROG
输入	DPTR0: D-Flash 字线地址
	寄存器 bank 3 中的 R0 (IRAM 地址:18H) : 32 字节 D-Flash 数据的 IRAM 起始地址
	32 字节 D-Flash 数据
输出	PSW.CY 0 = 正在进行 D-FLASH 编程 1 = DPTR0 未指向有效的 D-FLASH 字线地址
	DPTR0 加 20H
	Flash 定时器 NMI 被使能 (NMICON.NMIFLASHTIMER=1)
所需堆栈大小	8
已使用/破坏的资源	PSW.CY, A, SCU_PAGE, DPTR1
保留资源 ¹⁾	寄存器 bank 3 中的 R0-R7 (IRAM 地址:18H - 1FH)
	IRAM 地址 36H - 3DH
消耗的机器周期 ²⁾	904

闪存存储器

¹⁾ 保留资源中的数据在编程过程中一定不能修改（包括处理 Flash 定时器 NMI 中断）以保证正确的编程流程。

²⁾ 无等待状态的估算值。

D-Flash 编程结束后，编程子程序将禁止 Flash 定时器 NMI 中断（`NMICON.NMIFLASHTIMER = 0`）。

为了表明“D-Flash 编程结束”，用户可检查：

- 位 `NMICON.NMIFLASHTIMER` 被清零，或
- 寄存器 bank 3 中 R3（IRAM 地址 `1BH`）的值为 `03H`

用户有必要使用 `MOVC` 指令读取 D-Flash 的数据，人为检查 D-Flash 的数据以确保编程成功。

4.7.2 D-Flash 擦除

用户可调用 Flash 擦除子程序擦除 D-Flash bank 的扇区。每次擦除可选择擦除 1 个扇区、多个扇区或所有 10 个扇区。该子程序开始阶段，Flash 定时器 NMI 被使能，在整个擦除过程中每次定时器一下溢就转入 Flash 定时器的 NMI 中断服务程序。

调用该子程序前，用户必须保证寄存器 bank 3 中的 R3 和 R4 被置位。系统首先初始化擦除序列，退出子程序，然后返回用户程序。用户程序代码从上次中断的地方继续执行，直到下次 Flash 定时器 NMI 被触发，进入 Flash 定时器 NMI 中断服务程序（见图 4-6）。Flash 定时器 NMI 中断服务程序将首先检查 Flash 定时器 NMI 的状态位（NMISR.FNMIFLASHTIMER=1），在执行其余的中断服务程序之前确保 NMI 的请求来源于 Flash 定时器。

表 4-2 D-Flash 擦除子程序

子程序	DFF9 _H : DFLASH_ERASE
输入	寄存器 bank 3 中的 R3 (IRAM 地址 1B _H) 选择 D-Flash bank 的擦除扇区 最低有效位 (LSB) 代表扇区 0, 最高有效位 (MSB) 代表扇区 7
	寄存器 bank 3 中的 R4 (IRAM 地址 1C _H) 选择 D-Flash bank 的擦除扇区 最低有效位 (LSB) 代表扇区 8, 最高有效位 (MSB) 代表扇区 9
输出	Flash 定时器 NMI 被使能 (NMICON.NMIFLASHTIMER=1)
所需堆栈大小	8
已使用/破坏的资源	PSW.CY, A, SCU_PAGE, DPTR1
保留资源 ¹⁾	寄存器 bank 3 中的 R0-R7 (IRAM 地址 18 _H -1F _H)
	IRAM 地址 36 _H - 3D _H
花费的机器周期 ²⁾	358

¹⁾ 保留资源中的数据在擦除过程中一定不能修改（包括执行 FLASH 定时器 NMI 中断服务过程）以保证正确的擦除流程。

²⁾ 无等待状态的估算值。

D-Flash 擦除结束后，擦除子程序将禁止 Flash 定时器 NMI 中断 (NMICON.NMIFLASHTIMER = 0)。

为了表明“D-Flash 擦除结束”，用户可检查：

- 位 NMICON.NMIFLASHTIMER 被清零，或
- 寄存器 bank 3 中 R3 (IRAM 地址 1BH) 的值为 03H

用户有必要用 MOVC 指令检查 D-Flash 的数据以确保擦除成功。

5 中断系统

XC866 内核支持一个非屏蔽中断（NMI）和 14 个可屏蔽中断。除了支持标准中断功能（例如，可配置的中断优先级和中断屏蔽功能）之外，XC866 中断系统还提供了扩展中断功能，例如将每个中断矢量分配给多个中断源，从而增加了所支持的中断源数目；附加的状态寄存器用来检测和确定中断源。

5.1 非屏蔽中断

NMI 和常规中断类似，区别仅在于 NMI 具有最高优先级（和其他中断相比）来处理重要的系统事件。XC866 系统中，下列七种事件可分别产生 NMI：

- WDT 已发生预报警
- PLL 与外部晶振失锁
- Flash 定时器已溢出
- 监控模式下产生 JTAG 接收请求或用户中断请求
- V_{DD} 低于预报警电压（2.3V）
- V_{DDP} 低于预报警电压（4.0V，对应外部电压为 5.0V 的情况）
- Flash 纠错码出错

NMISR 寄存器用来保存这些事件的 NMI 中断请求标志。NMICON 寄存器中的相应位决定 NMI 请求是否被响应或忽略。当任何被使能的 NMI 请求被响应和服务时，软件可把 NMISR 寄存器中的 NMI 请求标志清零。

5.2 可屏蔽中断

所有常规中断被称为可屏蔽中断。可屏蔽中断可以被屏蔽，或在处理器处理完当前任务之后再被响应。可屏蔽中断可分为三类：内部中断、外部中断和扩展中断。

5.2.1 内部中断

系统共有 3 个内部中断，分别来自于定时器 0、定时器 1 和 UART。这些中断请求信号直接送至 XC800 内核，中断状态由内核控制。

一旦定时器 0 和定时器 1 溢出，寄存器 TCON 的中断标志 TF0 和 TF1 即被置位。响应中断进入相应的中断服务程序后，TF0 和 TF1 由硬件自动清零。

UART 的中断由两个串口中断的逻辑或产生。当完成一帧数据的接收或发送时，寄存器 SCON 中的中断标志 RI 和 TI 自动置位，这两个标志位必须由软件清零。

5.2.2 外部中断

7 个外部中断 EXT_INT [6:0] 从端口进入 XC866，它们可由正沿、负沿或正负双沿触发。寄存器 EXICON0 和 EXICON1 设置触发外部中断的有效沿。

外部中断 0 和外部中断 1 也可选择不用边沿检测触发。发送至 CPU 的中断请求信号（由边沿触发/不用边沿触发）可进一步设置为由电平触发或负跳变触发，通过置位或清零寄存器 TCON 中的 IT_x（x=0 或 1）实现。

如果外部中断为正沿（负沿）触发，外部中断引脚必须保持至少 1 个 CCLK 的低（高）电平，接着 1 个 CCLK 的高（低）电平，从而保证电平跳变能被采样到。如果外部中断 0 和 1 的边沿检测被旁路，外部中断引脚必须保持至少两个 CCLK 周期的“高”或“低”电平。

5.2.3 扩展中断

扩展中断主要由片内外设产生。系统共有 9 个扩展中断请求信号，XINTR_SRC [13:5]，它们向 CPU 发送中断请求，而后会接收到来自 CPU 相应的响应信号 XINTR_ACK [13:5]。

某些中断源的中断请求标志存放在各自的 SFR 中（如 TCON，T2CON，SCON）。其他扩展中断请求标志和外部中断请求标志存放在寄存器 IRCON0 和 IRCON1 中。

由于外设/器件产生的中断请求输入多于 CPU 所支持的数目，一些中断必须复用、共用一个输入向 CPU 发送中断。少数关键外设中断（如，定时器，CCU6）直接和 CPU 的中断输入相连。

每一个由相应中断标志申请的中断输入都可以由 SFR IEN0 和 IEN1 中的使能位分别使能或禁止。此外，IEN0 中还有一个全局使能位 EA，清零后将禁止所有的中断。

图 5-1 至图 5-5 给出中断请求源的概况，并标出中断请求标志和控制标志。

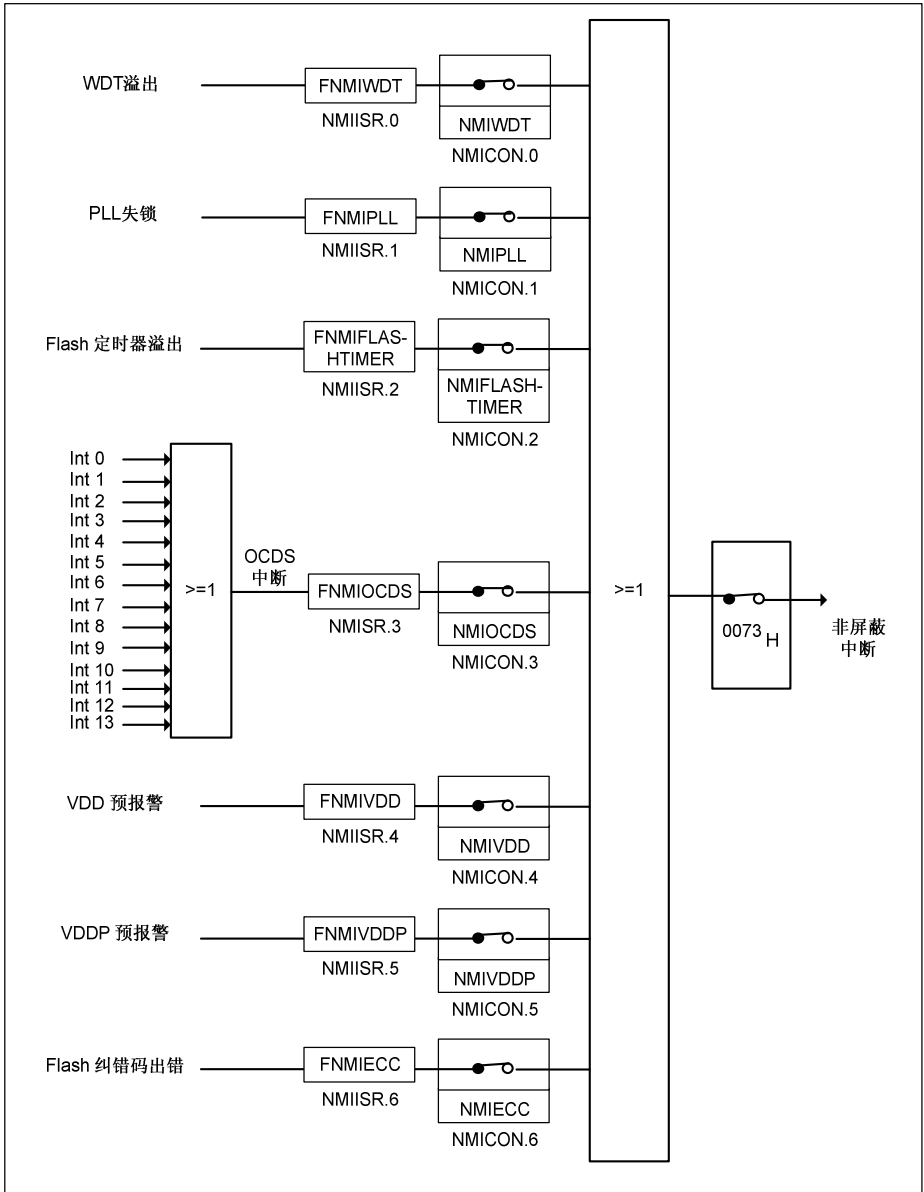


图 5-1 非屏蔽中断请求源

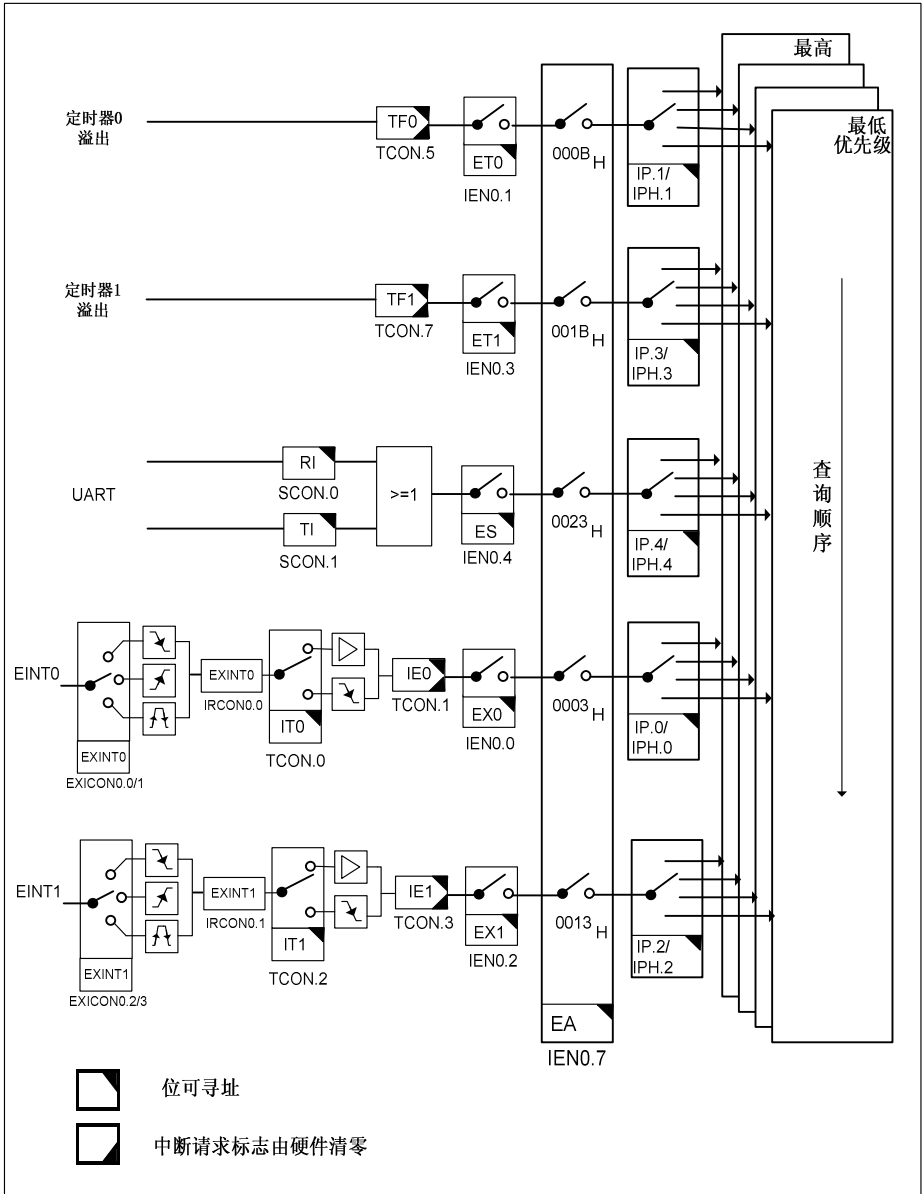


图 5-2 中断请求源 (第一部分)

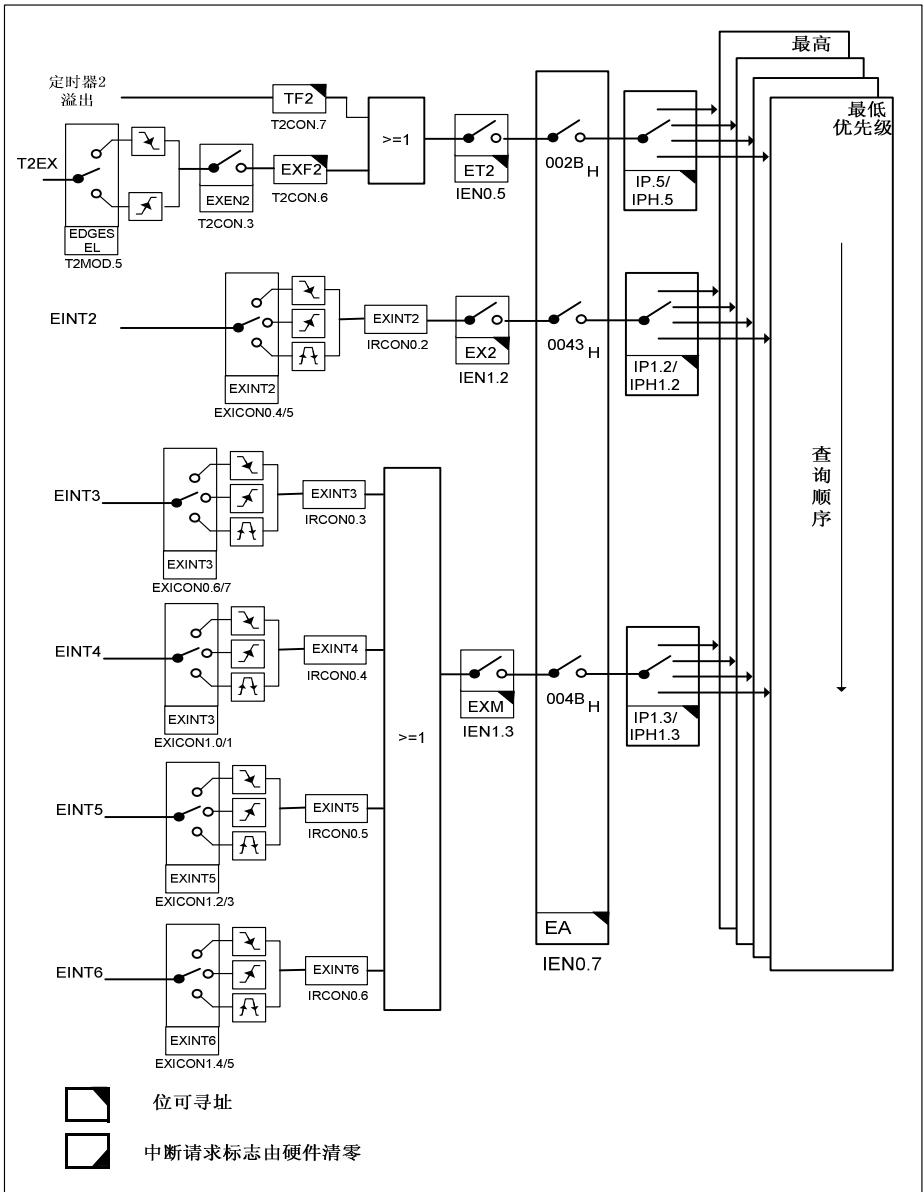


图 5-3 中断请求源 (第二部分)

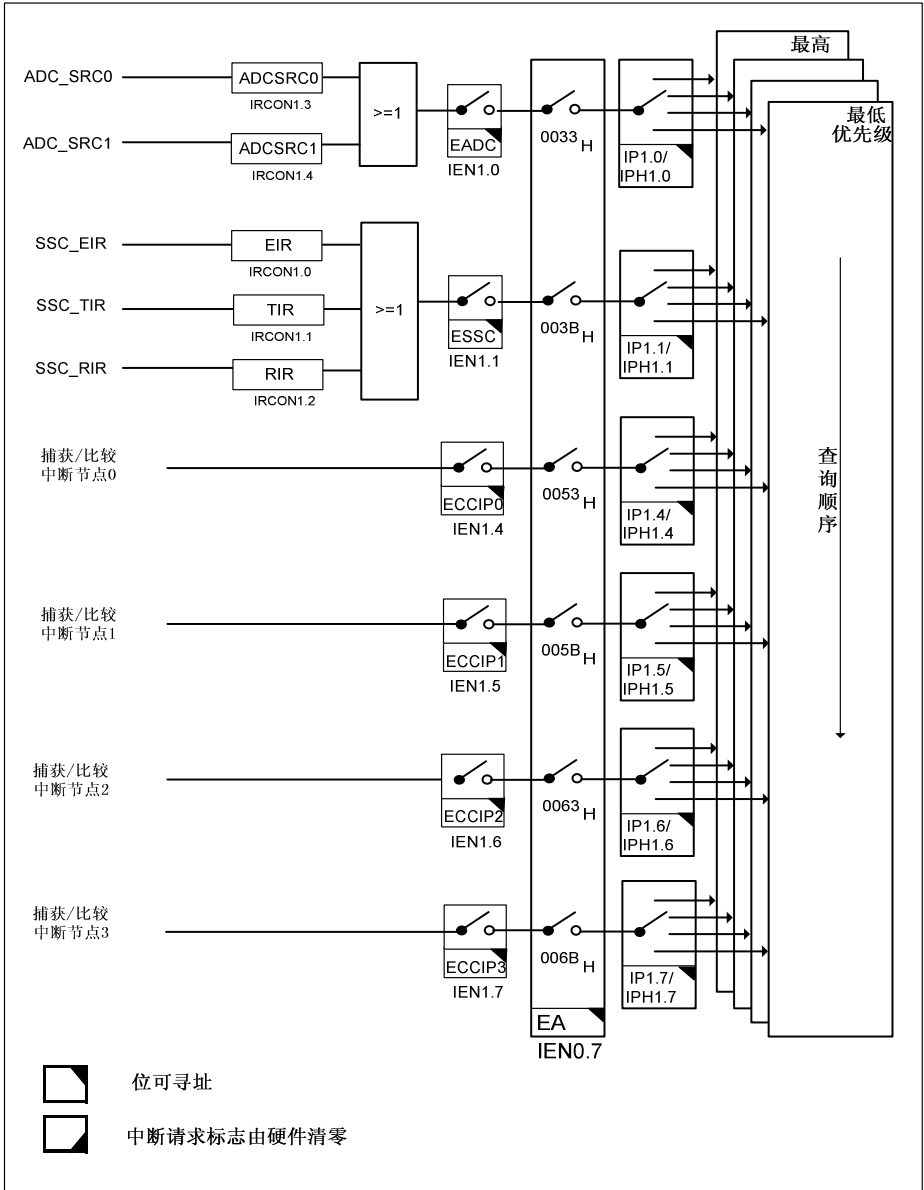


图 5-4 中断请求源 (第三部分)

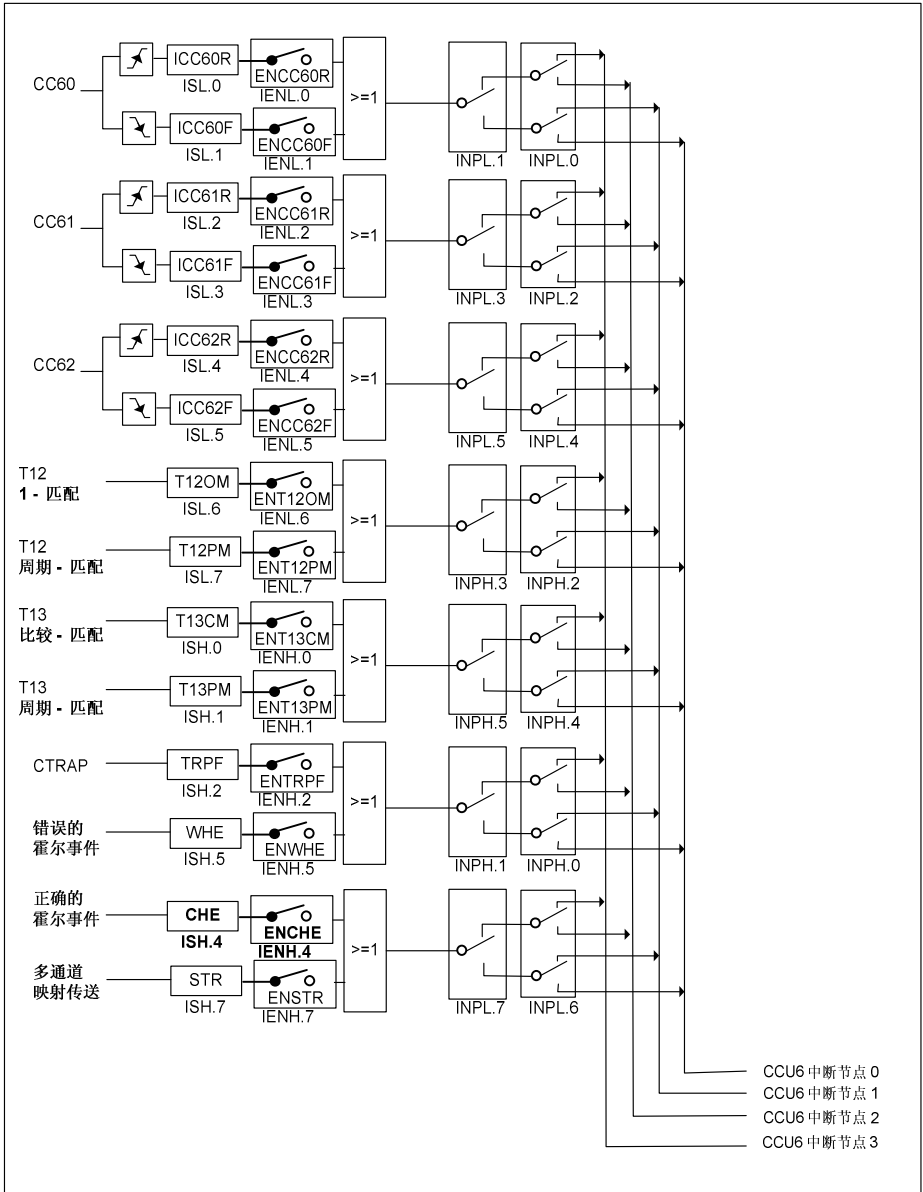


图 5-5 中断请求源 (第四部分)

5.3 中断源和中断向量

每个中断输入对应一个中断向量地址。从该中断向量转入相应中断源的中断服务程序。XC866 的中断源分配见表 5-1。

表 5-1 中断向量地址

中断输入	中断向量地址	中断请求源
NMI	0073 _H	WDT, PLL, Flash 接口定时器, OCDS, VDD 和 VDDP 预报警, Flash 纠错码
XINTR0	0003 _H	外部中断 0
XINTR1	000B _H	定时器 0
XINTR2	0013 _H	外部中断 1
XINTR3	001B _H	定时器 1
XINTR4	0023 _H	UART
XINTR5	002B _H	定时器 2
XINTR6	0033 _H	ADC_SRC[1:0]
XINTR7	003B _H	SSC
XINTR8	0043 _H	外部中断 2
XINTR9	004B _H	外部中断[6:3]
XINTR10	0053 _H	CCU6 中断节点指针 0
XINTR11	005B _H	CCU6 中断节点指针 1
XINTR12	0063 _H	CCU6 中断节点指针 2
XINTR13	006B _H	CCU6 中断节点指针 3

5.4 中断寄存器描述

5.4.1 中断使能寄存器

每个中断输入可通过置位或清零中断使能寄存器 IEN0 和 IEN1 中的相应位，被分别使能或禁止。寄存器 IEN0 中还包含了全局使能/禁止位（EA），清零后禁止所有中断。

很多中断源共用 NMI 中断，每个中断源可通过寄存器 NMICON 被分别使能或禁止。

复位后，IEN0, IEN1 和 NMICON 中的使能位全部清零，这意味着相应中断被禁止。

IEN0

中断使能寄存器 0

复位值: 00H

7	6	5	4	3	2	1	0
EA	0	ET2	ES	ET1	EX1	ET0	EX0
rw	r	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
EX0	0	rw	外部中断 0 使能 0 禁止外部中断 0 1 使能外部中断 0
ET0	1	rw	定时器 0 溢出中断使能 0 禁止定时器 0 溢出中断 1 使能定时器 0 溢出中断
EX1	2	rw	外部中断 1 使能 0 禁止外部中断 1 1 使能外部中断 1
ET1	3	rw	定时器 1 溢出中断使能 0 禁止定时器 1 溢出中断 1 使能定时器 1 溢出中断

符号	位序号	读写类型	功能描述
ES	4	rw	串行口中断使能 0 禁止串行口中断 1 使能串行口中断
ET2	5	rw	定时器 2 中断使能 0 禁止定时器 2 中断 1 使能定时器 2 中断
EA	7	rw	使能/禁止所有中断 0 中断不被响应 1 通过置位或清除相应使能位，分别使能或禁止每个中断源
0	6	r	保留 读操作返回 0；应写入 0

IEN1
中断使能寄存器 1
复位值: 00H

7	6	5	4	3	2	1	0
ECCIP3	ECCIP2	ECCIP1	ECCIP0	EXM	EX2	ESSC	EADC
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
EADC	0	rw	ADC 中断使能 0 禁止 ADC 中断 1 使能 ADC 中断
ESSC	1	rw	SSC 中断使能 0 禁止 SSC 中断 1 使能 SSC 中断
EX2	2	rw	外部中断 2 使能 0 禁止外部中断 2 1 使能外部中断 2

符号	位序号	读写类型	功能描述
EXM	3	rw	外部中断[6:3]使能 0 禁止外部中断[6:3] 1 使能外部中断[6:3]
ECCIP0	4	rw	CCU6 中断节点指针 0 使能 0 禁止 CCU6 中断节点指针 0 1 使能 CCU6 中断节点指针 0
ECCIP1	5	rw	CCU6 中断节点指针 1 使能 0 禁止 CCU6 中断节点指针 1 1 使能 CCU6 中断节点指针 1
ECCIP2	6	rw	CCU6 中断节点指针 2 使能 0 禁止 CCU6 中断节点指针 2 1 使能 CCU6 中断节点指针 2
ECCIP3	7	r	CCU6 中断节点指针 3 使能 0 禁止 CCU6 中断节点指针 3 1 使能 CCU6 中断节点指针 3

NMICON
NMI 控制寄存器
复位值: 00H

	7	6	5	4	3	2	1	0
0	NMIECC	NMI VDDP	NMIVDD	NMI OCDS	NMI-FLASH-TIMER	NMIPLL	NMI WDT	
	r	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
NMIWDT	0	rw	WDT NMI 使能 0 禁止 WDT NMI 1 使能 WDT NMI

符号	位序号	读写类型	功能描述
NMIPLL	1	rw	PLL 失锁 NMI 使能 0 禁止 PLL 失锁 NMI 1 使能 PLL 失锁 NMI
NMIFLASH-TIMER	2	rw	Flash 定时器 NMI 使能 0 禁止 Flash 定时器 NMI 1 使能 Flash 定时器 NMI
NMIOCDs	3	rw	OCDS NMI 使能 0 禁止 OCDS NMI 1 使能 OCDS NIM
NMIVDD	4	rw	VDD 预报警 NMI 使能 0 禁止 VDD NMI 1 使能 VDD NMI
NMIVDDP	5	rw	VDDP 预报警 NMI 使能 0 禁止 VDDP NMI 1 使能 VDDP NMI <i>注：当外部电源为 3.3V，用户必须禁止 NMIVDDP。</i>
NMIECC	6	rw	ECC NMI 使能 0 禁止 ECC NMI 1 使能 ECC NMI
0	7	r	保留 读操作返回 0；应写入 0

EXICON0

外部中断控制寄存器 0

复位值: 00_H

7 6 5 4 3 2 1 0

EXINT3	EXINT2	EXINT1	EXINT0
rw	rw	rw	rw

符号	位序号	读写类型	功能描述
EXINT0	[1:0]	rw	外部中断 0 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 边沿检测旁路
EXINT1	[3:2]	rw	外部中断 1 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 边沿检测旁路
EXINT2	[5:4]	rw	外部中断 2 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 保留
EXINT3	[7:6]	rw	外部中断 3 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 保留

EXICON1

外部中断控制寄存器 1

复位值: 00_H

7	6	5	4	3	2	1	0
0		EXINT6		EXINT5		EXINT4	
r		rw		rw		rw	

符号	位序号	读写类型	功能描述
EXINT4	[1:0]	rw	外部中断 4 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 保留
EXINT5	[3:2]	rw	外部中断 5 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 保留
EXINT6	[5:4]	rw	外部中断 6 触发选择 00 下降沿触发中断 01 上升沿触发中断 10 上升沿和下降沿均触发中断 11 保留
0	[7:6]	r	保留 读操作返回 0; 应写入 0

5.4.2 中断请求标志

中断请求源的中断请求标志存放在多个 SFR 中。这部分将详细描述中断请求标志的位置及含义。

IRCON0

中断请求寄存器 0

复位值: 00_H

7	6	5	4	3	2	1	0
0	EXINT6	EXINT5	EXINT4	EXINT3	EXINT2	EXINT1	EXINT0
r	rwh	rwh	rwh	rwh	rwh	rwh	rwh

符号	位序号	读写类型	功能描述
EXINT_x (X=0 - 6)	[6:0]	rwh	外部中断请求标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
0	7	r	保留 读操作返回 0；应写入 0

IRCON1

中断请求寄存器 1

复位值: 00H

7	6	5	4	3	2	1	0
0		ADC SRC1	ADC SRC0	RIR	TIR	EIR	
r		rwh	rwh	rwh	rwh	rwh	


符号	位序号	读写类型	功能描述
EIR	0	rwh	SSC 出错中断请求标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
TIR	1	rwh	SSC 发送中断请求标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
RIR	2	rwh	SSC 接收中断请求标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
ADCSRC0	3	rwh	ADC 中断请求 0 标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
ADCSRC1	4	rwh	ADC 中断请求 1 标志 该位由硬件置位，只能由软件清零 0 未产生中断请求 1 产生中断请求
0	[7:5]	r	保留 读操作返回 0；应写入 0

TCON

定时器控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
rwh	rw	rwh	rw	rwh	rw	rwh	rw

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
IT0	0	rw	外部中断 0 电平/边沿触发控制标志 0 选择低电平触发外部中断 0 1 选择下降沿触发外部中断 0
IE0	1	rwh	外部中断 0 请求标志 当检测到外部中断 0 的边沿跳变时由硬件置位。 处理器响应中断转入中断服务程序时, 该标志由硬件清零。
IT1	2	rw	外部中断 1 电平/边沿触发控制标志 0 选择低电平触发外部中断 1 1 选择下降沿触发外部中断 1
IE1	3	rwh	外部中断 1 请求标志 当检测到外部中断 1 的边沿跳变时由硬件置位。 处理器响应中断转入中断服务程序时, 该标志由硬件清零。
TF0	5	rwh	定时器 0 溢出标志 定时器/计数器 0 溢出时由硬件置位 处理器响应中断转入中断服务程序时, 该标志由硬件清零。
TF1	7	rwh	定时器 1 溢出标志 定时器/计数器 1 溢出时由硬件置位

符号	位序号	读写类型	功能描述
			处理器响应中断转入中断服务程序时，该标志由硬件清零。

SCON

串行通道控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
rw	rw	rw	rw	rw	rwh	rwh	rwh

阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
RI	0	rwh	串行接口接收中断标志 当完成一帧串行数据的接收，由硬件置位；必须由软件清零
TI	1	rwh	串行接口发送中断标志 当完成一帧串行数据的发送，由硬件置位；必须由软件清零

NMISR

NMI 状态寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
0	FNMI ECC	FNMI VDDP	FNMI VDD	FNMI OCDS	FNMI- FLASH- TIMER	FNMI PLL	FNMI WDT
r	rwh	rwh	rwh	rwh	rwh	rwh	rwh

符号	位序号	读写类型	功能描述
FNMIWDT	0	rwh	WDT NMI 标志 0 未产生 WDT NMI 1 WDT 已预报警
FNMIPLL	1	rwh	PLL 失锁 NMI 标志 0 未产生 PLL 失锁 NMI 1 PLL 已与外部晶振失锁
FNMIFLASH- TIMER	2	rwh	FLASH 定时器 NMI 标志 0 未产生 Flash 定时器 NMI 1 Flash 定时器已溢出
FNMIOCDS	3	rwh	OCDS NMI 标志 0 未产生 OCDS NMI 1 监控模式下已产生 JTAG 接收请求或用户中断请求
FNMIVDD	4	rwh	VDD 预报警 NMI 标志 0 未产生 VDD NMI 1 V _{DD} 已低于预报警电压 (2.3V)
FNMIVDDP	5	rwh	VDDP 预报警 NMI 标志 0 未产生 VDDP NMI 1 V _{DDP} 已低于预报警电压 (4.0V, 若外部电压为 5.0V)
FNMIIECC	6	rwh	ECC NMI 标志 0 未产生 ECC 错误

符号	位序号	读写类型	功能描述
			1 ECC 已出错
0	7	r	保留 读操作返回 0；应写入 0

NMISR 寄存器只能由软件清零，或在上电复位/ 硬件复位/ 压降复位时被置为缺省值。在其他复位方式下，如 WDT 复位或掉电唤醒复位时，该寄存器内容保持不变，从而使系统确定产生前次 NMI 的中断源。

5.4.3 中断优先级寄存器

每个中断源的优先级（共有四级优先级）可分别设定。有两对中断优先级寄存器来设定每个中断向量的优先级。第一对寄存器是 SFR IP 和 IPH。

IP

中断优先级寄存器

 复位值: 00_H

7	6	5	4	3	2	1	0
0	PT2	PS	PT1	PX1	PT0	PX0	
r	rw	rw	rw	rw	rw	rw	rw

IPH

中断优先级寄存器，高位字节

 复位值: 00_H

7	6	5	4	3	2	1	0
0	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	
r	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
PX0, PX0H	0	rw	外部中断 0 优先级
PT0, PT0H	1	rw	定时器 0 溢出中断优先级
PX1, PX1H	2	rw	外部中断 1 优先级
PT1, PT1H	3	rw	定时器 1 溢出中断优先级
PS, PSH	4	rw	串口中断优先级
PT2, PT2H	5	rw	定时器 2 中断优先级
0	[7:6]	r	保留 读操作返回 0; 应写入 0

第二对中断优先级寄存器是 SFR IP1 和 IPH1。

IP1
中断优先级寄存器 1
复位值: 00H

7	6	5	4	3	2	1	0
PCCIP3	PCCIP2	PCCIP1	PCCIP0	PXM	PX2	PSSC	PADC
rw	rw	rw	rw	rw	rw	rw	rw

IPH1
中断优先级寄存器 1, 高位字节
复位值: 00H

7	6	5	4	3	2	1	0
PCC IP3H	PCC IP2H	PCC IP1H	PCC IP0H	PXMH	PX2H	PSSCH	PADCH
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
PADC, PADCH	0	rw	ADC 中断优先级
PSSC, PSSCH	1	rw	SSC 中断优先级
PX2, PX2H	2	rw	外部中断 2 优先级
PXM, PXMH	3	rw	外部中断 3 至 6 优先级
PCCIP0, PCCIP0H	4	rw	CCU6 中断节点指针 0 优先级
PCCIP1, PCCIP1H	5	rw	CCU6 中断节点指针 1 优先级
PCCIP2, PCCIP2H	6	rw	CCU6 中断节点指针 2 优先级

符号	位序号	读写类型	功能描述
PCCIP3, PCCIP3H	7	rw	CCU6 中断节点指针 3 优先级

5.4.4 中断优先级

每对中断优先级寄存器中对应位置的位域共同决定了相应中断的优先级，见表 5-2。

表 5-2 中断优先级选择

IPH.X / IPH1.X	IP.X / IP1.X	优先级
0	0	优先级 0 (最低)
0	1	优先级 1
1	0	优先级 2
1	1	优先级 3 (最高)

注：由于 NMI 优先级最高，故不用表 5-2 进行优先级选择。

低优先级中断服务程序可响应更高级的中断，但不能响应同级或低级中断。最高级中断不响应其他任何中断。

如果两个或更多不同优先级的中断源同时请求中断时，首先响应最高优先级的中断请求。如果几个同优先级的中断源同时请求中断时，内部查询顺序将决定首先响应哪一个中断请求。因此，在相同优先级内又有由查询顺序决定的次级优先级结构，见表 5-3。

表 5-3 同级内的优先级结构

中断源	优先级
非屏蔽中断 (NMI)	(最高)
外部中断 0	1
定时器 0 中断	2
外部中断 1	3
定时器 1 中断	4
UART 中断	5
定时器 2 中断	6

中断源	优先级
ADC 中断	7
SSC 中断	8
外部中断 2	9
外部中断[6:3]	10
CCU6 中断节点指针 0	11
CCU6 中断节点指针 1	12
CCU6 中断节点指针 2	13
CCU6 中断节点指针 3	14

5.4.5 中断请求标志

中断请求标志位于不同的 SFR 中。表 5-4 列出中断请求标志的位置。中断请求标志的详细描述将在相关外设章节中给出。

表 5-4 中断请求标志位置

中断源	中断请求标志	SFR
定时器 0 中断	TF0	TCON
定时器 1 中断	TF1	TCON
定时器 2 中断	TF2	T2CON
	EXF2	T2CON
UART 中断	RI	SCON
	TI	SCON
外部中断 0	IE0	TCON
外部中断 1	IE1	TCON
外部中断 2	EXINT2	IRCON0
外部中断 3	EXINT3	IRCON0
外部中断 4	EXINT4	IRCON0
外部中断 5	EXINT5	IRCON0
外部中断 6	EXINT6	IRCON0
ADC 中断	ADCSRC0	IRCON1
	ADCSRC1	IRCON1
SSC 中断	EIR	IRCON1
	TIR	IRCON1
	RIR	IRCON1
CCU6 中断节点指针 0	见注 ¹⁾	INPL/INPH
CCU6 中断节点指针 1	见注 ¹⁾	INPL/INPH
CCU6 中断节点指针 2	见注 ¹⁾	INPL/INPH

中断源	中断请求标志	SFR
CCU6 中断节点指针 3	见注 ¹⁾	INPL/INPH
WDT NMI	FNMIWDT	NMISR
PLL NMI	FNMIPLL	NMISR
Flash 定时器 NMI	FNMIFLASHTIMER	NMISR
OCDS NMI	FNMIOCDS	NMISR
VDD NMI	FNMIVDD	NMISR
VDDP NMI	FNMIVDDP	NMISR
ECC NMI	FNMIECC	NMISR

¹⁾ 将 CCU6 产生的各个中断分配给不同的中断节点 [3:0]，由寄存器 INPL/INPH 选择。

5.5 中断处理

在每个机器周期的 P2 时采样中断标志，在下一个机器周期对采样到的中断标志进行查询。如果某中断标志在前一周期的 P2 已被置位，查询周期将发现该标志，中断系统将产生一个 LCALL 指令调用相应的中断服务程序。由硬件生成的 LCALL 指令在下列任意一种情况下都会推迟执行：

1. 同级或更高级的中断已在处理中。
2. 当前周期（查询周期）不是正在执行指令的最后一个周期。
3. 正在执行的指令是 RETI 或是对寄存器 IEN0/IEN1 或 IP, IPH/IP1, IP1H 的写操作。

上述任意一种情况都会推迟执行 LCALL 进入中断服务程序。条件 2 保证了正在执行的指令在进入中断服务程序之前可执行完毕。条件 3 保证了如果正在执行的指令是 RETI 或是对寄存器 IEN0/IEN1 或 IP, IPH/IP1, IP1H 的写操作时，进入中断服务程序之前必须至少再执行一条指令，该延迟保证了中断状态的改变可被 CPU 监测到。

中断查询在每个机器周期重复执行，查询到的值为前一机器周期 P2 的采样值。注意，如中断标志有效（置位），但由于上述的条件之一而未被响应；或上述阻止条件撤销后中断标志已不再有效，中断将不再被响应。换言之，曾经有效的中断请求标志若未能被 CPU 及时响应，将不被记忆。每个查询周期仅查询挂起的中断请求。

扩展中断的时序如图 5-6 所示。

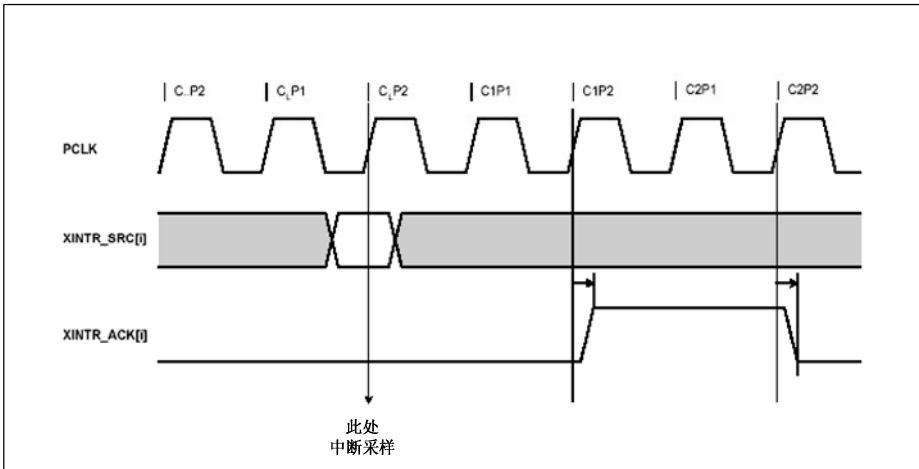


图 5-6 扩展中断时序

因此，处理器通过执行由硬件生成的 **LCALL** 指令调用相应的中断服务程序来响应中断。在有些情况下，硬件清除中断标志；另一些情况下，必须由用户软件清除中断标志。由硬件生成的 **LCALL** 会把程序计数器的内容压入堆栈（但不保存程序状态字的内容），并根据响应的的中断源重新将中断向量地址装入 **PC**，如表 5-1 所示。

中断服务程序执行到 **RETI** 指令时，程序返回继续执行调用中断之后的下条指令。**RETI** 指令通知处理器中断服务程序已执行完毕，然后从堆栈弹出两个字节重新装入 **PC**，继续执行被中断的程序。需要注意的是，**RETI** 指令非常重要，它通知处理器中断服务程序已执行完毕。简单的 **RET** 指令也可以返回到被中断的程序，但这样会使中断控制系统认为中断服务程序仍在执行。这种情况下，同级或低级中断就无法响应。

5.6 中断响应时间

如果识别到一个中断，在每个机器周期的 P2 将置位相应的中断请求标志，下一个机器周期之前不会被电路查询。如果中断请求有效并且响应中断的条件成立，下一条将执行硬件生成的调用指令，进入中断服务程序。调用指令本身占用两个周期，因此，从中断请求有效到执行中断服务程序的第一条指令需要至少三个完整的机器周期。如果中断请求被上述三个条件之一所阻滞，则需要更长的响应时间。如果一个同级或高优先级的中断已在处理，附加的等待时间取决于其他中断服务程序所耗的时间；如果正在执行的指令不是它的最后一个周期，则附加的等待时间不会超过三个机器周期，因为最长的指令（MUL 和 DIV 指令）仅四个机器周期；如果正在执行的指令是 RETI 指令或对寄存器 IEN0, IEN1, 或 IP(H), IP1(H)的写操作，则附加的等待时间不会超过五个机器周期（最多再用一个机器周期完成当前指令，再加四个周期完成下条指令，如果指令是 MUL 或 DIV 的话）。因此，在单中断系统中，如果不考虑等待状态，响应时间总是大于 3 个周期小于 9 个周期。

6 并行端口

XC866 有 27 个端口引脚，组成四个并行端口，P0 口到 P3 口。每个引脚都有一对可分别被使能或禁止的内部上拉/下拉器件。P0，P1 和 P3 是双向口，可用作通用 I/O（GPIO）或片内外设的 I/O 功能选择。设置为输出口时，可选择开漏输出模式。P2 口是单向输入口，用作通用输入、片内外设的输入功能选择以及 ADC 的模拟输入。

双向口特性：

- 引脚方向可设置
- 上拉/下拉器件可设置
- 开漏输出模式可设置
- 通过数字 I/O 口传送数据（GPIO）
- 片内外设的 I/O 功能选择

单向输入口特性：

- 上拉/下拉器件可设置
- 通过数字输入口接收数据（通用输入口）
- 片内外设的输入功能选择
- ADC 的模拟输入

6.1 基本端口操作

XC866 双向口引脚的结构框图如图 6-1 所示。每个引脚由一组控制位和数据位来配置，从而使用引脚时具有很大的灵活性。通过定义控制寄存器，可将每个引脚分别设置成输入或输出口。用户还可以把引脚设置成带有或者不带内部上拉/下拉器件的开漏输出模式。

每个双向口可被设置成输入或输出口。由寄存器 Px_DIR ($x=0, 1$ 或 3) 控制输入输出模式之间的切换、开启或关闭输出和输入驱动器。任何时刻端口只能被设置成输入或输出模式之。

输入模式下（复位后的缺省模式），输出驱动器关闭（高阻）。引脚上的实际电压值由施密特触发器翻译成逻辑 0 或 1，可从寄存器 Px_DATA 中读取。

输出模式下，输出驱动器被激活，将多选一输出选择器送出的值输出至端口引脚。在输出驱动器中，通过控制寄存器 Px_OD ，可将每个端口切换至开漏输出模式或正常模式（推挽模式）。

输出驱动器的前级多选一输出选择器使输出口可用作多种功能。如果引脚用作通用输出口，软件控制多选一输出选择器选通数据寄存器 Px_DATA ；软件可对 Px_DATA 置位或清零，从而直接影响引脚状态。如果引脚用作片内外设的输出口，多选一输出选择器会选通其他功能输出线（AltDataOut），将其送至输出驱动电路。由寄存器 $Px_ALTSEL0$ 和 $Px_ALTSEL1$ 选择引脚的其他功能。当引脚用作其他功能时，必须在寄存器 Px_DIR 中相应设置引脚方向。

每个引脚还可编程设置、激活内部的弱上拉或下拉器件。寄存器 Px_PUDEN 使能或禁止拉动器件，寄存器 Px_PUDSEL 选择激活上拉或下拉器件。

为了实现高速 I/O 数据传送，每个 I/O 口可以和外设单元的不同输入端（AltDataIn）直接相连。从引脚到数据寄存器 Px_DATA 和到 AltDataIn 的输入功能与当前引脚是输入还是输出操作无关。这意味着，当引脚用作输出时，引脚电平可由软件从 Px_DATA 中读出；或该电平可用作外设的输入。这为芯片应用提供了额外的好处：

- 引脚设置为通用输出口时，写入寄存器 Px_DATA 中的数据可用作片内外设的输入。从而能够通过软件来测试外设，不需添加外部电路。这样的例子有：触发定时器计数输入、产生外部中断，或软件仿真进入串口接收端的串行数据流。
- 引脚设置为其他功能输出时，外设驱动到引脚上的输出数据可由软件从 Px_DATA 读出，或被同一个外设或其他外设用作输入。从而能够用软件测试外设的功能；或用同一引脚为片内外设提供额外的连接，不需添加外部连线。

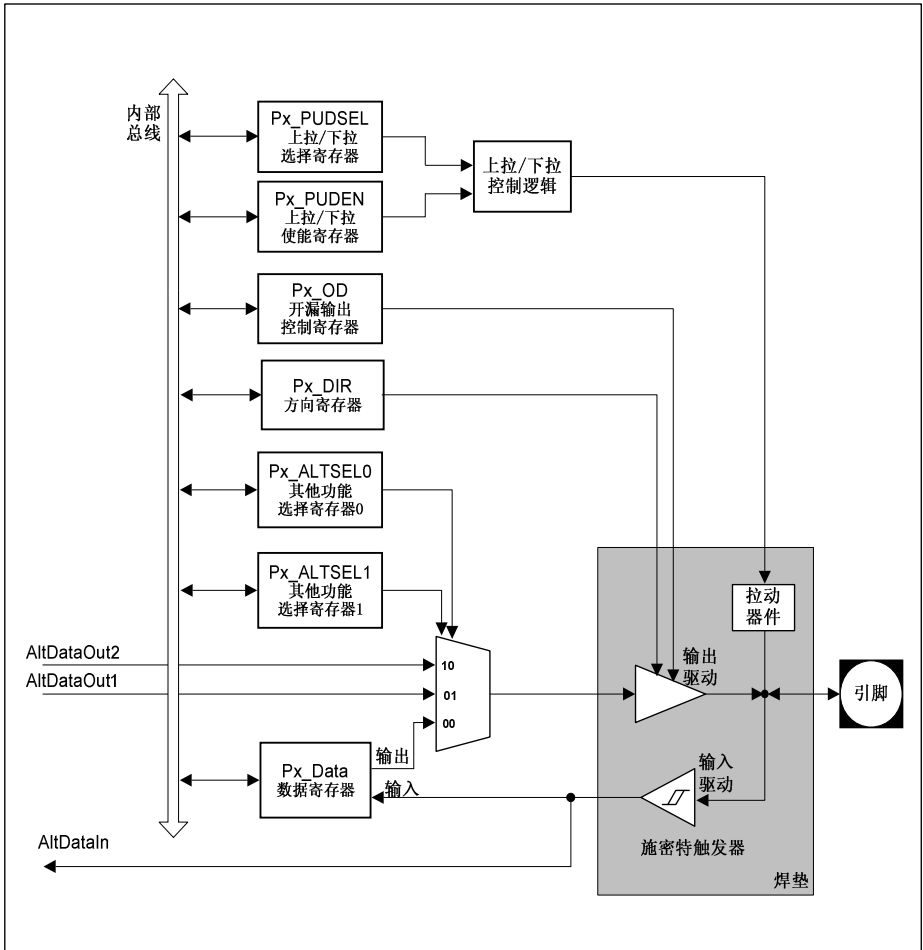


图 6-1 双向口基本结构

单向输入引脚结构如图 6-2 所示。P2 口引脚只工作在输入模式。引脚上的实际电压值由施密特触发器翻译成逻辑 0 或 1，可从寄存器 P_x_DATA 读出。每个引脚还可编程设置、激活内部的弱上拉或下拉器件。寄存器 P_x_PUDEN 使能或禁止拉动器件，寄存器 P_x_PUDSEL 选择激活上拉或下拉器件。模拟输入（AnalogIn）绕过数字电路和施密特触发器直接进入 ADC 的输入通道。

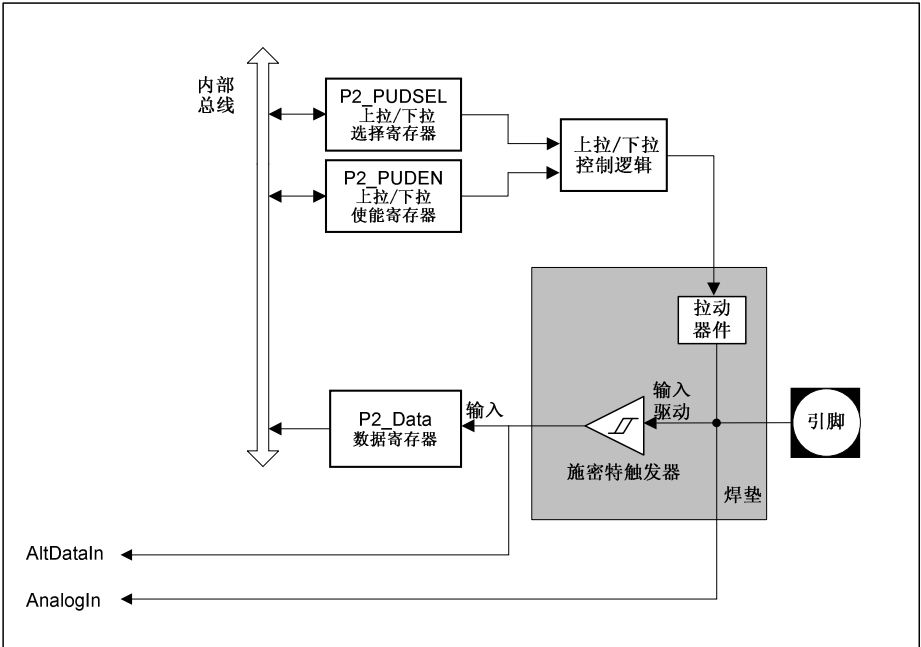


图 6-2 输入口基本结构

6.1.1 基本寄存器描述

每个并行口的控制位和数据位由一组 8 位寄存器实现。具有相同含义和功能的位存放在同一个寄存器中。这些寄存器将端口设置为 GPIO 口或其他功能 I/O 口。

P2 口不必实现表 6-1 所列出的全部寄存器。每个端口具体的寄存器定义见章节 6.3 至章节 6.6。本节仅旨在给出不同端口寄存器的概览。

表 6-1 端口寄存器

寄存器缩略名	寄存器完整名	具体描述请参见
Px_DATA	端口 x 数据寄存器	页 6-6
Px_DIR	端口 x 方向寄存器	页 6-6
Px_OD	端口 x 开漏输出控制寄存器	页 6-7
Px_PUDSEL	端口 x 上拉/下拉选择寄存器	页 6-8
Px_PUDEN	端口 x 上拉/下拉使能寄存器	页 6-8
Px_ALTSEL0	端口 x 其他功能选择寄存器 0	页 6-9
Px_ALTSEL1	端口 x 其他功能选择寄存器 1	页 6-9

6.1.1.1 数据寄存器

如果引脚用作通用输出口，输出数据写入数据寄存器 Px_DATA 中。如果引脚用作通用输入口，端口引脚的锁存值可从寄存器 Px_DATA 中读出。

注：若无外部驱动，用作输入的端口引脚将会锁存有效的内部上拉/下拉设置；用该有效的上拉/下拉值更新寄存器 Px_DATA。

Px_DATA

端口 x 数据寄存器

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	端口 x 引脚 n 的数据值 0 端口 x 引脚 n 的数据值 = 0 1 端口 x 引脚 n 的数据值 = 1

位 Px_DATA.n 只有在相应引脚设置为输出 (Px_DIR.n=1) 时可被写入，在引脚设置为输入 (Px_DIR.n=0) 时不能写入。当引脚设置为 GPIO、且方向为输出时，Px_DATA.n 的内容输出到指定引脚上。读取 Px_DATA 将返回寄存器的值，而不是相应 Px_DATA 引脚上的状态。

6.1.1.2 方向寄存器

端口引脚的方向由方向寄存器 Px_DIR 控制。

Px_DIR

端口 x 方向寄存器

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	端口 x 引脚 n 方向控制 0 设置为输入引脚 1 设置为输出引脚

6.1.1.3 开漏输出控制寄存器

每个输出引脚均可选择开漏输出模式。如果用 1 驱动，驱动器关闭，引脚的输出状态取决于内部上拉/下拉器件的设置。如果用 0 驱动，驱动器的下拉晶体管导通。

开漏输出模式由寄存器 Px_OD 控制。

Px_OD

端口 x 开漏输出控制寄存器

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	端口 x 引脚 n 开漏输出模式选择 0 正常模式；0 和 1 可有效输出 1 开漏输出模式；只有 0 可有效输出

6.1.1.4 上拉/下拉器件寄存器

端口引脚可选择使用内部上拉或下拉器件，从而端口可设置为具备以下输入特性：

- 三态
- 带有弱上拉的高阻态
- 带有弱下拉的高阻态

和以下输出特性：

- 推挽（选择上拉/下拉）
- 带有内部上拉的开漏输出
- 带有外部上拉的开漏输出

上拉/下拉器件可由寄存器 Px_PUDSEL 和 $Px_PU DEN$ 控制。寄存器 Px_PUDSEL 选择上拉/下拉类型；寄存器 $Px_PU DEN$ 使能或禁止上拉/下拉器件。可按每个引脚来选择上拉/下拉器件。

Px_PUDSEL

端口 x 上拉/下拉选择寄存器

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn ($n = 0 - 7$)	n	rw	端口 x 引脚 n 上拉/下拉选择 0 选择下拉器件 1 选择上拉器件

$Px_PU DEN$

端口 x 上拉/下拉使能寄存器

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn ($n = 0 - 7$)	n	rw	端口 x 引脚 n 上拉/下拉使能 0 禁止上拉或下拉 1 使能上拉或下拉

6.1.1.5 其他输入功能

一个端口引脚上的其他输入功能数目不限。每个 I/O 口的端口控制逻辑提供了以下的输入通路：

- 从寄存器读取数字输入值
- 直接读取数字输入值

6.1.1.6 其他输出功能

由多选一输出选择器选择其他输出功能。由以下寄存器控制选择：

- 寄存器 Px_ALTSEL0
- 寄存器 Px_ALTSEL1

由寄存器 Px_ALTSEL0 和 Px_ALTSEL1 控制其他功能选择。

Px_ALTSELn (n = 0 - 1)

端口 X 其他功能选择寄存器

复位值: 00H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

位 Px_ALTSEL0.Pn 和位 Px_ALTSEL1.Pn 的功能

Px_ALTSEL0.Pn	Px_ALTSEL1.Pn	功能
0	0	正常 GPIO 功能
1	0	其他功能输出 1
0	1	其他功能输出 2
1	1	保留

注：将 Px_ALTSEL0.Pn 和 Px_ALTSEL1.Pn 设置为已实现的其他输出功能。

6.2 寄存器映射

端口 SFR 在标准存储器区 (RMAP=0)，由四页构成。PORT_PAGE 寄存器地址 B2_H，包含了分页值和页控制信息。

PORT_PAGE

端口分页寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
OP		STNR		0	PAGE		
w		w		r	rw		

符号	位序号	读写类型	功能描述
PAGE	[2:0]	rw	分页位 写入时，该值表示新页的值 读出时，该值表示当前有效页的值
STNR	[5:4]	w	保存编号 该编号指明在哪个保存位域上执行 OP 定义的操作。 若 OP=10 _B ， PAGE 的内容在被新值覆盖之前保存在 STx 中 若 OP=11 _B ， PAGE 的内容被 STx 覆盖。写入 PAGE 的值不予理睬 00 选择 ST0 01 选择 ST1 10 选择 ST2 11 选择 ST3
OP	[7:6]	w	操作 0X 手动保存页模式，STNR 的值被忽略， PAGE 被直接写入 10 带有自动页保存的新页设置。当前写入 PAGE 中的内容被保存的同时，上次

符号	位序号	读写类型	功能描述
			写入 PAGE 的内容被保存在 STNR 规定的位域 STx 中 11 自动恢复页。对写入 PAGE 的内容不予理睬，PAGE 由 STNR 规定的位域 STx 中的内容覆盖
0	3	r	保留 读操作返回 0；应写入 0

端口 SFR 的地址列表见表 6-2。

表 6-2 页 0-3 的 SFR 地址列表

地址	页 0	页 1	页 2	页 3
80 _H	P0_DATA	P0_PUDSEL	P0_ALTSEL0	P0_OD
86 _H	P0_DIR	P0_PUDEN	P0_ALTSEL1	-
90 _H	P1_DATA	P1_PUDSEL	P1_ALTSEL0	P1_OD
91 _H	P1_DIR	P1_PUDEN	P1_ALTSEL1	-
A0 _H	P2_DATA	P2_PUDSEL	-	-
A1 _H	-	P2_PUDEN	-	-
B0 _H	P3_DATA	P3_PUDSEL	P3_ALTSEL0	P3_OD
B1 _H	P3_DIR	P3_PUDEN	P3_ALTSEL1	-

6.3 P0 口

P0 口是一个 6 位通用双向口。P0 口寄存器总结见表 6-3。

表 6-3 P0 口寄存器

寄存器缩略名	寄存器完整名
P0_DATA	P0 口数据寄存器
P0_DIR	P0 口方向寄存器
P0_OD	P0 口开漏输出控制寄存器
P0_PUDSEL	P0 口上拉/下拉选择寄存器
P0_PUDEN	P0 口上拉/下拉使能寄存器
P0_ALTSEL0	P0 口其他功能选择寄存器 0
P0_ALTSEL1	P0 口其他功能选择寄存器 1

6.3.1 功能

表 6-4 P0 口 I/O 功能

端口引脚	I/O	功能选择	连接信号	来自/送至模块
P0.0	输入	GPI	P0_DATA.P0	-
		ALT1	TCK_0	JTAG
		ALT2	T12HR_1	CCU6
		ALT3	CC61_1	CCU6
	输出	GPO	P0_DATA.P0	-
		ALT1	CLKOUT	片内 OSC
ALT2		CC61_1	CCU6	
P0.1	输入	GPI	P0_DATA.P1	-
		ALT1	TDI_0	JTAG
		ALT2	T13HR_1	CCU6

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
	输出	ALT3	RXD_1	UART
		GPO	P0_DATA.P1	-
		ALT1	-	-
		ALT2	COU61_1	CCU6
P0.2	输入	GPI	P0_DATA.P2	-
		ALT1	-	-
		ALT2	$\overline{\text{CTRAP_2}}$	CCU6
		ALT3	-	-
	输出	GPO	P0_DATA.P2	-
		ALT1	TDO_0	JTAG
		ALT2	TXD_1	UART
P0.3	输入	GPI	P0_DATA.P3	-
		ALT1	SCK_1	SSC
		ALT2	-	-
		ALT3	-	-
	输出	GPO	P0_DATA.P3	-
		ALT1	SCK_1	SSC
		ALT2	COU63_1	CCU6
P0.4	输入	GPI	P0_DATA.P4	-
		ALT1	M TSR_1	SSC
		ALT2	-	-
		ALT3	CC62_1	CCU6
	输出	GPO	P0_DATA.P4	-
		ALT1	M TSR_1	SSC
		ALT2	CC62_1	CCU6
P0.5	输入	GPI	P0_DATA.P5	-

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
		ALT1	MRST_1	SSC
		ALT2	EXINT0_0	外部中断 0
		ALT3	-	-
	输出	GPO	P0_DATA.P5	-
		ALT1	MRST_1	SSC
		ALT2	COU62_1	CCU6

6.3.2 寄存器描述
P0_DATA
P0 口数据寄存器

复位值: 00H

7	6	5	4	3	2	1	0
0		P5	P4	P3	P2	P1	P0
r		rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 5)	n	rw	P0 口引脚 n 的数据值 0 P0 口引脚 n 的数据值 = 0 (缺省值) 1 P0 口引脚 n 的数据值 = 1
0	[7:6]	r	保留 读操作返回 0; 应写入 0

P0_DIR
P0 口方向寄存器

复位值: 00H

7	6	5	4	3	2	1	0
0		P5	P4	P3	P2	P1	P0
r		rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n =0 - 5)	n	rw	P0 口引脚 n 方向控制 0 设置为输入引脚 (缺省值) 1 设置为输出引脚
0	[7:6]	r	保留 读操作返回 0; 应写入 0

P0_OD

P0 口开漏输出控制寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
0	P5	P4	P3	P2	P1		
r	rw	rw	rw	rw	rw		

符号	位序号	读写类型	功能描述
Pn (n = 0 - 5)	n	rw	P0 口引脚 n 开漏输出模式选择 0 正常模式; 0 和 1 可有效输出 (缺省值) 1 开漏输出模式; 只有 0 可有效输出
0	[7:6]	r	保留 读操作返回 0; 应写入 0

P0_PUDSEL

P0 口上拉/下拉选择寄存器

复位值: FF_H

7	6	5	4	3	2	1	0
0	P5	P4	P3	P2	P1		
r	rw	rw	rw	rw	rw		

符号	位序号	读写类型	功能描述
Pn (n = 0 - 5)	n	rw	P0 口引脚 n 上拉/下拉选择 0 选择下拉器件 1 选择上拉器件 (缺省值)
0	[7:6]	r	保留 读操作返回 0; 应写入 0

P0_PUDEN
P0 口上拉/下拉使能寄存器

 复位值: **C4_H**

7	6	5	4	3	2	1	0
0	P5	P4	P3	P2	P1	P0	P0
r	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n=0-5)	n	rw	P0 口引脚 n 上拉/下拉使能 0 禁止上拉或下拉 1 使能上拉或下拉 (缺省值)
0	[7:6]	r	保留 读操作返回 0; 应写入 0

P0_ALTSELn (n = 0 - 1)
P0 口其他功能选择寄存器

 复位值: **00_H**

7	6	5	4	3	2	1	0
0	P5	P4	P3	P2	P1	P0	P0
r	rw	rw	rw	rw	rw	rw	rw

表 6-5 位 P0_ALTSEL0.Pn 和位 P0_ALTSEL1.Pn 的功能

P0_ALTSEL0.Pn	P0_ALTSEL1.Pn	功能
0	0	正常 GPIO 功能
1	0	其他功能输出 1
0	1	其他功能输出 2
1	1	保留

6.4 P1 口

P1 口是一个 5 位通用双向口。P1 口寄存器总结见表 6-6。

表 6-6 P1 口寄存器

寄存器缩略名	寄存器完整名
P1_DATA	P1 口数据寄存器
P1_DIR	P1 口方向寄存器
P1_OD	P1 口开漏输出控制寄存器
P1_PUDSEL	P1 口上拉/下拉选择寄存器
P1_PUDEN	P1 口上拉/下拉使能寄存器
P1_ALTSEL0	P1 口其他功能选择寄存器 0
P1_ALTSEL1	P1 口其他功能选择寄存器 1

6.4.1 功能

表 6-7 P1 口 I/O 功能

端口引脚	I/O	功能选择	连接信号	来自/送至模块
P1.0	输入	GPI	P1_DATA.P0	-
		ALT1	RXD_0	UART
		ALT2	T2EX	定时器 2
		ALT3	-	-
	输出	GPO	P1_DATA.P0	-
		ALT1	-	-
ALT2		-	-	
P1.1	输入	GPI	P1_DATA.P1	-
		ALT1	-	-
		ALT2	EXINT3	外部中断 3

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
	输出	ALT3	-	-
		GPO	P1_DATA.P1	-
		ALT1	TDO_1	JTAG
		ALT2	TXD_0	UART
P1.5	输入	GPI	P1_DATA.P5	-
		ALT1	CCPOS0_1	CCU6
		ALT2	EXINT5	外部中断 5
		ALT3	-	-
	输出	GPO	P1_DATA.P5 ¹	-
		ALT1	-	-
		ALT2	-	-
P1.6	输入	GPI	P1_DATA.P6	-
		ALT1	CCPOS1_1	CCU6
		ALT2	T12HR_0	CCU6
		ALT3	EXINT6	外部中断 6
	输出	GPO	P1_DATA.P6 ²	-
		ALT1	-	-
		ALT2	-	-
P1.7	输入	GPI	P1_DATA.P7	-
		ALT1	CCPOS2_1	CCU6
		ALT2	T13HR_0	CCU6
		ALT3	-	-
	输出	GPO	P1_DATA.P7	-
		ALT1	-	-
		ALT2	-	-

¹⁾ P1.5 可用作 SSC 的软件片选功能

²⁾ P1.6 可用作 SSC 的软件片选功能

6.4.2 寄存器描述

P1_DATA

P1 口数据寄存器

复位值: 00H

7	6	5	4	3	2	1	0
P7	P6	P5	0		P1		P0
rw	rw	rw	r		rw		rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 1, 5 - 7)	n	rw	P1 口引脚 n 的数据值 0 P1 口引脚 n 的数据值 = 0 (缺省值) 1 P1 口引脚 n 的数据值 = 1
0	[4:2]	r	保留 读操作返回 0; 应写入 0

P1_DIR

P1 口方向寄存器

复位值: 00H

7	6	5	4	3	2	1	0
P7	P6	P5	0		P1		P0
rw	rw	rw	r		rw		rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 1, 5 - 7)	n	rw	P1 口引脚 n 方向控制 0 设置为输入引脚 (缺省值) 1 设置为输出引脚
0	[4:2]	r	保留 读操作返回 0; 应写入 0

P1_OD
P1 口开漏输出控制寄存器

 复位值: **00_H**

7	6	5	4	3	2	1	0
P7	P6	P5	0		P1		P0
rw	rw	rw	r		rw		rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 1, 5 - 7)	n	rw	P1 口引脚 n 开漏输出模式选择 0 正常模式; 0 和 1 可有效输出 (缺省值) 1 开漏输出模式; 只有 0 可有效输出
0	[4:2]	r	保留 读操作返回 0; 应写入 0

P1_PUDSEL
P1 口上拉/下拉选择寄存器

 复位值: **FF_H**

7	6	5	4	3	2	1	0
P7	P6	P5	0		P1		P0
rw	rw	rw	r		rw		rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 1, 5 - 7)	n	rw	P1 口引脚 n 上拉/下拉选择 0 选择下拉器件 1 选择上拉器件 (缺省值)
0	[4:2]	r	保留 读操作返回 0; 应写入 0

P1_PUDEN

P1 口上拉/下拉使能寄存器

复位值: **FF_H**

7	6	5	4	3	2	1	0
P7	P6	P5	0			P1	P0
rw	rw	rw	r			rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 1, 5 - 7)	n	rw	P1 口引脚 n 上拉/下拉使能 0 禁止上拉或下拉 1 使能上拉或下拉 (缺省值)

P1_ALTSELn (n = 0 - 1)

P1 口其他功能选择寄存器

复位值: **00_H**

7	6	5	4	3	2	1	0
P7	P6	P5	0			P1	P0
rw	rw	rw	r			rw	rw

表 6-8 位 P1_ALTSEL0.Pn 和位 P1_ALTSEL1.Pn 的功能

P1_ALTSEL0.Pn	P1_ALTSEL1.Pn	功能
0	0	正常 GPIO 功能
1	0	其他功能输出 1
0	1	其他功能输出 2
1	1	保留

6.5 P2 口

P2 口是一个 8 位通用单向输入口。P2 口寄存器总结见表 6-9。

表 6-9 P2 口寄存器

寄存器缩略名	寄存器完整名
P2_DATA	P2 口数据寄存器
P2_PUDSEL	P2 口上拉/下拉选择寄存器
P2_PUDEN	P2 口上拉/下拉使能寄存器

6.5.1 功能

表 6-10 P2 口 I/O 功能

端口引脚	I/O	功能选择	连接信号	来自/送至模块
P2.0	输入	GPI	P2_DATA.P0	-
		ALT1	CCPOS0_0	CCU6
		ALT2	EXINT1	外部中断 1
		ALT3	T12HR_2	CCU6
		ALT4	TCK_1	JTAG
		ANALOG	AN0	ADC
P2.1	输入	GPI	P2_DATA.P1	-
		ALT1	CCPOS1_0	CCU6
		ALT2	EXINT2	外部中断 2
		ALT3	T13HR_2	CCU6
		ALT4	TDI_1	JTAG
		ANALOG	AN1	ADC
P2.2	输入	GPI	P2_DATA.P2	-
		ALT1	CCPOS2_0	CCU6

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
		ALT2	-	-
		ALT3	$\overline{\text{CTRAP}}_1$	CCU6
		ALT4	-	-
		ANALOG	AN2	ADC
P2.3	输入	GPI	P2_DATA.P3	-
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
		ANALOG	AN3	ADC
P2.4	输入	GPI	P2_DATA.P4	-
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
		ANALOG	AN4	ADC
P2.5	输入	GPI	P2_DATA.P5	-
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
		ANALOG	AN5	ADC
P2.6	输入	GPI	P2_DATA.P6	-
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
		ANALOG	AN6	ADC
P2.7	输入	GPI	P2_DATA.P7	-
		ALT1	-	-

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
		ALT2	-	-
		ALT3	-	-
		ANALOG	AN7	ADC

6.5.2 寄存器描述

P2_DATA

P2 口数据寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
r	r	r	r	r	r	r	r

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	r	P2 口引脚 n 的数据值 0 P2 口引脚 n 的数据值 = 0 (缺省值) 1 P2 口引脚 n 的数据值 = 1

P2_PUDSEL

P2 口上拉/下拉选择寄存器

复位值: FF_H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	wr

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P2 口引脚 n 上拉/下拉选择 0 选择下拉器件 1 选择上拉器件

P2_PUDEN

P2 口上拉/下拉使能寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P2 口引脚 n 上拉/下拉使能 0 禁止上拉或下拉 (缺省值) 1 使能上拉或下拉

6.6 P3 口

P3 口是一个 8 位通用双向口。P3 口寄存器总结见表 6-11。

表 6-11 P3 口寄存器

寄存器缩略名	寄存器完整名
P3_DATA	P3 口数据寄存器
P3_DIR	P3 口方向寄存器
P3_OD	P3 口开漏输出控制寄存器
P3_PUDSEL	P3 口上拉/下拉选择寄存器
P3_PUDEN	P3 口上拉/下拉使能寄存器
P3_ALTSEL0	P3 口其他功能选择寄存器 0
P3_ALTSEL1	P3 口其他功能选择寄存器 1

6.6.1 功能

表 6-12 P3 口 I/O 功能

端口引脚	I/O	功能选择	连接信号	来自/送至模块
P3.0	输入	GPI	P3_DATA.P0	-
		ALT1	CC60_0	CCU6
		ALT2	CCPOS0_2	CCU6
		ALT3	-	-
	输出	GPO	P3_DATA.P0	-
		ALT1	CC60_0	CCU6
ALT2		-	-	
P3.1	输入	GPI	P3_DATA.P1	-
		ALT1	-	-
		ALT2	CCPOS1_2	CCU6

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
	输出	ALT3	-	-
		GPO	P3_DATA.P1	-
		ALT1	COU60_0	CCU6
		ALT2	-	-
P3.2	输入	GPI	P3_DATA.P2	-
		ALT1	CC61_0	CCU6
		ALT2	CCPOS2_2	CCU6
		ALT3	-	-
	输出	GPO	P3_DATA.P2	-
		ALT1	CC61_0	CCU6
		ALT2	-	-
P3.3	输入	GPI	P3_DATA.P3	-
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
	输出	GPO	P3_DATA.P3	-
		ALT1	COU61_0	CCU6
		ALT2	-	-
P3.4	输入	GPI	P3_DATA.P4	-
		ALT1	CC62_0	CCU6
		ALT2	-	-
		ALT3	-	-
	输出	GPO	P3_DATA.P4	-
		ALT1	CC62_0	CCU6
P3.5	输入	GPI	P3_DATA.P5	-

并行端口

端口引脚	I/O	功能选择	连接信号	来自/送至模块
		ALT1	-	-
		ALT2	-	-
		ALT3	-	-
	输出	GPO	P3_DATA.P5	-
		ALT1	COU62_0	CCU6
		ALT2	-	-
P3.6	输入	GPI	P3_DATA.P6	-
		ALT1	$\overline{\text{CTRAP}}_0$	CCU6
		ALT2	-	-
		ALT3	-	-
	输出	GPO	P3_DATA.P6	-
		ALT2	RSTOUT	内部复位
P3.7	输入	GPI	P3_DATA.P7	-
		ALT1	-	-
		ALT2	EXINT4	外部中断 4
		ALT3	-	-
	输出	GPO	P3_DATA.P7	-
		ALT1	COU63	CCU6
		ALT2	-	-

6.6.2 寄存器描述
P3_DATA
P3 口数据寄存器

复位值: 00H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P3 口引脚 n 的数据值 0 P3 口引脚 n 的数据值 = 0 (缺省值) 1 P3 口引脚 n 的数据值 = 1

P3_DIR
P3 口方向寄存器

复位值: 00H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P3 口引脚 n 方向控制位 0 设置为输入引脚(缺省值) 1 设置为输出引脚

P3_OD
P3 口开漏输出控制寄存器

 复位值: **00H**

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P3 口引脚 n 开漏输出模式选择 0 正常模式; 0 和 1 可有效输出 (缺省值) 1 开漏输出模式; 只有 0 可有效输出

P3_PUDSEL
P3 口上拉/下拉选择寄存器

 复位值: **BFH**

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P3 口引脚 n 上拉/下拉选择 0 选择下拉器件 1 选择上拉器件

P3_PUDEN

P3 口上拉/下拉使能寄存器

复位值: 40_H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
Pn (n = 0 - 7)	n	rw	P3 口引脚 n 上拉/下拉使能 0 禁止上拉或下拉 1 使能上拉或下拉

P3_ALTSELn (n = 0 - 1)

P3 口其他功能选择寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw

表 6-13 位 P3_ALTSEL0.Pn 和位 P3_ALTSEL1.Pn 的功能

P3_ALTSEL0.Pn	P3_ALTSEL1.Pn	功能
0	0	正常 GPIO 功能
1	0	其他功能输出 1
0	1	其他功能输出 2
1	1	保留

7 电源，复位和时钟管理

XC866 提供了一系列在重要条件下（如，电压下降）保证系统安全性能的特性。

CPU、存储器和外设的电压由嵌入式电压调节器（EVR）来调节；EVR 内部的检测电路确保供电电压在规定的工作范围之内。EVR 的主电压调节器和低功率电压调节器可分别被关闭，为不同的省电模式降低功耗。

XC866 时钟系统的核心是时钟产生单元（CGU），利用锁相环（PLL）和振荡器（OSC）单元产生主频。从主时钟得到的相位同步时钟信号被分配到整个系统中。时钟分频因子可编程设置，从而降低主频节省功耗。

7.1 内嵌电压调节器的电源系统

XC866 微控制器需要两种不同的电源供电：

- 嵌入式电压调节器（EVR）和端口需 3.3V 或 5.0V 供电
- CPU 核、存储器、片内振荡器和外设需 2.5V 供电

XC866 电源系统如图 7-1 所示。由外部电源引脚提供 3.3V 或 5.0V 电源；由 EVR 产生 2.5V 电源。EVR 有助于降低整个芯片的功耗及应用板的复杂度。

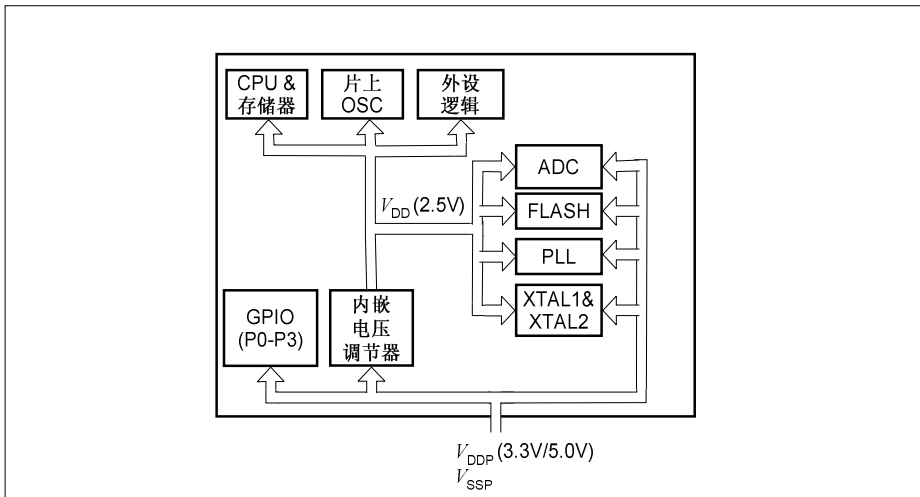


图 7-1 XC866 电源系统

EVR 特性:

- 输入电压 (V_{DDP}) : 3.3V/5.0V
- 输出电压 (V_{DD}) : 2.5V +/-7.5%
- 掉电模式下低功率电压调节器工作
- V_{DD} 和 V_{DDP} 预报警检测
- V_{DD} 压降检测

EVR 由一个主电压调节器和一个低功率电压调节器组成。正常工作模式下两个电压调节器均被使能。掉电模式下主电压调节器关闭，低功率电压调节器继续工作，在低功耗模式下为系统供电。

EVR 包含 V_{DD} 和 V_{DDP} 检测。有两种 V_{DD} 检测阈值电压：预报警电压 (2.3V) 和压降电压 (2.1V)。当 V_{DD} 低于 2.3V， V_{DD} 的 NMI 中断标志位 $NMISR.FNMIVDD$ 被置位；若 V_{DD} 的 NMI 中断使能 ($NMICON.NMIVDD$)，将向 CPU 发送 NMI 中断请求。当 V_{DD} 低于 2.1V，压降复位被激活，微控制器处于复位状态。

如果外接电源为 5.0V， V_{DDP} 只有一个 4.0V 的预报警阈值电压。当 V_{DDP} 低于 4.0V， V_{DDP} 的 NMI 中断标志位 $NMISR.FNMIVDDP$ 被置位；若 V_{DDP} 的 NMI 中断使能 ($NMICON.NMIVDDP$)，将向 CPU 发送 NMI 中断请求。

如果外接电源为 3.3V，用户必须将 $NMICON.NMIVDDP$ 清零来禁止 V_{DDP} 检测。掉电模式下， V_{DD} 检测关闭， V_{DDP} 检测继续工作。

EVR 还有一个 V_{DD} 上电复位 (POR) 检测，以保证系统正确上电。POR 的检测电压为 1.6V。在正常工作模式和掉电模式下均采用监控功能。上电过程中， V_{DD} 超过 1.6V 后，EVR 复位通常需再延长 300 μ s。正常工作模式下，主要由 V_{DD} 检测监控 V_{DD} ， V_{DD} 跌至 2.1V 以下时产生复位。掉电模式下，由 POR 监控 V_{DD} ， V_{DD} 低于 1.6V 时产生复位。

7.2 复位控制

XC866 有五种复位方式：上电复位、硬件复位、WDT 复位，掉电唤醒复位和压降复位。

XC866 首次上电时，必须定义某些引脚（见表 7-2）的状态以使器件能够正确启动。复位结束时，引脚采样值被锁存以选择所要的启动方式，直到下次上电复位或硬件复位时才能修改该值，从而保证了器件正常工作时状态稳定。

正常工作模式或掉电模式下可硬件复位。复位输入引脚 $\overline{\text{RESET}}$ 用于硬件复位。

检测到系统出现故障时，WDT 也能复位器件。

另一种需被检测的复位是器件掉电模式下的复位（唤醒复位）。上电复位时静态 RAM 的内容未定义；而掉电唤醒复位后，静态 RAM 的内容完好保留。

V_{DD} 电压跌至 2.1V 以下时触发压降复位。

7.2.1 复位类型

7.2.1.1 上电复位

电源电压 V_{DDP} 用于芯片上电。上电后首先复位 EVR 模块，包括：

1. 启动主电压调节器和低功率电压调节器
2. 当 V_{DD} 和 V_{DDP} 达到 V_{DD} 和 V_{DDP} 的检测阈值电压时，EVR 复位无效

系统启动时，PLL 和 OSC 断开，PLL 工作在基频。EVR 稳定之后，（假如 OSC 工作）PLL 和 OSC 连接，紧接着锁相检测、以确保 PLL 开始正常工作。接下来系统时钟一稳定，每个 4K 字节的 Flash bank 就会进入待读模式。

复位将锁存引脚 MBC、TMS 和 P0.0 的状态值，该值用来选择启动模式（见章节 7.2.3）。正确的复位将引导系统进入指定状态。从地址 0000_H 开始执行程序。

上电复位顺序如图 7-2 所示。

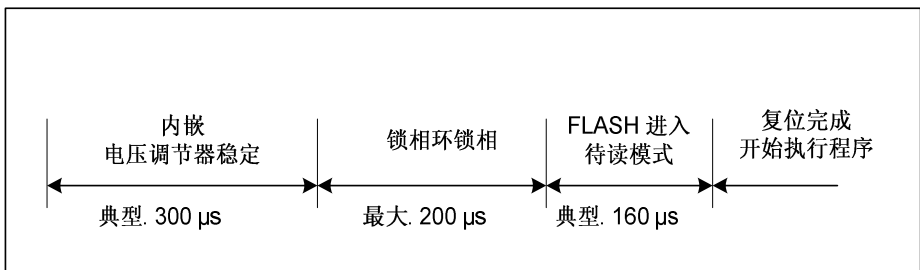


图 7-2 上电复位

7.2.1.2 硬件复位

当复位输入引脚 $\overline{\text{RESET}}$ 拉低时，开始执行外部硬件复位序列。引脚 $\overline{\text{RESET}}$ 必须拉低至少 1ms。 $\overline{\text{RESET}}$ 信号解除后，复位顺序和上电复位顺序相同，如图 7-2 所示。硬件复位将终止空闲模式或掉电模式。

复位将锁存引脚 MBC、TMS 和 P0.0 的状态值，该值用来选择启动模式（见章节 7.2.3）。

7.2.1.3 看门狗定时器复位

看门狗定时器复位是一种内部复位。WDT 包含一个必须周期刷新或清零的计数器。如果 WDT 没有被及时、正确刷新，它将向 CPU 发送 NMI 中断请求，在预先定义的超时周期之后系统复位。位 PMCON0.WDTRST 用来指示 WDT 的复位状态。

对 WDT 复位而言，因为 EVR 已稳定且不需要 PLL 锁相检测，WDT 复位时间大约 200 μs ，比其他复位时间短。

7.2.1.4 掉电唤醒复位

掉电模式下低功率电压调节器仍然工作，XC866 仍有供电。如果恰当的进入掉电模式，所有重要的系统状态将由软件保留在 Flash 中。

如果 XC866 处于掉电模式，有三种唤醒方式：

- 通过 RXD 唤醒
- 通过 EXINT0 唤醒
- 通过 RXD 或 EXINT0 唤醒

由控制位 PMCON0.WS 选择不同的唤醒方式。掉电唤醒可以有复位唤醒或无复位唤醒，由 PMCON0.WKSEL 选择。由位 PMCON0.WKRS 指示（有复位或无复位的）唤醒状态。

掉电唤醒复位序列如图 7-3 所示。EVR 大概需要 150 μs 达到稳定，比上电复位所需时间短。

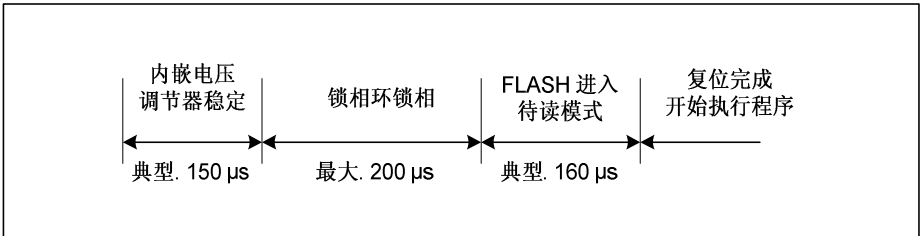


图 7-3 掉电唤醒复位

除以上三种唤醒方式外，还可以通过 $\overline{\text{RESET}}$ 硬件复位退出掉电模式。

7.2.1.5 压降复位

正常工作模式下，当内核供电电压 V_{DD} 跌至阈值电压 V_{DD_TH} (2.1V) 以下时，EVR 中的 V_{DD} 检测检测到电压下降，该压降将引起器件复位。掉电模式下， V_{DD} 由 EVR 中的 POR 监控， V_{DD} 跌至 1.6V 以下产生复位。

一旦发生压降复位，复位顺序和上电复位顺序相同，如图 7-2 所示。

7.2.2 模块复位行为

表 7-1 示出不同的复位类型如何影响 XC866 的功能。“■”表示该功能被复位为其缺省值。

表 7-1 复位对器件功能的影响

模块 / 功能	唤醒复位	WDT 复位	硬件复位	上电复位	压降复位
CPU 核	■	■	■	■	■
外设	■	■	■	■	■
片内静态 RAM	不受影响, 可靠	不受影响, 可靠	不受影响, 可靠	受影响, 不可靠	受影响, 不可靠
OSC, PLL	■	不受影响	■	■	■
端口引脚	■	■	■	■	■
EVR	电压调节器 开启	不受影响	■	■	■
FLASH	■	■	■	■	■
NMI	■	禁止	禁止	■	■

7.2.3 启动方案

XC866 复位时, 系统必须识别 (一旦复位结束) 启动不同工作模式的配置类型。因此, 激活特殊模式和状态所需的启动信息要由外部输入引脚来设置。上电复位或硬件复位后, 引脚 MBC、TMC 和 P0.0 共同选择不同的启动模式。表 7-2 示出 XC866 系统的启动模式选择。

表 7-2 XC866 启动选择

MBC	TMS	P0.0	模式类型	程序计数器初始值
1	x	x	用户模式: OSC/PLL 未旁路	0000 _H
0	0	x	BSL 模式: OSC/PLL 未旁路	0000 _H
0	1	0	OCDS 模式: OSC/PLL 未旁路	0000 _H


7.2.4 寄存器描述

PMCON0

功率模式控制寄存器 0

复位值: 见表 7-3

7	6	5	4	3	2	1	0
0	WDTRST	WKRS	WKSEL	SD	PD	WS	
r	rwh	rwh	rw	rw	rwh	rw	

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
WS	[1:0]	rw	唤醒源选择 00 无唤醒源 01 选择唤醒源 RXD（下降沿触发） 10 选择唤醒源 EXINT0（下降沿触发） 11 选择唤醒源 RXD（下降沿触发）或者 EXINT0（下降沿触发）
WKSEL	4	rw	唤醒复位选择位 0 无复位唤醒 1 有复位唤醒
WKRS	5	rwh	唤醒指示位 0 未发生唤醒 1 已唤醒 该位只能由硬件置位，软件清零
WDTRST	6	rwh	WDT 复位指示位 0 未发生 WDT 复位 1 WDT 已复位 该位仅能由硬件置位，软件清零
0	7	r	保留 读操作返回 0；应写入 0

表 7-3 寄存器 PMCON0 的复位值

复位源	复位值
上电复位/硬件复位/压降复位	0000 0000 _B
WDT 复位	0100 0000 _B
掉电唤醒复位	0010 0000 _B

7.3 时钟系统

XC866 的时钟系统执行以下功能：

- 获取并缓冲输入的时钟信号，产生主时钟
- 将相位同步的时钟信号分配到整个 XC866 系统中
- 将系统主时钟分频（到较低频率）用于省电模式

7.3.1 时钟产生单元

XC866 系统的时钟产生单元（CGU）由振荡器电路和 PLL 构成。XC866 的振荡器可以是片内（10 MHz）、也可以是片外振荡器（3 MHz 至 12 MHz）。如果不特殊指明，“振荡器”统指片内和片外振荡器。复位后，缺省使用片内振荡器；可由软件编程选择片外振荡器。PLL 能够将来自振荡器的低频时钟信号倍频成内部高速时钟，以提高系统性能。

CGU 的方框图如图 7-4 所示。

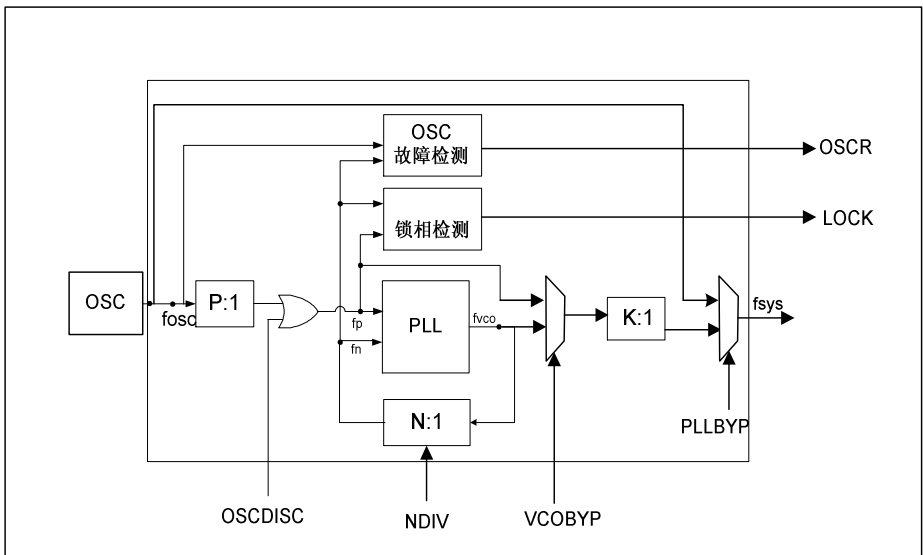


图 7-4 CGU 方框图

7.3.1.1 功能描述

系统启动时，PLL 和 OSC 断开，工作在压控振荡器（VCO）的基频。EVR 稳定之后，（假如 OSC 工作）PLL 和 OSC 连接，紧接着锁相检测、以确保 PLL 开始正常工作。复位结束后，若 OSC 在工作 OSCR 将被置 1；若 PLL 已锁相 LOCK 将被置 1。

失锁操作

若检测到 PLL 失锁，则不用 PLL 作系统时钟源（VCOBYP=1），只复位锁相标志（PLL_CON.LOCK=0），无其他操作，从而允许动态切换 PLL 的参数。

如果 PLL 与 OSC 失锁，置位 PLL 失锁 NMI 中断标志 NMISR.FNMIPLL；若 PLL NMI 被使能，将向 CPU 发送 NMI 中断请求。另外，复位 PLL_CON 中的 LOCK 标志。PLL 的 VCO 逐渐减速至基频。紧急事件可以在该基频时钟下处理。

在下次上电复位、硬件复位或成功执行锁相恢复之前，XC866 一直保持失锁状态。

失锁恢复

若 PLL 失锁，可通过软件重新锁相，必须执行下列步骤：

1. 将 OSC 和 PLL 断开（OSCDISC=1）
2. 将 PLL 的分频值 N 设为 16（PLL_CON.NDIV=0010_B）
3. 等待 50μs 直到 OSC 稳定
4. 置位 OSC_CON.ORDRES，重新启动 OSC 运行检测

如果 OSC_CON.OSCR 置位，则：

1. 选择 VCO 旁路模式（VCOBYP=1）
2. 重新将 OSC 连接到 PLL 上（OSCDISC=0）
3. 将 NDIV 因子重新设回原先的值
4. 必须置位 RESLD，然后检查 LOCK 标志。只有 LOCK 重新置位后才能退出 VCO 旁路模式，继续执行正常操作

如果 OSCR 和 LOCK 均未置位，必须进行紧急事件处理，例如由用户关闭系统。

改变 PLL 的参数

要改变 PLL 的参数，首先检查 OSC 是否在工作（OSC_CON.OSCR=1）。在这种情况下：

1. 选择 VCO 旁路模式（VCOBYP=1）
2. 将 OSC 连接到 PLL 上（OSCDISC=0）

3. 设置需要的 NDIV 分频值
4. 直到 LOCK 位被置位
5. 关闭 VCO 旁路模式

选择片外 OSC

选择片外 OSC 时，必须执行下列步骤：

1. 选择 VCO 旁路模式 (VCOBYP=1)
2. 将 OSC 和 PLL 断开 (OSCDISC=1)
3. 复位 XPD 对片外 OSC 上电
4. 置位 OSCSS，选择片外 OSC
5. 等待 1.5ms（根据片外 OSC 不同应相应调整延迟时间）直到片外 OSC 稳定
6. 置位 OSC_CON.ORDRES，重新启动 OSC 运行检测

如果 OSC_CON.OSCR 置位，则：

1. 选择 VCO 旁路模式 (VCOBYP=1)
2. 重新将 OSC 连接到 PLL 上 (OSCDISC=0)
3. 将 NDIV 因子重新设回原先的
4. 必须置位 RESLD，然后检查 LOCK 标志。只有 LOCK 重新置位后才能退出 VCO 旁路模式，继续正常操作

当使用片内 OSC 时，为了降低功耗，置位 XPD 可使 XTAL 掉电；但使用片外 OSC 时，置位 OSCPD 不会使片内 OSC 掉电。

7.3.2 时钟源控制

时钟系统提供了四种方式产生 CPU 时钟：

直接驱动 (PLL 旁路)

PLL 旁路时，系统时钟频率和外部时钟频率完全相同。XC866 中 PLL 旁路方式无效。

$$f_{sys} = f_{osc}$$

PLL 基频模式

系统时钟来自于 VCO 的基频除以因子 K。该模式下 VCO 旁路和 PLL 旁路必须无效。

$$f_{SYS} = \frac{1}{K} \times f_{VCObase}$$

预分频模式（VCO 旁路）

VCO 旁路时，系统时钟来自于振荡器时钟除以因子 P 和 K。

$$f_{SYS} = \frac{1}{P \times K} \times f_{OSC}$$

PLL 模式

系统时钟来自于振荡器时钟除以因子 P、乘以 N，再除以 K。该模式下 VCO 旁路和 PLL 旁路均必须无效。

$$f_{SYS} = \frac{N}{P \times K} \times f_{OSC}$$

正常工作时，系统工作在 PLL 模式。

针对不同的振荡器，典型输出频率 f_{SYS} 80MHz 所对应的参数选择如表 7-4 所示。

表 7-4 系统频率 (f_{SYS} 80MHz)

振荡器	f_{osc}	N	P	K	f_{sys}
片内	10 MHz	16	1	2	80 MHz
片外	10 MHz	16	1	2	80 MHz
	8 MHz	20	1	2	80 MHz
	5 MHz	32	1	2	80 MHz

XC866 系统中 P 值固定为“1”、K 值固定为“2”。为了获得需要的 f_{SYS} ，针对不同的振荡器输入，由位 NDIV 选择 N 值。见表 7-4，输出频率要在 75 MHz 到 80 MHz 之间。

表 7-5 示出 XC866 中 VCO 的频率范围。

表 7-5 VCO 频率范围

fVCOmin	fVCOmax	fVCOFREEmin	fVCOFREEmax	单位
150	200	40	130	MHz

7.3.3 时钟管理

时钟管理子模块从基本时钟产生出系统需要的所有时钟信号，包括：

- 基本的时钟减速电路
- 集中化时钟控制使能/禁止电路

图 7-5 示出从系统频率 f_{sys} 产生的时钟。正常工作模式下，不同模块的典型频率如下：

- CPU 时钟：CCLK, SCLK = 26.7 MHz
- CCU 时钟：FCLK = 26.7 MHz
- 其它外设：PCLK = 26.7 MHz
- Flash 接口时钟：CCLK3 = 80 MHz 和 CCLK = 26.7 MHz

振荡器时钟输出到引脚 CLKOUT (P0.0) 上。空闲模式下只有 CPU 时钟 CCLK 被关闭；掉电模式下 CCLK, SCLK, FCLK, CCLK3 全部关闭；若选择时钟减速模式，CPU 时钟及外设时钟将被分频，由位域 CMCON.CLKREL 选择可编程的分频因子。

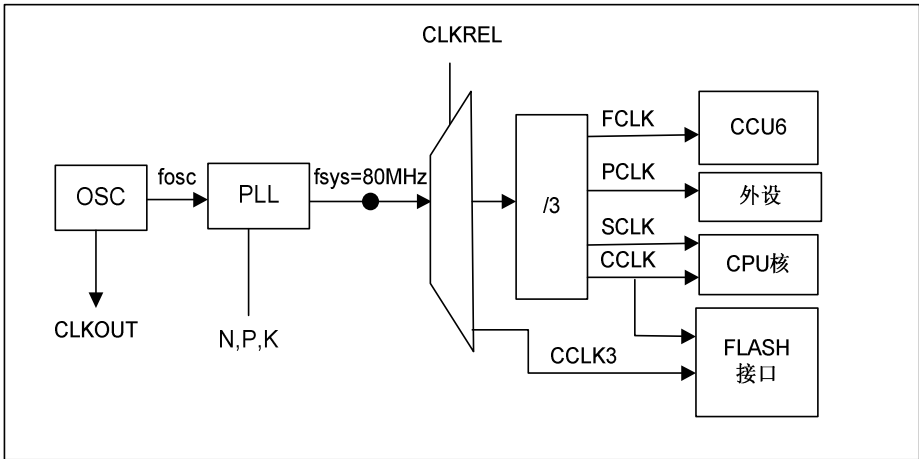


图 7-5 由 f_{sys} 产生的时钟

7.3.4 寄存器描述
OSC_CON
OSC 控制寄存器
复位值: 0000 1000_B

7	6	5	4	3	2	1	0
0		OSCPD	XPD	OSCSS	ORDRES	OSCR	
		rw	rw	rw	rwh	rh	

符号	位序号	读写类型	功能描述
OSCR	0	rh	OSC 运行状态位 该位指示检测到的 OSC 的工作状态 0 OSC 未工作 1 OSC 正在工作
ORDRES	1	rwh	OSC 运行检测复位 0 无操作 1 复位并重新启动 OSC 运行检测 该位将自动复位为 0。
OSCSS	2	rw	OSC 振荡源选择 0 选择片内 OSC 1 选择片外 OSC
XPD	3	rw	XTAL 掉电控制 0 XTAL 未掉电 1 XTAL 掉电
OSCPD	4	rw	片内 OSC 掉电控制 0 片内 OSC 未掉电 1 片内 OSC 掉电
0	[7:5]	r	保留 读操作返回 0；应写入 0

注：寄存器 OSC_CON 的复位值是 00001000_B。复位后过一个时钟周期如果振荡器工作，OSCR 将被置 1，于是会观察到 OSC_CON 的值为 00001001_B。

PLL_CON

PLL 控制寄存器

复位值: 0010 0000_B

7	6	5	4	3	2	1	0
NDIV				VCOPY P	OSCDIS C	RESLD	LOCK
rw				rw	rw	rwh	rh

符号	位序号	读写类型	功能描述
LOCK	0	rh	<p>PLL 锁相状态标志</p> <p>0 PLL 相位未锁定</p> <p>1 PLL 相位锁定</p>
RESLD	1	rwh	<p>重新启动锁相检测</p> <p>该位被置位将复位 PLL 锁相状态标志、并重启锁相检测。该位将自动复位为零，因此读取值始终为 0</p> <p>0 无作用</p> <p>1 复位锁相标志并重启锁相检测</p>
OSCDISC	2	rw	<p>OSC 断开</p> <p>0 OSC 连接到 PLL 上</p> <p>1 OSC 和 PLL 断开</p>
VCOPYP	3	rw	<p>PLL VCO 旁路模式选择</p> <p>0 正常模式（缺省状态）</p> <p>1 VCO 旁路模式（PLL 输出时钟来自振荡器时钟除以分频因子 P 和 K）</p>

符号	位序号	读写类型	功能描述
NDIV	[7:4]	rw	<p>PLL N 分频</p> <p>0000_B N = 14</p> <p>0001_B N = 15</p> <p>0010_B N = 16</p> <p>0011_B N = 17</p> <p>0100_B N = 18</p> <p>0101_B N = 19</p> <p>0110_B N = 20</p> <p>0111_B N = 21</p> <p>1000_B N = 24</p> <p>1001_B N = 28</p> <p>1010_B N = 30</p> <p>1011_B N = 32</p> <p>1100_B N = 40</p> <p>1101_B N = 42</p> <p>1110_B N = 45</p> <p>1111_B N = 50</p> <p>NDIV 是被保护位。当激活保护方案（见章节 3.3.4.1）该位不能直接写入。</p>

注：寄存器 PLL_CON 的复位值是 **00100000_B**。复位后过一个时钟周期如果 PLL 锁相，LOCK 将被置 1，于是会观察到 PLL_CON 的值为 **00100001_B**。

CMCON

时钟控制寄存器

复位值: 00_H

	7	6	5	4	3	2	1	0
0					CLKREL			
r					rw			

符号	位序号	读写类型	功能描述
CLKREL	[3:0]	rw	时钟分频 0000 _B $f_{sys} / 1$ 0001 _B $f_{sys} / 2$ 0010 _B $f_{sys} / 4$ 0011 _B $f_{sys} / 8$ 0100 _B $f_{sys} / 16$ 0101 _B $f_{sys} / 32$ 0110 _B $f_{sys} / 64$ 0111 _B $f_{sys} / 128$ 1000 _B $f_{sys} / 256$ 1001 _B $f_{sys} / 512$ 1010 _B $f_{sys} / 1024$ 1011 _B $f_{sys} / 2048$ 1100 _B 保留 1101 _B 保留 1110 _B 保留 1111 _B 保留
0	[7:4]	r	保留 读操作返回 0; 应写入 0

注: 寄存器 *OSC_CON*, *PLL_CON* 和 *CMCON* 在 *WDT* 复位期间不被复位。

8 省电模式

通过以下方法的组合，XC866 的省电模式为系统提供了灵活的功耗：

- 终止 CPU 时钟
- 终止系统各单元时钟
- 降低某些外设单元的时钟频率
- 具有快速重启能力的系统掉电

复位后，缺省（见图 8-1）选择进入有效模式（正常工作模式），系统以主频运行。可由软件选择从有效模式进入不同的省电模式，包括：

- 空闲模式
- 减速模式
- 掉电模式

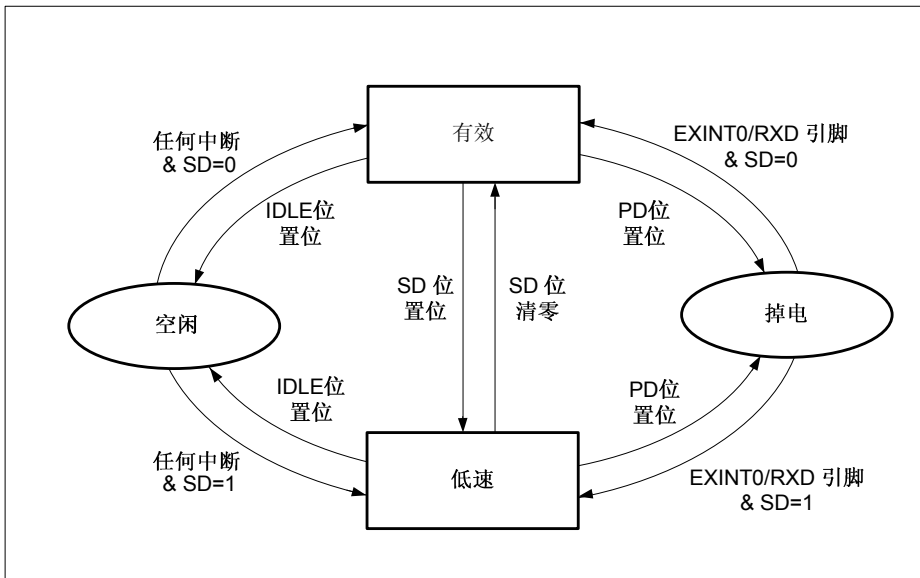


图 8-1 省电模式之间的转换

8.1 功能描述

本节描述了不同的省电模式，模式操作以及如何进入和退出这些模式。

8.1.1 空闲模式

空闲模式通过终止 CPU 时钟降低功耗。

空闲模式下，振荡器继续工作，但 CPU 时钟关闭，CPU 停止工作。输入时钟未关闭的外设仍继续工作。系统进入空闲模式前用户应关闭 WDT；否则 WDT 在溢出时将产生内部复位，从而中断了空闲模式。CPU 的状态被完整保存：堆栈指针、程序计数器、程序状态字、累加器，所有其他寄存器均保留进入空闲模式之前的数据；端口引脚保持空闲模式被激活前的逻辑状态。

PCON.IDLE 置 1 进入空闲模式。

下列任何一个条件发生时系统将返回有效模式：

- 激活任何被使能的中断可结束空闲模式。CPU 恢复运行，响应中断。执行完 RETI 指令，CPU 将返回执行将 IDLE 置 1 指令之后的下一条指令。
- 外部硬件复位信号（RESET）被拉低

8.1.2 减速模式

减速模式通过降低器件内部时钟降低功耗。

SFR PMCON0 中的 SD 置 1 将激活减速模式。位域 CMCON.CLKREL 用来选择不同的低速频率。CPU 和外设均以该低频时钟工作。对 SD 清零结束减速模式。

执行以下步骤，减速模式可以和空闲模式组合使用：

1. 置位 PMCON0.SD 激活减速模式
2. 置位 PCON.IDLE 激活空闲模式

有两种方式可结束该组合模式：

- 激活任何被使能的中断可结束空闲模式。CPU 恢复运行，响应中断。执行完 RETI 指令，将执行 IDEL 置 1 指令之后的下一条指令。但减速模式仍保持使能。如果要结束减速模式，必须在相应的中断服务程序中对 SD 清零，或在程序中任何不再需要该模式处对 SD 清零。
- 硬件复位也可结束该组合模式。

8.1.3 掉电模式

掉电模式下，OSC 和 PLL 均关闭，Flash 被带入掉电模式。主电压调节器关闭，但低功率电压调节器继续工作。因此，微控制器的所有功能均被终止，只有 Flash、片内 RAM、XRAM 和 SFR 的内容被保存。端口引脚保持掉电模式被激活前的逻辑状态；对于数字端口，用户必须注意，掉电模式下端口不可处于悬浮状态，可通过外部上拉/下拉或将端口设置为输出来实现。

掉电模式下时钟关闭，故不能由中断或 WDT 唤醒；只有接收到外部唤醒信号或复位信号时才能掉电唤醒。

进入掉电模式

PMCON0.PD 置 1 进入掉电模式。

如果采用外部掉电唤醒，进入掉电模式之前软件必须把 XC866 的外部环境准备好，从而在适当条件下触发这些信号。唤醒电路用来检测唤醒信号，激活系统上电。掉电过程该电路始终保持有效，和任何时钟无关。下列外部引脚的下降沿可触发退出掉电模式：

- 引脚 EXINT0
- 引脚 RXD
- 引脚 RXD 或 EXINT0

由 PMCON0.WS 选择唤醒源；由 PMCON0.WKSEL 选择有复位唤醒或无复位唤醒。系统进入掉电模式之前必须选择唤醒源和唤醒类型。

退出掉电模式

如果采用硬件复位退出掉电模式，器件处于硬件复位状态。

如果系统在进入掉电模式前已选择了唤醒源和唤醒类型，可由 EXINT0/RXD 引脚退出掉电模式。

由位 MODPISEL.URRIS 从两个 RXD 输入中选择一个；由位 MODPISEL.EXINT0IS 从两个 EXINT0 输入中选择一个。

如果进入掉电模式前 WKSEL 已置 1，系统将执行和上电复位类似的复位序列。因此，所有端口引脚被置为复位状态，该状态一直保持到执行程序改变引脚值。

如果进入掉电模式前 WKSEL 已清 0，系统将执行快速唤醒序列。端口引脚保持掉电模式时的有效状态，直到执行程序改变引脚值。

无复位掉电唤醒执行下列步骤：

1. 掉电模式下，EXINT0/RXD 引脚必须保持高电平。
2. 当 EXINT0/RXD 引脚拉低至少 100ns 后退出掉电模式。
3. 开启主电压调节器，大约 150 μ s 达到稳定。

4. 启动片内 OSC 和 PLL。通常片内 OSC 大约需要 500ns 达到稳定；检测到片内 OSC 时钟稳定到标称频率后 PLL 在 200 μ s 内达到稳定。
5. 接下来 Flash 将进入待读状态。这里不像正常复位通常需要 160 μ s，该部分的时间可忽略不计。
6. CPU 恢复运行。若唤醒源是 EXINT0，进入掉电模式前如果 EXINT0 被使能，则响应中断。执行完 RETI 指令，CPU 将返回执行将 PD 置 1 指令之后的下一条指令。若唤醒源是 RXD，CPU 返回执行 PD 置 1 指令之后的下一条指令。

8.1.4 外设时钟管理

功耗降低的程度取决于哪些外设在工作（外设个数）。实现某特定功能时，不必要的外设可通过门控电路关闭其输入时钟将其禁止。如空闲模式下，如果所有定时器被停止、ADC、CCU6 和串行接口均不工作，则最大程度降低了功耗。但是，用户必须小心决定在有效模式和空闲模式下，哪些外设应继续工作，哪些外设应被停止。

置位寄存器 PMCON1 中的相应位，可禁止（关闭时钟）外设 ADC、SSC、CCU6 和定时器 2。另外，复位 GLOBCTR.ANON 可禁止 ADC 的模拟部分。该特性将终止 f_{ADCl} 时钟产生，不需进行模数转换时功耗降低。

当使用片内 OSC 时，为了降低功耗，置位 OSC_CON.XPD 可使 XTAL 掉电；但使用片外 OSC 时，置位 OSC_CON.OSCPD 不会使片内 OSC 掉电。


8.2 寄存器描述

PMCON0

功率模式控制寄存器 0

复位值:见表 8-1

7	6	5	4	3	2	1	0
0	WDTRST	WKRS	WKSEL	SD	PD	WS	
r	rwh	rwh	rw	rw	rwh	rw	

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
WS	[1:0]	rw	唤醒源选择 00 无唤醒源 01 选择唤醒源 RXD（下降沿触发） 10 选择唤醒源 EXINT0（下降沿触发） 11 选择唤醒源 RXD（下降沿触发）或者 EXINT0（下降沿触发）
PD	2	rwh	掉电使能，高有效 该位置位芯片将进入掉电模式，由唤醒电路复位。PD 位是被保护位。当保护方案（见章节 3.3.4.1）有效时，该位不能直接写入。
SD	3	rw	减速使能，高有效 该位置位芯片将进入减速模式，由用户复位。SD 位是被保护位。当保护方案（见章节 3.3.4.1）有效时，该位不能直接写入。
WKSEL	4	rw	唤醒复位选择 0 无复位唤醒 1 有复位唤醒
WKRS	5	rwh	唤醒指示 0 未发生唤醒

符号	位序号	读写类型	功能描述
			1 已唤醒 该位只能由硬件置位，软件清零
0	7	r	保留 读操作返回 0；应写入 0

表 8-1 寄存器 PMCON0 的复位值

复位源	复位值
上电复位/硬件复位/压降复位	0000 0000 _B
WDT 复位	0100 0000 _B
掉电唤醒复位	0010 0000 _B

PCON

功率控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
SMOD	0			GF1	GF0	0	IDLE
rw	r			rw	rw	r	rw

阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
IDLE	[1:0]	rw	空闲模式使能 0 未进入空闲模式 1 进入空闲模式

MODPSEL

外设输入选择寄存器

复位值: 00H

7	6	5	4	3	2	1	0
0	JTAGD IS		JTAGC KS		0	EXINT0 IS	URRIS
r	rw		rw		r	rw	rw

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
URRIS	0	rw	UART 接收输入选择 0 选择 UART 接收输入 RXD_0 1 选择 UART 接收输入 RXD_1
EXINT0IS	1	rw	外部中断 0 输入选择 0 选择外部中断输入 EXINT0_0 1 选择外部中断输入 EXINT0_1
0	[3:2] [7:6]	r	保留 读操作返回 0; 应写入 0

PMCON1

功率模式控制寄存器 1

复位值: 00H

7	6	5	4	3	2	1	0
0				T2_DIS	CCU_ DIS	SSC_ DIS	ADC_ DIS
r				rw	rw	rw	rw

符号	位序号	读写类型	功能描述
ADC_DIS	0	rw	ADC 禁止请求, 高有效 0 ADC 正常工作 (缺省状态) 1 ADC 被关闭

符号	位序号	读写类型	功能描述
SSC_DIS	1	rw	SSC 禁止请求位, 高有效 0 SSC 正常工作 (缺省状态) 1 SSC 被关闭
CCU_DIS	2	rw	CCU6 禁止请求位, 高有效 0 CCU6 正常工作 (缺省状态) 1 CCU6 被关闭
T2_DIS	3	rw	定时器 2 禁止请求位, 高有效 0 定时器 2 正常工作 (缺省状态) 1 定时器 2 被关闭
0	[7:4]	r	保留 读操作返回 0; 应写入 0

ADC_GLOBCTR

全局控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
ANON	DW	CTC		0			
rw	rw	rw				r	



阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
ANON	7	rw	模拟部分开启控制 该位使能 ADC 的模拟部分并定义其工作模式 0 模拟部分关闭, 不能进行模数转换。为达到功耗最小, 内部模拟电路工作在掉电模式, 终止 f _{ADCl} 时钟产生。 1 模拟部分开启, 可进行模数转换。模拟


			部分的自动掉电功能被禁止。
0	[3:0]	r	保留 读操作返回 0；应写入 0

OSC_CON

OSC 控制寄存器

复位值: 0000 1000_B

7	6	5	4	3	2	1	0
0			OSCPD	XPD	OSCSS	ORDRE S	OSCR
r			rw	rw	rw	rwh	rh

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
XPD	3	rw	XTAL 掉电控制 0 XTAL 未掉电 1 XTAL 掉电
OSCPD	4	rw	片内 OSC 掉电控制 0 片内 OSC 未掉电 1 片内 OSC 掉电
0	[7:5]	r	保留 读操作返回 0；应写入 0

9 看门狗定时器

看门狗定时器 (WDT) 为检测硬件故障以及故障恢复提供了高度可靠和安全的方式。要在用户预先定义的时间内复位 WDT；CPU 必须在这个时间间隔内刷新 WDT 以防止 XC866 系统复位。因此，WDT 服务程序确保系统能够正常运行；确保系统能够在用户规定的时间周期内退出偶然出错状况。

特性:

- 16 位 WDT
- 定时器高 8 位的重载值可编程设定
- 窗界可编程设定
- 输入频率可选择: $f_{PCLK}/2$ 或 $f_{PCLK}/128$

9.1 功能描述

WDT 是以频率 $f_{PCLK}/2$ 或 $f_{PCLK}/128$ 递增计数的 16 位定时器，由两个 8 位定时器串联组成。通过服务 WDT 可将定时器高 8 位预置为用户规定的值，从而修改 WDT 的失效时间；每次服务 WDT 将复位定时器低 8 位。WDT 单元的方框图如图 9-1 所示。

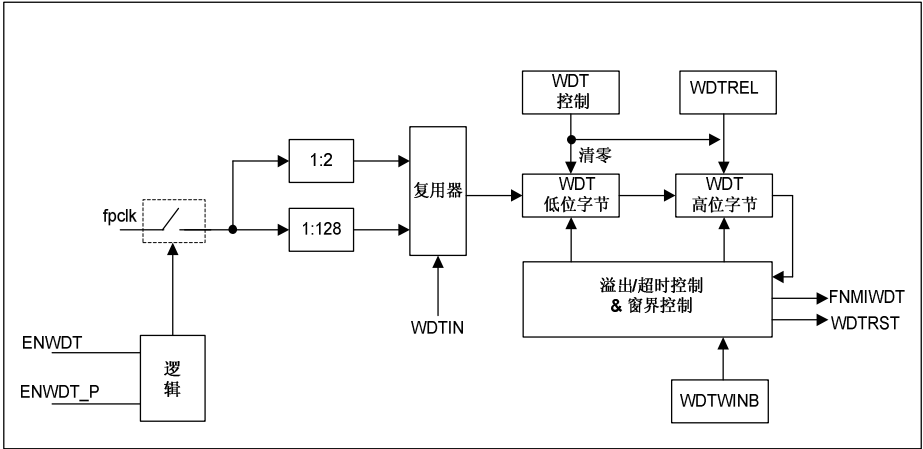


图 9-1 WDT 方框图

如果将 **WDTEN** 置 1 使能 WDT，定时器将从用户定义的初始值开始递增计数。CPU 必须在计数器溢出前服务 WDT，服务即执行刷新操作 (**WDTRS** 置 1)。刷新将把用户定义的初始值重新载入定时器，继续正常工作。

如果直到定时器溢出时 WDT 也未被刷新，则认为系统出错、正常运行模式被终止。产生 WDT NMI 请求 (**FNMWDT**)，进入预报警模式。预报警持续 30_H 个计数周期。在预报警阶段，不响应 WDT 刷新操作，也不能关闭 WDT，进而进行系统复位 (**WDTRST**) 且复位不能被终止。由位 **WDTRST** 指示 WDT 产生复位；一旦硬件检测到 **WDTRST** 信号被拉高，即对 **WDTRST** 置 1。如果刷新和溢出同时发生，WDT 将不会进入预报警阶段。

必须周期刷新 WDT 以保证定时器不会溢出。刷新定时器将对其低位字节清零，并将位域 **WDTREL** 中预先设置的值重新载入高位字节。刷新 WDT 时将 **WDTRS** 清零。

WDT 有一个在定时器计数过程中不允许刷新的“可编程窗界”。在窗界内刷新 WDT 无效；虽不产生 NMI 请求，但 WDT 会激活 **WDTRST**。窗界取值在 0000_H 和 **WDTWINB** 和 00_H 串联值之间。可由 **WINBEN** 使能该特性。

WDT 被刷新后，定时器从 $\langle \text{WDTREL} \rangle * 2^8$ 开始继续计数。WDT 的溢出周期有两种方式设定：

看门狗定时器

- 输入频率可由寄存器 WINCON 中 WDTIN 选择: $f_{PCLK}/2$ 或 $f_{PCLK}/128$ 。
- WDT 的高位字节重载值 WDTREL 可由寄存器 WDTREL 编程设定。

从 WDT 被刷新到下次溢出之间的溢出周期 P_{WDT} 可由下面的公式决定:

$$P_{WDT} = \frac{2^{1+(DTIN \times 2)} \times (WDTREL + 1)}{f_{PCLK}}$$

如果 WDT 的窗界刷新特性被使能, 若 WDTWINB 大于 WDTREL, 则溢出周期 P_{WDT} 缩短, 见图 9-2。用 WDTWINB 替换上面公式中的 WDTREL 即可计算 P_{WDT} 。为了使该窗界刷新特性有用, WDTWINB 不能小于 WINREL。

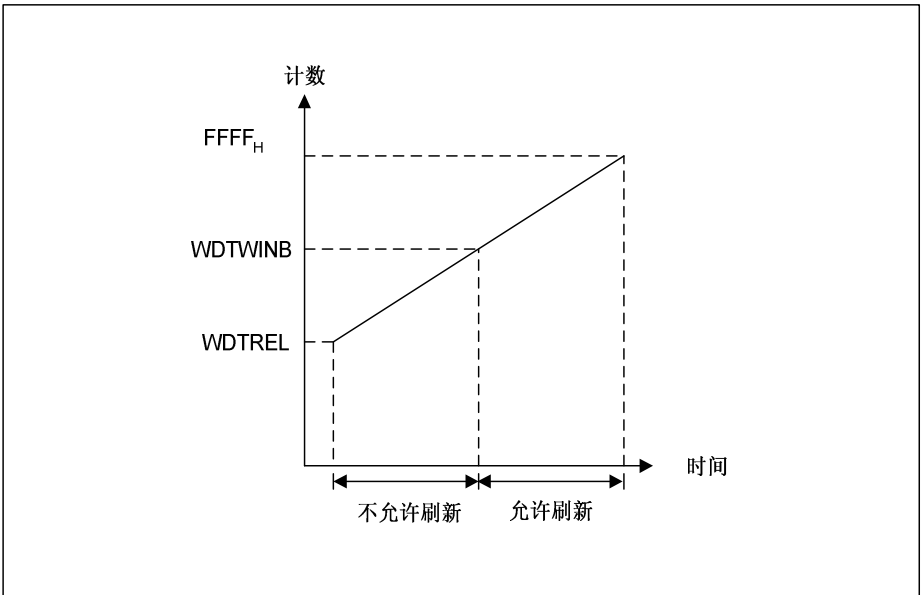


图 9-2 WDT 时间示意图

表 9-1 列出不同的模块输入时钟所对应的 WDT 溢出周期取值。数值取整到 3 位有效。

表 9-1 WDT 溢出周期范围

WDTREL 的重载值	f _{PCLK} 预分频	
	2 (WDTIN = 0)	128 (WDTIN = 1)
	26.7MHz	26.7MHz
FF _H	19.2 μs	1.23 ms
7F _H	2.48 ms	159 ms
00 _H	4.92 ms	315 ms

注：出于安全的考虑，建议用户在每次刷新 WDT 之前重新写入 WDTCON。

9.2 寄存器映射

WDT SFR 位于映射 SFR 区。表 9-2 列出这些 SFR 的地址。

表 9-2 SFR 地址列表

地址	寄存器名称
BB _H	WDTCON
BC _H	WDTREL
BD _H	WDTWINB
BE _H	WDTL
BF _H	WDTH

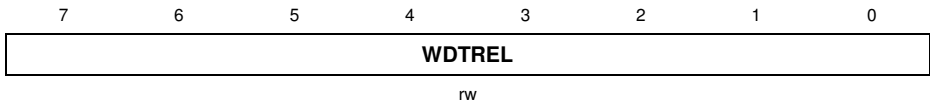
9.3 寄存器描述

WDT 的当前计数值存放在不可位寻址的只读寄存器 WDT 中。可位寻址控制寄存器 WDTCON 控制 WDT 的操作，该寄存器还选择输入时钟的预分频因子。WDTREL 规定定时器高位字节的重载值。

WDTREL

WDT 重载寄存器

复位值: 00H

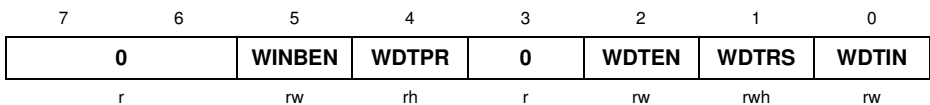


符号	位序号	读写类型	功能描述
WDTREL	[7:0]	rw	WDT 的重载值（针对 WDT 的高位字节） 新的重载值可写入 WDTREL 中。当定时器使能或下次刷新时该值被载入 WDT 的高 8 位字节

WDTCON

WDT 控制寄存器

复位值: 00H



符号	位序号	读写类型	功能描述
WDTIN	0	rw	WDT 输入频率选择 0 输入频率为 $f_{PCLK} / 2$ 1 输入频率为 $f_{PCLK} / 128$
WDTRS	1	rwh	WDT 刷新启动控制 高有效。该位置位将开始刷新 WDT，该位由硬件自动清零。

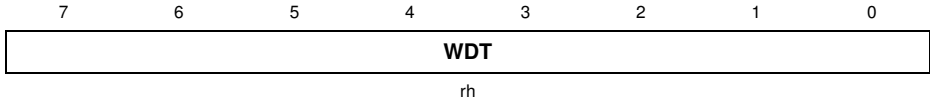
符号	位序号	读写类型	功能描述
WDTEN	2	rw	WDT 使能 0 禁止 WDT 1 使能 WDT WDTEN 是被保护位。保护方案（见章节 3.3.4.1）有效时该位不能直接写入。
WDTPR	4	rh	WDT 预报警模式标志 0 正常模式（复位后的缺省模式） 1 WDT 工作在预报警模式 检测到 WDT 出错时该位被置 1，WDT 产生 NMI 强制中断并进入预报警模式。预报警周期结束后芯片复位
WINBEN	5	rw	WDT 窗界控制 0 禁止 WDT 窗界特性（缺省状态） 1 使能 WDT 窗界特性
0	3, [7:6]	r	保留 读操作返回 0；应写入 0

看门狗定时器

WDTL

WDT 寄存器，低位字节

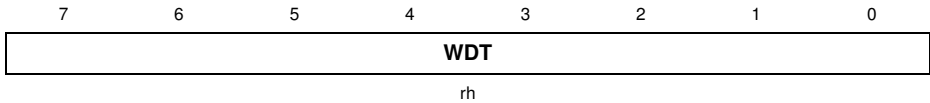
复位值: 00_H



WDTH

WDT 寄存器，高位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
WDT	WDTH [7:0] , WDTL [7:0]	rh	WDT 的当前计数值

WDTWINB

WDT 窗界计数寄存器

复位值: 00_H




符号	位序号	读写类型	功能描述
WDTWINB	[7:0]	rw	<p>WDT 窗界计数值</p> <p>该值可编程设定。窗界取值在 0000_H和 WDTWINB 和 00_H 串联值之间，在窗界内 WDT 不能被刷新，否则 WDRST 将有效。WDTWINB 与 WDTH 匹配。</p>

PMCON0

功率模式控制寄存器 0

复位值:见表 8-1

7	6	5	4	3	2	1	0
0	WDTRST	WKRS	WKSEL	SD	PD	WS	
r	rwh	rwh	rw	rw	rwh	rw	

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
WDTRST	6	rwh	WDT 复位指示 0 未发生 WDT 复位 1 WDT 已复位
0	7	r	保留 读操作返回 0; 应写入 0

10 串行接口

XC866 有两个与外部器件通信的串行接口，通用异步收发器（UART）和高速同步串行接口（SSC）。另外，可用 UART 支持局域网（LIN）协议。

UART 特性：

- 全双工异步模式
 - 8 位或 9 位的数据帧，先发送最低有效位（LSB）
 - 固定或可变的波特率
- 接收缓冲
- 多处理器通信
- 每帧数据发送或接收完成产生中断

LIN 特性：

- 主模式和从模式操作

SSC 特性：

- 主模式和从模式操作
 - 全双工或半双工操作
- 数据发送、接收双缓冲
- 灵活的数据格式
 - 数据位个数可编程：2 至 8 位
 - 移位方向可编程：先发送 LSB 或最高有效位（MSB）
 - 时钟极性可编程：移位时钟低电平空闲或高电平空闲
 - 时钟/数据相位可编程：在移位时钟的前沿或后沿进行数据移位
- 波特率可变
- 与串行外设接口（SPI）兼容
- 产生中断
 - 发送缓冲寄存器已空
 - 接收缓冲寄存器已满
 - 出错状况（接收、相位、波特率、发送错误）

10.1 UART

UART 是一个全双工异步收发器，可以同时接收和发送。它还具有接收缓冲，在第一个字节从接收寄存器读出之前，可开始接收第二个字节。但是如果第二个字节接收完毕时第一个字节仍未被读出，其中一个字节将会丢失。

10.1.1 UART 模式

UART 有 3 种异步模式。模式 1 UART 是一个 8 位串行口；模式 2 和模式 3 UART 是一个 9 位串行口。模式 2 和模式 3 的唯一区别在于波特率不同，模式 2 的波特率固定不变，模式 3 的波特率可变。通过专用波特率发生器产生可变的波特率。

通过把 SM0 和 SM1 设置为表 10-1 中所对应的值，来选择不同的工作模式。SM0 和 SM1 的值均为 0 的选择保留不用。

表 10-1 UART 的工作模式

SM0	SM1	工作模式	波特率
0	0	保留	—
0	1	模式 1: 8 位移位 UART	可变
1	0	模式 2: 9 位移位 UART	$f_{PCLK}/32$ 或 $f_{PCLK}/64$
1	1	模式 3: 9 位移位 UART	可变

10.1.1.1 模式 1, 8 位 UART, 波特率可变

模式 1 中 UART 是 8 位串行口。一位起始位 (0)、8 位数据位和一位停止位 (1) 组成的数据帧发送到 TXD 端或从 RXD 端接收，波特率取决于专用波特率发生器的下溢速率，波特率可变。

写入 SBUF 激活发送过程。数据被移入发送移位寄存器，“1”被装入第 9 位。在 16 分频计数器下次翻转后的机器周期的 P1 时刻，起始位发送到 TXD 上，一位时间后数据有效。再过一位后数据开始右移输出，从左边填入 0。当 MSB 移至移位寄存器的输出位置时，控制器作最后一次移位并置位 TI。

在 RXD 端检测到负跳变时启动接收（采样速率为波特率的 16 倍），16 分频计数器复位，并将 11111111_B 写入接收移位寄存器。如果检测到有效起始位 (0)（基于“三中取二”），则将其移入寄存器，接着移入 8 位数据。如果接收的第一位不是有效起始位，控制器将重新回去检测 RXD 的负跳变。当起始位移至寄存器的最左端时，控制器作最后一次移位，然后将 8 位数据装入 SBUF，停止位装入 RB8 (SCON.2)，置位 RI。上述动作只有满足以下条件时才发生：RI = 0 并且 SM2 = 0（见章节 10.1.2）或接收到的停止位 = 1。如果这些条件都不满足，所接收的数据就会丢失。

模式 1 发送/接收的相关时序如图 10-1 所示。

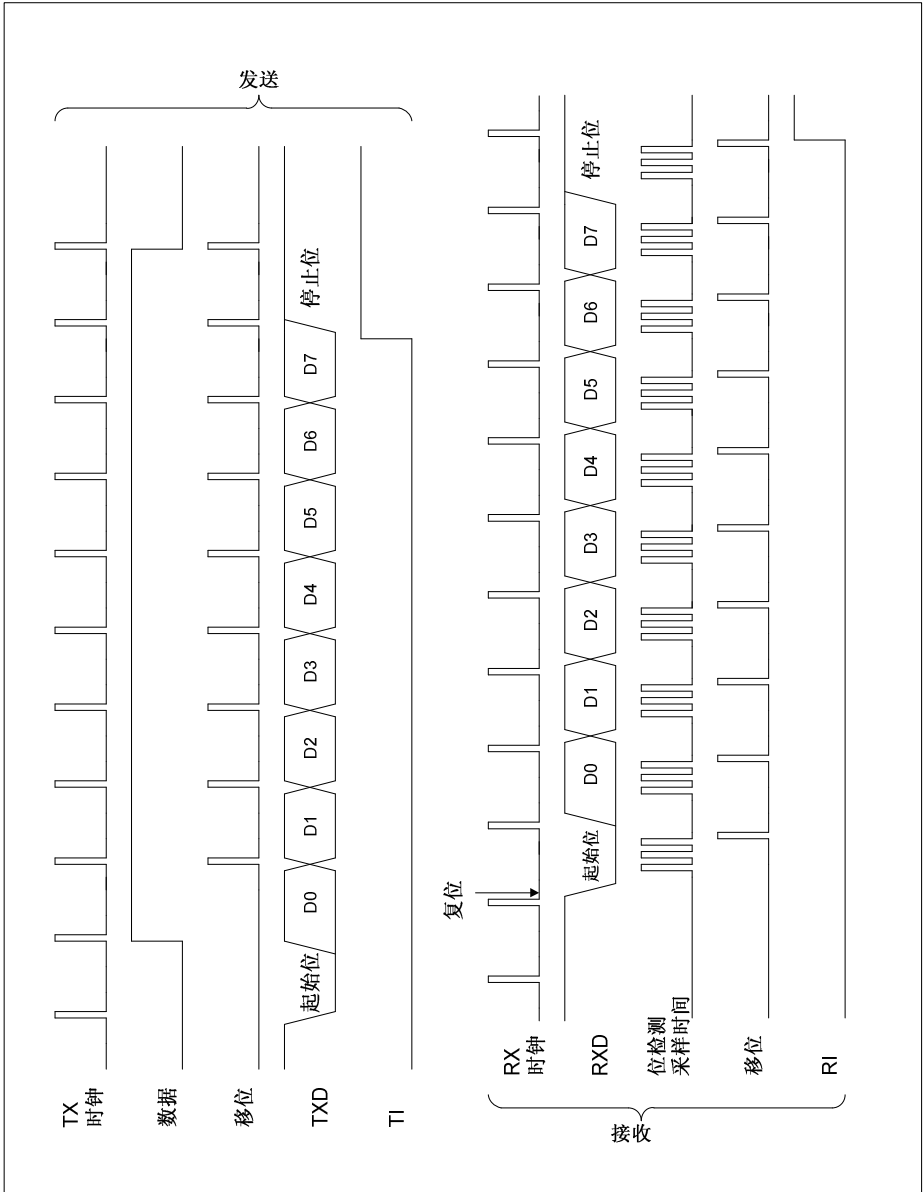


图 10-1 串行接口，模式 1，时序图

10.1.1.2 模式 2, 9 位 UART,波特率固定

模式 2 中 UART 是 9 位串行口。由一位起始位 (0)、8 位数据位、一位可编程数据位 (第 9 位), 和一位停止位 (1) 组成的数据帧发送到 TXD 端或从 RXD 端接收, 发送时第 9 位取自 TB8 (SCON.3); 接收时, 第 9 位存入 RB8 (SCON.2)。

写入 SBUF 激活发送过程。数据被移入发送移位寄存器, TB8 复制到第 9 位。在 16 分频计数器下次翻转后的机器周期的 P1 时刻, 起始位发送到 TXD 上, 一位时间后数据有效。再过一位后数据开始右移输出。第一次移位, 停止位 (1) 从左边移入, 此后填入 0。当 TB8 移至输出位置时, 控制器作最后一次移位并置位 TI。

在 RXD 端检测到负跳变时启动接收 (采样速率为波特率的 16 倍), 16 分频计数器复位, 并将 11111111_B 写入接收移位寄存器。如果检测到有效起始位 (0) (基于“三中取二”), 则将其移入寄存器, 接着移入 8 位数据。如果接收的第一位不是有效起始位, 控制器将重新回去检测 RXD 的负跳变。当起始位移至寄存器的最左端时, 控制器作最后一次移位, 然后将 8 位数据装入 SBUF, 第 9 位数据装入 RB8 (SCON.2), 置位 RI。上述动作只有满足以下条件时才发生: RI = 0 并且 SM2 = 0 (见章节 10.1.2) 或第 9 位 = 1。如果这些条件都不满足, 所接收的数据就会丢失。

传输波特率为 $f_{PCLK}/32$ 或 $f_{PCLK}/64$, 这取决于 PCON (功率控制寄存器) 中最高位 (SMOD) 的设置, 该位用于双倍频选择。

10.1.1.3 模式 3, 9 位 UART,波特率可变

除了波特率可变之外, 模式 3 和模式 2 完全相同。模式 3 的波特率取决于专用波特率发生器的下溢速率。

所有模式下, 数据的发送由执行任意一条以 SBUF 为目的寄存器的指令启动; 数据的接收由接收到有效起始位启动 (若 REN=1)。

数据帧发送或接收时完成, 串行接口产生中断请求。相应的中断请求标志分别为 TI 或 RI。如果不使用串行中断 (串行中断被禁止), 也可用 TI 和 RI 查询串行接口。

模式 2 和 3 发送/接收的相关时序如图 10-2 所示。

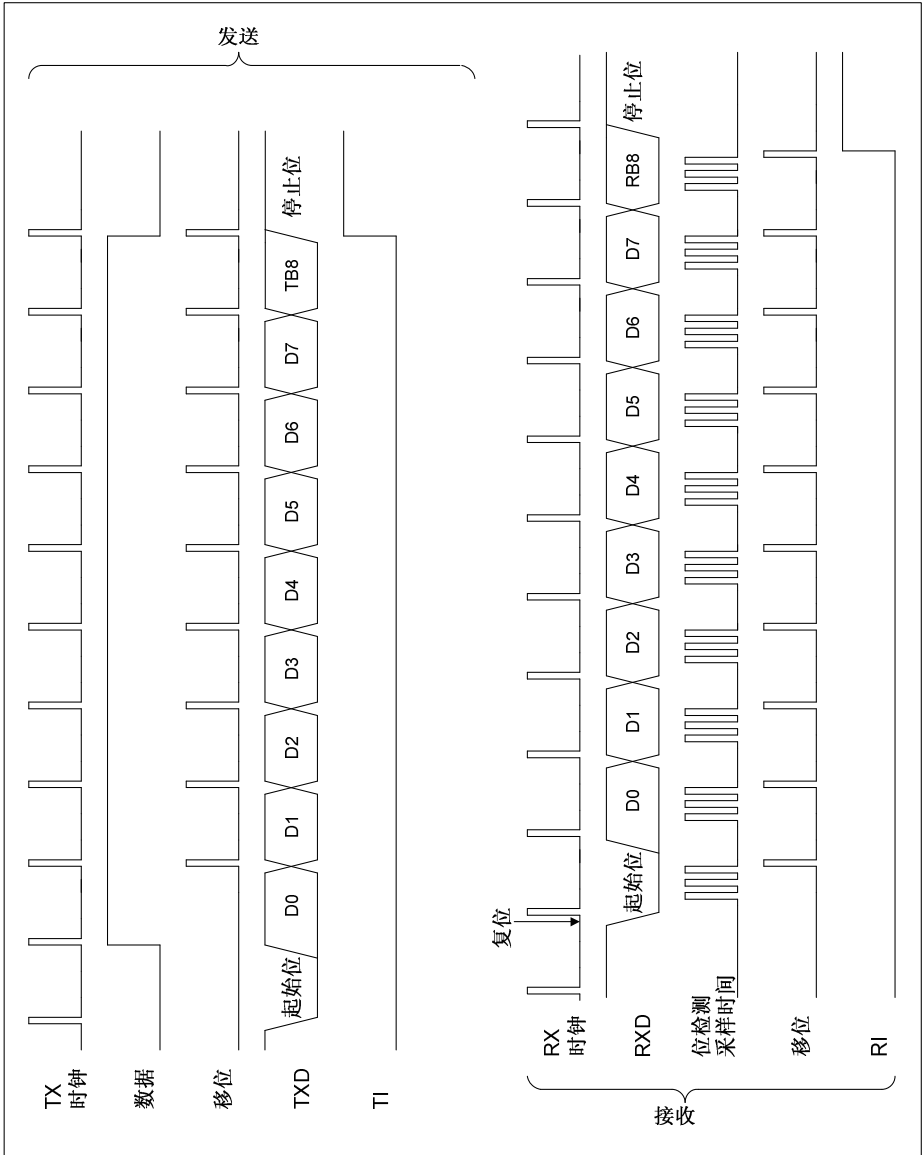


图 10-2 串行接口，模式 2 和 3，时序图

10.1.2 多处理器通信

模式 2 和模式 3 有一个专门的应用领域即多处理器通信，地址字节的第 9 位 = 1、数据字节的第 9 位 = 0。在这些模式下接收 9 位数据，第 9 位存入 RB8。始终以停止位来结束通信。串行口可编程为：当接收到停止位、仅当 RB8 = 1 时串口中断才有效。

可通过置位寄存器 SCON 中的位 SM2 选择该特性。下面为多处理器系统利用该特性的一种方法。

当主机需要发送一数据块给数台从机之一时，首先发送一个地址字节对目标从机进行识别，地址字节和数据字节通过第 9 位数据区别：地址字节的第 9 位为 1；数据字节的第 9 位为 0。SM2 = 1 时，数据字节不会使从机产生中断，而地址字节将中断所有从机，这样每个从机都会检查接收到的字节判断是否被寻址。被寻址的从机对 SM2 清零，准备接收主机送来的数据；未被寻址的从机的 SM2 仍为 1，对主机发送的数据不予理睬。

模式 1 时 SM2 用来检验停止位是否有效。模式 1 时，如果 SM2 = 1，只有接收到有效的停止位才可产生接收中断。

10.1.3 寄存器描述

UART 有两个 SFR，控制寄存器 SCON 和数据寄存器 SBUF。两个寄存器复位后均为 00_H。SCON 是串口控制和状态寄存器，该寄存器包含模式选择位、发送和接收的第 9 位 (TB8 和 RB8)，以及串口中断位 (TI 和 RI)。

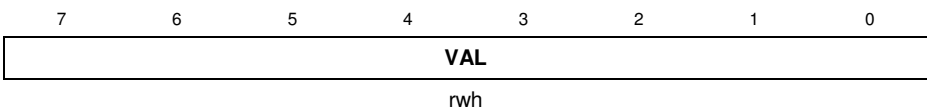
SBUF 是发送和接收缓冲寄存器。写入 SBUF 将加载发送移位寄存器并启动数据发送。该寄存器既用来发送也用来接收数据。发送数据写入该寄存器；接收数据从该寄存器读出，但两条数据通路互相独立。

从 SBUF 读取数据时是对物理上分开的接收寄存器进行访问。

SBUF

串行数据寄存器

复位值: 00_H



符号	位序号	读写类型	功能描述
VAL	[7:0]	rwh	串行接口缓冲寄存器

SCON
串行通道控制寄存器
复位值: 00H

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
rw	rw	rw	rw	rw	rwh	rwh	rwh

符号	位序号	读写类型	功能描述
RI	0	rwh	接收中断标志 模式 1, 2 和 3 中, 接收到停止位的中间时刻由硬件置位。必须由软件清零。
TI	1	rwh	发送中断标志 模式 1, 2 和 3 中, 开始发送停止位时由硬件置位。必须由软件清零。
RB8	2	rwh	串口接收的第 9 位 模式 2 和 3 中接收到的第 9 位数据; 模式 1 中该位是接收到的停止位。
TB8	3	rw	串口发送的第 9 位 模式 2 和 3 中发送的第 9 位数据
REN	4	rw	串口接收使能位 0 禁止串口接收 1 使能串口接收
SM2	5	rw	模式 2 和 3 中串口多处理器通信使能位 模式 2 或 3 中, 若 SM2 被置 1, 如果接收到的第 9 位数据 (RB8) 为 0, RI 不会被激活; 模式 1 中, 若 SM2 被置 1, 如果没有接收到有效的停止位 (RB8), RI 不会被激活。

符号	位序号	读写类型	功能描述		
SM1 SM2	6	rw	串口操作模式选择		
	7				
	SM0			SM1	模式选择
	0			0	模式 0: 保留
	0			1	模式 1: 8 位 UART, 波特率可变
1	0	模式 2: 9 位 UART, 波特率固定 ($f_{PCLK}/32$ 或 $f_{PCLK}/64$)			
1	1	模式 3: 9 位 UART, 波特率可变			

10.1.4 波特率产生

根据不同的操作模式，有几种产生串口波特率时钟的方式。

必须对“波特率时钟”和“波特率”加以区分。串行接口需要的时钟是波特率的 16 倍，用于内部同步。因此，波特率发生器必须为串口提供“波特率时钟”，它被 16 分频即得到真正的“波特率”。 f_{PCLK} 指输入时钟频率。

模式 2 中，波特率为 $f_{PCLK}/32$ 或 $f_{PCLK}/64$ ，这取决于 PCON（功率控制寄存器）中最高位（SMOD）的设置，该位用于双倍频选择。

模式 1 或 3 中，可变的波特率取决于专用波特率发生器的下溢速率。

模式 2 中串口的固定波特率由 PCON 的位 SMOD 控制，如下所示。模式 1 和 3 中由专用波特率发生器提供的可变波特率不受该位影响。

PCON

功率控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
SMOD	0			GF1	GF0	0	IDLE
rw	r			rw	rw	r	rw

阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
SMOD	7	rw	波特率双倍频使能 0 模式 2 中串口波特率不双倍频 1 模式 2 中串口波特率双倍频

10.1.4.1 波特率发生器

XC866 提供了一个专用波特率发生器，产生 UART 的波特率，具有 8 位可编程重载值和 3 位预分频因子。

波特率发生器的时钟 (f_{DIV}) 来自输入时钟 f_{PCLK} 经预分频。波特率定时器递减计数，由波特率发生器的运行控制位 **BCON.R** 启动或停止定时器工作。每次定时器下溢产生一个时钟脉冲，同时将 8 位重载寄存器中存放的值重新载入定时器。位域 **BCON.BRPRE** 选择预分频因子。

寄存器 **BG** 是具有双重功能的波特率发生器/重载寄存器。读取 **BG** 返回定时器的计数值；写入 **BG** 会更新重载寄存器的值。

每次写入 **BG** 时，重载寄存器的内容会自动重新载入定时器。但如果在执行 **BG** 写操作的同时 **BCON.R** 被清零，只有在 **BCON.R** 置位后的第一个指令周期定时器才能被重新载入。

波特率发生器的波特率取决于以下位和寄存器的值：

- 输入时钟 f_{PCLK}
- 寄存器 **BCON.BRPRE** 的值
- 8 位重载寄存器 **BG** 的值

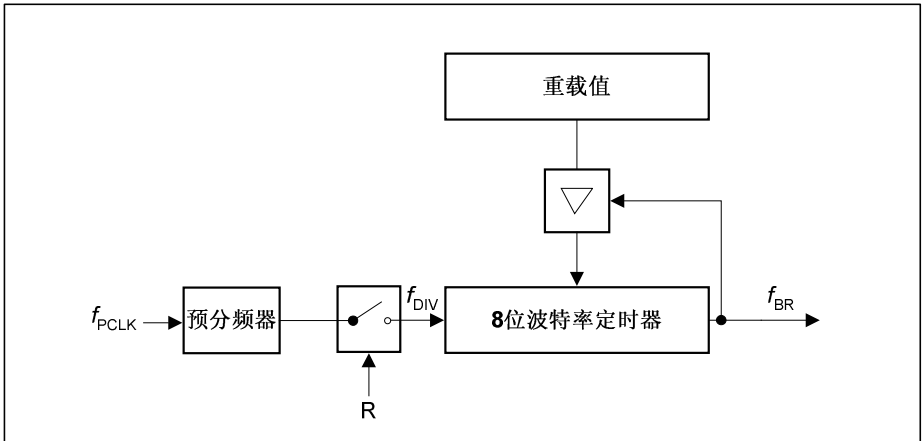


图 10-3 波特率发生器电路

串行接口需要的时钟是波特率的 16 倍，用于内部同步。因此，波特率发生器必须为串口提供“波特率时钟”，它被 16 分频即得到真正的“波特率”。下面的公式已包括该分频因子计算得到最终的波特率。

$$\text{波特率} = \frac{f_{\text{PCLK}}}{16 \times \text{PRE} \times \text{BG} + 1}$$

PRE（预分频）的值由 BCON.BRPRE 选择；BG 表示重载寄存器 BG.BR_VALUE 的值，为无符号 8 位整数。

输入频率 26.7MHz 对应可获得的最大波特率为 0.83 MBaud。波特率要设置小于 $f_{\text{PCLK}}/32$ 。

表 10-2 列出通常所使用的不同的波特率、对应的重载值以及与理想波特率相比较的波特率偏差。

表 10-2 UART 典型波特率

波特率 ($f_{\text{PCLK}}=26.7\text{MHz}$)	PRE	重载值	偏差
19.2 KBaud	1 (BRPRE=000)	86(56 _H)	-0.22%
9600 Baud	1 (BRPRE=000)	173(AD _H)	-0.22%
4800 Baud	2 (BRPRE=001)	173(AD _H)	-0.22%
2400 Baud	4 (BRPRE=010)	173(AD _H)	-0.22%

寄存器 BCON 包含了波特率发生器的控制位和预分频选择位。

BCON
波特率控制寄存器
复位值: 00H

7	6	5	4	3	2	1	0
BGSEL		T2EXIS	BRDIS	BRPRE			R
rw		rw	rw	rw			rw

符号	位序号	读写类型	功能描述
R	0	rw	波特率发生器运行控制 0 禁止波特率发生器工作 1 使能波特率发生器工作 <i>注: R=0 时 BR_VALUE 只能写入。</i>
BRPRE	[3:1]	rw	预分频选择 选择输入时钟 f_{DIV} (来自外设时钟经预分频) 000 $f_{DIV} = f_{PCLK}$ 001 $f_{DIV} = f_{PCLK}/2$ 010 $f_{DIV} = f_{PCLK}/4$ 011 $f_{DIV} = f_{PCLK}/8$ 100 $f_{DIV} = f_{PCLK}/16$ 101 $f_{DIV} = f_{PCLK}/32$ 其他: 保留
BRDIS	4	rw	波特率检测禁止 0 使能波特率检测 1 禁止波特率检测
T2EXIS	5	rw	T2EX 功能选择 0 选择 T2EX 用作波特率检测 1 选择 T2EX 用作定时器 2 功能
BGSEL	[7:6]	rw	检测波特率范围选择 00 10 kHz 到 20 kHz

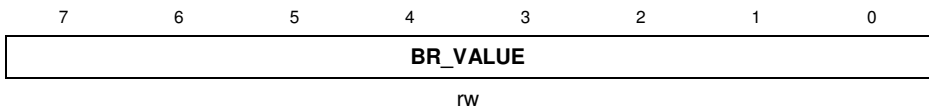
符号	位序号	读写类型	功能描述
			01 5 kHz 到 10 kHz
			10 2.5 kHz 到 5 kHz
			11 1.25 kHz 到 2.5 kHz

波特率定时器/重载寄存器 BG 包含了波特率定时器的 8 位重载值。

BG

波特率定时器/重载寄存器

复位值: 00H



符号	位序号	读写类型	功能描述
BR_VALUE	[7:0]	rw	波特率定时器/重载值 读操作将返回波特率定时器的 8 位计数值； 写操作将重新加载波特率定时器/重载值。 <i>注：R=0 时 BR_VALUE 只能写入。</i>

10.1.5 UART 接口

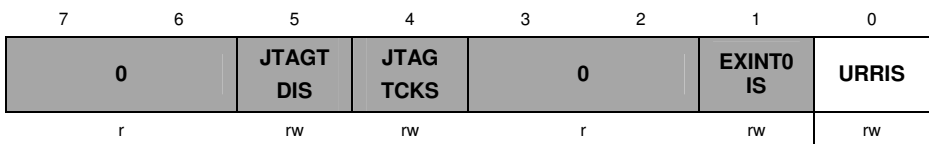
UART 有两条 I/O 线，TXD 用于数据发送；RXD 用于数据接收。

可选择从两个不同的输入源 RXD_0 和 RXD_1 将数据移入 UART，由位 MODPISEL.URRIS 控制选择。

MODPISEL

外设输入选择寄存器

复位值: 00H





阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
URRIS	0	rw	UART 接收输入选择 0 选择 UART 接收输入 RXD_0 1 选择 UART 接收输入 RXD_1

10.2 局域网互联网（LIN）

UART 支持局域网互联网（LIN）协议的主从模式操作。LIN 的波特率检测特使用定时器 2 检测 LIN 总线波特率，从而使 UART 与 LIN 的波特率同步、以进行数据发送和传输。

10.2.1 LIN 协议

LIN 是一种机动车内部网络互联的通信概念。该通信协议基于 SCI（UART）数据格式、单主机/多从机概念、无需固定的时间基准即可实现节点时钟同步。LIN 具有吸引力的一大特性是：从节点无需石英或陶瓷谐振器即可实现自同步，大大降低了硬件平台的成本。因此需要计算波特率用于报文帧传输。

LIN 的报文帧结构如图 10-4 所示。由以下部分组成：

- 报文头，由分隔域（13 位低位时间），同步域（55 μ s）和标识符域组成
- 响应间隔
- 数据字节（根据 UART 协议）
- 校验和

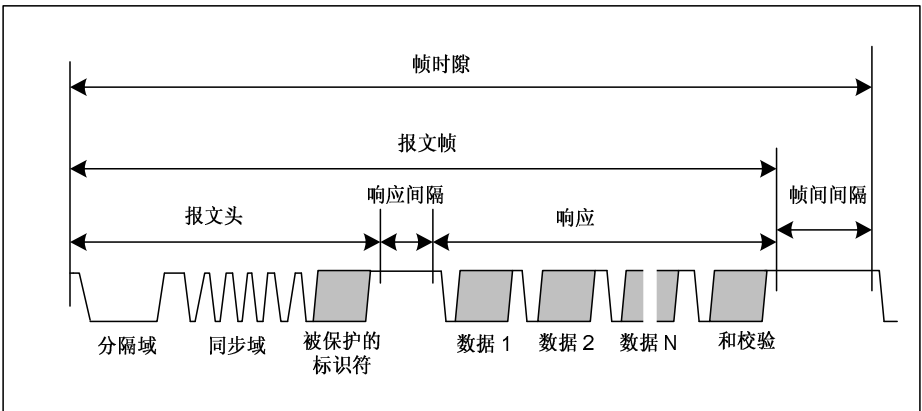


图 10-4 LIN 的帧结构

每个字节域串行发送，如图 10-5 所示。先发送 LSB；最后发送 MSB。起始位编码为值“0”（主控值），停止位编码为值“1”（退让值）。

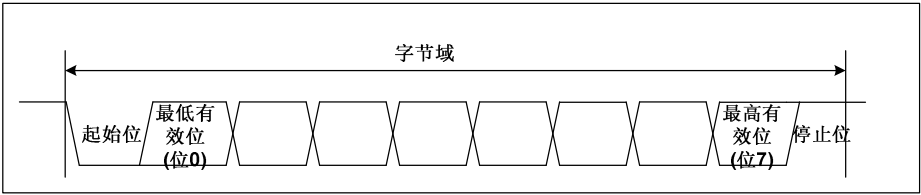


图 10-5 字节域结构

分隔域用作新的一帧的起始信令。只有该域不遵循图 10-5 所示的字节域结构。始终由主机任务（主机模式）产生分隔域，它必须至少保持包括起始位在内的 13 位主控值，随后为分隔符，如图 10-6 所示。分隔符必须至少保持 1 位标称位时间。

从节点将使用 11 位标称位时间的分隔域检测阈值。



图 10-6 分隔域

同步字节是确定时间基准的特定序列，值为 55H，如图 10-7 所示。

即使从机希望接收到字节域（假定字节域相互分开），它始终能够检测分隔域/同步序列。如果在检测阶段接收到字节域，分隔域/同步序列检测将中断正在进行的数据传输，开始新一帧数据的处理。

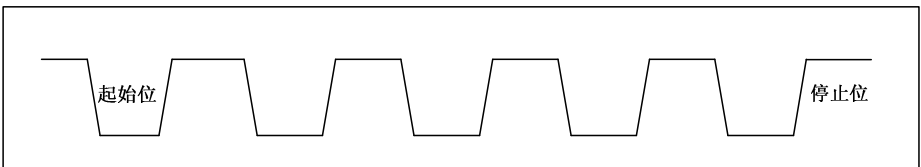


图 10-7 同步字节域

当主机发送了正确的标识符，相应的从机将接收和发送数据：

- 从机等待同步字节
- 从机与同步字节同步

- 从机查询标识符
- 从机根据标识符来确定接收或发送数据，或不做任何操作
- 发送数据时，从机发送 2、4 或 8 个数据字节，随后是校验字节

10.2.2 LIN 报文头传输

LIN 报文头传输只适用于主模式。在 LIN 通信中，主机任务决定在何时、将哪帧传输到总线上；还决定由哪个从机来接收每帧数据。主机和从机之间所需的握手信息由主机的报文头提供。

报文头由分隔序列、同步序列及随后的标识符组成。这三个域中只有分隔域不按通常的 8 位 UART 数据格式传输。分隔域必须至少包含 13 位主控值以保证从机能正确同步。

UART 可用来传输 20 位的分隔域，步骤如下：

步骤 1：将 UART 设置为工作在模式 1

- 波特率可变的 8 位 UART

步骤 2：把波特率设置为所需波特率的两倍

步骤 3：发送缓冲寄存器中写入 00_H以启动传输

采用两倍于所需波特率的波特率发送数据，由起始位、停止位和 8 位数据位组成的 10 位 UART 帧结构在 LIN 总线上可达到传输 20 位分隔域的效果。必须在接下来的同步域和标识符域中将波特率调整回原先的值。

10.2.3 LIN 波特率检测

在 LIN 通信协议中，要求从机在开始接收被保护的标识符前与主机同步。为此，每帧的起始序列为分隔域及随后的同步字节域。该序列是唯一的，为从机检测新的一帧的开始、并在接收标识符域之前与主机同步提供了足够信息。

为了检测 LIN 总线波特率，测量序列中两个下降沿之间的时间可计算位时间 T_{bit} 。在波特率检测模式下，捕获同步字节域中两位位时间，如图 10-8 所示。

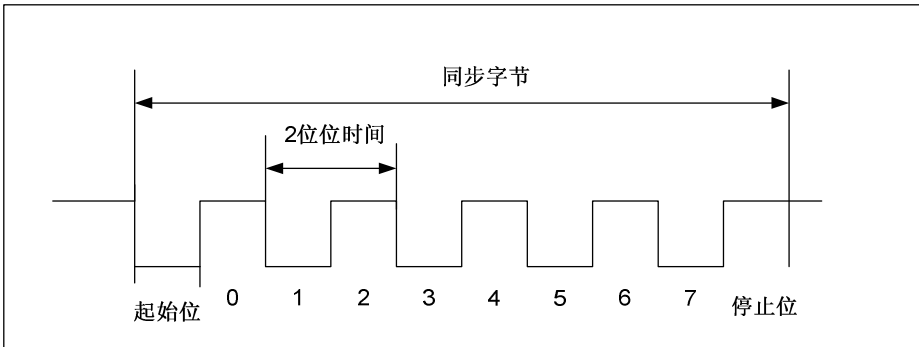


图 10-8 同步字节的位时间

BCON 寄存器的位 4-7（见页 10-12）用于 LIN 的波特率检测。位 BRDIS 使能/禁止波特率检测；位 T2EXIS 选择 T2EX 引脚（P1.0）用于波特率检测或作为正常定时器 2 的功能使用；如果用户知道 LIN 总线的波特率范围，应该在位 BGSEL 中对此进行规定，波特率检测单元将根据该信息、针对不同的波特率使用不同的采样率，从而保证正确的波特率检测。

通常执行下列步骤进行波特率检测：

步骤 1：在 RXD 的第一个下降沿：

- 如果系统处于掉电模式且 $PWCON0.WS = 01$ ，掉电唤醒将由引脚 RXD 激活。
- 一旦系统进入正常工作模式，软件执行下列设置：
 - 位 $PMCON1.T2_DIS$ 置 0（使能定时器 2）
 - 位 $BCON.BRDIS$ 置 0（使能波特率检测）
 - 位 $BCON.T2EXIS$ 置 0（T2EX 用作波特率检测）
 - 由位 $BCON.BGSEL$ 提供波特率范围
 - 位 $T2CON.P/RL2$ 和 $T2CON.EXEN2$ 均置 1。T2MOD.EDGESEL 置 0。（定时器 2 设置为由下降沿触发的捕获模式）

- UART 以估算的波特率运行，该波特率由波特率发生器产生（见章节 10.1.4.1）。

步骤 2：在同步字节的第 2 个下降沿：

- 硬件启动定时器 2 工作（同步字节域的位 1）。

步骤 3：在同步字节的第 3 个下降沿：

- 将触发定时器 2 捕获同步字节域中位 1 和位 2 的时间。定时器寄存器（THL2）的内容被捕获至 RC2 寄存器。捕获值是 LIN 的 2 位位时间。如果检测到捕获信号的同时计数器正在加 1，计数器首先加 1 然后执行捕获操作，从而保证始终能够捕获到定时器寄存器中的最新值。
- 当捕获完成，位 T2CON.EXF2 置位并可用来产生中断请求。
- 软件将利用定时器 2 的 RC2 的值得到波特率发生器的重载值 BG_VALUE 和预分频因子 BRPRE。
- 软件利用新的 BG 值和预分频因子更新波特率发生器，产生新的波特率。

软件重新在定时器 2 的重载寄存器（RC2）中装入 0000_H。

10.3 高速同步串行接口

同步串行接口 (SSC) 支持全双工和半双工同步通信。串行时钟信号由 SSC 内部的 16 位波特率发生器产生 (主模式), 或从外部主机接收得到 (从模式)。数据宽度、移位方向、时钟极性和相位均可编程设定, 从而能够和串行外设接口 (SPI) 兼容器件或使用其他同步串行接口的器件进行通信。

通过 TXD 和 RXD 线发送和接收数据, 通常这两条线分别和引脚 MTSR (主机发送/从机接收) 和 MRST (主机接收/从机发送) 相连。时钟信号从 MS_CLK (主机串行移位时钟) 输出或从 SS_CLK (从机串行移位时钟) 输入; 这两条时钟线通常和引脚 SCLK 相连。数据发送、接收双缓冲。

图 10-9 示出 SSC 的方框图。

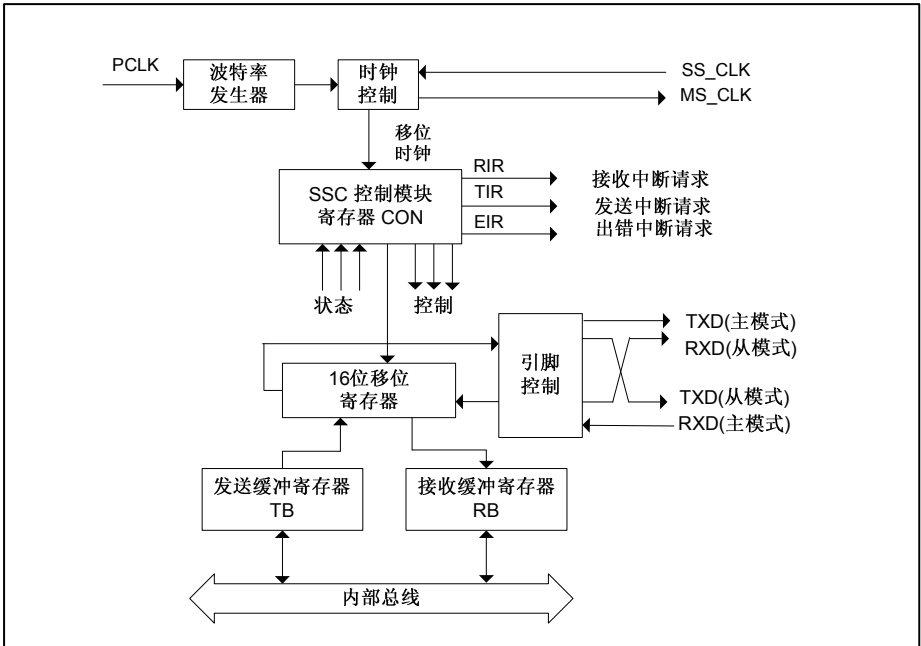


图 10-9 同步串行通道 SSC 方框图

10.3.1 基本操作

10.3.1.1 工作模式选择

由控制寄存器 CON 选择串行通道 SSC 的工作模式。该寄存器具有双重功能：

- 编程过程 (CON.EN=0 禁止 SSC)，存取控制位
- 工作工程 (CON.EN=1 使能 SSC)，存取状态标志

SSC 移位寄存器经引脚控制逻辑与发送线和接收线相连。串行数据的发送和接收被同步，发送接收同时发生，即接收数据和发送数据位数相同。发送数据写入发送缓冲寄存器 (TB)，移位寄存器一空即将 TB 的内容移入。SSC 主机 (CON.MS=1) 立刻开始发送；SSC 从机 (CON.MS=0) 在接收到有效的移位时钟才开始工作。开始传输时，忙碌标志 CON.BSY 被置位，发送中断请求线 (TIR) 被激活以指示寄存器 TB 可重新装入。编程设定的数据位 (2...8) 传输完成之后，将移位寄存器的内容移入接收缓冲寄存器 (RB)，接收中断请求线 (RIR) 被激活。如果没有进一步的数据传输 (TB 空)，同时对 CON.BSY 清零。该标志由硬件控制，不应由软件修改。

注：在给定的时间内每次只有一个 SSC 用作主机。

串行数据的传输可用很多方式设定：

- 数据宽度规定为 2 到 8 位
- 可先传输 LSB 或先传输 MSB
- 移位时钟可为低电平空闲或高电平空闲
- 可在移位时钟信号的前沿或后沿对数据移位
- 可根据模块时钟将波特率设定在某范围内
- 可产生移位时钟 (MS_CLK) 或接收移位时钟 (SS_CLK)

这些特性使 SSC 广泛适用于需要串行数据传输的应用领域。

数据宽度选择支持从 2 位到 8 位“字符”任意长度数据帧的传输。先传输 LSB (CON.HB=0) 可使 SSC 和 SSC 器件在同步模式下通信；或和其他串行接口、如 8051 的串行接口通信；先传输 MSB (CON.HB=1) 可使 SSC 与 SPI 接口兼容通信。

无论如何设定数据宽度以及选择先发送 LSB 还是 MSB，寄存器 TB 和 RB 中的传输数据始终右对齐，数据的 LSB 存放在寄存器的第 0 位。内部移位寄存器逻辑将重新安排数据传输顺序。TB 中未选中的数据位被忽略；RB 中未选中的数据位无效、接收中断服务程序将不予理睬。

时钟控制使 SSC 的发送和接收适用于多种不同的串行接口。用特定的一个移位时钟沿 (上升或下降) 移出数据；同时用另一个时钟沿锁存接收的数据。位 CON.PH 选择用上升沿或下降沿进行发送/接收。位 CON.PO 选择空闲状态下移位时钟电平。因此，对于高电平空闲时钟，时钟前沿为时钟下降沿，即 1 到 0 的跳变 (见图 10-10)。

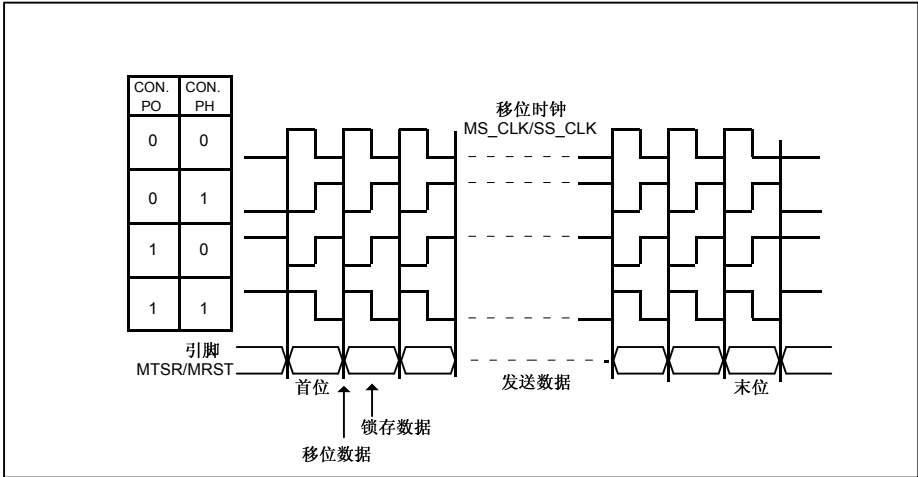


图 10-10 串行时钟相位和极性选择

对串行通信器件初始化，必须选择一个器件作为主机、所有其他器件作为从机。

10.3.1.2 全双工工作

由三条线将各个器件相互连接。由主机来定义这三条线：和主机数据输出线 TXD 相连的是发送线；和数据输入线 RXD 相连的是接收线；移位时钟线是 MS_CLK 或 SS_CLK。只有选作主机的器件产生移位时钟并从 MS_CLK 线上输出。由于所有的从机接收该时钟，它们的引脚 SCLK 必须切换到输入模式。外部连接是硬件连线，这些引脚的功能和方向由各个器件是主机操作还是从机操作决定。

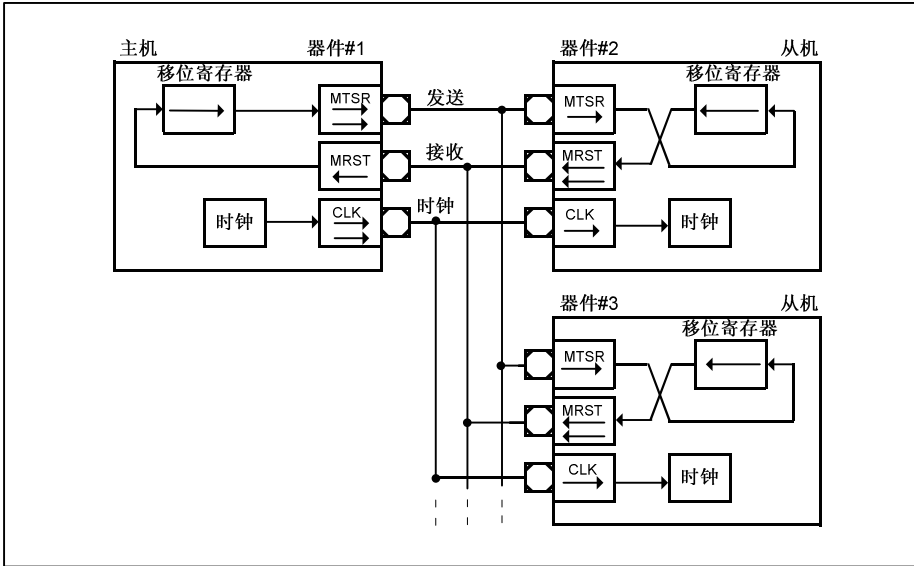


图 10-11 SSC 全双工设置

所有从机器件的数据输出引脚 MRST 共同连接到一条接收线上，如图 10-11 所示。数据传输时，每个从机从移位寄存器移出数据。有两种方式避免（不同从机同时传输数据引起）接收线上的数据传输冲突：

- 只有一个从机驱动接收线，即开启其 MRST 引脚驱动器。必须将所有其它从机的 MRST 引脚设置为输入，从而只有一个从机可将数据输出到主机的接收线上，其它从机只可能接收来自主机的数据。主机有两种方式选择期望获取数据的从机器件：通过分开的选择线或向该从机发送特殊指令。被选中的从机将 MRST 引脚切换到输出，直到它收到一个取消信号或指令。
- 从机在 MRST 引脚上采用开漏输出，形成线与连接。这种情况下接收线要外部上拉。所有不用作向主机发送数据的从机只发送“1”，从而避免了选中的从机发送到接收线上的数据遭破坏。因为这一高电平不能被有效的驱动到接收线上，只能由上拉器件保存；选中的从机发送“0”时可有效的将接收线拉低。主机有两种方式选择期望获取数据的从机器件：通过分开的选择线或向该从机发送特殊指令。

执行完 SSC 的初始化，可启用串行接口。主机器件的时钟线此时为设定的时钟极性。开始传输之前数据线上的值为 0 或 1。传输结束后，数据线将始终保持最后发送的数据位的逻辑电平。

串行接口

当串行接口开启工作，主机器件将发送数据写入 **TB** 启动数据传输。该数据被送入移位寄存器（假定此时寄存器已空），在波特率发生器产生的下一个移位时钟，将所选的发送数据的首位送到 **TXD** 线上（只有 **CON.EN=1** 才可发送）。根据选择的时钟极性，在 **MS_CLK** 线上产生时钟脉冲。同时在相反的时钟沿主机将输入线 **RXD** 上检测到的数据位锁存并移入移位寄存器，发送数据和接收数据进行了“交换”。由于时钟线和所有从机相连，从机移位寄存器将和主机移位寄存器同步移位——移出寄存器中的数据，移入输入线上检测到的数据。

随着传输开始，忙碌标志 **CON.BSY** 被置位，发送中断请求线 **TIR** 被激活以指示寄存器 **TB** 可重新装入。产生预规定个数的时钟脉冲（由数据宽度选择）之后，主机发送的数据全部保存在从机移位寄存器中；同时主机移位寄存器中锁存了选中从机发送的数据。主机和所有从机移位寄存器的内容移入接收缓冲寄存器 **RB**，接收中断请求线 **RIR** 被激活。如果没有进一步的数据传输（**TB** 空），同时对 **CON.BSY** 清零。该标志由硬件控制，不应由软件修改。

从机器件工作时，一旦发送缓冲寄存器中的内容移入移位寄存器，**SSC** 会立刻在输出引脚上输出所选中的数据首位（传输数据的 **MSB** 或 **LSB**）。直到 **SS_CLK** 的第一个时钟沿出现时才置位 **CON.BSY**。

注：无论是否发送或接收有效数据，SSC 的发送和接收总是同时发生。

注：在初始化主机 CLK 引脚时需多加注意，以免产生不希望的时钟跳变干扰其他器件。相关的方向控制寄存器将时钟引脚切换到输出之前，必须由控制寄存器 CON 选择时钟输出电平；由相关的 ALTSEL 寄存器准备该引脚其他功能输出选择；或将时钟空闲电平装入输出锁存器中。

10.3.1.3 半双工工作

在半双工模式下，只有一条数据线既用作接收又用作发送数据。数据交换线和每个器件的 **MTSR** 和 **MRST** 引脚连接，移位时钟线和 **SCLK** 引脚连接。

主机器件产生移位时钟控制数据的传输，同时从机器件接收主机发送的数据。由于所有的发送和接收引脚均连接到同一条数据交换线上，串行数据可在任意两个器件之间传输。

和全双工模式相同，由两种方式避免数据交换线上的数据传输冲突：

- 只有发送器件可开启其发送引脚驱动器
- 不发送数据的器件采用开漏输出并只发送“1”

因为数据输入和输出相连，发送器件将从输入引脚（主机器件对应 **MRST**，从机对应 **MTSR**）读回它发送的数据。若接收数据和发送数据不一致，通过这种方法可以检测到公共数据交换线上数据遭破坏。

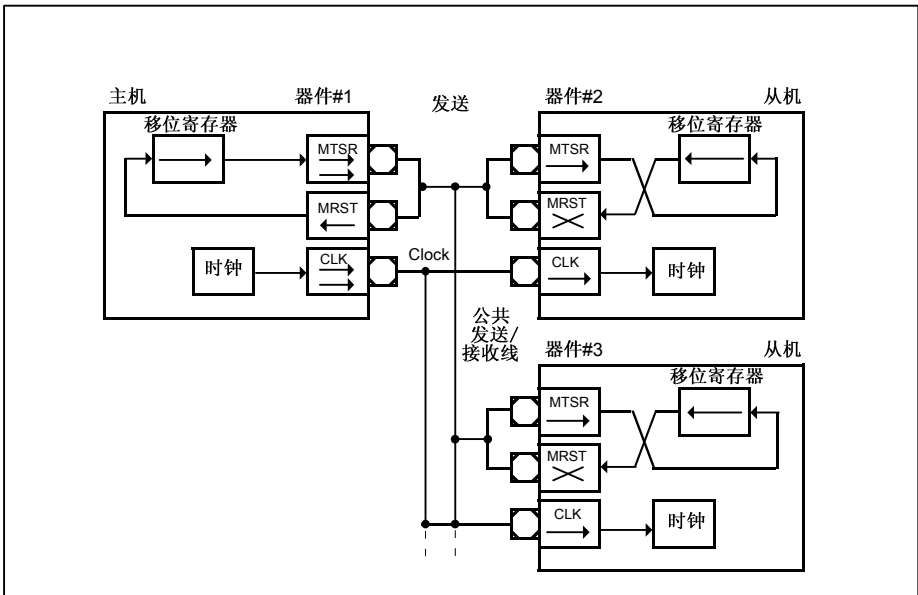


图 10-12 SSC 半双工设置

10.3.1.4 连续传输

当发送中断请求标志被置位，说明发送缓冲寄存器 TB 已空、并准备好可装入下个发送数据。如果当前发送结束时已将新数据装入 TB，该数据立刻被移入移位寄存器，没有任何附加延时开始发送下一轮数据。在数据线上两帧数据无间隔连续传输，例如，传输两个字节看上去和传输一个字的效果相同。该特性用来和可操作或需要每次传输 8 位以上数据的器件接口通信，这只需软件规定数据帧的总长度。该特性用来和字节宽器件和字宽器件接口通信。

注：该特性只允许传输数据宽度为基本数据宽度的整数倍，因为在传输过程中可能禁止/使能 SSC 来重新设定基本数据宽度。

10.3.1.5 端口控制

SSC 使用三条线和外部通信，如图 10-13 所示。引脚 SCLK 用作时钟线；引脚 MRST 和 MTSR 用作串行数据的输入/输出线。

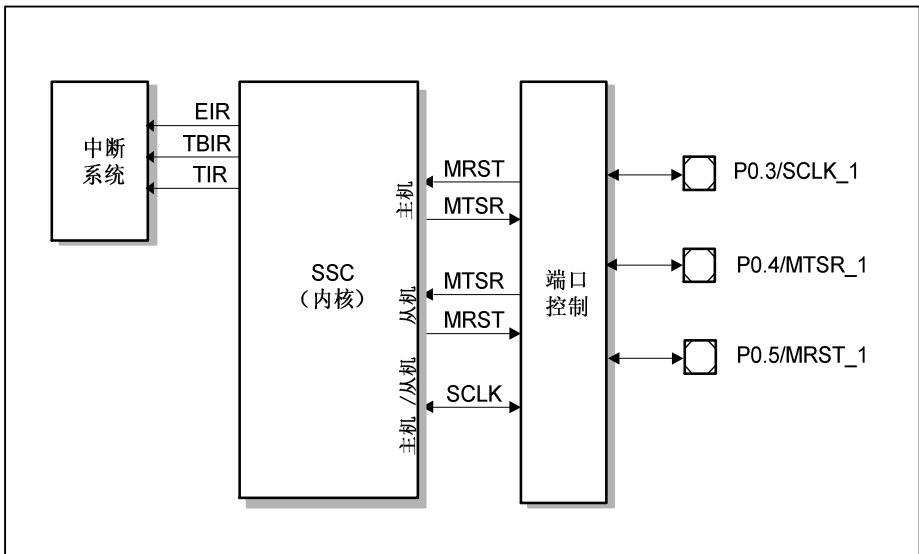


图 10-13 SSC 模块 I/O 接口

SSC 的 I/O 线操作取决于所选择的工作模式（主模式或从模式），端口线的方向也取决于工作模式。模式切换时 SSC 会自动使用正确的端口输出或输入线。

串行接口

由于 SSC I/O 线和通用 I/O 口的双向线连接，软件 I/O 控制用来控制分配给这些线的端口引脚。端口寄存器必须设置成交替输出和输入选择。在主模式和从模式之间切换时，端口寄存器必须重置。

10.3.1.6 波特率产生

串行通道 SSC 拥有专用的具有 16 位重载功能 16 位波特率发生器，从而波特率产生和定时器无关。图 10-14 示出 SSC 的波特率发生器。

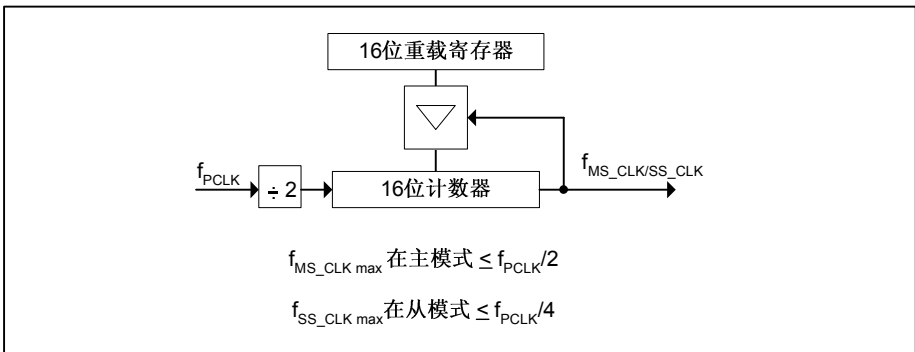


图 10-14 SSC 波特率发生器

波特率时钟来自模块时钟 f_{PCLK} 。定时器递减计数。寄存器 BR 具有双重功能，为波特率发生器/重载寄存器。SSC 被使能时，读取 BG 返回定时器的计数值；SSC 被禁止时，读取 BR 返回设定的重载值。该模式下，期望的重载值可写入 BR。

注：SSC 使能时务必不能对 BR 进行写操作。

给出重载值时用下面的公式计算波特率，或给出波特率时用下面的公式计算重载值：

$$\text{波特率} = \frac{PCLK}{2 \times \langle BR \rangle} \quad BR = \frac{f_{PCLK}}{2 \times \text{波特率}} - 1$$

$\langle BR \rangle$ 代表重载寄存器的值（16 位无符号整数），波特率等于 f_{MS_CLK/SS_CLK} ，如图 10-14 所示。

使用 26.7MHz 的模块时钟对应可获得的最大波特率为：主模式 13.3 MBaud ($\langle BR \rangle = 0000_H$)，从模式 6.7 MBaud ($\langle BR \rangle = 0001_H$)。

表 10-3 列出一些可能的波特率、对应的重载值以及波特率偏差值（假定模块时钟为 26.7 MHz）。

表 10-3 SSC 典型波特率 ($f_{hw_clk} = 26.7\text{MHz}$)

重载值	波特率 (= f_{MS_CLK/SS_CLK})	偏差
0000 _H	13.3 MBaud (仅限主模式)	0.0%
0001 _H	6.7 MBaud	0.0%
0009 _H	1.3 MBaud	0.0%
000C _H	1 MBaud	2.5%
0011 _H	750 kBaud	1.2%
0013 _H	666.7 kBaud	0.0%
0015 _H	600 kBaud	1.0%
001A _H	500 kBaud	1.2%
0031 _H	266.7 kBaud	0.0%
0042 _H	200 kBaud	0.5%
0063 _H	133.3 kBaud	0.0%
0084 _H	100 kBaud	0.25%
FFFF _H	203.45 kBaud	0.0%

10.3.1.7 检错机制

SSC 能够检测四种出错情况。主从模式下均可检测接收出错和相位出错；仅在从模式下检测发送出错和波特率出错。当检测到错误时，对应的出错标志位被置位，激活出错中断请求线 (EIR) 产生出错中断请求 (见图 10-15)。出错中断处理会检查出错标志以确定出错原因。出错标志不能被自动复位，中断被响应后必须由软件清零。从而出错使能位被置位时，可由中断响应出错情况；出错使能位未被置位时，可由软件查询出错情况。

注: 出错中断处理必须对相关 (被使能的) 出错标志清零以防止重复产生中断请求。

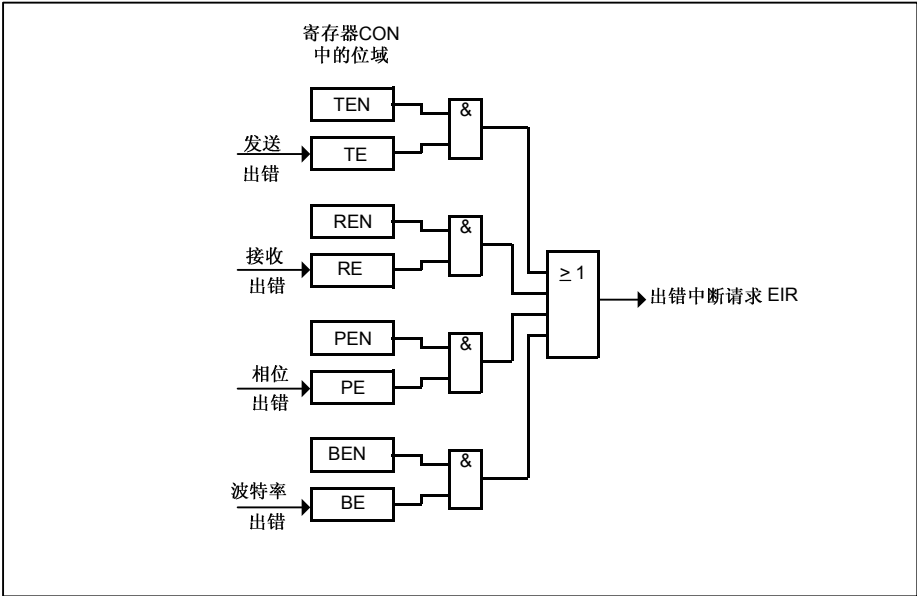


图 10-15 SSC 出错中断控制

接收出错（主模式或从模式）：当新的数据帧已接收完毕，但上个数据仍未从接收缓冲寄存器 RB 中读出时检测到接收出错。此时置位出错标志 CON.RE，CON.REN 使能时置位 EIR。接收缓冲寄存器 RB 中的旧数据将被新数据覆盖且不可恢复。

相位出错（主模式或从模式）：以模块时钟频率采样引脚 MRST（主模式）或引脚 MTSR（从模式）上的输入数据，在移位时钟信号 SCLK 锁存时钟沿的一个周期之前和两个周期之后若数据改变则检测到相位出错。此时置位出错标志 CON.PE，CON.PEN 使能时置位 EIR。

波特率出错（从模式）：当输入时钟信号和设定的波特率之间偏差超过 100%时检测到波特率出错。即偏差超过期望波特率的一倍或不到期望波特率的一半。此时置位出错标志 CON.BE，CON.BEN 使能时置位 EIR。使用该检错功能要求从机波特率发生器产生和主机相同的波特率。该特性可检测时钟线上错误的、多余的或丢失的脉冲（在某帧之内）。

注：如果发生该错误且位 CON.REN=1，将自动复位 SSC。检测到过少或过多的时钟脉冲时自动复位将重新初始化 SSC。

发送出错（从模式）：主机启动传输（SS_CLK 已有效），但从机发送缓冲寄存器 TB 在上次传输后仍未更新，检测到发送出错。此时置位出错标志 CON.TE，CON.TEN 使能时置位 EIR。如果发送缓冲器未更新即开始传输，从机将移位输出移位

串行接口

寄存器中“旧”的内容，它通常是上次传输时接收的数据。如果未选中该从机发送数据，这将可能导致半双工模式下发送/接收线上数据被破坏（开漏输出设置）。该模式要求不被选中发送数据的从机只发送“1”；也就是说，传输前必须将“FFFF_H”装入从机发送缓冲寄存器中。

注：带有上拉/下拉输出驱动器、未被选中发送数据的从机，通常会关闭它的输出驱动器。但为了避免可能的数据冲突或数据错误识别，建议在任何传输之前总是先加载从机发送缓冲寄存器。

可通过控制寄存器 CON 中的出错状态标志确定出错中断请求（接收、相位、波特率或发送出错）产生的原因。

注：错误状态标志 CON.TE, CON.RE, CON.PE, CON.BE 在进入出错中断服务程序时不能被自动复位，必须由软件清零。

10.3.2 中断

SSC 的中断类型由表 10-4 给出。

表 10-4 SSC 中断源

中断	信号	描述
发送开始	TIR	指示发送缓冲寄存器可重新装入新数据
发送结束	RIR	设定位数的数据已发送完成并移入接收缓冲寄存器
接收出错	EIR	当新的数据帧已接收完毕，但上个数据仍未从接收缓冲寄存器中读出时，产生该中断
相位出错	EIR	在移位时钟信号 SCLK 锁存时钟沿的一个周期之前和两个周期之后数据改变，产生该中断
波特率出错（仅针对从模式）	EIR	当输入时钟信号和设定的波特率之间偏差超过 100%时，产生该中断
发送出错（仅针对从模式）	EIR	主机启动传输，但从机的 TB 在上次传输后仍未更新，产生该中断

10.3.3 寄存器映射

SSC SFR 的地址见表 10-5。

表 10-5 SFR 地址列表

地址	寄存器名称
A9 _H	PISEL
AA _H	CONL
AB _H	CONH
AC _H	TBL
AD _H	RBL
AE _H	BRL
AF _H	BRH

10.3.4 寄存器描述

本章中描述的所有 SSC 寄存器名称，在本手册其他章节中引用时需加上模块名前缀“SSC_”，例如 SSC_PISEL。

10.3.4.1 端口输入选择寄存器

PISEL 寄存器控制 SSC 模块接收输入选择。

PISEL

端口输入选择寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
0					CIS	SIS	MIS
r					rw	rw	rw

符号	位序号	读写类型	功能描述
MIS	0	rw	主模式接收输入选择 0 主模式禁止接收输入 1 主模式使能接收输入
SIS	1	rw	从模式接收输入选择 0 从模式禁止接收输入 1 从模式使能接收输入
CIS	2	rw	从模式时钟输入选择 0 禁止时钟输入 1 使能时钟输入
0	[7:3]	r	保留 读操作返回 0；应写入 0

10.3.4.2 控制寄存器

串行通道 SSC 的工作模式由控制寄存器 CON 控制。该寄存器中包含用于模式和检错选择的控制位，以及用于错误识别的出错状态标志位。根据位 EN 的设置，使能控制功能或状态标志，使能主/从模式控制。

CON.EN = 0: 编程模式

CONL

控制寄存器，低位字节

复位值: 00_H

7	6	5	4	3	2	1	0
LB	PO	PH	HB	BM			
rw	rw	rw	rw	rw			

符号	位序号	读写类型	功能描述
BM	[3:0]	rw	数据宽度选择 0000 保留。不使用该组合 0001 - 0111 传输数据宽度为 2...8 (<BM>+1) <i>注: BM[3]固定为 0</i>
HB	4	rw	报文头控制 0 先发送/接收 LSB 1 先发送/接收 MSB
PH	5	rw	时钟相位控制 0 在时钟的前沿移位输出发送数据，后沿锁存接收数据 1 在时钟的前沿锁存接收数据，后沿移位输出发送数据
PO	6	rw	时钟极性控制 0 时钟线低电平空闲，时钟前沿为低到高跳变 1 时钟线高电平空闲，时钟前沿为高到低跳变
LB	7	rw	回环控制

符号	位序号	读写类型	功能描述
			0 正常输出 1 接收输入和发送输出相连（半双工模式）

CONH

控制寄存器，高位字节

复位值: 00H

7	6	5	4	3	2	1	0
EN	MS	0	AREN	BEN	PEN	REN	TEN
rw	rw	r	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
TEN	0	rw	发送出错中断使能 0 禁止发送出错中断 1 使能发送出错中断
REN	1	rw	接收出错使能 0 禁止接收出错中断 1 使能接收出错中断
PEN	2	rw	相位出错使能 0 禁止相位出错中断 1 使能相位出错中断
BEN	3	rw	波特率出错使能 0 禁止波特率出错中断 1 使能波特率出错中断
AREN	4	rw	自动复位使能 0 波特率出错时无操作 1 波特率出错时 SSC 被自动复位
MS	6	rw	主机选择 0 从模式。从 SCLK 接收移位时钟 1 主模式。产生移位时钟并从 SCLK 输出

符号	位序号	读写类型	功能描述
EN	7	rw	使能位 = 0 禁止发送和接收。存取控制位。
0	5	r	保留 读操作返回 0；应写入 0

CON.EN = 1: 工作模式
CONL
控制寄存器，低位字节
复位值: 00H

7	6	5	4	3	2	1	0
0				BC			
r				rh			

符号	位序号	读写类型	功能描述
BC	[3:0]	rh	位计数域 0001 - 1111 每次移位时更新移位计数器
0	[7:4]	r	保留 读操作返回 0; 应写入 0

CONH
控制寄存器，高位字节
复位值: 00H

7	6	5	4	3	2	1	0
EN	MS	0	BSY	BE	PE	RE	TE
rw	rw	r	rh	rwh	rwh	rwh	rwh

符号	位序号	读写类型	功能描述
TE	0	rwh	发送出错标志 0 发送未出错 1 从机缓冲寄存器未更新即开始传输数据
RE	1	rwh	接收出错标志 0 接收未出错 1 接收缓冲寄存器的内容在读出之前，新数据已接收完毕
PE	2	rwh	相位出错标志

符号	位序号	读写类型	功能描述
			0 相位未出错 1 接收数据在采样时钟沿附近改变
BE	3	rwh	波特率出错标志 0 波特率未出错 1 从机的实际波特率超过期望波特率的 2 倍或低于 0.5 倍
BSY	4	rh	忙碌标志 传输过程中被置位
MS	6	rw	主机选择 0 从模式。从 SCLK 接收移位时钟 1 主模式。产生移位时钟并从 SCLK 输出
EN	7	rw	使能位 = 0 发送和接收被使能。存取状态标志和主机/从机控制。
0	5	r	保留 读操作返回 0；应写入 0

注：访问 CON 之前先由 CON.EN 决定存取 CON 的目的（控制位或标志位）；也就是说，在编程模式下（CON.EN=0）在 CON 中写入 C057_H将初始化 SSC（CON.EN 为 0），然后开启 SSC 工作（CON.EN=1）。当写入 CON 时，确保保留位写入 0。

10.3.4.3 波特率定时器重载寄存器

SSC 的波特率定时器重载寄存器 BR 中包含波特率定时器的 16 位重载值。

BRL

波特率定时器重载寄存器，低位字节

复位值: 00H



rw

BRH

波特率定时器重载寄存器，高位字节

复位值: 00H



rw

符号	位序号	读写类型	功能描述
BR_VALUE	BRL 的 [7:0], BRH 的 [7:0]	rw	波特率定时器/重载寄存器的值 读取 BR 返回波特率定时器 16 位计数值； 写入 BR 将 BR_VALUE 装入波特率定时器重载寄存器中。

10.3.4.4 发送和接收缓冲寄存器

SSC 的发送缓冲寄存器 TB 中存放发送数据。

TBL

发送缓冲寄存器，低位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
TB_VALUE	[7:0]	rw	发送数据寄存器的值 TB_VALUE 是将发送的数据值，发送时 TB 中未选中的位不予理睬。

SSC 的接收缓冲寄存器 RB 中存放接收数据。

RBL

接收缓冲寄存器，低位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
RB_VALUE	[7:0]	rw	接收数据寄存器的值 RB 中存放接收数据 RB_VALUE，RB 中未选中的位无效，不予理睬。

11 定时器

XC866 有 3 个 16 位定时器：定时器 0、定时器 1 和定时器 2，在许多定时应用中非常有用，例如测量事件之间的时间间隔、产生规律时间间隔的信号。

定时器 0 和定时器 1 工作特性：

- 四种工作模式：
 - 模式 0：13 位定时器
 - 模式 1：16 位定时器
 - 模式 2：带有自动重载的 8 位定时器
 - 模式 3：两个 8 位定时器

定时器 2 工作特性：

- 递增/递减计数可选择
- 16 位自动重载模式
- 单通道、16 位捕获模式

11.1 定时器 0 和定时器 1

定时器 0 和定时器 1 递增计数，每个机器周期、或每两个 PCLK 输入时钟周期定时器寄存器加 1。定时器 0 和定时器 1 均有四种工作模式适用于不同的应用场合。

11.1.1 定时器基本操作

寄存器 TCON 和 TMOD 控制定时器的操作。若使能某定时器（使定时器工作），置位控制位 TCON.TR_x。

注：本章中的“x”表示 0 或 1。

每个定时器由两个 8 位寄存器，TL_x（低位字节）和 TH_x（高位字节）组成，复位缺省值为 00_H。TCON.TR_x 置位或清零不影响定时器寄存器的内容。

定时器溢出

定时器溢出时，定时器溢出标志 TCON.TF_x 置位，并在中断使能控制位 IEN0.ET 置位时产生中断。进入中断服务程序后溢出标志被自动清零。

定时器 0 工作在模式 3 时，定时器 1 的控制位 TR1、TF1 和 ET1 被 TH0 占用，见 11.1.2.4。

外部控制

定时器的开启/关闭不仅可以由软件控制，还可以由外部端口控制。必须首先设置寄存器 EXICON0，使 EXINTx 不通过中断管理单元的边沿检测电路。若定时器被使能 (TCON.TR_x = 1) 且 TMOD.GATE_x 置位，只有外部中断信号 EXINTx = 1 时定时器才工作，以便用于测量脉宽。但模式 3 下的定时器 1 操作不适用外部控制。

若 TMOD.GATE_x 被清零，定时器将只由软件控制。

11.1.2 定时器工作模式

定时器 0 和定时器 1 完全兼容，均可设定为四种不同的工作模式，如表 11-1 所示。寄存器 TMOD 的位域 T_xM 选择定时器的模式。

两个定时器在模式 0、1 和 2 时独立工作；在模式 3 时具有特定功能。

表 11-1 定时器 0 和定时器 1 工作模式

工作模式	功能描述
模式 0	13 位定时器 该定时器是一个带有 32 预分频的 8 位计数器。包含此工作模式仅仅是为了和 Intel 的 8048 器件兼容。
模式 1	16 位定时器 定时器寄存器 TLx 和 THx 级联构成一个 16 位计数器。
模式 2	具有自动重载的 8 位定时器 定时器寄存器 TLx 溢出时，THx 中用户定义的 8 位数据自动重新装入 TLx。
模式 3	定时器 0 作为两个独立的 8 位定时器工作 定时器寄存器 TLO 和 TH0 作为两个独立的 8 位定时器。即使定时器 1 被使能，它仍然停止计数、保持原先的计数值。

11.1.2.1 模式 0

在模式 0 时，定时器 0 或定时器 1 是带有 32 预分频的 8 位定时器，其结构如图 11-1 所示。

该模式下，定时器寄存器配置为 13 位寄存器。当计数值从全“1”翻转为全“0”时溢出标志 TF_x 置位，该标志可请求中断。当 $TR_x = 1$ 且 $GATE_x = 0$ 或 $EXINT_x = 1$ ($GATE_x = 1$ 时，允许定时器由外部输入 $EXINT$ 控制，以便用于脉宽测量) 时，定时器计数。 TR_x 是寄存器 $TCON$ 中的控制位； $GATE_x$ 位于寄存器 $TMOD$ 中。

13 位寄存器由 TH_x 的全 8 位和 TL_x 的低 5 位组成， TL_x 的高 3 位不起作用，可将其忽略。置位运行标志 (TR_x) 不对该寄存器清零。

模式 0 操作对于定时器 0 和定时器 1 是相同的。

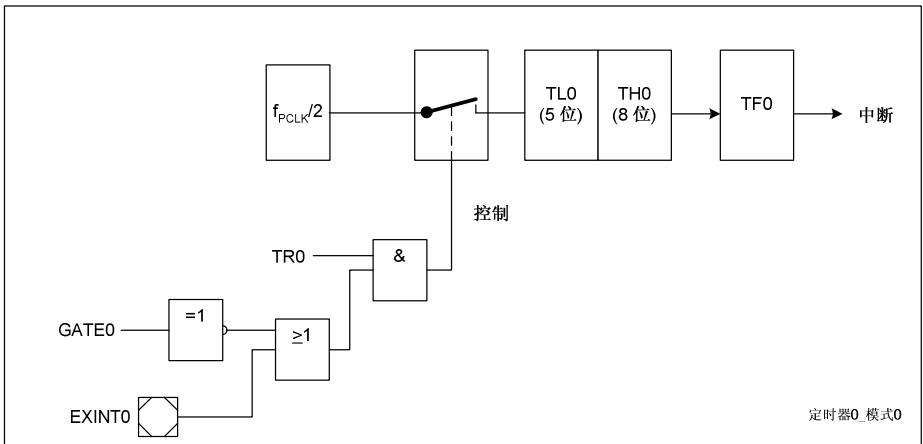


图 11-1 定时器 0，模式 0：13 位定时器

11.1.2.2 模式 1

模式 1 除了使用全部 16 位外，结构和模式 0 相同，见图 11-2。

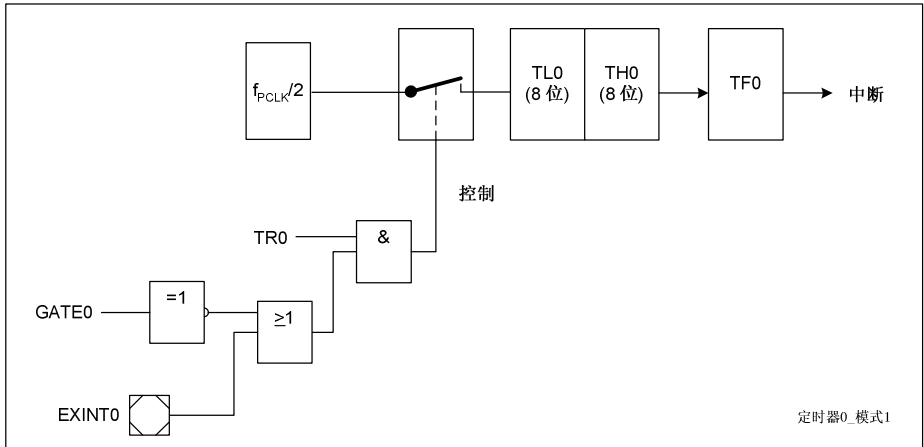


图 11-2 定时器 0，模式 1：16 位定时器

11.1.2.3 模式 2

模式 2 下，将定时器设置为可自动重载的 8 位计数器，定时器 0 的结构如图 11-3 所示。

TL_x 的溢出不仅置位 TF_x，而且将 TH_x 的值重新装入 TL_x，TH_x 的内容由软件预置，重载时 TH_x 内容不变。

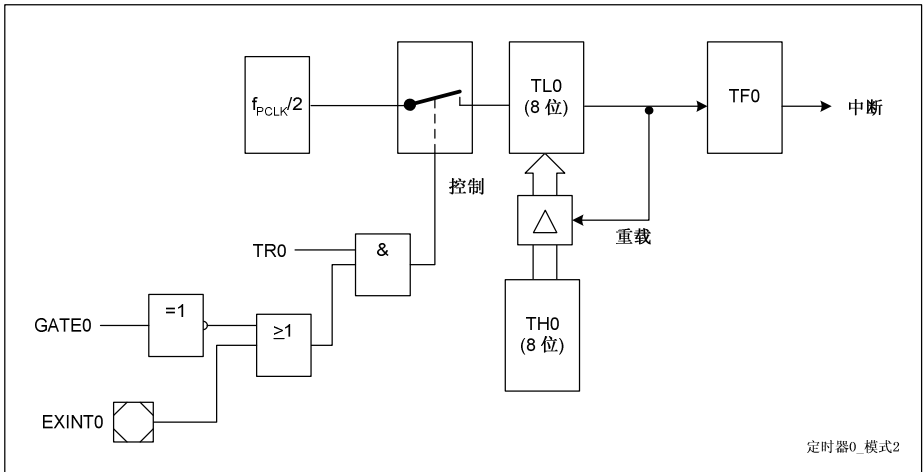


图 11-3 定时器 0，模式 2：8 位自动重载定时器

11.1.2.4 模式 3

模式 3 下，定时器 0 和定时器 1 的操作不同。定时器 0 的 TL0 和 TH0 作为两个独立的 8 位计数器；定时器 1 只保持原先计数值，效果与设置 TR1 = 0 相同。

模式 3 时的定时器 0 逻辑如图 11-4 所示。TL0 占用定时器 0 的控制位：GATE0，TR0 和 TF0，而 TH0 限定为定时器功能（计数机器周期），占用定时器 1 的控制位 TR1 和 TF1，TH0 溢出时将置位 TF1，并且在 ET1 置位时产生中断。

模式 3 可用于需要一个额外 8 位定时器的场合。当定时器 0 工作在模式 3 且 TR1 置位时，切换到模式 3 以外的任何其他模式可开启定时器 1、切换到模式 3 时定时器 1 关闭。

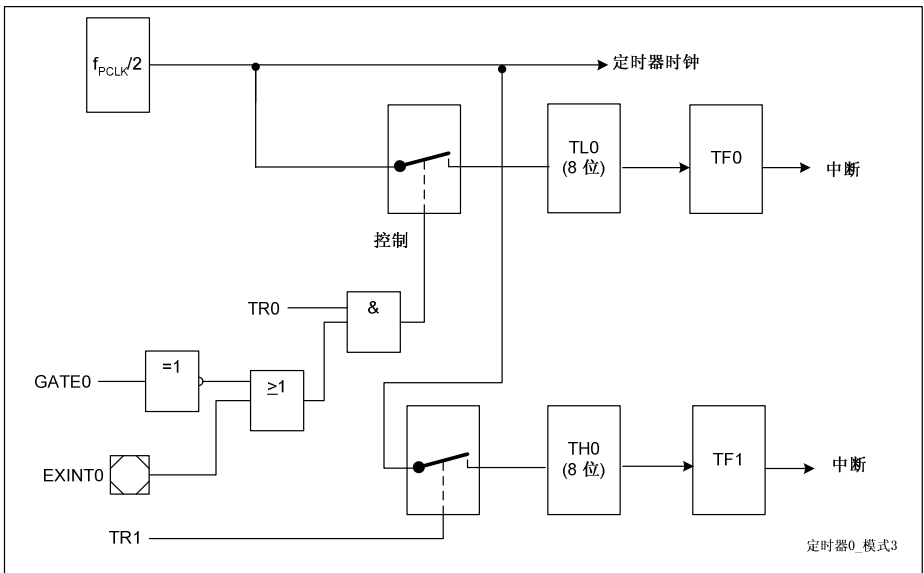


图 11-4 定时器 0，模式 3：两个 8 位定时器

11.1.3 寄存器映射

7 个 SFR 控制定时器 0 和定时器 1 的操作。可在标准（非映射）SFR 区和映射 SFR 区存取这些寄存器。

表 11-2 列出这些 SFR 的地址。

表 11-2 SFR 地址列表

地址	寄存器名称
88 _H	TCON
89 _H	TMOD
8A _H	TL0
8B _H	TL1
8C _H	TH0
8D _H	TH1

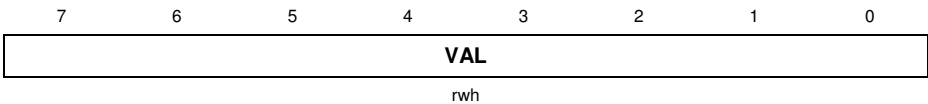
11.1.4 寄存器描述

根据工作模式不同，定时器 0 和定时器 1 的低位字节和高位字节寄存器可组成一个 16 位定时器。

TL_x (x = 0 - 1)

定时器 x 寄存器, 低位字节

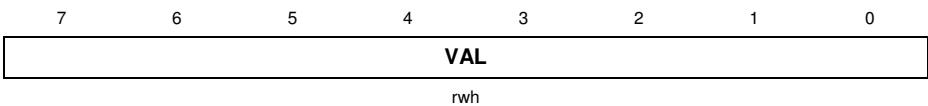
复位值: 00H



TH_x (x = 0 - 1)

定时器 x 寄存器, 高位字节

复位值: 00H



符号	位序号	读写类型	功能描述	
TL_x.VAL (x = 0 - 1)	[7:0]	rwh	定时器 0/1 低位字节寄存器	
			工作模式	功能描述
			0	“TL _x ” 保存 5 位预分频因子
			1	“TL _x ” 保存 16 位定时器的低 8 位
			2	“TL _x ” 保存 8 位定时器的值
3	TL0 保存 8 位定时器的值； TL1 未使用			

符号	位序号	读写类型	功能描述	
TH_x.VAL (x = 0,1)	[7:0]	rwh	定时器 0/1 高位字节寄存器	
			工作模式	功能描述
			0	“TH _x ” 保存 8 位计数器的值
			1	“TH _x ” 保存 16 位定时器中的高 8 位
			2	“TH _x ” 保存 8 位重载值
3	TH0 保存 8 位定时器的值； TH1 未使用			

寄存器 TCON 控制定时器 0 和定时器 1 的操作。

TCON

定时器控制寄存器

复位值: 00H

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
rw	rw	rw	rw	rw	rw	rw	rw

阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
TR0	4	rw	定时器 0 运行控制 0 定时器关闭 1 定时器运行
TF0	5	rw	定时器 0 溢出标志 当定时器 0 溢出时由硬件置位。当主机执行中断服务程序时溢出标志由硬件清零。
TR1	6	rw	定时器 1 运行控制¹⁾ 0 定时器关闭 1 定时器运行
TF1	7	rw	定时器 1 溢出标志 当定时器 1 ²⁾ 溢出时由硬件置位。当主机执行中断服务程序时溢出标志由硬件清零。

¹⁾ 若定时器 0 工作在模式 3，该控制位也影响 TH0

²⁾ 若定时器 0 工作在模式 3，TF1 由 TH0 置位

寄存器 TMOD 包含选择定时器 0 和定时器 1 工作模式的控制位。

TMOD
定时器模式寄存器
复位值: 00H

7	6	5	4	3	2	1	0
GATE1	0	T1M		GATE0	0	T0M	
rw	r	rw	rw	rw	r	rw	rw

符号	位序号	读写类型	功能描述										
T0M [1:0] T1M [5:4]	[1:0]	rw	模式选择位										
	[5:4]		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">T0M/T1M [1:0]</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>13 位定时器 (M8048 兼容模式)</td> </tr> <tr> <td>01</td> <td>16 位定时器</td> </tr> <tr> <td>10</td> <td>8 位自动重载定时器</td> </tr> <tr> <td>11</td> <td> 定时器 0: 定时器 0 被分成两个计数器。8 位定时器 TL0 由标准定时器 0 的控制位控制; 8 位定时器 TH0 由标准定时器 1 的控制位控制。 定时器 1: 保存 TH1 和 TL1 的内容 (定时器 1 停止工作)。 </td> </tr> </tbody> </table>	T0M/T1M [1:0]	功能	00	13 位定时器 (M8048 兼容模式)	01	16 位定时器	10	8 位自动重载定时器	11	定时器 0: 定时器 0 被分成两个计数器。8 位定时器 TL0 由标准定时器 0 的控制位控制; 8 位定时器 TH0 由标准定时器 1 的控制位控制。 定时器 1: 保存 TH1 和 TL1 的内容 (定时器 1 停止工作)。
	T0M/T1M [1:0]		功能										
	00		13 位定时器 (M8048 兼容模式)										
	01		16 位定时器										
10	8 位自动重载定时器												
11	定时器 0: 定时器 0 被分成两个计数器。8 位定时器 TL0 由标准定时器 0 的控制位控制; 8 位定时器 TH0 由标准定时器 1 的控制位控制。 定时器 1: 保存 TH1 和 TL1 的内容 (定时器 1 停止工作)。												
GATE0	3	rw	定时器 0 门控标志 0 仅当 TCON.TR0 = 1 时定时器 0 才运行 (软件控制)。 1 仅当引脚 EXINT0 = 1 (硬件控制) 且 TCON.TR0 = 1 时定时器 0 才运行。										
GATE1	7	rw	定时器 1 门控标志										

符号	位序号	读写类型	功能描述
			0 仅当 TCON.TR1 = 1 时定时器 1 才运行（软件控制）。 1 仅当引脚 EXINT1 = 1（硬件控制）且 TCON.TR1 = 1 时定时器 1 才运行。
0	2, 6	r	保留 读操作返回 0；应写入 0


IEN0 寄存器中包含定时器 0 和定时器 1 的中断使能控制位。

IEN0

中断使能寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
EA	0	ET2	ES	ET1	EX1	ET0	EX0
rw	r	rw	rw	rw	rw	rw	rw

 阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
ET0	1	rw	定时器 0 溢出中断使能 0 禁止定时器 0 中断 1 使能定时器 0 中断
ET1	3	rw	定时器 1 溢出中断使能¹⁾ 0 禁止定时器 1 中断 1 使能定时器 1 中断

¹⁾ 当定时器 0 工作在模式 3，该中断表明定时器 0 的 TH0 寄存器溢出。

11.2 定时器 2

定时器 2 是一个 16 位通用定时器，有两种操作模式：16 位自动重载模式和 16 位单通道捕获模式。如果预分频功能被禁止，定时器 2 以 PCLK/12 的输入时钟计数。

11.2.1 自动重载模式

寄存器 T2CON 中的 CP/RL2 置 0 时，选择自动重载模式。该模式下，定时器 2 计数至溢出后，将一个 16 位初始值重新装入定时器寄存器中，开始新一轮计数循环。置位寄存器 T2CON 的 TF2 表示计数溢出，从而向 CPU 发送中断请求信号。溢出标志 TF2 必须由软件清零。

根据寄存器 T2MOD 中 DCEN 控制位的设置，自动重载模式可进一步分为两种类型。

11.2.1.1 禁止递增/递减计数

若 DCEN = 0，则递增/递减计数选择被禁止，因此定时器只能递增计数。工作原理如图 11-5 所示。

若寄存器 T2CON 中的 EXEN2 = 0，置位 TR2 定时器开始递增计数，计数至最大值 FFFF_H 后溢出并置位 TF2，将寄存器 RC2 中的 16 位重载值重新装入定时器寄存器。该重载值在溢出前已由软件设置。新一轮计数循环开始，定时器同上一轮计数循环一样，从重载值开始递增计数。

若 EXEN = 1，置位 TR2 定时器开始递增计数至最大值 FFFF_H。计数溢出或输入引脚 T2EX 的负/正跳变（由寄存器 T2MOD 的位 EDGESEL 选择）均会引起 16 位重载，将寄存器 RC2 的内容重新装入定时器寄存器。若由溢出引起重载，溢出标志 TF2 置位。若由引脚 T2EX 的负/正跳变引起重载，寄存器 T2CON 中的 EXF2 置位。这两种情况均产生中断，定时器进入下一轮计数循环。EXF2 标志和 TF2 一样必须由软件清零。

注：当引脚 T2EX 用于定时器 2 时，必须置位 BCON.T2EXIS。

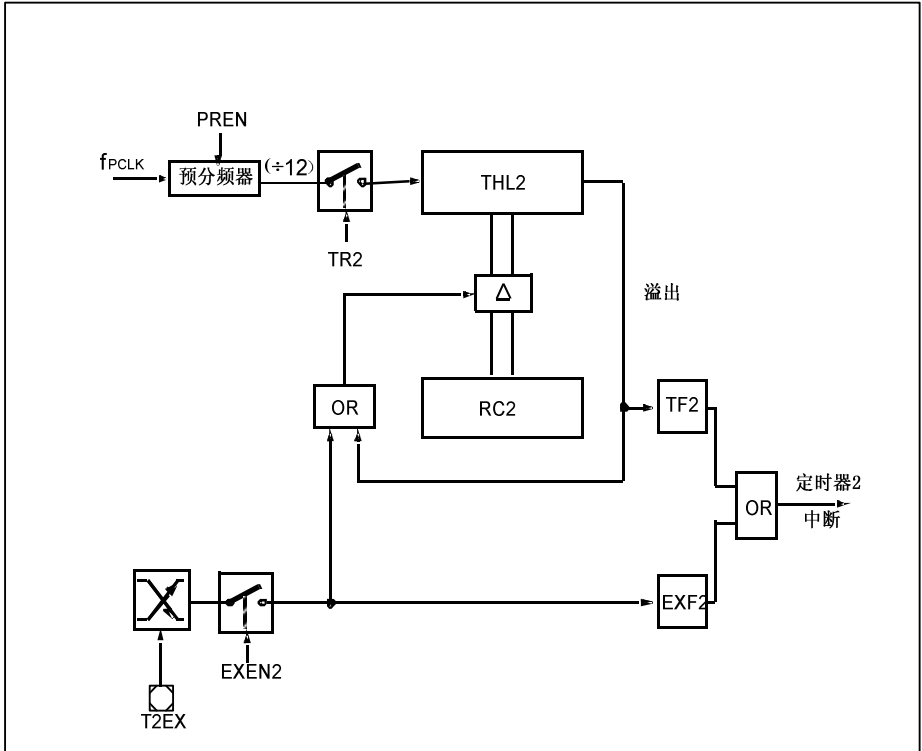


图 11-5 自动重载模式 (DCEN = 0)

11.2.1.2 使能递增/递减计数

若 $DCEN = 1$ ，可进行递增/递减计数选择。计数方向由输入引脚 T2EX 的电平决定。工作原理如图 11-6 所示。

引脚 T2EX 的逻辑电平为 1 时定时器 2 递增计数，因此定时器递增计数，计数至最大值 $FFFF_H$ 后溢出并置位 TF2，RC2 寄存器的 16 位重载值重新装入定时器寄存器。新一轮计数循环开始，定时器同上一轮计数循环一样，从重载值开始递增计数。该重载值在溢出前已由软件设置。

引脚 T2EX 的逻辑电平为 0 时定时器 2 递减计数。定时器递减计数并当 THL2 的值和寄存器 RC2 中的值相等时发生下溢。下溢后置位 TF2，并将值 $FFFF_H$ 重新载入定时器寄存器 THL2 中。新一轮计数循环开始，定时器同上一轮计数循环一样递减计数。

该模式下，定时器 2 递增/递减产生溢出时 EXF2 翻转，但该标志不会产生中断请求。

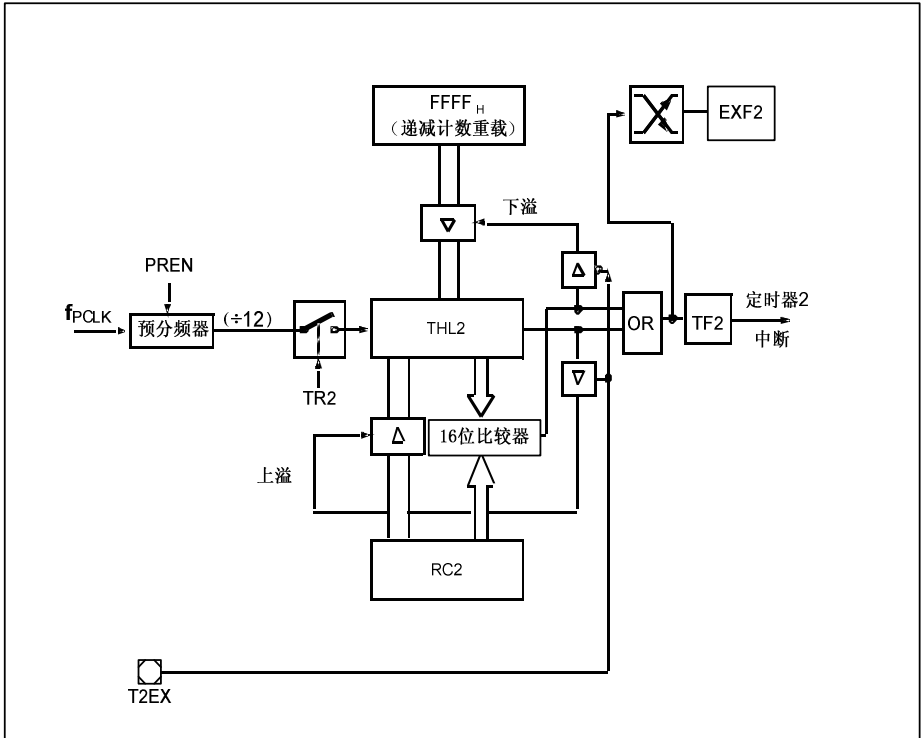


图 11-6 自动重载模式 (DCEN = 1)

11.2.2 捕获模式

置位寄存器 T2CON 中的 CP/RL2 和 EXEN2 进入 16 位捕获模式。该模式下，递减计数功能必须禁止。定时器是 16 位计数器并始终递增计数，计数至最大值 FFFF_H 后溢出，置位 TF2 并将 0000_H 重新装入定时器寄存器。TF2 置位会向 CPU 发送中断请求。

此外，在引脚 T2EX 的下降沿/上升沿（由 T2MOD.EDGESEL 选择），将定时器寄存器（THL2）的值捕获到寄存器 RC2 中。如果在计数器加 1 时检测到捕获信号，计数器先加 1 然后执行捕获操作，从而确保总能捕获到定时计数器的最新值。

执行完捕获操作，EXF2 置位、可用来产生中断请求。图 11-7 描述了定时器 2 的捕获功能。

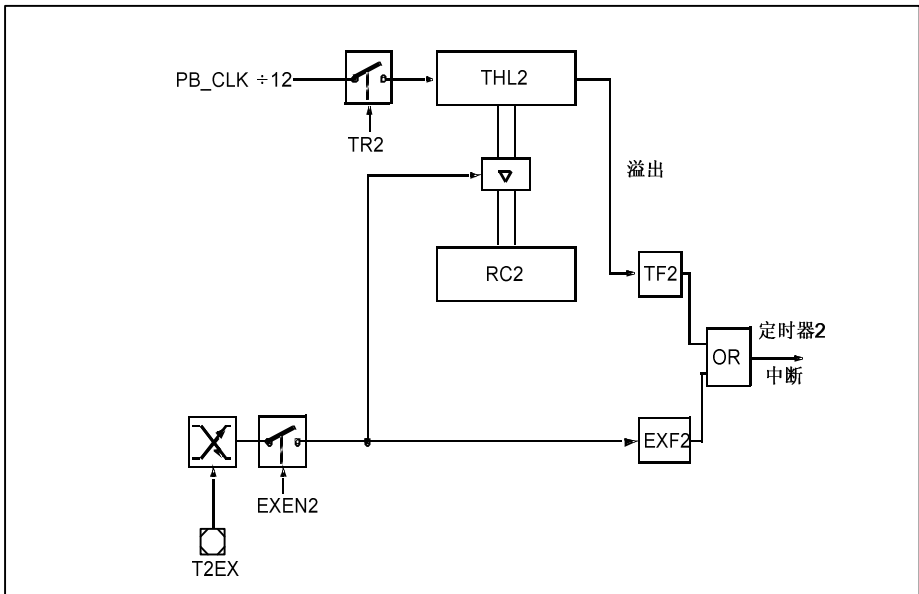


图 11-7 捕获模式

11.2.3 寄存器映射

本章中描述的所有定时器 2 寄存器名称，在本手册其他章节中引用时需加上模块名前缀“T2_”，例如 T2_T2CON。

定时器 2 的 SFR 位于标准（非映射）SFR 区。表 11-3 列出这些寄存器的地址。

表 11-3 SFR 地址列表

地址	寄存器名称
C0 _H	T2CON
C1 _H	T2MOD
C2 _H	RC2L
C3 _H	RC2H
C4 _H	T2L
C5 _H	T2H

11.2.4 寄存器描述

寄存器 T2MOD 用来设置定时器 2 工作在不同模式。

T2MOD

定时器 2 模式寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
0	EDGESEL	PREN	T2PRE		DCEN		
r	rw	rw	rw		rw		

符号	位序号	读写类型	功能描述
DCEN	0	rw	递增/递减计数器使能 0 禁止递增/递减计数功能 1 使能递增/递减计数且由引脚 T2EX 控制计数方向（递增 = 1，递减 = 0）
T2PRE	[3:1]	rw	定时器 2 预分频选择 选择定时器 2 的输入时钟（来自外设时钟）

符号	位序号	读写类型	功能描述
			000 $f_{T2} = f_{PCLK}$ 001 $f_{T2} = f_{PCLK} / 2$ 010 $f_{T2} = f_{PCLK} / 4$ 011 $f_{T2} = f_{PCLK} / 8$ 100 $f_{T2} = f_{PCLK} / 16$ 其他：保留
PREN	4	rw	预分频使能 0 禁止预分频功能，2/12 分频生效 1 使能预分频功能（见 T2PRE），2/12 分频旁路
EDGESEL	5	rw	捕获模式/重载模式的边沿选择 0 选择引脚 T2EX 的下降沿 1 选择引脚 T2EX 的上升沿
0	[7:6]	r	保留 读操作返回 0；应写入 0

寄存器 T2CON 控制定时器 2 的工作模式。此外还包含中断状态标志。

T2CON
定时器 2 控制寄存器
复位值: 00H

7	6	5	4	3	2	1	0
TF2	EXF2	0		EXEN2	TR2	0	CP/RL2
rwh	rwh	r	rw		rwh	r	rw

符号	位序号	读写类型	功能描述
CP/RL2	0	rw	捕获/重载模式选择 0 计数器溢出或引脚 T2EX 发生负/正跳变时重载（当 EXEN = 1）。 1 引脚 T2EX 发生负/正跳变时，捕获定时器 2 数据寄存器的内容（当 EXEN = 1）。 由位 EDGESEL 选择引脚 T2EX 负跳变还是正跳变。
TR2	2	rwh	定时器 2 启动/停止控制 0 停止定时器 2 1 启动定时器 2
EXEN2	3	rw	定时器 2 外部使能控制 0 禁止外部事件 1 捕获/重载模式时使能外部事件
EXF2	6	rwh	定时器 2 外部事件标志 捕获/重载模式下，若 EXEN2 = 1，在引脚 T2EX 上有负/正跳变时该位由硬件置位，除非 DCEN = 1，否则产生中断请求。该位必须由软件清零。
TF2	7	rwh	定时器 2 上溢/下溢标志 定时器 2 上溢/下溢时置位。必须由软件清零

符号	位序号	读写类型	功能描述
0	1, [5:4]	r	保留 读操作返回 0; 应写入 0

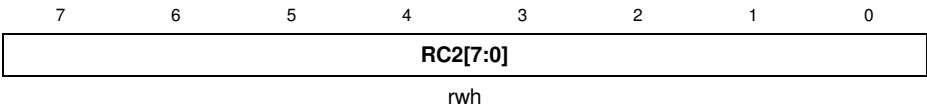
定时器

根据所选的工作模式不同，RC2 中存放定时器寄存器溢出时的 16 位重载值；或存放捕获到的当前定时器计数值。

RC2L

定时器 2 重载/捕获寄存器，低位字节

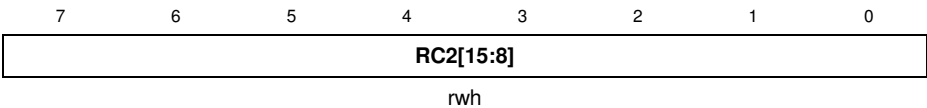
复位值: 00H



RC2H

定时器 2 重载/捕获寄存器，高位字节

复位值: 00H



符号	位序号	读写类型	功能描述
RC2	RC2L 的 [7:0], RC2H 的 [7:0]	rwh	<p>重载/捕获值</p> <p>若 $\overline{CP/RL2} = 0$，溢出时该寄存器的值装入定时器寄存器。</p> <p>若 $\overline{CP/RL2} = 1$，当 $EXEN2 = 1$，引脚 T2EX 发生负/正跳变时当前定时器计数值装入该寄存器。</p>

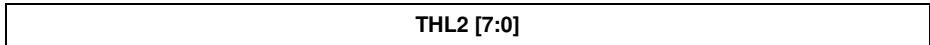
寄存器 T2 保存定时器 2 当前 16 位计数值。

T2L

定时器 2 寄存器，低字节

复位值: 00H

7 6 5 4 3 2 1 0



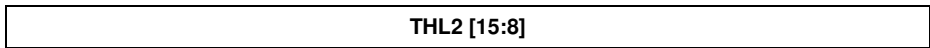
rwh

T2H

定时器 2 寄存器，高字节

复位值: 00H

7 6 5 4 3 2 1 0



rwh

符号	位序号	读写类型	功能描述
THL2	T2L 的 [7:0], T2H 的 [7:0]	rwh	定时器 2 计数值 该寄存器存放当前定时器的计数值

12 捕获/比较单元 6

捕获/比较单元 6 (CCU6) 中有两个独立的定时器 (T12, T13)、用来产生专门控制交流电机的脉宽调制 (PWM) 信号。CCU6 也支持块切换和多相电机的专用控制模式。CCU6 的方框图如图 12-1 所示。

定时器 T12 的 3 路通道可用作捕获和/或比较模式。定时器 T13 只工作在比较模式。

多通道控制单元产生的输出序列可由 T12 和/或 T13 调制。调制源可选择并组合使用。

定时器 T12 特性:

- 3 路捕获/比较通道，每路可用作捕获或比较
- 支持产生三相 PWM (6 路输出，每路信号对应上桥臂或下桥臂开关)
- 16 位精度，最大计数频率 = 外设时钟频率
- 单通道死区时间控制，避免功率级短路
- 同步刷新 T12/T13 寄存器
- 产生中间对齐和边缘对齐 PWM
- 支持单次模式
- 支持多中断请求源
- 类磁滞控制模式

定时器 T13 特性:

- 单一独立比较通道，单输出
- 16 位精度，最大计数频率 = 外设时钟频率
- 可与 T12 同步
- 周期 - 匹配和比较 - 匹配产生中断
- 支持单次模式

附加特性:

- 实现无刷直流驱动的块切换功能
- 利用霍尔序列进行位置检测
- 用于块切换的自动转速测量
- 综合错误处理

- 通过外部信号 $\overline{\text{CTRAP}}$ 快速紧急终止，无需 CPU 干预
- 用于多通道交流驱动的控制模式
- 输出电平可选，与功率级适配

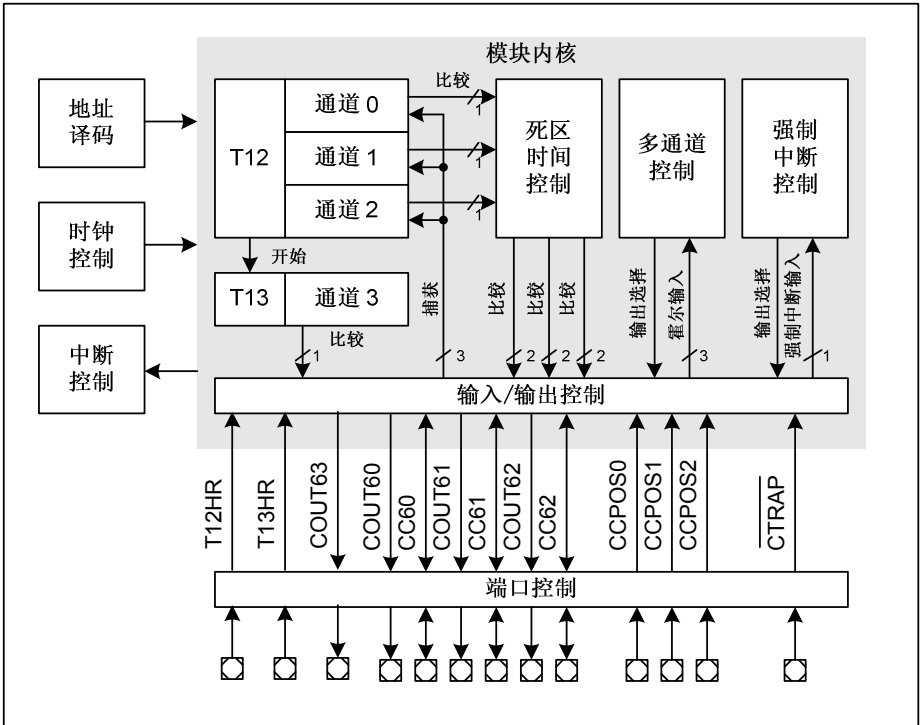


图 12-1 CCU6 方框图

12.1 功能描述

12.1.1 定时器 T12

定时器 T12 的三路通道用作捕获/比较模式。定时器 T12 的输入时钟范围由位域 T12CLK 设置，可从 f_{CCU6} 最大至 $f_{CCU6}/128$ 。为了支持更高的时钟频率，T12PRE = 1 时，可使用附加的分频因子 1/256 用于 T12 分频。

定时器的周期值、比较值、被动态选择位，以及被动态电平选择位均写入映射寄存器中，并非直接写入对应的实际寄存器；读操作直接访问实际寄存器（三路比较通道除外，它们的实际寄存器和映射寄存器均可读取）。置位映射传输使能位 STE12 将使能从映射寄存器到实际寄存器的传输。

如果映射传输使能，定时器下次回零时（边沿对齐模式下定时器被清零；或中间对齐模式下定时器递减计数到 1），映射寄存器的内容立刻被复制到对应的实际寄存器中。定时器 T12 工作在中间对齐模式时，当定时器（递增）计数到当前设定的周期值，映射寄存器的内容也将被复制到对应的实际寄存器中（如果已由 STE12 使能）。

当关闭定时器 T12，若 STE12 置位则立刻进行映射传输。传输结束位 STE12 将被自动清零。

定时器 T12 的功能概览如图 12-2 所示。

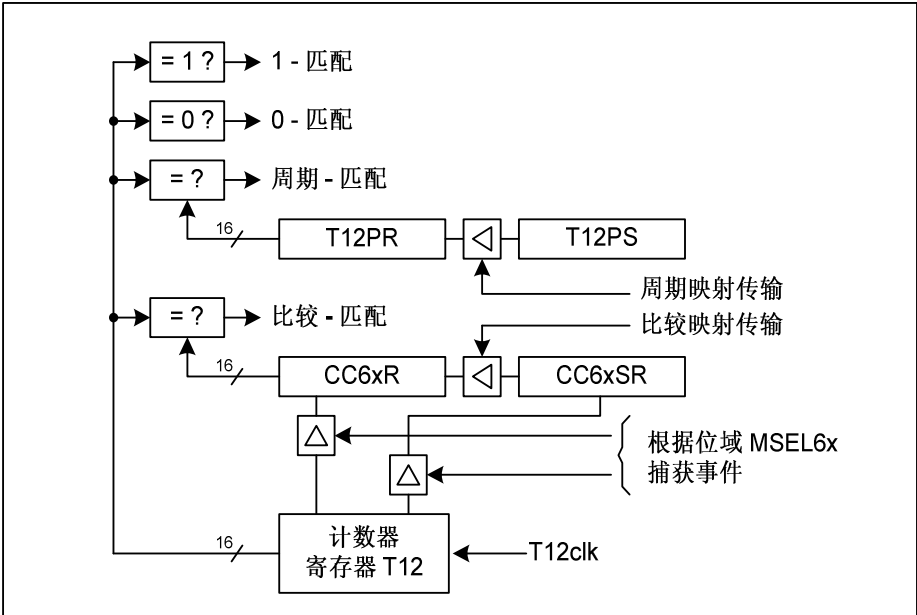


图 12-2 T12 功能概览

12.1.1.1 定时器设置

寄存器 T12 代表定时器 T12 的计数值。只有在定时器 T12 停止时才可写入；定时器 T12 工作时写操作无效。寄存器 T12 始终可由软件读取。

边沿对齐模式下，T12 递增计数；中间对齐模式下，T12 可递增和递减计数。

由硬件或软件控制 T12R 启动和停止定时器 T12。

- 位域 T12RSEL 定义了引脚 T12HR 上的事件：上升沿、下降沿、或二者任一，由硬件置位定时器运行控制位 T12R。
- 若位域 T12RSEL = 00_B，禁止由外部事件设置 T12R；定时器的运行位只能由软件控制。软件置位 T12RR 或 T12RS 复位或置位 T12R。
- 在单次模式下，根据位 T12SSC 定义的功能由硬件复位 T12R。若 T12SSC = 1，在下列情况硬件复位 T12R：
 - 边沿对齐模式下 T12 计数至周期值
 - 中间对齐模式下 T12 递减计数至 1

可置位 T12RES 将寄存器 T12 复位至 0。T12RES 的设置不影响运行位 T12R。

12.1.1.2 计数规则

以 T12 时钟作为输入，按如下计数规则定义计数序列：

T12 工作在边沿对齐模式下（位 CTM = 0）：

计数方向设置为递增计数（CDIR = 0）。检测到周期 - 匹配时计数器复位至 0，若 STE12 = 1 定时器 T12 发生映射寄存器传输。

T12 工作在中间对齐模式下（位 CTM = 1）：

- 递减计数检测到 1 - 匹配时，计数方向设置为递增计数（CDIR = 0）。
- 递增计数检测到周期 - 匹配时，计数方向设置为递减计数（CDIR = 1）。
- 若 STE12 = 1，在下列条件下发生映射寄存器传输：
 - 递增计数时检测到周期 - 匹配
 - 递减计数时检测到 1 - 匹配

定时器 T12 不工作时分频器复位，从而保证可重复产生同样的时序和延迟。

12.1.1.3 切换规则

三路比较通道并行工作。计数方向不同，比较 - 匹配的含义不同。为了使 PWM 信号和输出电平无关，引入两个不同的状态用于进行比较：主动态和被动态。用这两个状态产生所需的 PWM，该调制信号是 T13 的控制、强制中断控制和多通道控制的逻辑组合。如果将主动态视为 1、被动态视为 0，比较状态之间逻辑与组合。

- 主动态 AND 主动态 = 主动态
- 主动态 AND 被动态 = 被动态
- 被动态 AND 被动态 = 被动态

根据检测到的比较 - 匹配改变比较状态，由 CC6xST 表示。T12 的比较状态定义如下：

- 如果计数值低于比较值，为被动态
- 如果计数值高于比较值，为主动态

从而有以下的切换规则用于比较状态：

- 递增计数时，计数至比较值时，置位为主动态
- 递减计数时，计数至比较值时，复位为被动态
- 递增计数时，0 - 匹配但非比较 - 匹配时，复位为被动态
- 递增计数时，0 - 匹配同时也为比较 - 匹配时，置位为主动态

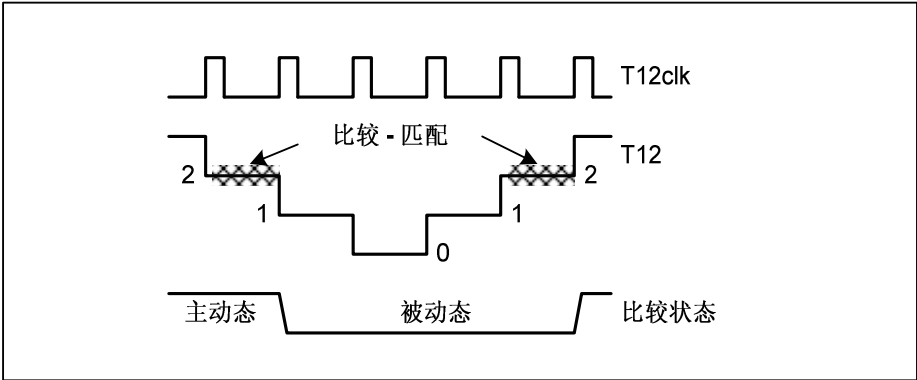


图 12-3 比较值 = 2 的比较状态

只有当定时器工作时才考虑切换规则。因此当定时器停止工作时，对定时器寄存器的写操作不会引起比较。

12.1.1.4 T12 的比较模式

比较模式下，寄存器 CC6xR (x = 0 - 2) 是 T12 的实际比较寄存器。CC6xR 中存放的值和 T12 的计数值进行比较（三路通道并行工作）。寄存器 CC6xR 只可由软件读取；通过映射寄存器传输来改变 CC6xR 的值，从寄存器 CC6xSR 中获取改变值。

寄存器 T12PR 包含定时器 T12 的周期值。周期值和 T12 的实际计数值比较，根据定义的计数规则执行相应的计数动作。

图 12-4 以中间对齐模式下无死区时间的比较模式为例。位 CC6xST 指示对应通道发生比较或捕获事件。下列事件将置位 CC6xST (若 CC6xST 当前为 0)：

- 软件置位 (MCC6xS)
- 若 T12 工作且 T12 的置位事件被使能，(T12 计数值大于比较值的) 比较置位事件
- 捕获置位事件

下列事件将复位 CC6xST (若 CC6xST 当前为 1)：

- 软件复位 (MCC6xR)
- 若 T12 工作且 T12 的复位事件被使能 (包括单次模式下 T12 计数到周期值时)，(T12 计数值低于比较值的) 比较复位事件
- 类磁滞控制模式下的复位事件

位 $CC6xPS$ 代表被动态选择位。定时器 $T12$ 的两条输出线 ($CC6x$, $COUT6x$) 可选择在 $CC6xST$ 为 0 时 ($CC6xPS = 0$) 或在 $CC6xST$ 为 1 时 ($CC6xPS = 1$) 处于被动态。

输出为被动态时所驱动的输出电平由位域 PSL 中的对应位来定义。

只有在定时器 $T12$ 工作时才可能硬件修改比较状态位。因此, 可用 $T12R$ 使能/禁止硬件修改。

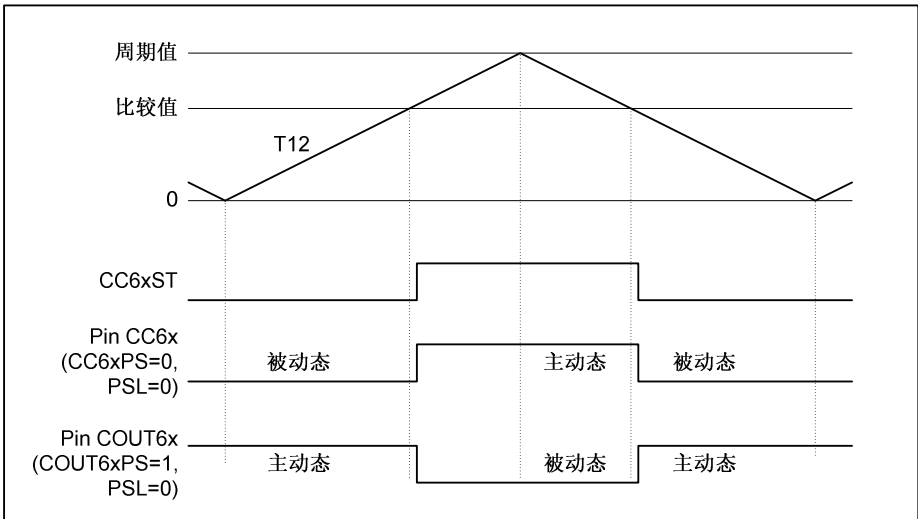


图 12-4 定时器 $T12$ 的比较状态

对于类磁滞控制模式 ($MSEL6x = 1001_B$) (见章节 12.1.1.9), 只有在输入 $CCPOSx = 1$ (无效) 时才可能设置比较状态位。

如果选择霍尔传感器模式 ($MSEL6x = 1000_B$) (见章节 12.1.6), 由定时器 $T12$ 修改比较通道 1 和 2 的比较状态位, 以指示设置时间已超时。

只有当 $CC6xST$ 被复位后才可置位; 只有被置位后才可复位。因此, 置位和复位 $CC6xST$ 的触发事件必须逻辑组合。置位和复位信号的逻辑或可触发死区时间计数器重载 (见图 12-5)。只有改变 $CC6xST$ 时才能触发重载, 产生正确的、带有死区时间的 PWM 信号; 边沿对齐和中间对齐模式下 PWM 的占空比在 0% 到 100% 之间。

12.1.1.5 0%和 100%的占空比

这些计数规则和切换规则确保了产生的 PWM 的占空比 (占空比 = 有效时间/PWM 总周期时间) 范围在 0% 到 100% 之间。为了获得 0% 的占空比 (比较状态永远无效),

比较值必须设置为 $T12P+1$ 的；比较值为 0 将会使 PWM 的占空比为 100%（比较状态始终有效）。

12.1.1.6 死区时间产生

大多数情况下，相互连接的功率开关的切换行为不对称，打开和关闭开关所需时间不同。若功率器件的开启时间小于关闭时间时通常会产生问题，这将导致反相器桥臂短路，整个系统将瘫痪。为了用硬件解决这个问题，CCU6 中提供了一个可编程死区时间计数器，可延迟切换信号从被动沿跳变至主动沿的时间（主动沿到被动沿不延迟）。

寄存器 T12DTC 控制定时器 T12 比较通道死区时间产生。每路通道可各自独立由位 DTEx 使能/禁止产生死区时间。如果允许产生死区时间，位域 DTM（以 T12CLK 为计数时钟的 8 位递减计数器）定义了从被动态跳变至主动态的延迟值。只有当死区时间计数器计数至 0 时才可被重载。

每路通道和各自的死区时间计数器、触发和使能信号一起独立工作。位域 DTM 的值对三路通道均有效。

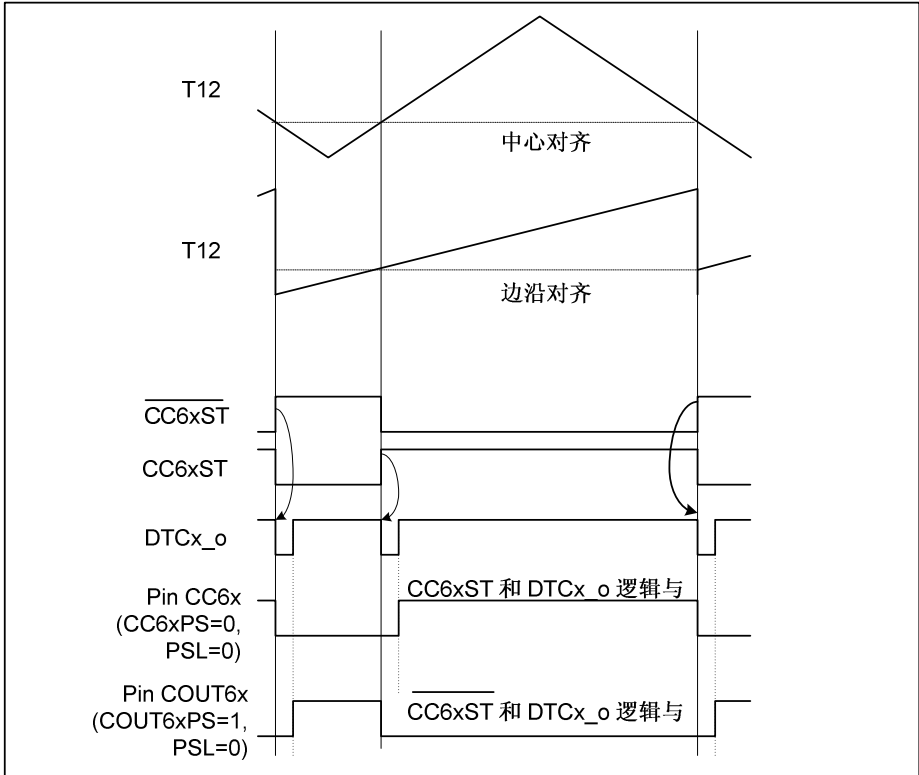


图 12-5 带有死区时间产生的 PWM 信号

12.1.1.7 捕获模式

捕获模式下，根据位域 MSEL6x 的选择，位 CC6xST 指示产生选中的捕获事件。

- MSEL6x = 01XX_B，双寄存器捕获模式（见表 12-5）
- MSEL6x = 101X_B 或 11XX_B，多输入捕获模式（见表 12-7）

可选择引脚 CC6x 或 CCPOSx 的上升沿 和/或 下降沿作为捕获事件，将定时器 T12 的内容捕获至寄存器 CC6xR 和 CC6xSR。要工作在捕获模式，捕获引脚必须设置为输入引脚。

由多种方式保存捕获值。双寄存器捕获模式，定时器的值保存到通道映射寄存器 CC6xSR 中；前次保存在该寄存器中的值立即被复制到通道寄存器 CC6xR 中。于是软件可检查新捕获的值，同时可能读取前次的捕获值。

注：捕获模式下，根据映射传输规则可请求映射传输，内容未改变的捕获/比较寄存器除外。

12.1.1.8 单次模式

当位 T12SSC 被置 1 时，选择定时器 T12 工作在单次模式。单次模式下，定时器 T12 在其计数周期结束时自动停止工作。图 12-6 示出在边沿对齐和中间对齐模式下，定时器周期结束时的行为。如果 T12SSC 被置位时检测到周期事件结束，位 T12R 和所有 CC6xST 被复位。

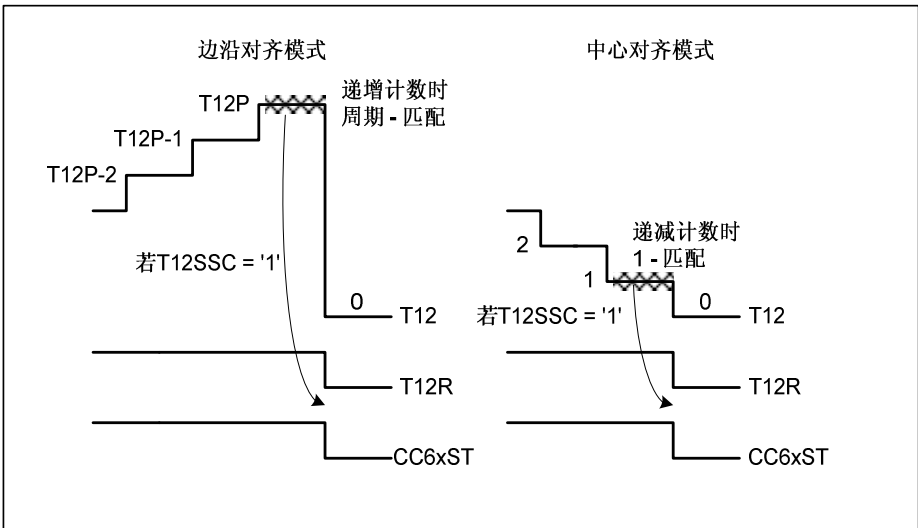


图 12-6 T12 单次模式

12.1.1.9 类磁滞控制模式

类磁滞控制模式 (MSEL6x = 1001_B) 提供了这样的可能性：如果输入 CCPOSx 变 0，通过复位 CC6xST 可关闭 PWM 输出。这可用作简单的电机控制特性，例如用比较器指示过流。当 CCPOSx = 0 时，相应通道的 PWM 输出驱动其被动电平。只有当 CCPOSx = 1 时才可能置位 CC6xST。图 12-7 给出类磁滞控制的例子。

可利用该模式将时间相关行为引入类磁滞控制器。标准的类磁滞控制器检测某值是否已超过极限，并根据比较结果关闭输出。根据运行条件，切换频率和占空比可一直改变。

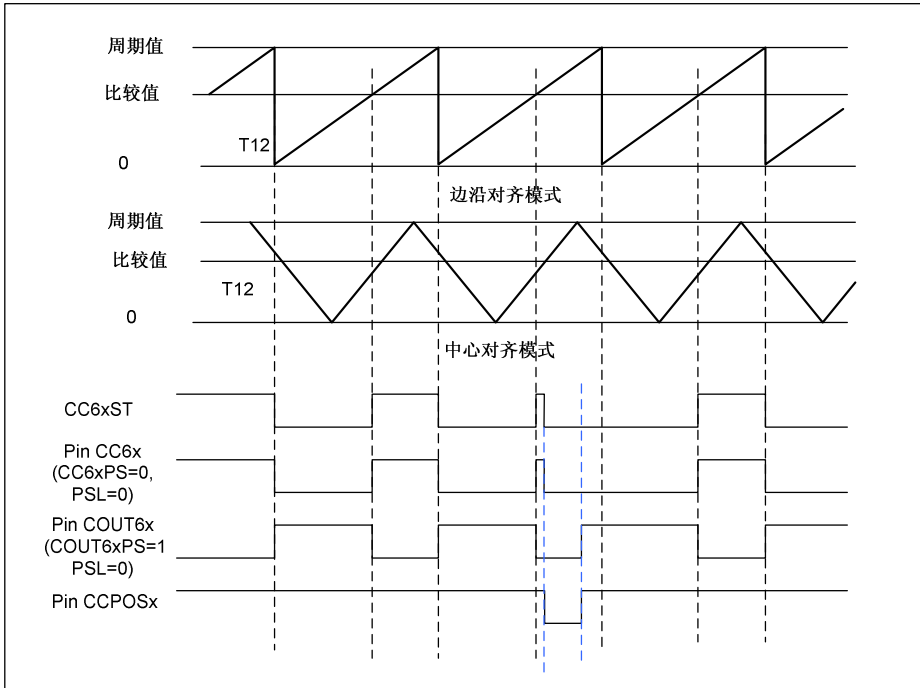


图 12-7 类磁滞控制模式

12.1.2 定时器 T13

定时器 T13 和定时器 T12 相似，差别仅在于 T13 只有一路比较通道。计数器只能递增计数（和 T12 的边沿对齐模式相似）。定时器 T13 的输入时钟范围可从 f_{CCU6} 最大至 $f_{CCU6}/128$ ，由位域 T13CLK 设置。为了支持更高的时钟频率，T13PRE = 1 时，可使用附加的分频因子 1/256 用作 T13 分频。

周期 - 匹配时，T13 的映射传输由位 STE13 使能。T13 映射传输时，寄存器 CC63SR 的内容被传送到寄存器 CC63R 中。这两个寄存器均可由软件读取，但只有映射寄存器可由软件写入。

位 CC63PS、T13IM 和 PSL63 具有映射位。T13 映射传输时，这些映射位的内容被传送给实际使用位。写操作写入映射位；读操作读取实际使用位的值。

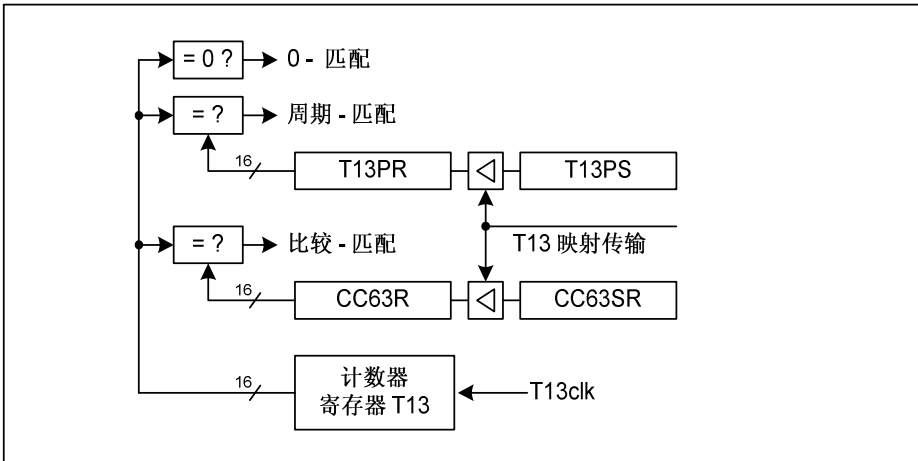


图 12-8 T13 功能概览

定时器 T13 计数和定时器 T12 在边沿对齐模式下的计数和切换规则相同。定时器 T13 的功能概览如图 12-8 所示。

12.1.2.1 定时器设置

寄存器 T13 代表定时器 T13 的计数值。只有在定时器 T13 停止时才可写入；定时器 T13 工作时写操作无效。寄存器 T13 始终可由软件读取。定时器 T13 只支持边沿对齐模式（递增计数）。

由硬件或软件控制 T13R 启动和停止定时器 T13。

- 软件置位 T13RR 或 T13RS 来复位或置位 T13R。

- 在单次模式下，若位 T13SSC = 1，当 T13 递增计数至周期值时，硬件复位 T13R。
- 位域 T13TEC 和 T13TED 选择触发事件置位 T13R，用于和不同的 T12 比较事件同步。

可通过置位 T13RES 将寄存器 T13 复位至 0。对 T13RES 的设置不会影响运行位 T13R。

12.1.2.2 比较模式

寄存器 CC63R 是 T13 的实际比较寄存器。CC63R 中保存的值和 T13 的计数值相比较。寄存器 CC63R 只可由软件读取；要更改 CC63R 的值，通过映射寄存器传输实现，从寄存器 CC63SR 中获取更改值。相应的映射寄存器 CC63SR 可由软件读写。

寄存器 T13PR 包含了定时器 T13 的周期值。周期值和 T13 的实际计数值比较，根据定义的计数规则执行相应的计数动作。

位 CC63ST 指示相应通道发生比较事件。下列事件将置位 CC63ST（若 CC63ST 当前为 0）：

- 软件置位（MCC63S）
- 若 T13 在运行且 T13 的置位事件被使能，（T13 计数值大于比较值）的比较置位事件

下列事件将复位 CC63ST（若 CC63ST 当前为 1）：

- 软件复位（MCC63R）
- 若 T13 在运行且 T13 的复位事件被使能（包括单次模式下 T13 周期结束时），（T13 计数值低于比较值）的比较复位事件

用 T13 的 PWM 信号调制其他输出信号。为了使 COUT63 和内部的调制信号无关，可由位 T13IM 和 COUT63PS 独立选择比较状态。

12.1.2.3 单次模式

当位 T13SSC 被置 1 时，选择定时器 T13 工作在单次模式。单次模式下，定时器 T13 在其计数周期结束时自动停止工作。如果 T13SSC 被置位时检测到周期事件结束，位 T13R 和 CC63ST 被复位。

12.1.2.4 T13 与 T12 同步

定时器 T13 可由 T12 的事件来同步。这些事件包括：

- 通道 0 的 T12 比较事件
- 通道 1 的 T12 比较事件
- 通道 2 的 T12 比较事件

- 通道 0、1 或 2 上任意 T12 比较事件
- T12 周期 - 匹配
- T12 0 - 匹配（递增计数）
- 输入 CCPOSx 的任意升降沿

位域 T13TEC 和 T13TED 选择用来启动定时器 T13 的事件。该事件硬件置位 T13R，T13 开始计数。和单次模式相结合，该特性可用来产生一个 T12 事件之后、时间可编程设定的延迟。

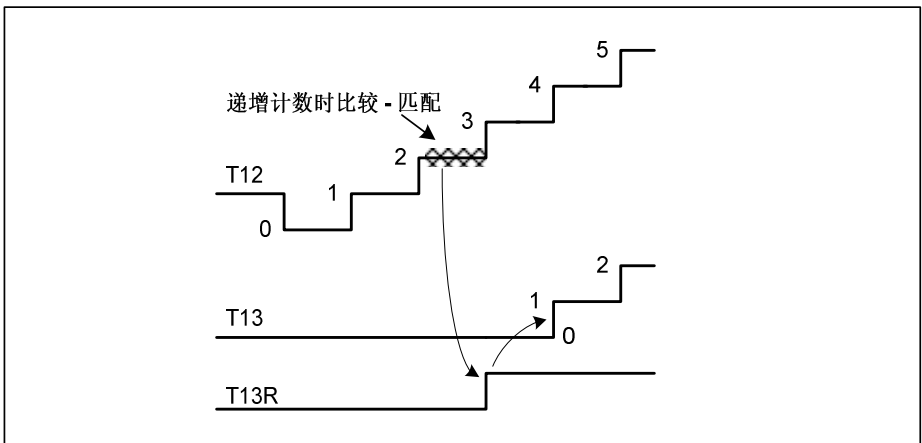


图 12-9 T13 与 T12 同步

图 12-9 所示为 T13 同步至 T12 的事件，图中举例选择的事件为递增计数时的比较 - 匹配（比较值 = 2）。T12 和 T13 的时钟可不同（使用其他分频因子）。为了简化，图中举例的 T12CLK 和 T13CLK 相等。

12.1.3 调制控制

调制控制部分将不同的调制源逻辑组合（CC6x_T12_o 和 COUT6x_T12_o 是 CC6xPS/COUT6xPS 设置后的输出信号；MOD_T13_o 是 T13 翻转调制（T13IM）之后的输出信号）。每种调制源可各自独立被使能用作输出。另外，强制中断功能也归入调制源，在强制中断状态下（若被使能）用来禁止相应输出线上的调制。

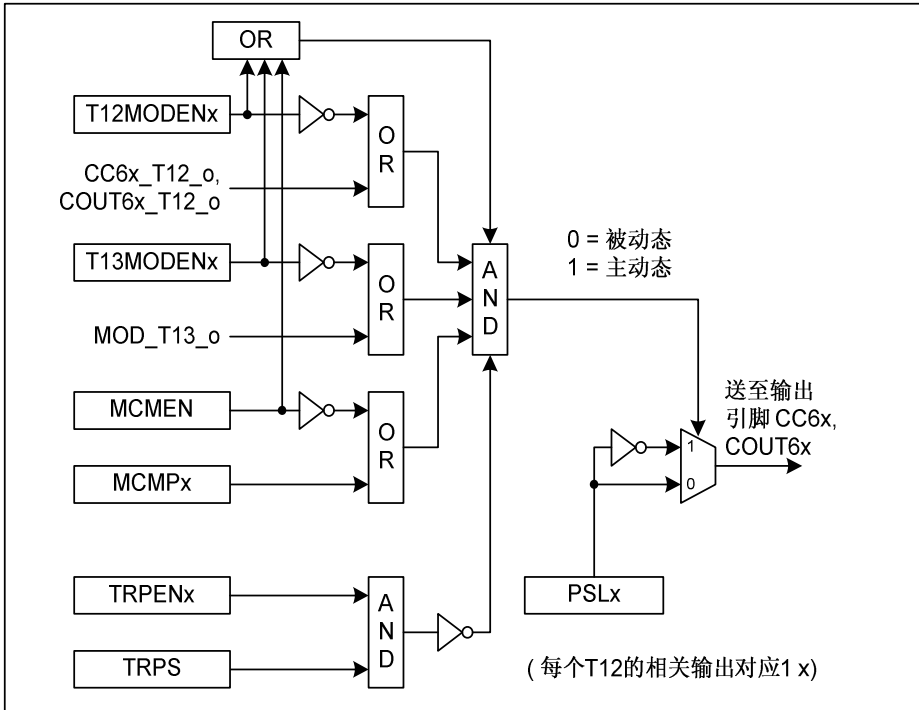


图 12-10 T12 相关输出的调制控制

对于图 12-10 所示 6 条 T12 相关输出线（由“x”表示）中的每条：

- T12MODENx 允许用定时器 T12 产生的 PWM 序列进行调制
- T13MODENx 允许用定时器 T13 产生的 PWM 序列进行调制
- MCMPx 选择多通道序列
- TRPENx 使能强制中断功能
- PSLx 定义了输出为被动态时所驱动的输出电平

如图 12-11 所示，T13 相关输出 COUT63 的调制控制部分将 T13 的输出信号（COUT63_T13_o 是 COUT63PS 设置后的输出信号）、使能位 ECT130 和强制中断功能逻辑组合。由位 PSL63 选择被动态的输出电平。

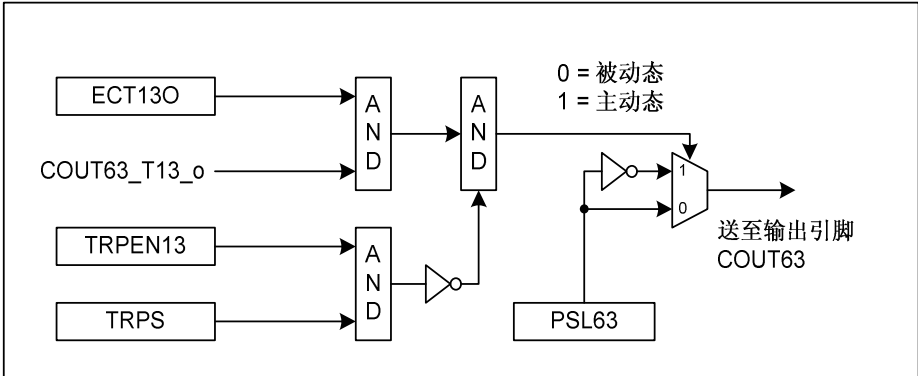


图 12-11 T13 相关输出 COUT63 的调制控制

图 12-12 给出 CC60 和 COUT60 调制控制的例子。

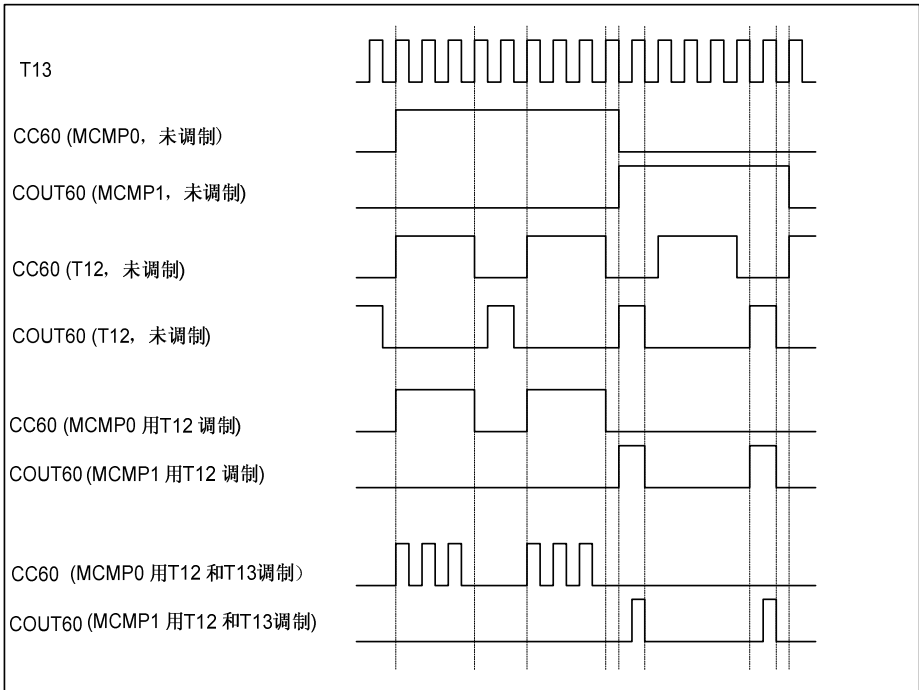


图 12-12 CC60 和 COUT60 的调制控制举例

12.1.4 强制中断处理

强制中断功能使得 PWM 输出能够响应输入引脚 $\overline{\text{CTRAP}}$ 的状态。强制中断输入有效时，该功能可用来关闭功率器件（如，当发生急停时）。

在强制中断状态下，选中的输出被强制为被动态，无有效调制。若 $\overline{\text{CTRAP}}$ 输入信号有效，硬件控制立刻进入强制中断状态，强制中断功能由位 TRPPEN 使能。也可由软件置位 TRPF（强制中断输入标志）进入该状态，从而使 TRPS = 1（强制中断状态指示标志）。当输入无效，软件控制退出该强制中断状态，退出动作和下列事件同步：

- （若 TRPM2 = 0） $\overline{\text{CTRAP}}$ 无效后 TRPF 由硬件自动复位
- （若 TRPM2 = 1） $\overline{\text{CTRAP}}$ 无效后 TRPF 必须由软件复位
- TRPF 复位后同步于 T12 的 PWM
（T12 边沿对齐模式的周期 - 匹配或中间对齐模式递减计数时 1 - 匹配）
- TRPF 复位后同步于 T13 的 PWM
（T13 周期 - 匹配）
- 不和 T12 或 T13 同步

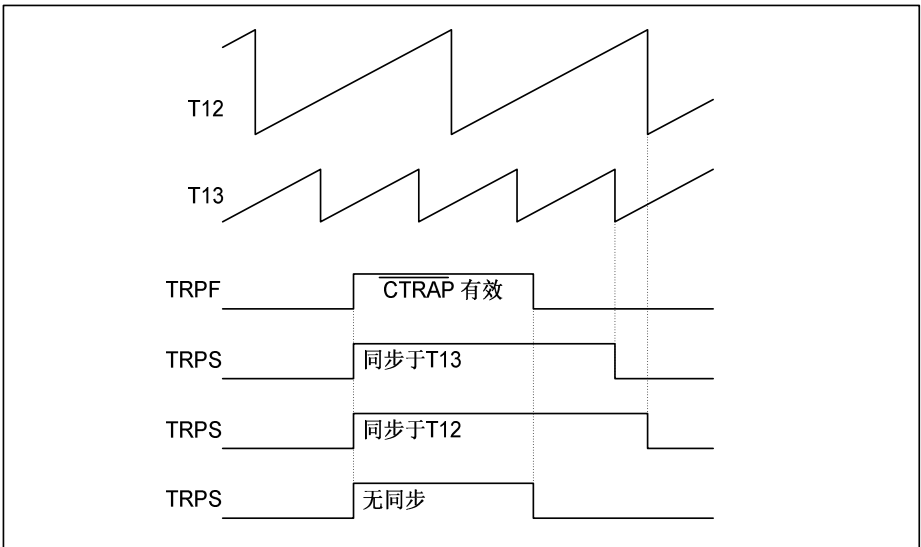


图 12-13 强制中断状态的同步 (TRPM2 = 0)

12.1.5 多通道模式

多通道模式可用来同时调制所有 6 路 T12 的相关输出。用位域 MCMP 选择可能有有效的输出。如果多通道模式被使能（位 MCMEN = 1），只有位域 MCMP 中被置 1 的位所对应的输出才有效。

位域 MCMP 有自己的映射位域 MCMPS，可由软件写入。可由 T12 或 T13 事件触发将 MCMPS 中的新值传送到 MCMP 中；传送动作和 T12 或 T13 事件同步。该结构允许由软件写入新值，在定义好的时刻硬件使用该值，并使映射传送和 PWM 的周期同步。这样就避免由不同步的调制源（T12, T13, SW）引起的多余脉冲。

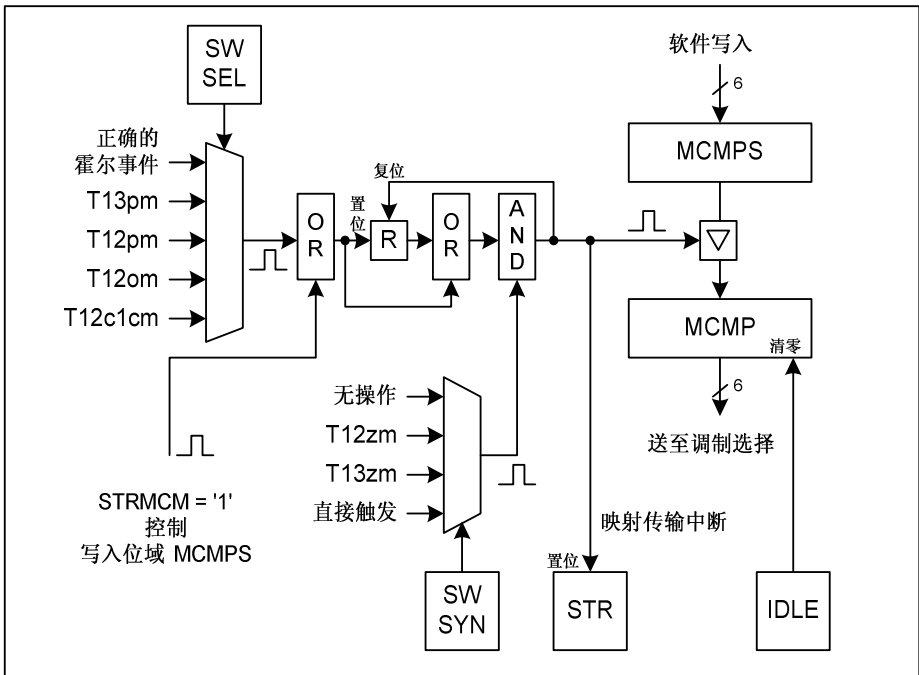


图 12-14 调制选择和同步

图 12-14 示出多通道模式的调制选择。由 SWSEL 选择触发位域 MCMP 更新的事件。如果所选事件发生，提示标志 R 被置位。该标志监控更新请求，更新时 R 被自动复位。为了使 MCMP 的更新和 T12 或 T13 产生的 PWM 同步，位域 SWSYN 可选择引起 MCMPS 内容传送到 MCMP 的同步事件。根据这种结构，在新的 PWM 周期更新 MCMP。

捕获/比较单元 6

也可由软件请求更新，映射传输请求位 **STRMCM** 置位时写入位域 **MCMP5**。如果执行写操作时该位被置位，标志 **R** 被自动置位。从而 **MCMP** 的更新完全由软件控制。

发生映射传输时可产生映射传输中断。

可能的硬件请求事件有：

- 递增计数时 **T12** 周期 - 匹配 (**T12pm**)
- 递减计数时 **T12 1** - 匹配 (**T12om**)
- **T13** 周期 - 匹配 (**T13pm**)
- 通道 1 的 **T12** 比较 - 匹配 (**T12c1cm**)
- 正确的霍尔事件

可能的硬件同步事件有：

- 递增计数时 **T12 0** - 匹配 (**T12zm**)
- **T13 0** - 匹配 (**T13zm**)

12.1.6 霍尔传感器模式

无刷直流电机中，下一个多通道状态值由霍尔输入序列决定。霍尔序列（CURH）和调制序列（MCMP）之间关系紧密。由于电机类型不同，驱动电机的调制序列有所不同。因此，灵活定义霍尔序列和相应调制序列之间的关系是有好处的。CCU6 具备这样的特性，通过存有实际霍尔序列（CURHS）、下次期望的霍尔序列（EXPHS）以及输出序列（MCMPS）的寄存器来实现。每个正确的霍尔事件发生时，新的霍尔序列和相应的输出序列（取自预定义的查找表）可由软件载入寄存器 MCMOUTS 中。也可软件写入映射寄存器 MCMOUTS，同时位 STRHP = 1。有相延迟时（由 T12 通道 1 产生），多通道模式映射传输（由位 STR 指示）时加载新序列。

12.1.6.1 霍尔序列采样

以模块时钟 f_{CCU6} 采样（CCPOSx 引脚上的）霍尔序列。利用死区时间计数器 DTC0（模式 MSEL6x = 1000_B）实现的硬件噪声滤波器可抑制霍尔输入的毛刺。霍尔事件到来时，DTC0 被重载并开始计数，在被检测到的事件和采样点之间产生一延迟。计数器计数至 1 时，采样 CCPOSx 输入信号（无噪声和毛刺），将采样值和当前的霍尔序列（CURH）以及期望霍尔序列（EXPH）进行比较。若采样序列和当前序列相同，这意味着 CCPOSx 的跳变沿由噪声毛刺造成，故不触发任何动作（延迟导致的隐含噪声滤波）。若采样序列和期望序列相同，CCPOSx 的跳变沿为正确的霍尔事件，位 CHE 置位并引起中断。

如果要求多通道模式和霍尔序列比较独立工作于定时器 T12，DTC0 的延迟产生可被旁路。这种情况下，定时器 T12 可用作其他用途。

位 HSYNC 定义了触发事件，触发采样霍尔输入序列，并和当前以及期望的霍尔序列相比较。也可在位 SWHC 中写入 1 软件触发霍尔比较。采样的硬件触发包括：

- 输入 CCPOSx (x = 0 - 2) 的任意升降沿
- T13 比较 - 匹配
- T13 周期 - 匹配
- T12 周期 - 匹配（递增计数）
- T12 1 - 匹配（递减计数）
- T12 通道 0 比较 - 匹配（递增计数）
- T12 通道 0 比较 - 匹配（递减计数）

正确的霍尔事件可作为寄存器 MCMOUTS 的传输请求事件。映射传输将新的 CURH 序列和 EXPH 序列从 MCMOUTS 复制到 MCMOUT 中。如果霍尔输入采样既非当前序列也非期望序列，位 WHE（错误霍尔事件）被置位，从而引起中断并设置空闲模式对 MCMP 清零（调制输出无效）。从空闲模式重新启动工作，必须由软件（位 STRHP 和位域 SWSEL/SWSYN）激活 MCMOUTS 的传输请求。

12.1.6.2 无刷直流控制

无刷直流电机中有一种特殊模式 ($MSEL6x = 1000_B$)，霍尔输入 (CCPOSx) 改变时触发该模式。在这种情况下，T12 的通道 0 用作捕获通道；通道 1 和 2 为比较通道（无输出调制），多通道模块和可能的 T13 调制一起触发输出切换。

检测到有效的霍尔事件跳变沿之后，T12 的计数值被捕获到通道 0 中（代表电机的实际转速）并将 T12 复位。通道 1 的定时器计数至比较值时，触发位域 MCMP 的映射传输切换到下个多通道状态。该触发事件可和某些条件相结合，用来实现噪声滤波（正确的霍尔事件）并将下个多通道状态与调制源同步（以避免输出毛刺）。如果不用霍尔传感器，而用无传感器反电动势技术，要利用通道 1 的比较功能产生位置输入到输出切换之间的相延迟。通道 2 的比较值可用作超时触发（中断），指示电机的目标转速远远低于期望转速（由异常的负载变化引起）。该模式下，必须禁止 T12 调制 ($T12MODENx = 0$)。

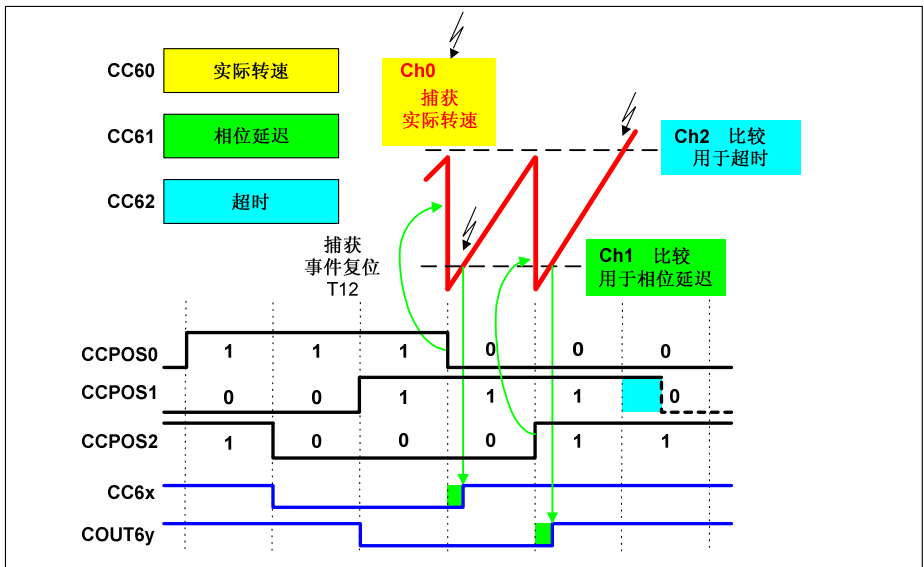


图 12-15 定时器 T12 无刷直流模式（所有 $MSEL6x = 1000_B$ ）

表 12-1 列出一个无刷直流电机控制块切换的例子。如果输入信号组合 CCPOS0-CCPOS2 改变状态，输出 CC6x 和 COUT6x 相应被设置为新状态。

图 12-16 示出左转模式的块切换；图 12-17 示出右转模式的块切换。这些图直接根据表 12-1 得到。

表 12-1 块切换控制表

模式	CCPOS0 – CCPOS2 输入			CC60 – CC62 输出			COUT60 – COUT62 输出		
	CC POS0	CC POS1	CC POS2	CC60	CC61	CC62	COUT60	COUT61	COUT62
左转, 0 相移	1	0	1	无效	无效	有效	无效	有效	无效
	1	0	0	无效	无效	有效	有效	无效	无效
	1	1	0	无效	有效	无效	有效	无效	无效
	0	1	0	无效	有效	无效	无效	无效	有效
	0	1	1	有效	无效	无效	无效	无效	有效
	0	0	1	有效	无效	无效	无效	有效	无效
右转	1	1	0	有效	无效	无效	无效	有效	无效
	1	0	0	有效	无效	无效	无效	无效	有效
	1	0	1	无效	有效	无效	无效	无效	有效
	0	0	1	无效	有效	无效	有效	无效	无效
	0	1	1	无效	无效	有效	有效	无效	无效
	0	1	0	无效	无效	有效	无效	有效	无效
减速	X	X	X	无效	无效	无效	有效	有效	有效
空闲 ¹⁾	X	X	X	无效	无效	无效	无效	无效	无效

¹⁾ 假如霍尔输入的采样值既不是当前的霍尔图样也不是期望的霍尔图样，位 WHE（错误霍尔事件）被置位，它可引起中断并置位空闲模式对 MCMP 清零（调制输出无效）。

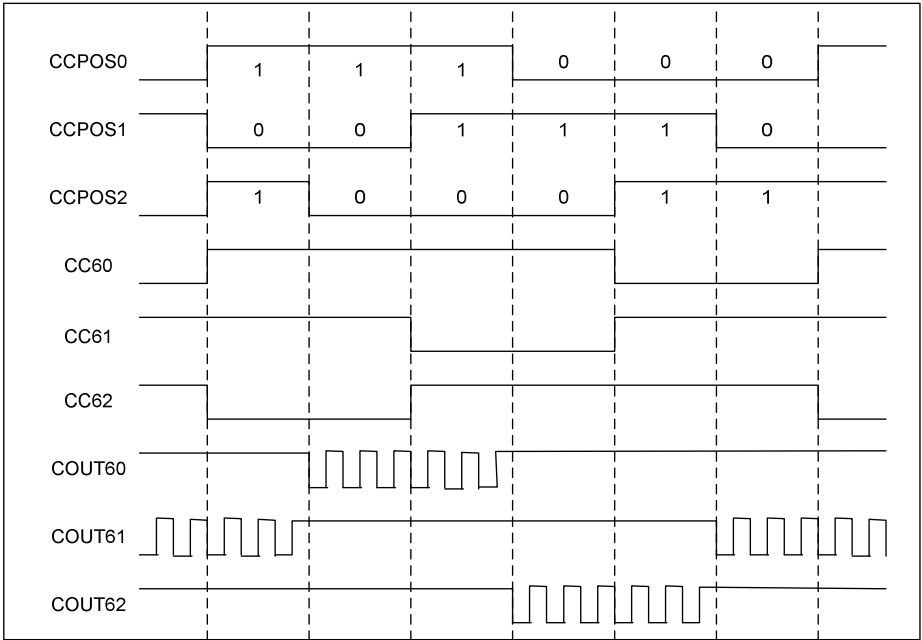


图 12-16 左转模式下的块切换

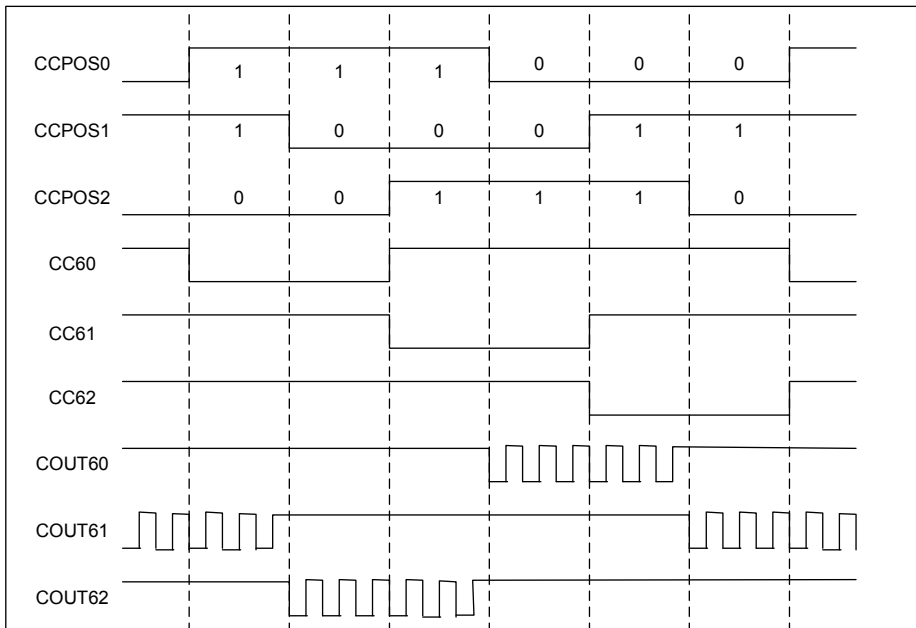


图 12-17 右转模式下的块切换

12.1.7 中断产生

可由中断事件触发中断，或由软件对寄存器 IS 的相应中断位置位产生中断。中断的产生和寄存器 IS 中的中断标志无关。寄存器 IS 只可读；对该寄存器的写操作不会改变其内容。对寄存器 ISS 或寄存器 ISR 的写操作可分别置位或复位寄存器 IS 中的各个位。

如果寄存器 IEN 中的相关中断使能位被使能，将产生中断。设置中断节点指针寄存器 INP，CCU6 模块的中断源可被映射到 4 条中断输出线上。

12.1.8 端口连接

表 12-2 给出如何设置位和位域，已满足 CCU6 I/O 线上的 I/O 功能要求。该表也给出外设输入选择寄存器的值。

表 12-2 CCU6 I/O 控制选择

端口线	PISEL 寄存器位	I/O 控制寄存器位	I/O
P3.6/ $\overline{CTRAP_0}$	ISTRP = 00 _B	P3_DIR.P6 = 0 _B	输入
P2.2/ $\overline{CTRAP_1}$	ISTRP = 01 _B	P2_DIR.P2 = 0 _B	输入
P0.2/ $\overline{CTRAP_2}$	ISTRP = 10 _B	P0_DIR.P2 = 0 _B	输入
P2.0/CCPOS0_0	ISPOS0 = 00 _B	P2_DIR.P0 = 0 _B	输入
P1.5/CCPOS0_1	ISPOS0 = 01 _B	P1_DIR.P5 = 0 _B	输入
P2.1/CCPOS1_0	ISPOS1 = 00 _B	P2_DIR.P1 = 0 _B	输入
P1.6/CCPOS1_1	ISPOS1 = 01 _B	P1_DIR.P6 = 0 _B	输入
P2.2/CCPOS2_0	ISPOS2 = 00 _B	P2_DIR.P2 = 0 _B	输入
P1.7/CCPOS2_1	ISPOS2 = 01 _B	P1_DIR.P7 = 0 _B	输入
P3.0/CC60	-	P3_DIR.P0 = 0 _B	输入
	-	P3_DIR.P0 = 1 _B	
	-	P3_ALTSEL0.P0 = 1 _B	
	-	P3_ALTSEL1.P0 = 0 _B	
P3.1/COU60	-	P3_DIR.P1 = 1 _B	输出

捕获/比较单元 6

端口线	PISEL 寄存器位	I/O 控制寄存器位	I/O
		P3_ALTSEL0.P1 = 1 _B	
		P3_ALTSEL1.P1 = 0 _B	
P3.2/CC61_0	ISCC61 = 00	P3_DIR.P2 = 0 _B	输入
	-	P3_DIR.P2 = 1 _B	输出
		P3_ALTSEL0.P2 = 1 _B	
		P3_ALTSEL1.P2 = 0 _B	
P0.0/CC61_1	ISCC61 = 01	P0_DIR.P0 = 0 _B	输入
	-	P0_DIR.P0 = 1 _B	输出
		P0_ALTSEL0.P0 = 0 _B	
		P0_ALTSEL1.P0 = 1 _B	
P3.3/COUT61_0	-	P3_DIR.P3 = 1 _B	输出
		P3_ALTSEL0.P3 = 1 _B	
		P3_ALTSEL1.P3 = 0 _B	
P0.0/COUT61_1	-	P0_DIR.P0 = 1 _B	输出
		P0_ALTSEL0.P0 = 0 _B	
		P0_ALTSEL1.P0 = 1 _B	
P3.4/CC62_0	ISCC62 = 00	P3_DIR.P4 = 0 _B	输入
	-	P3_DIR.P4 = 1 _B	输出
		P3_ALTSEL0.P4 = 1 _B	
		P3_ALTSEL1.P4 = 0 _B	
P0.4/CC62_1	ISCC62 = 01	P0_DIR.P4 = 0 _B	输入
	-	P0_DIR.P4 = 1 _B	输出
		P0_ALTSEL0.P4 = 0 _B	
		P0_ALTSEL1.P4 = 1 _B	
P3.5/COUT62_0	-	P3_DIR.P5 = 1 _B	输出
		P3_ALTSEL0.P5 = 1 _B	

捕获/比较单元 6

端口线	PISEL 寄存器位	I/O 控制寄存器位	I/O
		P3_ALTSEL1.P5 = 0 _B	
P0.5/COUT62_1	-	P0_DIR.P5 = 1 _B	输出
		P0_ALTSEL0.P5 = 0 _B	
		P0_ALTSEL1.P5 = 1 _B	
P3.7/COUT63_0	-	P3_DIR.P7 = 1 _B	输出
		P3_ALTSEL0.P7 = 1 _B	
		P3_ALTSEL1.P7 = 0 _B	
P0.3/COUT63_1	-	P0_DIR.P3 = 1 _B	输出
		P0_ALTSEL0.P3 = 0 _B	
		P0_ALTSEL1.P3 = 1 _B	
P1.6/T12HR_0	IST12HR = 00	P1_DIR.P6 = 0 _B	输入
P0.0/T12HR_1	IST12HR = 01	P0_DIR.P0 = 0 _B	输入
P2.0/T12HR_2	IST12HR = 10	P2_DIR.P0 = 0 _B	输入
P1.7/T13HR_0	IST13HR = 00	P1_DIR.P7 = 0 _B	输入
P0.1/T13HR_1	IST13HR = 01	P0_DIR.P1 = 0 _B	输入
P2.1/T13HR_2	IST13HR = 10	P2_DIR.P1 = 0 _B	输入

12.2 寄存器映射

CCU6 的 SFR 在标准存储器区 (RMAP=0)，由四页构成。CCU6_PAGE 寄存器位于地址 A3_H处，包含分页值和页控制信息。

CCU6_PAGE

CCU6 分页寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
OP		STNR		0	PAGE		
w		w		r	rw		

符号	位序号	读写类型	功能描述
PAGE	[2:0]	rw	分页位 写入时，该值表示新页的值 读出时，该值表示当前有效页的值
STNR	[5:4]	w	保存编号 该编号指明在哪个保存位域上执行 OP 定义的操作。 若 OP=10 _B ， PAGE 的内容在被新值覆盖之前保存在 STx 中 若 OP=11 _B ， PAGE 的内容被 STx 覆盖。写入 PAGE 的值不予理睬 00 选择 ST0 01 选择 ST1 10 选择 ST2 11 选择 ST3
OP	[7:6]	w	操作 0X 手动保存页模式，STNR 的值被忽略， PAGE 被直接写入 10 带有自动页保存的新页设置。当前写入 PAGE 中的内容被保存的同时，上次

符号	位序号	读写类型	功能描述
			写入 PAGE 的内容被保存在 STNR 规定的位域 STx 中 11 自动恢复页。对写入 PAGE 的内容不予理睬，PAGE 由 STNR 规定的位域 STx 中的内容覆盖
0	3	r	保留 读操作返回 0；应写入 0

下表中描述的所有 CCU6 寄存器名称，在本手册其他章节中引用时需加上模块名前缀“CCU6_”，例如 CCU6_CC63SRL。

CCU6 的 SFR 地址（非映射地址）列于表 12-3 中。

表 12-3 页 0 – 3 的 SFR 地址列表

地址	页 0	页 1	页 2	页 3
9A _H	CC63SRL	CC63RL	T12MSELL	MCMOUTL
9B _H	CC63SRH	CC63RH	T12MSELH	MCMOUTH
9C _H	TCTR4L	T12PRL	IENL	ISL
9D _H	TCTR4H	T12PRH	IENH	ISH
9E _H	MCMOUTSL	T13PRL	INPL	PISEL0L
9F _H	MCMOUTSH	T13PRH	INPH	PISEL0H
A4 _H	ISRL	T12DTCL	ISSL	PISEL2
A5 _H	ISRH	T12DTCH	ISSH	-
A6 _H	CMPMODIFL	TCTR0L	PSLR	-
A7 _H	CMPMODIFH	TCTR0H	MCMCTR	-
FA _H	CC60SRL	CC60RL	TCTR2L	T12L
FB _H	CC60SRH	CC60RH	TCTR2H	T12H
FC _H	CC61SRL	CC61RL	MODCTRL	T13L
FD _H	CC61SRH	CC61RH	MODCTRH	T13H

捕获/比较单元 6

地址	页 0	页 1	页 2	页 3
FE _H	CC62SRL	CC62RL	TRPCTRL	CMPSTATL
FF _H	CC62SRH	CC62RH	TRPCTRH	CMPSTATH

12.3 寄存器描述

表 12-4 示出 CCU6 模块的所有相关寄存器。

表 12-4 CCU6 模块寄存器

寄存器缩略名	寄存器完整名	具体描述请参见
系统寄存器		
PISEL0L	端口输入选择寄存器 0, 低位字节	页 12- 35
PISEL0H	端口输入选择寄存器 0, 高位字节	页 12- 37
PISEL2	端口输入选择寄存器 2	页 12- 38
T12 寄存器		
T12L	定时器 T12 计数寄存器, 低位字节	页 12- 39
T12H	定时器 T12 计数寄存器, 高位字节	页 12- 39
T12PRL	定时器 T12 周期寄存器, 低位字节	页 12- 40
T12PRH	定时器 T12 周期寄存器, 高位字节	页 12- 40
CC6xRL	通道 CC6x 捕获/比较寄存器, 低位字节	页 12- 41
CC6xRH	通道 CC6x 捕获/比较寄存器, 高位字节	页 12- 41
CC6xSRL	通道 CC6x 捕获/比较映射寄存器, 低位字节	页 12- 42
CC6xSRH	通道 CC6x 捕获/比较映射寄存器, 高位字节	页 12- 42
T12DTCL	定时器 T12 死区时间控制寄存器, 低位字节	页 12- 43
T12DTCH	定时器 T12 死区时间控制寄存器, 高位字节	页 12- 43
T13 寄存器		
T13L	定时器 T13 计数寄存器, 低位字节	页 12- 45
T13H	定时器 T13 计数寄存器, 高位字节	页 12- 45

寄存器缩略名	寄存器完整名	具体描述请参见
T13PRL	定时器 T13 周期寄存器, 低位字节	页 12- 46
T13PRH	定时器 T13 周期寄存器, 高位字节	页 12- 46
CC63RL	通道 CC63 捕获/比较寄存器, 低位字节	页 12- 47
CC63RH	通道 CC63 捕获/比较寄存器, 高位字节	页 12- 47
CC63SRL	通道 CC63 捕获/比较映射寄存器, 低位字节	页 12- 48
CC63SRH	通道 CC63 捕获/比较映射寄存器, 高位字节	页 12- 48
CCU6 控制寄存器		
CMPSTATL	比较状态寄存器, 低位字节	页 12- 49
CMPSTATH	比较状态寄存器, 高位字节	页 12- 50
CMPMODIFL	比较状态修改寄存器, 低位字节	页 12- 51
CMPMODIFH	比较状态修改寄存器, 高位字节	页 12- 51
TCTR0L	定时器控制寄存器 0, 低位字节	页 12- 52
TCTR0H	定时器控制寄存器 0, 高位字节	页 12- 54
TCTR2L	定时器控制寄存器 2, 低位字节	页 12- 56
TCTR2H	定时器控制寄存器 2, 高位字节	页 12- 58
TCTR4L	定时器控制寄存器 4, 低位字节	页 12- 59
TCTR4H	定时器控制寄存器 4, 高位字节	页 12- 60
调制控制寄存器		
MODCTRL	调制控制寄存器, 低位字节	页 12- 62
MODCTRH	调制控制寄存器, 高位字节	页 12- 63
TRPCTRL	强制中断控制寄存器, 低位字节	页 12- 65
TRPCTRH	强制中断控制寄存器, 高位字节	页 12- 66
PSLR	被动态电平寄存器	页 12- 68

寄存器缩略名	寄存器完整名	具体描述请参见
MCMOUTSL	多通道模式输出映射寄存器，低位字节	页 12- 70
MCMOUTSH	多通道模式输出映射寄存器，高位字节	页 12- 71
MCMOUTL	多通道模式输出寄存器，低位字节	页 12- 72
MCMOUTH	多通道模式输出寄存器，高位字节	页 12- 74
MCMCTR	多通道模式控制寄存器	页 12- 76
T12MSELL	定时器 T12 捕获/比较模式选择寄存器，低位字节	页 12- 78
T12MSELH	定时器 T12 捕获/比较模式选择寄存器，高位字节	页 12- 80

中断控制寄存器

ISL	中断状态寄存器，低位字节	页 12- 84
ISH	中断状态寄存器，高位字节	页 12- 85
ISSL	中断状态置位寄存器，低位字节	页 12- 87
ISSH	中断状态置位寄存器，高位字节	页 12- 89
ISRL	中断状态复位寄存器，低位字节	页 12- 91
ISRH	中断状态复位寄存器，高位字节	页 12- 93
IENL	中断使能寄存器，低位字节	页 12- 95
IENH	中断使能寄存器，高位字节	页 12- 97
INPL	中断节点指针寄存器，低位字节	页 12- 99
INPH	中断节点指针寄存器，高位字节	页 12- 101

注：对于所有的 **CCU6** 寄存器：只写位（由“w”表示）在读取时，值始终为 0。当硬件和软件同时请求修改某位时，软件占优。

12.3.1 系统寄存器

12.3.1.1 端口输入选择

寄存器 PISEL0 和 PISEL42 中包含选择模块真正的输入信号的控制位域，从而器件的引脚可根据应用需求进行配置。输出引脚由端口寄存器控制选择。

PISEL0L

端口输入选择寄存器 0，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
ISTRP		ISCC62		ISCC61		ISCC60	
rw		rw		rw		rw	

符号	位序号	读写类型	功能描述
ISCC60	[1:0]	rw	<p>CC60 的输入选择</p> <p>该位域定义了用作 CC60 捕获输入信号的端口引脚。</p> <p>00 选择 CC60_0 输入引脚</p> <p>01 保留</p> <p>10 保留</p> <p>11 保留</p>
ISCC61	[3:2]	rw	<p>CC61 的输入选择</p> <p>该位域定义了用作 CC61 捕获输入信号的端口引脚。</p> <p>00 选择 CC61_0 输入引脚</p> <p>01 选择 CC61_1 输入引脚</p> <p>10 保留</p> <p>11 保留</p>
ISCC62	[5:4]	rw	<p>CC62 的输入选择</p> <p>该位域定义了用作 CC62 捕获输入信号的端口引脚。</p> <p>00 选择 CC62_0 输入引脚</p> <p>01 选择 CC62_1 输入引脚</p>

符号	位序号	读写类型	功能描述
			10 保留 11 保留
ISTRP	[7:6]	rw	<p>$\overline{\text{CTRAP}}$ 的输入选择</p> <p>该位域定义了用作 $\overline{\text{CTRAP}}$ 输入信号的端口引脚。</p> <p>00 选择 $\overline{\text{CTRAP}_0}$ 输入引脚</p> <p>01 选择 $\overline{\text{CTRAP}_1}$ 输入引脚</p> <p>10 选择 $\overline{\text{CTRAP}_2}$ 输入引脚</p> <p>11 保留</p>

PISEL0H
端口输入选择寄存器 0，高位字节
复位值: 00H

7	6	5	4	3	2	1	0
IST12HR		ISOPS2		ISPOS1		ISPOS0	
rw		rw		rw		rw	

符号	位序号	读写类型	功能描述
ISPOS0	[1:0]	rw	CCPOS0 的输入选择 该位域定义了用作 CCPOS0 捕获输入信号的端口引脚。 00 选择 CCPOS0_0 输入引脚 01 选择 CCPOS0_1 输入引脚 10 保留 11 保留
ISPOS1	[3:2]	rw	CCPOS1 的输入选择 该位域定义了用作 CCPOS1 捕获输入信号的端口引脚。 00 选择 CCPOS1_0 输入引脚 01 选择 CCPOS1_1 输入引脚 10 保留 11 保留
ISPOS2	[5:4]	rw	CCPOS2 的输入选择 该位域定义了用作 CCPOS2 捕获输入信号的端口引脚。 00 选择 CCPOS2_0 输入引脚 01 选择 CCPOS2_1 输入引脚 10 保留 11 保留
IST12HR	[7:6]	rw	T12HR 的输入选择 该位域定义了用作 T12HR 输入信号的端口引脚。

符号	位序号	读写类型	功能描述
			00 选择 T12HR_0 的输入引脚
			01 选择 T12HR_1 的输入引脚
			10 选择 T12HR_2 的输入引脚
			11 保留

PISEL2
端口输入选择寄存器 2
复位值: 00_H

7	6	5	4	3	2	1	0
0						IST13HR	
r						rw	

符号	位序号	读写类型	功能描述
IST13HR	[1:0]	rw	T13HR 的输入选择 该位域定义了用作 T13HR 捕获输入信号的端口引脚。 00 选择 T13HR_0 的输入引脚 01 选择 T13HR_1 的输入引脚 10 选择 T13HR_2 的输入引脚 11 保留
0	[7:2]	r	保留 读操作返回 0; 应写入 0

12.3.2 定时器 T12 的相关寄存器

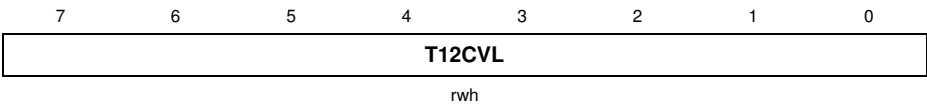
3 路通道的 PWM 序列是基于定时器 T12 产生的。定时器 T12 的相关寄存器（在定义好的条件下）可被同时刷新以保证三路 PWM 通道的一致性。

定时器 T12 支持捕获和比较模式，可独立选择这两种模式用于 3 路通道 CC60，CC61 和 CC62。

T12L

定时器 T12 计数寄存器，低位字节

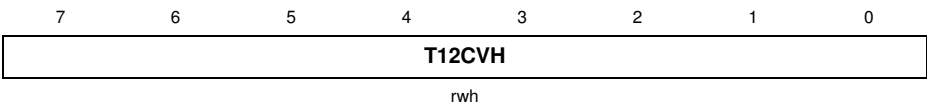
复位值: 00_H



T12H

定时器 T12 计数寄存器，高位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
T12CV	T12L 的 [7:0], T12H 的 [7:0]	rwh	定时器 T12 的计数值 该寄存器存放定时器 T12 的 16 位计数值。

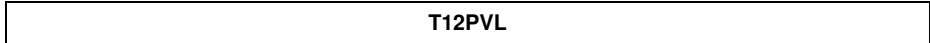
注：一旦定时器 T12 停止工作，内部时钟分频被复位以确保可重复产生同样的时序和延迟。

T12PRL

定时器 T12 周期寄存器，低位字节

复位值: 00H

7 6 5 4 3 2 1 0



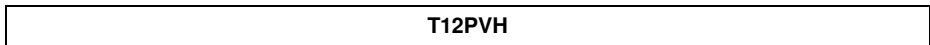
rwh

T12PRH

定时器 T12 周期寄存器，高位字节

复位值: 00H

7 6 5 4 3 2 1 0

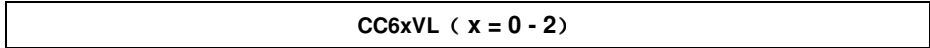


rwh

符号	位序号	读写类型	功能描述
T12PV	T12PRL 的 [7:0], T12PRH 的 [7:0]	rwh	定时器 T12 的周期值 T12PV 定义了用来产生周期 - 匹配的 T12 的计数值。计数到该值后，定时器 T12 被置 0（边沿对齐模式）或改变其计数方向为递减计数（中间对齐模式）。

CC6xRL (x = 0 - 2)
通道 CC6x 捕获/比较寄存器, 低位字节
复位值: 00_H

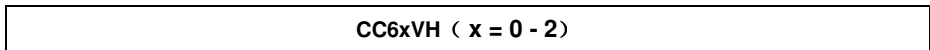
7 6 5 4 3 2 1 0



rh

CC6xRH (x = 0 - 2)
通道 CC6x 捕获/比较寄存器, 高位字节
复位值: 00_H

7 6 5 4 3 2 1 0



rh

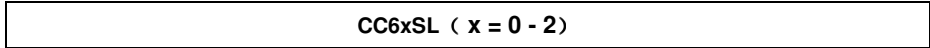
符号	位序号	读写类型	功能描述
CC6xV (x = 0 - 2)	CC6xRL 的 [7:0], CC6xRH 的 [7:0]	rh	通道 x 捕获/比较值 比较模式下, 位域 CC6xV 存放和 T12 计数值进行比较的值; 捕获模式下, T12 的捕获值可从该寄存器读出。

CC6xSRL (x = 0 - 2)

通道 CC6x 捕获/比较映射寄存器，低位字节

复位值: 00_H

7 6 5 4 3 2 1 0



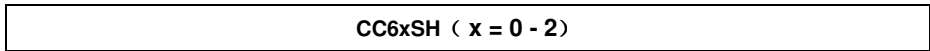
rwh

CC6xSRH (x = 0 - 2)

通道 CC6x 捕获/比较映射寄存器，高位字节

复位值: 00_H

7 6 5 4 3 2 1 0



rwh

符号	位序号	读写类型	功能描述
CC6xS (x = 0 - 2)	CC6xSRL 的 [7:0], CC6xSRH 的 [7:0]	rwh	通道 x 捕获/比较值映射寄存器 比较模式下，映射传输时位域 CC6xS 的内容被传送到位域 CC6xV 中；捕获模式下，T12 的捕获值可从该寄存器读出。

T12DTCL
定时器 T12 死区时间控制寄存器，低位字节
复位值: 00_H

7 6 5 4 3 2 1 0

DTM							
rw							

符号	位序号	读写类型	功能描述
DTM	[7:0]	rw	死区时间 位域 DTM 控制了选择输出从被动态切换至主动态的可编程延迟。主动态切换至被动态无延迟。

T12DTCH
定时器 T12 死区时间控制寄存器，高位字节
复位值: 00_H

7 6 5 4 3 2 1 0

0	DTR2	DTR1	DTR0	0	DTE2	DTE1	DTE0
r	rh	rh	rh	r	rw	rw	rw

符号	位序号	读写类型	功能描述
DTE0	0	rw	死区时间使能位 位 DTE _x (x = 0 - 2) 使能和禁止定时器 T12 每路比较通道 (0, 1, 2) 的死区时间产生。 0 禁止产生死区时间。(根据实际比较状态) 相应输出从被动态切换至主动态时无延迟。 1 使能产生死区时间。(根据实际比较状态) 相应输出从被动态切换至主动态时产生延迟, 延迟值由位域 DTM 设定。
DTE1	1		
DTE2	2		

符号	位序号	读写类型	功能描述
DTR0 DTR1 DTR2	4 5 6	rh	死区时间功能运行指示位 位 DTRx (x = 0 - 2) 指示定时器 T12 每路比较通道 (0, 1, 2) 产生死区时间的状态。 0 相应通道死区时间计数器的值为 0 1 相应通道死区时间计数器的值非 0
0	3, 7	r	保留 读操作返回 0; 应写入 0

注：死区时间计数器的时钟和 T12 时钟频率相同。该结构可在中间对齐和边沿对齐 PWM 模式下产生对称的死区时间。若 CC6x 的占空比为 50%，COU6x 的开启时间为： $0.5 * \text{周期值} - \text{死区时间}$ 。

注：死区时间计数器由位 DTRES 复位（不由位 T12RES 复位）。

12.3.3 定时器 T13 相关寄存器

单路通道 PWM 序列是基于定时器 T13 产生的。定时器 T13 的相关寄存器（在定义好的条件下）可被同时刷新以保证 PWM 信号的一致性。定时器 T13 可和定时器 T12 的不同事件进行同步。

定时器 T13 只支持比较通道 CC63 上的比较模式。

T13L

定时器 T13 计数寄存器，低位字节

复位值: 00_H

7	6	5	4	3	2	1	0
T13CVL							
rwh							

T13H

定时器 T13 计数寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
T13CVH							
rwh							

符号	位序号	读写类型	功能描述
T13CV	T13L 的 [7:0], T13H 的 [7:0]	rwh	定时器 T13 的计数值 该寄存器代表定时器 T13 的 16 位计数值。

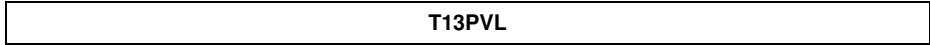
注：一旦定时器 T13 停止工作，内部时钟分频被复位以确保可重复产生同样的时序和延迟。

T13PRL

定时器 T13 周期寄存器，低位字节

复位值: 00H

7 6 5 4 3 2 1 0



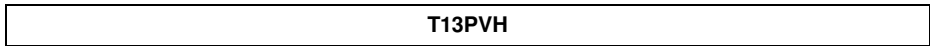
rwh

T13PRH

定时器 T13 周期寄存器，高位字节

复位值: 00H

7 6 5 4 3 2 1 0



rwh

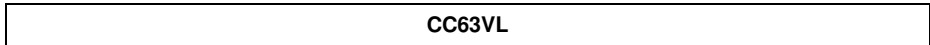
符号	位序号	读写类型	功能描述
T13PV	T13PRL 的 [7:0], T13PRH 的 [7:0]	rwh	定时器 T13 的周期值 T12PV 定义了用来产生周期 - 匹配的 T13 的计数值。计数到该值后，定时器 T13 被置 0。

CC63RL

通道 **CC63** 捕获/比较寄存器，低位字节

复位值: **00_H**

7 6 5 4 3 2 1 0



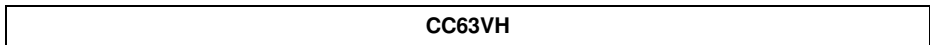
rh

CC63RH

通道 **CC63** 捕获/比较寄存器，高位字节

复位值: **00_H**

7 6 5 4 3 2 1 0



rh

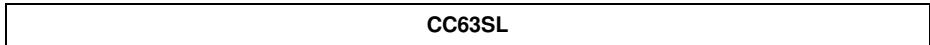
符号	位序号	读写类型	功能描述
CC63V	CC63RL 的 [7:0], CC63RH 的 [7:0]	rh	通道 CC63 比较值 位域 CC63V 存放和 T13 计数值进行比较的值。

CC63SRL

通道 **CC63** 捕获/比较映射寄存器，低位字节

复位值: **00H**

7 6 5 4 3 2 1 0



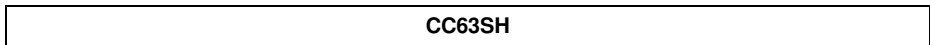
rw

CC63SRH

通道 **CC63** 捕获/比较映射寄存器，高位字节

复位值: **00H**

7 6 5 4 3 2 1 0



rw

符号	位序号	读写类型	功能描述
CC63S	CC63SRL 的 [7:0], CC63SRH 的 [7:0]	rw	通道 CC63 比较值映射寄存器 映射传输时位域 CC63S 的内容被传送到位域 CC63V 中。

12.3.4 捕获/比较控制寄存器

寄存器 CMPSTAT 中存放着监控当前捕获和比较状态的状态位；以及决定比较通道主动/被动态的控制位。

CMPSTATL

比较状态寄存器，低位字节

复位值: 00_H

7	6	5	4	3	2	1	0
0	CC63ST	CCPOS 2	CCPOS 1	CCPOS 0	CC62ST	CC61ST	CC60ST
r	rh	rh	rh	rh	rh	rh	rh

符号	位序号	读写类型	功能描述
CC60ST CC61ST CC62ST CC63ST 1)	0 1 2 6	rh	捕获/比较状态位 位 CC6xST 监控捕获/比较通道的状态。位 CC6xST (x = 0 - 2) 和 T12 相关；CC63ST 和 T13 相关。 0 比较模式下，定时器计数值小于比较值；捕获模式下，软件复位该位后还未检测到所选择的沿跳变。 1 比较模式下，定时器计数值大于或等于比较值；捕获模式下，检测到所选择的沿跳变。
CCPOS0 CCPOS1 CCPOS2	3 4 5	rh	霍尔序列采样值 位 CCPOSx (x = 0 - 2) 指示和当前以及期望的霍尔序列进行比较的输入霍尔序列的采样值。事件 hcrdy (霍尔比较准备就绪) 发生时采样。 0 输入 CCPOSx 的采样值为 0 1 输入 CCPOSx 的采样值为 1
0	7	r	保留 读操作返回 0；应写入 0

¹⁾ 这些位根据 T12 和 T13 的切换规则置位和复位。

CMPSTATH

比较状态寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
T13IM	COU63 PS	COU62 PS	CC62PS	COU61 PS	CC61PS	COU60 PS	CC60PS
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

符号	位序号	读写类型	功能描述
CC60PS	0	rwh	<p>比较输出的被动态选择</p> <p>位 CC6xPS 和 COU6xPS (x = 0 -2) 选择相应比较通道的被动态。被动态时输出引脚驱动 (由寄存器 PSLR 定义的) 被动电平。位 CC6xPS 和 COU6xPS 和 T12 相关, 位 COU63PS 和 T13 相关。</p> <p>0 CC6xST 为 0 时相应的比较输出驱动被动电平。</p> <p>1 CC6xST 为 1 时相应的比较输出驱动被动电平。</p> <p>捕获模式下不使用这些位。</p>
CC61PS	2		
CC62PS	4		
COU60PS	1		
COU61PS	3		
COU62PS	5		
COU63PS	6		
T13IM²⁾	7	rwh	<p>T13 反相调制</p> <p>位 T13IM 将 T13 信号反相用于调制 CC6x 和 COU6x (x = 0 - 2) 信号。</p> <p>0 T13 输出不反相</p> <p>1 T13 输出反相用于调制</p>

¹⁾ 这些位具有映射位, 与 T12 和 T13 捕获/比较寄存器同时更新。读操作读取实际使用位的值, 写操作写入映射位。

²⁾ 该位具有映射位, 与 T13 的比较和周期寄存器同时更新。读操作读取实际使用位的值, 写操作写入映射位。

捕获/比较单元 6

寄存器 CMPMODIF 中存放的控制位可允许软件修改捕获/比较状态。

CMPMODIFL
比较状态修改寄存器，低位字节
复位值: 00H

7	6	5	4	3	2	1	0
0	MCC63S	0			MCC62S	MCC61S	MCC60S
r	w	r			w	w	w

CMPMODIFH
比较状态修改寄存器，高位字节
复位值: 00H

7	6	5	4	3	2	1	0
0	MCC63R	0			MCC62R	MCC61R	MCC60R
r	w	r			w	w	w

符号	位序号	读写类型	功能描述
MCC60S¹⁾	0	w	捕获/比较状态修改位 这些位用作软件置位 (MCC6xS) 或复位 (MCC6xR) 相应的 CC6xST。 例如在相应比较定时器停止工作的情况下，该特性允许用户用软件修改输出线状态。通过一个数据写指令即可实现 CC6xST 的位操作。 MCC6xR, MCC6xS = 0, 0 未改变 CC6xST 0, 1 置位 CC6xST 1, 0 复位 CC6xST 1, 1 保留 (翻转)
MCC61S¹⁾	1		
MCC62S¹⁾	2		
MCC63S¹⁾	6		
MCC60R²⁾	0		
MCC61R²⁾	1		
MCC62R²⁾	2		
MCC63R²⁾	6		
0	[5:3], 7	r	保留 读操作返回 0; 应写入 0

¹⁾ 这些位域存放在比较状态修改寄存器的低位字节中。

²⁾ 这些位域存放在比较状态修改寄存器的高位字节中。

寄存器 TCTR0 控制定时器 T12 和 T13 的基本功能。

TCTR0L

定时器控制寄存器 0, 低位字节

复位值: 00H

7	6	5	4	3	2	1	0
CTM	CDIR	STE12	T12R	T12PRE	T12CLK		
rw	rh	rh	rh	rw	rw		

符号	位序号	读写类型	功能描述
T12CLK	[2:0]	rw	<p>定时器 T12 输入时钟选择</p> <p>根据等式 $f_{T12} = f_{CCU6} / 2^{<T12CLK>}$, 选择取自外设时钟的定时器 T12 的输入时钟。</p> <p>000 $f_{T12} = f_{CCU6}$ 001 $f_{T12} = f_{CCU6} / 2$ 010 $f_{T12} = f_{CCU6} / 4$ 011 $f_{T12} = f_{CCU6} / 8$ 100 $f_{T12} = f_{CCU6} / 16$ 101 $f_{T12} = f_{CCU6} / 32$ 110 $f_{T12} = f_{CCU6} / 64$ 111 $f_{T12} = f_{CCU6} / 128$</p>
T12PRE	3	rw	<p>定时器 T12 预分频</p> <p>为了支持更高的时钟频率, 允许额外的分频因子 1/256 用于 T12 分频。</p> <p>0 禁止 T12 的额外分频 1 使能 T12 的额外分频</p>
T12R¹⁾	4	rh	<p>定时器 T12 运行位</p> <p>T12R 启动和停止定时器 T12 工作。软件通过置位 T12RS 或 T12RR 来置位/复位 T12R; 或根据 T12SSC 定义的功能硬件复位 T12R。</p> <p>0 定时器 T12 被停止工作 1 定时器 T12 在运行</p>

符号	位序号	读写类型	功能描述
STE12	5	rh	<p>定时器 T12 映射传输使能</p> <p>检测到 T12 映射传输事件时，位 STE12 使能或禁止将 T12 的周期值、比较值、被动态选择位和电平从映射寄存器中映射传输到实际寄存器中。映射传输后硬件对 STE12 清零。</p> <p>递增计数时 T12 的映射传输事件为周期 - 匹配；递减计数时为 1 - 匹配</p> <p>0 禁止映射寄存器传输 1 使能映射寄存器传输</p>
CDIR	6	rh	<p>定时器 T12 的计数方向</p> <p>根据 T12 的计数规则对该位置位或复位。</p> <p>0 T12 递增计数 1 T12 递减计数</p>
CTM	7	rw	<p>T12 工作模式</p> <p>0 边沿对齐模式： T12 始终递增计数，计数至周期之后从 0 开始继续计数</p> <p>1 中间对齐模式： 检测到周期 - 匹配时 T12 递减计数，检测到 1 - 匹配时 T12 递增计数</p>

¹⁾ (由 T12SSC, T12RR 或 T12RS) 同时置位/复位 T12R 将不起作用, T12R 保持原值。

TCTR0H
定时器控制寄存器 0, 高位字节
复位值: 00H

7	6	5	4	3	2	1	0
0	STE13		T13R	T13PRE	T13CLK		
r	rh		rh	rw	rw		

符号	位序号	读写类型	功能描述
T13CLK	[2:0]	rw	定时器 T13 输入时钟选择 根据等式 $f_{T13} = f_{CCU6} / 2^{<T13CLK>}$, 选择取自外设时钟的定时器 T13 的输入时钟。 000 $f_{T13} = f_{CCU6}$ 001 $f_{T13} = f_{CCU6} / 2$ 010 $f_{T13} = f_{CCU6} / 4$ 011 $f_{T13} = f_{CCU6} / 8$ 100 $f_{T13} = f_{CCU6} / 16$ 101 $f_{T13} = f_{CCU6} / 32$ 110 $f_{T13} = f_{CCU6} / 64$ 111 $f_{T13} = f_{CCU6} / 128$
T13PRE	3	rw	定时器 T13 预分频 为了支持更高的时钟频率, 允许额外的分频因子 1/256 用于 T13 分频。 0 禁止 T13 的额外分频 1 使能 T13 的额外分频
T13R¹⁾	4	rh	定时器 T13 运行位 T13R 启动和停止定时器 T13 工作。软件通过置位 T13RS 或 T13RR 来置位/复位 T13R; 或根据 T13SSC、T13TEC 和 T13TED 定义的功能硬件置位/复位 T13R。 0 定时器 T13 被停止工作 1 定时器 T13 在运行

符号	位序号	读写类型	功能描述
STE13	5	rh	<p>定时器 T13 映射传输使能</p> <p>检测到 T13 映射传输事件时，位 STE13 使能或禁止将 T13 的周期值、比较值、被动态选择位和电平从映射寄存器中映射传输到实际寄存器中。映射传输后硬件对 STE13 清零。</p> <p>T13 的映射传输事件为周期 - 匹配</p> <p>0 禁止映射寄存器传输</p> <p>1 使能映射寄存器传输</p>
0	[7:6]	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

¹⁾ (由 T13SSC, T13TEC, T13RR 或 T13RS) 同时置位/复位 T13R 将不起作用, T13R 保持原值。

注: 只有当定时器 T12 不工作时 (T12R = 0), 对位域 T12CLK 或位 T12PRE 的写入才有效。只有当定时器 T13 不工作时 (T13R = 0), 对位域 T13CLK 或位 T13PRE 的写入才有效。

捕获/比较单元 6

寄存器 TCTR2 控制定时器 T12 和 T13 的单次模式和同步功能。两个定时器均可工作在单次模式；该模式下，定时器在一个计数周期之后自动停止计数，此时计数值为 0。单次模式与 T13 和 T12 同步相结合，可在定义好的 T12 的 PWM 事件之后产生延迟可编程的事件。例如，该特性可用来在一个特定的延迟之后（以避免切换噪声引起的问题）、和 PWM 事件同步的触发模数转换。

TCTR2L

定时器控制寄存器 2，低位字节

复位值: 00_H

7	6	5	4	3	2	1	0
0	T13TED		T13TEC			T13SSC	T12SSC
r	rw		rw			rw	rw

符号	位序号	读写类型	功能描述
T12SSC	0	rw	<p>定时器 T12 单次模式控制</p> <p>该位控制 T12 的单次模式。</p> <p>0 禁止单次模式，对 T12R 无硬件操作。</p> <p>1 使能单次模式，下列情况下硬件复位 T12R:</p> <ul style="list-style-type: none"> - 边沿对齐模式下 T12 计数至周期值 - 中间对齐模式下 T12 递减计数时计数至 1 <p>位 T12R 复位的同时复位 CC6xST (x = 0 - 2)。</p>
T13SSC	1	rw	<p>定时器 T13 单次模式控制</p> <p>该位控制 T13 的单次模式。</p> <p>0 对 T13R 无硬件操作。</p> <p>1 使能单次模式，T13 计数至周期值后硬件复位 T13R。</p> <p>位 T13R 复位的同时复位 CC63ST。</p>
T13TEC	[4:2]	rw	T13 触发事件控制

符号	位序号	读写类型	功能描述
			<p>根据下列组合，位域 T13TEC 选择启动 T13 的触发事件（自动置位 T13R 和 T12 的比较信号同步）：</p> <p>000 无动作</p> <p>001 通道 0 的 T12 比较事件触发置位 T13R</p> <p>010 通道 1 的 T12 比较事件触发置位 T13R</p> <p>011 通道 2 的 T12 比较事件触发置位 T13R</p> <p>100 通道 0, 1 或 2 的任意 T12 比较事件触发置位 T13R</p> <p>101 T12 的周期 - 匹配置位 T13R</p> <p>110 T12 的 0 - 匹配（递增计数时）置位 T13R</p> <p>111 输入 CCPOSx 的任意升降沿置位 T13R</p>
T13TED¹⁾	[6:5]	rw	<p>T13 触发事件方向</p> <p>当检测到由 T13TEC 定义的触发事件，位域 T13TED 给出控制自动置位 T13R 的附加信息。</p> <p>00 无动作</p> <p>01 T12 递增计数时</p> <p>10 T12 递减计数时</p> <p>11 和 T12 的计数方向无关</p>
0	7	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

¹⁾ 举例说明：

如果 T12 的任意比较事件将启动定时器 T13(T13TEC = 100_B)，触发事件的方向可设定为：

- 递增计数 >> 只有当 T12 递增计数时，T12 通道 0, 1, 2 的比较 - 匹配触发 T13R
- 递减计数 >> 只有当 T12 递减计数时，T12 通道 0, 1, 2 的比较 - 匹配触发 T13R

捕获/比较单元 6

- 和位 CDIR 无关 >> 每次 T12 通道 0, 1, 2 的比较 - 匹配均触发 T13R

定时器的计数方向取自位 CDIR 的值。因此，如果 T12 工作在边沿模式（只递增计数），只有当位域 T13TED = 01_B或 11_B时 T13 可自动被启动。

TCTR2H

定时器控制寄存器 2，高位字节

复位值: 00_H

	7	6	5	4	3	2	1	0
	0			T13RSEL			T12RSEL	
	r			rw			rw	

符号	位序号	读写类型	功能描述
T12RSEL	[1:0]	rw	<p>定时器 T12 外部运行选择</p> <p>位域 T12RSEL 定义了信号 T12HR 的事件，可硬件置位运行位 T12R。</p> <p>00 禁止由外部事件设制 T12R</p> <p>01 检测到 T12HR 的上升沿置位 T12R</p> <p>10 检测到 T12HR 的下降沿时置 T12R</p> <p>11 检测到 T12HR 的任意沿时置 T12R</p>
T13RSEL	[3:2]	rw	<p>定时器 T13 外部运行选择</p> <p>位域 T13RSEL 定义了信号 T13HR 的事件，可硬件置位运行位 T13R。</p> <p>00 禁止由外部事件设制 T13R</p> <p>01 检测到 T13HR 的上升沿置位 T13R</p> <p>10 检测到 T13HR 的下降沿时置 T13R</p> <p>11 检测到 T13HR 的任意沿时置 T13R</p>
0	[7:4]	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

捕获/比较单元 6

寄存器 TCTR4 可通过独立的置位和复位条件，软件控制运行位 T12R 和 T13R。另外，定时器（运行时）可被复位，位 STE12 和 STE13 可由软件控制。

TCTR4L

定时器控制寄存器 4，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
T12STD	T12STR	0	DTRES	T12RES	T12RS	T12RR	
w	w	r	w	w	w	w	w

符号	位序号	读写类型	功能描述
T12RR	0	w	定时器 T12 运行复位 对该位置位将复位 T12R。 0 T12R 不受影响 1 T12R 被清零，T12 停止工作
T12RS	1	w	定时器 T12 运行置位 对该位置位将置位 T12R。 0 T12R 不受影响 1 T12R 被置位，T12 计数
T12RES	2	w	定时器 T12 复位 0 对 T12 无影响 1 T12 计数寄存器被复位为 0。根据切换规则切换输出信号。T12RES 的置位对 T12R 无影响。
DTRES	3	w	死区时间计数器复位 0 对死区时间计数器无影响 1 三路死区时间计数通道被复位为 0
T12STR	6	w	定时器 T12 映射传输请求 0 无操作 1 STE12 被置位，允许映射传输
T12STD	7	w	定时器 T12 映射传输禁止 0 无操作

捕获/比较单元 6

符号	位序号	读写类型	功能描述
			1 STE12 被复位，不能触发映射传输
0	[5:4]	r	保留 读操作返回 0；应写入 0

TCTR4H

定时器控制寄存器 4，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
T13STD	T13STR	0			T13RES	T13RS	T13RR
w	w	r			w	w	w

符号	位序号	读写类型	功能描述
T13RR	0	w	定时器 T13 运行复位 对该位置位将复位 T13R。 0 T13R 不受影响 1 T13R 被清零，T13 停止工作
T13RS	1	w	定时器 T13 运行置位 对该位置位将置位 T13R。 0 T13R 不受影响 1 T13R 被置位，T13 计数
T13RES	2	w	定时器 T13 复位 0 对 T13 无影响 1 T13 计数寄存器被复位为 0。根据切换规则切换输出信号。T13RES 的置位对 T13R 无影响。
T13STR	6	w	定时器 T13 映射传输请求 0 无操作 1 STE13 被置位，允许映射传输
T13STD	7	w	定时器 T13 映射传输禁止

符号	位序号	读写类型	功能描述
			0 无操作 1 STE13 被复位，不能触发映射传输
0	[5:4]	r	保留 读操作返回 0；应写入 0

注：同时置位和复位同一位将不起作用（例如，同时在 T13RR 和 T13RS 中写入 1 不会改变 T13R），相应位保持原值。

12.3.5 调制控制寄存器

12.3.5.1 全局模块控制

寄存器 MODCTR 中的控制位，用来使能由定时器 T12 和 T13 的 PWM 序列调制相应的输出信号。另外，可使能多通道模式用作输出信号的附加调制源。

MODCTRL

调制控制寄存器，低位字节

复位值: 00H

7	6	5	4	3	2	1	0	
MCMEN	0	T12MODEN						
rw	r	rw						

符号	位序号	读写类型	功能描述
T12MODEN	[5:0]	rw	<p>定时器 T12 调制使能</p> <p>置位该位域将使能由 T12 的 PWM 序列调制相应的比较通道。这些位和下列输出信号对应：</p> <p>位 0 CC60 的调制</p> <p>位 1 COUT60 的调制</p> <p>位 2 CC61 的调制</p> <p>位 3 COUT61 的调制</p> <p>位 4 CC62 的调制</p> <p>位 5 COUT62 的调制</p> <p>调制的使能特性定义如下：</p> <p>0 禁止由 T12 的 PWM 序列调制相应的输出信号</p> <p>1 使能由 T12 的 PWM 序列调制相应的输出信号</p>
MCMEN	7	rw	<p>多通道模式使能</p> <p>0 禁止由位域 MCMP 中的多通道序列调制相应的输出信号</p> <p>1 使能由位域 MCMP 中的多通道序列调制相应的输出信号</p>

捕获/比较单元 6

符号	位序号	读写类型	功能描述
0	6	r	保留 读操作返回 0；应写入 0

MODCTRH

调制控制寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
ECT130	0	T13MODEN					
rw	r	rw					

符号	位序号	读写类型	功能描述
T13MODEN	[5:0]	rw	<p>定时器 T13 调制使能</p> <p>置位该位域将使能由 T13 的 PWM 序列调制相应的比较通道。这些位和下列输出信号对应：</p> <p>位 0 CC60 的调制</p> <p>位 1 COUT60 的调制</p> <p>位 2 CC61 的调制</p> <p>位 3 COUT61 的调制</p> <p>位 4 CC62 的调制</p> <p>位 5 COUT62 的调制</p> <p>调制的使能特性定义如下：</p> <p>0 禁止由 T13 的 PWM 序列调制相应的输出信号</p> <p>1 使能由 T13 的 PWM 序列调制相应的输出信号</p>
ECT130	7	rw	<p>定时器 T13 比较输出使能</p> <p>0 禁止 COUT63 的其他功能选择</p> <p>1 使能 COUT63 的其他功能选择，用于输出 T13 的 PWM 信号</p>
0	6	r	保留

符号	位序号	读写类型	功能描述
			读操作返回 0；应写入 0

捕获/比较单元 6

寄存器 TRPCTR 控制强制中断功能。TRPCTR 中存放着每路输出信号独立的使能位以及强制中断条件下选择不同操作的控制位。强制中断条件是输入引脚 $\overline{\text{CTRAP}}$ 为低电平，由（寄存器 IS 中的）位 TRPF 监控（反相电平）。当 TRPF = 1（强制中断输入有效）时，（寄存器 IS 中的）强制中断状态位 TRPS 被置 1。

TRPCTRL

强制中断控制寄存器，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
0					TRPM2	TRPM1	TRPM0
r					rw	rw	rw

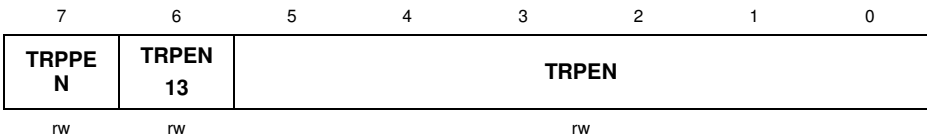
符号	位序号	读写类型	功能描述
TRPM0, TRPM1	[1:0]	rw	<p>强制中断模式控制位 0,1</p> <p>这两位定义了在强制中断条件再次无效时，输出信号如何退出中断状态。</p> <p>退出中断状态时，和驱动 PWM 的定时器同步可避免不需要的短脉冲。</p> <p>TRPM0 和 TRPM1 的组合如下：</p> <p>00 当检测到 T12（递增计数）的 0 匹配时退出强制中断状态（根据 TRPM2 返回正常操作）（和 T12 同步）</p> <p>01 当检测到 T13 的 0 匹配时退出强制中断状态（根据 TRPM2 返回正常操作）（和 T13 同步）</p> <p>10 保留</p> <p>11 不需要和 T12 或 T13 同步，立刻退出强制中断状态（根据 TRPM2 返回正常操作）</p>
TRPM2	2	rw	<p>强制中断模式控制位 2</p> <p>0 输入 $\overline{\text{CTRAP}}$ 一无效就退出强制中断状态（返回正常操作 = 位 TRPS = 0）。如果输入 $\overline{\text{CTRAP}}$ 为 1，位 TRPF 由硬件自动清零。如果</p>

符号	位序号	读写类型	功能描述
			<p>TRPF 为 0 并且检测到（由 TRPM0 和 TRPM1 决定的）同步条件时，位 TRPS 由硬件自动清零。</p> <p>1 输入 $\overline{\text{CTRAP}}$ 无效后，软件一旦复位 TRPF（TRPF 不被硬件清零）就退出强制中断状态（返回正常操作 = 位 TRPS = 0）。如果 TRPF 为 0 并且检测到（由 TRPM0 和 TRPM1 决定的）同步条件时，位 TRPS 由硬件自动清零。</p>
0	[7:3]	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

TRPCTRH

强制中断控制寄存器，高位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
TRPEN	[5:0]	rw	<p>强制中断使能控制</p> <p>置位这些位将使能下列所对应的输出信号的强制中断功能：</p> <p>位 0 CC60 的强制中断功能</p> <p>位 1 COUT60 的强制中断功能</p> <p>位 2 CC61 的强制中断功能</p> <p>位 3 COUT61 的强制中断功能</p> <p>位 4 CC62 的强制中断功能</p> <p>位 5 COUT62 的强制中断功能</p> <p>强制中断功能的使能特性定义如下：</p>

符号	位序号	读写类型	功能描述
			<p>0 禁止相应输出信号的强制中断功能。输出状态和位 TRPS 无关。</p> <p>1 使能相应输出信号的强制中断功能。位 TRPS = 1 时，输出信号被设置为被动态。</p>
TRPEN13	6	rw	<p>定时器 T13 的强制中断使能控制</p> <p>0 禁止 T13 的强制中断功能。即使当 TRPS = 1，（如果选择 T13 且被使能）定时器 T13 也提供产生 PWM 功能。</p> <p>1 使能 T13 的强制中断功能。位 TRPS = 1 时定时器 T13 的 PWM 输出信号被设置为被动态。</p>
TRPPEN	7	rw	<p>强制中断引脚使能</p> <p>0 禁止基于输入引脚 $\overline{\text{CTRAP}}$ 的强制中断功能。只能通过置位 TRPF 由软件产生强制中断。</p> <p>1 使能基于输入引脚 $\overline{\text{CTRAP}}$ 的强制中断功能。通过软件置位 TRPF 或 $\overline{\text{CTRAP}} = 0$ 产生强制中断。</p>

捕获/比较单元 6

寄存器 PSLR 定义了由模块的输出引脚驱动的被动态电平。被动态电平值由处于被动态的输出引脚驱动。输出引脚处于主动态时驱动和被动态电平反相的主动态电平。被动态电平可使被驱动的输出电平和所连接的功率级的驱动极性（反相或不反相）相匹配。

PSLR

被动态电平寄存器

复位值: 00_H

7	6	5	4	3	2	1	0	
PSL63	0	PSL						
rwh	r	rwh						

符号	位序号	读写类型	功能描述
PSL ¹⁾	[5:0]	rwh	<p>比较输出的被动态电平</p> <p>该位域定义了模块输出处于被动态时驱动的被动态电平值。</p> <p>位 0 输出 CC60 的被动态电平</p> <p>位 1 输出 COUT60 的被动态电平</p> <p>位 2 输出 CC61 的被动态电平</p> <p>位 3 输出 COUT61 的被动态电平</p> <p>位 4 输出 CC62 的被动态电平</p> <p>位 5 输出 COUT62 的被动态电平</p> <p>每位的电平值定义如下：</p> <p>0 被动态电平为 0。</p> <p>1 被动态电平为 1。</p>
PSL63 ²⁾	7	rwh	<p>输出 COUT63 的被动态电平</p> <p>该位定义了输出引脚 COUT63 的被动态电平值</p> <p>0 被动态电平为 0。</p> <p>1 被动态电平为 1。</p>
0	6	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

捕获/比较单元 6

¹⁾ 位域 PSL 具有映射寄存器，从而避免在更新输出电平值时产生不期望的脉冲。T12 映射传输时更新该位域。读操作读取实际使用位的值，写操作写入映射位。

²⁾ 位域 PSL63 具有映射寄存器，从而避免在更新输出电平值时产生不期望的脉冲。T13 映射传输时更新该位。读操作读取实际使用位的值，写操作写入映射位。

12.3.5.2 多通道控制

寄存器 MCMOUTS 中存放着多通道模式输出状态的控制位。另外，可选择合适的信号用于由霍尔传感器控制的块切换。该寄存器为寄存器 MCMOUT 的映射寄存器（可写），指示当前的有效信号。

MCMOUTSL

多通道模式输出映射寄存器，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
STRMCM	0	MCMP5					
w	r	rw					

符号	位序号	读写类型	功能描述
MCMP5	[5:0]	rw	多通道 PWM 序列映射 位域 MCMP5 为位域 MCMP 的映射位域。由寄存器 MCMCTR 定义的传输条件触发多通道映射传输。
STRMCM	7	w	MCMP5 的映射传输请求 对该位置位将立刻用写入位域 MCMP5 的值更新位域 MCMP。该功能可用软件触发更新。读取该位时始终为 0。 0 根据定义的硬件动作更新位域 MCMP。对位域 MCMP5 的写操作不会修改位域 MCMP 的值。 1 位域 MCMP 由位域 MCMP5 的写入值更新。
0	6	r	保留 读操作返回 0；应写入 0

MCMOUTSH

多通道模式输出映射寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
STRHP	0	CURHS			EXPHS		
w	r	rw			rw		

符号	位序号	读写类型	功能描述
EXPHS	[2:0]	rw	<p>期望霍尔序列映射</p> <p>位域 EXPHS 是位域 EXPH 的映射位域。当检测到霍尔输入引脚 CCPOS_x (x = 0 - 2) 的沿跳变时，该位域被传送到位域 EXPH 中。</p>
CURHS	[5:3]	rw	<p>当前霍尔序列映射</p> <p>位域 CURHS 是位域 CURH 的映射位域。当检测到霍尔输入引脚 CCPOS_x (x = 0 - 2) 的沿跳变时，该位域被传送到位域 CURH 中。</p>
STRHP	7	w	<p>霍尔序列映射传输请求</p> <p>对这些位置位将立刻用写入位域 CURHS 和 EXPHS 中的值更新位域 CURH 和 EXPH。该功能可用软件触发更新。读取该位时始终为 0。</p> <p>0 根据定义的硬件动作更新位域。对位域 CURHS 和 EXPHS 的写操作不会修改位域 CURH 和 EXPH 的值。</p> <p>1 位域 CURH 和 EXPH 由位域 CURHS 和 EXPHS 的写入值更新。</p>
0	6	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

寄存器 MCMOUT 规定了当前使用的多通道的控制位。

MCMOUTL

多通道模式输出寄存器，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
0	R	MCMP					
w	rh	rh					

符号	位序号	读写类型	功能描述
MCMP¹⁾	[5:0]	rh	<p>多通道 PWM 序列</p> <p>MCMP 的值通过传输映射从位域 MCMPS 写入。MCMP 中存放着多通道模式的输出序列。如果寄存器 MODCTR 中的 MCMEN 使能该模式，下列输出信号的输出状态可被修改：</p> <p>位 0 输出 CC60 的多通道状态</p> <p>位 1 输出 COUT60 的多通道状态</p> <p>位 2 输出 CC61 的多通道状态</p> <p>位 3 输出 COUT61 的多通道状态</p> <p>位 4 输出 CC62 的多通道状态</p> <p>位 5 输出 COUT62 的多通道状态</p> <p>多通道序列可将相关的输出设置为被动态。</p> <p>0 输出被设为被动态。T12 或 T13 产生的 PWM 不起作用</p> <p>1 可输出 T12 或 T13 产生的 PWM (根据寄存器 MODCTR)</p>
R	6	w	<p>提示标志</p> <p>该提示标志指明。映射传输发生后且 MCMEN = 0，该位被清零。</p> <p>0 未发生从 MCMPS 到 MCMP 的映射传输请求</p> <p>1 所选的触发源已产生从 MCMPS 到</p>

符号	位序号	读写类型	功能描述
			MCMP 的映射传输请求，但由于所选的同步条件还未满足，映射传输未发生
0	7	r	保留 读操作返回 0；应写入 0

¹⁾ 当 IDLE = 1 时，位域 MCMP 被清零。

MCMOUTH

多通道模式输出寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
0		CURH			EXPH		
r		rh			rh		

符号	位序号	读写类型	功能描述
EXPH¹⁾	[2:0]	rh	<p>期望霍尔序列</p> <p>EXPH 的值通过传输映射从位域 EXPHS 写入。每次检测到霍尔输入引脚的跳变沿之后进行比较，从而检测是否产生了下次期望的霍尔序列或者错误序列。</p> <p>如果霍尔输入引脚的当前霍尔序列和位域 EXPH 相等，置位 CHE（正确霍尔事件）并（当位 ENCHE 使能中断时）产生中断请求。</p> <p>如果霍尔输入引脚的当前霍尔序列和位域 CURH 和 EXPH 都不相等，置位 WHE（错误霍尔事件）并（当位 ENWHE 使能中断时）产生中断请求。</p>
CURH	[5:3]	rh	<p>当前霍尔序列</p> <p>CURH 的值通过传输映射从位域 CURHS 写入。每次检测到霍尔输入引脚的跳变沿之后进行比较，从而检测是否产生了下次期望的霍尔序列或者错误序列。</p> <p>如果当前的霍尔输入序列和位域 CURH 相等，霍尔输入引脚上检测到跳变沿无效（如，毛刺）。</p>
0	[7:6]	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

捕获/比较单元 6

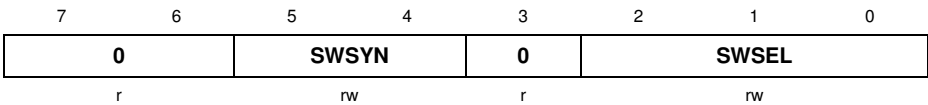
¹⁾ 寄存器 EXPH 和 CURH 中的位域和输入引脚 CCPOS_x ($x = 0 - 2$) 上的霍尔序列按照下面次序对应: (EXPH.2, EXPH.1, EXPH.0), (CURH.2, CURH.1, CURH.0), (CCPOS2, CCPOS1, CCPOS0)。

寄存器 MCMCTR 中存放着多通道模式功能的控制位。

MCMCTR

多通道模式控制寄存器

复位值: 00H



符号	位序号	读写类型	功能描述
SWSEL	[2:0]	rw	<p>切换选择</p> <p>位域 SWSEL 选择以下的触发请求源（下次多通道事件）触发从 MCMPS 到 MCMP 的映射传输。映射传输前，触发请求保存在提示标志 R 中；映射传输时标志 R 被自动清零。映射传输和位域 SWSYN 所选择的事件同步发生。</p> <p>000 不产生触发请求</p> <p>001 检测到 CCPOSx 上的正确霍尔事件</p> <p>010 检测到 T13 周期 - 匹配（递增计数）</p> <p>011 检测到 T12 1 - 匹配（递减计数）</p> <p>100 检测到 T12 通道 1 比较 - 匹配（相延迟功能）</p> <p>101 检测到 T12 周期 - 匹配（递增计数）；否则保留，不产生触发请求</p>
SWSYN	[5:4]	rw	<p>切换同步</p> <p>位域 SWSYN 触发已被请求的从 MCMPS 到 MCMP 的映射传输（标志 R 由 SWSEL 选择的事件置位）。该特性可使输出和用作调制的（T12 或 T13 的）PWM 同步。</p> <p>00 触发事件直接引起映射传输</p>

捕获/比较单元 6

符号	位序号	读写类型	功能描述
			01 T13 0 - 匹配触发映射传输 10 T12 0 - 匹配（递增计数）触发映射传输 11 保留，无操作
0	3, [7:6]	r	保留 读操作返回 0；应写入 0

注：只有位 MCMEN = 1 时，才能使能由硬件产生映射传输请求。

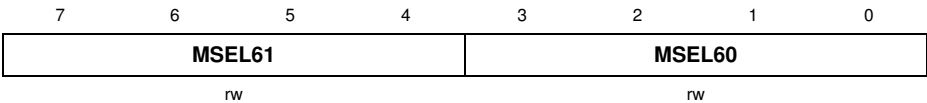
捕获/比较单元 6

寄存器 T12MSEL 中存放的控制位，用来选择定时器 T12 三路通道的捕获/比较功能。

T12MSELL

T12 捕获/比较模式选择寄存器，低位字节

复位值: 00_H



符号	位序号	读写类型	功能描述
MSEL60, MSEL61	[3:0], [7:4]	rw	<p>捕获/比较模式选择</p> <p>该位域选择定时器 T12 三路捕获/比较通道的工作模式。根据以下定义，每路通道 (n = 0 - 2) 可独立编程设定为比较或捕获模式：</p> <p>0000 比较输出被禁止；引脚 CC6n 和 COUT6n 可用作 I/O。无捕获操作。</p> <p>0001 比较结果从引脚 CC6n 输出，COUT6n 可用作 I/O。无捕获操作。</p> <p>0010 比较结果从引脚 COUT6n 输出，CC6n 可用作 I/O。无捕获操作。</p> <p>0011 比较结果从引脚 COUT6n 和 CC6n 输出。无捕获操作。</p> <p>01XX 双寄存器捕获模式，见表 12-5。</p> <p>1000 霍尔传感器模式，见表 12-6。为使能霍尔跳变沿检测，必须将 MSEL6x (x = 0 - 2) 全部设置为霍尔传感器模式。</p> <p>1001 类磁滞模式，见表 12-6。</p> <p>101X 多输入捕获模式，见表 12-7。</p>

符号	位序号	读写类型	功能描述
			11XX 多输入捕获模式，见表 12-7。

T12MSELH

T12 捕获/比较模式选择寄存器，高位字节

复位值: 00H

7	6	5	4	3	2	1	0
DBYP		HSYNC			MSEL62		
rw		rw			rw		

符号	位序号	读写类型	功能描述
MSEL62	[3:0]	rw	<p>捕获/比较模式选择</p> <p>该位域选择定时器 T12 三路捕获/比较通道的工作模式。根据以下定义，每路通道 (n = 0 - 2) 可独立编程设定为比较或捕获模式：</p> <p>0000 比较输出被禁止；引脚 CC6n 和 COUT6n 可用作 I/O。无捕获操作。</p> <p>0001 比较结果从引脚 CC6n 输出，COUT6n 可用作 I/O。无捕获操作。</p> <p>0010 比较结果从引脚 COUT6n 输出，CC6n 可用作 I/O。无捕获操作。</p> <p>0011 比较结果从引脚 COUT6n 和 CC6n 输出。无捕获操作。</p> <p>01XX 双寄存器捕获模式，见表 12-5。</p> <p>1000 霍尔传感器模式，见表 12-6。为使能霍尔跳变沿检测，必须将 MSEL6x (x = 0 - 2) 全部设置为霍尔传感器模式。</p> <p>1001 类磁滞模式，见表 12-6。</p> <p>101X 多输入捕获模式，见表 12-7。</p> <p>11XX 多输入捕获模式，见表 12-7。</p>
HSYNC	[6:4]	rw	<p>霍尔同步</p> <p>位域 HSYNC 定义了采样霍尔输入序</p>

符号	位序号	读写类型	功能描述
			<p>列、并和当前及期望的霍尔序列位域进行比较的触发源。在所有模式下，可由软件对 SWHC 写入 1 作为触发。</p> <p>000 任一输入引脚 CCPOSx 的任意升降沿触发采样。</p> <p>001 T13 的比较 - 匹配触发采样。</p> <p>010 T13 的周期 - 匹配触发采样。</p> <p>011 硬件触发霍尔采样被关闭。</p> <p>100 T12 (递增计数时) 的周期 - 匹配触发采样。</p> <p>101 T12 (递减计数时) 的 1 - 匹配触发采样。</p> <p>110 T12 通道 0 (递增计数时) 的比较 - 匹配触发采样。</p> <p>111 T12 通道 0 (递减计数时) 的比较 - 匹配触发采样。</p>
DBYP	7	rw	<p>延迟旁路</p> <p>位域 DBYP 决定了采样霍尔输入序列的触发源 (由 HSYNC 选择) 是否利用定时器 T12 的死区时间计数器 DTC0 产生附加延迟; 或者延迟被旁路。</p> <p>0 延迟旁路无效。在触发源信号有效之后死区时间计数器 DTC0 产生附加延迟。</p> <p>1 延迟旁路有效。采样霍尔序列不使用死区时间计数器 DTC0。</p>

注: 捕获模式下, CC6x 输入的所有跳变沿将置位寄存器 IS 中的相应中断状态标志。为了监控多输入捕获模式下 CCPOSx 输入端所选择的捕获事件, 检测到选中事件时置位相应通道的 CC6xST。必须由软件复位中断状态位和 CC6xST。

表 12-5 双寄存器捕获模式

描述	
双寄存器捕获模式	
0100	在输入引脚 CC6n 的上升沿将 T12 的内容保存在 CC6nR 中；在输入引脚 CC6n 的下将沿将 T12 的内容保存在 CC6nSR 中。
0101	在输入引脚 CC6n 的上升沿将 CC6nSR 中的内容复制到 CC6nR 中；T12 的实际计数值立即保存在映射寄存器 CC6nSR 中。该特性可用于测量 CC6n 引脚上连续两个上升沿之间的时间。COUT6n 是 I/O 引脚。
0110	在输入引脚 CC6n 的下降沿将 CC6nSR 中的内容复制到 CC6nR 中；T12 的实际计数值立即保存在映射寄存器 CC6nSR 中。该特性可用于测量 CC6n 引脚上连续两个下降沿之间的时间。COUT6n 是 I/O 引脚。
0111	在输入引脚 CC6n 的任意（升降）沿将 CC6nSR 中的内容复制到 CC6nR 中；T12 的实际计数值立即保存在映射寄存器 CC6nSR 中。该特性可用于测量 CC6n 引脚上连续两个跳变沿之间的时间。COUT6n 是 I/O 引脚。

表 12-6 组合的 T12 模式

描述	
组合的 T12 模式	
1000	<p>霍尔传感器模式：</p> <p>通道 0 为捕获模式，通道 1 和 2 为比较模式。检测到有效的霍尔事件，将 T12 的内容捕获到 CC60 中（作为实际转速的参考值）；CC61 可产生霍尔事件和输出切换之间的相延迟；如果期望的霍尔事件出现太晚，CC62 可作为超时触发。若使用霍尔信号，MSEL0, MSEL1 和 MSEL2 必须设置为 1000_B。该模式下，在检测到有效的霍尔事件后定时器 T12 的内容被捕获至 CC60 并复位；为了避免噪声影响，当检测到霍尔输入的跳变沿时，启动通道 0 的死区时间计数器，计数到 000001_B 时，采样霍尔输入并进行序列比较。</p>
1001	<p>带有死区时间的类磁滞控制模式：</p> <p>CCPOSx 输入信号的下降沿用作复位 CC6nST。因此输出信号立刻被切换到被动态，当 CCPOSx 信号变高且 CC6nST 由比较事件置位后，输出切换回主动态（带有死区时间）。</p>

表 12-7 多输入捕获模式

描述	
多输入捕获模式	
1010	在输入引脚 CC6n 的上升沿将 T12 的定时器值保存在 CC6nR 中；在输入引脚 CCPOSx 的下降沿将 T12 的定时器值保存在 CC6nSR 中。
1011	在输入引脚 CC6n 的下降沿将 T12 的定时器值保存在 CC6nR 中；在输入引脚 CCPOSx 的上升沿将 T12 的定时器值保存在 CC6nSR 中。
1100	在输入引脚 CC6n 的上升沿将 T12 的定时器值保存在 CC6nR 中；在输入引脚 CCPOSx 的上升沿将 T12 的定时器值保存在 CC6nSR 中。
1101	在输入引脚 CC6n 的下降沿将 T12 的定时器值保存在 CC6nR 中；在输入引脚 CCPOSx 的下降沿将 T12 的定时器值保存在 CC6nSR 中。
1110	在输入引脚 CC6n 的任意沿将 T12 的定时器值保存在 CC6nR 中；在输入引脚 CCPOSx 的任意（升降）沿将 T12 的定时器值保存在 CC6nSR 中。
1111	保留（无捕获或比较动作）

12.3.6 中断控制寄存器

ISL

捕获/比较中断状态寄存器，低位字节

复位值: 00H

7	6	5	4	3	2	1	0
T12PM	T12OM	ICC62F	ICC62R	ICC61F	ICC61R	ICC60F	ICC60R
rh	rh	rh	rh	rh	rh	rh	rh

符号	位序号	读写类型	功能描述
ICC60R, ICC61R, ICC62R	0, 2, 4	rh	捕获，比较 - 匹配上升沿标志 比较模式下，T12 递增计数时检测到比较 - 匹配；捕获模式下，检测到输入 CC6x (x = 0 - 2) 的上升沿 0 该位复位后事件还未发生 1 已检测到以上所描述的事件
ICC60F, ICC61F, ICC62F	1, 3, 5	rh	捕获，比较 - 匹配下降沿标志 比较模式下，T12 递减计数时检测到比较 - 匹配；捕获模式下，检测到输入 CC6x (x = 0 - 2) 的下降沿 0 该位复位后事件还未发生 1 已检测到以上所描述的事件
T12OM	6	rh	定时器 T12 1 - 匹配标志 0 该位复位后还未检测到定时器 T12 (递减计数时) 的 1 - 匹配 1 已检测到定时器 T12 (递减计数时) 的 1 - 匹配
T12PM	7	rh	定时器 T12 周期 - 匹配标志 0 该位复位后还未检测到定时器 T12 (递增计数时) 的周期 - 匹配 1 已检测到定时器 T12 (递增计数时) 的周期 - 匹配

ISH

捕获/比较中断状态寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
STR	IDLE	WHE	CHE	TRPS	TRPF	T13PM	T13CM
rh	rh	rh	rh	rh	rh	rh	rh

符号	位序号	读写类型	功能描述
T13CM	0	rh	<p>定时器 T13 比较 - 匹配标志</p> <p>0 该位复位后还未检测到定时器 T13 的比较 - 匹配</p> <p>1 已检测到定时器 T13 的比较 - 匹配</p>
T13PM	1	rh	<p>定时器 T13 周期 - 匹配标志</p> <p>0 该位复位后还未检测到定时器 T13 的周期 - 匹配</p> <p>1 检测到定时器 T13 的周期 - 匹配</p>
TRPF	2	rh	<p>强制中断标志</p> <p>若 TRPPEN = 1 且 $\overline{CTRAP} = 0$ 时硬件置位强制中断标志 TRPF，软件也可对该位置位。如果 TRPM2 = 0，输入 \overline{CTRAP} 变高时（TRPPEN = 1）硬件复位 TRPF；如果 TRPM2 = 1，必须软件复位 TRPF 以退出强制中断状态。</p> <p>0 还未检测到强制中断条件</p> <p>1 已检测到强制中断条件（输入 $\overline{CTRAP} = 0$ 或由软件控制）</p>
TRPS¹⁾	3	rh	<p>强制中断状态</p> <p>0 强制中断未发生</p> <p>1 强制中断已发生。位 TRPF = 1 时置位 TRPS。根据寄存器 TRPCTR 所选择的模式复位 TRPS。</p>
CHE²⁾	4	rh	<p>正确霍尔事件</p>

符号	位序号	读写类型	功能描述
			0 该位复位后还未检测到跳变至正确的（期望）霍尔事件 1 已检测到跳变至正确的（期望）霍尔事件
WHE³⁾	5	rh	错误霍尔事件 0 该位复位后还未检测到跳变至错误的（非期望）霍尔事件 1 已检测到跳变至错误的（非期望）霍尔事件
IDLE⁴⁾	6	rh	空闲状态 该位和 WHE （错误霍尔事件）一起被置位， IDLE 必须由软件复位 0 无动作 1 位域 MCMP 被清零，将所选的输出设置为被动态
STR	7	rh	多通道模式映射传输请求 多通道模式下，当发生从 MCMOUTS 到 MCMOUT 的传输映射时该位被置位 0 未发生传输映射 1 已发生传输映射

¹⁾ 在强制中断状态，所选的输出设为被动态。被动态时所驱动的逻辑电平由寄存器 **PSLR** 中的相应位定义。如果强制中断条件不再有效，但所选的同步条件还未发生，**TRPS = 1** 且 **TRPF = 0** 可以发生。

²⁾ 在每个有效的霍尔沿，将 **EXPH** 的内容和引脚 **CCPOSx** 序列进行比较，若二者相等，置位 **CHE**。

³⁾ 在每个有效的霍尔沿，将 **EXPH** 的内容和引脚 **CCPOSx** 序列进行比较，若 **CURH** 及 **EXPH** 与 **CCPOSx** 均不相等，置位 **WHE**（错误的霍尔事件）。

⁴⁾ 只要 **IDLE = 1**，位域 **MCMP** 保持为 0。

注：寄存器 IS 中不是所有位均可产生中断，还加入了状态位，这些位的置位和复位操作和其他位相似。

注：中断的产生和寄存器 IS 中的值无关，例如：相应位已被置位，将产生中断（若被使能）。检测到寄存器 IS 中相应位（硬件或软件产生）的置位条件时，触发产生中断。

捕获/比较单元 6

注：比较模式（及霍尔模式）下，只有在定时器运行（TxR = 1）时才能产生和定时器相关的中断。捕获模式下，定时器 T12 不工作也能产生捕获中断。

ISSL

捕获/比较中断状态置位寄存器，低位字节

复位值: 00_H

7	6	5	4	3	2	1	0
ST12PM	ST12OM	SCC62F	SCC62R	SCC61F	SCC61R	SCC60F	SCC60R
w	w	w	w	w	w	w	w

符号	位序号	读写类型	功能描述
SCC60R	0	w	置位捕获，比较 - 匹配上升沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC60R
SCC60F	1	w	置位捕获，比较 - 匹配下降沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC60F
SCC61R	2	w	置位捕获，比较 - 匹配上升沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC61R
SCC61F	3	w	置位捕获，比较 - 匹配下降沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC61F
SCC62R	4	w	置位捕获，比较 - 匹配上升沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC62R
SCC62F	5	w	置位捕获，比较 - 匹配下降沿标志 0 无操作 1 将置位寄存器 IS 中的 ICC62F
ST12OM	6	w	置位定时器 T12 1 - 匹配标志 0 无操作 1 将置位寄存器 IS 中的 T12OM
ST12PM	7	w	置位定时器 T12 周期 - 匹配标志

符号	位序号	读写类型	功能描述
			0 无操作
			1 将置位寄存器 IS 中的 T12PM

注：如果硬件置位相应的状态标志引起中断，软件置位具有相同的作用。

ISSH

捕获/比较中断状态置位寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
SSTR	SIDLE	SWHE	SCHE	SWHC	STRPF	ST13PM	ST13CM
w	w	w	w	w	w	w	w

符号	位序号	读写类型	功能描述
ST13CM	0	w	置位定时器 T13 比较 - 匹配标志 0 无操作 1 将置位寄存器 IS 中的 T13CM
ST13PM	1	w	置位定时器 T13 周期 - 匹配标志 0 无操作 1 将置位寄存器 IS 中的 T13PM
STRPF	2	w	置位强制中断标志 0 无操作 1 将置位寄存器 IS 中的 TRPF 和 TRPS
SWHE	3	w	软件霍尔比较 0 无操作 1 触发霍尔比较
SCHE	4	w	置位正确霍尔事件标志 0 无操作 1 将置位寄存器 IS 中的 CHE
SWHE	5	w	置位错误霍尔事件标志 0 无操作 1 将置位寄存器 IS 中的 WHE
SIDLE	6	w	置位 IDLE 标志 0 无操作 1 将置位寄存器 IS 中的 IDLE
SSTR	7	w	置位 STR 标志

符号	位序号	读写类型	功能描述
			0 无操作
			1 将置位寄存器 IS 中的 STR

捕获/比较单元 6

寄存器 ISR 中存放着各自的中断请求复位控制位，软件复位相应标志。

ISRL
捕获/比较中断状态复位寄存器，低位字节
复位值: 00H

7	6	5	4	3	2	1	0
RT12PM	RT12OM	RCC62F	RCC62R	RCC61F	RCC61R	RCC60F	RCC60R
w	w	w	w	w	w	w	w

符号	位序号	读写类型	功能描述
RCC60R	0	w	复位捕获，比较 - 匹配上升沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC60R
RCC60F	1	w	复位捕获，比较 - 匹配下降沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC60F
RCC61R	2	w	复位捕获，比较 - 匹配上升沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC61R
RCC61F	3	w	复位捕获，比较 - 匹配下降沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC61F
RCC62R	4	w	复位捕获，比较 - 匹配上升沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC62R
RCC62F	5	w	复位捕获，比较 - 匹配下降沿标志 0 无操作 1 将复位寄存器 IS 中的 ICC62F
RT12OM	6	w	复位定时器 T12 1 - 匹配标志 0 无操作 1 将复位寄存器 IS 中的 T12OM

符号	位序号	读写类型	功能描述
RT12PM	7	w	复位定时器 T12 周期 - 匹配标志 0 无操作 1 将复位寄存器 IS 中的 T12PM

ISRH

捕获/比较中断状态复位寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
RSTR	RIDLE	RWHE	RCHE	0	RTRPF	RT13PM	RT13CM
w	w	w	w	r	w	w	w

符号	位序号	读写类型	功能描述
RT13CM	0	w	复位定时器 T13 比较 - 匹配标志 0 无操作 1 将复位寄存器 IS 中的 T13CM
RT13PM	1	w	复位定时器 T13 周期 - 匹配标志 0 无操作 1 将复位寄存器 IS 中的 T13PM
RTRPF	2	w	复位强制中断标志 0 无操作 1 将复位寄存器 IS 中的 TRPF（当输入 $\overline{CTRAP} = 0$ 且 TRPEN = 1 时，该位不起作用）
RCHE	4	w	复位正确霍尔事件标志 0 无操作 1 将复位寄存器 IS 中的 CHE
RWHE	5	w	复位错误霍尔事件标志 0 无操作 1 将复位寄存器 IS 中的 WHE
RIDLE	6	w	复位 IDLE 标志 0 无操作 1 将复位寄存器 IS 中的 IDLE
RSTR	7	w	复位 STR 标志 0 无操作 1 将复位寄存器 IS 中的 STR

符号	位序号	读写类型	功能描述
0	3	r	保留 读操作返回 0；应写入 0

IENL
捕获/比较中断使能寄存器，低位字节
复位值: 00_H

7	6	5	4	3	2	1	0
EN T12PM	EN T12OM	EN CC62F	EN CC62R	EN CC61F	EN CC61R	EN CC60F	EN CC60R
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
ENCC60R	0	rw	通道 0 捕获，比较 - 匹配上升沿中断使能 0 当寄存器 IS 中的 ICC60R 被置位时，不产生中断 1 当寄存器 IS 中的 ICC60R 被置位时产生中断。由位域 INPCC60 选择被激活的中断线。
ENCC60F	1	rw	通道 0 捕获，比较 - 匹配下降沿中断使能 0 当寄存器 IS 中的 ICC60F 被置位时，不产生中断 1 当寄存器 IS 中的 ICC60F 被置位时产生中断。由位域 INPCC60 选择被激活的中断线。
ENCC61R	2	rw	通道 1 捕获，比较 - 匹配上升沿中断使能 0 当寄存器 IS 中的 ICC61R 被置位时，不产生中断 1 当寄存器 IS 中的 ICC61R 被置位时产生中断。由位域 INPCC61 选择被激活的中断线。
ENCC61F	3	rw	通道 1 捕获，比较 - 匹配下降沿中断使能 0 当寄存器 IS 中的 ICC61F 被置位时，不产生中断 1 当寄存器 IS 中的 ICC61F 被置位

捕获/比较单元 6

符号	位序号	读写类型	功能描述
			时产生中断。由位域 INPCC61 选择被激活的中断线。
ENCC62R	4	rw	通道 2 捕获, 比较 - 匹配上升沿中断使能 0 当寄存器 IS 中的 ICC62R 被置位时, 不产生中断 1 当寄存器 IS 中的 ICC62R 被置位时产生中断。由位域 INPCC62 选择被激活的中断线。
ENCC62F	5	rw	通道 2 捕获, 比较 - 匹配下降沿中断使能 0 当寄存器 IS 中的 ICC62F 被置位时, 不产生中断 1 当寄存器 IS 中的 ICC62F 被置位时产生中断。由位域 INPCC62 选择被激活的中断线。
ENT12OM	6	rw	T12 1 - 匹配中断使能 0 当寄存器 IS 中的 T12OM 被置位时, 不产生中断 1 当寄存器 IS 中的 T12OM 被置位时产生中断。由位域 INPT12 选择被激活的中断线。
ENT12PM	7	rw	T12 周期 - 匹配中断使能 0 当寄存器 IS 中的 T12PM 被置位时, 不产生中断 1 当寄存器 IS 中的 T12PM 被置位时产生中断。由位域 INPT12 选择被激活的中断线。

IENH

捕获/比较中断使能寄存器，高位字节

复位值: 00_H

7	6	5	4	3	2	1	0
ENSTR	ENIDLE	ENWHE	ENCHE	0	ENTRPF	EN T13PM	EN T13CM
rw	rw	rw	rw	r	rw	rw	rw

符号	位序号	读写类型	功能描述
ENT13CM	0	rw	<p>T13 比较 - 匹配中断使能</p> <p>0 当寄存器 IS 中的 T13CM 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 T13CM 被置位时产生中断。由位域 INPT13 选择被激活的中断线。</p>
ENT13PM	1	rw	<p>T13 周期 - 匹配中断使能</p> <p>0 当寄存器 IS 中的 T13PM 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 T13PM 被置位时产生中断。由位域 INPT13 选择被激活的中断线。</p>
ENTRPF	2	rw	<p>强制中断标志中断使能</p> <p>0 当寄存器 IS 中的 TRPF 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 TRPF 被置位时产生中断。由位域 INPERR 选择被激活的中断线。</p>
ENCHE	4	rw	<p>正确霍尔事件中断使能</p> <p>0 当寄存器 IS 中的 CHE 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 CHE 被置位时产生中断。由位域 INPCHE 选择被激活的中断线。</p>
ENWHE	5	rw	<p>错误霍尔事件中断使能</p>

符号	位序号	读写类型	功能描述
			<p>0 当寄存器 IS 中的 WHE 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 WHE 被置位时产生中断。由位域 INPEER 选择被激活的中断线。</p>
ENIDLE	6	rw	<p>空闲使能</p> <p>该位使能检测到错误霍尔事件（位 WHE 被置位）后自动进入空闲状态（位 IDLE 被置位）。空闲状态下位域 MCMP 自动被清零。</p> <p>0 检测到错误霍尔事件时，位 IDLE 不自动被置位。</p> <p>1 检测到错误霍尔事件时，位 IDLE 自动被置位</p>
ENSTR	7	rw	<p>多通道模式映射传输中断使能</p> <p>0 当寄存器 IS 中的 STR 被置位时，不产生中断。</p> <p>1 当寄存器 IS 中的 STR 被置位时产生中断。由位域 INPCHE 选择被激活的中断线</p>
0	3	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

INPL

捕获/比较中断节点指针寄存器，低位字节

复位值: 40_H

7	6	5	4
3	2	1	0
INPCHE		INPCC62	
rw		rw	
INPCC61		INPCC60	
rw		rw	

符号	位序号	读写类型	功能描述
INPCC60	[1:0]	rw	<p>通道 0 中断的中断节点指针</p> <p>该位域定义了当位 ICC60R（若由位域 ENCC60R 使能）或 ICC60F（若由位域 ENCC60F 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1 10 选择中断输出线 SR2 11 选择中断输出线 SR3</p>
INPCC61	[3:2]	rw	<p>通道 1 中断的中断节点指针</p> <p>该位域定义了当位 ICC61R（若由位域 ENCC61R 使能）或 ICC61F（若由位域 ENCC61F 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1 10 选择中断输出线 SR2 11 选择中断输出线 SR3</p>
INPCC62	[5:4]	rw	<p>通道 2 中断的中断节点指针</p> <p>该位域定义了当位 ICC62R（若由位域 ENCC62R 使能）或 ICC62F（若由位域 ENCC62F 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1</p>

符号	位序号	读写类型	功能描述
			10 选择中断输出线 SR2 11 选择中断输出线 SR3
INPCHE	[7:6]	rw	CHE 中断的中断节点指针 该位域定义了当位 CHE（若由位域 ENCHE 使能）或 STR（若由位域 ENSTR 使能）被置位时所激活的中断输出线。 00 选择中断输出线 SR0 01 选择中断输出线 SR1 10 选择中断输出线 SR2 11 选择中断输出线 SR3

INPH

捕获/比较中断节点指针寄存器，高位字节

复位值: 39_H

7	6	5	4
3	2	1	0
0		INPT13	INPT12
r		rw	rw
		INPERR	
			rw

符号	位序号	读写类型	功能描述
INPERR	[1:0]	rw	<p>错误中断的中断节点指针</p> <p>该位域定义了当位 TRPF（若由位域 ENTRPF 使能）或 WHE（若由位域 ENWHE 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1 10 选择中断输出线 SR2 11 选择中断输出线 SR3</p>
INPT12	[3:2]	rw	<p>定时器 T12 中断的中断节点指针</p> <p>该位域定义了当位 T12OM（若由位域 ENT12OM 使能）或 T12PM（若由位域 ENT12PM 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1 10 选择中断输出线 SR2 11 选择中断输出线 SR3</p>
INPT13	[5:4]	rw	<p>定时器 T13 中断的中断节点指针</p> <p>该位域定义了当位 T13CM（若由位域 ENT13CM 使能）或 T13PM（若由位域 ENT13PM 使能）被置位时所激活的中断输出线。</p> <p>00 选择中断输出线 SR0 01 选择中断输出线 SR1</p>

符号	位序号	读写类型	功能描述
			10 选择中断输出线 SR2 11 选择中断输出线 SR3
0	[7:6]	r	保留 读操作返回 0；应写入 0

13 模数转换单元

XC866 中包含一个带有多选一八路模拟输入通道、10 位高性能模数转换单元（ADC）。ADC 采用逐次逼近技术，最多可对 8 种不同的模拟通道的电压电平进行转换。

特性：

- 逐次逼近
- 8 位或 10 位精度
（总计不可校正误差分别为 ± 1 LSB 和 ± 2 LSB）
- 8 路模拟通道
- 四个独立的结果寄存器
（可配置为 FIFO）
- 用于 CPU 低速访问的结果数据保护
（待读模式）
- 单次转换模式
- 自动扫描功能
- 转换结果边界检测
- 数据压缩滤波
（最多累加 2 个转换结果）
- 两个独立的、优先级可编程的转换请求源
- 转换请求触发方式可选择
- 灵活的中断产生方式，中断服务节点可配置
- 采样时间可编程
- 时钟分频可编程
- 转换运行的取消/重启特性
- 集成采样与保持电路
- 偏移误差补偿
- 低功耗模式

13.1 结构概述

ADC 模块由两大部分组成，即模拟部分和数字部分，每部分包含独立的功能模块。

模拟部分包括：

- 模拟输入多路选择
（选择转换通道）
- 模拟转换
（如，作为 ADC 组成部分的电容网络和比较器）
- 模拟转换的数字控制部分
（控制模数转换过程并产生转换结果）

数字部分定义和控制 ADC 模块的功能，包括：

- 数字数据和转换请求的处理
（控制转换触发机制并处理转换结果）
- 器件内部数据总线的总线接口
（控制中断和寄存器访问）

ADC 模块的方框图如图 13-1 所示。模拟输入通道 x ($x= 0 - 7$) 对应端口引脚 P2.x/ANx。

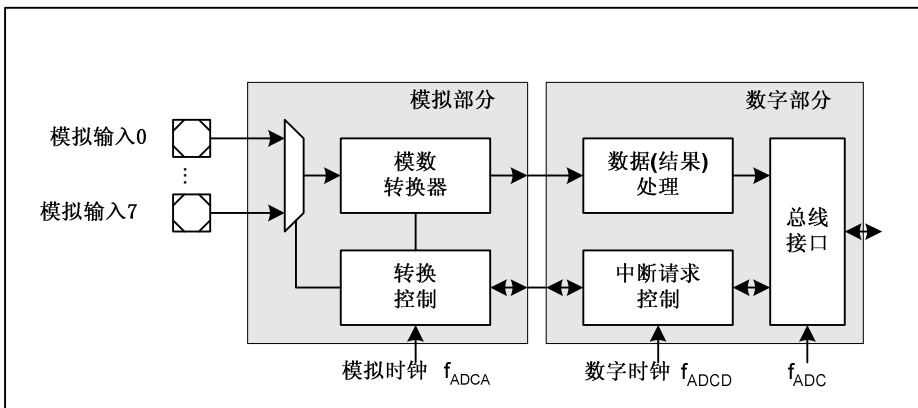


图 13-1 ADC 组成单元

13.2 时钟方案

公共模块时钟 f_{ADC} 产生 ADC 模块中模拟和数字部分所需的各种时钟信号：

- f_{ADCA} 为模拟部分的输入时钟。
- f_{ADCI} 为模拟部分的内部时钟（定义转换时间长度和采样时间的基准），该时钟基于输入时钟 f_{ADCA} 在模拟部分内产生，用来产生模拟单元所需的正确占空比。
- f_{ADCD} 为数字部分的输入时钟。该时钟用于仲裁器（定义仲裁周期的长短）和其他数字控制单元（如，寄存器和中断的产生）。

模拟部分内部时钟 f_{ADCI} 的最大频率限制在 10MHz。因此，ADC 模块的时钟预分频值必须编程设定保证 f_{ADCI} 不会超过 10MHz。预分频因子由寄存器 GLOBCTR 中的位域 CTC 选择。当无需 ADC 工作在最大工作性能时，可选择预分频因子为 32。

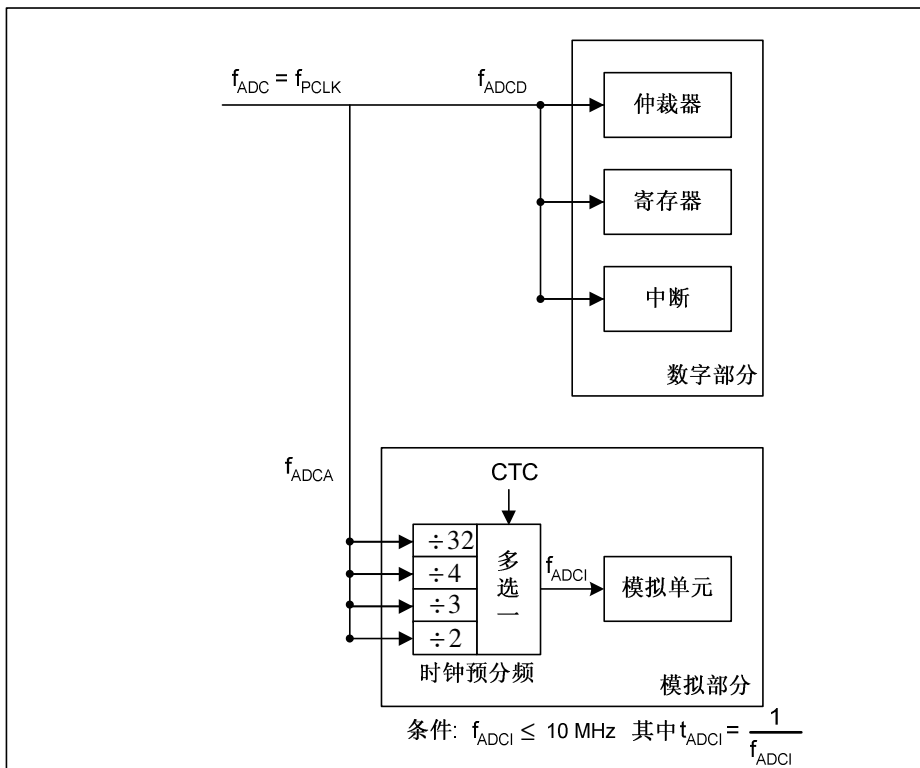


图 13-2 时钟方案

模块时钟 $f_{ADC} = 26.7 \text{ MHz}$ 对应的模拟时钟频率 f_{ADCI} 选择见表 13-1。

表 13-1 f_{ADCI} 频率选择

模块时钟 f_{ADC}	CTC	预分频因子	模拟时钟 f_{ADCI}
26.7 MHz	00 _B	÷ 2	13.3 MHz (不可用)
	01 _B	÷ 3	8.9 MHz
	10 _B	÷ 4	6.7 MHz
	11 _B (缺省值)	÷ 32	833.3 kHz

因为 f_{ADCI} 不能超过 10MHz， f_{ADC} 为 26.7 MHz 时位 CTC 不应设置为 00_B。减速模式下 f_{ADC} 可能减至 13.3 MHz，或 6.7 MHz 等，只要 f_{ADCI} 不会超过 10MHz，模拟时钟 CTC 可设置为 00_B。但是必须注意，若减速模式下 f_{ADC} 速度过慢，可能导致电容器电荷损失而造成模数转换出错增多。

13.2.1 转换时间

模数转换过程由下列阶段组成：

- 同步阶段 (t_{SYN})
- 采样阶段 (t_S)
- 转换阶段
- 结果写入阶段 (t_{WR})

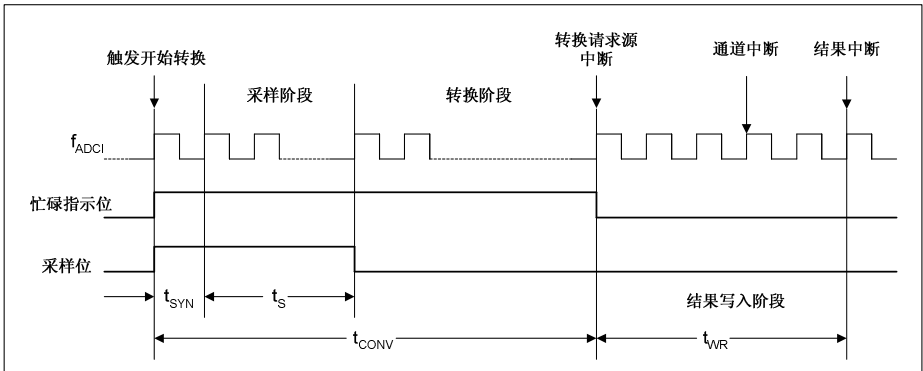


图 13-3 转换时序

同步阶段 t_{SYN}

触发开始转换（数字部分）和开始采样（模拟部分）之间需要一个 f_{ADCI} 周期进行同步，转换开始启动将置位忙碌指示位（BUSY）和采样位（SAMPLE）。

采样阶段 t_S

该阶段采样模拟输入电压。内部电容器阵列连接到选中的模拟输入通道上，加载需转换的模拟电压。模拟电压还被送至内部的电压比较器。采样开始时，寄存器 GLOBSTR 中的 SAMPLE 和 BUSY 标志被置位。该阶段对所有模拟输入通道都一样，并由寄存器 INPCR0 中的位 STC 控制：

$$t_S = (2 + STC) \times t_{ADCI} \quad [13.1]$$

转换阶段

在该阶段，采用二进制权重电容网络的逐次逼近技术，将模拟电压转换为 8 位或 10 位的数字量。转换开始时，SAMPLE 标志复位（表明采样结束），BUSY 标志继续置位。只有在转换结束且相应的转换请求源（触发转换的请求源）中断产生时 BUSY 标志才复位。

结果写入阶段 t_{WR}

转换阶段结束时，在执行完极限检查、三个 f_{ADCI} 周期之后产生相应（转换通道）的通道中断。一旦转换结果写入目标结果寄存器，即产生结果中断。

转换总时间 t_{CONV}

模数转换总时间（同步 + 采样 + 电荷重分配） t_{CONV} 由下面公式给出：

$$t_{CONV} = t_{ADC} \times (1 + r \times (3 + n + STC)) \quad [13.2]$$

其中

$r = CTC + 2$ （ $CTC = 00_B, 01_B$ 或 10_B ），

$r = 32$ （ $CTC = 11_B$ ），

$CTC =$ 转换时间控制，

$STC =$ 采样时间控制，

$n = 8$ 或 10 （分别对应 8 位或 10 位转换），

$t_{ADC} = 1 / f_{ADC}$

举例：

$STC = 00H$ ，

$CTC = 01B$ ，

$f_{ADC} = 26.7 \text{ MHz}$ ，

$n = 10$ ，

$$t_{CONV} = t_{ADC} \times (1 + 3 \times (3 + 10 + 0)) = 1.5 \mu s$$

13.3 低功耗模式

为了降低功耗，当无需模数转换时，可将 ADC 模块部分或完全禁止：

- ADC 模块的模拟部分可通过复位 ANON 来禁止。这会终止时钟 f_{ADCI} 的产生从而降低功耗。只有再次使能模拟部分 ($ANON = 1$) 才可能进行转换。唤醒大约需要 100 ns。

禁止 ADC 模拟部分的寄存器描述参见章节 13.7.1。

- 如果完全不需要 ADC 功能，可关闭时钟输入 (f_{ADC}) 完全禁止该模块，从而最大程度节省了功耗。可通过置位寄存器 PMCON1 中的 ADC_DIS 来实现，具体描述见下表。更多信息参见章节 8.1.4 的外设时钟管理。

PMCON1

功率模式控制寄存器 1

复位值: 00H

7	6	5	4	3	2	1	0
0				T2_DIS	CCU_DIS	SSC_DIS	ADC_DIS
r				rw	rw	rw	rw



阴影位的功能此处不做描述

符号	位序号	读写类型	功能描述
ADC_DIS	0	rw	ADC 禁止请求位，高有效 0 ADC 正常工作（缺省状态） 1 ADC 被禁止
0	[7:4]	r	保留 读操作返回 0；应写入 0

13.4 功能描述

ADC 的功能包括:

- 带有各自独立寄存器的两个不同的（顺序和并列）转换请求源。请求源用来触发由外部事件（同步至 PWM 信号）、序列方案等引起的转换。
- 仲裁器规则的扫描请求源，以确定对优先级最高的通道进行下次转换。每个请求源的优先级可单独编程设定，从而获得的灵活性可覆盖期望的应用领域。
- 八路通道中每路的控制寄存器定义了每种模拟输入口的行为（例如中断行为，结果寄存器指针，通道级别指针，等）。
- 输入综合控制寄存器给出通用的通道控制信息（采样时间）。
- 四个结果寄存器（不是每个模拟输入通道对应一个结果寄存器）用来保存转换结果并控制数据压缩。可产生 FIFO 型的结果数据。
- 转换结果的累加阶段将最新的转换结果累加到目标结果寄存器所保存的转换结果上，使得 CPU 在低频工作时，仍可快速、连续进行转换而不会造成数据丢失。

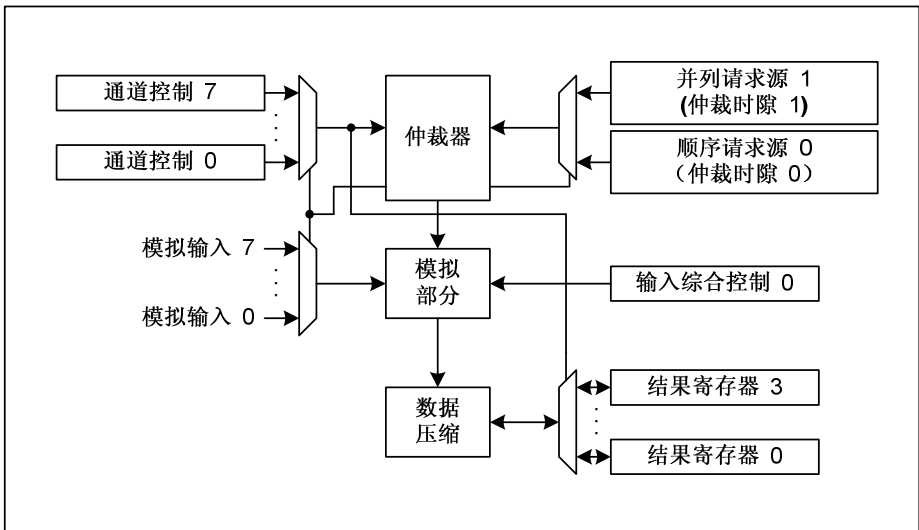


图 13-4 ADC 功能框图

13.4.1 请求源仲裁

仲裁器可由位 ARBM 选择工作在两种模式：

- 持续模式

该模式下，即使无转换请求挂起，仲裁器仍将连续查询请求源。

- 由挂起的转换请求启动仲裁

该模式下，只有在至少有一个转换请求挂起的情况下，仲裁器才开始查询请求源。

一旦开始仲裁，仲裁器查询两个转换请求源（仲裁时隙 x 的请求源 x ， $x = 0-1$ ）以确定优先级最高的的模拟通道进行转换。每个仲裁时隙，仲裁器查询请求源的挂起请求信号（REQPND）和通道编号有效信号（REQCHNRV）。所有仲裁时隙的总和称为一个仲裁周期。仲裁时隙在参与仲裁之前必须被使能（ASENx = 1）。

每个请求源均有由位 PRIOx 设定的优先级。仲裁器从请求源 0（仲裁时隙 0）开始，首先检查请求源是否有转换请求挂起（REQPND）。如果发现不止一个相同优先级的请求源有转换请求挂起，则选择对首先发现的请求源所指定的通道进行转换。仲裁器还要检查 REQCHNRV 信号，只有当 REQCHNRV = 1（且 REQPND = 1）时才开始进行转换。如果两个请求源设定的优先级相同，由于请求源 0 和仲裁时隙 0 相连，则先对请求源 0 所指定的通道进行转换。

一个完整的仲裁周期 t_{ARB} 计算如下：

$$t_{ARB} = 4 \times t_{ADCD} \quad [13.3]$$

中断优先级和仲裁控制的寄存器描述请参见章节 13.7.2。

13.4.2 转换启动模式

每个仲裁周期结束时，仲裁器应已确定了优先级最高的请求源和一个挂起转换请求。仲裁器保存仲裁结果，即通道编号，采样时间以及目标结果寄存器，为下面的工作做准备。

如果模拟部分正处于空闲状态，则立即开始转换。如果已有转换正在处理中，仲裁结果和当前正在进行转换的优先级进行比较。如果当前转换的优先级为同级或高优先级，继续完成当前转换，该转换结束后可立即进行下一个转换。只要模拟部分一空闲且仲裁器输出转换请求，就开始进行转换。

如果新的转换请求具有高优先级（和当前转换相比），有两种转换启动模式（由位 CSM_x 来选择， $x = 0 - 1$ ）：

- 等待开始：

该模式下，当前转换正常完成，该转换结束后立即处理已挂起的转换请求。转换尽快开始。

- 取消-插入-重复：

该模式下，一旦发现高优先级的、新的转换请求，则立即中止当前转换，并尽快开始新的转换。被中止的转换请求保存在申请该转换的请求源中，参与下轮仲裁。请求源有效（包括挂起转换或当前转换）时优先级不能由软件改变。转换运行的最后 3 个时钟周期内中止不被响应。

转换启动控制的相关寄存器描述参见章节 13.7.2。

13.4.3 通道控制

每路通道均有各自的控制信息来定义保存转换结果的目标结果寄存器（见章节 13.7.4）。所有通道唯一的公共控制信息为采样时间，由输入综合控制寄存器定义（见章节 13.7.5）。

13.4.4 顺序请求源

13.4.4.1 概述

位于仲裁时隙 0 的顺序请求源请求对通道 0 至通道 7 进行顺序转换。队列寄存器保存被请求通道编号及一些附加控制信息，从而通道转换顺序不受通道次序的限制，可自由编程设定。附加控制信息用来（在请求通道转换完成时）使能请求源中断以及自动添加过程。

顺序请求源由队列寄存器（Q0R0），备份寄存器（QBUR0）和模式控制寄存器（QMR0）组成。最新请求的转换被中止后，备份寄存器保存其信息。如果备份寄存器中包含被中止的转换请求（ $V = 1$ ），在处理队列中的转换请求之前，先处理该转换请求；也就是说，只有当备份寄存器中的位 V 被清零时才可开始进行队列中的转换。如果备份寄存器中的位 V 没有被置位，当转换开始时队列寄存器中的位 V 被复位。如果备份或队列寄存器中有一个有效的请求（ $V = 1$ ），请求源可参与转换请求源仲裁。

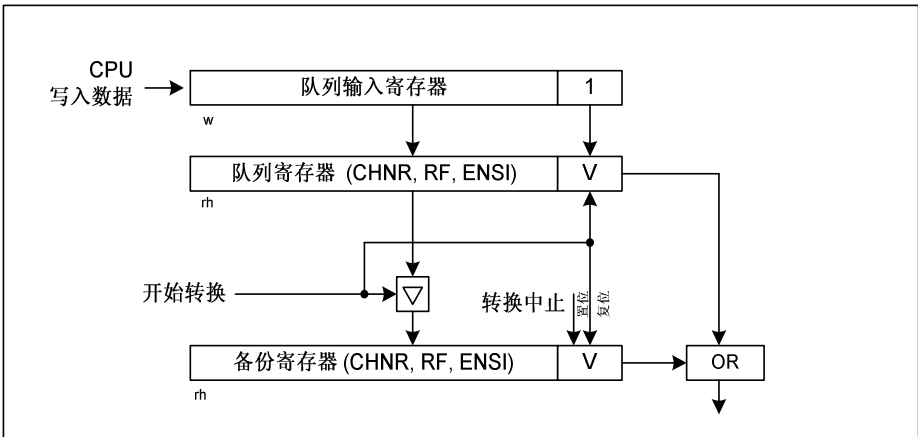


图 13-5 顺序请求源的基本结构

激活自动添加特性（ $RF = 1$ ）允许在成功执行转换开始后，在队列中自动重新插入该挂起请求。否则，挂起的请求执行完后被丢弃。但自动添加特性被使能时，软件不应队列输入寄存器写入数据。

顺序转换请求的入口寄存器为只写队列输入寄存器（QINR0）。如果队列阶段为空（ $V = 0$ ），写入数据将被保存在队列中（位 V 变 1），否则写操作被忽略。

顺序请求源寄存器的描述见章节 13.7.6。

13.4.4.2 请求源控制

如果请求源请求的转换和外部触发事件无关 ($EXTR = 0$)，请求有效位 $V = 1$ 将信号 $REQPND$ 和 $REQCHNRV$ 置 1 直接请求转换。在这种情况下，若 $V = 0$ 则无转换被请求。门控机制通过位 $ENGT$ 使用户能够使能/禁止转换请求。

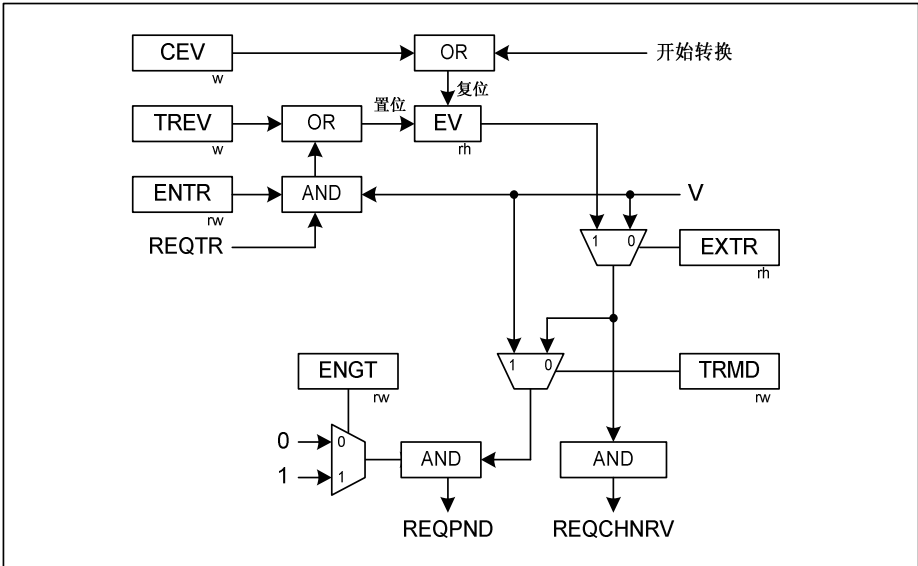


图 13-6 顺序请求源控制

如果转换和外部触发事件有关 ($EXTR = 1$)，信号 $REQTR$ 被考虑在内 ($ENTR = 1$) 或者软件控制写入 $TREV = 1$ 。这两个操作置位事件标志 EV 。事件标志 $EV = 1$ 表明已有外部事件发生，可请求转换（只有当转换请求有效 $V = 1$ 时 EV 才可被置位）。在这种情况下，信号 $REQCHNRV$ 来自位 EV 。

在有效请求位已置位的情况下，位 $TRMD$ （触发模式）可在参与仲裁之前等待事件被检测，从而保证了以最小延时响应外部事件。如果不需要该特性 ($TRMD = 0$)，用事件标志位 EV 产生信号 $REQPND$ 和 $REQCHNRV$ 。

13.4.5 并行请求源

13.4.5.1 概述

位于仲裁时隙 1 的并行请求源并行产生一路或多路通道 4 至通道 7 的转换请求。转换请求始终按照预先定义的次序（高路通道在低路通道之前）一个接一个（在不同的、分开的仲裁周期中）处理。

并行请求源由转换请求控制寄存器（CRCR1），转换请求挂起寄存器（CRPR1）和转换请求模式寄存器（CRMR1）组成。当选择的加载事件（LED）发生时，转换请求控制寄存器中的内容被复制（覆盖）到转换请求挂起寄存器中。事件的类型定义了请求源的行为以及触发方式。

如果转换挂起寄存器的内容不为 0，送至仲裁器的转换请求被激活。值为 1 的最高挂起位规定了转换通道的编号。要参与请求源仲裁信号，REQPND 和 REQCHNRV 必须为 1。

并行请求源寄存器的描述参见章节 13.7.7。

13.4.5.2 请求源控制

所有的转换请求挂起位逻辑或得到中间信号 PND，用来产生 REQPND 和 REQCHNRV。信号 PND 由位 ENGT 控制，从而使用户能够使能/禁止转换请求。见图 13-7。

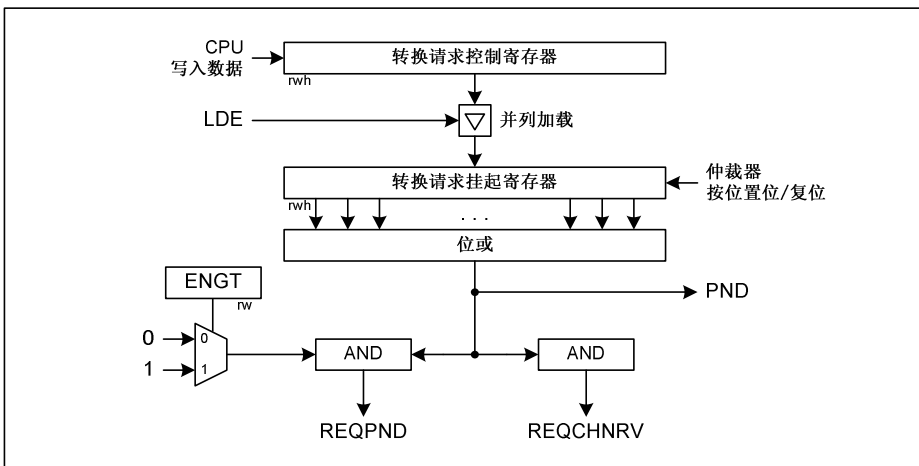


图 13-7 并行请求源控制

并列加载的加载事件可为：

- 输入线 REQTR 上的外部触发，见章节 13.4.5.3。
- 在转换请求控制寄存器的特定地址处进行写操作，见章节 13.4.5.4。
- 在请求源模式寄存器中写入 LDEV = 1，见章节 13.4.5.4。
- 请求源内部操作（转换完成且 PND = 0 时的自动扫描模式）。

转换请求控制/挂起寄存器中的每一位（位 x ， $x = 4 - 7$ ）对应一路模拟输入通道。每位所在的位置定义了通道编号。转换请求挂起寄存器中的位由仲裁器按位置位或复位：

- 当仲裁器指明某通道开始转换时，转换请求挂起寄存器中的相应位自动复位。
- 当仲裁器指明某通道转换被中止时，转换请求挂起寄存器中的相应位自动置位。

（由请求源请求的）转换完成且 PND = 0 时，（如果中断被使能）可产生请求源中断。这些规则仅适用于请求源已触发了转换的情况。

13.4.5.3 外部触发

并列请求源（以及顺序请求源）的转换请求可和外部触发事件同步。对于并列请求源，通过将重载事件耦合到请求触发输入 REQTR 上来实现。

13.4.5.4 软件控制

并列请求源的加载事件还可以由软件控制以两种方式产生：

- 可在两处不同的地址（CRCR1 和 CRPR1）对转换请求控制寄存器写入。访问 CRCR1 时，写操作仅改变该寄存器中的位状态。访问 CRPR1 时，在写操作之后一个时钟周期将产生加载事件。该自动加载事件只用一条 MOVE 指令即可启动转换。在这种情况下，即将转换的通道信息在 MOVE 指令中作为参数给出。
- 可由软件对位 LDEV 写入 1 来触发加载事件。在这种情况下，加载事件不包含任何即将转换的通道信息，而始终以转换请求控制寄存器中的通道信息作为目标转换通道。从而能够在另一地址写入转换请求控制寄存器而不触发加载事件。

13.4.5.5 自动扫描

并列请求源具有自动扫描功能。如果使能自动扫描模式，若并列请求源触发了转换，转换完成且 PND = 0 时产生加载事件。该自动重载特性不需外部触发或软件操作，可持续扫描通道 4 至通道 7，查询是否有挂起的转换请求。

13.4.6 待读模式

待读模式可用于所有的转换请求源，使 CPU 能够独立处理每次的转换结果而不丢失数据。若结果寄存器中的转换结果还未被 CPU 读出就已被新转换结果覆盖，会产生数据丢失。

待读模式下，如果目标结果寄存器已包含有效数据（由置位的有效标志示出），在某特定通道上请求源产生的转换请求将被禁止（不可能进行转换）。直到目标结果寄存器中的有效标志被清零（数据无效），才开始进行请求通道的转换。置位 WFR（参见章节 13.7.8）使能结果寄存器的待读模式。

13.4.7 转换结果的产生

13.4.7.1 概述

ADC 结果的产生由下面几部分组成：

- 极限检查单元，将转换结果和两个选择的边界值（BOUND0 和 BOUND1）相比较。根据极限检查结果可产生通道中断。
- 数据压缩滤波累加转换结果，将新的转换结果叠加到选中的结果寄存器所保存的转换结果上。
- 四个结果寄存器保存转换结果。软件可从结果寄存器中读出转换结果。每路输入通道可各自独立选择保存转换结果的结果寄存器。

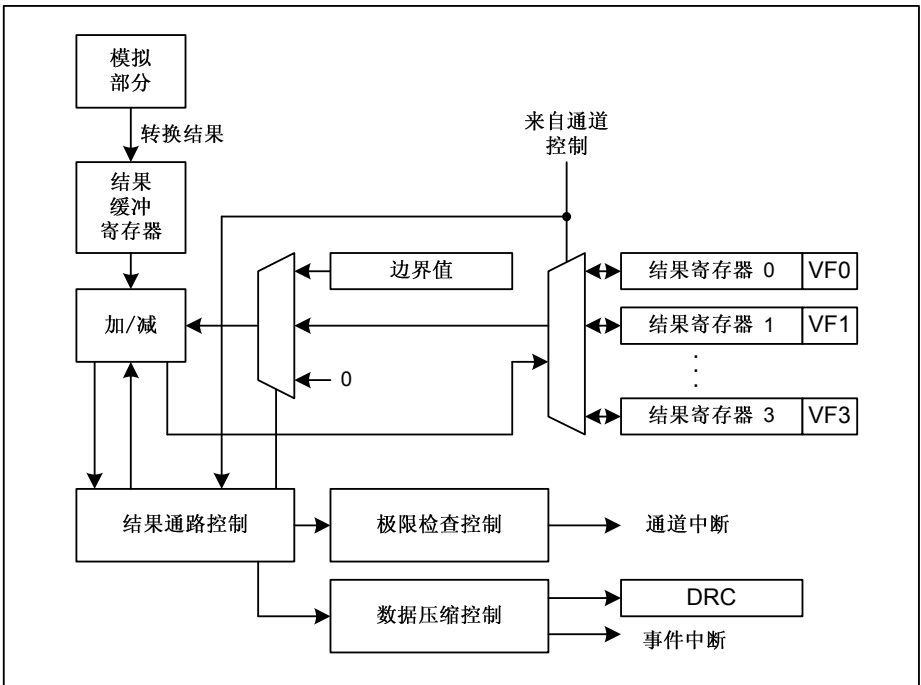


图 13-8 结果通路

结果寄存器描述参见章节 13.7.8。

13.4.7.2 极限检查

极限检查和数据压缩滤波基于同一个加/减结构。送给极限检查单元的转换结果先和 BOUND0 比较，再和 BOUND1 比较。极限检查根据结果标志（“小于”比较）可产生通道中断。当数据压缩滤波的有效结果保存到选择的结果寄存器中，极限检查可变为有效。

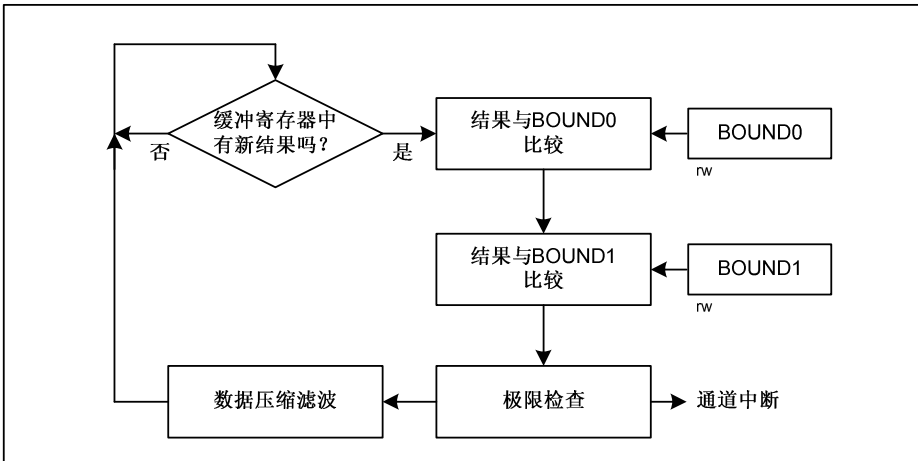


图 13-9 极限检查流程

13.4.7.3 数据压缩滤波

每个结果寄存器均可被控制使能或禁止数据压缩滤波。数据压缩模块可累加转换结果，用于抗混迭滤波或结果平均。以 A – B – B – A 次序进行通道转换，可实现伪并行双模拟口采样。结果寄存器 A 保存 A 通道两次转换结果的累加和；结果寄存器 B 保存 B 通道两次转换结果的累加和。

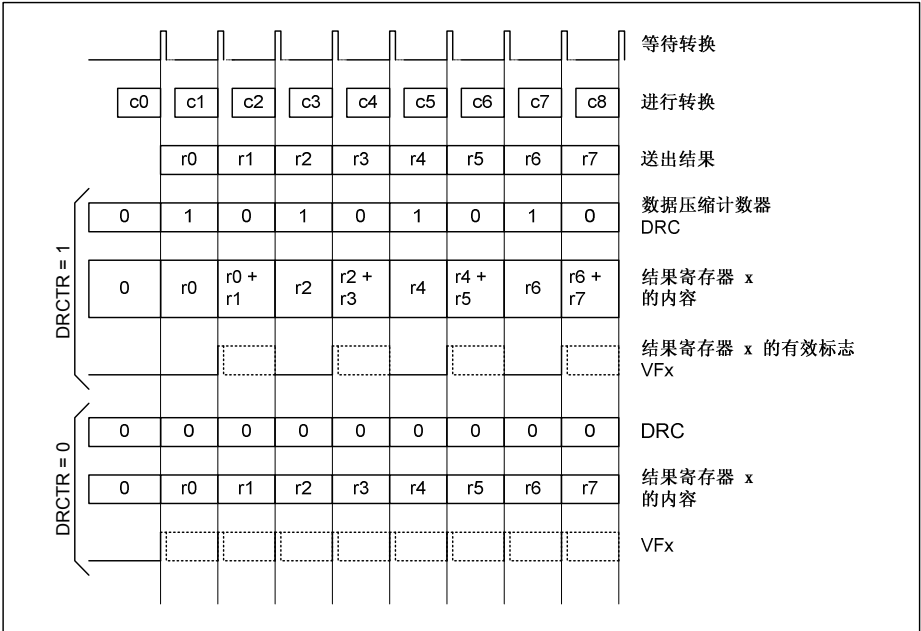


图 13-10 数据压缩流程

如果 DRC 为 0 且有新转换结果产生，重载值重新装入 DRC（由结果控制寄存器中的位 DRCTR 定义），将 DRC 为 0 的转换结果叠加到结果寄存器中的转换结果上（代替结果寄存器中的前次内容）。于是完整的结果被保存在选择的结果寄存器中。如果重载值为 0（数据压缩滤波被禁止），每次转换结果被保存在结果寄存器中，因此产生结果事件、置位结果寄存器的有效位（VF）。如果重载值为 1（数据压缩滤波被使能），将两次的转换结果累加，此时不产生结果事件、有效位不被置位。

如果 DRC 为 1 且有新转换结果产生，数据压缩滤波将新的转换结果叠加到结果寄存器已保存的转换结果上，DRC 减 1。叠加后完整的结果保存在选择的结果寄存器中。此时产生结果事件、置位结果寄存器的有效位。

使能或禁止数据压缩滤波，有可能使到达结果寄存器的通路的周期相同。另外，由于最多有两个转换结果相加（两个 10 位结果相加最大得到 11 位的结果），所以可避免结果寄存器溢出。

13.4.7.4 FIFO 型结果寄存器

四个结果寄存器可被独立设置为具有 2, 3 或 4 级 FIFO。从而允许放宽 CPU 的访问时间保存测量结果。

如果结果寄存器 x （和读取方式无关）的 FIFO 机制被使能（ $FEN = 1$ ），将执行下列操作（结果寄存器 $x + 1$ 的设置对下列操作无影响）：

如果有效标志 VFx 未被置位（结果寄存器中未包含有效数据），（结果寄存器 $x + 1$ 的） $VFx+1$ 被置位，结果寄存器 $x + 1$ 的内容会被转移到结果寄存器 x 中。接着 VFx 被置位、 $VFx+1$ 被复位。 VFx 的置位可产生事件中断。

若前一个寄存器（ $x - 1$ ）不能用作 FIFO 寄存器，新数据保存在结果寄存器 x 时产生结果中断 x 。

13.4.7.5 结果寄存器的读取方式

为了使应用领域更加广泛，结果寄存器 x （ $x = 0$ 至 3）的内容可从不同的地址上以不同的方式读取（见图 13-11）：

- 正常读取 RESR x L/H：
此方式读取 8 位或 10 位的转换结果。
- 读取 RESR x L/H：
此方式读取累加的 9 位或 11 位转换结果。

所有（有累加或无累加）的转换结果均保存在结果寄存器中，可从具有不同的数据排列及数据宽度的寄存器 RESR x L/H 或 RESR x L/H 中读取。

当使能数据压缩滤波（ $DRCTR = 1$ ），应从 RESR x L/H 中读取数据，因为该寄存器中保存着 9 位（R8:R0）或 11 位（R10:R0）的累加转换结果。读取 RESR x L/H 给出（无 MSB）累加结果。

当禁止数据压缩滤波（ $DRCTR = 0$ ），用户可从 RESR x L/H 或 RESR x L/H 中读取 8 位或 10 位的转换结果。尤其对于（无累加的）8 位转换，用一条指令即可从 RESR x H 中读出结果。因此，用户可根据应用要求选择不同的读取方式。

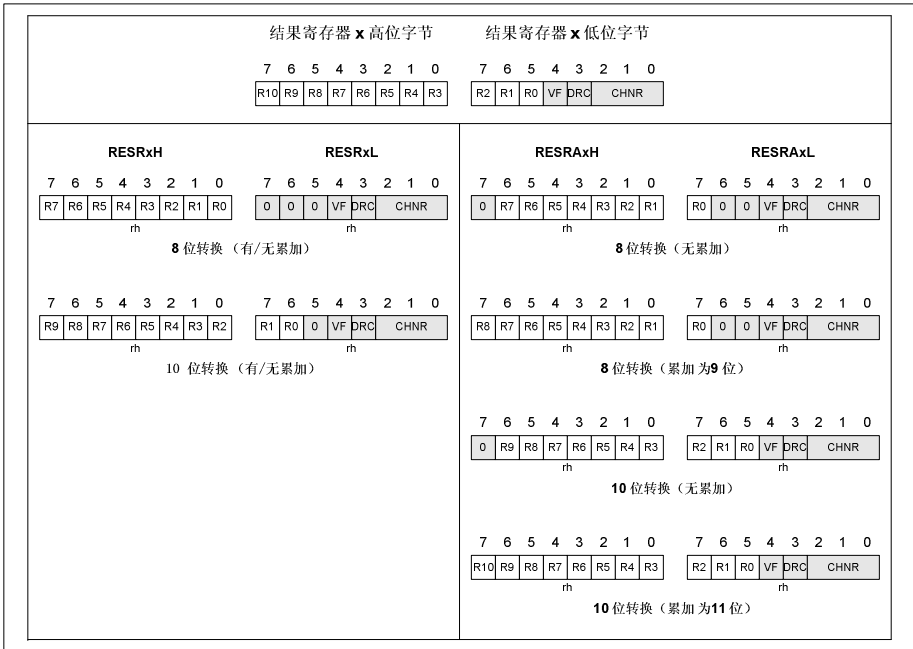


图 13-11 结果寄存器读取

13.4.8 中断

ADC 提供了两个可由不同的中断源激活的中断服务请求输出线 SR [1:0]。

ADC 的中断结构支持两种不同的中断源：

- 事件中断：由转换请求源事件（源中断）或结果寄存器事件（结果中断）激活。
- 通道中断：任何输入通道转换完成时激活该中断。根据极限检查控制位来使能通道中断。每路输入通道独立设定。

对于每条 SR 输出线，中断压缩单元都是所有中断脉冲的逻辑或组合。

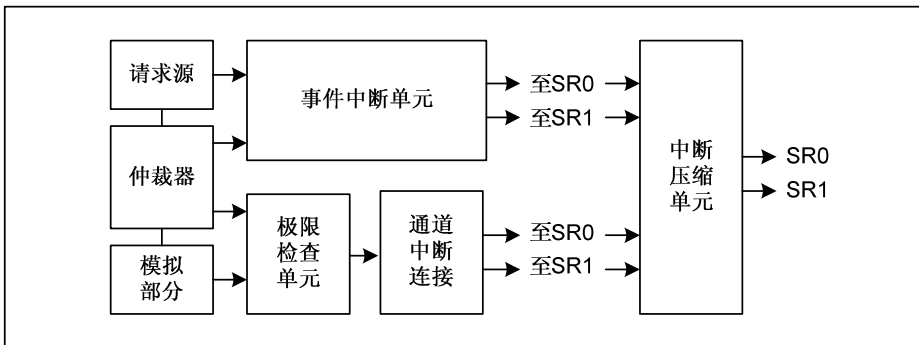


图 13-12 中断概览

中断寄存器的描述参见章节 13.7.9。

13.4.8.1 事件中断

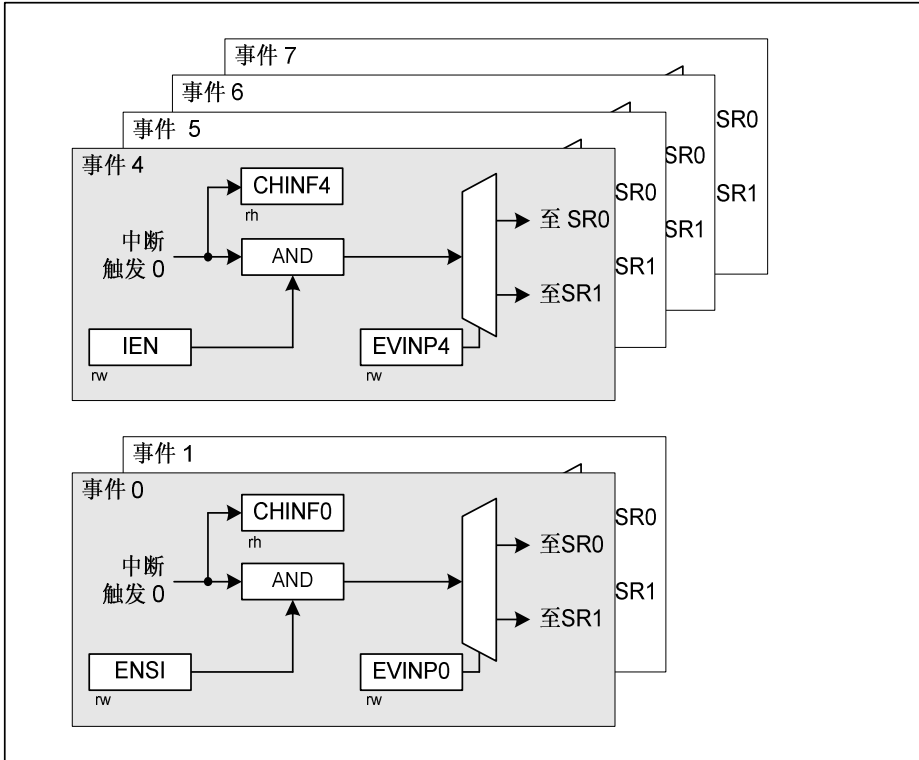


图 13-13 事件中断结构

请求源和结果寄存器可产生事件中断。事件中断使能位对应请求源寄存器的位 ENSI 和结果控制寄存器的位 IEN。每种事件的中断节点指针（EVINP）可选择目标中断服务请求输出线。

当被请求的通道转换完成时产生请求源事件：

- 事件 0：顺序请求源 0（仲裁时隙 0）的请求源事件
- 事件 1：并列请求源 1（仲裁时隙 1）的请求源事件

根据数据压缩控制（参见章节 13.4.7.3）产生结果寄存器事件：

- 事件 4：结果寄存器 0 的结果寄存器事件
- 事件 5：结果寄存器 1 的结果寄存器事件

- 事件 6: 结果寄存器 2 的结果寄存器事件
- 事件 7: 结果寄存器 3 的结果寄存器事件

13.4.8.2 通道中断

当转换完成并满足所选的极限检查条件时产生通道中断。因此一次只有一路通道中断可被激活。将每路通道的转换结果和两个选择的边界相比较，根据极限检查的结果触发中断。

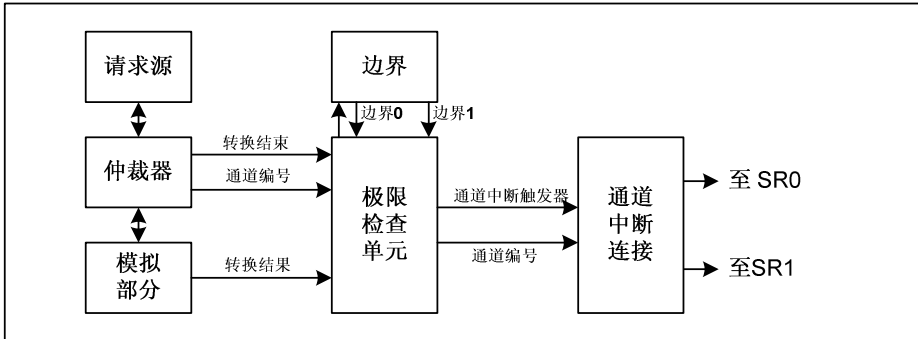


图 13-14 通道中断概览

极限检查单元将两个边界（BOUND0 和 BOUND1）和转换结果进行比较。转换结果空间被划分为 3 个区：

- 区 I：转换结果小于两个边界值。
- 区 II：转换结果在两个边界值之间。
- 区 III：转换结果大于两个边界值。

转换完成后，根据下列条件（由极限检查控制位 LCC 选择）可触发通道中断：

- LCC = 000：不触发中断，通道中断被禁止。
- LCC = 001：转换结果不在区 I 时产生通道中断。
- LCC = 010：转换结果不在区 II 时产生通道中断。
- LCC = 011：转换结果不在区 III 时产生通道中断。
- LCC = 100：始终产生通道中断（和边界无关）。
- LCC = 101：转换结果在区 I 时产生通道中断。
- LCC = 110：转换结果在区 II 时产生通道中断。
- LCC = 111：转换结果在区 III 时产生通道中断。

模数转换单元

通道中断节点指针 CHINPx (x = 0 至 7) 选择中断服务请求输出 (SR[1:0])，由通道中断触发激活。见图 13-15。

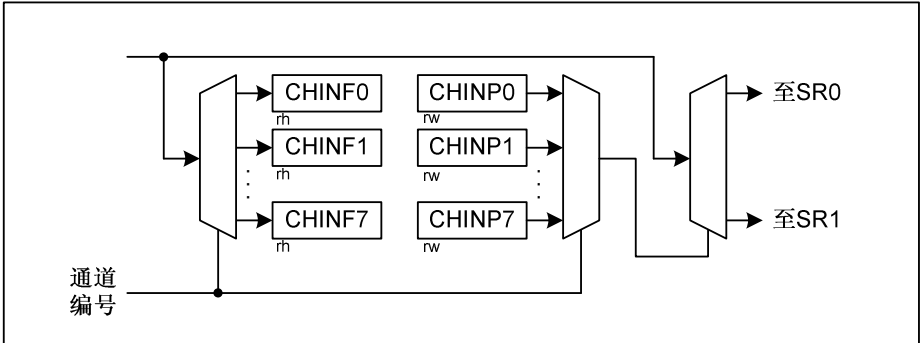


图 13-15 通道中断连接

13.4.9 外部触发输入

顺序请求源和并列请求源各自有一个请求触发输入 REQTRx (x = 0 -1)，通过该请求触发输入可启动转换请求。多路选择器根据位 ETRSELx 的值，从八路外部触发输入中 (ETRx0 至 ETRx7) 选择一路送至 REQTRx。和 ADC 同步的外部触发请求可不通过同步模块 (旁路)，这由位 SYNENx 的选择实现。

外部触发控制寄存器的描述参见章节 13.7.9。

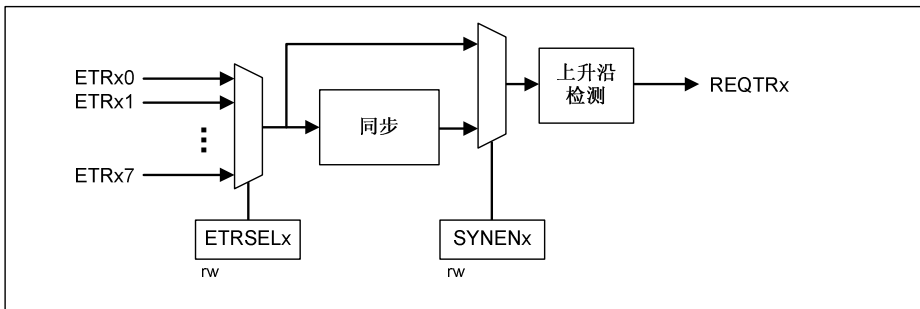


图 13-16 外部触发输入

ADC 的外部触发输入由 CCU6 模块产生的事件驱动。见表 13-2。

表 13-2 外部触发输入源

外部触发输入	CCU6 事件
ETRx0	定时器 13 周期 - 匹配
ETRx1	定时器 13 比较 - 匹配
ETRx2	定时器 12 周期 - 匹配
ETRx3	通道 0 定时器 12 比较 - 匹配
ETRx4	通道 1 定时器 12 比较 - 匹配
ETRx5	通道 2 定时器 12 比较 - 匹配
ETRx6	多通道模式的映射传输事件
ETRx7	多通道模式正确的霍尔事件

13.5 ADC 初始化序列

下列步骤旨在指导用户如何初始化 ADC 模块。一些步骤可根据应用需求的不同相应改变或省略：

1. 设置全局控制功能：
 - 选择转换宽度 (GLOBCTR.DW)
 - 选择模拟时钟 f_{ADCl} 分频因子 (GLOBCTR.CTC)

2. 设置仲裁控制功能：
 - 选择请求源 x
 - 优先级 (PRAR.PRIOx)
 - 转换启动模式 (PRAR.CSMx)
 - 使能仲裁时隙 x (PRAR.ASENx)
 - 选择仲裁模式 (PRAR.ARBm)

3. 设置通道控制信息：
 - 选择通道 x
 - 极限检查控制 (CHCTR_x.LCC)
 - 目标结果寄存器 (CHCTR_x.RESRSEL)
 - 选择所有通道的采样时间 (INPCR0.STC)

4. 设置结果控制信息：
 - 使能/禁止结果寄存器 x
 - 数据压缩 (RCR_x.DRCTR)
 - 事件中断 (RCR_x.IEN)
 - FIFO 功能 (RCR_x.FEN)
 - 待读模式 (RCR_x.WFR)
 - 读操作复位有效标志 (RCR_x.VFCTR)

5. 设置中断控制功能：
 - 选择通道 x 的中断节点指针 (CHINPR.CHINPx)
 - 选择事件 x 的中断节点指针 (EVINPR.EVINPx)

6. 设置极限检查边界：

- 选择所有通道的极限检查边界 (LCBR.BOUND0, LCBR.BOUND1)
7. 设置外部触发控制功能:
- 选择请求源 x 外部触发输入 (ETRCR.ETRSELx)
 - 使能/禁止请求源 x 外部触发输入同步功能 (ETRCR.SYENx)
8. 建立顺序请求源:
- 使能转换请求 (QMR0.ENG)
 - 使能/禁止外部触发器 (QMR0.ENTR)
 - 选择触发模式 (QMR0.TRMD)
9. 建立并列请求源:
- 使能转换请求 (CRMR1.ENG)
 - 使能/禁止外部触发 (CRMR1.ENTR)
 - 使能/禁止请求源中断 (CRMR1.ENS)
 - 使能/禁止自动扫描 (CRMR1.SCAN)
10. 开启模拟部分:
- 置位 GLOBCTR.ANON (等待 100 ns)
11. 启动顺序请求:
- (将信息如 REQCHNR, RF, ENSI 和 EXTR) 写入 QINR0
 - 用章节 13.4.4.2 描述的任何方式产生挂起的转换请求
12. 启动并列请求:
- 将要转换的通道编号写入 CRCR1 (无加载事件) 或写入 CRPR1 (自动加载事件)
 - 用章节 13.4.5.2 描述的任何方式产生加载事件 (若加载事件还未产生), 触发挂起的转换请求。
13. 等待 ADC 转换完成:
- 请求源中断指示请求源请求的转换已完成
 - 通道中断指示相应的通道转换 (并已执行极限检查) 已完成

- 结果中断指示相应结果寄存器中（有/无累加或 FIFO）的结果已准备好可以读取。

14. 读取 ADC 的结果

13.6 寄存器映射

ADC 的 SFR 在标准存储器区 (RMAP=0)，由七页构成。ADC_PAGE 寄存器位于地址 D1H 处，包含分页值和页控制信息。

ADC_PAGE

ADC 分页寄存器

复位值: 00H

	7	6	5	4	3	2	1	0
	OP		STNR		0	PAGE		
	w		w		r	rw		

符号	位序号	读 写 类 型	功能描述
PAGE	[2:0]	rw	分页位 写入时，该值表示新页的值 读出时，该值表示当前有效页的值
STNR	[5:4]	w	保存编号 该编号指明在哪个保存位域上执行 OP 定义的操作。 若 OP=10 _B ， PAGE 的内容在被新值覆盖之前保存在 ST _x 中 若 OP=11 _B ， PAGE 的内容被 ST _x 覆盖。写入 PAGE 的值不予理睬 00 选择 ST0 01 选择 ST1 10 选择 ST2 11 选择 ST3
OP	[7:6]	w	操作 0X 手动保存页模式，STNR 的值被忽略， PAGE 被直接写入 10 带有自动页保存的新页设置。当前写入

符号	位序号	读 写 类 型	功能描述
			<p>PAGE 中的内容被保存的同时，上次写入 PAGE 的内容被保存在 STNR 规定的位域 STx 中</p> <p>11 自动恢复页。对写入 PAGE 的内容不予理睬，PAGE 由 STNR 规定的位域 STx 中的内容覆盖</p>
0	3	r	<p>保留</p> <p>读操作返回 0；应写入 0</p>

下表中描述的所有 ADC 寄存器名称，在本手册其他章节中引用时需加上模块名前缀“ADC_”，例如 ADC_GLOBCTR。

ADC 的 SFR 地址列于表 13-3 和表 13-4 中。

表 13-3 页 0 - 2 的 SFR 地址列表

地址	页 0	页 1	页 2
CA _H	GLOBCTR	CHCTR0	RESR0L
CB _H	GLOBSTR	CHCTR1	RESR0H
CC _H	PRAR	CHCTR2	RESR1L
CD _H	LCBR	CHCTR3	RESR1H
CE _H	INPCR0	CHCTR4	RESR2L
CF _H	ETRCR	CHCTR5	RESR2H
D2 _H	-	CHCTR6	RESR3L
D3 _H	-	CHCTR7	RESR3H

表 13-4 页 3 - 6 的 SFR 地址列表

地址	页 3	页 4	页 5	页 6
CA _H	RESRA0L	RCR0	CHINFR	CRCR1
CB _H	RESRA0H	RCR1	CHINCR	CRPR1
CC _H	RESRA1L	RCR2	CHINSR	CRMR1
CD _H	RESRA1H	RCR3	CHINPR	QMR0
CE _H	RESRA2L	VFCR	EVINFR	QSR0
CF _H	RESRA2H	-	EVINCR	Q0R0
D2 _H	RESRA3L	-	EVINSR	QBUR0/QINR0
D3 _H	RESRA3H	-	EVINPR	-

13.7 寄存器描述

13.7.1 基本功能寄存器

寄存器 GLOBCTR 中的控制位控制模拟部分和转换时间。

GLOBCTR

全局控制寄存器

复位值: 30_H

7	6	5	4	3	2	1	0
ANON	DW	CTC		0			
rw	rw	rw		r			

符号	位序号	读写类型	功能描述
CTC	[5:4]	rw	<p>转换时间控制</p> <p>该位定义了内部模拟时钟 f_{ADCI} 的分频因子。该时钟提供了模拟转换和采样时间计算的内部时间基准。</p> <p>00 $f_{ADCI} = 1/2 \times f_{ADCA}$ 01 $f_{ADCI} = 1/3 \times f_{ADCA}$ 10 $f_{ADCI} = 1/4 \times f_{ADCA}$ 11 $f_{ADCI} = 1/32 \times f_{ADCA}$ (缺省值)</p>
DW	6	rw	<p>数据宽度</p> <p>该位定义了转换精度。</p> <p>0 10 位的转换结果 (缺省值) 1 8 位的转换结果</p>
ANON	7	rw	<p>模拟部分开启控制</p> <p>该位使能 ADC 的模拟部分并定义其工作模式。</p> <p>0 模拟部分关闭, 不能进行转换。为了使功耗最低, 内部模拟电路处于掉电模式, 停止产生 f_{ADCI}。 1 模拟部分开启, 可进行转换。模拟部分的自动掉电功能被禁止。</p>

模数转换单元

符号	位序号	读写类型	功能描述
0	[3:0]	r	保留 读操作返回 0；应写入 0

寄存器 GLOBSTR 中包含指示当前转换状态的指示位。

GLOBSTR
全局状态寄存器
复位值: 00_H

7	6	5	4	3	2	1	0
0	CHNR			0	SAMPL E	BUSY	
r	rh			r	rh	rh	

符号	位序号	读写类型	功能描述
BUSY	0	rh	模拟部分忙碌指示位 该位指示当前正在进行转换。 0 模拟部分空闲 1 转换正在进行
SAMPLE	1	rh	采样阶段指示位 该位指示当前正在对模拟输入信号采样。 0 模拟部分未处于采样阶段 1 模拟部分正处于采样阶段
CHNR	[5:3]	rh	通道编号 该位指示正在转换哪路模拟输入通道。新转换开始时更新该信息。
0	2, [7:6]	r	保留 读操作返回 0；应写入 0。

13.7.2 优先级和仲裁寄存器

寄存器 PRAR 中的控制位定义转换请求源的优先级和转换启动模式，以及在仲裁时隙使能/禁止转换请求处理。

PRAR

优先级和仲裁寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
ASEN1	ASEN0	0	ARB	CSM1	PRIO1	CSM0	PRIO0
rw	rw	r	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
PRIO0	0	rw	转换请求源 0 的优先级 该位定义了顺序请求源 0 的优先级。 0 低优先级 1 高优先级
CSM0	1	rw	转换请求源 0 的转换启动模式 该位定义了顺序请求源 0 的转换启动模式。 0 选择等待开始模式 1 选择取消-插入-重复模式
PRIO1	2	rw	转换请求源 1 的优先级 该位定义了并列请求源 1 的优先级。 0 低优先级 1 高优先级
CSM1	3	rw	转换请求源 1 的转换启动模式 该位定义了并列请求源 1 的转换启动模式。 0 选择等待开始模式 1 选择取消-插入-重复模式
ARB	4	rw	仲裁模式 该位定义了选择哪种仲裁模式 0 持续仲裁 (缺省值)

模数转换单元

符号	位序号	读写类型	功能描述
			1 由挂起的转换请求启动仲裁
ASENx (x = 0 - 1)	[7:6]	rw	<p>仲裁时隙 x 使能位</p> <p>这两位分别使能仲裁周期中的的一个仲裁时隙。ASEN0 使能仲裁时隙 0，ASEN1 使能仲裁时隙 1。</p> <p>如果仲裁时隙被禁止，和这个时隙相连的转换请求源产生的挂起的转换请求不参与仲裁。</p> <p>0 禁止相应的仲裁时隙 1 使能相应的仲裁时隙</p>
0	5	r	<p>保留</p> <p>读操作返回 0；应写入 0。</p>

13.7.3 外部触发控制寄存器

寄存器 ETRCR 中的控制位选择外部触发输入信号源以及使能与外部触发输入同步。

ETRCR

外部触发控制寄存器

复位值: 00_H

	7	6	5	4	3	2	1	0
SYNEN1	SYNEN0	ETRSEL1			ETRSEL0			
rw	rw	rw			rw			

符号	位序号	读写类型	功能描述
ETRSELx (x = 0 - 1)	[2:0], [5:3]	rw	转换请求源 x 的外部触发选择 该位定义了选择哪种外部触发输入信号。 000 选择触发输入 ETRx0 001 选择触发输入 ETRx1 111 选择触发输入 ETRx7
SYNENx (x = 0 - 1)	6, 7	rw	同步使能位 0 同步阶段不在请求外部触发输入 REQTRx 的路径上 1 同步阶段位于请求外部触发输入 REQTRx 的路径上

13.7.4 通道控制寄存器

通道控制寄存器 CHCTR_x 中的控制位选择目标结果寄存器以及控制极限检查；寄存器 CHCTR_x 定义了输入通道 x 的设置。

CHCTR_x (x = 0 - 7)

通道控制寄存器 x

复位值: 00_H

7	6	5	4	3	2	1	0
0	LCC			0		RESRSEL	
r	rw			r		rw	

符号	位序号	读写类型	功能描述
RESRSEL	[1:0]	rw	结果寄存器选择 该位定义了选择哪个结果寄存器保存该通道转换的结果。 00 选择结果寄存器 0 01 选择结果寄存器 1 10 选择结果寄存器 2 11 选择结果寄存器 3
LCC	[6:4]	rw	极限检查控制 该位定义了极限检查的操作 参见章节 13.4.8.2
0	[3:2], 7	r	保留 读操作返回 0；应写入 0。

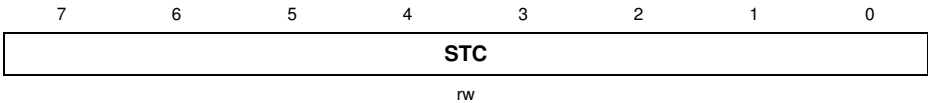
13.7.5 输入综合控制寄存器

寄存器 INPCR0 中的控制位控制输入通道采样时间。

INPCR0

输入综合控制寄存器 0

复位值: 00_H



符号	位序号	读写类型	功能描述
STC	[7:0]	rw	采样时间控制 该位域定义了以时钟周期 f_{ADCI} 为单位的附加采样时间。可将两个模拟时钟周期的采样时间扩展为编程设定的采样值。

13.7.6 顺序请求源寄存器

这些寄存器包含顺序请求源 0 的控制和状态位。

寄存器 QMR0 中的控制位将顺序请求源设定为期望模式。

QMR0

队列模式寄存器

复位值: 00H

7	6	5	4	3	2	1	0
CEV	TREV	FLUSH	CLR V	TRMD	ENTR	0	ENGT
w	w	w	w	rw	rw	r	rw

符号	位序号	读写类型	功能描述
ENGT	0	rw	门控使能 该位使能请求源的门控功能。 0 门控线始终为 0。请求源关闭。 1 门控线始终为 1。请求源开启。
ENTR	2	rw	外部触发使能 该位使能外部触发功能。如果该位使能，寄存器 QOR0 或 QBUR0 中至少有一个 V 位被置位时，检测到外部触发输入 REQTR 的上升沿时置位 EV。 0 禁止外部事件 1 使能外部事件
TRMD	3	rw	触发模式选择 该位定义了选择哪种触发模式。触发模式 0，输出线 REQPND 和 REQCHNRV 同时有效；触发模式 1，信号 REQPND 比 REQCHNRV 提前有效。 0 选择触发模式 0 1 选择触发模式 1
CLR V	4	w	V 位清零 0 无操作 1 寄存器 QOR0 或 QBUR0 中的 V 位被复

模数转换单元

符号	位序号	读写类型	功能描述
			位。 若 QBUR0.V = 1, 则 QBUR0.V 复位 若 QBUR0.V = 0, 则 QOR0.V 复位
FLUSH	5	w	队列全部清零控制 0 无操作 1 对列寄存器中所有 V 位和 EV 位全部被复位。队列中无有效请求输入
TREV	6	w	触发事件控制 0 无操作 1 软件产生触发事件。若请求源等到了触发事件, 启动转换请求。
CEV	7	w	事件清零控制 0 无操作 1 位 EV 被清零
0	1	r	保留 读操作返回 0; 应写入 0。

寄存器 QSR0 中的状态位指示顺序请求源的状态。

QSR0

队列状态寄存器

复位值: 20_H

7	6	5	4	3	2	1	0
Rsv	0	EMPTY	EV	0			
r	rh		rh	r			

符号	位序号	读写类型	功能描述
EV	4	rh	<p>检测事件指示</p> <p>该位指示当 V = 1 时检测到有事件发生。一旦置位，被请求的转换开始进行时该位自动复位。</p> <p>0 未检测到事件发生 1 检测到事件已发生</p>
EMPTY	5	rh	<p>队列已空指示</p> <p>该位指示队列（Q0R0）中是否包含有效的请求输入。如果对列已满（EMPTY = 0），新的请求输入不予理睬。</p> <p>0 对列已满（1 个有效输入） 1 对列已空</p>
Rsv	7	r	<p>保留</p> <p>读操作返回 1；应写入 0。</p>
0	[3:0], 6	r	<p>保留</p> <p>读操作返回 0；应写入 0。</p>

模数转换单元

寄存器 Q0R0 中的监控位监控顺序请求源状的当前状态。

Q0R0

队列 0 寄存器 0

复位值: 00_H

7	6	5	4	3	2	1	0
EXTR	ENSI	RF	V	0	REQCHNR		
rh	rh	rh	rh	r	rh		

符号	位序号	读写类型	功能描述
REQCHNR	[2:0]	rh	请求通道编号 该位给出将要被请求或正在被请求的通道编号。
V	4	rh	请求通道有效指示 该位指示 REQCHNR, RF, ENSI 和 EXTR 是否有效。当有效输入写入队列输入寄存器 QINR0 时置位 V。 0 数据无效 1 数据有效
RF	5	rh	重新添加指示 该位指示挂起的请求执行完后（开始转换）是被丢弃还是被自动添加到请求队列的顶端 0 转换开始之后丢弃请求 1 转换开始之后将请求重新添加到队列中
ENSI	6	rh	请求源中断使能 该位指示转换结束后是否产生请求源中断。若转换结束并且 ENSI = 1, 中断触发有效 0 禁止请求源中断 1 使能请求源中断
EXTR	7	rh	外部触发 该位定义了转换请求源是否和外部触发事件有关 事件标志位（位 EV）指示外部事件是否已

模数转换单元

符号	位序号	读写类型	功能描述
			发生并可请求转换 0 位 EV 未用作启动转换请求 1 位 EV 用作启动转换请求
0	3	r	保留 读操作返回 0；应写入 0。

寄存器 QBUR0 和 QINR0 共用一个寄存器地址。该寄存器地址上的读操作将对应取出 QBUR0 寄存器的“rh”位；相同寄存器地址上的写操作将对应 QINR0 寄存器的“w”位。

寄存器 QBUR0 中的监控位监控被中断顺序请求源的状态。

QBUR0

队列备份寄存器 0

复位值: 00_H

7	6	5	4	3	2	1	0
EXTR	ENSI	RF	V	0	REQCHNR		
rh	rh	rh	rh	r	rh		

符号	位序号	读写类型	功能描述
REQCHNR	[2:0]	rh	请求通道编号 当开始进行 Q0R0 请求的转换时该位由位域 Q0R0.REQCHNR 更新
V	4	rh	请求通道有效指示 该位指示 REQCHNR, RF, ENSI 和 EXTR 是否有效。当正在进行的转换被退出时置位 V。转换开始后该位复位。 0 备份寄存器中不包含有效数据，因为该数据所描述的转换未退出 1 数据有效。在处理 Q0R0 请求的转换之前，先请求处理被退出的转换
RF	5	rh	重新添加指示 当开始进行 Q0R0 请求的转换时，该位由位

模数转换单元

符号	位序号	读写类型	功能描述
			Q0R0.RF 更新
ENSI	6	rh	请求源中断使能 当开始进行 Q0R0 请求的转换时，该位由位 Q0R0.ENSI 更新
EXTR	7	rh	外部触发 当开始进行 Q0R0 请求的转换时，该位由位 Q0R0.EXTR 更新
0	3	r	保留 读操作返回 0；应写入 0。

寄存器 QINR0 是顺序请求源的入口寄存器。

QINR0

队列输入寄存器 0

复位值: 00H

7	6	5	4	3	2	1	0
EXTR	ENSI	RF	0	REQCHNR			
w	w	w	r	w			

符号	位序号	读写类型	功能描述
REQCHNR	[2:0]	rh	请求通道编号 该位域定义了被请求的通道编号
RF	5	rh	重新添加指示 该位定义了重新添加功能
ENSI	6	rh	请求源中断使能 该位定义了请求源中断功能
EXTR	7	rh	外部触发 该位定义了外部触发功能
0	[4:3]	r	保留

模数转换单元

符号	位序号	读写类型	功能描述
			读操作返回 0；应写入 0。

13.7.7 并行请求源寄存器

这些寄存器中存放并行请求源 1 的控制和状态位。

寄存器 CRCR1 包含的控制位在加载事件发生时被复制到挂起寄存器 (CRPR1) 中。该寄存器可从两个地址处访问 (一个读地址; 两个写地址)。第一个读写操作的地址为 CRCR1 的地址; 第二个写操作的地址对应 CRPR1 的地址。对 CRPR1 的写操作将使数据写入 CRCR1 并在一个时钟周期后自动产生加载事件。

CRCR1

转换请求控制寄存器 1

复位值: 00H

7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	0			
rwh	rwh	rwh	rwh	r			

符号	位序号	读写类型	功能描述
CHx (x = 4 - 7)	x	rwh	<p>通道位 x</p> <p>每一位对应一路模拟通道, 由寄存器中位域的位置定义通道编号 x。加载事件发生时, 转换请求挂起寄存器中的相应位 x 被该位覆盖。</p> <p>0 并行请求源未请求该模拟通道 x 进行转换</p> <p>1 并行请求源将请求该模拟通道 x 进行转换</p>
0	[3:0]	r	<p>保留</p> <p>读操作返回 0; 应写入 0。</p>

模数转换单元

寄存器 CRPR1 中的控制位请求进行相应模拟通道转换。该寄存器只读；对该地址的写操作将使数据写入 CRCR1 并在一个时钟周期后自动产生加载事件。

CRPR1

转换请求挂起寄存器 1

复位值: 00_H

7	6	5	4	3	2	1	0
CHP7	CHP6	CHP5	CHP4	0			
rwh	rwh	rwh	rwh	r			

符号	位序号	读写类型	功能描述
CHPx (x = 4 - 7)	x	rwh	<p>通道挂起位 x</p> <p>写操作： 该地址写操作的目标寄存器为 CRCR1</p> <p>读操作： 每一位对应一路模拟通道，由寄存器中位的位置定义通道编号 x。 (转换开始时) 仲裁器自动复位或(转换退出时) 再次置位相应的模拟通道。</p> <p>0 并列请求源未请求该模拟通道 x 进行转换</p> <p>1 并列请求源请求该模拟通道 x 进行转换</p>
0	[3:0]	r	<p>保留</p> <p>读操作返回 0；应写入 0。</p>

注：可从该寄存器读取的位通常标示为“rh”，不可由写操作直接修改。写操作将修改 CRCR1 寄存器中的位（故这些位标示为“rwh”），并一个时钟周期后产生加载事件。

寄存器 CRMR1 中的控制位将并列请求源设定为期望模式。

CRMR1
转换请求模式寄存器 1
复位值: 00_H

7	6	5	4	3	2	1	0
Rsv	LDEV	CLRPND	SCAN	ENSI	ENTR	0	ENGT
r	w	w	w	rw	rw	r	rw

符号	位序号	读写类型	功能描述
ENGT	0	rw	门控使能 该位使能请求源的门控功能。 0 门控线始终为 0。请求源关闭。 1 门控线始终为 1。请求源开启。
ENTR	2	rw	外部触发使能 该位使能外部触发功能。如果该位被使能，检测到外部触发输入 REQTR 的上升沿时产生加载事件 0 禁止外部事件 1 使能外部事件
ENSI	3	rw	请求源中断使能 该位使能请求源中断。当该请求源的最后一个挂起请求完成时（且 PND = 0）产生中断 0 禁止请求源中断 1 使能请求源中断
SCAN	4	rw	自动扫描使能 该位使能自动扫描功能。如果该位被使能，转换（被该请求源请求）完成且 PND = 0 时自动产生加载事件 0 禁止自动扫描功能 1 使能自动扫描功能
CLRPND	5	w	挂起位清零

模数转换单元

符号	位序号	读写类型	功能描述
			0 无操作 1 复位寄存器 CRPR1 中的位
LDEV	6	w	产生加载事件 0 无操作 1 产生加载事件
Rsv	7	r	保留 读操作返回 1；应写入 0。
0	1	r	保留 读操作返回 0；应写入 0。

13.7.8 结果寄存器

结果寄存器给出转换结果，也可给出导致结果寄存器最新更新的通道编号。结果寄存器可从不同的地址上以不同的方式读取。以下位域可根据所选择的读取地址，从结果寄存器中读取。转换结果的排列及宽度的详细描述，参见章节 13.4.7.5。

符号	位序号	读写类型	功能描述
RESULT	RESR _x L[7:6], RESR _x H 或 RESR _A xL[7:5], RESR _A xH	rh	转换结果 该位域包含了转换结果或数据压缩滤波的结果。
CHNR	[2:0]	rh	通道编号 该位域包含了最新更新结果寄存器的通道编号
DRC	3	rh	数据压缩计数器 该位指示还有多少转换结果需要累加以产生最终的数据压缩结果 0 结果寄存器中的值为最终转换结果。当该位域置 0 时有效标志自动置位 1 必须再累加一次转换结果以获得结果寄存器的最终转换值。当该位域置 1 时有效标志自动复位
VF	4	rh	结果寄存器 x 的有效标志 该位指示结果寄存器 x 中的内容是否有效 0 结果寄存器 x 中不包含有效数据 1 结果寄存器 x 中包含有效数据

正常读取 RESRx

该读取方式给出 8 位或 10 位的转换结果以及 3 位的通道编号。假如位 RCRx.VFCTR 被置位，读指令访问寄存器的高位字节时相应的有效标志被清零。

RESRxL (x = 0 - 3)

结果寄存器 x，低位字节

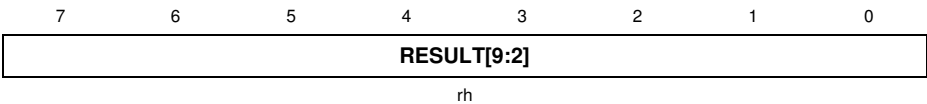
复位值: 00_H



RESRxH (x = 0 - 3)

结果寄存器 x，高位字节

复位值: 00_H



累加读取 RESRAx

该读取方式给出累加的 9 位或 11 位的转换结果以及 3 位的通道编号。假如位 RCRx.VFCTR 被置位，读指令访问寄存器的高位字节时相应的有效标志被清零。

RESRAxL (x = 0 - 3)

结果寄存器 x，累加读取低位字节

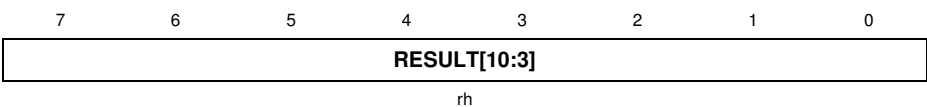
复位值: 00_H



RESRAxH (x = 0 - 3)

结果寄存器 x，累加读取高位字节

复位值: 00_H



模数转换单元

对寄存器 VFCR 中某位写 1 时，将对寄存器 RESR_x/RESR_{Ax} 中的相应有效标志清零。如果硬件事件触发置位某位 VF_x，同时 VFC_x = 1，VF_x 被清零（软件占优）。

VFCR
有效标志清零寄存器
复位值: 00H

7	6	5	4	3	2	1	0
0				VFC3	VFC2	VFC1	VFC0
r				w	w	w	w

符号	位序号	读写类型	功能描述
VFC_x (x = 0-3)	x	w	结果寄存器 x 有效标志清零 0 无操作 1 位 VF _x 被复位
0	[7:4]	r	保留 读操作返回 0；应写入 0。

结果控制寄存器 RCR_x 中包含控制结果寄存器行为的控制位以及状态监控位。

RCR_x (x = 0 - 3)
结果控制寄存器 x
复位值: 00H

7	6	5	4	3	2	1	0
VFCTR	WFR	FEN	IEN	0		DRCTR	
rw	rw	rw	rw	r		rw	

符号	位序号	读写类型	功能描述
DRCTR	0	rw	数据压缩控制 该位定义了有多少转换结果累加用于数据压缩。它定义了位 DRC 的重载值。 0 禁止数据压缩滤波。DRC 的重载值为 0，故每个转换结果即为累加结果 1 使能数据压缩滤波。DRC 的重载值为

模数转换单元

符号	位序号	读写类型	功能描述
			1, 故 2 个转换结果相累加
IEN	4	rw	<p>中断使能</p> <p>该位使能与结果寄存器 x 相关的事件中断。DRC 被置 0 时 (减 1 或重载后) 产生事件中断</p> <p>0 禁止事件中断 1 使能事件中断</p>
FEN	5	rw	<p>FIFO 使能</p> <p>该位使能结果寄存器 x 的 FIFO 功能</p> <p>0 禁止 FIFO 功能 1 使能 FIFO 功能</p>
WFR	6	rw	<p>待读模式</p> <p>该位使能结果寄存器 x 的待读模式</p> <p>0 禁止待读模式 1 使能待读模式</p>
VFCTR	7	rw	<p>有效标志控制</p> <p>该位使能结果寄存器 x 有效标志的复位功能 (通过读取高位字节实现)</p> <p>0 读取 RESRxH/RESRAxH 时 VF 不改变 (缺省值) 1 读取 RESRxH/RESRAxH 时 VF 复位</p>
0	[3:1]	r	<p>保留</p> <p>读操作返回 0; 应写入 0。</p>

13.7.9 中断寄存器

寄存器 CHINFR 监控被激活的通道中断标志。

CHINFR

通道中断标志寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
CHINF7	CHINF6	CHINF5	CHINF4	CHINF3	CHINF2	CHINF1	CHINF0
rh	rh	rh	rh	rh	rh	rh	rh

符号	位序号	读写类型	功能描述
CHINF_x (x = 0- 7)	x	rh	通道 x 的中断标志 该位监控通道 x 的中断状态 0 通道 x 未产生通道中断 1 通道 x 已产生通道中断

对寄存器 CHINCR 中某位写 1 时，将对寄存器 CHINFR 中的相应通道中断标志清零。如果硬件事件触发置位某位 CHINF_x，同时 CHINC_x = 1，CHINF_x 被清零（软件占优）。

CHINCR

通道中断清零寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
CHINC7	CHINC6	CHINC5	CHINC4	CHINC3	CHINC2	CHINC1	CHINC0
w	w	w	w	w	w	w	w

符号	位序号	读写类型	功能描述
CHINC_x (x = 0- 7)	x	w	通道 x 中断标志清零 0 无操作 1 位 CHINFR.x 被复位

模数转换单元

对寄存器 CHINSR 中某位写 1 时，置位寄存器 CHINFR 中的相应通道中断标志，并产生中断脉冲。

CHINSR
通道中断置位寄存器
复位值: 00_H

7	6	5	4	3	2	1	0
CHINS7	CHINS6	CHINS5	CHINS4	CHINS3	CHINS2	CHINS1	CHINS0
w	w	w	w	w	w	w	w

符号	位序号	读写类型	功能描述
CHINSx (x = 0-7)	x	w	通道 x 中断标志置位 0 无操作 1 位 CHINFR.x 被置位并产生中断脉冲

寄存器 CHINPR 定义了中断服务请求输出线，SR_x (x = 0 或 1)，产生通道中断时该输出被激活。

CHINPR
通道中断节点指针寄存器
复位值: 00_H

7	6	5	4	3	2	1	0
CHINP7	CHINP6	CHINP5	CHINP4	CHINP3	CHINP2	CHINP1	CHINP0
rw	rw	rw	rw	rw	rw	rw	rw

符号	位序号	读写类型	功能描述
CHINPx (x = 0-7)	x	rw	通道 x 中断节点指针 该位定义了通道 x 产生中断时哪条 SR 有效 0 线 SR0 有效 1 线 SR1 有效

寄存器 EVINFR 监控被激活的事件中断标志。

EVINFR

事件中断标志寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
EVINF7	EVINF6	EVINF5	EVINF4	0		EVINF1	EVINF0
rh	rh	rh	rh	r		rh	rh

符号	位序号	读写类型	功能描述
EVINF_x (x = 0- 1, 4 - 7)	[1:0], [7:4]	rh	事件 x 的中断标志 该位监控事件 x 的中断状态 0 事件 x 未产生事件中断 1 事件 x 已产生事件中断
0	[3:2]	r	保留 读操作返回 0; 应写入 0。

对寄存器 EVINCR 中某位写 1 时, 将对寄存器 EVINFR 中的相应事件中断标志清零。如果硬件事件触发置位某位 EVINF_x, 同时 EVINC_x = 1, EVINF_x 被清零 (软件占优)。

EVINCR

事件中断标志清零寄存器

复位值: 00_H

7	6	5	4	3	2	1	0
EVINC7	EVINC6	EVINC5	EVINC4	0		EVINC1	EVINC0
w	w	w	w	r		w	w

符号	位序号	读写类型	功能描述
EVINC_x (x = 0- 1, 4 - 7)	[1:0], [7:4]	w	事件 x 中断标志清零 0 无操作 1 位 EVINFR.x 被复位

模数转换单元

符号	位序号	读写类型	功能描述
0	[3:2]	r	保留 读操作返回 0；应写入 0。

对寄存器 EVINSR 中某位写 1 时，置位寄存器 EVINFR 中的相应通道中断标志，并产生中断脉冲（如果中断被使能）。

EVINSR
事件中断标志置位寄存器
复位值: 00_H

7	6	5	4	3	2	1	0
EVINS7	EVINS6	EVINS5	EVINS4	0	EVINS1	EVINS0	
w	w	w	w	r	w	w	

符号	位序号	读写类型	功能描述
EVINSx (x = 0- 1, 4 - 7)	[1:0], [7:4]	w	事件 x 中断标志置位 0 无操作 1 位 EVINFR.x 被置位
0	[3:2]	r	保留 读操作返回 0；应写入 0。

寄存器 EVINPR 定义了中断服务请求输出线，SR_x（x = 0 或 1），产生事件中断时该输出被激活。

EVINPR
事件中断节点指针寄存器
复位值: 00_H

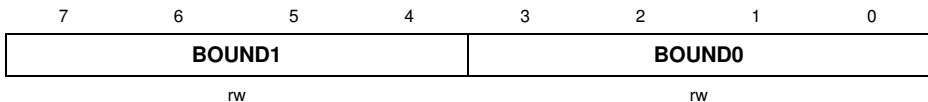
7	6	5	4	3	2	1	0
EVINP7	EVINP6	EVINP5	EVINP4	0	EVINP1	EVINP0	
rw	rw	rw	rw	r	rw	rw	

符号	位序号	读写类型	功能描述
----	-----	------	------

模数转换单元

符号	位序号	读写类型	功能描述
EVINPx (x = 0 - 1, 4 - 7)	[1:0], [7:4]	rw	事件 x 中断节点指针 该位定义了产生事件 x 中断时哪条 SR 有效 0 线 SR0 有效 1 线 SR1 有效
0	[3:2]	r	保留 读操作返回 0; 应写入 0。

寄存器 LCBR 的位域定义了极限检查所需比较值（边界）的四个 MSB。将四个“0”（8 位转换）或六个“0”（10 位转换）串联在位域 BOUND0 和 BOUND1 定义的值之后构成最终结果，和转换结果进行比较。举例说明：BOUND1 (B_H) 的复位值将构成最终比较值 B0_H 用于 8 位比较，2C0_H 用于 10 位比较。

LCBR
极限检查边界寄存器
复位值: B7_H


符号	位序号	读写类型	功能描述
BOUND (x = 0 - 1)	[3:0], [7:4]	rw	极限检查的边界 该位定义了极限检查所需比较值的四个 MSB。极限检查的结果用来产生中断。

14 片上调试支持

片上调试支持（OCDS）提供了基于 XC800 系统软件开发与调试所需的基本功能。

OCDS 设计基于以下原则：

- 利用 XC800 核内置的调试功能
- 增加最少的硬件开销
- 监控器程序支持大部分操作
- 利用标准接口与主机（调试器）通信

特性：

- 在指定的指令地址范围内设置断点
- 在内部 RAM 地址设置断点
- 在 FLASH/RAM 代码区支持无数量限制的软件断点
- 处理外部断点
- 程序代码逐行调试

14.1 功能描述

OCDS 的功能框图如图 14-1 所示。监控器模式控制 (MMC) 是 OCDS 系统的核心, 连接 OCDS 的控制信号, 支持总体调试功能。MMC 主要通过调试接口和 XC800 核进行通信, 并接收复位和时钟信号。MMC 根据来自 CPU 的内存地址和控制信号, 可对专用附加存储器, 即监控器 ROM (存储程序) 和监控器 RAM (存储工作数据和监控器堆栈) 进行恰当访问。OCDS 系统可通过 JTAG¹⁾ 接口访问, JTAG 接口是测试和调试专用接口, 通常在应用中不使用。MBC 引脚专门用于外部设置与调试控制。

注: 这里描述的所有调试功能通常只用于 XC866 启动后工作在 OCDS 模式下。

注: 启动设置选择的详细信息, 参见章节 7.2.3。

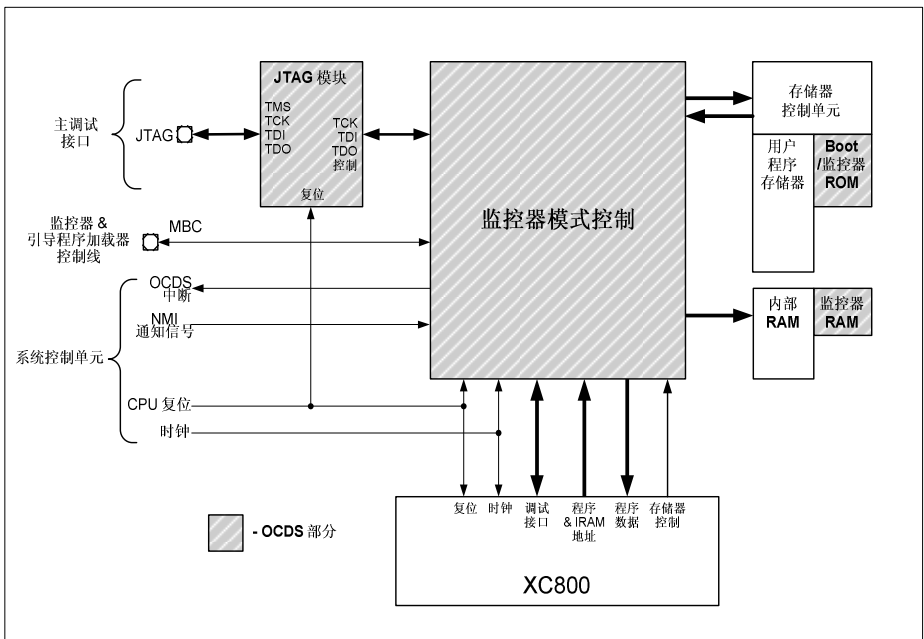


图 14-1 XC866 OCDS: 功能框图

¹⁾ JTAG 端口引脚可设置为端口 0 (首选调试端口) 或端口 1 和 2 (次选调试端口)。在和 OCDS 系统连接时, 用户必须将 JTAG 引脚 (TCK 和 TDI) 设置为输入口。

14.2 调试

片上调试系统可以分为两部分描述。第一部分涵盖了调试事件的产生；第二部分描述了调试事件产生时的调试动作。

- 调试事件
 - 硬件断点
 - 软件断点
 - 外部断点
- 调试事件动作
 - 调用监控器程序
 - 激活 MBC 引脚

XC866 调试操作是基于 OCDS 硬件和专用软件，即监控器程序的紧密作用。

14.2.1 调试事件

OCDS 系统可识别不同的调试事件，通常称其为断点。

根据断点事件处理的时间，可划分为以下三种断点：

- 指令执行前断点

断点刚好发生在断点指令（引起该断点的指令）执行之前。因此，该断点指令为用户程序中的下一条指令，但只在相应的调试动作发生后执行。

- 指令执行后断点

在断点指令（引起该断点的指令）执行之后断点立即发生。因此在相应的调试动作发生时，该断点指令已被执行。

- 立即断点

这类事件和 XC866 内部的指令执行不同步，在这种情况下无“引起调试事件的指令”。调试事件一旦发生，OCDS 即刻执行调试动作。

14.2.1.1 硬件断点

通过监控 XC866 系统的特定地址总线产生硬件断点。与硬件断点类型相关的总线和被设置在特定寄存器中的断点地址进行连续比较。

硬件断点可根据两种类型进行分类：

- 根据监控的地址总线

- **指令地址断点**

监控程序存储器地址（PROGA）

- **IRAM 地址断点**

监控内部数据存储器地址（SOURCE_A, DESTIN_A）

- 根据比较方式

- 点断点

只与一个地址值比较；只在匹配到该值时产生断点事件

- 区间断点

与两个地址值比较；监控的地址值在两个设定的地址值之间（闭区间）时产生断点事件。

指令地址断点

这类指令指针（IP）断点在满足以下条件时产生：断点地址和将要执行的指令的首字节地址匹配，即从程序存储器中取回的指令操作码的起始地址。

注：对于 2 字节和 3 字节指令，第二和第三个指令字节的地址不产生断点。

IP 断点属于“指令执行前断点”类型，只有在相应调试动作执行之后才执行该断点指令。

XC866 的 OCDS 支持指令地址的点断点和区间断点（参见页 14-5 “硬件断点设置”）。

IRAM 地址断点

这类断点在满足以下条件时产生：当一条指令读写内部数据存储器（IRAM）时，断点地址和该 IRAM 的地址匹配。

IRAM 断点属于“指令执行后断点”类型，在断点指令执行之后立即执行相应调试动作。

XC866 OCDS 只支持 IRAM 地址的区间断点。

当内部数据存储器为 RAM 时，OCDS 能够区分对 IRAM 读操作产生断点和写操作产生断点。

硬件断点设置

XC866 中的 OCDS 允许最多设置 4 个硬件断点 HWBP x ($x = 0 - 3$) (16 位值)，各种设置如下所示：

- **HWBP0 和 HWBP1**
 - 两个点断点
指令地址 = HWBP0 和 指令地址 = HWBP1，或
 - 一个区间断点
HWBP0 \leq 指令地址 \leq HWBP1
- **HWBP2**
 - 一个点断点
指令地址 = HWBP2，或
 - 一个区间断点
HWBP2L \leq IRAM 读地址 \leq HWBP2H
- **HWBP3**
 - 一个点断点
指令地址 = HWBP3，或
 - 一个区间断点
HWBP3L \leq IRAM 写地址 \leq HWBP3H

在 XC866 中程序存储器地址为 16 位宽，内部数据存储器地址(读和写)为 8 位宽。因此 HWBP2 和 HWBP3 的全值可用于产生 IP 断点，而高位字节和低位字节 HWBP x L 和 HWBP x H ($x = 2 - 3$) 分别用于产生 IRAM 断点。

若将区间断点的上下地址设置为同一地址值，将会产生点断点。

14.2.1.2 软件断点

这些断点使用 XC800 专用的（非 8051 标准）TRAP 指令，该指令在扩展操作寄存器（EO）的 TRAP_EN 位设置成 1 时，被 CPU 译码。

当读取到 TRAP 指令时，产生一个“指令执行前断点”，执行相应的断点动作。

实际上软件断点的行为与指令地址的点断点相似，区别在于前者由程序代码产生；后者由专门的硬件逻辑（比较逻辑）实现。

在用户程序中通过替换操作码设置软件断点，数量不限。但是，只能在可写存储器（RAM/FLASH）范围内实现。

注：为了在调试事件发生后继续执行用户程序，外部调试器必须恢复当前软件断点所在地址的原代码。

14.2.1.3 外部断点

这类调试事件属于“立即断点”，可通过下面两种方式产生：

- 通过 JTAG 接口请求；利用特殊序列，连接到 JTAG 接口上的外部设备可以中断 XC866 上运行的用户程序，启动调试进程。
- XC866 工作时，将专用监控器和启动程序加载器控制线（MBC）置低；用于响应外部的异步事件。

14.2.2 调试动作

当调试事件发生时，OCDS 系统可根据当前的设置以两种方式响应：

14.2.2.1 调用监控器程序

XC866 带有存储在非易失监控器 ROM 中（见表 14-1）的片上监控器程序。激活该程序是 OCDS 识别调试事件时的主要和基本响应。

OCDS 硬件确保监控器始终安全启动，并在调试动作发生时完全独立于当前的系统状态。此外，其他的中断申请将不会干扰监控器的正常运行。

一旦监控器启动，它将运行在自己的堆栈和数据工作存储器（见图 14-1 中监控器 RAM），从而确保当控制返回到用户程序时，所有的 CPU 和内存资源和进入监控器前相同。

XC866 监控器包括以下功能：

- 通过 JTAG 接口与外部调试器通信
- 可读写任意存储器位置和包括指令指针在内的 SFR
- 设置 OCDS 以及设置/删除断点
- 单指令执行（逐行模式）

注：本手册不提供监控器程序功能以及 JTAG 通信协议的具体描述。

14.2.2.2 激活 MBC 引脚

OCDS 响应调试事件时可将 MBC 引脚置低。

该功能有两种可选择的设置：

- 作为监控器程序启动时的附加动作

片上调试支持

- 作为 CPU 暂时（四个 SCLK 时钟周期）挂起时 OCDS 的唯一响应；这是对外部事件的最快响应。

14.3 寄存器描述

从编程人员的角度看，OCDS 由 8 个寄存器地址来表示（见表 14-1），这些寄存器均位于映射 SFR 区。

表 14-1 OCDS 直接寻址寄存器

寄存器缩略名	地址（映射）	寄存器完整名
MMCR	F1 _H	监控器模式控制寄存器
MMSR	F2 _H	监控器模式状态寄存器
MMBPCR	F3 _H	监控器模式断点控制寄存器
MMICR	F4 _H	监控器模式中断控制寄存器
MMCR2	E9 _H	监控器模式控制寄存器 2
MMDR	F5 _H	监控器模式数据寄存器
HWBPSR	F6 _H	硬件断点选择寄存器
HWBPDR	F7 _H	硬件断点数据寄存器

另外，有 8 个硬件断点寄存器，通过 HWBPSR 和 HWBPDR 间接访问（见表 14-2）。

表 14-2 OCDS 间接访问寄存器

寄存器缩略名	寄存器完整名
HWBP0L	硬件断点 0 低位寄存器
HWBP0H	硬件断点 0 高位寄存器
HWBP1L	硬件断点 1 低位寄存器
HWBP1H	硬件断点 1 高位寄存器
HWBP2L	硬件断点 2 低位寄存器
HWBP2H	硬件断点 2 高位寄存器
HWBP3L	硬件断点 3 低位寄存器
HWBP3H	硬件断点 3 高位寄存器

片上调试支持

注：OCDS 寄存器专门用于片上监控程序，强烈建议用户不要使用这些寄存器，这会导致系统出现不可预测的行为。

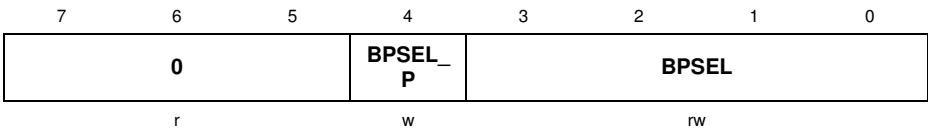
只有当 XC866 不从 OCDS 模式启动，并且无外部设备和 JTAG 接口连接时，硬件断点寄存器才可用作通用寄存器。见表 14-1 表 14-2，描述如下。

HWBPSR

硬件断点选择寄存器

映射 SFR (F6H)

复位值: 00H



符号	位序号	读写类型	功能描述
BPSEL	[3:0]	rw	断点寄存器选择
BPSEL_P	4	w	位保护 0 BPSEL 不可更改 1 BPSEL 可更改
0	[7:5]	r	保留 读操作返回 0；应写入 0

表 14-3 HWBPSR [3:0]：选择硬件断点寄存器

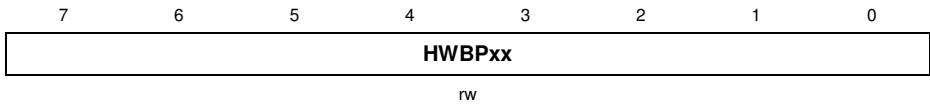
BPSEL	寄存器选择	BPSEL	寄存器选择
0xxx	保留	-	-
1000	HWBP0L	1001	HWBP0H
1010	HWBP1L	1011	HWBP1H
1100	HWBP2L	1101	HWBP2H
1110	HWBP3L	1111	HWBP3H

HWBPDR

硬件断点数据寄存器

映射 SFR (F7H)

复位值: 00H



符号	位序号	读写类型	功能描述
HWBPxx	[7:0]	rw	数据写入/取自当前 HWBPSR 选中的 HWBPxx 寄存器 (见表 14-3)

14.3.1 JTAG 标识寄存器

该寄存器是 JTAG 模块内的只读寄存器，用于识别和 JTAG 接口连接的器件。当指令寄存器中的指令为 IDCODE (操作码 04H) 时，寄存器中的内容可读出；复位后情况相同。

和 XC866 Flash 系列器件对应的 JTAG 标识寄存器的值由表 14-4 给出。

表 14-4 JTAG 标识符总结

器件类型	器件名称	JTAG 标识符
Flash	XC866L-4FR	1010 0083H
	XC866-4FR	100F 5083H
	XC866L -2FR	1010 2083H
	XC866-2FR	1010 1083H

15 索引

15.1 关键词索引

本章节列出了很多关键词，以便查阅与 XC866 的结构、功能单元或模块功能相关的详细描述。

A

- Accumulator (累加器), 2-4**
- Alternate function (其他功能), 6-9**
 - Input (输入), 6-9
 - Output (输出), 6-9
- Analog input clock (模拟输入时钟), 13-3**
- Analog-to-Digital Converter (模数转换单元), 13-1**
 - Interrupt (中断), 13-21
 - Channel (通道), 13-24
 - Event (事件), 13-22
 - Node pointer (节点指针), 13-25
 - Low power mode (低功耗模式), 13-7
 - Module clock (模块时钟), 13-3
 - Register description (寄存器描述), 13-33
 - Register map (寄存器映射), 13-30
- Arbitration round (仲裁周期), 13-9**
- Arbitration slot (仲裁时隙), 13-9**
- Arithmetic (运算), 2-2**
- Asynchronous modes (异步模式), 10-2**
- Automatic refill (自动添加), 13-11**
- Autoscan (自动扫描), 13-14**

B

- Baud rate (波特率), 10-10**
 - Baud rate clock (波特率时钟), 10-11
 - Baud rate generation (波特率产生), 10-28
 - Baud-rate generator (波特率发生器), 10-10
- Bit protection scheme (位保护方案), 3-13**
- Bitaddressable (位可寻址), 3-11**
- Boot option (启动选择), 7-6**
 - BSL mode (启动程序加载器模式), 7-6

- OCDS mode (片上调试支持模式), 7-7
- User mode (用户模式), 7-6
- Boot ROM operation mode (Boot ROM 工作模式), 3-43**
 - Bootstrap Loader mode (启动程序加载器模式), 3-43
 - OCDS mode (片上调试支持模式), 3-44
 - User mode (用户模式), 3-43
- Boot ROM (启动 ROM), 3-1**
- Booting scheme (启动方案), 7-6**
- BootStrap Loader (启动程序加载器), 3-43, 4-6, 4-9**
- Brownout reset (压降复位), 7-5**
- Buffer mechanism (缓存机制), 4-4**

C

- Cancel-Inject-Repeat (取消-插入-重复), 13-10**
- Capture/Compare Unit 6 (捕获/比较单元 6), 12-1**
 - Register description (寄存器描述), 12-32
 - Register map (寄存器映射), 12-29
- Center Processing Unit (中央处理单元), 2-1**
- Circular stack memory (环形堆栈存储器), 4-4**
- Clock Source (时钟源), 7-12**
- Clock System (时钟系统), 7-10**
- Continuous transfer operation (连续传输), 10-27**
- Conversion error (转换出错), 13-4**
- Conversion phase (转换阶段), 13-5**
- Correction algorithm (纠错算法), 4-8**
- CPU (中央处理单元), 2-1**

D

- Data Flash (数据 FLASH), 4-2, 4-3**
- Data memory (数据存储器), 3-3**
- Data Pointer (数据指针), 2-4**
- Data reduction (数据压缩), 13-17**
 - Counter (计数器), 13-18
- Debug (调试), 14-3**
 - Event (事件), 14-3
- D-Flash (数据 FLASH), 4-2, 4-3**
- Digital input clock (数字输入时钟), 13-3**
- Direct drive (直接驱动), 7-12**
- Direct feed-through (直连), 6-4**

Document (文档)

- Acronyms (缩略词), 1-14
- Terminology (术语), 1-13
- Textual convention (命名规则), 1-12

Dynamic error detection (动态检错), 4-8

E

EEPROM emulation (电可擦除可编程 ROM 仿真), 4-4

Embedded voltage regulator (嵌入式电压调节器), 7-1

- Features (特性), 7-1
- Low power voltage regulator (低功率电压调节器), 7-2
- Main voltage regulator (主电压调节器), 7-2
- Threshold voltage levels (阈值电压), 7-2

Error Correction Code (纠错码), 4-8

Extended operation (扩展操作), 2-6

External break (外部断点), 14-6

- Break now (立即断点), 14-6

External data memory (外部数据存储器), 3-3

External oscillator (片外振荡器), 7-10, 7-12

F

Flash device (FLASH 器件), 3-1

Flash program memory (FLASH 程序存储器), 3-1

Flash Timer NMI (FLASH 定时器非屏蔽中断), 4-11

Flash (闪存), 4-1

- Endurance (耐受能力), 4-4
- Erase mode (擦除模式), 4-7
- Non-volatile (非易失), 4-1
- Operating mode (工作模式), 4-7
- Power-down mode (掉电模式), 4-7
- Program mode (编程模式), 4-7
- Program width (编程宽度), 4-6
- Ready-to-read mode (待读模式), 4-7
- Sector (扇区), 4-3

Full-duplex operation (全双工工作), 10-23

G

Gate disturb (门干扰), 4-6

GPIO (通用输入/输出), 6-1

H

Half-duplex operation (半双工工作), 10-26

Hall sensor mode (霍尔传感器模式)

Actual hall pattern (实际霍尔序列), 12-21

Block commutation (块切换), 12-22

Brushless-DC (无刷直流), 12-21, 12-22

Correct hall event (正确霍尔事件), 12-21

Expected hall pattern (期望霍尔序列), 12-21

Hall pattern (霍尔序列), 12-21

Modulation pattern (调制序列), 12-21

Noise filter (噪声滤波器), 12-21

Hamming 码, 4-8

Hardware breakpoint (硬件断点), 14-4

Hardware reset (硬件复位), 7-4

High-impedance (高阻), 6-2

I

Idle mode (空闲模式), 7-14, 8-2

In-Application Programming (在应用编程), 4-10

Input class (输入综合), 13-8

Instruction decoder (指令译码器), 2-2

Instruction timing (指令时序), 2-8, 2-10

CPU state (CPU 状态), 2-8

Mnemonic (助记符), 2-10

Wait state (等待状态), 2-8

In-System Programming (在系统编程), 4-9

Internal analog clock (内部模拟时钟), 13-3

Maximum frequency (最大频率), 13-3

Internal data memory (内部数据存储), 3-3

Internal handling (中断处理), 5-28

Internal RAM (内部随机存取存储器), 3-1

Interrupt priority (中断优先级), 5-24

Interrupt request flag (中断请求标志), 5-26

Interrupt response time (中断响应时间), 5-30

Interrupt source and vector (中断源和中断向量), 5-2, 5-8

Interrupt system (中断系统), 5-1

Register description (寄存器描述), 5-9

J

JTAG ID (JTAG 标识), 14-10

K

Kernel register (核寄存器), 6-5

Direction control register (方向寄存器), 6-6

L

Limit checking (极限检查), 13-17

LIN (局域互联网), 10-16–10-20

Baud rate detection (波特率检测), 10-19

Break field (分隔域), 10-17

Header transmission (头信息传输), 10-18

LIN frame (LIN 的帧结构), 10-16

LIN protocol (LIN 协议), 10-16

Synch byte (同步字节), 10-17

M

Maskable interrupt (可屏蔽中断), 5-1

Extended (扩展中断), 5-2

External (外部中断), 5-2

Internal (内部中断), 5-1

Memory organization (存储器结构), 3-1

Special Function Register (特殊功能寄存器), 3-4

Address extension by mapping (映射地址扩展), 3-4

Mapped (映射 SFR 区), 3-4

Standard (标准 SFR 区), 3-4

Address extension by paging (分页地址扩展), 3-7

Local address extension (局部地址扩展), 3-7

Save and restore (保存和恢复), 3-9

Minimum erase width (最小擦除宽度), 4-4

Modulation (调制), 12-15

Monitor mode control (监控器模式控制), 14-2

Monitor RAM (监控器 RAM), 14-2

Data (数据存储), 14-6

Stack (堆栈), 14-6

Monitor ROM (监控器 ROM), 14-2

Multi-channel mode (多通道模式), 12-19

Multifold replication (多方复制), 4-4
Multiprocessor communication (多处理器通信), 10-7

N

Non-maskable interrupt (非屏蔽中断), 5-1
Events (事件), 5-1

O

On-chip Debug Support (片上调试支持), 14-1
Register description (寄存器描述), 14-8
Register map (寄存器映射), 14-8
On-chip oscillator (片内振荡器), 7-10

P

P0 register description (P0 口寄存器描述), 6-5, 6-15
P1 register description (P1 口寄存器描述), 6-21
P2 register description (P2 口寄存器描述), 6-27
P3 register description (P3 口寄存器描述), 6-32
Parallel ports (并行端口), 6-1
Bidirection port structure (双向口结构), 6-3
Driver (驱动器), 6-2, 6-7
General port structure (基本端口结构), 6-3
General register description (基本寄存器描述), 6-5
Input port structure (单向输入口结构), 6-4
Kernel register (核寄存器)
Open drain control register (开漏输出控制寄存器), 6-7
Normal mode (正常模式), 6-2, 6-7
Open drain mode (开漏输出模式), 6-2, 6-7
Parallel request source (并列请求源), 13-13
Permanent arbitration (持续仲裁), 13-9
Personal computer host (PC 主机), 4-9
Phase-Locked Loop (锁相环), 7-10
Changing PLL parameters (改变 PLL 的参数), 7-11
Loss-of-Lock operation (失锁操作), 7-11
Loss-of-Lock recovery (失锁恢复), 7-11
Pin (引脚)
Configuration (配置), 1-5
Definition and function (定义及功能), 1-6

PLL

Loss-of-Lock (失锁), 7-11
Startup (启动), 7-11
PLL base mode (PLL 基频模式), 7-13
PLL bypass (PLL 旁路), 7-12
PLL mode (PLL 模式), 7-13
Power control (电源控制), 2-7
Power on reset (上电复位), 7-2, 7-3
Power saving mode (省电模式), 8-1
Power supply system (电源系统), 7-1
Power-down mode (掉电模式), 7-14, 8-3
 Entering power-down mode (进入掉电模式), 8-3
 Exiting power-down mode (退出掉电模式), 8-3
Power-down wake-up reset (掉电唤醒复位), 7-4
Prescaler mode (预分频模式), 7-13
Prewarning period (预报警阶段), 9-2
Processor architecture (处理器结构), 2-1
 Instruction timing (指令时序)
 Machine cycle (机器周期), 2-8
 Register description (寄存器描述), 2-4
Program counter (程序计数器), 2-3
Program Flash (程序 Flash), 4-2, 4-3
Program Memory (程序存储器), 3-3
Program status word (程序状态字), 2-5
Pull-down device (下拉器件), 6-7
Pull-up device (上拉器件), 6-7
Pulse width modulation (脉宽调制), 12-1

R

Read access time (读取时间), 4-1
Receive-buffered (接收缓冲), 10-2
Request gating (请求源门控机制), 13-12
Request trigger (转换请求触发), 13-12, 13-14, 13-26
 CCU6 Event (CCU6 事件), 13-26
Reset control (复位控制), 7-3
 Module behavior (模块复位行为), 7-6
Result FIFO (FIFO 型结果数据), 13-19
Result read view (结果读取), 13-19
 Accumulated (累加读取), 13-19
 Normal (正常读取), 13-19
ROM devices (ROM 器件), 3-1, 3-2

ROM program memory (ROM 程序存储器), 3-1
RS-232, 4-9

S

Sample phase (采样阶段), 13-5
Schmitt-Trigger (施密特触发器), 6-2, 6-4
Sectorization (分区), 4-3
Sequential request source (顺序请求源), 13-11
Serial data (串行数据), 6-2
Serial interface 串行接口, 10-1–10-20
Serial port (串行端口), 6-2
Slow-down mode (减速模式), 7-14, 8-2
Software breakpoint (软件断点), 14-5
Break before make (指令执行前断点), 14-5
Source priority (请求源优先级), 13-9
Special Function Register area (特殊功能寄存器区), 3-1
Stack pointer (堆栈指针), 2-4
Synchronization phase (同步阶段), 13-5
Synchronous serial interface (同步串行接口), 10-21
Data Width (数据宽度), 10-22
Error detection (检错), 10-29
Baud rate error (波特率出错), 10-30
Phase error (相位出错), 10-30
Receive error (接收出错), 10-30
Transmit error (发送出错), 10-31
Interrupt (中断), 10-29
Master mode (主模式), 10-21
Operating mode (工作模式), 10-22
Right-aligned (右对齐), 10-22
Slave mode (从模式), 10-21

T

Timer 0 and Timer 1 (定时器 0 和定时器 1), 11-1–11-12
External control (外部控制), 11-2
Mode 0, 13-bit timer (模式 0, 13 位定时器), 11-3
Mode 1, 16-bit timer (模式 1, 16 位定时器), 11-4
Mode 2, 8-bit automatic reload timer (模式 2, 8 位自动重载定时器), 11-5
Mode 3, two 8-bit timers (模式 3, 两个 8 位定时器), 11-6
Timer operation (定时器运行), 11-1

Timer overflow (定时器溢出), 11-1

Timer 2 (定时器 2), 11-13–11-22

Auto-Reload mode (自动重载模式), 11-13

Capture mode (捕获模式), 11-16

Up/Down Count Disabled (递增/递减计数禁止), 11-13

Up/Down Count Enabled (递增/递减计数使能), 11-14

Timer T12 (定时器 T12), 12-3

Capture mode (捕获模式), 12-9

Center-aligned mode (中间对齐模式), 12-5

Compare mode (比较模式), 12-6

Dead-time (死区时间), 12-8

Duty cycle (占空比), 12-8

Edge-aligned mode (边沿对齐模式), 12-5

Hysteresis-like control mode (类磁滞控制模式), 12-10

Shadow transfer (映射传输), 12-3

Single-shot mode (单次模式), 12-10

Three-phase PWM (3相 PWM), 12-1

Timer T13 (定时器 T13), 12-12

Compare mode (比较模式), 12-13

Shadow transfer (映射传输), 12-12

Single-shot mode (单次模式), 12-13

Total conversion time (转换总时间), 13-5

Trap handling (强制中断处理), 12-18

Tristate (三态), 6-7

U

UART (通用异步收发器), 10-2–10-15

Interrupt request (中断请求), 10-5

Mode 1, 8-bit UART (模式 1, 8 位 UART), 10-2

Mode 2, 9-bit UART (模式 2, 9 位 UART), 10-5

Mode 3, 9-bit UART (模式 3, 9 位 UART), 10-5

V

VCO bypass (压控振荡器旁路), 7-13

W

Wait-for-read mode (待读模式), 13-15

Wait-for-start (等待开始), 13-10

Watchdog timer reset (看门狗定时器复位), 7-4

Watchdog timer (看门狗定时器), 9-1–9-9

Input frequency (输入频率), 9-3

Servicing (服务, 刷新), 9-2

Time period (溢出周期), 9-3

Window boundary (窗界), 9-2

Wordline address (字线地址), 4-5

Write result phase (结果写入阶段), 13-5

Writer buffers (写缓冲存储器), 4-6

X

XC866 register overview (XC866 寄存器概览), 3-14

XRAM, 3-1

15.2 寄存器索引

本章节列出 XC866 中的 SFR，以便查阅。

A		D
A, 2-4		
ADC_PAGE, 13-31		DPH, 2-4
		DPL, 2-4
B		E
B, 2-4		
BCON, 10-13		EO, 2-6
BG, 10-14		ETRCR, 13-38
BRH, 10-40		EVINCR, 13-58
BRL, 10-40		EVINFR, 13-58
		EVINPR, 13-59
		EVINSR, 13-59
C		EXICON0, 5-13
CC63RH, 12-48		EXICON1, 5-14
CC63RL, 12-48		
CC63SRH, 12-49		F
CC63SRL, 12-49		
CC6xRH (x = 0 - 2), 12-42		FEAH, 4-9
CC6xRL (x = 0 - 2), 12-42		FEAL, 4-9
CC6xSRH (x = 0 - 2), 12-43		
CC6xSRL (x = 0 - 2), 12-43		G
CCU6_PAGE, 12-30		
CHCTR _x (x = 0 - 7), 13-39		GLOBCTR, 8-8, 13-34
CHINCR, 13-56		GLOBSTR, 13-35
CHINFR, 13-56		
CHINPR, 13-57		H
CHINSR, 13-57		
CMCON, 7-19		HWBPDR, 14-11
CMPMODIFH, 12-52		HWBPSR, 14-9
CMPMODIFL, 12-52		
CMPSTATH, 12-51		I
CMPSTATL, 12-50		
CONH, 10-36		ID, 3-19
CONL, 10-35		IEN0, 5-9, 11-12
CRCR1, 13-48		IEN1, 5-10
CRMR1, 13-50		IENH, 12-99
CRPR1, 13-49		IENL, 12-97
		INPCR0, 13-40
		INPH, 12-103

INPL, 12-101

IP, 5-21

IP1, 5-22

IPH, 5-21

IPH1, 5-22

IRCON0, 5-15

IRCON1, 5-16

ISH, 12-86

ISL, 12-85

ISRH, 12-95

ISRL, 12-93

ISSH, 12-91

ISSL, 12-89

L

LCBR, 13-60

M

MCMCTR, 12-77

MCMOUTH, 12-75

MCMOUTL, 12-73

MCMOUTSH, 12-72

MCMOUTSL, 12-71

MMBPCR, 14-8

MMCR, 14-8

MMCR2, 14-8

MMDR, 14-8

MMICR, 14-8

MMSR, 14-8

MODCTR, 12-64

MODCTRL, 12-63

MODPISEL, 8-7, 10-14

N

NMICON, 5-11

NMISR, 5-19

O

OSC_CON, 7-16, 8-9

P

P0_ALTSEL0, 6-18

P0_ALTSEL1, 6-18

P0_DATA, 6-16

P0_DIR, 6-16

P0_OD, 6-17

P0_PUDEN, 6-18

P0_PUDEL, 6-17

P1_ALTSEL0, 6-24

P1_ALTSEL1, 6-24

P1_DATA, 6-22

P1_DIR, 6-22

P1_OD, 6-23

P1_PUDEN, 6-24

P1_PUDEL, 6-23

P2_DATA, 6-28

P2_PUDEN, 6-29

P2_PUDEL, 6-28

P3_ALTSEL0, 6-35

P3_ALTSEL1, 6-35

P3_DATA, 6-33

P3_DIR, 6-33

P3_OD, 6-34

P3_PUDEN, 6-35

P3_PUDEL, 6-34

PASSWD, 3-14

PCON, 2-7, 8-6, 10-10

PISEL, 10-34

PISEL0H, 12-38

PISEL0L, 12-36

PISEL2, 12-39

PLL_CON, 7-17

PMCON0, 7-8, 8-5, 9-9

PMCON1, 8-7, 13-8

PORT_PAGE, 6-11

PRAR, 13-36

PSLR, 12-69

PSW, 2-5

Px_ALTSELn, 6-10

Px_DATA, 6-6

Px_DIR, 6-6

Px_OD, 6-7

Px_PUDEN, 6-8

Px_PUDEL, 6-8

Q

Q0R0, 13-44
QBUR0, 13-45
QINR0, 13-46
QMRO, 13-41
QSR0, 13-43

R

RBL, 10-41
RC2H, 11-21
RC2L, 11-21
RCRx (x = 0 - 3), 13-54
RESRAxH (x = 0 - 3), 13-53
RESRAxL (x = 0 - 3), 13-53
RESRxH (x = 0 - 3), 13-53
RESRxL (x = 0 - 3), 13-53

S

SBUF, 10-7
SCON, 5-18, 10-8
SCU_PAGE, 3-12
SP, 2-4
SYSCON0, 3-6

T

T12DTCH, 12-44
T12DTCL, 12-44
T12H, 12-40
T12L, 12-40
T12MSELH, 12-81
T12MSELL, 12-79
T12PRH, 12-41
T12PRL, 12-41
T13H, 12-46
T13L, 12-46
T13PRH, 12-47
T13PRL, 12-47
T2CON, 11-19
T2H, 11-22
T2L, 11-22
T2MOD, 11-17

TBL, 10-41
TCON, 5-17, 11-10
TCTR0H, 12-55
TCTR0L, 12-53
TCTR2H, 12-59
TCTR2L, 12-57
TCTR4H, 12-61
TCTR4L, 12-60
TH_x (x = 0 - 1), 11-8
TL_x (x = 0 - 1), 11-8
TMOD, 11-11
TRPCTRH, 12-67
TRPCTRL, 12-66

V

VFCCR, 13-54

W

WDTCON, 9-6
WDTH, 9-8
WDTL, 9-8
WDTREL, 9-6
WDTWINB, 9-8

<http://www.infineon.com>

英飞凌科技出版