



MC51F7424B 用户手册 V1.0.7

SinoMCU 8 位增强型 8051 单片机



1T 8051 内核 FLASH 型 MCU，主频高达 24MHz，18KB FLASH ROM，2KB SRAM，128B EEPROM-like，12 位 ADC，2 个模拟比较器，1 个运算放大器，6 通道 16 位 PWM（电机专用），4 个 16 位定时器，2 路 UART，SPI，硬件乘法器、双两线调试口

主要特性

CPU

- ◇ 1T 高速增强型 8051 内核
- ◇ 双 DPTR

硬件乘法器

- ◇ 集成单周期执行时间 16 x 16 bit 硬件乘法器
- ◇ 集成 8 周期执行时间 32 / 16 bit 硬件除法器
- ◇ 集成 1 周期 32bit 硬件移位单元

存储器

- ◇ 片上 18K 字节 FLASH，支持 18k main flash 或 16k main flash+2k bootloader
- ◇ 2304 (256+2048) 字节 RAM
- ◇ 片上 128 字节 EEPROM-LIKE
- ◇ 支持代码分区保护功能（有效防止非法读、写）
- ◇ 支持在电路编程（ICP）
- ◇ 支持 bootloader（ISP）功能，仅 16k main flash+2k bootloader 模式下有效

时钟源

- ◇ 片上高精度 48MHz RC 高速振荡器（常温 3.0V 电压下 ±1% 精度；全工作条件下 ±2.5% 精度）
- ◇ 片上 32KHz 超低功耗 RC 低速振荡器，供看门狗定时器和 T3 使用
- ◇ 外部可选接 32768Hz 或 4~24MHz 晶体振荡器

电源管理

- ◇ 工作电压 2.0V~5.5V
- ◇ 内建 LDO 稳压器，用于 CPU 内核电压
- ◇ 3 种工作模式：NORMAL/STOP/SLEEP
- ◇ 增加高级能耗控制功能，满足用户低功耗的需求

复位

- ◇ 支持 5 种复位类型：上电复位（POR/PDR）、nRST 外部复位、低电压复位（LVR）、看门狗复位（WDT）、软件复位
- ◇ LVR 支持电压 4 级可选：2.1V、2.5V、3.5V、4.1V
- ◇ 看门狗复位，可选 8 种溢出时间

I/O

- ◇ 最多 30 个双向通用 I/O 口（32 pin 封装下）
- ◇ 支持 4 种输入/推挽输出/开漏输出/高阻模式，内部上拉电阻独立控制
- ◇ 内建数字功能端口全映射模块

定时器/计数器

- ◇ 2 个 16 位 T0/T1 定时器，兼容标准 8051
- ◇ 1 个 16 位增强型 T2 定时器，兼容 8052 的 T2，带输入捕获和输出比较功能

- ◇ 1 个 16 位 T3 时基定时器，可连接到外部 32768Hz 晶振，在停止/休眠模式下具有唤醒功能
- ◇ Timer0~2 计数溢出触发 ADC 采样

PWM 定时器

- ◇ 1 路独立 8 位 PWM0，可作通用定时器
- ◇ 3 组共 6 路 16 位 PWM1 阵列，可互补输出且死区时间可调，并具有故障保护中断功能
- ◇ PWM1 可在选片上高速时钟或其分频下独立工作
- ◇ PWM1 三种时基计数模式：边沿对齐计数、中心对齐计数和单次计数模式
- ◇ PWM1 两种波形输出模式：互补模式与独立模式
- ◇ PWM1 提供 PWM 周期匹配、归零匹配和占空比比较匹配中断
- ◇ PWM1 支持周期匹配、归零匹配信号后分频功能
- ◇ PWM1 提供寄存器重载使能位以保证寄存器重载同步
- ◇ PWM1 计数周期、归零和占空比触发 ADC 采样
- ◇ PWM1 支持触发 Timer2 开始计数

12 位高精度 ADC

- ◇ 12 位高精度逐次逼近型，高达 11 个外部输入通道 + 内部 5 路通道
- ◇ 参考电压可选：内部参考、VDD、Vref 引脚输入
- ◇ 多种触发启动方式
- ◇ 支持数字比较功能
- ◇ 内建温度传感器，连接至 ADC 通道

2 路 UART

- ◇ 2 路 UART 模块，UART0 兼容 8051 标准
- ◇ 增强 UART0 支持“帧出错”检测及自动地址识别
- ◇ 支持 8 位同步半双工、8 位/9 位异步全双工等 4 种工作方式
- ◇ UART1 自带波特率发生器，支持波特率微调

SPI

- ◇ 支持全双工，3 线/4 线同步模式，主/从机可选
- ◇ 支持主机模式错误用以防止主机冲突

模拟比较器 CMP0/1

- ◇ 正/反向输入支持高达 10（CMP0）或 9（CMP1）路外部端口输入
- ◇ 正/反向输入可选择内部信号
- ◇ 内部 LDO 输出
- ◇ GND/VDD
- ◇ 专用 6-bit DAC 参考
- ◇ 内置施密特窗口和滤波电路

本公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。

同时保留在未通知的情况下，对本文档做更改的权利。

晟矽微电

<http://www.sinomcu.com/>



◇ 输出可作为 PWM1 刹车检测信号

运算放大器 OPA

- ◇ 正/反向输入支持高达 7 路外部端口输入
- ◇ 内部固定多档倍数放大可选
- ◇ 内部输出直接作为 ADC 内部通道进行采样

中断

- ◇ INTOX、T0、INT1X、T1、UART0、CMP0/1、ADC、T2、SPI、PWM1、PWM0、PWM1FB、UART1、CRC/CSS、T3 共 15 个中断源
- ◇ 4 级中断优先级可设
- ◇ 其中 INTnX (n=0~1, X=0~7) 支持多重映射输入, 8 选 1 分别对应两个中断源 (INT0/INT1); 支持上

升沿触发方式和高电平脉宽测量功能

循环冗余校验算法模块 (符合 CRC-16 标准)

双两线调试与编程接口

- ◇ 两组调试和编程接口任意二选一, 支持自动识别
- ◇ 支持 boot loader 引导装载程序

96-bit unique ID

工作环境温度

- ◇ -40°C~105°C

封装形式

- ◇ LQFP32/QFN32/**SOP28**/QFN24/SSOP24/TSSOP20/**SOP20**

芯片对比

模块/功能		MC51F7424	MS51F7424B
Core	时钟	高频时钟源分频, 由寄存器控制 系统时钟源切换, 由用户配置字控制	高频时钟源分频和系统时钟源切换, 系统时钟源切换, 由用户配置字控制
	内核&指令集	双 DPTR 第一代 8051 内核	双 DPTR 第一代 8051 内核
存储器	ROM	18K	18K
	BootLoad	2K rom size (占用 FLASH, 此时 FLASH 只有 16K)	2K rom size (占用 FLASH, 此时 FLASH 只有 16K)
	SRAM	256+2048 (idata+idata)	256+2048 (idata+idata)
	SFR 地址映射	SFR0+XSFR	SFR0+XSFR
	EEPROM	128	128
时钟源	HIRC	48M 固定 2 分频	48M 固定 2 分频
	LIRC	32K	32K
	EXT	不支持	32768 + (4M~24M)
复位	复位	POR/LVR/RST/WDT/SOFT	POR/LVR/RST/WDT/SOFT
	LVR 电压	2.1V/2.5V/3.5V/4.1V	2.1V/2.5V/3.5V/4.1V
I/O	数量	28	30
	模式	4 种 (高速/低速/停止/休眠)	4 种 (高速/低速/停止/休眠)
	上/下拉	全上拉	全上拉
	数字全映射	支持	支持
	外部中断	2×8	2×8
定时器	16 位 T0/T1	支持触发 ADC 采样	支持触发 ADC 采样
	16 位 T2	支持触发 ADC 采样	支持触发 ADC 采样
	16 位 T3	√	√
PWM	PWM0	1 路	1 路
	PWM1	16 位, 3 对互补输出+6 路独立输出 支持电机专用控制 支持 HIRC48M 独立时钟	16 位, 3 对互补输出+6 路独立输出 支持电机专用控制 支持 HIRC48M 独立时钟
	PWM1FBCR 寄存器 bit0 位 (FBSTA) 初值	0	1
ADC	输入通道	ANO~AN10 (9 路, 无 AN2、AN9), GND, LDO, TEMP, OPA	ANO~AN10 (11 路), VDD , GND, LDO, TEMP, OPA



模块/功能		MC51F7424	MS51F7424B
	参考电压	VDD, 1.62V (int), VREF (pin+1.2/2.4v), LDO	VDD, 1.62V (int), VREF (pin+1.2/2.4v), LDO
	温度传感器	支持	支持
	ADC 数字比较器	支持	支持
模拟比较器	外设数量	2 路 (CMP0/1)	2 路 (CMP0/1)
	正反向输入通道	每个比较器 (CMP0/1) 支持 9 路输入	CMP0 支持 10 路, CMP1 支持 9 路输入
	DAC 参考	支持 6-bit 分辨率 电压源支持所有外部通道	支持 6-bit 分辨率 电压源支持所有外部通道
	施密特窗口电路	4 档可选	4 档可选
	PWM1 刹车	支持	支持
运算放大器	外设数量	1 (OPA0)	1 (OPA0)
	正反向输入通道	放大器 (OPA0) 支持 7 路输入	放大器 (OPA0) 支持 7 路输入
	内部放大增益	1/2/4/8/16/32/64	1/2/4/8/16/32/64
	输出通道	3 路可选	3 路可选
	伪差分方式	不支持	不支持
	正向内建直流偏置	不支持	不支持
LVD		不支持	不支持
通讯	UART	UART0 (T1/T2 波特率发生器) UART1 (支持独立波特率发生器)	UART0 (T1/T2 波特率发生器) UART1 (支持独立波特率发生器)
	SPI	1 路	1 路
	IIC	无	无
其他	FLASH 编程	ICP, IAP, ISP (bootloader)	ICP, IAP, ISP (bootloader)
	软件 LCD	不支持	不支持
	unique ID	支持	支持
	功耗	-	-
	工作温度	-40℃~105℃	-40℃~105℃
	2 线烧写/仿真口	2 组	2 组
	sleep 唤醒	不可使用	可以使用



1 产品介绍

1.1 概述

本产品是一款高速低功耗 1T 周期 8051 内核 8 位增强型 FLASH 微控制器芯片，较传统 8051 而言，运行更快、效率更高。

最高运行频率为 24MHz，片上集成了 18K 字节 FLASH ROM 和 2304 字节 RAM（内部 RAM 256B 加上外部 RAM 2048B），内置 1 高频 RC 振荡器和 1 个低频 RC 振荡器，1 组外部晶体振荡器驱动接口，可用户选项配置为高频 4~24MHz 或 32768Hz 晶体，最大包含 30 个双向 GPIO 口，集成灵活的二选一调试和编程接口，具有 4 个 16 位定时器、3 组 6 通道 16 位共周期且占空比可调的 PWM 定时器（带有死区互补和刹车保护功能），集成了 1 个 11 外部通道高精度 12 位 ADC 模块，2 路 UART，以及 1 路 SPI，集成了 2 组模拟电压比较器和 1 组模拟运算放大器，同时内部集成 4 级可选的 LVR、看门狗定时器、CRC 校验等容错功能，极大地提高了芯片的可靠性。

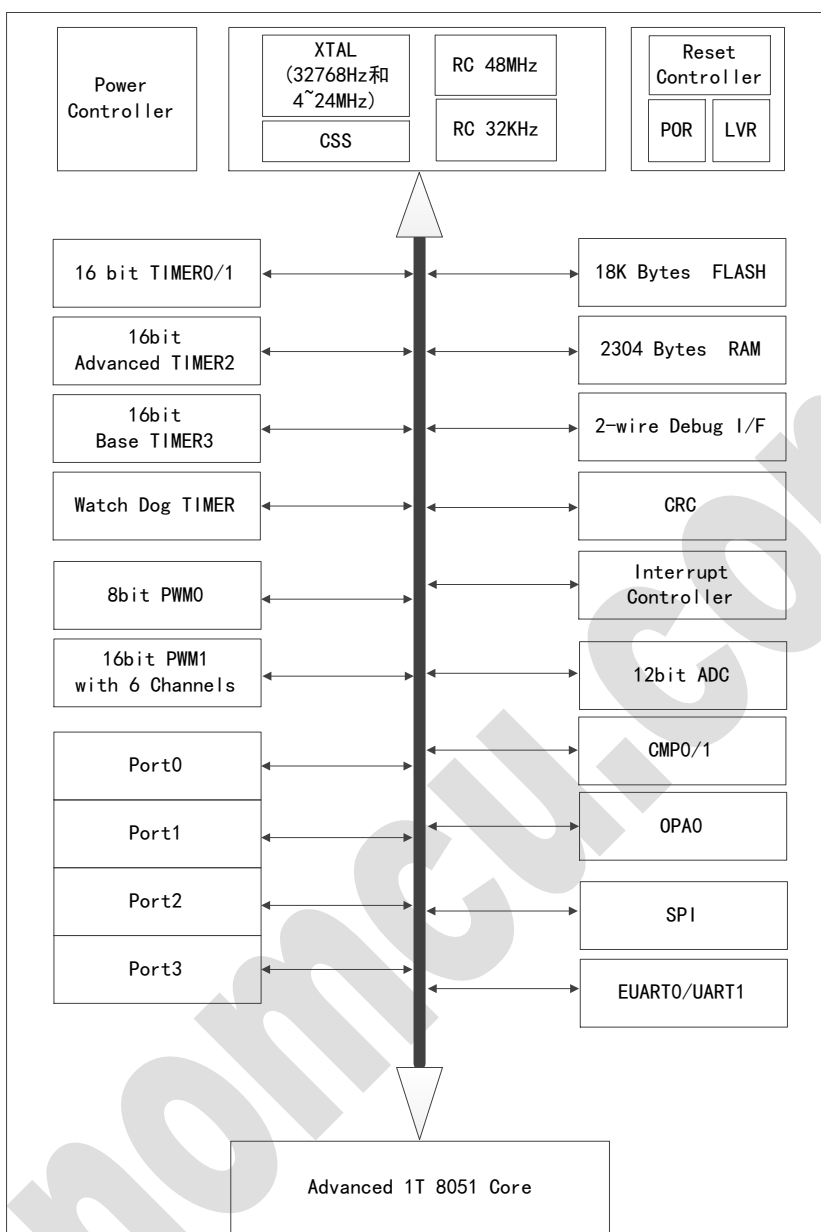
本产品内建数字端口全映射模块，内建温度传感器和运算放大器输出连接至 ADC 通道，供电电压为 2.0V 至 5.5V，温度范围 -40° C 至 +105° C。支持一系列的省电模式保证低功耗应用的要求。

提供包括 QFN32/24 紧凑型封装形式。

本产品为工业级 IC，具有较强的抗干扰性能，内部集成了电机专用 PWM 模块、高精度 ADC、模拟比较器以及运放模块，简化了系统设计，降低了整体成本，可广泛用于电机控制应用领域。



1.2 系统框图

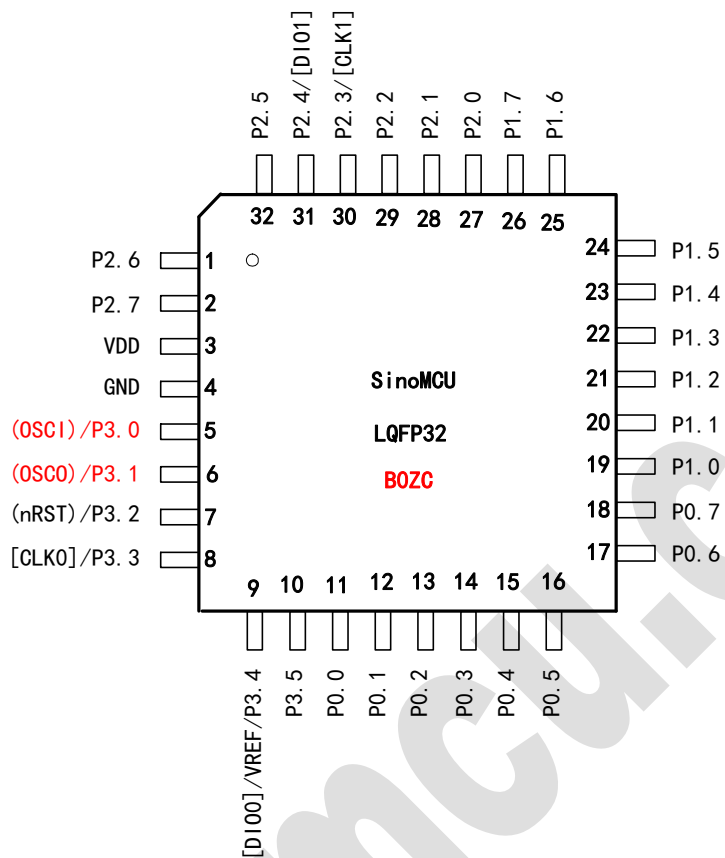




1.3 引脚排列

1.3.1 LQFP32

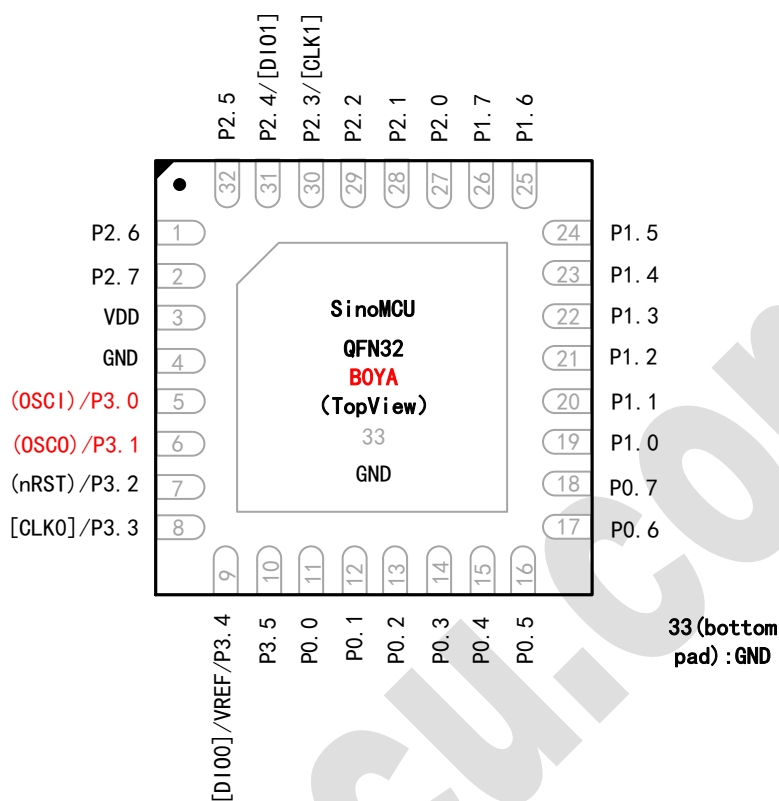
MC51F7424B0ZC



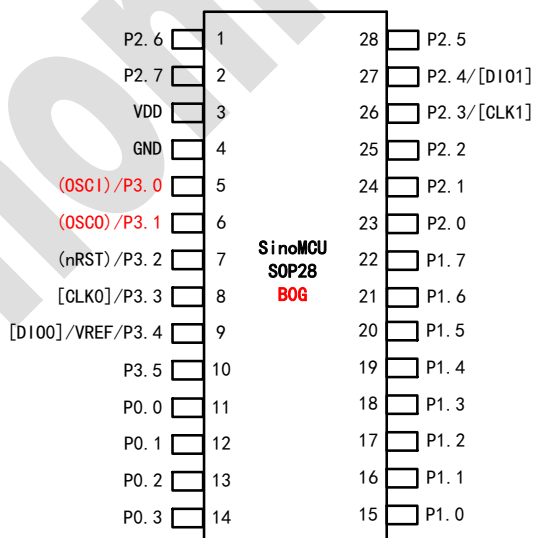


1.3.2 QFN32

MC51F7424B0YA



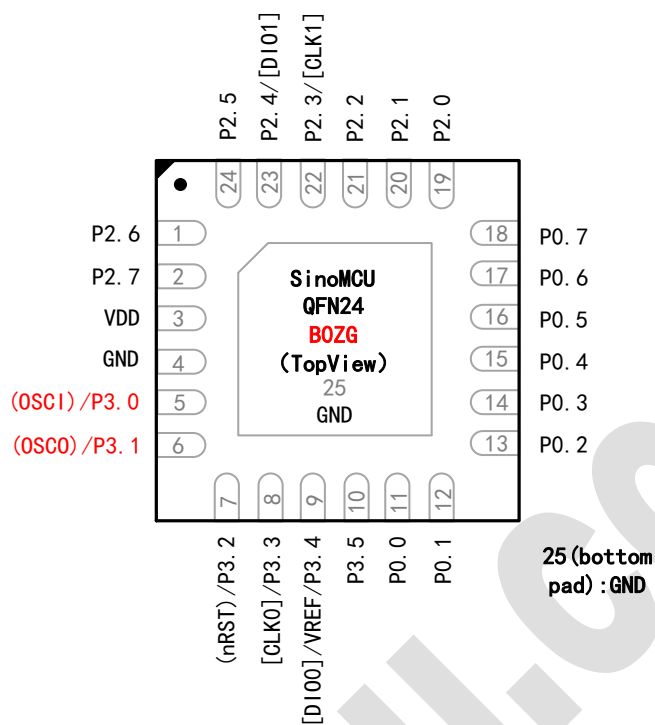
MC51F7424B0G





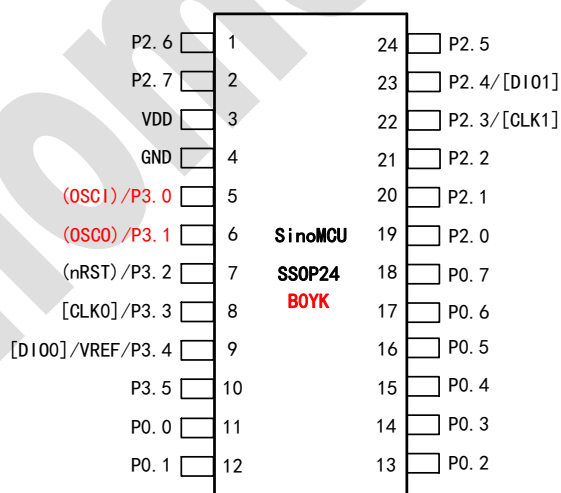
1.3.3 QFN24

MC51F7424B0ZG



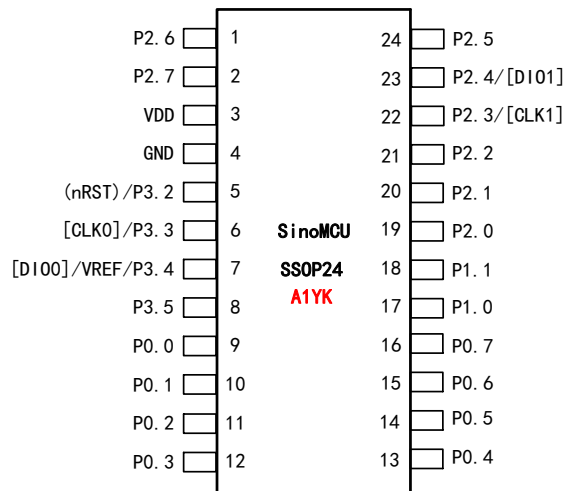
1.3.4 SSOP24

MC51F7424B0YK



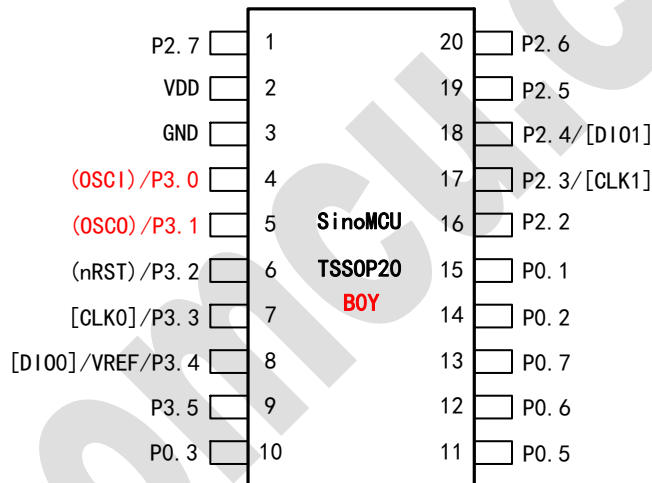


MC51F7424A1YK

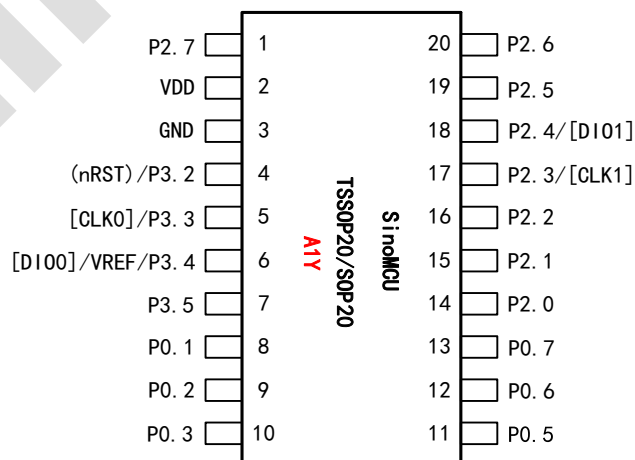


1.3.5 TSSOP20/SOP20

MC51F7424B0Y



MC51F7424A1Y



注 1: 图中, 用方括号 “[] ” 括起来的功能, 代表烧写、调试专用口; 用圆括号 “ () ” 括起来的, 代表这些复用功



能为在“用户配置字”中设定；其中“nXXX”的n，代表其为低电平有效。

注 2：原则上同一个引脚同一时刻只可选择一种功能使用，当一个引脚同时打开了多种复用功能时，其中最高优先级的功能有效，其他功能则被屏蔽。

1.4 引脚说明

引脚汇总表

引脚编号					引脚名	属性	数字全映射	功能符号	
LQFP/ QFN32	QFN/SS OP24	SSOP24	TSSOP20	TSSOP20 /SOP20				非全映射数字功能	模拟功能
B0	B0	A1YK	B0	A1Y					
1	1	1	20	20	P2.6	I/O	√	INT11	AIN4 CMP03 OPA03
2	2	2	1	1	P2.7	I/O	√	INT10	AIN3 CMP02 OPA02
3	3	3	2	2	VDD	VDD	—	—	—
4	4	4	3	3	GND	GND	—	—	—
5	5	—	4	—	P3.0	I/O	√	INT00	(OSCI) AIN9
6	6	—	5	—	P3.1	I/O	√	INT01	AIN2 CMP01 OPA01 (OSCO)
7	7	5	6	4	P3.2	I/O	√	INT02 (nRST) 注 1	AIN1 CMP00 OPA00
8	8	6	7	5	P3.3	I/O	√	INT03 [CLK0] 注 1	AIN0
9	9	7	8	6	P3.4	I/O	√	[DI00]	VREF AIN10
10	10	8	9	7	P3.5	I/O	√	INT04	OPA0OUT
11	11	9	—	—	P0.0	I/O	√	INT05	—
12	12	10	15	8	P0.1	I/O	√	INT06	CMP18
13	13	11	14	9	P0.2	I/O	√	INT07	CMP17
14	14	12	10	10	P0.3	I/O	√	—	CMP16
15	15	13	—	—	P0.4	I/O	√	—	CMP09
16	16	14	11	11	P0.5	I/O	√	—	CMP15
17	17	15	12	12	P0.6	I/O	√	—	CMP14
18	18	16	13	13	P0.7	I/O	√	—	CMP13
19	—	17	—	—	P1.0	I/O	√	—	—
20	—	18	—	—	P1.1	I/O	√	—	—
21	—	—	—	—	P1.2	I/O	√	—	—
22	—	—	—	—	P1.3	I/O	√	INT17	—
23	—	—	—	—	P1.4	I/O	√	INT16	—
24	—	—	—	—	P1.5	I/O	√	INT15	—
25	—	—	—	—	P1.6	I/O	√	INT14	—
26	—	—	—	—	P1.7	I/O	√	—	—
27	19	19	—	14	P2.0	I/O	√	—	CMP12 OPA0OUT
28	20	20	—	15	P2.1	I/O	√	—	CMP08 OPA07
29	21	21	16	16	P2.2	I/O	√	—	AIN8 CMP07 OPA06
30	22	22	17	17	P2.3	I/O	√	[CLK1]	AIN7 CMP06 OPA05
31	23	23	18	18	P2.4	I/O	√	INT13 [DI01]	AIN6 CMP05/11 OPA04
32	24	24	19	19	P2.5	I/O	√	INT12 EADTRG	AIN5 CMP04/10



引脚编号					引脚名	属性	数字全映射	功能符号	
LQFP/ QFN32	QFN/SS OP24	SSOP24	TSSOP20	TSSOP20 /SOP20				非全映射数字功能	模拟功能
B0	B0	A1YK	B0	A1Y					OPA0OUT

1.5 订购信息

产品名称	封装形式	备注
MC51F7424B0ZC	LQFP32	
MC51F7424B0YA	QFN32	4x4
MC51F7424B0G	SOP28	
MC51F7424B0ZG	QFN24	4x4
MC51F7424B0YK/ MC51F7424A1YK	SSOP24	引脚间距 (e=0.635)
MC51F7424B0Y/ MC51F7424A1Y	TSSOP20	引脚间距 (e=0.65)
MC51F7424A1Y	SOP20	引脚间距 (e=1.27)

注：产品名称中的倒数第二位 x 代表本产品的出货包装信息，最终的具体类型请与我司销售人员确认。

1.6 基本概念和术语

- ✧ **R/W**：代表 Read/Write 的含义，即可读可写。当仅有 R 时，代表只读；当仅有 W 时，代表只写。
- ✧ **R*/W, R/W***：R*代表仅可读 0 或 1；W*代表仅可写 0 或 1。
- ✧ **0/1/X/-**：其中 0=数字逻辑 0（低电平）；1=数字逻辑 1（高电平）；X=不确定（高低电平皆可）；-=保留，有点类似于 X（高低电平皆可）。
- ✧ **B/b**：B 代表 Byte 的含义，即字节，表示了 8 位数据长度；b 代表 bit 的含义，即位，表示 1 位数据长度。
- ✧ **双两线调试接口 (Two-wire)**：是一种类似于 SWD 的调试接口形式，是晟矽微电子特有的调试/编程接口。为用户精简了调试/编程接口电路，方便了硬件工程师的 PCB 设计；通过两组调试接口交替仿真，让工程师可以做到“所有端口的全覆盖仿真”。
- ✧ **ICP**：代表在电路编程，即 In Circuit Programming 的首字母缩写。用户可通过电路板上预留的两线调试和编程接口 (Two-wire)，在印刷电路板组装以后 (PCBA)，直接将程序下载到芯片中。
- ✧ **IAP**：代表在应用编程，即 In Application Programming 的首字母缩写。只要芯片可以运行，用户就可以通过程序对芯片本身进行编程。
- ✧ **FLASH 的分区保护功能**：用户可以部分或者全部保护 FLASH 区。可以防范用 Flash 区中的代码在不可信的情况下被读出，也可以防范程序跑飞对 Flash 的意外擦除或者内容改写，还可以为用户的二次开发提供有效手段，这是一种高可靠性的安全防护措施。



2 电气参数

下述无特殊说明的情况时， $T_a=25^{\circ}\text{C}$ ， $V_{DD}-GND=5V$ 。

2.1 极限参数

参数	符号	值	单位
直流供电电压	V_{DD}/GND	$-0.3 \sim 5.5$	V
I/O 输入/输出电压	V_{in}/V_{out}	$GND-0.3 \sim V_{DD}+0.3$	V
工作环境温度	T_a	$-40 \sim 105$	$^{\circ}\text{C}$
储存环境温度	T_{stg}	$-55 \sim 150$	$^{\circ}\text{C}$
片上 FLASH/EEPROM 操作温度	T_{flash}	$-40 \sim 105$	$^{\circ}\text{C}$

注：流过 V_{DD} 、 GND 电流总和的最大值在 $5.0V/25^{\circ}\text{C}$ 下必须小于 200mA 。本规格仅采用本芯片在上述条件内测试获得，对超出上述条件的“特殊应用”不做保证。若芯片长期处于极限参数的最大或者最小条件下工作，可能会直接影响器件的可靠性并加速器件的老化。

2.2 推荐工作条件

参数	符号	最小值	最大值	单位	备注
工作电压	$V_{DD}-GND$	2.0	5.5	V	CORE 可运行于 HIRC@375KHz~24MHz LIRC@32KHz 外部 OSC@4~24MHz 外部 OSC@32.768KHz 建议按需开启 LVR。
工作环境温度	T_a	-40	105	$^{\circ}\text{C}$	-

2.3 直流特性参数

特性	符号	引脚	条件	最小	典型	最大	单位
输入低电压 (斯密特输入方式)	V_{il}	所有输入脚	I/O 输入模式下	$GND-0.3$	-	$0.2V_{DD}$	V
输入高电压 (斯密特输入方式)	V_{ih}	所有输入脚	I/O 输入模式下	$0.8V_{DD}$	-	$V_{DD}+0.3$	V
输入漏电流	I_{ilc}	所有输入脚	I/O 输入模式下， $V_{in}=V_{DD}$ 或者 GND	-1	-	1	μA
输出漏电流	I_{olc}	所有输入脚	I/O 输出模式下， $V_{out}=V_{DD}$ 或者 GND	-1	-	1	μA
输出高电流 (推挽输出)	I_{oh01} I_{oh02}	所有 I/O	I/O 输出模式下， $V_{oh}=0.9V_{DD}$	-	10 20	-	mA
输出低电流 (推挽输出)	I_{ol01} I_{ol02}	所有 I/O	I/O 输出模式下， $V_{ol}=0.1V_{DD}$	-	20 40	-	mA
内部上拉电阻	R_{pu}	P0, P1, P2, P3	$V_{DD}=5V$ ， $V_{in}=0$ ，输入上拉功能使能	-	20	-	$\text{K}\Omega$
输入 RESET 低电平	V_{ilrst}	nRST pin	作为 RESET 使用时	GND	-	$0.2V_{DD}$	V
输入 RESET 高电平	V_{ihrst}	nRST pin	作为 RESET 使用时	$0.8V_{DD}$	-	V_{DD}	V
RESET 引脚的内部上拉电阻	R_{rst}	nRST pin	作为 RESET 使用时	20	-	600	$\text{K}\Omega$
工作电流 (所有未作说明的功能模块关闭；所有引脚无负载；CORE 执行 NOP 指令)	I_{ddc}	V_{DD}	$V_{DD}=5V$ ， $F_{CORE}=24\text{MHz}@HIRC$	-	7.5	-	mA
			$V_{DD}=5V$ ， $F_{CORE}=12\text{MHz}@HIRC$	-	5	-	mA
			$V_{DD}=5V$ ， $F_{CORE}=6\text{MHz}@HIRC$	-	3.5	-	mA
			$V_{DD}=5V$ ， $F_{CORE}=3\text{MHz}@HIRC$	-	2	-	mA
			$V_{DD}=5V$ ， $F_{CORE}=32\text{KHz}@LIRC$ ，HIRC 关，XTAL 关	-	50	-	μA
待机电流 (停止模式)	I_{stop}	V_{DD}	$V_{DD}=5V$ ， $F_{CORE}=24\text{MHz}@HIRC$ ；CORE 停止，LIRC 关，XTAL 关；WDT 关，LVR 开；所有引脚无负载，INTn 唤醒开；其他功	-	1.5	-	mA



特性	符号	引脚	条件	最小	典型	最大	单位
			能模块关				
			VDD=5V, $F_{\text{CORE}}=32\text{KHz}@LIRC$; CORE 停止, HIRC 关, XTAL 关; WDT 关, LVR 开; 所有引脚无负载, INTn 唤醒开; 其他功能模块关	-	70	-	uA
待机电流 (休眠模式)	Isleep	VDD	VDD=5V, CORE 休眠, HIRC/LIRC 关, XTAL 关; WDT 关, LVR 开; 所有引脚无负载, INTn 唤醒开; 其他功能模块关	-	5	-	uA
待机后定时唤醒的平均电流 (休眠模式)	I _{sa}	VDD	VDD=5V, CORE 休眠, HIRC 关, XTAL 关, LIRC 开, T3 的 1S 定时唤醒开; WDT 关, LVR 开; 所有引脚无负载; 其他功能模块关	-	12	-	uA
WDT 电流	I _{wdt}	VDD	VDD=5V, CORE 休眠, HIRC/LIRC 关, XTAL 关; WDT 工作; 其他功能模块关	-	-	3	uA
LVR 电流	I _{lvr}	VDD	VDD=5V, CORE 休眠, HIRC/LIRC 关, XTAL 关; WDT 关, LVR 开; 其他功能模块关	-	-	3	uA
RAM 保持电压	V _{RAM}	VDD	-	-	1.3	-	V

2.4 LVR 特性参数

特性	符号	条件	最小	典型	最大	单位
LVR 设定电压 1	V _{lvr1}	LVR 使能, VDD=2.0~5.5V	-	2.1	-	V
LVR 设定电压 2	V _{lvr2}		-	2.5	-	V
LVR 设定电压 3	V _{lvr3}		-	3.5	-	V
LVR 设定电压 4	V _{lvr4}		-	4.1	-	V
LVR 回滞电压	V _{lvrs}	-	-	100	-	mV

2.5 比较器特性参数

特性	符号	条件	最小	典型	最大	单位
工作电流	I _{COMP}	V _{DDA} =5.0V	-	50	100	μA
工作温度	T _A		-40	25	105	°C
输入失调电压	V _{IO}	T _A =25°C	-	5	10	mV
输入共模电压	V _{ICM}	-	0.1	-	V _{DD} -0.1	V
输入电流	I _{IN}		-	0	1	μA
小信号响应时间	T _{rs1}	VDD=5V, 无迟滞, 正反相输入 10mV 差分变化	-	-	0.5	uS
大信号响应时间	T _{rs2}	VDD=5V, 无迟滞, 正反相输入 100mV 差分变化	-	-	0.3	uS
施密特窗口 1	V _{smt1}	VDD=5V, CnSMT[1:0]=00b, 双边		0		mV
施密特窗口 2	V _{smt2}	VDD=5V, CnSMT[1:0]=01b, 双边		15 注	20	mV
施密特窗口 3	V _{smt3}	VDD=5V, CnSMT[1:0]=10b, 双边		30	40	mV
施密特窗口 4	V _{smt4}	VDD=5V, CnSMT[1:0]=11b, 双边		90	110	mV
内部参考 DAC 分辨率	Nr			6		Bits

注: 表中施密特窗口值为单边窗口值, 实际双边对称设计。

2.6 运算放大器特性参数

特性	符号	条件	最小	典型	最大	单位
工作电压	V _{DDA}	-	V _{DD}	5.0	5.5	V
工作电流	I _{AMP}	V _{DDA} =5.0V	-	0.7	1	mA
工作温度	T _A	-	-40	25	105	°C
输入共模电压范围	V _{IN}	-	V _{SS}	-	V _{DD}	V
输入失调电压	V _{OFFSET}	-	-	±2	±4	mV
输入失调电压随温度变化率	dV _{OFFSET} /dT	-	-	18	25	μV/°C
共模抑制比	CMRR		60	90	-	dB
电压抑制比	PSRR		60	90	-	dB
开环增益	A _{OL}		60	80	-	dB



特性	符号	条件	最小	典型	最大	单位
增益带宽	GBW		2	8	—	MHz
压摆率	SR		4.5	6	—	V/ μ S
电阻性负载	R_{LOAD}		4	—	—	k Ω
电容性负载	C_{LOAD}		—	—	50	pF
驱动电流	I_{LOAD}		—	—	2	mA
高饱和输出电压	VOH_{SAT}	$R_L=4\text{ k}\Omega$, 输入 V_{DDA}	$V_{DDA}-200$	—	—	mV
低饱和输出电压	VOL_{SAT}	$R_L=4\text{ k}\Omega$, 输入 V_{SS}	—	—	$V_{SS}+200$	mV
相位裕度	PM		—	60	—	°
关闭状态到唤醒建立时间	T_{WAKEUP}	$R_L \geq 4\text{ k}\Omega$, $C_L \leq 50\text{pF}$, 电压跟随器	—	2.5	5	μ s
增益精度	P_{GAIN}	$T_A=-40\sim 105^\circ\text{C}$, $V_{DD}=5.0\text{V}$	—	± 0.5	± 1	%
增益值 1	GAIN1	反馈阻值/负端阻值		1		
增益值 2	GAIN2	反馈阻值/负端阻值		2		
增益值 3	GAIN3	反馈阻值/负端阻值		4		
增益值 4	GAIN4	反馈阻值/负端阻值		8		
增益值 5	GAIN5	反馈阻值/负端阻值		16		
增益值 6	GAIN6	反馈阻值/负端阻值		32		
增益值 7	GAIN7	反馈阻值/负端阻值		64		

2.7 ADC 特性参数

下表为分辨率 12bit 参数

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V_{adc}	—	2.5	—	5.5	V
分辨率	Nr	12-bit 模式	—	12	—	bit
		10-bit 模式	—	10	—	bit
		8-bit 模式	—	8	—	bit
		6-bit 模式	—	6	—	bit
转换率	F_s	12-bit 模式	—	—	900	ksps
积分线性误差	INL	$V_{DD}=5\text{V}$, $F_{adc}=1\sim 16\text{MHz}$, 采样时间 $=8\text{Clk}$, 12bit 精度	—	—	± 2	LSB
微分线性误差	DNL	$F_{ADC}=16\text{MHz}$	—	—	± 1	LSB
上限偏置误差	E_{ot}	$F_{ADC}=16\text{MHz}$	—	± 1	± 4	LSB
下限偏置误差	E_{ob}	$F_{ADC}=16\text{MHz}$	—	± 1	± 4	LSB
零点偏移误差	E_z	$F_{ADC}=16\text{MHz}$	—	± 0.5	± 4	LSB
总绝对误差	E_{ad}	$F_{ADC}=16\text{MHz}$	—	± 4	—	LSB
转换时钟	F_{adc}	$V_{DD}=5\text{V}$	—	—	16	MHz
		$V_{DD}=3\text{V}$	—	—	16	MHz
转换时间	T_{con}	$V_{DD}=5\text{V}$, $F_{adc}=16\text{MHz}$, 采样时间 $=3\text{Clk}$, 12-bit 模式	17	—	—	T_{adc}
ADC 输入电压	V_{ain}	—	GND	—	V_{REF}/Gain	V
ADC 输入阻抗	R_{ain}	—	2	—	—	M Ω
ADC 输入电流	I_{ain}	—	—	—	10	μ A
ADC 动态电流	I_{add}	AD 转换中	—	1	1.5	mA
ADC 静态电流	I_{ads}	ADEN 位=0	—	0.1	1	μ A
模拟信号源推荐阻抗	Z_{ain}	—	—	—	10	K Ω
内部参考电压 1	V_{ref}	@ $V_{DD} \geq 2.5\text{V}$	1.59	1.62	1.65	V
启动时间	T_{on}				1.5	μ s
内部参考电压 2	V_{ref}	@ $V_{DD} \geq 2.5\text{V}$, 全温度, EXVRS[1:0]=10b	2.31	2.35	2.39	V
		@ $V_{DD} \geq 2.5\text{V}$, 全温度, EXVRS[1:0]=01b	1.16	1.18	1.20	V
启动时间	T_{on}	4.7 μ F 钽电容+0.1 μ F 瓷片电容@ V_{REF} pin	—	3	—	ms
		0.1 μ F 瓷片电容@ V_{REF} pin	—	100	—	μ s



2.8 温度传感器特性

特性	符号	条件	最小	典型	最大	单位
V_{SENSE} 温度线性度	$T_L^{(注1)}$		-	± 1	± 2	$^{\circ}\text{C}$
平均斜率	$Avg_Slope^{(注1)}$		-	2.82	-	$\text{mV}/^{\circ}\text{C}$
30 $^{\circ}\text{C}$ ($\pm 5^{\circ}\text{C}$) 的电压 ^(注2)	V_{30}		704	751	798	mV
启动时间	$t_{START}^{(注1)}$	Buffer 启动时间	4	-	10 ^(注3)	μs
读取温度时的 ADC 采样时间	$t_{S_temp}^{(注1)}$		4	-	-	μs

注 1: 由设计保证, 未经生产测试。

注 2: 在 $V_{DD} = 3.3\text{ V} \pm 10\text{ mV}$ 测量。V30 ADC 转换结果存储于 TS_CAL1 字节中。请参见下表<3 温度传感器校准值>。

注 3: ADEN 使能后, 延迟 256 μs 后, 才可以使能 TSEN, 再等待 t_{START} 时间后可以 ADC 采样。只要 ADEN 保持开启, 后续的再次使能 TSEN, 只需等待 t_{START} 时间后就可以 ADC 采样。

由于工艺差别, 温度传感器的偏移因芯片而异, 因此未校准的内部温度传感器仅适合检测温度变化的应用。

为提高温度传感器的测量精度, 每个器件都单独出厂校准。温度传感器的出厂校准数据储存在系统存储区, 访问模式为只读。

温度传感器校准值

名称	说明	存储地址
VADC_Temp1	内部温度传感器通道 ADC 原始数据 (常温), $V_{DD}=3.0\text{V} (\pm 10\text{mV})$	1884H - 1885H
TSens_Temp1	获取 ADC_Temp1 的常温温度值, 单位 0.0625 $^{\circ}\text{C}$ 举例, 读取的值 0x190(400), 则实际温度 $400 \times 0.0625^{\circ}\text{C} = 25^{\circ}\text{C}$	1888H - 1889H

2.9 片上 FLASH 特性

特性	符号	条件	最小	典型	最大	单位
Page sizes	Spage			128		bytes
Flash 读取电流	Iread	@24MHz		1.2	1.6	mA
编程电流	Iprog	字节编程			2.8	mA
		Page 编程			3.5	mA
擦除电流	Ierase	Sector 擦除			2	mA
字节编程时间	Tbprog	Byte 编程	6		7.5	μs
擦除时间	Terase	Sector/block 擦除	4		5	ms
	Tcerase	Chip 擦除	20		40	ms
FLASH 读写循环次数	N_{FNDUR}	-	20000	-	-	cycle
FLASH 数据保存时间	T_{PRET}	-	100	-	-	year

注: 片上 FLASH 支持 IAP, 但可以通过 ICP 进行操作, 但当 IAP 操作时芯片的 F_{sys} 必须处于 HIRC 时钟下 (推荐 2MHz 以上, LIRC 时钟或者外部 OSC 时钟下禁止 IAP 操作)。

2.10 交流电气参数

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	F_{HIRC}	$V_{DD}=5\text{V}$, 常温	-1%	48	+1%	MHz
		$T=-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$, $V_{DD}=2.4 \sim 5.5\text{V}$	-2.5%	48	+2.5%	MHz
HIRC 起振时间	T_{HIRC}	$V_{DD}=2.4 \sim 5.5\text{V}$	-	-	50	μs
LIRC 振荡频率	F_{LIRC}	$V_{DD}=5\text{V}$	-5%	32	+5%	KHz
LIRC 起振时间	T_{LIRC}	$V_{DD}=2.0 \sim 5.5\text{V}$	-	-	150	μs
32768Hz 振荡器起振时间	T_{LOSC}	$V_{DD}=2.0 \sim 5.5\text{V}$	-	-	1	s
32768Hz 振荡器匹配电容	C_{LOSC}	$V_{DD}=2.0 \sim 5.5\text{V}$	-	22 ^(注)	-	pF
高频晶体起振时间	T_{HOSC}	$T=-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$, $V_{DD}=2.0 \sim 5.5\text{V}$			1	ms
外部复位脉冲宽度	T_{RST}	-	10	-	-	μs

注: 晶体负载电容请参考所购买的晶体振荡器规格进行匹配。



3 中央处理器

3.1 CPU 概述

CPU 为 1T 标准的高速 8051 内核，其指令完全兼容传统 8051 单片机。

CPU 包含复位初值为 0000H 的程序计数器 PC，包含 1 个专用累加器 A、辅助寄存器 B、堆栈指针 SP 和数据指针 DPTR/DPTR1，以及程序状态寄存器 PSW。

堆栈指针 SP 是 1 个 8 位专用寄存器，保存堆栈栈顶的地址信息。在执行 PUSH、子程序调用、中断响应等指令时，SP 先加 1 再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减 1。堆栈栈顶可以是片上内部 RAM（00H~FFH）的任意地址，系统复位后，SP 初始化为 07H，堆栈实际从 08H 地址开始，用户可自行调整堆栈栈顶地址。

程序状态寄存器 PSW 用来保存指令执行中的各种状态。

数据指针 DPTR/DPTR1 是两组 16 位专用寄存器，其高字节寄存器为 DPH/DPH1 表示，低字节寄存器为 DPL/DPL1 表示。在一组 DPTR/DPTR1 中，它们既可以作为一个 16 位寄存器来处理，也可作为两个独立 8 寄存器来处理。兼容模式下仅使用 DPTR，但用户可通过设定 DPS 寄存器同时使用这两组，关于 DPS 的详细内容，请参考相关章节内容。

3.1.1 程序状态寄存器（PSW, 0xD0/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PSW	CY	AC	FO	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] CY - 进/借位标志位

0: 算术或逻辑运算中，最高位（Bit7）没有进位或借位发生；

1: 算术或逻辑运算中，最高位（Bit7）有进位或借位发生。

BIT[6] AC - 进/借位辅助标志位（可在 BCD 码加减法运算时方便调整）

0: 算术或逻辑运算中，半字节（Bit3）没有辅助进位或借位发生；

1: 算术或逻辑运算中，半字节（Bit3）有辅助进位或借位发生。

BIT[5] FO - 用户自定义标志位

BIT[4:3] RS[1:0] - 工作寄存器组选择位

RS[1:0]	当前使用的工作寄存器组 0~3
00	工作寄存器组 0（00H~07H）
01	工作寄存器组 1（08H~0FH）
10	工作寄存器组 2（10H~17H）
11	工作寄存器组 3（18H~1FH）

BIT[2] OV - 溢出标志位

0: 除下述置位条件，其余情况清零；

1: 累加器 A 中的 bit6 和 bit7 位仅一位发生进位或借位时，置 OV 标志；

即 a) OV 标志表示有符号的 8 位数字加减法的结果超出了限制（大于 127 或小于 -128）。

b) 当乘法结果大于 255，或 c) 试图除以 0 时，也会置 OV 标志。

BIT[1] F1 - 用户自定义标志位

BIT[0] P - 奇偶校验位

0: 表示累加器 A 中的二进制结果中有 1 的个数为偶数个；

1: 表示累加器 A 中的二进制结果中有 1 的个数为奇数个。

3.2 程序存储器（FLASH ROM）

程序存储器 FLASH ROM 总的大小为 18K 字节，地址范围为 0000H~47FFH，该 FLASH ROM 可反复擦写 2 万次，数据保存年限至少 10 年。

FLASH ROM 通常需要先擦除后写入。本芯片的 FLASH ROM 支持在电路编程（ICP）和在应用编程（IAP）。

注：ICP 和 IAP 相关内容，请参考相关章节的内容。

3.3 附加数据区（EEPROM-LIKE）

本芯片内置 128bytes 附加数据区，可作为 EEPROM-LIKE 使用，支持页擦除/字节烧写。

本芯片的 EEPROM-LIKE 支持在电路编程（ICP）和在应用编程（IAP），也可通过 MOVX 指令进行寻址。

注：ICP 和 IAP 相关内容请参考相关章节的内容。

注：使用烧写下载工具执行芯片全擦时，此 EEPROM-LIKE 区域会被擦除。

3.4 随机存储器（RAM）

随机数据存储器 RAM 包括内部 256 字节和外部 2048 字节共 2304 字节的 SRAM。

内部 RAM 的地址为 00H~FFH，其中高 128 字节（80H~FFH）与特殊功能寄存器 SFR 重叠，且只能间接寻址，低 128 字



节（00H~7FH）可直接寻址也可以间接寻址。其中最低的 32 字节（00H~17H）作为通用寄存器分成四组 8 个寄存器，程序视这些寄存器为 R0 到 R7，具体通过程序状态字 PSW 的 RS[1:0] 位来选择哪一个寄存器组会被使用；接下来的 16 个字节（20H~2FH）还支持位寻址；剩下的字节（30H~7FH）作为通用 RAM，可用来设置栈指针，用作“栈”使用。

外部 RAM 的地址为 0000H~07FFH，这部分空间必须通过 MOVX 指令来寻址。

特殊功能寄存器 SFR 地址为 80H~FFH，只能直接寻址。CPU 自动通过指令区分是访问 SFR 还是内部 RAM 的高 128 字节；SFR 通过 DPS 的 SFRSL0 位切换，拓展了 SFR 的范围。

注 1：本芯片不支持 P0 和 P2 口的外部 RAM 扩展功能。

注 2：PSW 的 RS[1:0] 相关内容请参考相关章节的内容。

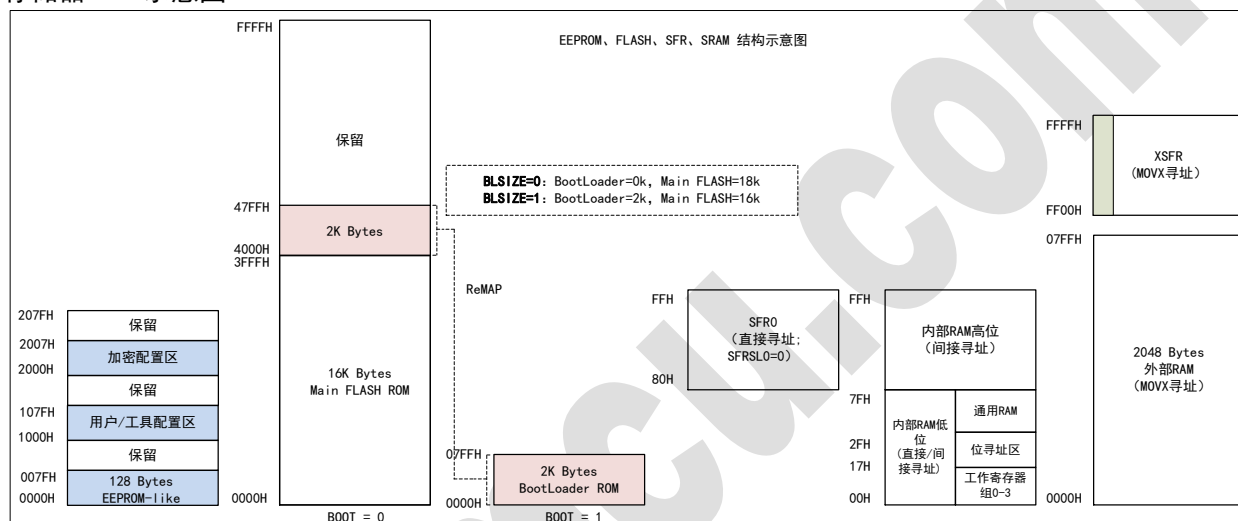
注 3：DPS 的 SFRSL0 相关内容请参考相关章节的内容。

3.5 存储器结构

本芯片的程序存储器 FLASH、随机存储器 RAM 以及非易失性数据存储器 EEPROM 分别是独立编址的。由于没有提供外部存储器扩展的总线接口，所以并不能像传统的 8051 通过 P2 端口进行外扩。

本芯片片上存储器的结构，如下图所示：

存储器 MAP 示意图



该芯片的 FLASH 空间由 8 位宽的存储单元组成，即可以保存代码又可以保存数据。当前 FLASH 有 18 x 8 x 128 页组成（每页 128 Bytes）或 4 扇区（每个扇区 4KB，即 4 页）组成。在 FLASH “分区保护” 的时候，以扇区为操作单位，具体在“用户配置字”中设定，请参考相关章节的内容。

ROM 空间操作汇总表

操作形式	ICP	IAP	ISP
FLASH 代码保护	支持	不支持	支持
FLASH 页擦除	支持	不支持	支持
FLASH 块擦除	支持（不保护的情况下）	不支持	支持（不保护的情况下）
FLASH 整体擦除	支持	不支持	不支持
FLASH 读/写	支持（不保护的情况下）	不支持	支持（不保护的情况下）
EEPROM-LIKE 擦除	支持	支持	支持
EEPROM-LIKE 读/写	支持	支持	支持

3.6 寻址方式

CPU 指令的寻址方式有以下 7 种：立即寻址、直接寻址、间接寻址、寄存器寻址、相对寻址、变址寻址、位寻址。

立即寻址

立即寻址也称为立即数，它是在指令操作数中直接给出参加运算的操作数。

直接寻址

在直接寻址方式中，指令操作数域给出的是参加运算操作数的地址。直接寻址方式只能用来表示特殊功能寄存器、内部数据寄存器和位地址空间。其中特殊功能寄存器和位地址空间只能用直接寻址方式访问。

间接寻址

间接寻址采用 R0 或 R1 前加“@”符号来表示。

寄存器寻址

寄存器寻址寻址时对选定的工作寄存器 R7~R0、累加器 A、通用寄存器 B、地址寄存器和进位标志 CY 中的数进行操作。



作。其中寄存器 R7~R0 由指令码的低 3 位表示，A、B、DPTR 及进位标志 CY 隐含在指令码中，因此寄存器寻址也包含一种隐含寻址方式。寄存器工作区的选择由程序状态寄存器 PSW 中的 RS[1:0]来决定。指令操作数指定的寄存器均指当前工作区的寄存器。

相对寻址

相对寻址是将程序计数器 PC 中的当前值与指令第二字节给出的数相加，其结果作为转移指令的转移地址。转移地址也成为转移目的地址，PC 中的当前值成为基地址，指令第二字节给出的数成为偏移量。由于目的地址是相对于 PC 中的基地址而言，所以这种寻址方式称为相对寻址。偏移量为带符号的数，所能表示的范围为 (+127) ~ (-128)。这种寻址方式主要用于转移指令。

变址寻址

在变址寻址方式中，指令操作数指定一个存放变址基址的变址寄存器。变址寻址时，偏移量与变址基值相加，其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

语句“MOV A, @A+DPTR”表示累加器 A 为偏移量寄存器，其内容与地址寄存器 DPTR 中的内容相加，其结果作为操作数的地址，取出该单元中的数送入累加器 A 中。

位寻址

位寻址是指对一些可进行位操作的内部数据存储器 RAM 和特殊功能寄存器进行位操作时的寻址方式。在进行位操作时，借助于进位标志 CY 作为位操作累加器，指令操作数直接给出该位的地址，然后根据操作码的性质对该位进行位操作。位地址与字节直接寻址中的字节地址编码方式完全一样，主要由操作指令的性质加以区分，使用时应特别注意。

3.7 特殊功能寄存器映射

特殊功能寄存器 SFR 的地址位于 80H~FFH，有些可以位寻址，有些不能位寻址。能够进行位寻址操作的寄存器的地址末位数都是“0”或“8”，这些寄存器在需要改变单个位的数值时非常方便。所有的特殊功能寄存器只能通过直接寻址方式进行寻址。

特殊功能寄存器列表 (SFR0)

起始地址	可位寻址	不可位寻址							SFR 页
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H	PWRCR	PWRCR1	ADASCRO	ADASCR1	ADASCR2	ADCCR2	ADRL	ADRH	SFR0
F0H	B	AUXC	ADCCRO	ADCCR1	ADCCR3	ADCCR4	OSCSS	XPAGE	SFR0
E8H	RSTFLAG	IAPCR	IAPADL	IAPADH	IAPDB	IAPCR1	IAPMD	IAPKEY	SFR0
E0H	A	MDUCR	MDUDRA0	MDUDRA1	MDUDRA2	MDUDRA3	MDUDRB0	MDUDRB1	SFR0
D8H	DPS	PWM1RLN	PWM1ANDL	PWM1ANDH	PWM1BNDL	PWM1BNDH	PWM1CNDL	PWM1CNDH	SFR0
D0H	PSW	PWM1OE	PWM1ADL	PWM1ADH	PWM1BDL	PWM1BDH	PWM1CDL	PWM1CDH	SFR0
C8H	T2CON	T2MOD	RCAPL2	RCAPH2	TL2	TH2	T2CON1	WDTCR	SFR0
C0H	OSCM	POMODO	POMOD1	POPU	CMP1CRO	CMP1CR1	CMP1CR2	CMP1CR3	SFR0
B8H	PWM1IF	IP	IPH	IP1	IPH1	PWM1CR1	PWM1CR2	PWM1FCR	SFR0
B0H	P3	P3MOD0	P3MOD01	P3PU	EXTEN1	EXTEN	PWM1MANCRO	PWM1MANCR1	SFR0
A8H	IE	IE1	EXTIS2	EXTIS1	EXTIS	SPCON	SPSTA	SPDAT	SFR0
A0H	P2	P2MOD0	P2MOD1	P2PU	CMPOCRO	CMPOCR1	CMPOCR2	CMPOCR3	SFR0
98H	SCON	SBUF	SADDR	SADEN	SCON_1	SBUF_1	BRGCR	PXODM	SFR0
90H	P1	P1MOD0	P1MOD1	P1PU	BRGL	BRGH	TL3	TH3	SFR0
88H	TCON	TMOD	TL0	TL1	TH0	TH1	TCON1	T3CR	SFR0
80H	P0	SP	DPL	DPH	DPL1	DPH1	SLPCR	PCON	SFR0
起始地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	SFR 页
可位寻址	不可位寻址								

扩展特殊功能寄存器列表 (XSFR)

起始地址	偏移地址								SFR 页
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
FFF8H~FFC0H	-	-	-	-	-	-	-	-	XSFR
FFB8H	-	-	-	-	-	-	-	-	XSFR
FFB0H	CMPOO_MAP	-	CMP1O_MAP	-	-	-	-	-	XSFR
FFA8H	PWMO_MAP	-	-	-	-	-	-	-	XSFR
FFA0H	PWM1A_MAP	PWM1AN_MAP	PWM1B_MAP	PWM1BN_MAP	PWM1C_MAP	PWM1CN_MAP	PWM1FB_MAP	-	XSFR
FF98H	nSS_MAP	SCK_MAP	MISO_MAP	MOSI_MAP	-	-	-	-	XSFR
FF90H	-	-	-	-	-	-	-	-	XSFR
FF88H	TXO_MAP	RXO_MAP	-	-	TX1_MAP	RX1_MAP	-	-	XSFR
FF80H	TO_MAP	T1_MAP	T2_MAP	-	-	-	-	T2EX_MAP	XSFR
FF78H	-	-	-	-	-	-	-	-	XSFR
FF70H	OPAOCRO	OPAOCR1	-	-	-	-	-	-	XSFR



起始地址	偏移地址								SFR 页
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
FF68H	PWMOCR	PWMOP	PWMOD	-	-	-	-	-	XSFR
FF60H	CRCCR	-	CRCH	CRCL	CRCSTH	CRCSTL	CRCEDH	CRCEDL	XSFR
FF58H	-	-	-	-	-	-	-	-	XSFR
FF50H	-	-	-	-	-	-	-	-	XSFR
FF48H	PWM1CRO	PWM1IE	PWM1PH	PWM1PL	-	PWM1DTOL	-	PWM1DT1L	XSFR
FF40H	ADCMPLGH	ADCMPLG	ADCMPLH	ADCMPLL	ADCCALOS	ADSEQRO	ADSEQR1	ADSEQR2	XSFR
FF38H~FF00H	-	-	-	-	-	-	-	-	XSFR
起始地址	偏移地址								SFR 页
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

3.8 SFR 的页切换与硬件乘除法器

芯片内部扩展了一页 SFR，即新增的 SFR1 页。SFR1 页用于存放芯片上新增外设的寄存器。默认情况下的 SFR 页与 51 系列保持兼容，并将其编号为 SFR0 页。除去可位寻址的 16 个 SFR 始终落在 SFR0 页上面，其余的 SFR，在 SFR1 页与 SFR0 页上位置是重叠的，它们通过 DPS 中的 SFRSL0 位进行区分。上电复位后，SFR0 页有效。

同时，芯片内部的扩展了一个新的 AUXC 寄存器，满足了 16 位乘法和 16 位除法。当用户需要完成 16 位*8 位或者 16 位/8 位运算时，通过增强型指令“MUL”和“DIV”可以一次完成这个运算。上电复位后，该功能关闭，仅兼容 51 系列的 8 位*8 位或者 8 位/8 位运算。

3.8.1 数据指针选择寄存器（DPS, 0xD8/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPS	-	SFRSL0	-	-	DIV16	MUL16	-	SEL
R/W	-	R/W	-	-	R/W	R/W	-	R/W
复位值	-	0	-	-	0	0	-	0

BIT[6] SFRSL0 - SFR 页切换低位

SFRSL0	对应操作的 SFR 页面	SFR 页号
0	标准 SFR（默认）	SFR0
1	扩展第一页 SFR（保留）	SFR1

注：本芯片 SFR1 页保留，用户不可操作

BIT[3] DIV16 - 16 位硬件除法使能位

0：使用 8 位硬件除法器；

1：使用 16 位硬件除法器。

BIT[2] MUL16 - 16 位硬件乘法使能位

0：使用 8 位硬件乘法器；

1：使用 16 位硬件乘法器。

寄存器位	对应操作	计算结果		
		AUXC	B	A
MUL16 位	MUL16=0: 8 位乘法	(A) * (B)	-	积高位字节 积低位字节
	MUL16=1: 16 位乘法	(AUXC A) * (B)	积高位字节 积中位字节	积低位字节
DIV16 位	DIV16=0: 8 位除法	(A) / (B)	-	余数 商低位字节
	DIV16=1: 16 位除法	(AUXC A) / (B)	商高位字节 余数	商低位字节

3.9 用户配置字

芯片内部包含一块额外的 FLASH 空间用于保存芯片的配置信息。芯片为保证系统正常工作，会将关键模块的配置预先存储于此区域，在上电或其他复位发生后会将配置信息载入寄存器中，通过寄存器确定关键模块的初始工作状态。这部分配置信息，即为用户配置字。

注：“用户配置字”在芯片正常操作时不允许修改，否则芯片会出现不可预知的情况。

用户配置字详细说明

符号	功能说明
BOOTS	BOOTS - 启动 ROM 区域选择 0: 复位后 MCU 从 main flash rom 区 0000H 启动; 1: 复位后 MCU 从 bootloader rom 区 0000H 启动; 注: 若 BLSIZE=0, 选择 BOOTS=1 无效, 复位后依旧从 main flash rom 区 0000H 启动 注: 除软复位 (SWRST) 外, 其他复位此位内容加载至 IAPCR1->BOOT (BIT7)。



符号	功能说明
BLSIZE	Bootloader Rom 区空间大小选择 0: BootLoader=0k, Main FLASH=18k 1: BootLoader=2k, Main FLASH=16k 注: 若设置为 BootLoader=2k (BLSIZE=1), 烧写模式下 (ICP) 仍可以通过访问第 17/18k 空间 (4000H~47FFH) 访问。
FMOD[3:0]	芯片工作模式选择位: 0: 双时钟运行模式: HIRC+LIRC 1: 双时钟运行模式: HIRC+LOSC, LOSC @OSCI/OSCO 2: 保留 3: 双时钟运行模式: HOSC+LIRC, HOSC @OSCI/OSCO 4: 保留 5: 保留 注 1: 双时钟运行模式, Core 可根据软件的设置工作在高速或低速运行模式, 控制位 CLKS 可调; 6: 单时钟高速运行模式: HIRC 7: 单时钟高速运行模式: HOSC, HOSC @OSCI/OSCO 8: 保留 注 2: 单时钟 HOSC 模式下, LOSC 被常关闭; 9: 单时钟低速运行模式: LIRC 注 3: 单时钟 LIRC 模式下, HIRC、HOSC 和 LOSC 将被常关闭; ; 10: 单时钟低速运行模式: LOSC, LOSC @OSCI/OSCO 11: 保留 其他: 双时钟运行模式: HIRC+LIRC 注 4: 单时钟 LOSC 模式下, HIRC 和 HOSC 将被常关闭; 注 5: 当 OSCI/OSCO 被选择有效时, 管脚其他复用功能无效, 请务必外接晶体振荡器及其振荡电路, 否则影响芯片的稳定性。当未被选择时, 引脚用作 GPIO 和其他复用功能。 注 6: 当选择了“单时钟低速运行模式”时, Core 无法切换到高速运行模式, 且芯片上 HIRC/HOSC 时钟源及其相关的功能均不可用。
FSYS_FILS[2:0]	系统时钟滤波选择位: 000: 7.5 ns (Fsys 最大频率 24MHz) 001: 15 ns (Fsys 最大频率 16MHz) 010: 30 ns (Fsys 最大频率 8MHz) 011: 60 ns (Fsys 最大频率 4MHz) 100: 80 ns (Fsys 最大频率 2MHz) 其他: 保留 注 1: 此滤波可以提高晶振电路抗干扰能力。
EXOSC_WUT[1:0]	外部高频晶振预热时间选择位: 00: 最长预热时间 01: 长预热时间 10: 短预热时间 11: 最短预热时间 注: 此位仅在 FMOD[3:0] 选择外部高频晶振 (FHOSC) 时, 有效
WDTM[1:0]	WDT 工作模式选择位: 0: 始终关闭 WDT: “看门狗”功能无效; 1: 自动关闭 WDT: 在停止模式/休眠模式下, 自动关闭“看门狗”; 2: 始终开启 WDT: 在停止模式/休眠模式下, “看门狗”继续工作, 故需要手动喂狗。 注: 当本位选择了“始终开启‘看门狗’”时, 请务必按时手动喂狗, 否则将会产生“WDT 复位”。
RSTEN	允许外部复位操作位: 0: 外部复位功能无效: 对应的 nRST 口可用于 GPIO 或者其他复用功能, 该 PIN 不可用于外部复位; 1: 外部复位功能有效: nRST 口自动内部带上拉, 低电平时复位有效, 该 PIN 仅可用于外部复位, 其他功能无效。 注: 当本位选择了“外部复位功能无效”时, 本芯片将无外部复位功能可用。
LVRSLP	休眠模式下 LVR 功能的选择位: 0: 休眠模式下 LVR 无效: LVR 功能将在 Core 进入休眠模式下, 自动关闭; 1: 休眠模式下 LVR 有效: LVR 功能始终有效; 注: “停止模式”下, LVR 不可控, 将始终有效。



符号	功能说明
VLVRS	LVR 电压等级选择位： 0: 2.1V; 1: 2.5V; 2: 3.5V; 3: 4.1V; <i>注：本位与 LVRSLP 位联动，当 LVRSLP 为“停止模式/休眠模式下 LVR 无效”时，则 LVR 功能将在停止模式/休眠模式下，自动关闭。</i>
OCDEN	OCD 调试端口运行状态使能： 0: 芯片运行状态，关闭 OCD 调试端口； 1: 芯片运行状态，使能 OCD 调试端口，OCD 组由 OCDSEL 决定。
OCDSEL	运行状态 OCD 组选择： 0: 使能 OCD0 调试口 (DIO0/CLK0)，同时关闭 OCD1 调试口（默认）； 1: 使能 OCD0 调试口 (DIO1/CLK1)，同时关闭 OCD0 调试口。 <i>注：此位只在 OCDEN=1 时有效。</i> <i>注：OCDEN 和 OCDSEL 控制位，不影响上电或复位的 OCD 访问控制；仅控制芯片运行状态是否打开内部 OCD 通路，除此之外，还要配置对应的 OCD 组端口为输入状态，内部 OCD 控制器才能收到控制时序。</i>
SCnP (n=0~4)	FLASH 分区保护控制位： 0: n 分区加密块不加密：对应的 n 分区加密块无限制，FLASH 内容可正常读出； 1: n 分区加密块一级加密：对应的 n 分区一级加密限制，FLASH 内容禁止外部调试器/编程器读出，但允许内部 MOVN 指令在“脱机”情况下的读出、“调试”情况下无效； 2: n 分区加密块二级加密：对应的 n 分区二级加密限制，FLASH 内容禁止外部调试器/编程器读出，同时禁止内部 MOVN 指令任何情况下的读出； <i>注：该芯片中，每 4K 字节大小设定为一个分区加密块，根据 FLASH ROM 空间地址从低字节到高依次递增排列，即 SC0P 对应 0x0000~0x0FFF 区域，SC1P 对应 0x1000~0x1FFF 区域，SC2P 对应 0x2000~0x2FFF 区域，SC3P 对应 0x3000~0x3FFF 区域，SC4P 对应 0x4000~0x47FF 区域（2K）。</i>



4 系统时钟与工作模式

芯片支持 4 种时钟源：

- ◇ 内部高频 RC 时钟 FHIRC 48MHz
- ◇ 内部低频 RC 时钟 FLIRC 32kHz
- ◇ 外部高频晶体时钟 FHOSC 4~24MHz
- ◇ 外部低频晶体时钟 FLOSC 32.768KHz

支持 2 种时钟被选择为系统时钟 Fsys：

- ◇ 高频时钟 FHCLK，由 FHOSC 或 FHIRC 二分频时钟产生
- ◇ 低频时钟 FLCLK，由 FLIRC 或 FLOSC 产生

支持 1 组 GPIO 作为外部晶体时钟接口，并支持多种双时钟或单时钟模式选择。

用户配置字 EXOSC_DRV，选择高频晶体驱动能力等级，仅在 FMODE[3:0] 选择外部高频晶振（FHOSC）时有效。

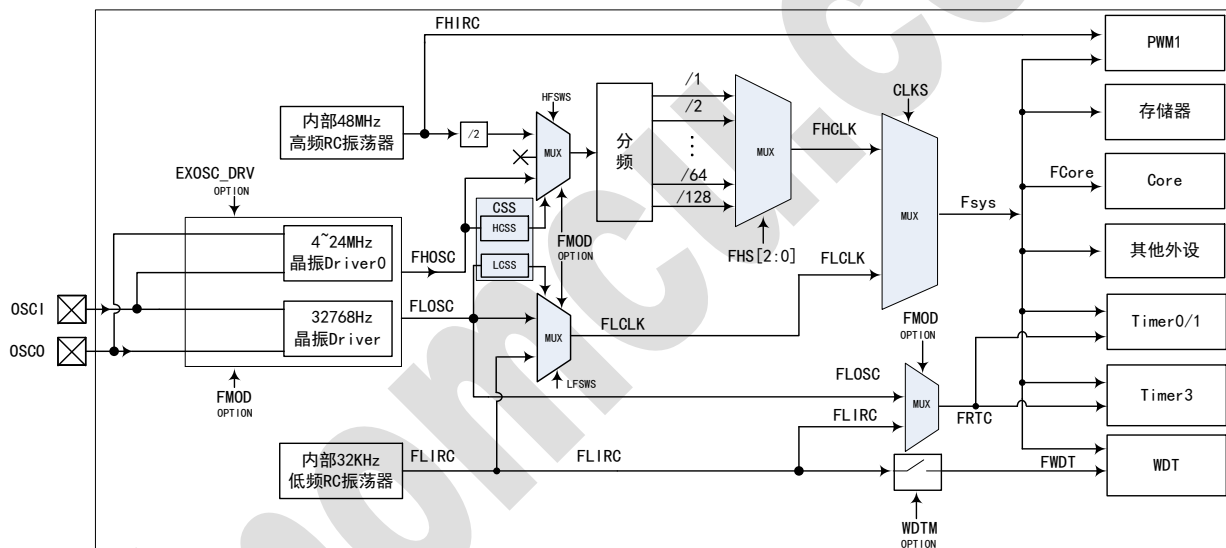
用户配置字 FMODE，选择双时钟及单时钟模式下的时钟匹配及外部晶体 pin 脚匹配。

用户配置字 FMODE，还控制 Timer3 方式 1 和 Timer0/1 的低频时钟源，若 FMODE 选择时钟包含 FLOSC，则选择 FLOSC 作为 Timer3/0/1 时钟源，否则选择 FLIRC 作为时钟源。

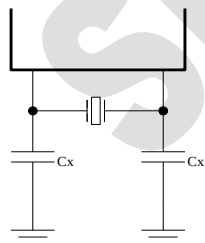
FHS 控制位，HIRC/2 或 HOSC 二选后时钟分频选择。

4.1 系统时钟框图

系统时钟示意图



4.2 外接晶体振荡器



使用外部晶体作为内部模块的低频时钟源时，连接方式见左图。

用户配置字 FMODE，选择高频体振荡器或低频振荡器有效及 PIN 脚选择。

用户配置字 EXOSC_DRV，选择高频晶体驱动能力等级。

低频一般是接 32768Hz 晶体，通常 Cx 是必须的。

高频支持 4~24MHz 晶体，电容根据实际使用的晶体进行匹配。用户可通过用户配置字 FHS 配置成 16/8/2/1 分频用于系统高频时钟 FLCK。

在实际使用中，用户应使晶体离 OSCI/OSCO 或 OSCOX/OSCIX 引脚的距离尽可能短，这样有助于振荡器的起振和振荡的稳定性。

下表列出典型频率 32.768KHz 晶振选用电容 Cx 的推荐值和相应最低起振电压参考值。

晶体频率 (Hz)	电容 Cx (pF)	最低起振电压 (V)
32768	10	2.0

注：因为晶体的品牌很多，电容值仅为推荐值，起振电压仅供参考，具体参数请根据实际使用的晶振性能而定。

4.3 内置 48MHz 高频 RC 振荡器

芯片内置高精度 48MHz RC 振荡器 HIRC，2 分频后，用户可通过寄存器 OSCSS 的 FHS 位配置成 128/64/32/16/8/4/2/1 分频用于系统高频时钟 FHCK。



HIRC 也可以直接提供 PWM1 时钟，用户可配置相关控制位选择 Fsys 或 FHIRC。

内置的高频 RC 振荡器，可在芯片内部进行调校，出厂时即已调校至 48MHz@5V/25℃，用户使用时无需重新调校。

4.4 内置 32KHz 低频 RC 振荡器

芯片内置一个低频 RC 振荡器 LIRC，该振荡器可用于系统低频时钟 FLCK。该振荡器频率典型值为 32KHz。

FLIRC 作为 WDT 固定工作时钟。

通过用户配置字 FMOD，可配置 FLIRC 为 timer3 方式 1（时基模式）的低频时钟。若 FMOD 选择时钟包含 FLOSC，则选择 FLOSC 作为 Timer3 时钟源，否则选择 FLIRC 作为时钟源。

4.5 系统时钟相关寄存器

系统时钟寄存器汇总表

地址 (SFRn /H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
SFR0/ F9	PWRCR1	高级能耗控制 寄存器 1	-	-	CRC_CLK EN	-	-	T3_CLKE N	T2_CLKE N	TO1_CLK EN	1111 1111
SFR0/ F8	PWRCR	高级能耗控制 寄存器	-	-	PWM1_CL KEN	PWM0_CL KEN	SPI_CLK EN	ADC_CLK EN	LCD_CLK EN	OCD_CLK EN	1111 1111
SFR0/ C0	OSCM	振荡模式寄存 器	XTSPD	STBHXT	STBH	STBXT	-	CLKS	LFEN	HFEN	1000 ~x1x
SFR0/ 87	PCON	电源控制寄存 器	UART1EN	-	-	UART0EN	SMOD0	SSTAT0	SLEEP	STOP	0—0 0000
SFR0/ 86	SLPCR	模式保护控制 寄存器	SLPCR7	SLPCR6	SLPCR5	SLPCR4	SLPCR3	SLPCR2	SLPCR1	SLPCR0	00000 000
SFR0/ F6	OSCSS	时钟安全寄存 器	CSSSEN	FHS2	FHS1	FHS0	CSSIE	HFSWS	LFSWS	CSSF	1010 0000

4.6 时钟安全检测 CSS

芯片内置时钟安全检测功能，以提高系统可靠性。当检测到外部晶体振荡器（HOSC 或 LOSC）发生异常（如，晶体振荡器停振等），对应的中断标志 CSSF 置位，若开启中断使能，则发生中断。

若 HOSC 外部振荡器配置为系统时钟，当检测到振荡器异常，硬件自动切换系统时钟为 HIRC，HFSWS 置位。若用户检测到 HOSC 时钟恢复正常，可清零 HFSWS，切换时钟为 HOSC。

若 LOSC 外部振荡器配置为系统时钟，当检测到振荡器异常，硬件自动切换系统时钟为 LIRC，LFSWS 置位。若用户检测到 LOSC 时钟恢复正常，可清零 LFSWS，切换时钟为 LOSC。

用户配置字 FMOD，选择高频振荡器或低频振荡器有效。CSSSEN 位控制相应的 CSS 检测模块开启。

寄存器 OSCM 的 STBHXT 位，反映外部晶体 HOSC 的稳定状态，STBXT 位，反映外部晶体 LOSC 的稳定状态，用户通过读取此 2 位，可以判断晶体稳定或异常状态。

CSSF 标志位在振荡器发生时异常置位，软件清零，若中断使能控制 CSS_CRCIE 和 CSSIE 同时使能，则发生中断；CSS 与 CRC 复用同一个中断入口。

CSS 功能默认使能，为防止晶体启动时，异常误判引起系统时钟切换，内部设计为有效的稳定预热时间后，启动 CSS。

FHS[2:0] 高频时钟分频控制，1/2/4.../64/128 分频可选。

4.6.1 时钟安全寄存器（OSCSS, 0xF6/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCSS	CSSSEN	FHS2	FHS1	FHS0	CSSIE	HFSWS	LFSWS	CSSF
R/W	R/W	R/W	R/W	R/W	R/W	R/W*	R/W*	R/W*
复位值	1	0	1	0	0	0	0	0

BIT[7] CSSSEN - 时钟安全检测模块使能，默认开启

0: 时钟安全检测模块关闭；

1: 时钟安全检测模块开启。

BIT[6:4] FHS[2:0] - 高频时钟分频控制

000: 1 分频；

001: 2 分频

010: 4 分频（默认）

011: 8 分频

100: 16 分频；



- 101: 32 分频。
110: 64 分频。
111: 128 分频。
- BIT[3] **CSSIE** - 时钟安全中断使能
0: 关闭;
1: 开启, IE1 寄存器的 CSS_CRCIE 同时置 1, 中断使能才生效。
注: 此控制位与 IE1 的 CSS_CRCIE 联动控制, 若要使能 CSS 中断, 必须同时置位 CSS_CRCIE 和 CSSIE; 若要禁止 CSS 中断, 仅需清零 CSS_CRCIE 和 CSSIE 任意一位。CSS 和 CRC 复用同一个中断入口。
- BIT[2] **HFSWS** - 高频晶体停振切换恢复, 硬件置 1, 软件清零
0: 软件清零, 高频时钟源切换为 HOSC 时钟;
1: HOSC 发生停振事件, HCLK 时钟源切换为 HIRC。
- BIT[1] **LFSWS** - 低频晶体停振切换恢复, 硬件置 1, 软件清零
0: 软件清零, 低频时钟源切换为 LOSC 时钟;
1: LOSC 发生停振事件, LCLK 时钟源切换为 LIRC。
注: 若 HOSC 或 LOSC 外部振荡器配置为系统时钟, 当检测到振荡器异常, 硬件自动切换系统时钟为 HIRC 或 LIRC, HFSWS 或 LFSWS 硬件置 1, 用户可以通过清零此 2 位, 恢复系统时钟为 HOSC 或 LOSC。
- BIT[0] **CSSE** - 时钟安全检测异常中断标志, 硬件置 1, 软件清零
0: 时钟安全检测异常标志未发生;
1: 时钟安全检测异常标志发生。

4.7 振荡器预热和复位预热

振荡器时钟预热

振荡器类型	高频晶体振荡器 HOSC 4~24MHz	低频晶体振荡器 LOSC 32.768KHz	内部 RC 高频振荡器 HIRC 48MHz	内部 RC 低频振荡器 LIRC 32KHz
预热时钟数	00: 2^{18} clk 01: 2^{16} clk 10: 2^{13} clk 11: 2^{11} clk 注	2^{14} clk	2^7 clk	2^3 clk

注: 高频晶体时钟预热, 对应用户配置字 EXOSC_WUT, 可以由用户选择 4 档预热时间。

复位预热

复位类型	上电/外部复位	LVR/WDT/SWRST
预热时间	10ms+振荡器预热	≈振荡器预热

芯片上电后, 先进行上电预热, 再进行振荡器预热, 完成后运行程序。

4.8 工作模式

CORE 支持高速运行模式、低速运行模式、停止模式、休眠模式共 4 种工作模式。

工作模式	模式描述	功耗
高速运行模式	CORE 时钟运行在高频时钟 (FHLCK) 上, 所有外设均可正常工作。	高
低速运行模式	CORE 时钟运行在低频时钟 (FLCLK) 上, 大部分外设可正常工作。	中
停止模式	CORE 时钟停止 (FHLCK 时钟源可选关闭), 程序计数器 (PC) 挂起, 但所有外设可继续正常工作, 用户可随时通过各种外设产生的中断快速唤醒 CORE。	低
休眠模式	CORE 时钟停止 (FHLCK 时钟源关闭、FLCLK 时钟源可选关闭), 程序计数器 (PC) 挂起, 仅有限的外设可以继续工作, 用户仅可通过这部分外设产生的中断唤醒 CORE。	最低

4.8.1 工作模式与时钟切换

通过用户配置字可配置运行模式为下面列表中 7 种之一, 通过软件可配置进入停止模式和休眠模式。

芯片上电或复位默认时钟为 HIRC, 执行配置字加载后切换为用户预设时钟。

振荡器	(双时钟) HIRC+LIRC (CLKS=0/1)	(双时钟) HIRC+LOSC (CLKS=0/1)	(双时钟) HOSC+LIRC (CLKS=0/1)	-	停止模式 (当 STOP=1)
FHIRC	ON	ON	OFF, 注 5	-	HFEN 决定, 注 2
FLIRC	ON	ON, 注 5	ON	-	ON
FHOSC	OFF	OFF	ON	-	HFEN 决定, 注 2



FLOSC	OFF	ON	OFF	-	
振荡器	(单时钟) HIRC (CLKS=0)	(单时钟) HORC (CLKS=0)	(单时钟) LIRC (CLKS 恒=1)	(单时钟) LOSC (CLKS 恒=1)	休眠模式 (当 SLEEP=1)
FHIRC	ON	OFF, 注 5	OFF	OFF	关闭, 注 3
FLIRC	ON, 注 5	ON, 注 5	ON	ON, 注 5	LFEN 和 WDTM 决定, 注 4
FHOSC	OFF	ON	OFF	OFF	关闭, 注 3
FLOSC	OFF	OFF	OFF	ON	LFEN 决定, 注 4

注 1: 为了尽量降低功耗, 内部高频振荡器 (HIRC 和 HOSC) 在某些场合下必须“停止”工作。单时钟低速运行用户配置字“FMOD”选择工作模式; 当进入“单时钟运行模式”后, 系统时钟固定, 即使用户强行切换时钟 (CLKS=0 或 1), 切换也会不成功;

注 2: 当进入“停止模式”后, 若用户之前选择了“双时钟运行模式”, 可以通过进模式前切换为 LCLK (LIRC 或 LOSC), 然后关闭 HIRC 和 HOSC (设置 HFEN=0), 进一步降低功耗。

注 3: 在进入“休眠模式”前, 即使用户之前选择了“单时钟高速运行模式”, 且用户选择了不关闭 HCLK (设置 HFEN=1), 一旦进入“休眠模式”后, HCLK (HIRC 和 HOSC) 也将自动停止工作。

注 4: 当进入“休眠模式”后, 用户可以通过关闭 LCLK (设置 LFEN=0), 进一步降低功耗。特别注意, 当用户进入休眠模式时关闭 LCLK, T3 不可用来唤醒“休眠模式”下的芯片; T3 需要使用 LIRC 时钟源, 才可以唤醒“休眠模式”下的芯片。但 INTO/1 任何情况下, 都可以直接用来唤醒“休眠模式”下的芯片, 并再次启动 LIRC。只不过这样的唤醒时间会比开启 LIRC 进入“休眠模式”的时间长, 因为芯片多了一个 LIRC 起振和稳定的时间。

注 5: 双时钟运行模式下, LIRC 保持常开, 保证 WDT 工作时钟和 LOSC 停振切换时钟;

单时钟运行模式下, 若用户配置字 WDTM 开启 WDT, LIRC 保持常开;

单时钟模式 (LOSC), LIRC 保持关闭, 当发生停振事件, 硬件开启 LIRC 并切换至 LIRC, 切回 LOSC 后硬件关闭 LIRC;

单时钟模式 (HOSC) 和单时钟模式 (HIRC), 则 LIRC 保持关闭

运行模式下 HOSC 被选择, HIRC 保持关闭, 当发生停振事件, 硬件开启 HIRC 并切换至 HIRC, 切回 HOSC 后硬件关闭 HIRC。

4.8.2 振荡模式寄存器 (OSCM, 0xC0/SFRO)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	XTSPD	STBHXT	STBH	STBXT	-	CLKS	LFEN	HFEN
R/W	R/W	R	R	R	-	R/W	R/W	R/W
复位值	1	0	0	0	-	X	1	X

BIT[7] XTSPD - 外部 32768Hz 振荡器加速模式控制位

0: 32768Hz 振荡器常规模式, 由软件清 0;

1: 32768Hz 振荡器加速模式, 由软件或硬件置 1。

注: 当使用 32768Hz 振荡器时, 此位在系统发生复位后, 由硬件自动置 1, 以加速 32768Hz 振荡器的起振时间; 如有需要, 此位也可由软件置 1 以便在模式切换时加速外部振荡器的起振, 或软件清 0 以降低功耗。

BIT[6] STBHXT - 外部晶体高频振荡器稳定标志位

0: HOSC 高频振荡器停振或未稳定, 由硬件清 0, 可读但写入无效;

1: HOSC 高频振荡器已稳定运行, 由硬件置 1, 可读但写入无效。

BIT[5] STBH - 内部 RC 高频振荡器稳定标志位

0: HIRC 高频振荡器停振或未稳定, 由硬件清 0, 可读但写入无效;

1: HIRC 高频振荡器已稳定运行, 由硬件置 1, 可读但写入无效。

BIT[4] STBXT - 外部 32768Hz 振荡器稳定标志位

0: 低频振荡器停振或未稳定, 由硬件清 0, 可读但写入无效;

1: 低频振荡器已稳定运行, 由硬件置 1, 可读但写入无效。

注: 当 T3 需要使用该振荡器时, 必须等待 STBXT 置 1, 置 1 说明振荡器已经稳定, 方可切换; 否则 T3 将出现不可预知的情况。

BIT[2] CLKS - CORE 工作时钟选择位 (复位初值取决于时钟振荡模式)

0: 高频时钟 FHCLK 作为系统时钟 Fsys;

1: 低频时钟 FLCLK 作为系统时钟 Fsys。

注 1: 当用户设定了“用户配置字”中设为“单时钟低速运行模式”, 该位设为 0 无效, 系统固定为 1。

注 2: 当用户设定了“用户配置字”中设为“单时钟高速运行模式”, 该位设为 1 无效, 系统固定为 0。

注 3: 当用户设定了“用户配置字”中设为“双时钟运行模式”, 该位复位值 0, 默认选择高频时钟, 系统可配置 0 或 1 切换时钟。

注 4: “双时钟运行模式”下, 当用户需要切换振荡器时, 必须等待 STBH 置 1, 振荡器已经稳定, 方可切换; 否则芯片将出现不可预知的情况。

BIT[1] LFEN - 低频振荡器控制位



0: 任意模式下, 当芯片进入“休眠模式”后, 低频振荡器(LIRC/LOSC)自动停止; 停止模式, 低频振荡器(LIRC/LOSC)始终工作;

1: 任意模式下, 当芯片在进入“休眠模式”后, 低频振荡器(LIRC/LOSC)始终工作。

注 1: 这里的“任意模式”指, “单时钟高速/低速运行模式”和“双时钟运行模式”, 用户通过设定“用户配置字”中的 FMOD 位决定。

注 2: 若用户配置字中的“WDT 工作模式选择位”处于“始终开启看门狗”状态, LFEN 控制的 LIRC 是不能被直接关闭的, 设置 LFEN=0 对 LIRC 无效。

BIT[0] HFEN - 高频振荡器控制位(复位初值取决于时钟振荡模式)

0: 双时钟运行模式下, 在低频运行模式(CLS=1)下进入“停止模式”, 高频振荡器(HIRC/HOSC)停止;

1: 双时钟运行模式下, 在低频运行模式下(CLS=1)下进入“停止模式”, 高频振荡器(HIRC/HOSC)工作。

注 1: 此控制位仅在“双时钟运行模式”、“低频运行下(CLS=1)”和进入“停止模式”三个条件同时成立下有效, 其余情况无效。

注 2: 当用户设定了“用户配置字”的 FMOD 为“单时钟低速运行模式”, 当芯片进入“停止模式”, 该位的设置无效, 即使设为 1, 高频振荡器不会工作。

注 3: 当用户设定了“用户配置字”的 FMOD 为“单时钟高速运行模式”, 当芯片进入“停止模式”, 该位的设置无效, 即使设为 0, 高频振荡器保持工作。

注 4: 当用户设定了“用户配置字”的 FMOD 为“双时钟运行模式”, 且当芯片进入“休眠模式”, 该位的设置无效, 即使设定为 1, 高频振荡器也将自动停止; 仅当芯片进入“停止模式”, 该位的设置才有效; 当 Core 工作于高频运行模式(CLS=0)下, 用户再去设置 HFEN=0, 芯片有自我保护机制, 设置 HFEN=0 无效。

4.8.3 停止模式

在高速或低速运行模式时, 通过先对寄存器 SLPCR 写入“55H”、再将寄存器 PCON 中 STOP 位置“1”, 则系统进入停止模式, 芯片进行相应处理:

- ✧ CORE 停止工作;
- ✧ LD01P5 开启
- ✧ 高频振荡器根据 HFEN 位及生效条件, 决定停止或工作; 低频振荡器保持工作
- ✧ RAM 内容保持不变
- ✧ 所有的输入输出端口保持不变;
- ✧ 全部外设可以根据用户的设定, 停止或保持继续工作;

以下情况可退出停止模式:

- ✧ 发生一个有效的中断请求(可以参考中断章节的内容);
- ✧ 发生一个有效的复位(可以参考中断章节的内容);

芯片退出休眠模式后, 经过振荡等待后, 会将 STOP 清 0, 以高速/低速模式恢复工作。

4.8.4 休眠模式

在高速或低速运行模式时, 通过先对寄存器 SLPCR 写入“55H”、再将寄存器 PCON 中 SLEEP 位置“1”, 则系统进入休眠模式, 芯片进行相应处理:

- ✧ CORE 停止工作;
- ✧ FLASH 进入低功耗模式
- ✧ LD01P5 进入低功耗模式
- ✧ 高频振荡器停止, 低频振荡器根据 LFEN 位决定停止或工作; 用户配置字 WDTM 选择“WDT 始终开启”, 则 LFEN=0 无效;
- ✧ RAM 内容保持不变;
- ✧ 所有的输入输出端口保持不变;
- ✧ 部分外设可以根据用户的设定, 停止或保持继续工作;

以下情况可退出休眠模式:

- ✧ 发生外部中断请求(可以参考中断章节的内容);
- ✧ 发生定时器 T3 中断(可以参考中断章节的内容);
- ✧ 发生一个有效的复位(可以参考中断章节的内容);

芯片退出休眠模式后, 经过振荡等待后, 会将 SLEEP 清 0, 以高速/低速模式恢复工作。

4.8.5 电源控制寄存器(PCON, 0x87/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	UART1EN	-	-	UART0EN	SMOD0	SSTAT0	SLEEP	STOP
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
复位值	0	-	-	0	0	0	0	0

BIT[1] SLEEP - 休眠模式控制位



- 0: 未进入休眠模式, 退出休眠模式后硬件清 0;
1: 进入休眠模式 (必须紧跟解锁保护后), 用户是无法读取到 (1) 这个值的。

BIT[0]

STOP - 停止模式控制位

- 0: 未进入停止模式, 退出停止模式后硬件清 0;
1: 进入停止模式 (必须紧跟解锁保护后), 用户是无法读取到 (1) 这个值的。

注 1: 以上两个模式控制位, 停止/休眠模式都是 CORE 的节能模式, CORE 同一时刻只会进入其中一种模式, 并不会嵌套进入; 若这两位同时置 1, CORE 优先进入休眠 SLEEP 模式。

注 2: 特别注意, 对上述两位 (PCON[1:0]) 操作时, 必须先将寄存器 SLPCR 写 “55H”, 然后立刻将 STOP/SLEEP 位置 1, 才能使系统进入停止/休眠模式, 否则下个时钟周期到来后, 本次操作失效。

4.8.6 模式保护控制寄存器 (SLPCR, 0x86/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SLPCR	SLPCR7	SLPCR6	SLPCR5	SLPCR4	SLPCR3	SLPCR2	SLPCR1	SLPCR0
W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0

BIT[7:0]

SLPCR[7:0] - 模式切换保护位

SLPCR[7:0]	模式保护允许位
55H	允许进入停止/休眠模式
其他	禁止进入停止/休眠模式

注: 该寄存器为只写寄存器, 只有先将寄存器 SLPCR 写 “55H” 再写 STOP/SLEEP 位置 1 的连续 2 条指令, 才能使系统进入停止/休眠模式, 否则下个时钟周期到来后, 寄存器 SLPCR 和 STOP/SLEEP 位中先前写入的值将被硬件清 0。

CORE 进入 STOP 停止模式, C 语言代码应用举例:

```
#include <intrins.h>
.....
sfr SLPCR = 0x86;
sfr PCON = 0x87;

SLPCR = 0x55;           //解锁保护寄存器
PCON |= 0x01;           //CORE 进入停止模式
_nop_();                 //NOP 指令, 不可少
_nop_();                 //NOP 指令, 不可少
```

CORE 进入 SLEEP 休眠模式, C 语言代码应用举例:

```
#include <intrins.h>
.....
sfr SLPCR = 0x86;
sfr PCON = 0x87;

SLPCR = 0x55;           //解锁保护寄存器
PCON |= 0x02;           //CORE 进入休眠模式
_nop_();                 //NOP 指令, 不可少
_nop_();                 //NOP 指令, 不可少
```

4.9 高级能耗控制

芯片内置了一个高级的能耗控制寄存器, 在当今要求嵌入式设备低功耗的时代, 用户可以通过配置本芯片的该寄存器, 进一步降低芯片本身的功耗。

4.9.1 高级能耗控制寄存器 (PWRCR, 0xF8/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWRCR	-	-	PWM1_CLKEN	PWM0_CLKEN	SPI_CLKEN	ADC_CLKEN	-	OCD_CLKEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



复位值	1	1	1	1	1	1	1	1
-----	---	---	---	---	---	---	---	---

BIT[5] PWM1_CLKEN - PWM1 时钟控制位

0: 关闭 PWM1 模块的时钟;

1: 打开 PWM1 模块的时钟。

BIT[4] PWM0_CLKEN - PWM0 时钟控制位

0: 关闭 PWM0 模块的时钟;

1: 打开 PWM0 模块的时钟。

BIT[3] SPI_CLKEN - SPI 时钟控制位

0: 关闭 SPI 模块的时钟;

1: 打开 SPI 模块的时钟。

BIT[2] ADC_CLKEN - ADC 时钟控制位

0: 关闭 ADC 模块的时钟;

1: 打开 ADC 模块的时钟。

BIT[0] OCD_CLKEN - OCD 时钟控制位

0: 关闭 OCD (片上调试) 模块的时钟;

1: 打开 OCD (片上调试) 模块的时钟。

注: 该寄存器中的位需要在模块不使用的情况下, 才能被关闭; 否则芯片有保护机制, “关闭”动作会失败。

4.9.2 高级能耗控制寄存器 1 (PWRCR1, 0xF9/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWRCR1	—	—	CRC_CLKEN	—	—	T3_CLKEN	T2_CLKEN	T01_CLKEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

BIT[5] CRC_CLKEN - CRC 时钟控制位

0: 关闭 CRC 模块的时钟;

1: 打开 CRC 模块的时钟。

BIT[2] T3_CLKEN - T3 时钟控制位

0: 关闭 T3 模块的时钟;

1: 打开 T3 模块的时钟。

BIT[1] T2_CLKEN - T2 时钟控制位

0: 关闭 T2 模块的时钟;

1: 打开 T2 模块的时钟。

BIT[0] T01_CLKEN - T01 时钟控制位

0: 关闭 T01 模块的时钟;

1: 打开 T01 模块的时钟。

注: 该寄存器中的位需要在模块不使用的情况下, 才能被关闭; 否则芯片有保护机制, “关闭”动作会失败。



5 复位

片上有五种复位方式

- ✧ 上电复位 POR;
- ✧ 外部复位;
- ✧ 低电压复位 LVR;
- ✧ 看门狗复位;
- ✧ 软件复位。

任何一种复位发生后，系统将会重新从 0000H 地址处开始执行指令；另外系统还会将大部分的特殊功能寄存器重置为默认复位值。并且可用通过寄存器标志位可以确定复位源。

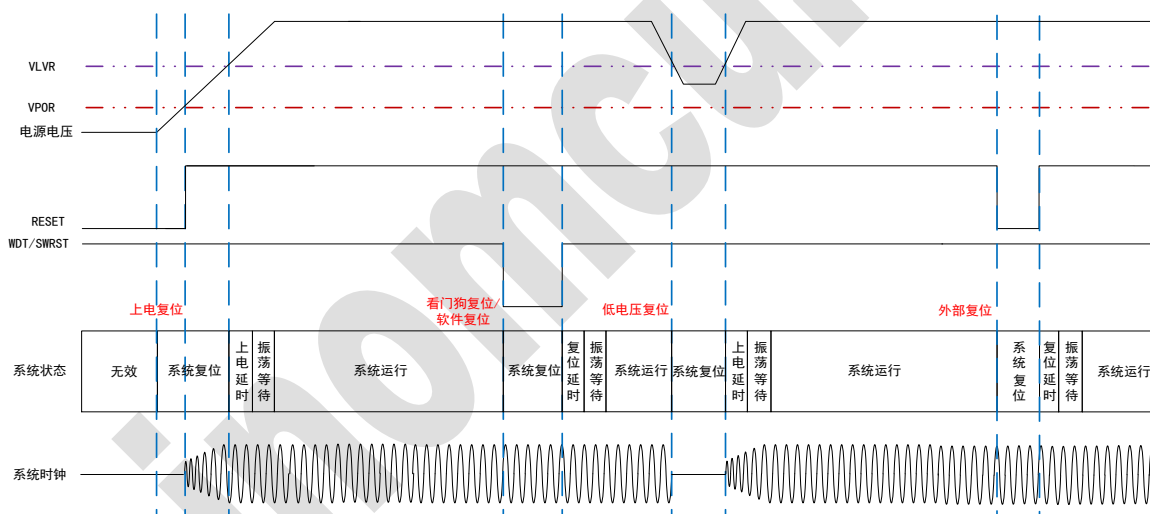
复位寄存器汇总表

地址 (SFRn /H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
SFR0/ E8	RSTFLAG	复位标志寄存器	PORF	LVRF	RSTF	WDTRF	SWRF	-	-	SWRST	00000 --0
SFR0/ CF	WDTCR	看门狗定时器 控制寄存器	-	-	-	-	-	WDTOT2	WDTOT1	WDTOT0	00000 111

5.1 复位条件

CORE 根据复位方式的不同，执行不同的复位动作，根据配置信息完成相应的模块配置后，在外部条件满足时才结束复位，系统重新开始工作。

系统状态与五种复位的关系示意图



5.2 上电复位

在上电过程中，当 VDD 低于参考电压门限值，上电复位将保持芯片为复位模式。这种设计使芯片在 VDD 不满足执行读取存储器时，不访问程序存储器空间。如果从程序存储器读取并执行一个不确定的操作码，可能会使芯片甚至是整个系统进入错误状态。VDD 上升到参考门限电压以上，系统工作，所选的振荡器起振，程序从 0000H 开始执行，同时 PORF 上电标志位将置 1。若发生上电复位，上电标志 PORF 置 1，且上电后内部 RAM 的值为不确定，强烈建议用户初始化 RAM 再使用。

5.3 外部复位

外部复位功能是否开启可以通过芯片配置信息进行设置，选择外部复位功能后复位引脚的内部上拉电阻自动有效。外部复位引脚 RST 是施密特结构的，低电平有效。当外复位引脚为高电平时，系统正常运行；为低电平时，系统产生复位。若发生外部复位，程序从 0000H 开始执行。同时外部复位 RSTF 标志位将置 1。

5.4 低电压复位

芯片的 LVR 电压有 4 级可选：2.1V、2.5V、3.5V、4.1V。LVR 的电压检测电路有一定的回滞特性，通常回滞电压为 0.05V 左右，则当电源电压下降到 LVR 电压时 LVR 复位有效，而电压需要上升到 LVR 电压+0.05V 时 LVR 复位才会解除。



若发生低电压复位，程序从 0000H 开始执行。同时低电压 LVRF 标志位将置 1。

LVR 通过芯片的“用户配置字”进行设置，软件无法调整复位电压等级。

5.5 看门狗复位

看门狗复位是一种对程序正常运行的保护机制。正常情况下，用户软件需要按时对 WDT 定时器进行清零操作，保证 WDT 不溢出。若出现异常状况，程序未按时对 WDT 定时器清零，WDT 会溢出从而产生看门狗复位，系统重新初始化，返回受控状态。若发生看门狗复位，程序从 0000H 地址开始执行。同时看门狗复位 WDTRF 标志位将被置 1。

WDT 开启与关闭通过“用户配置字”进行设置，软件无法调整；WDT 只能通过用户配置字的 WDTM 选择“始终开启 WDT”、“停止/休眠模式关闭 WDT”、“始终关闭 WDT”等 3 种工作模式。

(1) 若选择始终开启 WDT 模式，在停止/休眠模式下 WDT 依然运行，WDT 溢出后将复位芯片。

(2) 若选择停止/休眠模式下关闭 WDT 模式，则 WDT 在停止/休眠模式下被硬件自动关闭，并在芯片恢复运行后自动继续。

(3) 若选择始终关闭 WDT 模式，则整个 WDT 模块功能将失效，“用户配置字”中默认为此项。

当 WDT 开启时，默认为最大溢出时间，用户可以通过改变 WDTOT[2:0] 的值，来调整的 WDT 溢出时间。因为 WDT 的时钟源来自 LIRC，故一旦在“用户配置字”中选择“始终开启 WDT”后，LIRC 将不能在软件中被关闭，将始终为 WDT 提供时钟源。

WDT 设计为一个向上计数的计数器，计数到设定值以后产生溢出，触发对应的看门狗复位事件。该计数器的值对用户不可见，但用户可以通过读/写一次 WDTCR 寄存器，实现“喂狗操作”，该动作发生的同时，硬件上自动完成一次对 WDT 计数器的重载。

看门狗定时器控制寄存器 (0xCF, SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTCR	—	—	—	—	—	WDTOT2	WDTOT1	WDTOT0
R/W	—	—	—	—	—	R/W	R/W	R/W
复位值	—	—	—	—	—	1	1	1

BIT[2:0] WDTOT[2:0] - WDT 溢出时间控制位

WDT[2:0]	WDT 溢出时间
000	8ms
001	16ms
010	32ms
011	64ms
100	128ms
101	256ms
110	512ms
111	1024ms (复位值)

注 1：上表中的“WDT 溢出时间”是一个理论的溢出时间，实际溢出时间和 LIRC 有关，请以 LIRC 规格为准；

注 2：寄存器的配置（用户软件上）仅可控制看门狗的溢出时间；但是看门狗定时器开启与否，必须在“用户配置字”中设定。

注 3：在 WDT 的溢出时间之内，读/写一次 WDTCR 寄存器，即完成一次“喂狗操作”；此时硬件会自动重载一次 WDT 计数器的值，并继续开始向下递减计数。

喂狗操作（清 WDT），C 语言代码应用举例：

```
sfr WDTCR = 0xCF;
```

```
WDTCR = 0x03; //WDT 复位之前，喂狗操作（清 WDT），再从 64ms 开始重新倒数计时
```

5.6 软件复位

软件复位是一种对可控的程序自发的复位行为，有点类似于外部复位。正常情况下，用户软件需要连续两次对 SWRST 位写两次，启动软件复位。若发生软件复位，程序从 0000H 开始执行。同时软件复位 SWRF 标志位将置 1。

注：软件复位，不会加载用户配置字 BOOTS 位，其他复位（除 wdt 复位）会加载 BOOTS 至 IAPCR1->BOOT(BIT7)。

复位标志寄存器 (0xE8, SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTFLAG	PORF	LVRF	RSTF	WDTRF	SWRF	—	—	SWRST
R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W
复位值	0	0	0	0	0	—	—	0

BIT[0] SWRST - 软件复位



0: 无软件复位要求;

1: 启动软件复位 (连续写两遍有效), 系统软件复位后硬件清 0, 并将 SWRF 标志位置 1。

注 1: 只有对该位连续写 2 条相同的指令, 才能使系统进行软件复位, 否则下个时钟周期到来后, 寄存器中先前写入的值将被硬件自动清掉, 保持为 0, 这是一种简单的保护机制。

注 2: 写入 1 成功后, 芯片直接“软件复位”, 用户查询不到该值为 1 的状态。

软件复位, C 语言代码应用举例:

```
sfr RSTFLAG = 0xE8;
```

```
RSTFLAG |= 0x01;
```

```
RSTFLAG |= 0x01;
```

```
//软件复位生效, 以下代码将不会执行
```

5.7 复位标志寄存器

系统在热复位的情况下, 用户可以通过查看复位标志寄存器, 确认系统复位的原因。

5.7.1 复位标志寄存器 (RSTFLAG, 0xE8/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTFLAG	PORF	LVRF	RSTF	WDTRF	SWRF	—	—	SWRST
R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W
复位值	0	0	0	0	0	—	—	0

BIT[7] PORF - 上电复位标志位

0: 未发生上电复位;

1: 曾经发生过上电复位, 硬件置 1 但不会清 0, 需要用户手动清。

BIT[6] LVRF - 低电压复位标志位

0: 未发生低电压复位;

1: 曾经发生过低电压复位, 硬件置 1 但不会清 0, 需要用户手动清。

BIT[5] RSTF - 外部复位标志位

0: 未发生外部复位;

1: 曾经发生过外部复位, 硬件置 1 但不会清 0, 需要用户手动清。

BIT[4] WDTRF - 看门狗复位标志位

0: 没有发生过看门狗定时器溢出复位;

1: 曾经发生过看门狗定时器溢出复位, 硬件置 1 但不会清 0, 需要用户手动清。

注: 特别注意, 该标志位在程序查询的同时, 兼具喂狗的作用; 故不建议用户在开启 WDT 后反复查询该标志位, 否则可能会造成 WDT 正常的“复位功能”失效。

BIT[3] SWRF - 软件复位标志位

0: 没有发生过软件复位;

1: 曾经发生过软件复位, 硬件置 1 但不会清 0, 需要用户手动清。

注 1: 以上 5 个复位标志在置位 (=1) 后, 需要手动清除, 否则硬件不会自动清零 (=0); 建议用户在每次使用这些标志位之前, 将以上 5 个“复位标志位”全部清零。

注 2: 若不清零, 不同复位原因标志位之间可以累积记录; 但对于同一复位原因标志位, 前后复位之间, 对已经置位 (=1) 的标志位将无影响, 所以用户可能因此检查不到“重复复位”的标志位。

5.8 各种复位与低功耗模式的关系

复位与停止模式和休眠模式的关系

复位源	停止模式中 (STOP=1) 复位有效	休眠模式中 (SLEEP=1) 复位有效
POR 复位	Yes	Yes
外部复位	Yes	Yes
低电压复位	Yes	Yes ^{注 2}
看门狗复位	Yes	Yes
软件复位	No	No

注 1: 通过上表中的部分复位操作, 虽然可以让芯片从停止模式和休眠模式迁移出来, 虽然这是一种退出“低功耗模式”的手段, 但是程序将从 0x0000 地址开始执行, 因此可能造成实际运行结果与用户的预想的结果截然不同。

注 2: 通过“用户配置字中”关闭休眠模式下的 LVR 功能, 则不会产生 LVR 复位。



6 I/O端口

6.1 端口特性

芯片共有 3 组 8 位端口的 P0、P1 和 P2，以及一组 2 位端口的 P3，总共 4 组 I/O 端口。

每个 I/O 都有 4 种输入/输出模式可选：推挽输出模式、输入模式、高阻态模式、开漏输出模式。

推挽输出模式：此模式下能输出很强的驱动能力。

输入模式：此模式下可实现输入，而根据用户的配置实现不带上拉电阻、带上拉电阻 2 种输入方式。

高阻态模式：此模式下既不输入也不输出，既不是高电平也不是低电平，“高阻态模式”可以用来隔绝芯片和外围电路之间的相互影响，近似于断开。

开漏输出模式：此模式下仅有输出低驱动能力，无输出高驱动能力，可配置内部上拉电阻

6.2 端口相关寄存器

端口寄存器汇总表

地址 (SFR n/H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
SFR0/ C3	P0PU	P0 端口上拉电 阻控制寄存器	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU	00000 000
SFR0/ C2	P0MOD1	P0 端口模式寄 存器 1	P07M1	P07M0	P06M1	P06M0	P05M1	P05M0	P04M1	P04M0	1010 1010
SFR0/ C1	P0MOD0	P0 端口模式寄 存器 0	P03M1	P03M0	P02M1	P02M0	P01M1	P01M0	P00M1	P00M0	1010 1010
SFR0/ B3	P3PU	P3 端口上拉电 阻控制寄存器	-	-	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU	--00 0000
SFR0/ B2	P3MOD1	P3 端口模式寄 存器 1	-	-	-	-	P35M1	P35M0	P34M1	P34M0	---- 1010
SFR0/ B1	P3MOD0	P3 端口模式寄 存器 0	P33M1	P33M0	P32M1	P32M0	P31M1	P31M0	P30M1	P30M0	1010 1010
SFR0/ B0	P3	P3 端口数据寄 存器	-	-	P35D	P34D	P33D	P32D	P31D	P30D	--xx xxxx
SFR0/ A3	P2PU	P2 端口上拉电 阻控制寄存器	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU	00000 000
SFR0/ A2	P2MOD1	P2 端口模式寄 存器 1	P27M1	P27M0	P26M1	P26M0	P25M1	P25M0	P24M1	P24M0	1010 1010
SFR0/ A1	P2MOD0	P2 端口模式寄 存器 0	P23M1	P23M0	P22M1	P22M0	P21M1	P21M0	P20M1	P20M0	1010 1010
SFR0/ A0	P2	P2 端口数据寄 存器	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D	xxxx xxxx
SFR0/ 9F	PXODM	端口输出驱动 控制寄存器	-	P30DM0	-	P20DM0	-	P10DM0	-	P00DM0	-0-0 -0-0
SFR0/ 93	P1PU	P1 端口上拉电 阻控制寄存器	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU	00000 000
SFR0/ 92	P1MOD1	P1 端口模式寄 存器 1	P17M1	P17M0	P16M1	P16M0	P15M1	P15M0	P14M1	P14M0	1010 1010
SFR0/ 91	P1MOD0	P1 端口模式寄 存器 0	P13M1	P13M0	P12M1	P12M0	P11M1	P11M0	P10M1	P10M0	1010 1010
SFR0/ 90	P1	P1 端口数据寄 存器	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D	xxxx xxxx
SFR0/ 80	P0	P0 端口数据寄 存器	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D	xxxx xxxx

6.3 端口模式及数据控制

每个 I/O 都有 4 种输入/输出模式可选：推挽输出模式、输入模式、高阻态模式、开漏输出模式。

推挽输出模式：此模式下能输出很强的驱动能力。

输入模式：此模式下可实现输入，而根据用户的配置实现不带上拉电阻、带上拉电阻 2 种输入方式。

高阻态模式：此模式下既不输入也不输出，既不是高电平也不是低电平，“高阻态模式”可以用来隔绝芯片和外围电路之间的相互影响，近似于断开。

开漏输出模式：此模式下仅有输出低驱动能力，无输出高驱动能力，可配置内部上拉电阻。

端口数据寄存器 P0/P1/P2/P3，全部支持位寻址。



6.3.1 P0 端口模式寄存器 0 (P0MOD0, 0xC1/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0MOD0	P03M1	P03M0	P02M1	P02M0	P01M1	P01M0	P00M1	P00M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

6.3.2 P0 端口模式寄存器 1 (P0MOD1, 0xC2/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0MOD1	P07M1	P07M0	P06M1	P06M0	P05M1	P05M0	P04M1	P04M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

BIT[7:0] P0nM[1:0] (n = 7 - 0) - P0 端口模式选择位, 初始状态为高阻态模式

6.3.3 P1 端口模式寄存器 0 (P1MOD0, 0x91/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1MOD0	P13M1	P13M0	P12M1	P12M0	P11M1	P11M0	P10M1	P10M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

6.3.4 P1 端口模式寄存器 1 (P1MOD1, 0x92/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1MOD1	P17M1	P17M0	P16M1	P16M0	P15M1	P15M0	P14M1	P14M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

BIT[7:0] P1nM[1:0] (n = 7 - 0) - P1 端口模式选择位, 初始状态为高阻态模式

6.3.5 P2 端口模式寄存器 0 (P2MOD0, 0xA1/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2MOD0	P23M1	P23M0	P22M1	P22M0	P21M1	P21M0	P20M1	P20M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

6.3.6 P2 端口模式寄存器 1 (P2MOD1, 0xA2/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2MOD1	P27M1	P27M0	P26M1	P26M0	P25M1	P25M0	P24M1	P24M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

BIT[7:0] P2nM[1:0] (n = 7 - 0) - P2 端口模式选择位, 初始状态为高阻态模式

6.3.7 P3 端口模式寄存器 0 (P3MOD0, 0xB1/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3MOD0	P33M1	P33M0	P32M1	P32M0	P31M1	P31M0	P30M1	P30M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

6.3.8 P3 端口模式寄存器 1 (P3MOD1, 0xB2/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3MOD0	-	-	-	-	P35M1	P35M0	P34M1	P34M0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
复位值	-	-	-	-	1	0	1	0

BIT[3:0] P3nM[1:0] (n = 5 - 0) - P3 端口模式选择位, 初始状态为高阻态模式

端口输入/输出模式对应表

PxnM1 (x = 0-3; n = 7-0)	PxnM0 (x = 0-3; n = 7-0)	IO 工作模式
0	0	输入模式
0	1	推挽输出模式



1	0	高阻态模式（复位值）
1	1	开漏输出模式

6.3.9 P0 端口数据寄存器（P0, 0x80/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

BIT[7:0] P0nD (n = 7 - 0) - P0 端口数据位

6.3.10 P1 端口数据寄存器（P1, 0x90/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

BIT[7:0] P1nD (n = 7 - 0) - P1 端口数据位

6.3.11 P2 端口数据寄存器（P2, 0xA0/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

BIT[7:0] P2nD (n = 7 - 0) - P2 端口数据位

6.3.12 P3 端口数据寄存器（P3, 0xB0/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3	-	-	P35D	P34D	P33D	P32D	P31D	P30D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	X	X	X	X	X	X

BIT[1:0] P3nD (n = 5 - 0) - P3 端口数据位

端口输出电平对应表

PxnD[1:0] (x = 0-3; n = 7-0)	IO 电平
0	低电平
1	高电平
X	不确定（复位值）

注：P0、P1、P2、P3 口的数据值在复位之后为（X）不确定状态，建议用户使用之前先写入一个确定的初始电平值。

6.4 输入上拉电阻控制

P0、P1、P2、P3 端口的每位都内置有独立的输入上拉电阻控制。无论端口处于输入或输出模式都配置有效。

6.4.1 P0 端口上拉电阻控制寄存器（P0PU, 0xC3/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0PU	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] P0nPU0 (n = 7 - 0) - P0 端口上拉电阻选择位 0

6.4.2 P1 端口上拉电阻控制寄存器（P1PU, 0x93/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1PU	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] P1nPU0 (n = 7 - 0) - P1 端口上拉电阻选择位 0



6.4.3 P2 端口上拉电阻控制寄存器 (P2PU, 0xA3/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2PU	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] P2nPU0 (n = 7 - 0) - P2 端口上拉电阻选择位 0

6.4.4 P3 端口上拉电阻控制寄存器 (P3PU, 0xB3/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3PU	-	-	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	0	0	0	0	0	0

BIT[1:0] P3nPU0 (n = 5 - 0) - P3 端口上拉电阻选择位 0

端口上拉电阻选择对应表

PxnPU (x = 0-3; n = 7-0)	上拉电阻选择
0	无上拉电阻控制 (复位值)
1	输入带内部上拉 (阻值约 20K)

6.5 驱动电流控制

6.5.1 端口输出驱动控制寄存器 (PXODM, 0x9F/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PXODM	-	P3ODM0	-	P2ODM0	-	P1ODM0	-	P0ODM0
R/W	-	R/W	-	R/W	-	R/W	-	R/W
复位值	-	0	-	0	-	0	-	0

BIT[6] P3ODM0 - P3 端口输出驱动控制选择

BIT[4] P2ODM0 - P2 端口输出驱动控制选择

BIT[2] P1ODM0 - P1 端口输出驱动控制选择

BIT[0] P0ODM0 - P0 端口输出驱动控制选择

PnODM0 (n = 3 - 0)	输出模式下, GPIO 的驱动电流选择
0	输出电流大小参考 Ioh01 / IoI01 的值 (复位值)
1	输出电流大小参考 Ioh02 / IoI02 的值

注: 本芯片同一时刻 P0 口、P1 口、P2 口、P3 口只能为一种输出驱动电流方式, 且一组端口上的一个 PIN 不能单独配置。

6.6 数字功能端口全映射 (DFM)

本芯片部分端口可以复用为其他功能模块的输入/输出端口, 数字输入功能无优先级, 允许多对一映射, 数字输出功能的优先级: 不支持端口全映射功能→支持全映射功能输出优先级列表→GPIO 输出, 用户可根据需要通过寄存器配置成需要功能。

芯片内置数字功能端口全映射模块 (DFM), 用户可以通过软件配置 FFM 模块将绝大多数的数字功能配置在任意一个非电源端口上。提高外设引脚布局灵活性, 可以实现 PCB 过程引脚布局错误后重分配, 升级 PCB 最小改动项。数字功能为输入功能 (T0/1/2 外部输入、RXD 等等) 特性时, 允许多对一映射, 即多种数字输入功能分配到同一 IO 端口上。

数字功能为输出功能 (T0/1/2 时钟输出、TXD 等等) 特性时, 如果将多个输出特性数字功能引脚分配到同一 IO 端口, 遵循固定的优先级, 只能有一个输出有效。

6.6.1 可全映射的数字功能*

功能模块	名称	类型	描述
定时器	T0	I/O	Timer0 的外部输入或 Timer0 的溢出输出
	T1	I/O	Timer1 的外部输入或 Timer1 的溢出输出
	T2	I/O	Timer2 的外部输入或 Timer2 的溢出输出
	T2EX	I	Timer2 的重载触发/输入捕捉输入
PWM	PWM0	O	PWM0 独立输出端口
	PWM1A	O	PWM1 输出通道 A
	PWM1B	O	PWM1 输出通道 B



	PWM1C	0	PWM1 输出通道 C
	PWM1AN	0	PWM1 输出通道 AN
	PWM1BN	0	PWM1 输出通道 BN
	PWM1CN	0	PWM1 输出通道 CN
	PWM1FB	I	PWM1 刹车反馈输入
UART	RX0	I	UART0 的 RXD 端口
	TX0	0	UART0 的 TXD 端口
	RX1	I	UART1 的 RXD 端口
	TX1	0	UART1 的 TXD 端口
SPI	nSS	I	SPI 从机片选端口
	SCK	I/O	SPI 主机时钟输出/从机输入
	MISO	I/O	SPI 主机数据输入/从机数据输出
	MOSI	I/O	SPI 主机数据输出/从机数据输入
CMP	CMP00	0	比较器 0 输出
	CMP10	0	比较器 0 输出

6.6.2 不可全映射的功能

DFM 不支持全映射的功能包括：模拟功能、外部中断 INTx、外部复位 nRST、[CLKx]/[DIOx]和 ADTRG，详情参考<引脚说明>章节。

6.6.3 端口控制相关寄存器

端口映射寄存器放置于扩展的 XSFR 区，采用 MOVX 来访问。

起始地址	偏移地址								SFR 页
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
FF8H~FFC0H	-	-	-	-	-	-	-	-	XSFR
FFB8H	-	-	-	-	-	-	-	-	XSFR
FFB0H	CMP00_MAP	=	CMP10_MAP	=	-	-	-	-	XSFR
FFA8H	PWMO_MAP	-	-	-	-	-	-	-	XSFR
FFA0H	PWM1A_MAP	PWM1AN_MAP	PWM1B_MAP	PWM1BN_MAP	PWM1C_MAP	PWM1CN_MAP	PWM1FB_MAP	-	XSFR
FF98H	nSS_MAP	SCK_MAP	MISO_MAP	MOSI_MAP	-	-	-	-	XSFR
FF90H	-	-	-	-	-	-	-	-	XSFR
FF88H	TX0_MAP	RX0_MAP	-	-	TX1_MAP	RX1_MAP	-	-	XSFR
FF80H	T0_MAP	T1_MAP	T2_MAP	-	-	-	-	T2EX_MAP	XSFR

6.6.4 端口映射控制寄存器 (xxx_MAP, 0xFFxx/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
xxx_MAP	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0
R/W	-	-	R/W	R/W	-	R/W	R/W	R/W
复位值	-	-	1	1	-	1	1	1

BIT[5:4] PORTSEL[1:0] - 全映射 PORT 选择位

00: P0

01: P1

10: P2

11: P3

BIT[2:0] PINSEL[2:0] - 全映射 PIN 选择位

PINSEL[2:0] = 0~7, 对应端口的 0~7 PIN 脚。

注：寄存器复位值为 0x37，复位后的所有 IO 为 GPIO 功能，用户若要复用数字功能前须先配置上述寄存器，否则功能不能使用。

6.6.5 全映射配置应用说明

数字功能为输入功能（T0/1/2 外部输入、RXD 等等）特性时，允许多对一映射，即多种数字输入功能分配到同一 IO 端口上。

数字功能为输出功能（T0/1/2 时钟输出、TXD 等等）特性时，如果将多个输出特性数字功能引脚分配到同一 IO 端口，遵循固定的优先级，只能有一个输出有效。

应用举例：

UART0 的 TX0 和 RX0 映射到 P1.0 和 P1.1 上，配置步骤如下：

1、配置代码



```
TX0_MAP = 0x10;    //TX0 map to P1.0  
RX0_MAP = 0x11;    //RX0 map to P1.1
```

2、启动 UART0

输出多对一映射：

多路输出映射到同一端口，仅 1 路有效，优先级如下：

功能	优先级
TX0	0 最高级
RX0	1
SCK_OUT	2
MISO_S0	3
MOSI_MO	4
CMP00	5
CMP10	6
PWM1A	7
PWM1AN	8
PWM1B	9
PWM1BN	10
PWM1C	11
PWM1CN	12
PWM0	13
TO_OUT	14
T1_OUT	15
T2_OUT	16
TX1	17
RX1	18 最低

应用举例：

PWM0_MAP、TO_MAP 同时配置到 P2.0 端口，TO 配置为溢出反转输出，DFM 模块根据上表优先级将 PWM0 配置有效，TO 输出映射无效。

当所有的映射都不选择指定端口映射，那么这个端口作为 GPIO；

若设置 TO_MAP、T1_MAP、T2_MAP 都为 0x31，且 T0/T1/T2 配置 Tx 引脚为输入，则从 P3.1 端口输入的信号将同时作用于 3 个模块。

当 RX0 和 TX0 配置为一个端口，则 TXD 和 RXD 内部相连。

6.7 模拟功能

设置端口模式为高阻模式，并根据需求开启相应的模块控制位，端口复用为模拟功能。

端口模拟功能复用优先级高于数字功能，模拟输入功能允许多对一映射，即多种模拟输入功能分配到同一 IO 端口上。

若同一个端口复用多个模拟输出功能，且多个输出功能开启，开启的输出功能遵循固定的优先级，只能有一个输出有效。

优先级按照<引脚说明>章节模拟功能从上往下，优先级从高至低；用户配置字控制优先级最高。

举例：

引脚编号			功能描述	
LQFP	QFN	QFN	模拟功能	模拟复用优先级
1	28	21	模拟输出 0	高
			模拟输出 1	低
			模拟输出 2	最高

注：模拟输出 0 和模拟输出 1，按照从上往下，优先级从高到底；（模拟输出 2）为用户配置字控制，若开启优先级最高。



7 硬件乘除法器 (MDU)

7.1 特性

✧ 硬件实现在 1 周期内完成有符号/无符号 16-bit X 16-bit 乘法

✧ 硬件实现在 8 周期内完成有符号/无符号 32-bit ÷ 16-bit 除法 (注: 除数为负数时需要特别注意, 详见 7.3 描述)

✧ 硬件实现在 1 周期内完成有符号/无符号 32-bit 数据最多 31 次移位

乘/除法、移位单元是由寄存器 MDUCR、MDUDRA0 - MDUDRA3、MDUDRB0 - MDUDRB1 完成 16-bit X 16-bit 数乘法、32-bit ÷ 16-bit 数除法或者 32-bit 无符号数据左/右移位的运算。由于运算过程由硬件完成, 转化速率快, 高效, 为程序运行节省了大量的时间。

7.2 16-bit X 16-bit 数乘法

进行 16-bit X 16-bit 乘法操作时, 乘数和被乘数分别存放于寄存器 [MDUDRA1:MDUDRA0] 和 [MDUDRB1:MDUDRB0] 中。根据乘法交换律, 乘数和被乘数存放的寄存器也可以互相交换。

操作流程如下:

1、写被乘数和乘数到寄存器 [MDUDRB1:MDUDRB0] 和 [MDUDRA1:MDUDRA0]

2、设置寄存器 MDUCR (MD[1:0] = 00, SIGN = 0) 选择无符号乘法操作, 操作数用原码表示; SIGN = 1 选择有符号乘法操作, 操作数用补码表示, 最高位表示符号位; 置位 START 则启动运算。

注: 启动之前, 对乘法操作的寄存器设置没有顺序要求。

4、当硬件完成操作后, 会自动将 MDUCR 中的 START 清 0。此时, 可以从相应寄存器中读取乘积

注: 由于乘法操作只需要一个系统周期执行时间, 因此可以不用判断完成标志位 (START), 直接读取乘法结果。

5、16-bit X 16-bit 乘积是一个 32-bit 数据, 存放于 [MDUDRA3:MDUDRA0], 其中 MDUDRA3 是 MSB, MDUDRA0 是 LSB。

注: 当读取乘积时, 对读取顺序没有要求, 建议大端或小端顺序读取 (根据实际数据处理需求)。读取操作不会改变寄存器中的乘积值。乘积将一直保留在 [MDUDRA3:MDUDRA0], 除非被硬件/软件改写。

注: 对于无符号的乘法操作, 操作数都是以原码的方式写入和结果以原码的形式储存; 对于有符号的乘法操作, 操作数都是以补码的形式写入和结果以补码方式储存。

7.3 32-bit ÷ 16-bit 数除法

进行 32-bit ÷ 16-bit 除法时, [MDUDRA3:MDUDRA0] 用于存放除法操作的被除数; [MDUDRB1:MDUDRB0] 用于存放除法操作的除数。32-bit ÷ 16-bit 除法结果有一个 32-bit 商和一个 16-bit 余数。其中, 商存放于 [MDUDRA3:MDUDRA0], MDUDRA3 是 MSB, MDUDRA0 是 LSB。余数存放于 [MDUDRB1:MDUDRB0], 其中 MDUDRB1 是高位, MDUDRB0 是低位。

操作流程如下:

1、写除数到寄存器 [MDUDRB1:MDUDRB0], 写被除数到寄存器 [MDUDRA3:MDUDRA0]

2、设置寄存器 MDUCR (MD[1:0] = 01, SIGN = 0) 选择无符号除法操作, 操作数用原码表示; SIGN = 1 选择有符号除法操作, 操作数用补码表示, 最高位表示符号位; 置位 START 启动运算

注: 启动之前, 对除法操作的寄存器设置没有顺序要求。

3、由于除法操作需要 8 个系统周期执行时间, 如果读出 START 的值为 1 则表示运算未完成。

4、当硬件完成操作后, 会自动将 MDUCR 中的 START 清 0。此时, 可以从相应寄存器中读取商和余数

注: 读取商或者余数时, 对读取顺序没有要求。读取操作不会改变寄存器中的商和余数。商和余数将一直保留在相应的寄存器中, 除非被硬件/软件改写

注: 对于有符号除法运算, 对取余数规则描述如下: $-10 \div 3$, 商为-3, 余数为-1。而不是商为-4, 余数为+2

注: 有符号除法中, 仅当除数为负数时需特别注意, 商没问题, 余数的符号是和 KEIL C 中得到的结果相反的 (application note 中有详细说明)

注: 对于无符号的除法操作, 操作数都是以原码的方式写入和结果以原码形式储存; 对于有符号的除法操作, 操作数都是以补码的形式写入和商和余数以补码方式储存

7.4 32-bit 移位

32-bit 数据移位结果仍然是 32-bit 数据, 32-bit 操作数存放于 [MDUDRA3:MDUDRA0], 移位次数存放在 MDUDRB0 中, 结果仍然存放于 [MDUDRA3:MDUDRA0], MDUDRA3 是 MSB, MDUDRA0 是 LSB。

无符号数不论是左移操作还是右移操作, 移出数据的空位均是补 0。有符号数左移, 空位补 0, 最高位可能会溢出, 这点和 KEIL C 编译器一致; 右移, 空位补符号位 (正数右移, 最高位补 0; 负数右移最高位补 1) 操作也与 KEIL C 移位结果一致。

操作流程如下:



- 1、写待移位数据到寄存器[MDUDRA3:MDUDRA0]，写待移位的次数到 MDUDRB0 (DB[4:0]的值即为移位次数)
- 2、设置寄存器 MDUCR (MD[1:0] = 10)，选择移位操作，设置 SDIR 选择移位方向，置位 START 位开始移位运算；
注：启动之前，对移位操作的寄存器设置没有顺序要求。
- 3、当硬件完成操作后，会自动将 MDUCR 中 START 清 0。此时，可以从相应的寄存器中读取移位结果
注：由于移位操作只需要一个系统周期执行时间，因此可以不用判断完成标志位 (START)，直接读取移位操作后的结果
注：读取操作不会改变寄存器中的移位结果。移位结果将一直保留在相应的寄存器中，除非被硬件/软件改写

7.5 快速清空操作数/结果寄存器

MDU 模块内置快速清除所有操作数/结果寄存器功能。将 MDUCR 寄存器中的 MD[1:0] = 11，置位 START 开始清 0 操作。将在一个系统周期时间内由硬件清 0 所有操作数结果寄存器 (MDUDRA3~MDUDRA0, MDUDRB1~MDUDRB0) 的值。

7.6 异常状态

如果除法操作中除数为 0，则不会执行除法操作，操作数寄存器[MDUDRA3:MDUDRA0]和[MDUDRB1:MDUDRB0]保持原有值不变。硬件自动将操作控制寄存器 MDUCR 的位 START 清 0 并置位 DVERR 标志表示异常情况，其余位保持不变，不会影响用户程序继续运行。DVERR 异常标志位会在下一次启动运算时硬件自动清 0。

7.7 注意事项

- 注 1：在硬件执行除法操作没有结束时试图读取运算结果（即读取寄存器 MDUDRA3~MDUDRA0 和 MDUDRB1~MDUDRB0 中任意一个或者几个），会得到不可预知的结果。但是，读取动作不会影响硬件继续运算并得到正确结果。当硬件完成运算后，仍然可以读取到正确的结果。
- 注 2：移位操作中，如果移位次数值为 0 (DB[4:0] = 0)，则不会移位，操作数寄存器 MDUDRA3~MDUDRA0 和 MDUDRB1~MDUDRB0 保持原有值不变。硬件自动将操作控制寄存器 MDUCR 的位 START 清 0，其余位保持不变，不会影响用户程序继续运行。
- 注 3：在运算未完成时 (START = 1)，任何对 MDUDRA3~MDUDRA0 和 MDUDRB1~MDUDRB0 的写入操作都是无效的。
- 注 4：有符号除法中，除数为负数时用法要特别注意。

7.8 硬件乘除法相关寄存器

地址 (SFR n/H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR0/E1	MDUCR	运算控制寄存器	MD1	MD0	SDIR	-	-	SIGN	DVERR	START	0000000
SFR0/E2	MDUDRA0	运算/结果寄存器 A 低 8 位	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	0000000
SFR0/E3	MDUDRA1	运算/结果寄存器 A 次低 8 位	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	0000000
SFR0/E4	MDUDRA2	运算/结果寄存器 A 次高 8 位	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	0000000
SFR0/E5	MDUDRA3	运算/结果寄存器 A 高 8 位	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24	0000000
SFR0/E6	MDUDRB0	运算/结果寄存器 B 低位	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0000000
SFR0/E7	MDUDRB1	运算/结果寄存器 B 高位	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	0000000

7.8.1 运算控制寄存器 (MDUCR, 0xE1/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUCR	MD1	MD0	SDIR	-	-	SIGN	DVERR	START
R/W	R/W	R/W	R/W	-	-	R/W	R/W	R/W
复位值	0	0	0	-	-	0	0	0

BIT[7:6] MD[1:0] - 位移、乘除法选择位

MD[1:0]	运算功能选择位
00	执行乘法操作 (默认)
01	执行除法操作



	10	执行移位操作
	11	执行所有操作数/结果寄存器清 0 操作
BIT[5]	SDIR - 位移方向选择位, 软件设置 (只有进行移位操作时有效) 0: 32bit 数据左移 (默认); 1: 32bit 数据右移。	
BIT[2]	SIGN - 乘除法算法符号使能位 0: 进行无符号数操作 (默认); 1: 进行有符号数操作。	
BIT[1]	DVERR - 除法运算异常标志位, 硬件置位, 并在下一次运算启动时硬件自动清 0 0: 无除数为 0 的情况发生; 1: 有除数为 0 的情况发生 (不可软件写 1, 写 1 无效)。	
BIT[0]	STRAT - 运算启动标志位, 只能硬件清 0, 软件置 1 0: 运算已结束或未启动; 1: 启动乘/除法、移位运算或操作数结果寄存器清 0 操作, 运算过程中标志为 1, 运算完成后标志由硬件清 0。	

7.8.2 运算/结果寄存器 A 低 16 位 (MDUDRA[1:0], 0xE2/0xE3/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRA0	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRA1	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

MDUDRA0 BIT[7:0]

MDUDRA1 BIT[7:0] DA[15:0] - 运算/结果 A 低 16 位

MD[1:0]	运算功能选择位
00	作 16bit X 16bit 乘法操作时 (MD[1:0] = 00) MDUDRA0/1 存放 16bit 乘数, DA15 为 MSB, DA0 为 LSB
01	作 32bit ÷ 16bit 除法操作时 (MD[1:0] = 01) MDUDRA0/1 存放 32bit 被除数的低 16 位, DA15 为 MSB, DA0 为 LSB
10	作 32bit 移位操作时 (MD[1:0] = 10) MDUDRA0/1 存放 32bit 移位操作数的低 16 位, DA15 为 MSB, DA0 为 LSB
-	运算完成时 (START = 0) MDUDRA0/1 存放 32bit 结果的低 16 位, DA15 为 MSB, DA0 为 LSB

7.8.3 运算/结果寄存器 A 高 16 位 (MDDRA[3:2], 0xE4/0xE5/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRA2	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRA3	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

MDUDRA2 BIT[7:0]

MDUDRA3 BIT[7:0] DA[31:16] - 运算/结果 A 高 16 位

MD[1:0]	运算功能选择位
01	作 32bit ÷ 16bit 除法操作时 (MD[1:0] = 01) MDUDRA2/3 存放 32bit 被除数的高 16 位, DA31 为 MSB, DA16 为 LSB
10	作 32bit 移位操作时 (MD[1:0] = 10) MDUDRA2/3 存放 32bit 移位操作数的高 16 位, DA31 为 MSB, DA16 为 LSB
-	运算完成时 (START = 0) MDUDRA2/3 存放 32bit 结果的高 16 位, DA31 为 MSB, DA16 为 LSB



7.8.4 运算/结果寄存器 B 16 位 (MDUDRB[1:0], 0xE6/0xE7/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRB0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MDUDRB1	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

MDUDRB0 BIT[7:0]

MDUDRB1 BIT[7:0] DB[15:0] - 运算/结果 B 16 位

MD[1:0]	运算功能选择位
00	作 16bit X 16bit 乘法操作时 (MD[1:0] = 00) 存放另一个 16bit 乘数, DB15 为 MSB, DB0 为 LSB
01	作 32bit ÷ 16bit 除法操作时 (MD[1:0] = 01) 存放 16bit 除数, DB15 为 MSB, DB0 为 LSB
10	作 32bit 位移操作时 (MD[1:0] = 10) 存放位移的次数 (此时只有 DB[4:0] 位有效) 例: DB[4:0] = 1; 移位 1 次
-	运算完成时 (START = 0) 存放 16bit 余数, DB15 为 MSB, DB0 为 LSB



8 定时器/计数器

8.1 定时器 T0/T1

芯片包含 2 个兼容 8051 标准的 16 位定时器 T0 和 T1，定时器的两个 8 位数据寄存器 (THx\TLx, x=0, 1) 可作为一个 16 位寄存器来访问。

定时器具有 4 种工作方式，通过定时器模式寄存器的方式选择位 (TxM[1:0]) 来选择定时器 Tx 的工作方式。

8.1.1 方式 0：13 位计数器/定时器

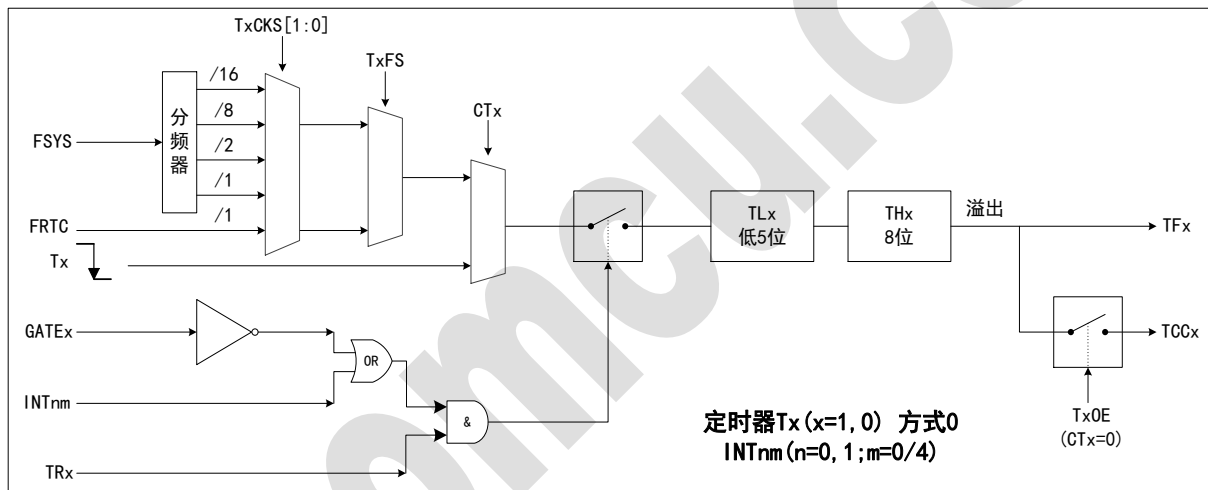
在方式 0 中，定时器 Tx (x=0, 1) 为 13 位计数器/定时器。递增计数寄存器 THx[7:0] 存放 13 位计数器/定时器的高 8 位，TLx[4:0] 存放低 5 位，而 TLx[7:5] 的值不确定，在读取时应被忽略。当 13 位计数寄存器溢出时，溢出标志 TFx 将被置 1，如果 Tx 中断被允许，将会产生定时器 Tx 中断。

CTx 位选择计数器/定时器 Tx 的时钟源。如果 CTx = 1，Tx 用作计数器，Tx 引脚将被设置成输入端口（需其他复用功能未复用该端口），Tx 的下降沿将使 Tx 计数寄存器加 1；如果 CTx = 0，Tx 用作定时器，可选择低频时钟 FRTC 或系统时钟 FSys（或其分频）作为 Tx 的时钟源。

注：用户配置字 FMOD，控制 Timer0/1 的低频时钟源，若 FMOD 选择时钟包含 FLOSC，则选择 FLOSC 作为 Timer0/1 时钟源，否则选择 FLIRC 作为时钟源。

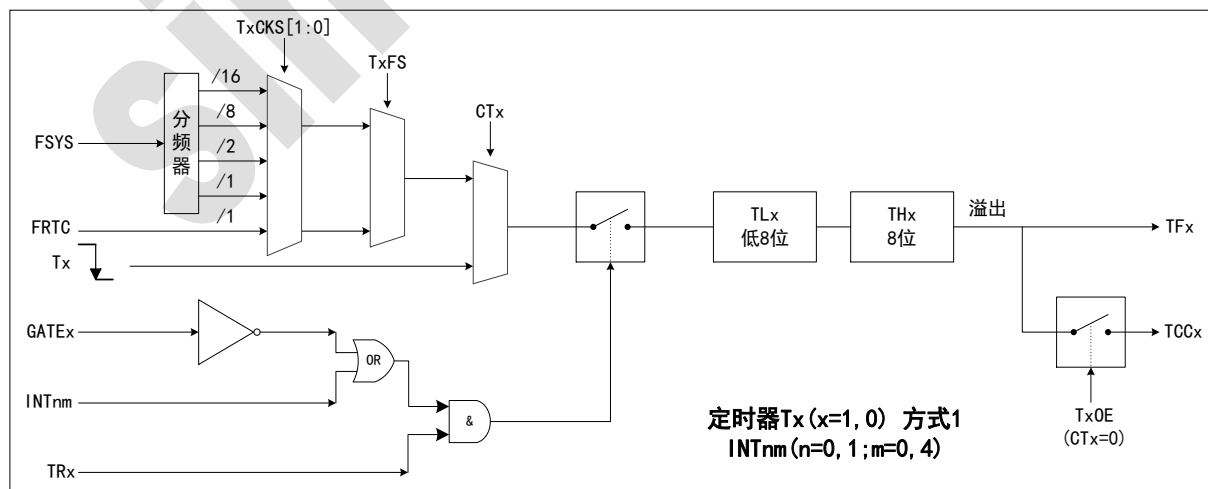
当 GATEx = 0，或者 GATEx = 1 且 INTnm 有效时，TRx 置 1 将启动定时器。GATEx 置 1，允许定时器 Tx 由外部输入信号 INTnm 控制，用于测量 INTnm 引脚上输入高电平的脉冲宽度。TRx 位置 1 不强行复位定时器，即 TRx 置 1 时计数寄存器将从上次 TRx 清 0 时的值恢复计数。所以在启动定时器之前应设定计数寄存器的复位值。

当 Tx 用作定时器时，可设置寄存器 TCON1 中的 TxOE 位使 Tx 计数值溢出时可以对 Tx 引脚电平自动翻转。仅当 TxOE 被置 1，Tx 引脚自动设置为输出有效（需其他复用功能未复用该端口）。



8.1.2 方式 1：16 位计数器/定时器

除了使用 16 位计数寄存器 THx/TLx 外，方式 1 的运行与方式 0 相同，打开和设置计数器/定时器也与方式 0 相同。

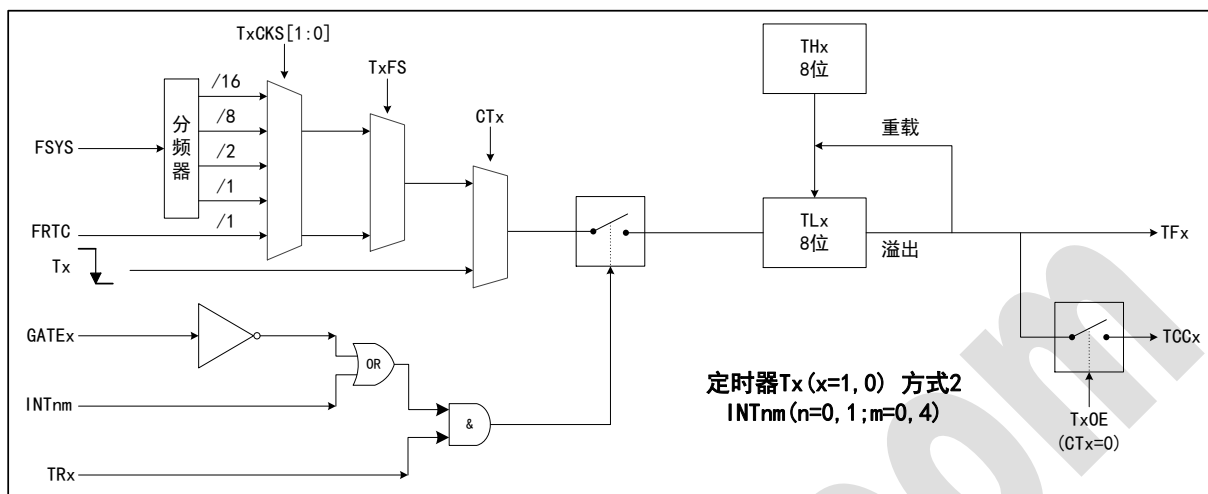


8.1.3 方式 2：8 位自动重载计数器/定时器

方式 2 中，定时器 Tx 用作 8 位自动重载计数器/定时器。TLx 存放计数值，THx 存放重载值。当递增计数器 TLx 溢

出至 0x00 时, 溢出标志 TFx 将被置 1, 寄存器 THx 的值重新载入计数器 TLx 中, 而 THx 的重载值不变。如果 Tx 中断被允许, 将会产生定时器 Tx 中断。在此方式下, 首次打开定时器开始计数之前, 应将 TLx 初始化为重载值, 确保首次溢出时间与后续一致。

仅 T1 可以用作 UART0 的波特率发生器，T0 不可作为波特率发生器。除自动重载功能以外，方式 2 中的计数器/定时器的使能和设置与方式 1 和 0 基本相同。



在 UART0 方式 1 和方式 3 中波特率可由该方式提供, 波特率的计算公式如下:

$$\text{T1产生的波特率} = \frac{1}{16} \times \frac{\text{Ft1}}{[256 - (\text{TH1})]}$$

其中 F_{T1} 为 T1 实际的工作频率, TH1 为实际设定值。

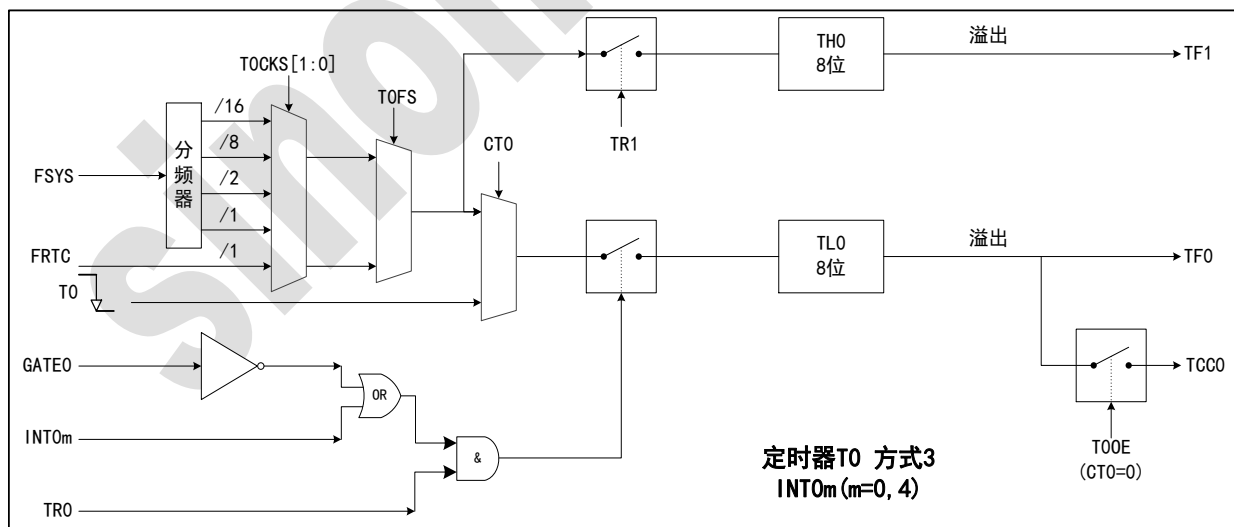
注：当用定时器 T1 作为波特率发生器时，T1 的 TH1 值不可设置为 0xFF，并应该同时关闭相应 T1 的中断使能位。

8.1.4 方式3: 两个8位计数器/定时器 (只限于定时器 T0)

在方式 3 中, 定时器 T0 用作两个独立的 8 位计数器/定时器, 分别由 TL0 和 TH0 控制。TL0 使用定时器 T0 的控制 (TCON 中) 和状态 (TMOD 中) 位: TR0、CT0、GATE0 和 TF0。TL0 可采用振荡器时钟信号用作定时器, 也可采用外部输入信号用作计数器。

TH0 只能用作定时器功能, 时钟来自振荡器时钟信号。TH0 借用定时器 T1 的控制位 TR1 控制使能, 溢出时借用定时器 T1 溢出标志 TF1 置 1, 产生 T1 中断。

当 T0 工作在方式 3 时, 定时器 T1 可以工作在方式 0、1 或 2, 但是无法将 TF1 标志置 1, 无法产生中断。TH1 和 TL1 只能用作定时器功能, 时钟源来自振荡时钟信号, GATE1 无效。因为 TR1 被定时器 0 占用, 所以定时器 T1 改由方式设置来控制使能/关闭。设为方式 0/1/2 时 T1 使能, 设为方式 3 时 T1 被关闭。



8.2 T0/T1 相关寄存器

T0/T1 寄存器汇总表

地址	寄存器	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位
----	-----	----	-------	-------	-------	-------	-------	-------	-------	-------	----



(SFR n/H)	名										值 (B)
SFR0/8E	TCON1	定时器控制寄存器 1	T1FS	T1OE	T1CKS1	T1CKS0	T0FS	T0OE	T0CKS1	T0CKS0	00000000
SFR0/8D	TH1	定时器计数高位寄存器 1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10	00000000
SFR0/8C	TH0	定时器计数高位寄存器 0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00	00000000
SFR0/8B	TL1	定时器计数低位寄存器 1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10	00000000
SFR0/8A	TL0	定时器计数低位寄存器 0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00	00000000
SFR0/89	TMOD	定时器模式寄存器	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0	00000000
SFR0/88	TCON	定时器控制寄存器	TF1	TR1	TF0	TR0	-	-	-		0000---0

8.2.1 定时器模式寄存器 (TMOD, 0x89/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMOD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7、3] GATE_x - 定时器/计数器 Tx 门控制位 (x=1,0)

0: TR_x 置 1 时, 定时器 Tx 立刻开始计数;

1: TR_x 置 1 且外部 INT_{xn} (x=1,0 与 Tx 保持一致; n=4) 信号同时有效 (具体可参考中断章节内容), 定时器 Tx 才开始计数。

注: 当 GATE_x=1 时, T0/T1 只能通过 INTO4 和 INT14 测量外部信号的脉宽, 且仅测量高电平的脉宽。

BIT[6、2] CT_x - Tx 的计数器/定时器方式选择位 (x=1,0)

0: 定时器方式, 振荡器时钟信号触发, 即内部有一个 clock, Tx 的计数器递增一次;

1: 计数器方式, 外部输入下降沿触发, 即 Tx 端口上一个下降沿的有效信号, Tx 的计数器递增一次, 此时 Tx 口作为 Tx 定时器/计数器的专用输入口, 且不能做其它复用功能。

BIT[5:4] T1M[1:0] - 定时器/计数器 T1 工作方式选择位

T1M[1:0]	定时器 T1 工作方式
00	方式 0
01	方式 1
10	方式 2
11	禁止 (T1 将停止工作)

注: T1 没有工作方式 3, 用户强制使用 T1 的方式 3, T1 将停止工作。

BIT[1:0] T0M[1:0] - 定时器/计数器 T0 工作方式选择位

T0M[1:0]	定时器 T0 工作方式
00	方式 0
01	方式 1
10	方式 2
11	方式 3

8.2.2 定时器控制寄存器 (TCON, 0x88/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	-	-	-	
R/W	R/W	R/W	R/W	R/W	-	-	-	
复位值	0	0	0	0	-	-	-	

BIT[7、5] TF_x - 定时器/计数器 Tx 溢出标志位 (x=1,0)

0: 定时器/计数器 Tx 无溢出, 当中断响应后硬件清 0, 可由软件清 0;

1: 定时器/计数器 Tx 溢出时, 由硬件置 1。

BIT[6、4] TR_x - 定时器/计数器 Tx 使能位 (x=1,0)

0: 定时器/计数器 Tx 中止递增, TH_x 和 TL_x 寄存器的数值将停在“中止”时候的数值;

1: 定时器/计数器 Tx 开始递增, 或者 (TH_x 和 TL_x 寄存器不重新赋值) 继续递增。

8.2.3 定时器控制寄存器 1 (TCON1, 0x8E/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON1	T1FS	T1OE	T1CKS1	T1CKS0	T0FS	T0OE	T0CKS1	T0CKS0



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7、3] **TxFS** - 定时器/计数器方式时, Tx 时钟源选择位 ($x=1, 0$)

0: Tx 时钟源为系统时钟 Fsys;

1: Tx 时钟源为低频时钟 F_{RTC};

BIT[6、2] **TxOE** - 定时器/计数器 Tx 输出使能位 ($x=1, 0$) (CTx=0 时有效)

0: 禁止 Tx 溢出时翻转 Tx 端口电平, 端口用于通用 I/O 或其他复用模块;

1: 使能 Tx 端口复用于定时器 Tx, Tx 溢出时翻转端口电平, 并且第一次反转输出的是高电平 (逻辑 1), 第二次反转输出的是低电平 (逻辑 0), 之后依次交替反转。

BIT[5:4, 1:0] **TxCKS[1:0]** - 定时器/计数器 Tx ($x=1, 0$) 时钟分频选择位

TxCKS[1:0]	Tx 频率 (TxFS=0)	Tx 频率 (TxFS=1)
00	Fsys	F _{RTC}
01	Fsys/2	禁止使用
10	Fsys/8	
11	Fsys/16	

注: Tx 的工作频率 F_{Tx} 与用户选择的实际频率和分频比有关; 在低频 (TxFS=1) 的情况下, 用户仅有一挡 (TxCKS[1:0]=00) 可选择, 若强行选择了“禁止”状态, Tx 将出现不可预知的情况。

8.2.4 定时器计数高位寄存器 0 (TH0, 0x8C/SFR0) 和定时器计数高位寄存器 1 (TH1, 0x8D/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
THx	THx7	THx6	THx5	THx4	THx3	THx2	THx1	THx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] **THx[7:0]** - 定时器/计数器 Tx 高 8 位计数器 ($x=1, 0$)

8.2.5 定时器计数低位寄存器 0 (TL0, 0x8A/SFR0) 和定时器计数低位寄存器 1 (TL1, 0x8B/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TLx	TLx7	TLx6	TLx5	TLx4	TLx3	TLx2	TLx1	TLx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] **TLx[7:0]** - 定时器/计数器 Tx 低 8 位计数器 ($x=1, 0$)

注 1: 无论是读或者写 TLx/THx 的值, 必须先停止 (TRx=0), 否则 Tx 将产生不可预知的情况。

注 2: 写操作时: 必须先写 TLx, 再写 THx; 否则 TLx 的值不能立即反映到芯片上; 读操作时: 必须先读 TLx, 再读 THx, 否则将读取到错误的值。

8.3 定时器 T2

芯片包含 1 个 16 位增强型定时器 T2, 兼容 52 系列中标准型 T2, 但功能更强。T2 是一个向上递增型定时器/计数器。T2 具有 4 种工作在模式: 自动重载模式、捕获模式、波特率发生器模式、比较模式。

T2 工作方式	T2CON1		T2CON
	Bit 6	Bit 5	Bit 0
	CM2	BR2EN	CPRL2
方式 0: 16 位重载方式	0	0	0
方式 1: 16 位捕获方式	0	0	1
方式 2: 16 位波特率发生器方式	0	1	0
方式 3: 16 位比较方式	1	0	0
无效状态	其他值		

8.3.1 方式 0: 16 位重载方式定时器/计数器方式

T2 的这种工作方式和 T0/T1 的工作方式 2 类似, 只不过 T0/T1 是 8 位的, 而 T2 则是 16 位的。

T2 的自动重载模式中, 时钟源由 CT2 位决定是对 Fsys 计数, 还是对 T2 端口输入信号计数。RCAPH2 和 RCAPL2 保存着重载模式的数值。一旦产生了重载触发事件, 硬件会自动把 RCAPH2 和 RCAPL2 中的值分别装入 TH2 和 TL2 中。重载触发事件指, T2 定时器溢出 (TF2=1) 或者是外部捕获信号有效 (T2EX 端口出现有效信号, EXTF2=1), 两者都可产生 T2 中断, 并重新装载 TH2/TL2, 使其开始新一轮的计数。

标准情况: 使能 T2, 将 T2 设置为方式 0, 设置时钟频率、定时器模式 (CT2=0) 等等参数。启动 T2 (TR2=1), 打开重载功能 LD2EN=1, 且禁止外部捕获信号 T2EX 输入 (EX2EN=0), 该情况不产生外部触发事件 (T2EX 端口无有效信号,

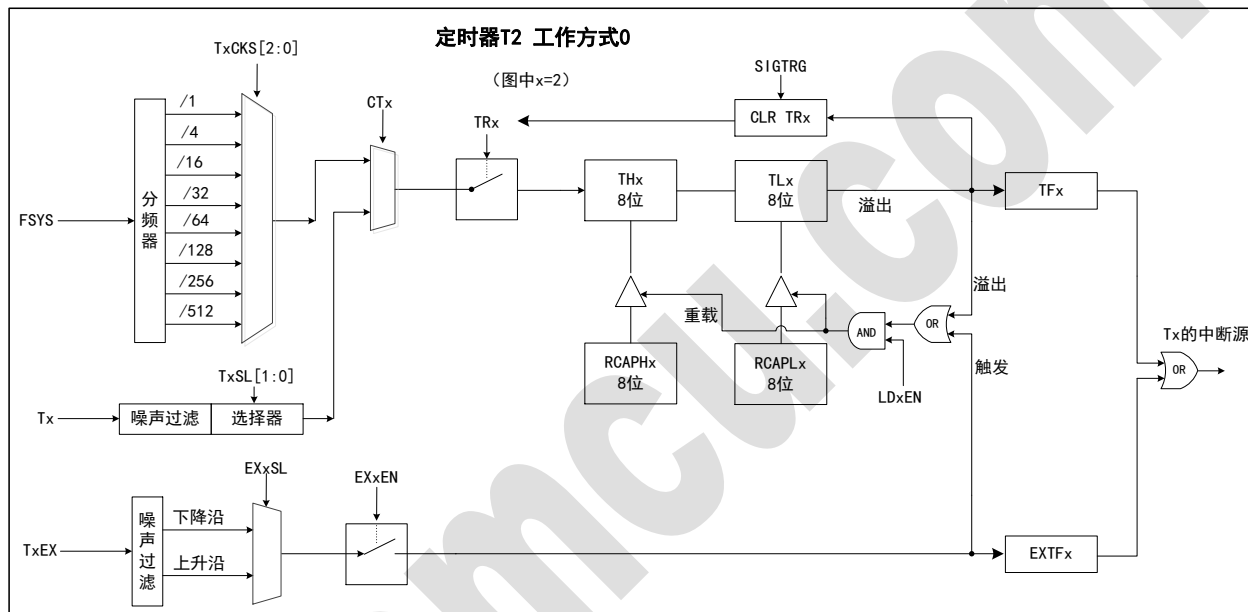


始终是 $EXTF2=0$), 仅通过 T2 定时器产生溢出 ($TF2=1$) 产生触发事件的情况。向上递增计数中, $TH2/TL2$ (从 $RCMPH2/RCMPL2$ 装载) 的值, 开始递增计数直到 $0x0$ 产生溢出, 最终产生 $TF2$ 标志位置位 ($TF2=1$) 和 T2 中断。

特殊情况一: 用户觉得内部时钟源不好, 可以改变时钟源。当 $CT2=1$ 时, 时钟源切换到外部, 通过 T2 口送入。用户还可通过 $T2SL$ 位, 决定有效信号的方式, 当 $T2SL=01B$ 时上升沿有效, 当 $T2SL=10B$ 时下降沿有效, 当 $T2SL=11B$ 时双沿有效。当选择外部时钟时, 由于内部系统时钟采样同步, 外部最大不能超过 $F_{sys}/2$ 。

特殊情况二: 不重载或者 T2EX 加入触发重载控制。当 $LD2EN$ 位=0 关闭重载功能, 则 T2 仅就做一次定时任务, T2 的溢出和外部 T2EX 的有效信号都不再重载 $TH2$ 和 $TL2$ 。若用户需要再次开启 T2 的方式 0, 则先让 $TR2=0$ 停止 T2, 对 $RCAPH2/RCAPL2$ 重新赋值后 (对 $TH2/TL2$ 进行赋值无效), 让 $TR2=1$ 再次启动 T2, T2 方可正常工作。当 $T2OE=1$, T2EX 端口的信号可以参与触发重载, 外部触发信号与 T2 溢出的信号同一优先级。若 T2EX 有效信号和 T2 溢出信号同时来到, 两者都会产生中断, 且 2 个标志位的产生, 就意味着会有 2 次中断请求, 用户需要根据自己的应用编写不同的应用程序, 以满足自己的实际需要。 $EX2SL$ 控制有效信号的类型, 0 为下降沿有效, 1 为上升沿有效。当通过外部 T2EX 触发重载了 $TH2/TL2$ 并产生中断, 其中断标志位是 $EXTF2$, 而不是 $TF2$, 用户可以通过查询 $EXTF2$ 标志位/ $TF2$ 标志位, 具体识别本次 T2 中断事件的类型。

特殊情况三: 当 $SIGTRG$ 为 0 时, 定时器定时溢出后会自动重载继续定时 ($TR2$ 不会变为 0); 当 $SIGTRG$ 为 1 时, 定时器定时溢出后会自动停止定时器 (将 $TR2$ 变为 0), 可以利用这个方式将 PWM1, TIMER2 和 ADC 组成一个事件级联 (如: PWM1 周期触发 TIMER2 单次定时, TIMER2 定时溢出触发 ADC)



8.3.2 方式 1: 16 位输入捕获方式

输入捕获功能, 主要用于测量脉冲宽度或周期。在 T2 的捕获方式中, 时钟源由 $CT2$ 位决定是对 F_{sys} 或其分频计数, 还是对 T2 端口输入信号计数。T2 的 T2EX 端口的输入由 $EX2EN$ 位控制, 当在该方式下打开了 T2EX 端口后, 芯片始终通过监视并捕获 T2EX 端口上的有效信号。 $RCAPH2$ 和 $RCAPL2$ 保存最近一次捕获的结果。一旦产生了输入捕获事件, 硬件自动把 $TH2$ 和 $TL2$ 中的当前值分别送到 $RCAPH2$ 和 $RCAPL2$ 中保存。输入捕获事件指, T2EX 上产生一个有效的输入捕获信号有效, 并被芯片识别 (T2 硬件对 $EXTF2$ 标志位置 1) 的过程。注意若捕获事件发生前的事件间隔超过最大计数, 同样会产生 T2 定时器的溢出 ($TF2$ 标志位置 1) 引起 T2 中断, 但用户可以通过标志位的不同进行区分。另外, 当 $RCAPH2$ 和 $RCAPL2$ 中保存了 $TH2$ 和 $TL2$ 的溢出值 ($0x0$), 也是一种异常情况, 用户应该做特别处理。

标准情况: 使能 T2, 将 T2 设置为方式 1, 设置时钟频率、定时器模式 ($CT2=0$) 等等参数。开启 T2EX 端口 ($EX2EN=1$), 并选择捕获极性为上升沿 $EX2SL=1$ (下降沿 $EX2SL=0$) 有效, 并持续对其进行监视。启动 T2 ($TR2=1$), $TH2/TL2$ 从 $0x0$ 向上开始计数。当 T2EX 端口上产生有效的触发信号 ($EXTF2=1$) 时, 硬件立即将 $TH2/TL2$ 的值保存到 $RCAPH2/RCAPL2$ 中, 同时产生 $EXTF2$ 标志位和 T2 中断。每一次的 T2EX 有效信号都会触发一次捕获, 并覆盖上一次的 $RCAPH2/RCAPL2$ 值。

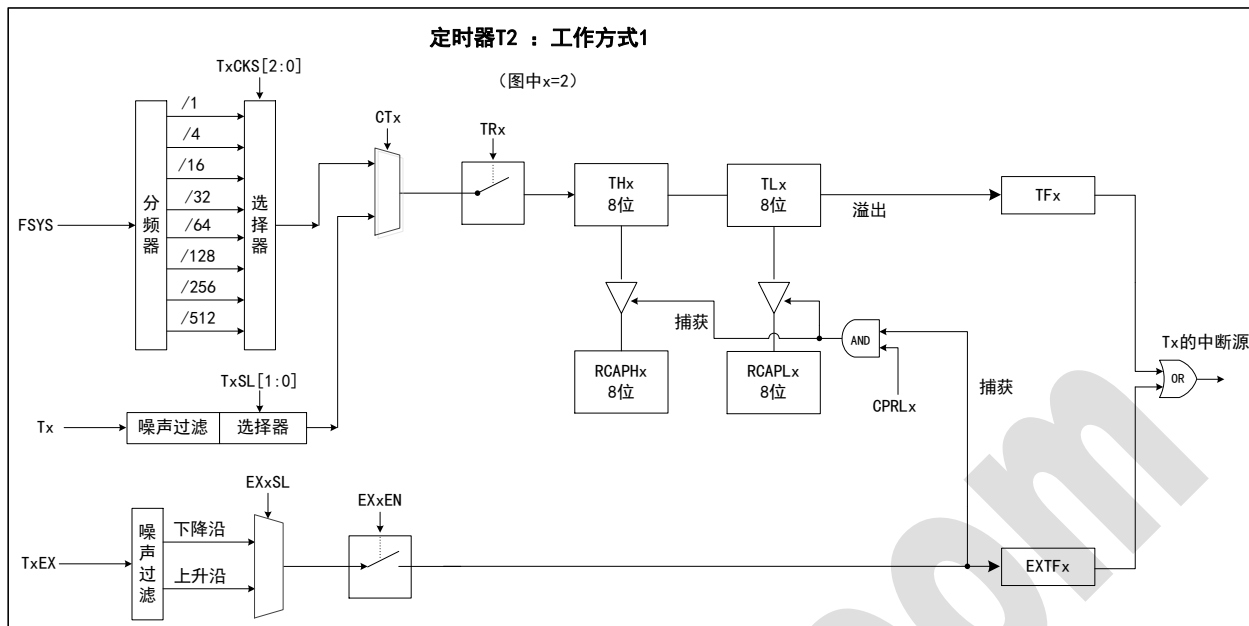
特殊情况一: 用户觉得内部时钟源不好, 可以改变时钟源。当 $CT2=1$ 时, 时钟源切换到外部, 通过 T2 口送入。用户还可通过 $T2SL$ 位, 决定有效信号的方式, 当 $T2SL=01B$ 时上升沿有效, 当 $T2SL=10B$ 时下降沿有效, 当 $T2SL=11B$ 时双沿有效。当选择外部时钟时, 由于内部系统时钟采样同步, 外部最大不能超过 $F_{sys}/2$ 。

特殊情况二: 假设测量 T2EX 端口上的一个低电平, 但由于测量的波形时间特别长, 且超过了 T2 的最大计数时间。T2 启动 ($TR2=1$) 开始计数, T2EX 端口下降沿执行第一次捕获, 上升沿未到达之前, 将产生 T2 溢出, 发生 $TF2$ 中断。用户可以在进入中断后通过标志位判定中断原因, 并记录溢出次数, 等待 T2EX 上的触发信号到来。这样, 用户的中断处理函数增加一定溢出处理, 便可以可以实现长时间间隔的有效沿捕获。假设, 首次触发捕获后, 经过了 3 个溢出时间中断后, 有效的上升沿到来, 那么本次测量 2 次有效沿宽度的时间计数值如下:

有效沿宽度= $RCAP1$ (最后一次中断取得的 $RCAPH2$ 和 $RCAPL2$) $+ 0xFFFF * 3$ (前 3 次 T2 的溢出时间) $- RCAP0$ (首次捕获



值 RCAPH2 和 RCAPL2)。



8.3.3 方式 2：16 位波特率发生器方式

通过设置 RCLKU0 位和 TCLKU0 位来选择 T2 定时器作为 UART0 的波特率发生器。UART0 的接收器和发送器可以同时使用 T1 或者 T2；UART0 的接收器和发送器也可以设置不同的波特率，但 T2 同一时刻只可以提供一种波特率，若用户同时需要两种不同的波特率，可以共同使用 T2 与 T1 配合实现。T1 和 T2 物理连接 UART0 的控制位，分别通过 RCLKU0、TCLKU0 位控制，因为兼容性的关系这些控制位都分布在 T2 的控制寄存器中。对 UART0 的配置，请用户参考相关章节的内容。

T2 作为波特率发生器模式，LD2EN 控制无效，保持重载方式，溢出将不会发生溢出中断，T2EX 可以作为独立的外部中断使用。

在此方式下，若 EX2EN 置位（外部信号有效），则当 T2EX 端口上出现有效的信号，仍会产生 EXTF2 标志位和 T2 中断，但不会引起 TH2/TL2 的重载和重新计数。此模式下，T2EX 可以作为独立的外部中断功能使用。

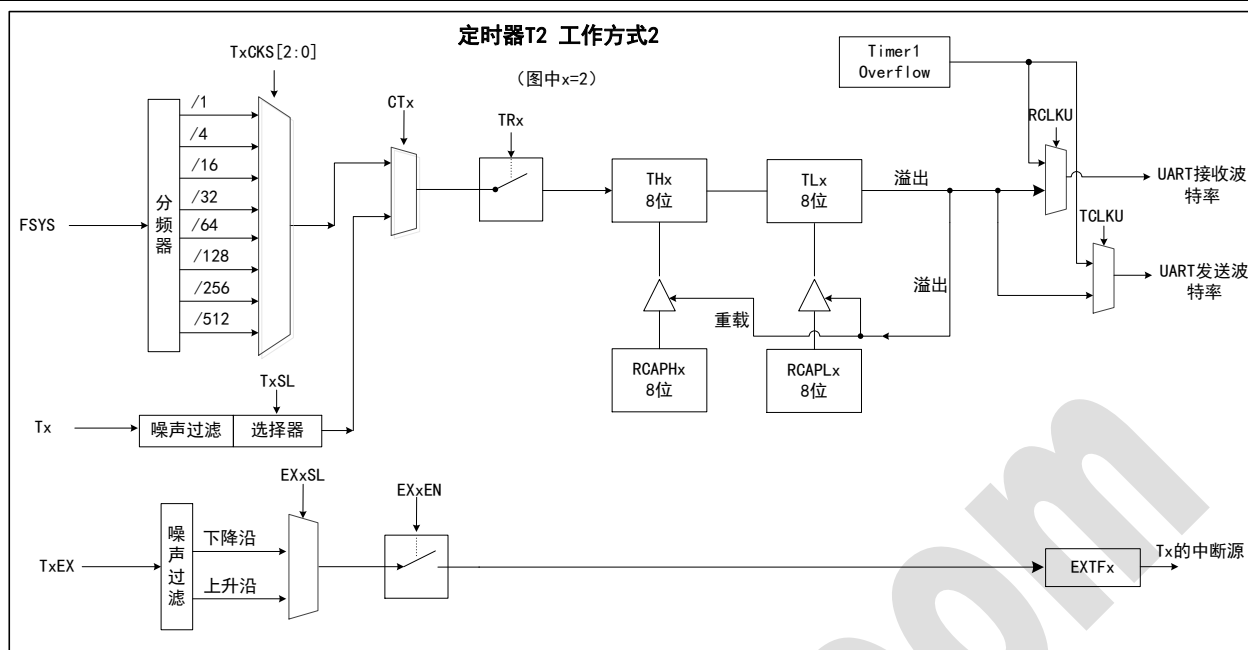
特殊情况一：用户觉得内部时钟源不好，可以改变时钟源。当 CT2=1 时，时钟源切换到外部，通过 T2 口送入。用户还可通过 T2SL 位，决定有效信号的方式，当 T2SL=01B 时上升沿有效，当 T2SL=10B 时下降沿有效，当 T2SL=11B 时双沿有效。当选择外部时钟时，由于内部系统时钟采样同步，外部最大不能超过 Fsys/2。

在 UART0 方式 1 和方式 3 中波特率可由该方式提供，波特率的计算公式如下：

$$\text{T2产生的波特率} = \frac{1}{16} \times \frac{\text{Fsys}}{\{\text{Pre_scale} \times [65536 - ((\text{RCAPH2}) \times 256 + (\text{RCAPL2}))]\}}$$

其中 n 为 0 或者 1，Fsys 为系统时钟频率，预分频值 Pre_scale 为 T2 时钟选择位 T2CKS[2:0]的分频系数，RCAPH2/RCAPL2 为实际设定值。

注：当用定时器 T2 作为波特率发生器时，T2 的 TH2/TL2 和 RCAPH2/RCAPL2 值不可设置为 0xFFFF；同时，应该关闭 T2 的中断使能位，不让 T2 进入中断。



8.3.4 方式3：16位输出比较方式

在T2的比较模式中，时钟源仅来源于对Fsys或其分频，T2EX端口作输入无效。T2端口仅作输出使用（若用户不想输出也可以不输出），不可作为时钟输入，但T2端口最终是否输出，可由T2OE位控制，且该端口的输出初始电平值由T2SL[1:0]位控制。在T2端口允许输出的前提下，每次比较事件后T2端口上可以指定输出一个高/低电平（CM2ON=0，初始电平由T2SL决定），或者控制其翻转一次电平（CM2ON=1）。

比较模式下，T2从0开始计数，TH2和TL2计数计到与RCMPH2和RCMPL2的值相同后，发生输出比较匹配事件，硬件产生一次TF2置位的过程，若LD2EN=1，则TH2和TL2重载0x0继续计数。若LD2EN=0，TH2/TL2计数至比较匹配时不自动重载。当中断打开，每次比较匹配（TF2置位）将触发中断。输出比较功能，可用于产生类方波。

标准情况：使能T2，将T2设置为方式3，设置时钟频率、定时器模式（CT2=0）等等参数。重载有效（LD2EN=1），T2启动（TR2=1），向上递增计数时，TH2/TL2从0x0开始计到RCMPH2/RCMPL2中的值产生比较匹配事件，则TF2标志位置位（TF2=1）和产生T2中断，并将0x0载入TH2/TL2重新开始计数；允许外部输出T2OE=1，且当T2SL0=0时，CM2ON=1，第一次比较匹配事件发生后T2端口上的电平由初始低电平翻转至高电平，第二次比较匹配事件后则由高电平翻转至低电平，之后交替翻转。

特殊情况：重载无效（LD2EN=0），T2启动（TR2=1），基本同“标准情况”，每次比较匹配事件发生后，T2端口上输出的电平翻转一次后保持不变。当T2OE=0时，将没有输出，T2端口可用作其它复用功能，这样的话仅实现了一个定时中断功能。

注：初始电平的配置（T2SL0和CM2ON）会立即生效，T2OE开启，对应的电平状态即会传递至GPIO，若要实现更准确的单脉冲输出，请配置GPIO电平为预期脉冲电平的相反电平，T2SL0和CM2ON配置脉冲初始电平，然后尽可能的同时启动T2OE和TR2。



地址 (SFR n/H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
SFR0/ CE	T2CON1	T2 控制寄存器 1	CM2ON	CM2	BR2EN	RCLKU1	TCLKU1	EX2SL	SIGTRG	T2EN	0000 000
SFR0/ CD	TH2	T2 计数高位寄 存器	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20	00000 000
SFR0/ CC	TL2	T2 计数低位寄 存器	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20	00000 000
SFR0/ CB	RCAPH2	T2 重载/捕获高 位寄存器	RCAPH27	RCAPH26	RCAPH25	RCAPH24	RCAPH23	RCAPH22	RCAPH21	RCAPH20	00000 000
SFR0/ CA	RCAPL2	T2 重载/捕获低 位寄存器	RCAPL27	RCAPL26	RCAPL25	RCAPL24	RCAPL23	RCAPL22	RCAPL21	RCAPL20	00000 000
SFR0/ C9	T2MOD	工作模式寄存 器	LD2EN	T2CKS2	T2CKS1	T2CKS0	T20E	T2SL1	T2SL0	-	00000 00-
SFR0/ C8	T2CON	T2 控制寄存器	TF2	EXTF2	RCLKU0	TCLKU0	EX2EN	TR2	CT2	CPRL2	00000 000

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	TF2	EXTF2	RCLKU0	TCLKU0	EX2EN	TR2	CT2	CPRL2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

1: UART0 用 T2 溢出率作为发送波特率。



注：当 UART 的波特率选择用 T1 或者 T2 作为接收用时，用户应当避免 T1 或者 T2 再做他用。

BIT[3] **EX2EN** - T2EX 端口作为重载触发/输入捕获使能位

0: 禁止 T2EX 端口作为重载触发/输入捕获功能，或者用于通用 IO 或其他复用模块（由 T2CON 的 C/T2 位决定）；

1: 使能 T2EX 端口作为重载触发/输入捕获功能，具体是重载触发还是输入捕获功能，由此时 T2 的具体工作方式决定（工作方式 0=重载触发；工作方式 1=输入捕获），激活 T2 执行一次重载触发/输入捕获的事件。

注：有效触发的形式，由 EX2SL 位决定，即 EX2SL=0 为下降沿触发，EX2SL=1 为上升沿触发。

BIT[2] **TR2** - 定时器/计数器 T2 使能位

0: 停止定时器/计数器 T2；

1: 开始定时器/计数器 T2，或者再次写入 1 后复位 T2，让 T2 重新开始计数，即定时器将从 0x0/0xFFFF 开始向上/向下计数，具体计数方向视 DC2EN 位而定。

BIT[1] **CT2** - T2 的计数器/定时器方式选择位

0: T2 工作于定时器方式，T2 的时钟源来自内部；

1: T2 工作于计数器方式，T2 端口作为 T2 的外部时钟源专用输入口，且不能做其它复用功能。

BIT[0] **CPRL2** - 定时器/计数器 T2 重载触发/输入捕获方式选择位

0: T2 工作于带重载触发功能的定时器/计数器方式；

1: T2 工作于带输入捕获功能的定时器/计数器方式。

T2 工作方式整理表

T2 工作方式	T2CON1		T2CON
	Bit 6	Bit 5	Bit 0
	CM2	BR2EN	CPRL2
方式 0: 16 位重载方式	0	0	0
方式 1: 16 位捕获方式	0	0	1
方式 2: 16 位波特率发生器方式	0	1	0
方式 3: 16 位比较方式	1	0	0
无效状态	其他值		

注：当 CM2、BR2EN、CPRL2 同时置位时，比较模式优先于波特率模式优先于捕获模式。

8.4.2 T2 工作模式寄存器（T2MOD, 0xC9/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2MOD	LD2EN	T2CKS2	T2CKS1	T2CKS0	T2OE	T2SL1	T2SL0	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
复位值	0	0	0	0	0	0	0	0

BIT[7] **LD2EN** - 定时器/计数器 T2 溢出重载控制位

0: 在重载模式（方式 0/方式 1/方式 3）中，T2 溢出后不自动重载 TH2/TL2；

1: 在重载模式（方式 0/方式 1/方式 3）中，T2 溢出后自动重载 TH2/TL2，使能后将连续重载。

BIT[6:4] **T2CKS[2:0]** - 定时器/计数器 T2 时钟分频选择位

T2CKS[2:0]	T2 频率 (CT2M=0)
000	Fsys
001	Fsys/4
010	Fsys/16
011	Fsys/32
100	Fsys/64
101	Fsys/128
110	Fsys/256
111	Fsys/512

BIT[3] **T2OE** - 定时器/计数器 T2 输出使能位

0: 禁止 T2 溢出时翻转 T2 端口电平，端口用于通用 IO 或其他复用模块；

1: 使能 T2 端口复用于定时器 T2，T2 溢出时翻转一次端口电平，初始电平可由 T2SL[1:0]控制。

BIT[2:1] **T2SL[1:0]** - T2 端口输入/输出信号选择位

T2SL[1:0]	T2 端口有效输入信号选择 ^{注1}	T2 端口初始输出信号选择 ^{注2}
00	无效	禁止
01	上升沿有效	高电平
10	下降沿有效	低电平
11	双边沿同时有效	禁止



注 1: 仅当 C/T2 位置 1 后, T2 端口上的输入信号才受控于该控制位。

注 2: 仅在输出比较的情况下, 即“工作方式 3”, 且 T2OE=1 时, 该位决定了 T2 端口的初始电平值, 若输出比较模式中, 用户需要了解当前 T2 端口的实际电平值, 可以通过查询对应的 (P2.0) 端口的数据值获得。

8.4.3 T2 控制寄存器 1 (T2CON1, 0xCE/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON1	CM2ON	CM2	BR2EN	=	=	EX2SL	SIGTRG	T2EN
R/W	R/W	R/W	R/W	=	=	R/W	R/W	R/W
复位值	0	0	0	=	=	0	0	0

BIT[7] CM2ON - 比较方式下 T2 端口输出电平翻转控制位

0: T2 端口在比较模式 (方式 3) 下, 每次触发后 T2 端口自动翻转一次输出电平;

1: T2 端口在比较模式 (方式 3) 下, 每次触发后 T2 端口输出 (初始) 电平保持不变。

注: T2 端口上的初始电平由 T2SL 决定。

BIT[6] CM2 - 比较方式选择位

0: T2 工作于 (方式 0、方式 1、方式 2) 非 16 位输出比较模式;

1: T2 工作于 (方式 3) 16 位输出比较模式。

BIT[5] BR2EN - UART 波特率发生器模式使能位

0: T2 工作于 (方式 0、方式 1、方式 3) 非波特率发生器模式;

1: T2 工作于 (方式 2) 波特率发生器模式。

BIT[2] EX2SL - T2EX 端口输入捕获/重载触发信号有效沿/电平选择位

0: T2EX 端口作为输入捕获或者重载触发用时, 下降沿为有效输入信号;

1: T2EX 端口作为输入捕获或者重载触发用时, 上升沿为有效输入信号。

EX2SL	(当 EX2EN=1) T2EX 端口有效输入信号选择
1	上升沿有效
0	下降沿有效

BIT[1] SIGTRG - Timer2 单次计数使能位

0: 连续计数模式;

1: 单次计数模式。

BIT[0] T2EN - 定时器/计数器 T2 模块使能位

0: 关闭 T2 模块;

1: 打开 T2 模块。

8.4.4 T2 计数高位寄存器 (TH2, 0xCD/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] TH2[7:0] - 定时器/计数器 T2 高 8 位计数器设定值

8.4.5 T2 计数低位寄存器 (TL2, 0xCC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] TL2[7:0] - 定时器/计数器 T2 低 8 位计数器设定值

注 1: 无论是读或者写 TL2/TH2 的值, 必须先停止 (TR2=0), 否则 T2 将产生不可预知的情况。

注 2: 写操作时: 必须先写 TL2, 再写 TH2; 否则 TL2 的值不能立即反映到内部 counter 上; 读操作时: 必须先读 TL2, 再读 TH2, 否则将读取到错误的值。

8.4.6 T2 重载/捕获高位寄存器 (RCAPH2, 0xCB/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCAPH2	RCAPH27	RCAPH26	RCAPH25	RCAPH24	RCAPH23	RCAPH22	RCAPH21	RCAPH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] RCAPH2[7:0] - 定时器/计数器 T2 重载/捕获高 8 位寄存器

8.4.7 T2 重载/捕获低位寄存器 (RCAPL2, 0xCA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	-------	-------	-------	-------	-------	-------	-------	-------



RCAPL2	RCAPL27	RCAPL26	RCAPL25	RCAPL24	RCAPL23	RCAPL22	RCAPL21	RCAPL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] RCAPL2[7:0] - 定时器/计数器 T2 重载/捕获低 8 位寄存器

注 1: 无论是读或者写 RCAPL2/RCAPH2 的值, 必须先停止 ($TR2=0$), 否则 T2 将产生不可预知的情况。

注 2: 写操作时: 必须先写 RCAPL2, 再写 RCAPH2; 否则 RCAPH2 的值不能立即反映到内部 counter 上; 读操作时: 必须先读 RCAPL2, 再读 RCAPH2, 否则将读取到错误的值。。

8.5 定时器 T3

芯片包含 1 个 16 位基础定时器 T3, T3 是一个向上递增型定时器/计数器, 可选择系统时钟 (同步时钟) 或低频时钟 FRTC (异步时钟)。

8.5.1 方式 0: 系统时钟 (同步时钟) 模式

当 T3 作定时使用时, 可通过 CT3 位 ($=0$) 设定, T3 选择系统时钟 (同步时钟) 模式。此时, 其时钟源为系统时钟或其分频, 通过 T3CKS 设置, 可选择多种分频 (具体值参考 T3 控制寄存器说明); 启动 TR3, T3 从 TL3/TH3 设定的初值开始递增计数, 溢出后产生 T3 中断, 并重新加载 TL3/TH3 设定的初值重新开始递增计数。

8.5.2 方式 1: 低频时钟 (异步时钟) 模式

当 T3 作定时使用时, 可通过 CT3 位 ($=1$) 设定, T3 选择低频时钟 (异步时钟) 模式。此时, 通过用户配置字的 FMOD 位设定其时钟源为外部 32768Hz 振荡器或其分频, 还是选择内部 32KHz 低频振荡器或其分频, 通过 T3CKS 设置, 可选择多种分频 (具体值参考 T3 控制寄存器说明)。方式 1 类似于方式 0, 它们的本质类似, 只是时钟的来源不同。

注: 用户配置字 FMOD, 控制 Timer3 方式 1 的低频时钟源, 若 FMOD 选择时钟包含 FLOSC, 则选择 FLOSC 作为 Timer3/0/1 时钟源, 否则选择 FLIRC 作为时钟源。

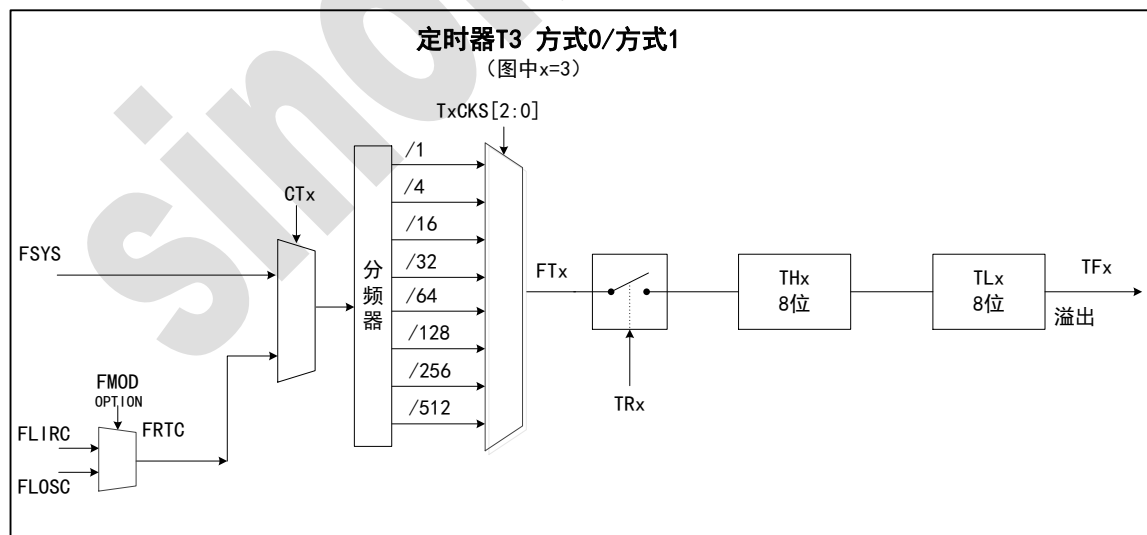
通过配置字 FMOD 配置, 可以选择内置 32KHz 低频振荡器或外部 32768Hz 晶体振荡器作为时钟源, 通过设定 OSCM 寄存器的 LFEN 位确定 T3 是否在停止模式 (STOP) / 休眠模式 (SLEEP) 下具备唤醒功能。

当 LFEN=1, 停止模式/休眠模式下选择的低频时钟源 (FIRC 或 FLOSC) 继续工作, 当寄存器 TH3/TL3 溢出后会产生 T3 中断, 并通过中断从而实现在停止模式/休眠模式下唤醒芯片的功能。

当 LFEN=0, 停止模式下低频时钟源 (FIRC 或 FLOSC) 将继续工作, 则 T3 在停止模式具备唤醒芯片的功能; 休眠模式下低频时钟源 (FIRC 或 FLOSC) 停止, T3 不具备唤醒芯片的功能。

若选择外部 32768Hz 振荡器作为时钟源时 (需在用户配置字中 FMOD 设置相关值), 需连接外部 32768Hz 振荡器到对应的 OSC1/OSC0 端口, 这种模式下可以实现对时间的精准计时。

用户应充分考虑在停止模式/休眠模式下是否需要使用 T3 作为唤醒源, 并做相应的设定, 当内置 32KHz 振荡器还是外部 32768Hz 振荡器, 始终“开启”为 T3 提供时钟源的情况下, 会增加芯片的功耗。





8.6 T3 相关寄存器

T3 寄存器汇总表

地址 (SFR n/H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR0/97	TH3	T3 计数高位寄存器	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30	00000000
SFR0/96	TL3	T3 计数低位寄存器	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30	00000000
SFR0/8F	T3CR	T3 控制寄存器	TR3	CT3	T3F	—	T3EN	T3CKS2	T3CKS1	T3CKS0	000—0000

8.6.1 T3 控制寄存器 (T3CR, 0x8F/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3CR	TR3	CT3	T3F	—	T3EN	T3CKS2	T3CKS1	T3CKS0
R/W	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W
复位值	0	0	0	—	0	0	0	0

BIT[7] TR3 - 定时器/计数器 T3 使能位

0: 停止定时器/计数器 T3;

1: 打开定时器/计数器 T3, 每次写入 1, 将从 TH3 和 TL3 的起始值开始递增计数;

BIT[6] CT3 - T3 模式选择位

0: 时钟源为系统时钟或其分频;

1: 时钟源为内置 32KHz 振荡器或者外部 32768Hz 振荡器或其分频, 具体由“用户配置字”决定。

注 1: 当该位=1 时, 时钟源由“用户配置字”决定, 若用户配置字中打开外部 32768Hz 振荡器 (相应的端口 OSC1 和 OSC0 上已经接了振荡器), 那么选 32768Hz 振荡器或其分频作为时钟源, 否则为选 32KHz 内部振荡器或其分频作为时钟源。

注 2: 当该位=1 时, 由于外部没有 32768Hz 振荡器, (LFEN=0) 32KHz 内部振荡器也停止时, T3 将停止工作。

BIT[5] T3F - 定时器/计数器 T3 溢出标志位

0: 定时器/计数器 T3 无溢出, 中断响应后硬件清 0, 也可由软件清 0;

1: 定时器/计数器 T3 溢出时, 由硬件置 1。

BIT[3] T3EN - 定时器/计数器 T3 模块使能位

0: 关闭定时器/计数器 T3 模块;

1: 打开定时器/计数器 T3 模块。

BIT[2:0] T3CKS[2:0] - T3 基础定时器时钟分频选择位

对应频率选择列表:

T3CKS[2:0]	分频
000	Fsys
001	Fsys/4
010	Fsys/16
011	Fsys/32
100	Fsys/64
101	Fsys/128
110	Fsys/256
111	Fsys/512

8.6.2 T3 计数高位寄存器 (TH3, 0x97/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH3	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] TH3[7:0] - 定时器/计数器 T3 高 8 位计数器设定值

8.6.3 T3 计数低位寄存器 (TL3, 0x96/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL3	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



复位值	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[7:0] **TL3[7:0]** - 定时器/计数器 T3 低 8 位计数器设定值

注 1: 无论是读或者写 TL3/TH3 的值, 必须先停止 ($TR3=0$), 否则 T3 将产生不可预知的情况。

注 2: 写操作时: 必须先写 TL3, 再写 TH3; 否则 TL3 的值不能立即反映到内部 counter 上; 读操作时: 必须先读 TL3, 再读 TH3, 否则将读取到错误的值。



9 8位单路PWM0

9.1 单路 PWM0 概述

片上提供 1 路 8 位 PWM0 模块，可设置独立中断向量入口，输出极性可选，该 PWM 也可用作具有重载功能的通用定时器使用。输出引脚为 PWM0。

PWM0 周期的计算公式：

$$\text{PWM0周期} = \frac{\text{PWMOP} + 1}{\text{Fsys}}$$

注：PWM0 周期的时钟设定，固定为 Fsys 或 Fsys 的分频，最大不应该超过 16MHz。

PWM0 占空比（默认为高电平的时间）的计算公式：

$$\text{PWM0占空比} = \frac{\text{PWMOD} + 1}{\text{PWMOP} + 1}$$

注 1：假设为（默认）高电平有效，当 PWMOD=0 时，占空比为最小 0%，输出恒为 0，但每个周期之间存在一个 clock（由 1 到 0）切换的时间；当 PWMOD < PWMOP，占空比为 PWMOD+1 的高电平时间；当 PWMOD ≥ PWMOP 时，占空比为最大 100%，输出恒为 1。

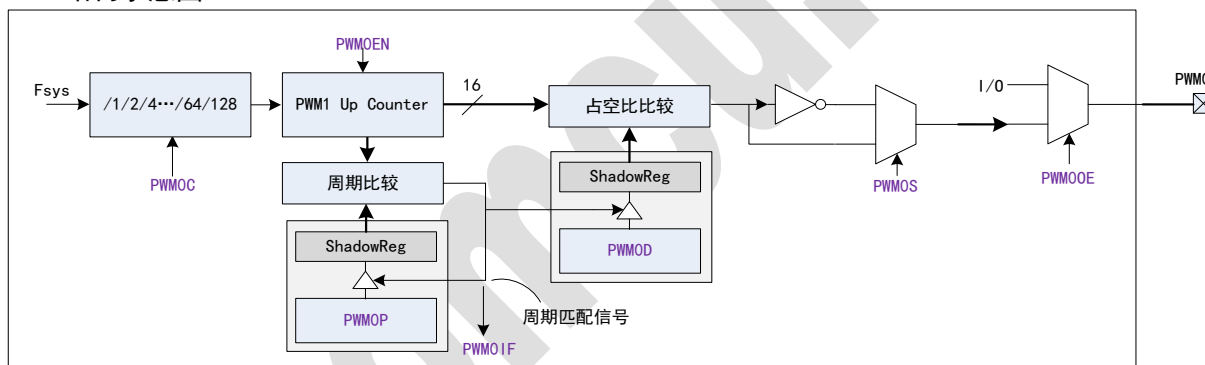
注 2：在 PWM0 输出波形时，若当前周期或占空比被修改，不会立即生效，需要等到下个周期来到才会相应改变。

注 3：占空比所代表的电平，由 PWMOS 位决定，用户可以切换。当 PWMOS 设为 1 时，这里的 PWM0 占空比指“低电平”所占的时间。

注 4：当用户需要 PWM0 输出恒 0/1 时，推荐下面两种做法：A. 在 PWM 功能下反向控制极限位，并以 100% 占空比输出，此时将获得需要的电平值；B. 给 Px 端口指定一个 0/1 电平，并通过改变 PWM0OE 位退出 PWM 功能，切换至 GPIO 的输出功能，直接输出指定的电平值。

9.2 PWM0 结构图

PWM0 结构示意图



9.3 单路 PWM0 相关寄存器

PWM0 寄存器汇总表

地址 (SFRn /H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
XSFR/F F68	PWMOCR	PWM0 控制寄存器	PWMOEN	PWMOPIE	PWMOIF	PWMOOE	PWMOS	PWMOC2	PWMOC1	PWMOCO	0000 0000
XSFR/F F69	PWMOP	PWM0 周期设置 寄存器	PWMOP7	PWMOP6	PWMOP5	PWMOP4	PWMOP3	PWMOP2	PWMOP1	PWMOP0	00000 000
XSFR/F F6A	PWMOD	PWM0 占空比设 置寄存器	PWMOD7	PWMOD6	PWMOD5	PWMOD4	PWMOD3	PWMOD2	PWMOD1	PWMOD0	00000 000

9.3.1 PWM0 控制寄存器（PWMOCR, 0xFF68/XSFR）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOCR	PWMOEN	PWMOPIE	PWMOIF	PWMOOE	PWMOS	PWMOC2	PWMOC1	PWMOCO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7]

PWMOEN - PWM0 使能位

0：关闭 PWM0 模块，关闭后 PWM 停止计数，输出也将停止；

1：打开 PWM0 模块，PWM0 计数器从 0 开始向上计数，PWM0 对应的占空比较器不断比较 PWMOD 与



PWMOCNT 的值，当两者相同时反转一次输出波形，比较 PWMOP 与 PWMOCNT 的值，当周期结束时再反转一次，并继续新一轮周期的比较反转。

注 1: PWM0EN 即使计数的控制位，配置 PWM0 时须先配置周期寄存器和占空比寄存器，再设置 PWM0EN 位为 1。

注 2: PWMOCNT 是一个内部寄存器，对用户不可见，向上自增，将在 PWM0EN 启动后开始计数或者重新计数。

BIT[6] PWM0PIE - PWM0 周期中断使能

0: 关闭;

1: 开启。

注: 此控制位与 IE1 的 PWMIE 联动控制，若要使能 PWMOP 中断，必须同时置位 PWM0PIE 和 PWMIE；若要禁止 PWMOP 中断，仅需清零 PWMIE 和 PWM0PIE 任意一位。PWMIFB、PWMIP 和 PWMOP 复用同一个中断入口。

BIT[5] PWM0IF - PWM0 中断标志位

0: PWM0 定时器未溢出（计数未超过周期寄存器值），由软件清 0；

1: PWM0 定时器已溢出（计数超过周期寄存器值），由硬件置 1。

BIT[4] PWM0OE - PWM0 输出使能位

0: 禁止 PWM0 复用功能，端口用于通用 I/O 或其他复用模块；

1: 打开 PWM0 复用功能，作为 PWM0 输出波形用。

注 1: PWM0EN 设置为 0 时，硬件会同时关闭 PWM0OE，需要再次开启 PWM，需要重新配置 PWM0EN=1 和 PWM0OE=1。

注 2: 当禁止 PWM0 复用功能时，PWM0 可用作为“具有重载功能的通用定时器”使用。

BIT[3] PWM0S - PWM0 输出极性选择位

0: PWM0 输出占空比有效期间为高电平；

1: PWM0 输出占空比有效期间为低电平。

BIT[2:0] PWM0C[2:0] - PWM0 工作时钟频率选择位

PWM0C[2:0]	PWM0 工作频率
000	Fsys
001	Fsys/2
010	Fsys/4
011	Fsys/8
100	Fsys/16
101	Fsys/32
110	Fsys/64
111	Fsys/128

注: 在 PWM0 输出波形时，请勿随意修改频率，否则会影响 PWM0 的输出结果；建议按先关闭 PWM0，修改后再打开 PWM0 的顺序操作。

9.3.2 PWM0 周期设置寄存器 (PWMOP, 0xFF69/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOP	PWMOP7	PWMOP6	PWMOP5	PWMOP4	PWMOP3	PWMOP2	PWMOP1	PWMOP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMOP[7:0] - 设置 PWM0 的周期。

9.3.3 PWM0 占空比设置寄存器 (PWMOD, 0xFF6A/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOD	PWMOD7	PWMOD6	PWMOD5	PWMOD4	PWMOD3	PWMOD2	PWMOD1	PWMOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMOD [7:0] - 设置 PWM0 的占空比，即 PWM0 周期内先输出有效电平的时钟个数。

注: 在 PWM0 配置时，建议先设置周期和占空比寄存器，再配置控制寄存器。



10 16位增强型PWM1阵列（电机控制）

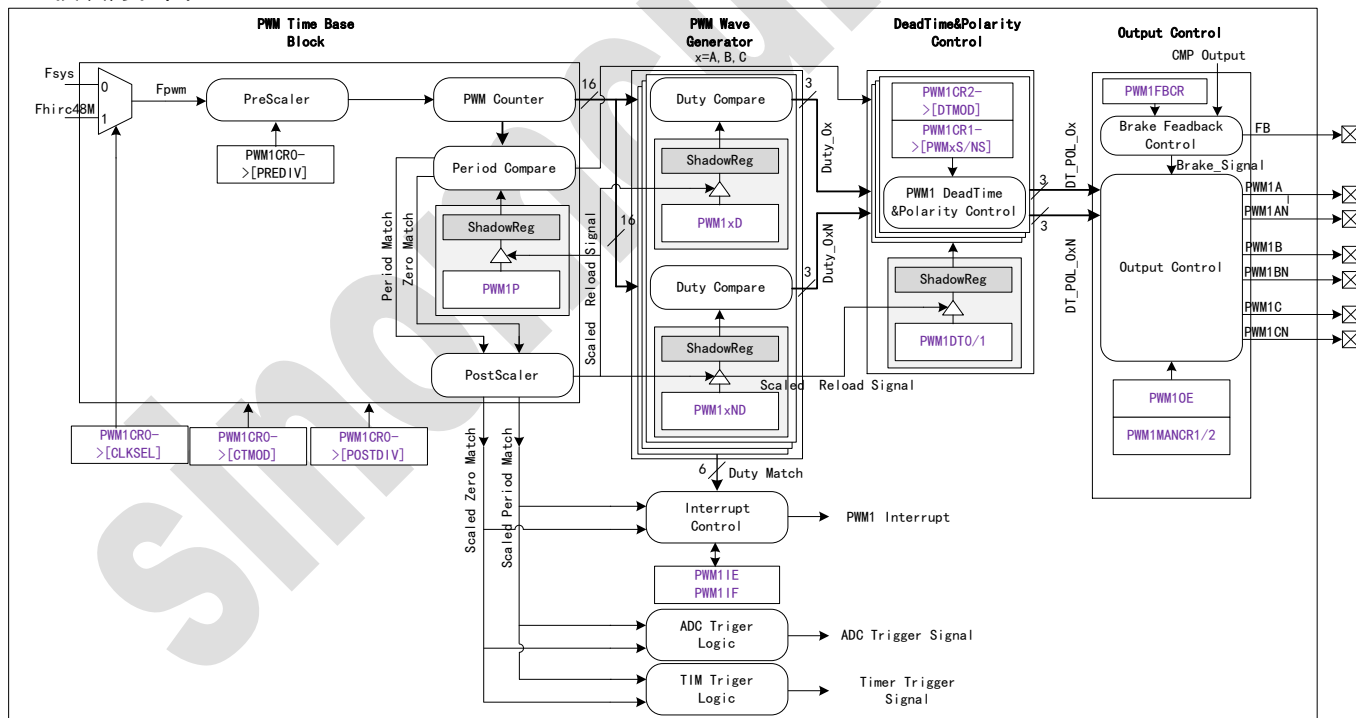
10.1 概述

- ◇ 重要寄存器受保护寄存器 PWMLEN 控制
- ◇ 16 位时基计数器
- ◇ 三种时基计数模式：边沿对齐计数、中心对齐计数和单次计数模式
- ◇ 两种波形输出模式：互补模式与独立模式
- ◇ 提供 PWM 周期匹配、归零匹配和占空比较匹配中断
- ◇ 3 路互补 PWM 输出或 6 路独立 PWM 输出，输出极性可选择
- ◇ 提供故障检测功能可紧急关闭 PWM 输出
- ◇ 提供寄存器重载使能位以保证寄存器重载同步
- ◇ 4 档预分频 / 后分频功能
- ◇ 可切换手动控制 PWM 输出
- ◇ 可选择自动触发 Timer 或 ADC 转换

芯片内建一个 16 位电机控制 PWM 阵列。PWM 模块由时基模块（PWM Time Base Block）、波形发生模块（Wave Generator）、死区&极性控制模块（Dead&Polarity Control）、刹车检测（Brake Feedback Control）以及输出控制模块（Output Control）组成。

时基模块中通过一个 16 位计数器（PWM Counter）对 PWM 时钟（PWM Clock）计数，计数器的值与周期寄存器比较产生周期匹配信号、与 0 比较产生归零信号、与占空比寄存器比较产生占空比匹配信号，结合波形发生模块产生原始的六路 PWM 信号 Duty_Ox 和 Duty_OxN (x=A, B, C)，该原始 PWM 信号经过死区&极性控制模块后产生带死区和极性的 PWM 信号 DT_POL_Ox 和 DT_POL_OxN (x= A, B, C)，最后的输出控制模块决定是否将 DT_POL_Ox 和 DT_POL_OxN (x= A, B, C) 信号输出至 PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 上。

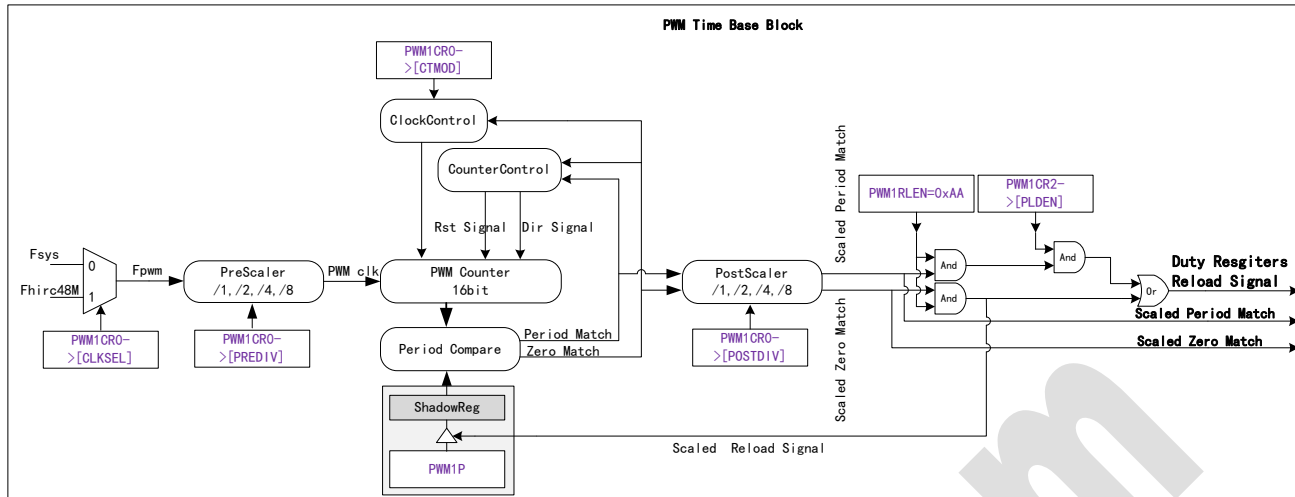
PWM 模块方框图



10.2 PWM 时基模块

PWM 模块中的时基模块（Time Base Block）主要由一个 16 位的计数器结合一个预分频器和一个后分频器组成。如图所示，波形发生器（Wave Generator）用时基模块中 16 位计数器的值与占空比寄存器比较并结合死区/极性逻辑（DeadTime&Polarity Control）和输出控制逻辑（Output Control）最终产生 PWM 波形。

时基模块逻辑框图



10.2.1 PWM 工作时钟

PWM1 时钟源可以选择系统时钟 F_{sys} 或 Hirc48M 时钟, 选择后的时钟 F_{pwm} 经过预分频提供给 PWM 时基计数器。PWMCRO 寄存器的 PREDIV 控制位控制分频系数 1/1、1/2、1/4 或 1/8 分频。

通过软件置位/清零 PWM10E 寄存器中的 PWM1EN 位可以使能/禁止 PWM 时基模块。PWM1EN 位从 0 置 1 时, PWM 计数器 (PWM Counter) 开始对 PWM 时钟从 0 开始向上计数。PWMDEN 位清 0 时, 预分频计数器和 PWM 计数器都将清 0。

注: 任何复位都将导致 PWM1 时钟预分频计数器和 PWM 计数器清零。

10.2.2 PWM 时基工作方式

PWM1 时基中 16 位的 PWM 计数器有 3 种工作方式, 由 PWMCRO 寄存器中的 CTMOD[1:0]控制位确定。

边沿对齐计数模式

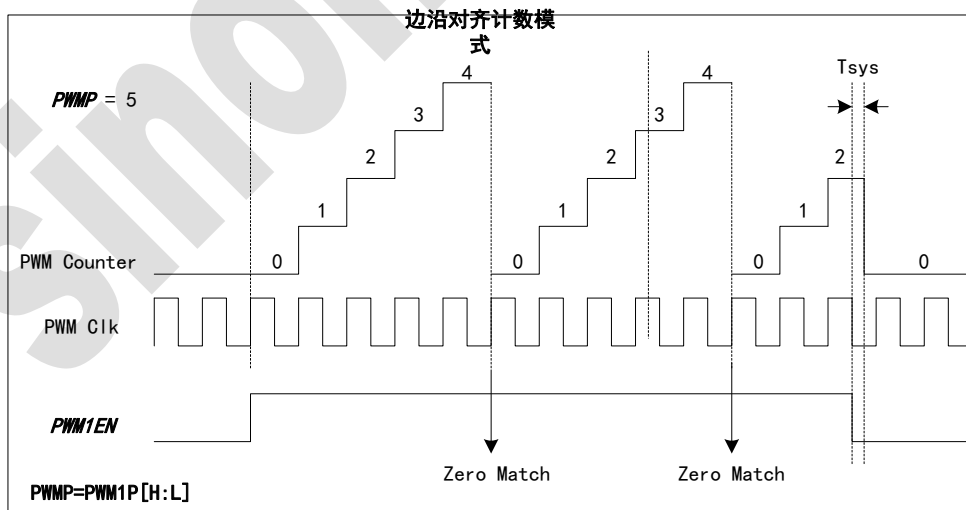
设置 PWMCR0 中 CTMOD[1:0]为 00，时基模块中的 PWM 计数器将工作在边沿对齐计数模式下。

此模式下，设置 PWM1EN 位置 1，PWM 计数器将从 0 开始向上计数，直到与 PWM 周期寄存器 PWM1PL/H 匹配，然后 PWM 计数器复位为 0 并继续向上计数，如此往复。

PWM 计数器复位为 0 的时刻产生归零匹配 (Zero Match)。

设置 PWM1EN 位清 0，PWM Counter 将停止计数并在下一个系统时钟边沿复位为 0 并停止计数。

边沿对齐计数模式时序图 (PWM1PL/H = 5)



中心对齐计数模式

设置 PWMCRO 中 CTMOD[1:0] 为 01, 时基模块中的 PWM 计数器将工作在中心对齐计数模式下。

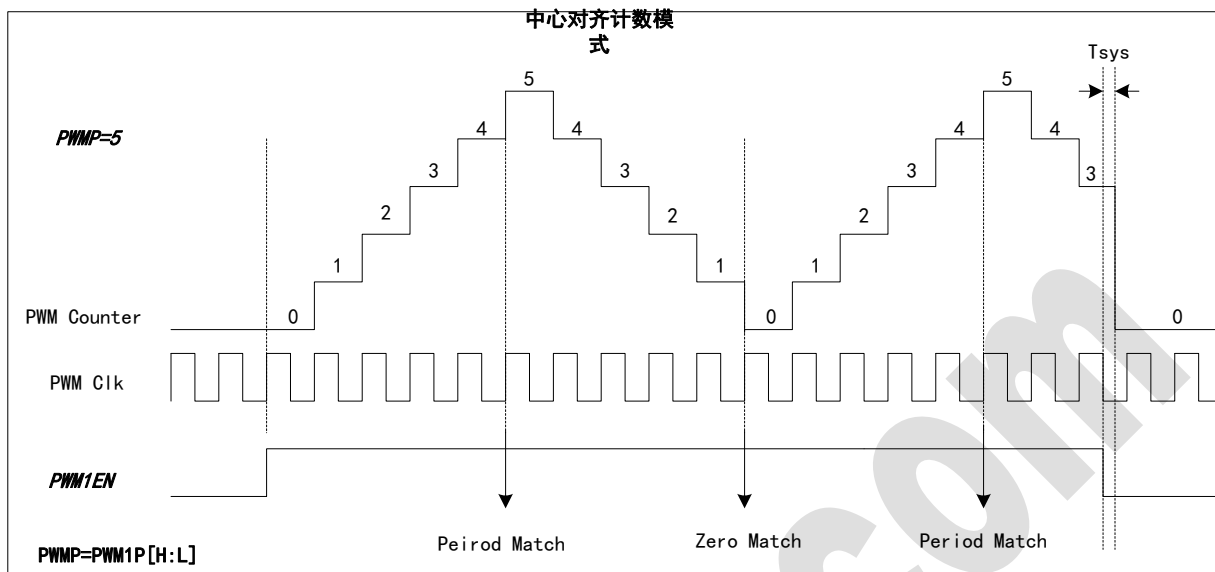
此模式下，设置 PWM1EN 位置 1，PWM 计数器将从 0 开始向上计数，直到与 PWM 周期寄存器 PWM1PL/H 匹配，然后 PWM 计数器开始向下计数直至归 0，如此往复。



PWM 计数器向下计数至零的时刻产生归零匹配 (Zero Match), PWM 计数器与周期寄存器匹配时刻产生周期匹配 (Period Match)。

设置 PWM1EN 位清 0, PWM Counter 将停止计数并在下一个系统时钟边沿复位为 0 并停止计数。

中心对齐计数模式时序图 (PWM1PL/H = 5)



单次计数模式

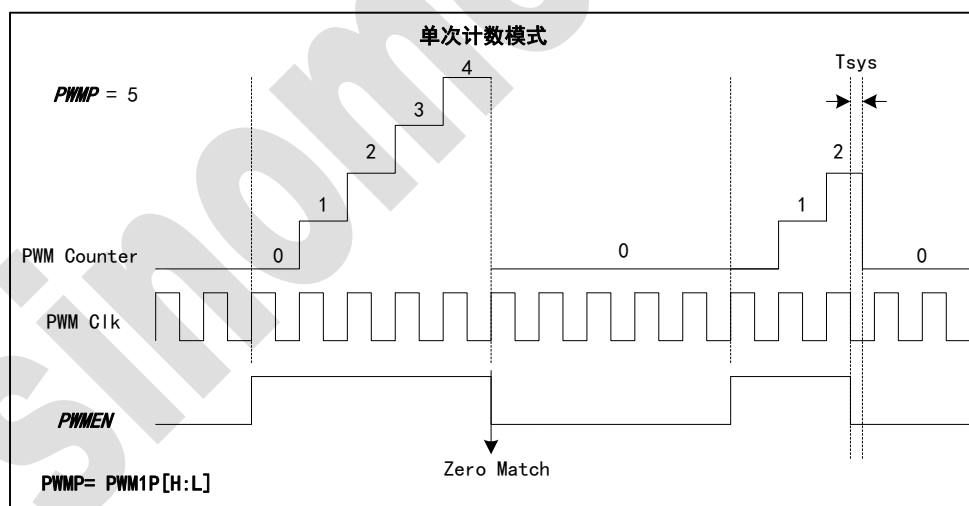
设置 PWM1CRO 中 CTMOD[1:0] 为 10 或 11, 时基模块中的 PWM 计数器将工作在单次计数模式下。

此模式下, 设置 PWM1EN 位置 1, PWM 计数器将从 0 开始向上计数, 直到与 PWM 周期寄存器 PWM1PL/H 匹配时, PWM 计数器复位为 0, 同时 PWM1EN 位将由硬件清 0, PWM1 输出高阻状态。

PWM 计数器复位为 0 的时刻产生归零匹配 (Zero Match)。

设置 PWM1EN 位清 0, PWM Counter 将停止计数并在下一个系统时钟边沿复位为 0 并停止计数。

单次计数模式时序图 (PWM1PL/H = 5)



10.2.3 后分频

PWM 计数器运行时产生的周期匹配信号 (Period Match) 和归零匹配信号 (Zero Match) 经过后分频计数器 (PostScaler), 可以实现 1/1、1/2、1/4 和 1/8 分频。通过寄存器 PWM1CRO 中的 POSTDIV[1:0] 位可以选择以上四档后分频。若设置 POSTDIV[1:0] = 00 则后分频系数为 1/1, 周期匹配信号和归零匹配信号不会被分频。

后分频应用场景:

- 1、通过后分频配置, 可以实现多个周期更新一次占空比, 不需要每个周期更新 PWM 的占空比
- 2、后分频后的周期匹配信号 (Scaled Period Match) 和归零匹配信号 (Scaled Zero Match) 结合 PWM1LEN 寄存器值用于控制周期寄存器 PWM1PL/H 以及占空比寄存器 PWM1xD[H, L]/PWM1xND[H, L] (x=A, B, C) 的重载。
- 3、后分频后的周期匹配信号 (Scaled Period Match) 和归零匹配信号 (Scaled Zero Match) 可用于产生归零匹配中断和周期匹配中断, 详见“时基中断”章节。
- 4、后分频后的周期匹配信号 (Scaled Period Match) 和归零匹配信号 (Scaled Zero Match) 可用于触发 ADC,



Timer2。在中心对齐模式下触发 timer2，可用分频后的周期匹配信号触发、或分频后的归零匹配信号触发，也可以同时选择两者。中心对齐模式触发 ADC，可以选择在分频后的周期匹配信号触发，或分频后的归零匹配信号触发，也可以同时选择两者。而在边沿对齐模式下，只能用分频后的归零匹配信号触发。

注：后分频只对周期匹配信号和归零匹配信号有效，对占空比匹配时刻无效，比如，允许占空比中断和归零匹配信号中断的情况下，设置后分频为 8 分频时，归零匹配信号进中断会在 8 个 PWM 周期后才进行一次中断，而占空比匹配信号则依然会在每个 PWM 周期触发中断。

10.2.4 时基中断

时基中断包括分频后的周期匹配信号所触发的周期中断和分频后的归零匹配信号所触发的归零中断。分频后的周期匹配信号发生时将令寄存器 PWM1IF 中的 PWMPIF 位置 1，此时若 PWM1IE 中的 PWMPIE 位为 1 则将触发周期中断；分频后的归零匹配信号发生时将令寄存器 PWM1IF 中的 PWMZIF 位置 1，此时若 PWM1IE 中的 PWMZIE 位为 1 则将触发归零中断。PWMPIF 位和 PWMZIF 位需由软件写 0 清除。

PWM1CR0 中 POSTDIV[1:0]段等于 0 时，PWM 计数器的周期匹配信号和归零匹配信号不会被分频，故周期匹配信号每次发生时都将令 PWMPIF 位置 1，归零匹配信号每次发生时都将令 PWMZIF 位置 1。

10.2.5 PWM 周期寄存器的自动重载

周期寄存器 PWM1PL 和 PWM1PH 都有一个缓存寄存器 (ShadowReg)，用户能够读写 PWM1PL/H 但不能操作其缓存寄存器。PWM 时基计数器运行时实际上是与周期寄存器的 ShadowReg 进行比较以产生周期匹配信号。

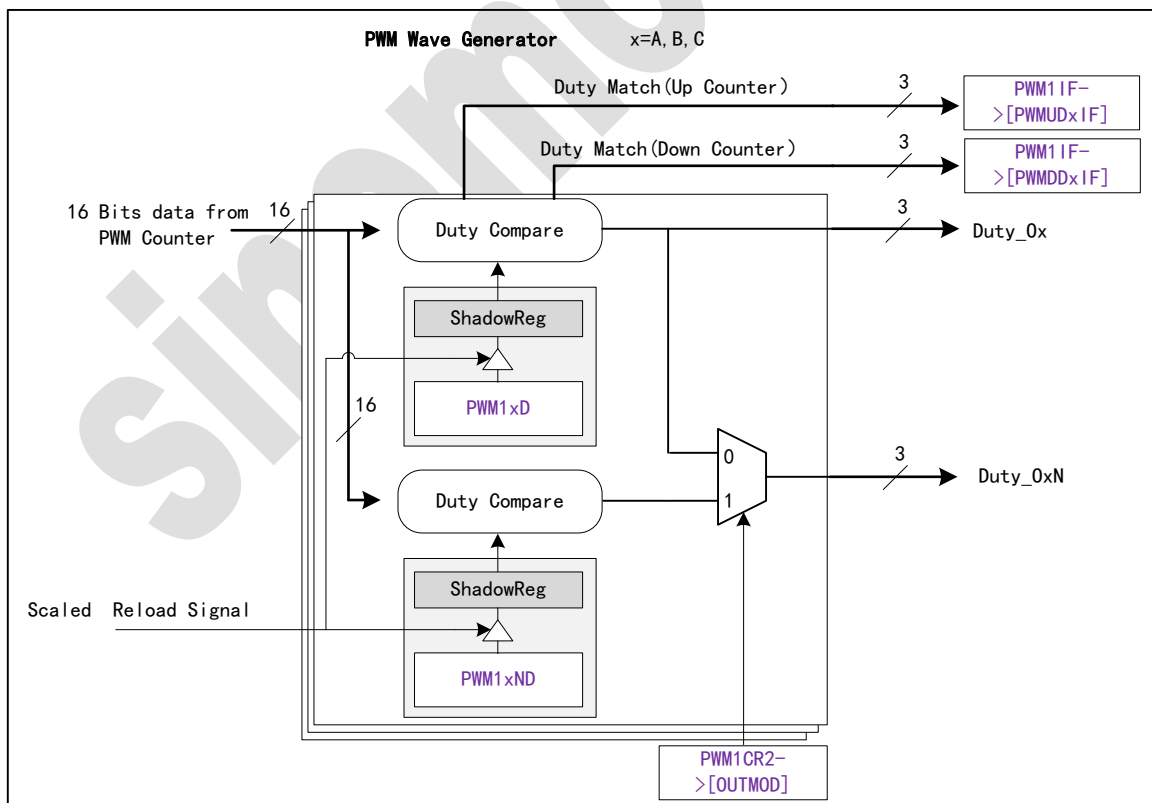
由于存在缓冲寄存器，因此对周期寄存器的读写有特殊规定：

当 PWM1RLEN 寄存器的值设为 0x55 时，才允许对这些寄存器进行修改，然后只有当 PWM1RLEN 寄存器中的值设为 0xAA 时，分频后的归零匹配信号发生时才会将 PWM1PL 和 PWM1PH 寄存器锁存至对应的缓存寄存器 (ShadowReg) 中，这样可以避免寄存器修改过程中发生重载，也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改，必须要有对 PWM1RLEN 寄存器的配合操作才能完成。

10.3 PWM 波形发生模块

PWM 波形发生模块使用 PWM 时基模块中 PWM 计数器的值与 6 个 16 位占空比缓存寄存器值进行比较，从而产生原始的 6 路 PWM 波形：Duty_0x 和 Duty_0xN (x=A, B, C)。该原始 6 路 PWM 波形信号经过死区&极性逻辑和输出控制逻辑最终经由 6 个 PWM 引脚输出。

波形发生模块逻辑框图



10.3.1 PWM 输出模式

PWM 模块中包含有 3 个波形发生模块，对应于 3 对 PWM 输出：PWM1A/PWM1AN、PWM1B/PWM1BN 和 PWM1C/PWM1CN。对



于每对 PWM 输出可以分别配置为互补模式或独立模式。

互补输出模式

设置 PWM1CR1 寄存器中 OUTMOD 位为 0, PWM1x/PWM1xN 工作在互补模式, 16 位 PWM 计数器与占空比寄存器 PWM1xDL/H 比较以产生 Duty_0x 和 Duty_0xN 波形, 因此 PWM1x/PWM1xN 引脚上输出的 PWM 波形使用的是同一占空比寄存器 PWM1xDL/H。在互补输出模式下, 可由寄存器 PWM1CR1 设置 PWM1x/PWM1xN 引脚上输出波形的极性、并可由寄存器 PWM1DTOL/H 和 PWM1DT1L/H 设置死区 (详见 9.5 章节“PWM 死区与极性控制模块”)。

注: $x=A, B, C$

独立输出模式

设置 PWM1CR1 寄存器中 OUTMOD 位为 1, PWM1x/PWM1xN 工作在独立模式, 16 位 PWM 计数器与占空比寄存器 PWM1xDL/H 比较以产生 Duty_0x 波形, 16 位 PWM 计数器与占空比寄存器 PWM1xNDL/H 的值比较以产生 Duty_0xN 波形, 因此 PWM1x 和 PWM1xN 引脚上输出的 PWM 波形使用的是不同的占空比寄存器 PWM1xDL/H 与 PWM1xNDL/H。在独立输出模式下, 可由寄存器 PWM1CR1 设置 PWM1x/PWM1xN 引脚上输出波形的极性、但是寄存器 PWM1DTOL/H 和 PWM1DT1L/H 将无效, 即独立输出模式下 PWM 的输出无死区 (详见 9.5 章节“PWM 死区与极性控制模块”)。

注: $x=A, B, C$

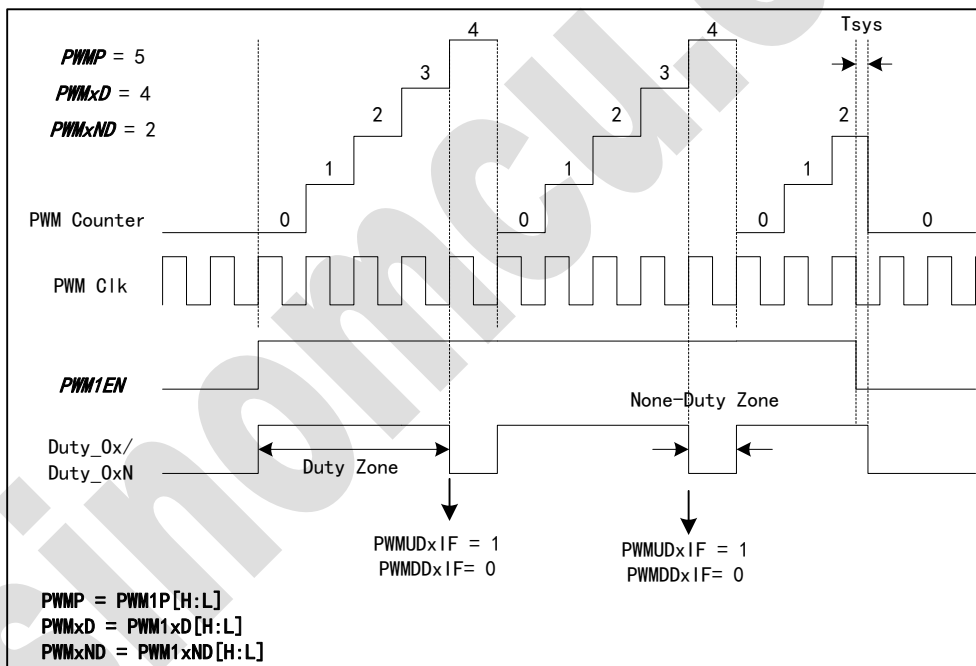
10.3.2 边沿对齐计数模式下的 PWM1A/B/C 原始波形

设置 PWM1CR0 中 CTMOD[1:0] = 00, 将 PWM 时基设置为边沿对齐计数模式。

在互补输出模式下 (PWM1CR1→OUTMOD = 0), PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图所示。PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形变高, PWM 计数器与占空比寄存器 PWM1xDL/H 匹配后 Duty_0x 和 Duty_0xN 波形将变低直至 PWM 计数器归零, 如此往复。

注: $x=A, B, C$

边沿对齐计数、互补输出模式下的 PWM 原始波形

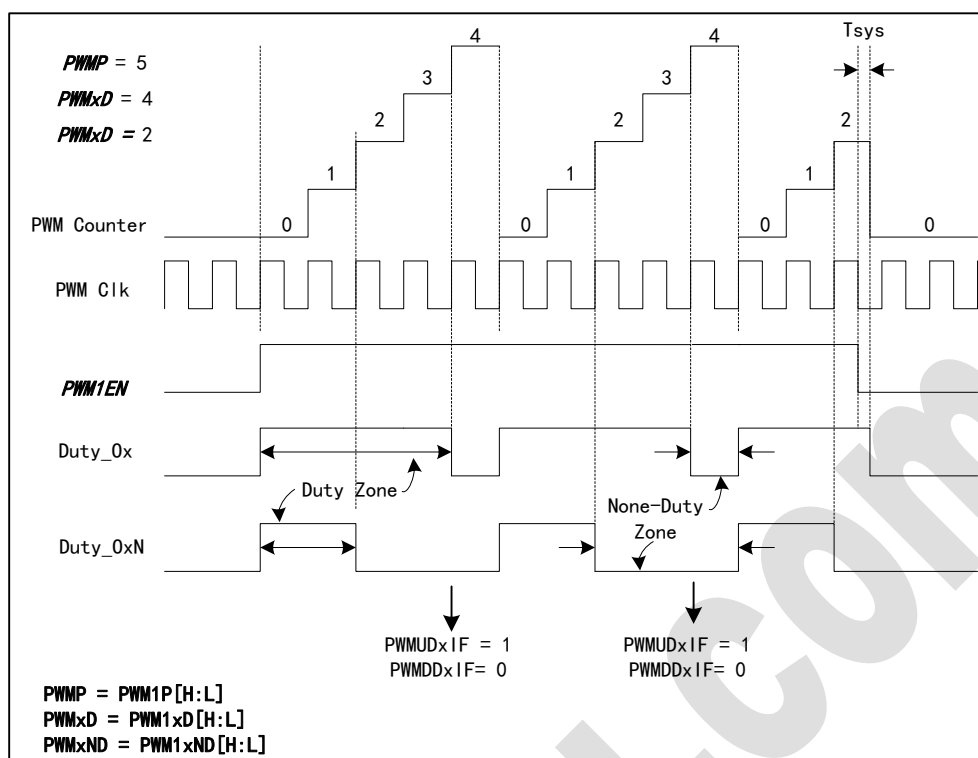


在独立输出模式下 (PWM1CR1→OUTMOD = 1), PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图 8 所示。PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形变高, PWM 计数器与占空比寄存器 PWM1xDL/H 匹配后 Duty_0x 波形将变低直至 PWM 计数器归零; PWM 计数器与占空比寄存器 PWM1xNDL/H 匹配后 Duty_0xN 波形将变低直至 PWM 计数器归零, 如此往复。

注: $x=A, B, C$



边沿对齐计数、独立输出模式下的 PWM 原始波形



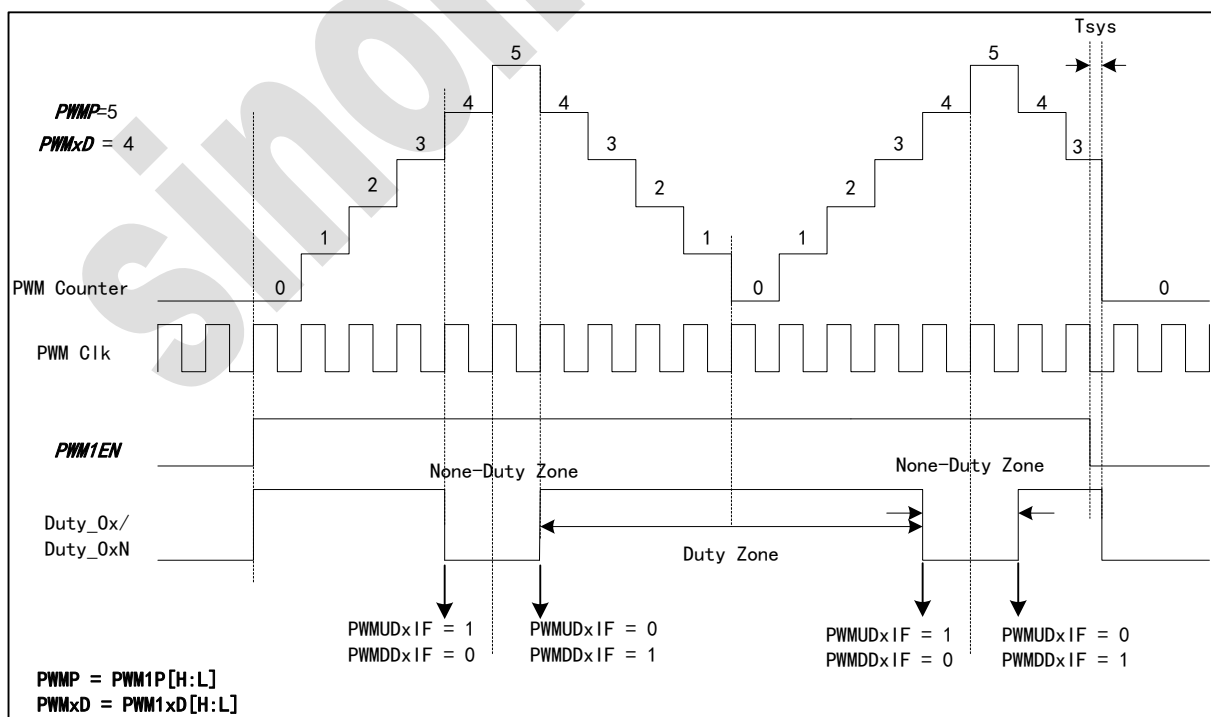
10.3.3 中心对齐计数模式下的 PWM1A/B/C 原始波形

设置 PWM1CR0 中 CTMOD[1:0] = 01，将 PWM 时基设置为中心对齐计数模式。

在互补输出模式下 (PWM1CR1->OUTMOD = 0)，设置 PWM1CR1->SYMC=0，PWM 工作在对称输出模式，PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图所示。PWM1EN 置 1 后，Duty_0x 和 Duty_0xN 波形保持为高，PWM 计数器增计数时与 PWM1xDL/H 匹配后 Duty_0x 和 Duty_0xN 波形将变低直至 PWM 计数器减计数时再次与 PWM1xDL/H 匹配，如此往复。

注：x=A, B, C

中心对齐计数、互补输出模式下的 PWM 原始波形（对称波形输出）

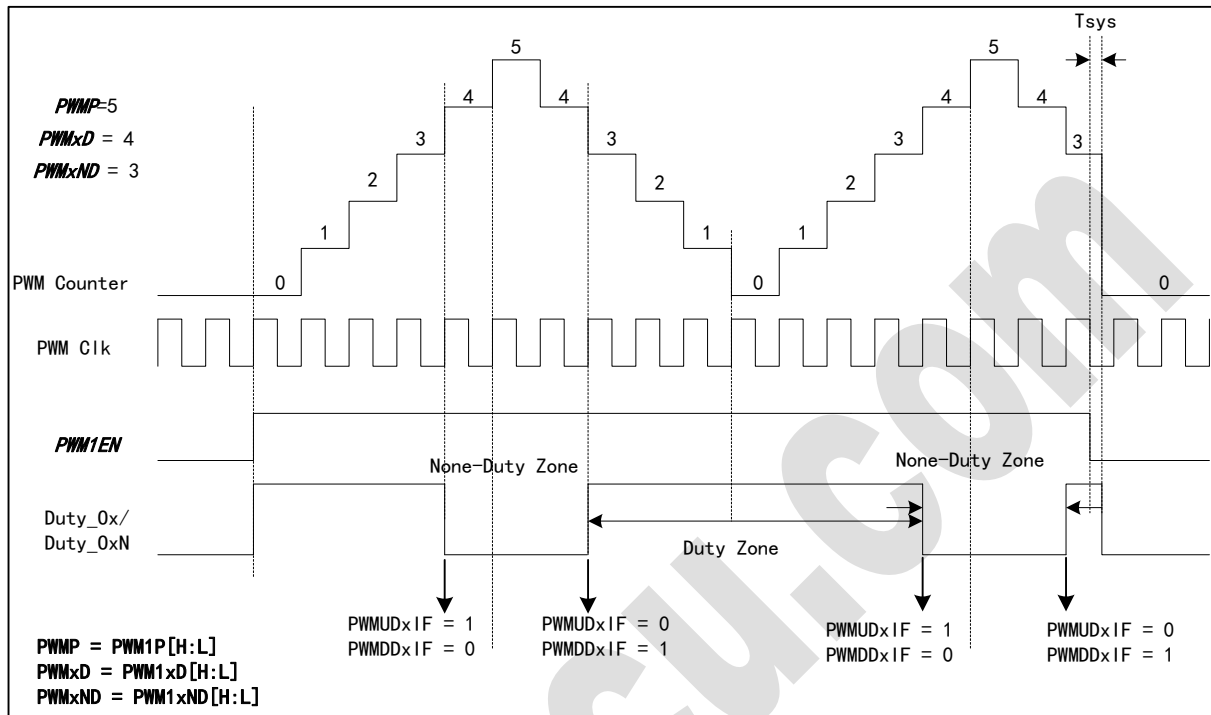




在**互补输出模式**下 (PWM1CR1->OUTMOD = 0), 设置 PWM1CR1->SYMC=1, PWM 工作在非**对称输出模式**, PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图所示。PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形保持为高, PWM 计数器增计数时与 PWM1xDL/H 匹配后 Duty_0x 和 Duty_0xN 波形将变低, 直至 PWM 计数器减计数时与 PWM1xNDL/H 匹配变高, 如此往复。

注: $x=A, B, C$

中心对齐计数、互补输出模式下的 PWM 原始波形 (非对称波形输出)

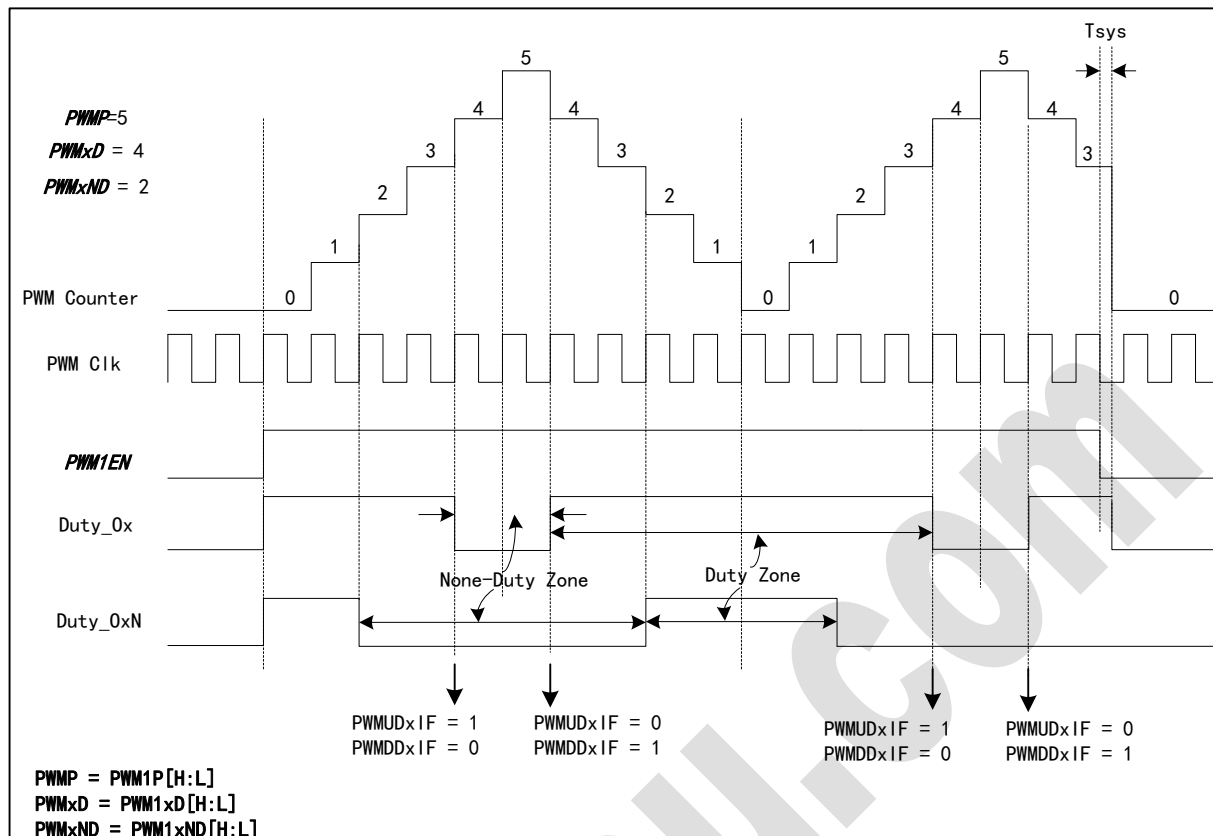


在**独立输出模式**下 (PWM1CR1->OUTMOD = 1), PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图所示。PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形变高, PWM 计数器增计数时与 PWM1xDL/H 匹配后 Duty_0x 波形将变低直至 PWM 计数器减计数时再次与 PWM1xDL/H 匹配; PWM 计数器增计数时与 PWM1xNDL/H 匹配后 Duty_0xN 波形将变低直至 PWM 计数器减计数时再次与 PWM1xNDL/H 匹配, 如此往复。

注: $x=A, B, C$



中心对齐计数、独立输出模式下的 PWM 原始波形



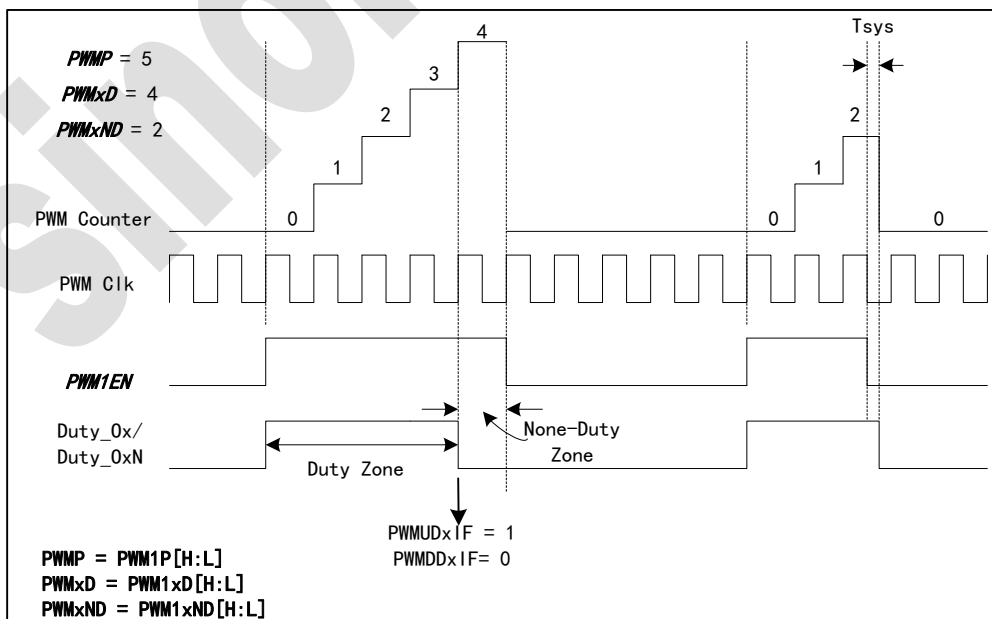
10.3.4 单次计数模式下的 PWM1A/B/C 原始波形

设置 PTCON 中 CTMOD[1:0] = 10 或 11, 将 PWM 时基设置为单次计数模式。

在互补输出模式下 (PWM1CR1->OUTMOD = 0), PWM 原始输出波形 Duty_0x 和 Duty_0xN 如图所示。PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形变高, PWM 计数器增计数时与 PWM1xDL/H 匹配后 Duty_0x 和 Duty_0xN 波形将变低。

注: x=A, B, C

单次计数、互补输出模式下的 PWM 原始波形

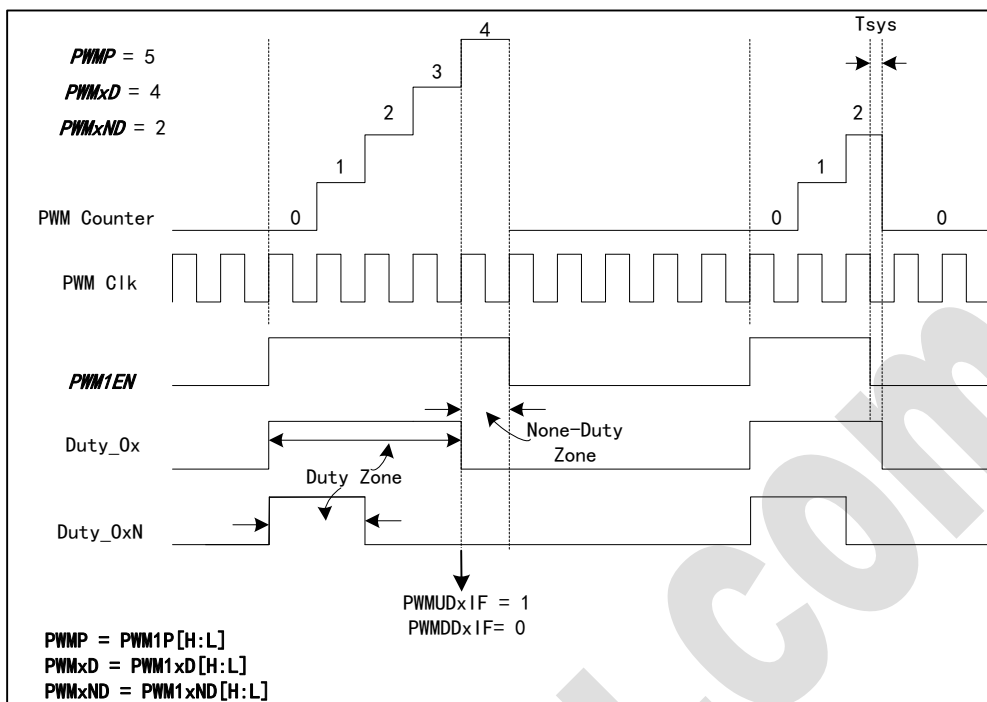


在独立输出模式下, PWM1EN 置 1 后, Duty_0x 和 Duty_0xN 波形变高, PWM 计数器增计数时与 PWM1xDL/H 匹配后 Duty_0x 波形将变低; PWM 计数器增计数时与 PWM1xNDL/H 匹配后 Duty_0xN 波形将变低。



注: $x=A, B, C$

单次计数、独立输出模式下的 PWM 原始波形



10.3.5 占空比寄存器的自动重载

6 个占空比寄存器都带有对应的缓存寄存器 (ShadowReg), 16 位的 PWM 计数器实际上是与 ShadowReg 比较从而控制 PWM 原始波形的变化。用户能够读写 6 个占空比寄存器, 却不能操作其对应的缓存寄存器。

由于存在缓冲寄存器, 和周期寄存器一样, 对占空比寄存器的读写有特殊规定:

当 PWM1LEN 寄存器的值设为 0x55 时, 才允许对占空比寄存器进行修改, 然后只有当 PWM1LEN 寄存器中的值设为 0xAA 时, 且分频后的归零信号发生时才会将占空比寄存器值锁存至对应的缓存寄存器 (ShadowReg) 中, 这样可以避免寄存器修改过程中发生重载, 也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改, 必须要有对 PWM1LEN 寄存器的配合操作才能完成。

另外, 在中心对齐模式下, 和周期寄存器不同, 占空比寄存器除了可以用归零匹配信号重载外, 还可以用周期匹配信号重载, 如果将 PWM1CR2 中 PLDEN 位为 1, 则分频后的周期匹配信号硬件自动将 6 个占空比寄存器锁存至对应的缓存寄存器中 (称之为重载)。

因此, 在边沿计数模式和单次计数模式下, 由于时基模块只有归零匹配信号, 故 PWM 计数器每次复位为 0 时都将发生一次占空比寄存器的重载; 中心对齐计数模式下, 时基模块能产生归零匹配以及周期匹配信号, 故 PWM 计数器每次复位为 0 以及与周期寄存器匹配时都可以重载占空比寄存器。

10.3.6 占空比中断

在 PWM 时基计数器向上计数时, 不论在何种模式, 当计数值与占空比寄存器 PWM1xDL/H 匹配时, PWM1IF 寄存器中 PWMUDxIF 位硬件置 1, 此时若 PWM1IE 寄存器中的 PWMUDxIE 位为 1, 则触发占空比中断。

在中心对齐计数模式中, PWM 时基计数器向下计数时, 当计数值与 PWM1xDL/H 匹配时, PWM1IF 寄存器中 PWMDDxIF 位硬件置 1, 此时若 PWM1IE 寄存器中的 PWMDDxIE 位为 1, 则触发占空比中断。

注 1: 不论互补输出模式还是独立输出模式, 占空比中断只是在 PWM 计数器与 PWM1xDL/H 值匹配时产生, 与 PWM1xNDL/H 值匹配时不会改变占空比中断标志。

注 2: PWM1IF 寄存器 PWMDDxIF 位, 仅在在中心对齐计数模式中, PWM 时基计数器向下计数时产生; PWM1IF 寄存器中 PWMUDxIF 位在任何模式的时基计数器向上计数产生, 包括中心对齐计数模式。

注 3: PWMDDxIF 和 PWMUDxIF, 软件可写零清零。

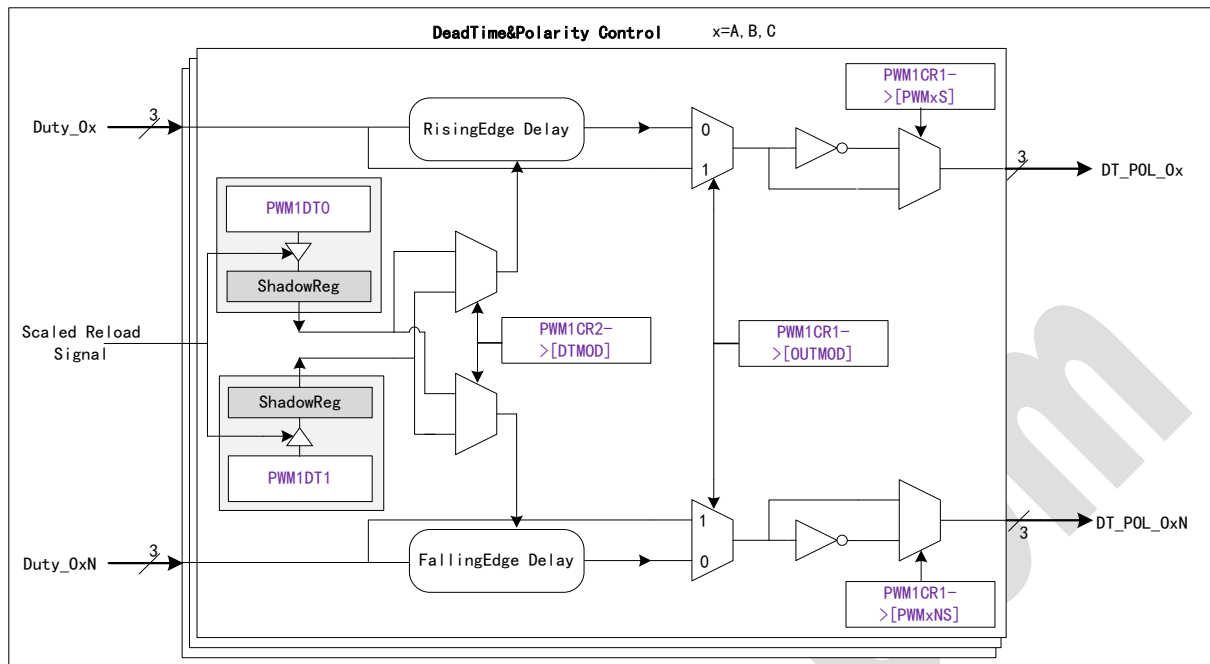
10.4 PWM 死区与极性控制模块

波形发生模块 (Wave Generator) 产生的 PWM 原始波形 Duty_0x 和 Duty_0xN 经过死区与极性控制模块 (Dead&Polarity Control) 产生最终的 PWM 波形 DT_POL_0x 和 DT_POL_0xN。带有死区与极性的 PWM 信号 DT_POL_0x/DT_POL_0xN 由输出控制模块 (Output Control) 和刹车检测模块 (Brake Feedback Control) 决定是否由引



脚 PWM1x/PWM1xN 送出, (x = A, B, C)。

死区与极性控制模块的逻辑框图



10.4.1 PWM 死区的实现

如死区与极性控制模块框图所示,死区控制逻辑通过将 PWM 原始信号 Duty_0x 的上升沿和 Duty_0xN 的下降沿进行延迟控制实现。PWM1CR1 寄存器中 OUTMOD 位为 1 时, PWM 原始信号 Duty_0x/Duty_0xN 不会经过死区控制逻辑,即 PWM1x/PWM1xN 引脚输出为独立模式时,无死区; PWM1CR1 寄存器中 OUTMOD 位为 0 时, PWM 原始信号 Duty_0x/Duty_0xN 将经过死区逻辑,即 PWM1x/PWM1xN 引脚为互补模式时,将引入死区。(x=A, B, C)

死区时间由寄存器 PWM1D0/1 决定,死区控制方式由 PWM1CR2 寄存器的 DTMOD[1:0] 控制。

死区延迟时间 (Valid Edge Delay) = PWM1DTx * T_{PWM CLK}, x=0/1

DTMOD [1:0]	PWM1A 通道 上升沿死区	PWM1A 通道 下降沿死区	PWM1B 通道 上升沿死区	PWM1B 通道 下降沿死区	PWM1C 通道 上升沿死区	PWM1C 通道 下降沿死区
00	PWM1D0	PWM1D1	PWM1D0	PWM1D1	PWM1D0	PWM1D1
01	PWM1D1	PWM1D1	PWM1D0	PWM1D0	PWM1D0	PWM1D0
10	PWM1D0	PWM1D0	PWM1D1	PWM1D1	PWM1D0	PWM1D0
11	PWM1D0	PWM1D0	PWM1D0	PWM1D0	PWM1D1	PWM1D1

注 1: DTMOD[1:0] 只在互补模式下有效,因为在独立模式下无死区。PWM1A/B/C 的上升沿死区只会影响 PWM1A、PWM1B、PWM1C 的输出波形; PWM1A/B/C 的下降沿死区只会影响 PWM1AN、PWM1BN、PWM1CN 的输出波形。

注 2: 在不同的极性设置下,插入死区后的 PWM 波形也将不同,详见“不同极性设置下带死区的 PWM 波形”章节。

注 3: PWM1DTx = 0 时 (x=0, 1), 原始 PWM 波形对应的有效沿不会触发死区延时。

10.4.2 PWM 极性设置

为了有效说明 PWM 引脚上波形的极性,将一个 PWM 周期划分为“占空比时区 (Duty Zone)”与“非占空比时区 (None-Duty Zone)”,各种时基计数模式下,占空比时区和非占空比时区的划分参见章节 9.4.1~9.4.4 框图所示。由寄存器 PWM1CR1 中的位 PWMxS/PWMxNS 来决定占空比时区期间 DT_POL_0x 和 DT_POL_0xN 的电平。

PWMxS 控制 DT_POL_0x 的极性。PWMxS = 0 时, DT_POL_0x 的占空比时区期间为高电平,非占空比时区期间为低电平; PWMxS = 1 时, DT_POL_0x 的占空比时区期间为低电平,非占空比时区期间为高电平。

PWMxNS 控制 DT_POL_0xN 的极性。PWMxNS = 0 时, DT_POL_0xN 的占空比时区期间为低电平,非占空比时区期间为高电平; PWMxNS = 1 时, DT_POL_0xN 的占空比时区期间为高电平,非占空比时区期间为低电平。

结合章节 9.4.1~9.4.4 框图所示的 PWM 原始波形 Duty_0x/Duty_0xN 和章节 9.5 模块框图可以清晰的反映死区与极性的实现。

10.4.3 PWM 死区的自动重载

PWM 死区寄存器 (PWM1D0/1),也和占空比、周期一样,有对应的缓冲寄存器 (ShadowReg),自动重载功能仅可以在归零匹配信号发生时重载。

由于存在缓冲寄存器,对死区寄存器的读写有特殊规定:



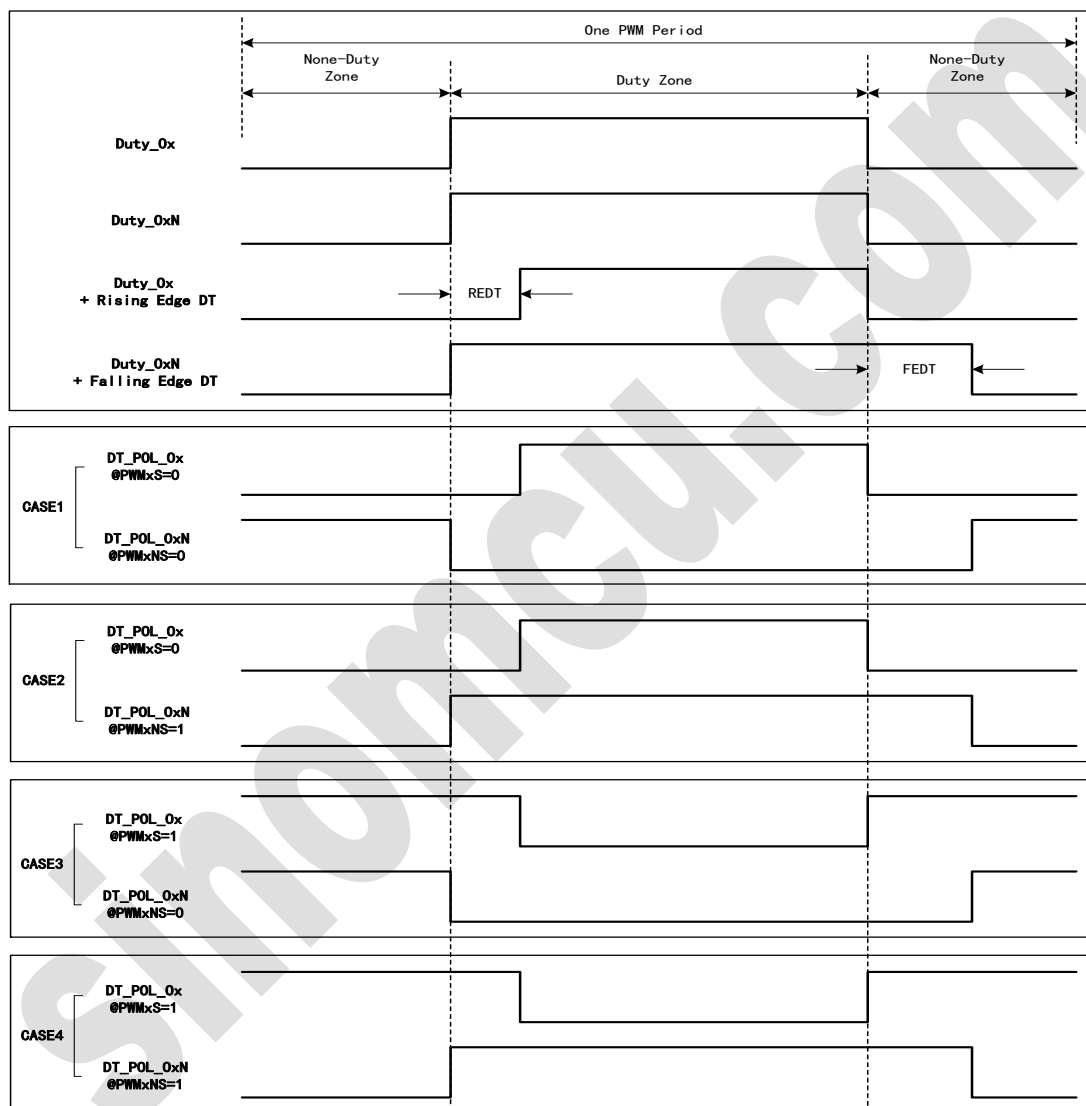
当 PWM1RLEN 寄存器的值设为 0x55 时，才允许对死区寄存器（PWM1DT0/1）进行修改，然后只有当 PWM1RLEN 寄存器中的值设为 0xAA 时，且分频后的归零信号发生时才会将对应寄存器值锁存至对应的缓存寄存器（ShadowReg）中，这样可以避免寄存器修改过程中发生重载，也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改，必须要有对 PWM1RLEN 寄存器的配合操作才能完成。

10.4.4 不同极性设置下带死区的 PWM 波形

下图给出了一对在 DTMOD[1:0]=00 时的 PWM 输出 DT_POL_0x/DT_POL_0xN 设置为互补输出模式，所有的极性组合下带有死区的波形。

注：PWM 启动第一个周期前，增加死区保护；中心对齐模式，计数 0 启动也会有额外增加死区控制，以防初始态上下桥开关管切换导致的瞬态导通。

不同极性设置下的死区波形



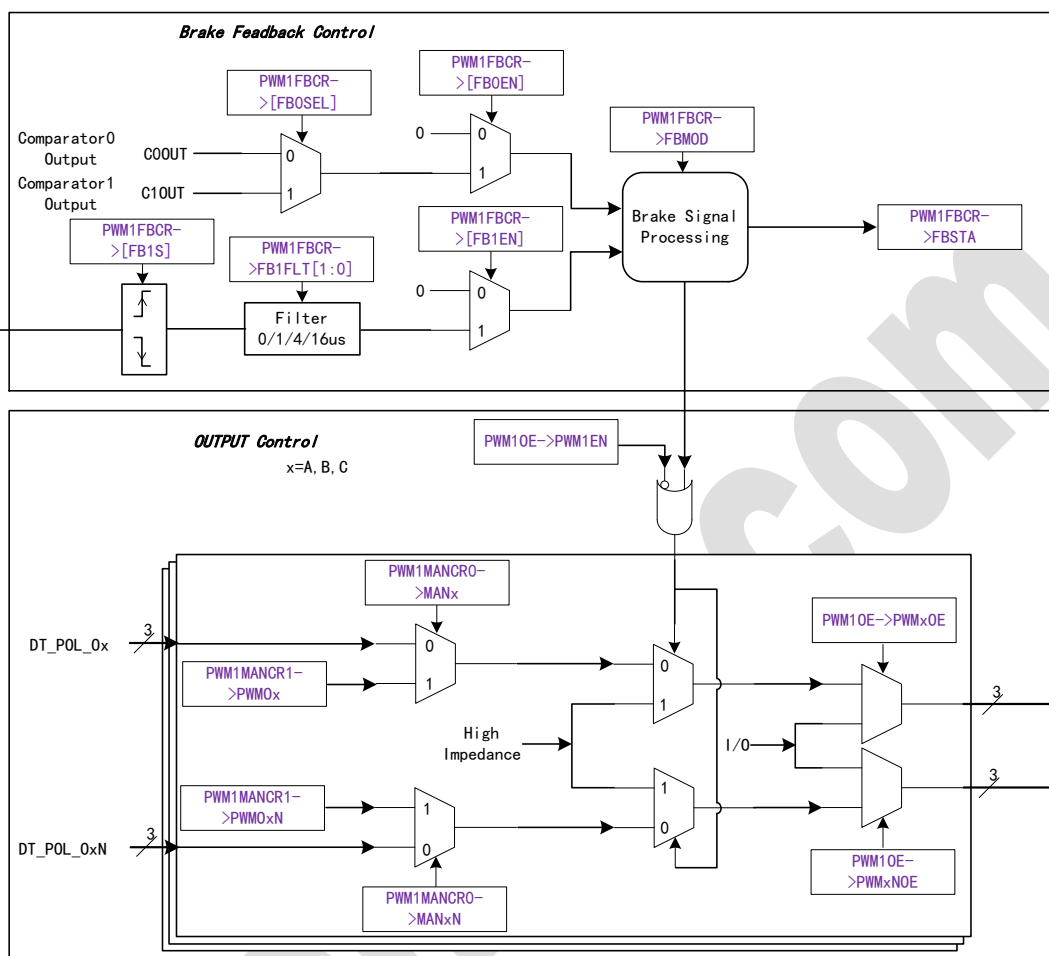
注：图中的 REDT 表示上升沿死区，FEDT 表示下降沿死区。上图是 DTMOD[1:0]=00 时的情况，DTMOD[1:0]为其他值时的情况，可参考 PWM1CR2 寄存器部分的说明。



10.5 PWM1A/B/C 输出控制与刹车检测

PWM 输出控制与刹车检测模块 (Output Control & Brake Feedback Control) 用于决定最终的 PWM 波形 DT_POL_Ox/DT_POL_OxN 是否在 PWM1x/PWM1xN 引脚上输出。

输出控制与刹车检测模块的逻辑框图



刹车检测模块 (Brake Feedback Control), 当 PWM1FBCR 寄存器中 FBOEN 位或 FB1EN 位为 1 时, 使能该模块功能, 若 FBOEN 和 FB1EN 都为 0, 则禁止该模块功能。

PWM 刹车检测应用场景:

当刹车事件发生时 (比如过流) 可以切断 PWM 的输出, 进入无效驱动状态 (输出高阻态), 进而达到保护外部功率器件的目的, 由于硬件控制, 因此响应速度非常快。参见章节 9.6<输出控制与刹车检测模块的逻辑框图>, 刹车检测模块结合输出控制模块 (Output Control) 能够实现前述的保护功能。

10.5.1 PWM 刹车信号输入源的选择

可以选择模拟比较器 0/1 的输出或 FB 引脚输入作为刹车检测输入信号, 参见章节 9.6<输出控制与刹车检测模块的逻辑框图>。若刹车信号来自比较器 0/1 输出, 高电平为有效信号, 即比较器输出由低变高时将自动切断 PWM 的输出; 若刹车信号来自 FB 引脚, 可以选择高电平或低电平有效, 对应的有效电平发生时将自动切断 PWM 的输出。

参见章节 9.6<输出控制与刹车检测模块的逻辑框图>, 若 FBOEN = 1, 则模拟比较器 0 或 1 的输出会接入刹车检测输入信号。一旦比较器的输出变高, 并保持滤波时间 (比较器模块控制), 则六路 PWM 引脚 PWM1x/PWM1xN 立即输出高阻态 (x=A, B, C)。如果 FBOEN=0, 则禁止模拟比较器的刹车检测功能。控制位 FBOSEL 控制模拟比较器 0 或 1 其中 1 路输出作为刹车检测输入信号, 同一时刻仅 1 路比较输出可被选择。

同理, 若 FB1EN=1, 则 FB 引脚输入接入刹车检测输入信号, 一旦 FB 引脚输入变高 (FB1S=0 时) 或变低 (FB1S=1 时), 并保持滤波时间, 则六路 PWM 引脚 PWM1x/PWM1xN 立即输出高阻态 (x=A, B, C)。如果 FB1EN=0, 则禁止 FB 引脚的刹车检测功能, FB 引脚作为普通 IO 使用 (但比较器的刹车检测功能不受影响)。

注: FBOEN 和 FB1EN 可以同时打开, 任意一路刹车信号有效, 都会自动切断 PWM 的输出。

PWM 应用场景: 逆变桥控制

下图为逆变桥控制电路, 六个管子为高电平驱动的功率管。

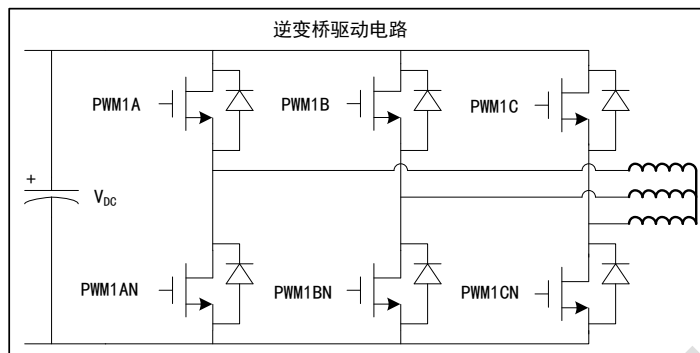
设置 PWMxS/PWMxNS = 0 (x=A, B, C), 则 PWM1A/B/C 的输出在占空比时区期间为高电平, PWM1AN/BN/CN 的输出在占



空比时区期间为低电平（见“PWM极性的设置”章节）；

设置 PWM1CR1 寄存器中 OUTMOD 位为 0，即 PWM1A 与 PWM1AN 互补输出、PWM1B 与 PWM1BN 互补输出、PWM1C 与 PWM1CN 互补输出。

使能刹车检测后，若发生刹车事件，则六路 PWM 立即输出高阻态。



10.5.2 PWM 刹车输入信号滤波

FB 引脚输入信号

当 FB1EN = 1 时，FB 引脚输入接入刹车检测输入信号，此时可以通过设置寄存器 PWM1FBCR 中的 FB1FLT[1:0] 位调整此输入信号的滤波时间。

FB1FLT [1:0] = 00b 时，无滤波作用；刹车检测输入检测到有效信号立即触发保护，自动切断 PWM 的输出。

FB1FLT [1:0] = 01/10/11b 时，对应滤波时间为 1 μ s、4 μ s、16 μ s；刹车检测输入的有效信号电平至少保持 FB1FLT [1:0] 定义的时间长度，才会触发保护，自动切断 PWM 的输出；检测到的无效电平也要至少保持 FB1FLT [1:0] 定义的时间长度，刹车检测才认为 FB 引脚的有效电平信号消失，PWM 输出根据保护模式（章节 9.6.3 介绍）执行动作。

当 FB1S=0 时，FB 刹车检测信号变为低电平有效；当 FLT2S=1 时，FB 刹车检测信号变为高电平有效。

模拟比较器 0/1 输出信号

FBOEN=1 时，模拟比较器 0 或 1 的输出会接入刹车检测输入信号，当比较器输出由低电平变为高电平时立即触发保护，比较器 0/1 的输出滤波时间和相关配置在模拟比较器相关寄存器中设置。（详见运算放大器和模拟比较器模块章节）

注：根据功率管的特性，恰当的设置滤波时间可以滤除刹车检测输入信号上的噪声。

10.5.3 PWM 刹车保护模式

刹车保护模式：逐次模式。

逐次模式

设置 PWM1FBCR 寄存器中 FBMOD=1，配置刹车保护模式为逐次模式。

在逐次模式下，六路 PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 是否输出 PWM 波形直接由模拟比较器滤波后的输出信号控制。当检测到有效刹车信号（比较器输出高电平），立即触发保护，六路 PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 立即输出高阻态。FBSTA 作为状态位使用，硬件置位，硬件清零；FBSTA 直接反映滤波之后的刹车信号状态，若滤波之后的刹车信号为高，则 FBSTA 位为 1；若滤波之后的刹车信号为低，则 FBSTA 位为 0。

若有效刹车信号没有消失，六路 PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 将保持输出高阻态，软件无法使六路 PWM 引脚恢复正常输出。

当有效刹车信号消失后，六路 PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 会在有效刹车信号消失后的最近一次 PWM 计数器归零匹配或周期匹配时自动恢复正常输出，FBSTA 位也将清零。

10.5.4 PWM 输出控制模块

PWM 输出引脚与 I/O 功能复用

PWM 引脚 PWM1x/PWM1xN (x=A, B, C) 与 I/O 功能复用，由寄存器 PWM1OE 控制。令 PWM1OE 中的 PWMxOE 位为 1，可以将 PWM1x 引脚设置为 PWM 输出功能，反之为 I/O 功能；令 PWM1OE 中的 PWMxNOE 位为 1，可以将 PWM1xN 引脚设置为 PWM 输出功能，反之为 I/O 功能。参见章节 9.6<输出控制与刹车检测模块的逻辑框图>。

手动控制 PWM 输出

当 PWMxOE/PWMxNOE 位为 1，且 PWM1EN 位为 1 时，若 PWMMANCRO 寄存器中的 MANx/MANxN 位为 1，则引脚 PWM1x/PWM1xN (x=A, B, C) 将输出 PWMMANCRO 寄存器中 PWM0x/PWM0xN 位的值；若 PWMMANCRO 寄存器中的 MANx/MANxN 位为 0，则引脚 PWM1x/PWM1xN (x=A, B, C) 将输出 PWM 波形 DT_POL_Ox/DT_POL_OxN。

此功能和 I/O 口的输出功能类似，唯一的区别是，在手动控制 PWM 输出模式下，其输出会受到 PWM 刹车检测输入信号控制，而在 I/O 输出模式下，不受该信号控制。参见章节 9.6<输出控制与刹车检测模块的逻辑框图>。

更改 PWMMANCRO 和 PWMMANCRO1 寄存器时，引脚 PWM1x/PWM1xN (x=A, B, C) 输出的改变与系统时钟同步，输出状态改



变立即生效。

注：无论 MAN_x/MAN_xN 是否为 1，当寄存器 $PWM1OE$ 中的 PWM_xOE/PWM_xNOE 为 0 时，引脚 $PWM1x/PWM1xN$ ($x=A, B, C$) 都将成为普通 I/O 口。

时基停止时 PWM 引脚的输出

$PWM1OE$ 中 $PWM1EN$ 位为 0 时，时基模块将停止工作，同时若 $PWM1OE$ 中 PWM_xOE/PWM_xNOE 位为 1，则 PWM_x/PWM_x1 引脚将输出高阻态。具体参见章节 9.6<输出控制与刹车检测模块的逻辑框图>。

10.6 事件触发功能

PWM 时基模块产生的分频后的归零匹配信号 (Scaled Zero Match) 和分频后的周期匹配信号 (Scaled Period Match) 可以用来触发启动 ADC 或者 Timer2。

10.6.1 PWM 计数器触发 Timer2

配置寄存器 $PWM1CR2$ 中的 $TRGTIM[1:0]$ 位且 Timer2 工作在自动触发模式 ($PWMTRG=1$)，可以设置 PWM 计数器自动触发启动 Timer2。触发方式配置参见下表。

PWM 计数器匹配触发启动 TIMER2 控制

PWM1CR2→ TRGTIM[1:0]	T2CON1→ SIGTRG	自动触发启动 Timer2 单次方式
00	x	PWM 模块不会触发启动 Timer2
01	1	中心对齐模式下，分频后的 PWM 周期匹配信号启动 Timer2 定时，溢出后自动停止定时 边沿对齐或单次模式下，分频后的 PWM 归零匹配信号启动 Timer2 定时，溢出后自动停止定时
10	1	中心对齐模式下，分频后的 PWM 归零匹配信号启动 Timer2 定时，溢出后自动停止定时 边沿对齐或单次模式下，分频后的 PWM 归零匹配信号启动 Timer2 定时，溢出后自动停止定时
11	1	中心对齐模式下，分频后的 PWM 周期匹配和归零匹配信号均启动 Timer2 定时，溢出后自动停止定时 边沿对齐或单次模式下，分频后的 PWM 归零匹配信号启动 Timer2 定时，溢出后自动停止定时

10.6.2 PWM 计数器触发 ADC

若寄存器 $PWM1CR2$ 中的 $ZTRGAD = 1$ ，则分频后的归零信号可以自动触发一次 ADC 序列的转换 (如果 ADC 模块中的 $ADCCR0→ADEN$ 位和 $ADCCR4→ADCTRG3$ 位被允许，将会启动一次序列转换)；若 $ZTRGAD = 0$ ，则归零信号不会自动触发 ADC 序列。

PWM1CR2→ ZTRGAD	ADCCR4→ ADCTRG3	ADCCR0→ ADEN	自动触发启动 ADC 方式
x	0	0	ADC 工作在非自动触发模式
0	1	1	PWM 模块分频后归零信号不会触发启动 ADC
1	1	1	PWM 模块分频后归零信号会触发启动 ADC

若寄存器 $PWM1CR2$ 中的 $PTRGAD = 1$ ，则分频后的周期匹配信号可以自动触发一次 ADC 序列的转换 (如果 ADC 模块中的 $ADCCR0→ADEN$ 位和 $ADCCR4→ADCTRG3$ 位被允许，将会启动一次序列转换)；若 $PTRGAD = 0$ ，则周期匹配信号不会自动触发 ADC 序列。

PWM1CR2→ PTRGAD	ADCCR4→ ADCTRG3	ADCCR0→ ADEN	自动触发启动 ADC 方式
x	0	0	ADC 工作在非自动触发模式
0	1	1	PWM 模块分频后周期匹配信号不会触发启动 ADC
1	1	1	PWM 模块分频后周期匹配信号会触发启动 ADC

上述，PWM 自动触发启动 AD，必须配置 ADC 模块中的 $ADCCR0→ADEN$ 位和 $ADCCR4→ADCTRG3$ 位被允许。

注：若 PWM 选择 HIRC 48MHz 时钟源，则 PWM1 无法硬件触发 ADC。

10.7 PWM 模块的中断总汇

PWM 模块的中断包括时基模块产生的 PWM 计数器归零中断、PWM 计数器值与周期寄存器匹配产生的周期中断 (详见



“时基中断”章节)以及 PWM 计数器值与占空比寄存器 PWM1xDL/H (x=A,B,C) 匹配时产生的占空比中断 (详见 “占空比中断” 章节)。这些中断共享一个中断向量 (详见 “中断” 章节)。

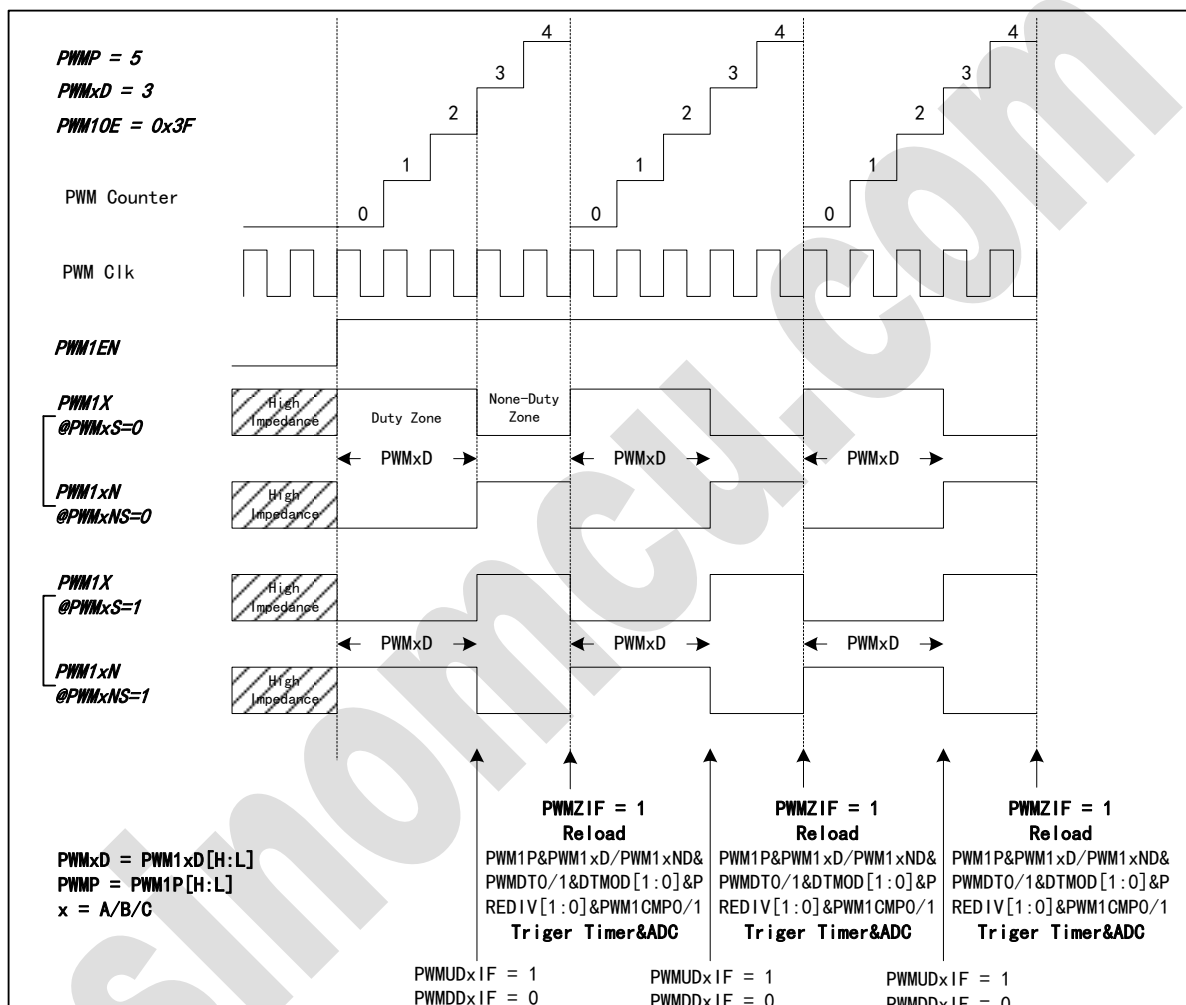
10.8 注意事项

10.8.1 边沿对齐计数模式中占空比寄存器值为 0 或周期值

边沿对齐计数模式中, 若占空比寄存器为 0 或周期值, 则根据不同的极性设置, PWMx/PWMx1 应输出占空比为 0%或 100%的波形。令 $PWMOE = 0xBF$ 、 $PWMDTOL/H = 0$ 、 $PWMDT1L/H = 0$ 在边沿对齐时基计数、互补输出模式下, 不同占空比值的 PWM 引脚输出的波形、中断标志触发位置、事件触发位置、寄存器重载位置如图 8-1-16~图 8-1-18 所示。

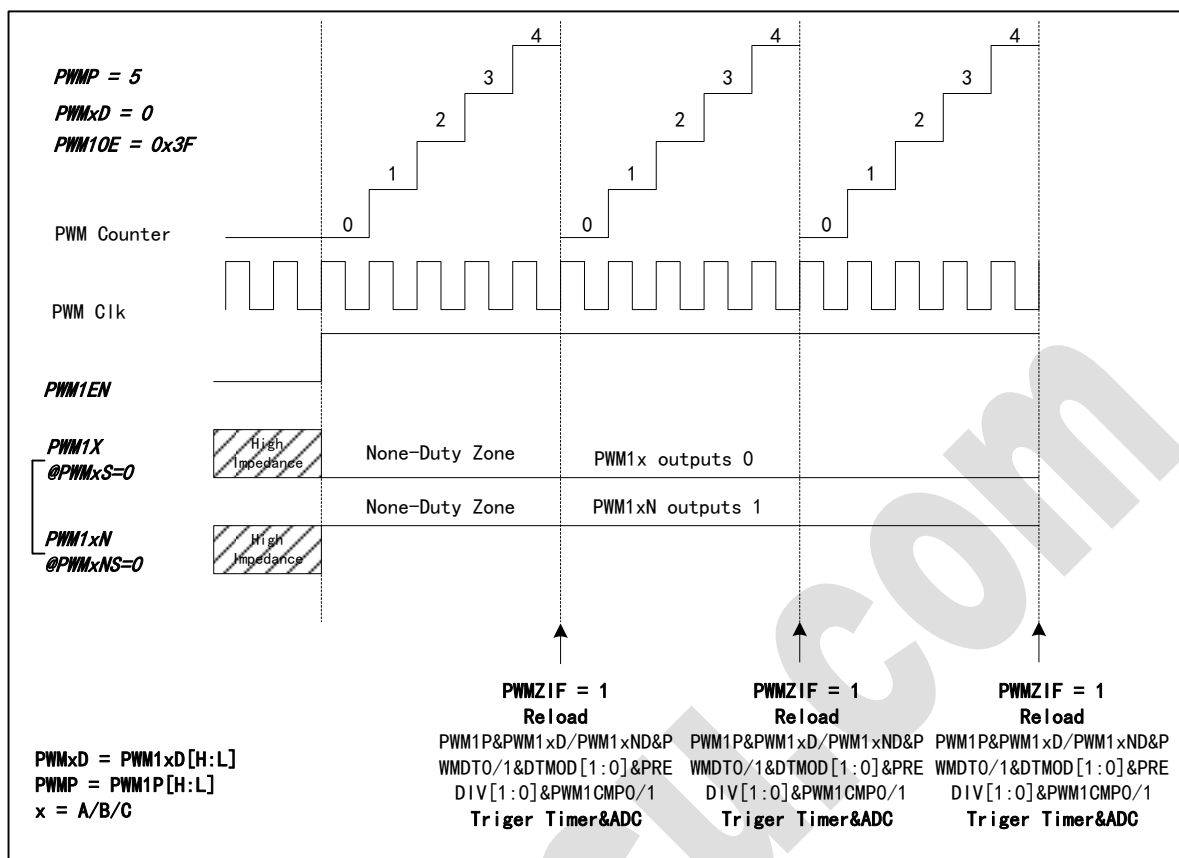
注: PWM 时基 Enable 时刻, 虽然时基计数器为 0, 但不会产生归零(zero match)信号, 因此不会置 PWMZIF 标志, 也不会产生事件触发信号, 而寄存器重载信号在 $PWMEN=0$ 时是立即发生的。

边沿对齐计数、互补输出模式、占空比大于 0&小于周期值



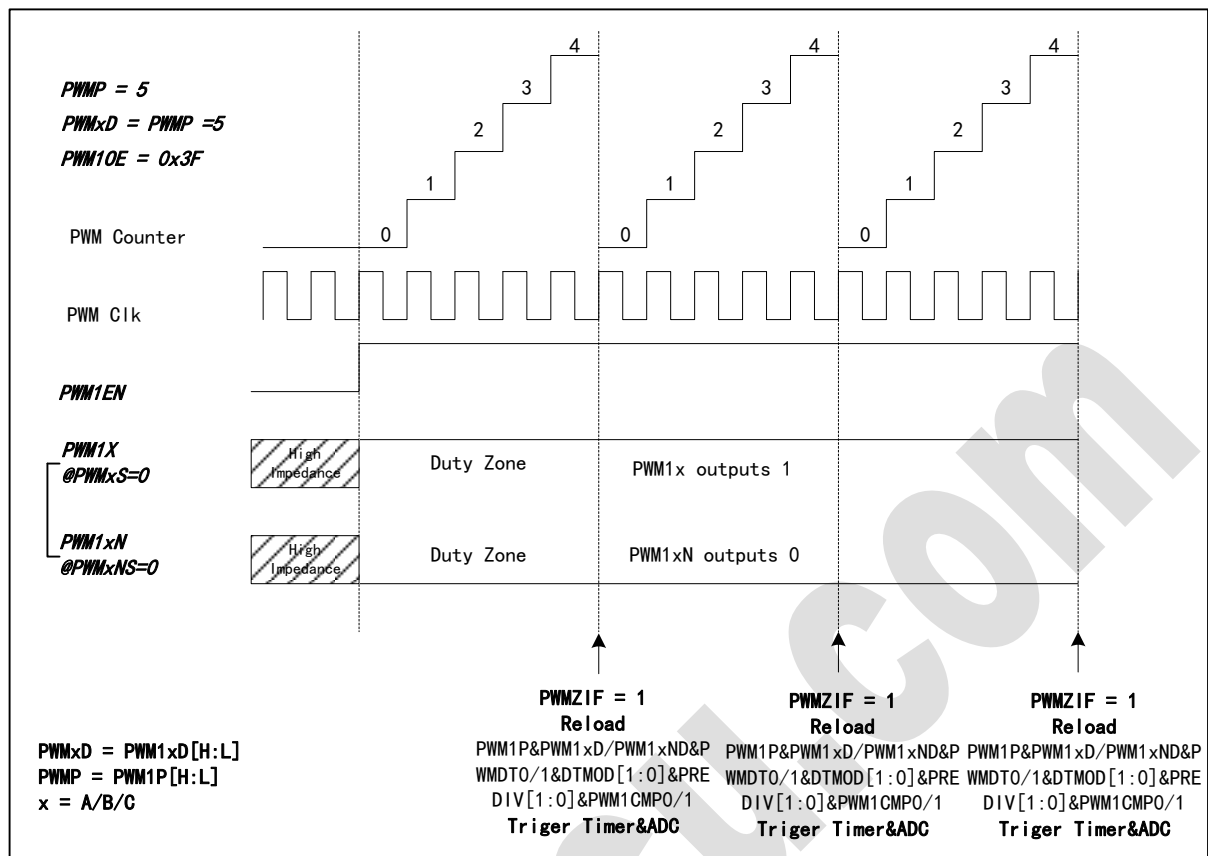


边沿对齐计数、互补输出模式，占空比=0





边沿对齐计数、互补输出模式，占空比=周期值



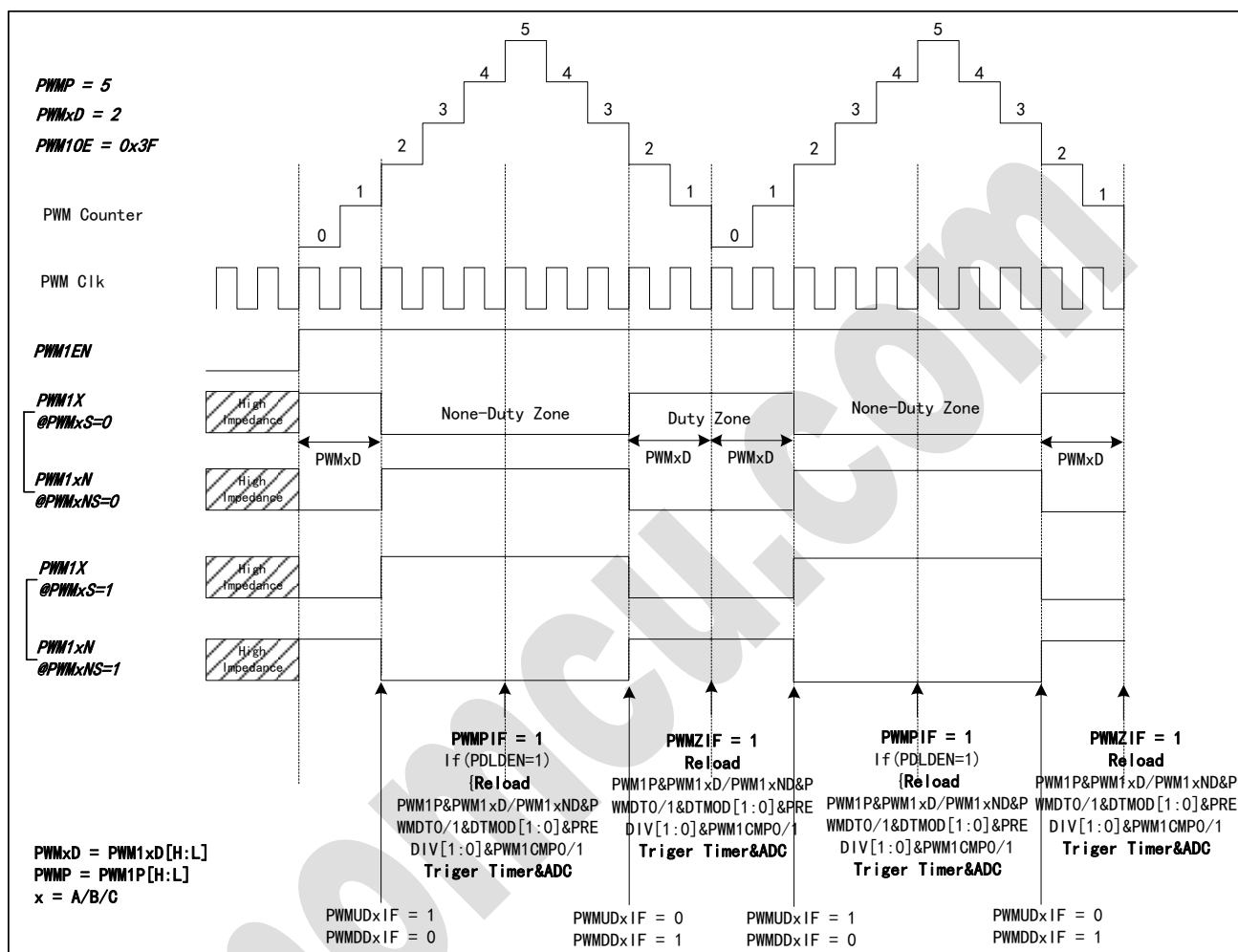


10.8.2 中心对齐计数模式中占空比寄存器值为 0 或周期值

中心对齐计数模式中，若占空比寄存器为 0 或周期值，则根据不同的极性设置，PWM1x/PWM1xN 应输出占空比为 0% 或 100% 的波形。令 PWM1OE = 0xBF、PWM1DTOL/H = 0、PWM1DT1L/H = 0，在边沿对齐计数、互补输出模式下，不同占空比值的 PWM 引脚输出的波形、中断标志触发位置、事件触发位置、寄存器重载位置如下图所示。

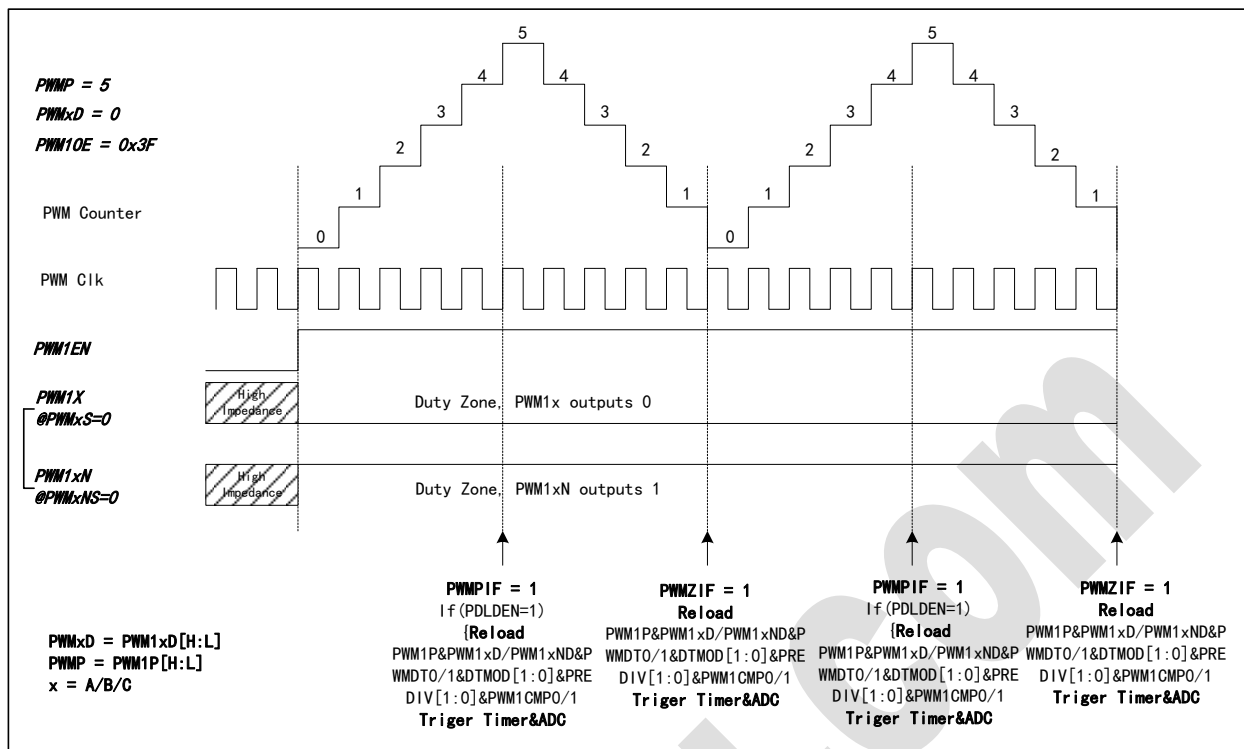
注：在中心对齐计数模式中，当占空比寄存器为 0 或周期值时，占空比较匹配中断标志不会置起。

中心对齐计数、互补输出模式、占空比大于 0 且小于周期值

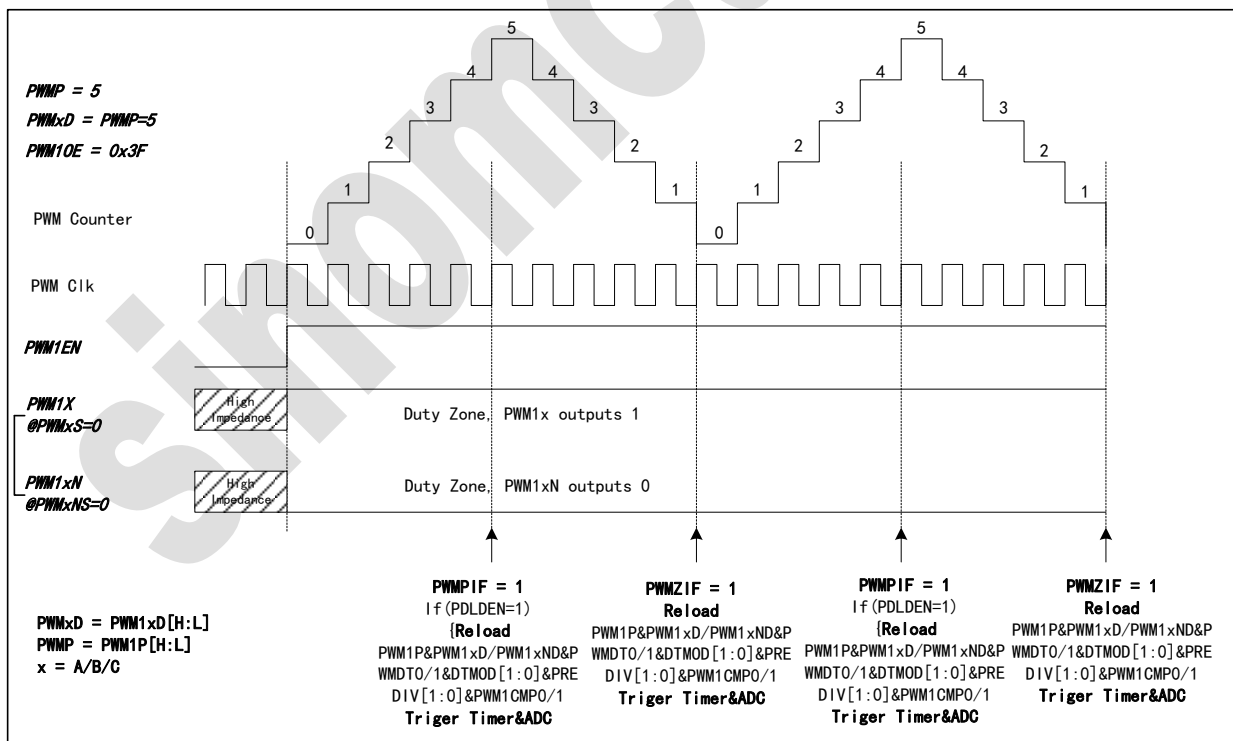




中心对齐计数、互补输出模式、占空比=0



中心对齐计数、互补输出模式、占空比=周期值



10.8.3 周期寄存器值为 0

不管时基计数器在何种模式，若周期寄存器为 0，则根据不同的极性设置，PWM1x/PWM1xN 应输出占空比为 0%的波



形。此时，所有中断标志都不会产生，也不会产生事件触发信号，但重载信号会产生。

10.8.4 仿真状态的 PWM 输出

支持在线仿真，在仿真状态运行和正常状态运行没有区别，如果在仿真状态停止运行或单步运行，PWM 输出会切换为高阻态，保证不会误触发外部功率管。

10.8.5 进入 STOP 模式，PWM1A/B/C 的波形输出状态

进入 STOP 模式后，PWM 输出口不受影响，比如 PWM0 口原先输出 1K 的方波，进入 STOP 模式后会继续输出 1K 的方波，但是如果设置了 PWM1 中断，会唤醒 STOP 模式，继续执行 IDLE 后面的程序。

10.8.6 进入 SLEEP 模式，PWM1A/B/C 的波形输出状态

进入 SLEEP 模式后，PWM 输出口会输出高阻状态，即使设置了 PWM1 中断，也不能唤醒 SLEEP 模式，SLEEP 模式必须通过其他方式唤醒，具体参考工作模式章节。

10.8.7 周期寄存器和占空比寄存器写入

周期寄存器和占空比寄存器，由 2 个 8 位寄存器组合，写入顺序必须先写低位再写高位。代码编写过程，请避免调用 SFR16 宏命令定义占空比寄存器（SFR 区），以及避免使用 16 位变量定义周期寄存器（XSFR 区），以防止高低位写入反序。



10.9 代码示例

以下是输出 PWM 波形的一个简单示例程序

```
#include "MC51Fxxxx.h" /*用户配置字选择单时钟源内部 24M HIRC*/
void Pwm1_Isr(void) interrupt 11 using 1
{
    PWM1IF = 0;
    PWM1RLEN = 0x55;           //解保护，允许修改 PWM 受保护寄存器
    PWM1AD16++;                //每次中断将 PWM 占空比加 1
    PWM1RLEN = 0xAA;           //启动加载
}
void System_Initial()
{
    EA = 1;
    IE1 |= 0x04;
}
void Pwm1_Initial()
{
    PWM1RLEN = 0x55;           //解保护，允许修改 PWM 受保护寄存器
    PWM1OE = 0x3f;             //将 6 个 I/O 口设置为 PWM 输出口
    PWM1CR0 = 0x00;            //PWM 时钟选择系统时钟，设置边沿对齐模式，预分频为 1，后分频为 1
    PWM1CR1 = 0x00;            //PWM 输出互补模式 (PWM1x 与 PWM1xN 波形互补，x=A,B,C)
    PWM1PL = 0x70;             //PWM 的周期设为 6000 (0x1770) Fsys clk, 先写低位再写高位
    PWM1PH = 0x17;
    PWM1ADL = 0;               //PWM1A 占空比设置为 0 Fsys clk
    PWM1ADH = 0;
    PWM1BDL = 0xE8;            //PWM1B 占空比设置为 1000 (0x03E8) Fsys clk
    PWM1BDH = 0x03;
    PWM1CDL = 0xB8;            //PWM1C 占空比设置为 3000 (0x0BB8) Fsys clk
    PWM1CDH = 0x0B;
    PWM1IE = 0x40;             //PWM 归零匹配中断允许
    PWM1OE |= 0x80;            //打开 PWM 总时基
    PWM1RLEN = 0xAA;           //启动加载，将 PWM 带缓存的寄存器值载入 (周期，占空比，死区等)
}

void main()
{
    System_Initial();
    Pwm1_Initial();
    while(1);
}
```

10.10 增强型 PWM1 寄存器

PWM1 模块所使用寄存器如下：

功能	名称	寄存器描述
PWM 计数器时钟 (时基) 设置	PWM1CR0	设置 PWM 时基的预分频、PWM 模块中控制信号的后分频以及 PWM 计数器的运行模式
PWM 控制寄存器	PWM1CR1	设置 PWM 波形的输出极性及输出模式 (互补/独立)
	PWM1CR2	PWM 触发事件的使能、重载以及死区方式选择
PWM 输出控制	PWM1OE	设置 PWM 引脚输出 PWM 模块产生的信号或者作为普通 I/O 口
	PWM1MANCRO	设置 PWM 模块输出 PWM 波形或直接输出 PWM1MANCR1 中定义的电平
	PWM1MANCR1	定义 PWM 模块直接输出的电平
PWM 死区控制	PWM1DTOL	设置 PWM 原始波形上升沿触发死区时间
	PWM1DTIL	设置 PWM 原始波形下降沿触发死区时间
刹车保护	PWM1FBCR	设置刹车保护功能



PWM 中断	PWM1IE	PWM 中断使能位控制
	PWM1IF	PWM 中断标志位
寄存器重载使能控制	PWM1RLN	禁止/使能占空比寄存器/周期寄存器/PWM 时基预分频的重载
PWM 周期寄存器	PWM1PL	16 位周期寄存器
	PWM1PH	
PWM 占空比寄存器	PWM1ADL	独立输出模式 PWM1xDL/H (x=A, B, C) 控制 PWM1x (x=A, B, C) 通道原始波形 PWM1xNDL/H (x=A, B, C) 控制 PWM1xN (x=A, B, C) 通道原始波形 互补输出模式+对称波形 PWM1xDL/H (x=A, B, C) 同时控制 PWM1x 和 PWM1xN (x=A, B, C) 通道原始波形 PWM1xNDL/H (x=A, B, C) 无效 互补输出模式+非对称波形（仅在中心对齐计数模式下有效） PWM1xDL/H (x=A, B, C) 同时控制 PWM1x 和 PWM1xN (x=A, B, C) 通道 增计数 过程原始波形 PWM1xNDL/H (x=A, B, C) 同时控制 PWM1x 和 PWM1xN (x=A, B, C) 通道 减计数 过程原始波形
	PWM1ADH	
	PWM1ANDL	
	PWM1ANDH	
	PWM1BDL	
	PWM1BDH	
	PWM1BNDL	
	PWM1BNDH	
	PWM1CDL	
	PWM1CDH	
	PWM1CNDL	
	PWM1CNDH	

PWM1 寄存器汇总表

地址 (SFRn/ H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
XSFR/F F48	PWM1CR 0	PWM1 控制寄存器 0	CLKSEL	-	CTMOD1	CTMOD0	PREDIV1	PREDIV0	POSTDIV 1	POSTDIV 0	0-00 0000
SFR0/B D	PWM1CR 1	PWM1 控制寄存器 1	OUTMOD	SYMC	PWMCNS	PWMBNS	PWMANS	PWMCSS	PWMBSS	PWMAS	0000 0000
SFR0/B E	PWM1CR 2	PWM1 控制寄存器 2	TRGTIM1	TRGTIMO	PTRGAD	ZTRGAD	PLDEN	OSYNC	DTMOD1	DTMOD0	0000 0-00
XSFR/F F4A	PWM1PH	PWM1 周期寄存器高位	PWMP15	PWMP14	PWMP13	PWMP12	PWMP11	PWMP10	PWMP9	PWMP8	0000 0000
XSFR/F F4B	PWM1PL	PWM1 周期寄存器低位	PWMP7	PWMP6	PWMP5	PWMP4	PWMP3	PWMP2	PWMP1	PWMP0	0000 0000
SFR0/B 8	PWM1IF	PWM1 中断标志寄存器	PWMP1F	PWMP2F	PWMP3F	PWMP4F	PWMP5F	PWMP6F	PWMP7F	PWMP8F	0000 0000
SFR0/D 2	PWM1AD L	PWM1A 占空比寄存器低位	PWMAD7	PWMAD6	PWMAD5	PWMAD4	PWMAD3	PWMAD2	PWMAD1	PWMAD0	0000 0000
SFR0/D 3	PWM1AD H	PWM1A 占空比寄存器高位	PWMAD15	PWMAD14	PWMAD13	PWMAD12	PWMAD11	PWMAD10	PWMAD9	PWMAD8	0000 0000
SFR0/D 4	PWM1BD L	PWM1B 占空比寄存器低位	PWMBD7	PWMBD6	PWMBD5	PWMBD4	PWMBD3	PWMBD2	PWMBD1	PWMBD0	0000 0000
SFR0/D 5	PWM1BD H	PWM1B 占空比寄存器高位	PWMBD15	PWMBD14	PWMBD13	PWMBD12	PWMBD11	PWMBD10	PWMBD9	PWMBD8	0000 0000
SFR0/D 6	PWM1CD L	PWM1C 占空比寄存器低位	PWMCD7	PWMCD6	PWMCD5	PWMCD4	PWMCD3	PWMCD2	PWMCD1	PWMCD0	0000 0000
SFR0/D 7	PWM1CD H	PWM1C 占空比寄存器高位	PWMCD15	PWMCD14	PWMCD13	PWMCD12	PWMCD11	PWMCD10	PWMCD9	PWMCD8	0000 0000
SFR0/D 9	PWM1RL EN	PWM1 寄存器修改重载控制寄存器	PWMRL7	PWMRL6	PWMRL5	PWMRL4	PWMRL3	PWMRL2	PWMRL1	PWMRL0	0000 0000
SFR0/D A	PWM1AN DL	PWM1AN 占空比寄存器低位	PWMAND7	PWMAND6	PWMAND5	PWMAND4	PWMAND3	PWMAND2	PWMAND1	PWMAND0	0000 0000
SFR0/D B	PWM1AN DH	PWM1AN 占空比寄存器高位	PWMAND15	PWMAND14	PWMAND13	PWMAND12	PWMAND11	PWMAND10	PWMAND9	PWMAND8	0000 0000
SFR0/D C	PWM1BN DL	PWM1BN 占空比寄存器低位	PWMBND7	PWMBND6	PWMBND5	PWMBND4	PWMBND3	PWMBND2	PWMBND1	PWMBND0	0000 0000
SFR0/D D	PWM1BN DH	PWM1BN 占空比寄存器高位	PWMBND15	PWMBND14	PWMBND13	PWMBND12	PWMBND11	PWMBND10	PWMBND9	PWMBND8	0000 0000



SFR0/D E	PWM1CN DL	PWM1CN 占空比寄存器低位	PWMCND7	PWMCND6	PWMCND5	PWMCND4	PWMCND3	PWMCND2	PWMCND1	PWMCND0	0000 0000
SFR0/D F	PWM1CN DH	PWM1CN 占空比寄存器高位	PWMCND1 5	PWMCND1 4	PWMCND1 3	PWMCND1 2	PWMCND1 1	PWMCND1 0	PWMCND9	PWMCND8	0000 0000
SFR0/B 6	PWM1MA NCR0	PWM1 手动输出控制寄存器 0	-	-	MANCN	MANBN	MANAN	MANC	MANB	MANA	--00 0000
SFR0/B 7	PWM1MA NCR1	PWM1 手动输出控制寄存器 1	-	-	PWMOCN	PWMOBN	PWMOAN	PWMOC	PWMOB	PWMOA	--00 0000
SFR0/D 1	PWM1OE	PWM1 输出使能寄存器	PWM1EN	-	PWMCNOE	PWMBNOE	PWMANOE	PWMCOE	PWMBOE	PWMAOE	0-00 0000
SFR0/B F	PWM1FB CR	PWM1 刹车控制寄存器	FBOEN	FBOSEL	FB1EN	FB1S	FB1FLT1	FB1FLT0	FBMOD	FBSTA	0000 0001
XSFR/F F49	PWM1IE	PWM1 中断使能控制寄存器	PWMP1E	PWMZ1E	PWMDDC1 E	PWMUDC1 E	PWMDDB1 E	PWMUDB1 E	PWMDDA1 E	PWMUDA1 E	0000 0000
XSFR/ FF4D	PWM1DT OL	PWM1 死区控制寄存器 0 低位	PWMDT07	PWMDT06	PWMDT05	PWMDT04	PWMDT03	PWMDT02	PWMDT01	PWMDT00	0000 0000
XSFR/ FF4F	PWM1DT 1L	PWM1 死区控制寄存器 1 低位	PWMDT17	PWMDT16	PWMDT15	PWMDT14	PWMDT13	PWMDT12	PWMDT11	PWMDT10	0000 0000

10.10.1 PWM1 控制寄存器 0 (PWM1CR0, 0xFF48/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR0	CLKSEL	-	CTMOD1	CTMOD0	PREDIV1	PREDIV0	POSTDIV1	POSTDIV0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	-	0	0	0	0	0	0

BIT[7] CLKSEL - PWM 时钟源选择

00: 选择系统时钟 F_{sys}

01: 选择 HIRC48M 时钟

BIT[5:4] CTMOD[1:0] - 计数工作模式选择

00: 边沿对齐计数模式, 在此模式下, 时基计数器只会发生归零匹配

01: 中心对齐计数模式, 在此模式下, 时基计数器会发生归零匹配和周期匹配

10: 单次计数模式, 在此模式下, 时基计数器只会发生归零匹配

11: 保留

BIT[3:2] PREDIV[1:0] - 时钟预分频选择

00: PWM 计数器时钟 = $F_{pwm}/1$

01: PWM 计数器时钟 = $F_{pwm}/2$

10: PWM 计数器时钟 = $F_{pwm}/4$

11: PWM 计数器时钟 = $F_{pwm}/8$

注: F_{pwm} 由 CLKSEL 控制, 选择系统时钟 F_{sys} 或 HIRC48M 时钟。

BIT[1:0] POSTDIV[1:0] - 时钟后分频选择

00: 无后分频功能

01: 中断、重载信号与事件触发信号将 2 分频

10: 中断、重载信号与事件触发信号将 4 分频

11: 中断、重载信号与事件触发信号将 8 分频

10.10.2 PWM1 控制寄存器 1 (PWM1CR1, 0xBD/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR1	OUTMOD	SYMC	PWMCNS	PWMBNS	PWMANS	PWMCS	PWMBBS	PWMAS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] OUTMOD - 输出模式选择

0: 3 路互补 PWM 输出

1: 6 路独立 PWM 输出

BIT[6] SYMC - 对称控制 (仅在中心对齐计数模式下互补输出时有效)



- 0: 互补输出对称 PWM 波形
 1: 互补输出非对称 PWM 波形
- BIT[5, 4, 3] **PWMxNS** (x=A, B, C) - PWMxN 通道输出极性选择, x=A, B, C
 0: PWMxN 占空比期间输出低电平, 其余时间输出高电平
 1: PWMxN 占空比期间输出高电平, 其余时间输出低电平
- BIT[2, 1, 0] **PWMxS** (x=A, B, C) - PWMx 通道输出极性选择, x=A, B, C
 0: PWMx 占空比期间输出高电平, 其余时间输出低电平
 1: PWMx 占空比期间输出低电平, 其余时间输出高电平

10.10.3 PWM1 控制寄存器 2 (PWM1CR2, 0xBE/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CR2	TRGTIM1	TRGTIMO	PTRGAD	ZTRGAD	PLDEN	OSYNC	DTMOD1	DTMOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- BIT[7:6] **TRGTIM[1:0]** - PWM 计数器匹配触发启动 TIMER2 控制
 00: PWM 模块不会触发启动 Timer2
 01: 中心对齐模式下, 分频后的 PWM 周期匹配信号启动 Timer2
 边沿对齐或单次模式下, 分频后的 PWM 归零匹配信号启动 Timer2
 10: 中心对齐模式下, 分频后的 PWM 归零匹配信号启动 Timer2
 边沿对齐或单次模式下, 分频后的 PWM 归零匹配信号启动 Timer2
 11: 中心对齐模式下, 分频后的 PWM 周期匹配和归零匹配信号均启动 Timer2
 边沿对齐或单次模式下, 分频后的 PWM 归零匹配信号启动 Timer2
 (Timer2 的 PWM 触发控制开启后, 上述设置才有效)
- BIT[5] **PTRGAD** - PWM 计数器与周期匹配启动 AD 控制
 0: PWM 模块周期匹配时不会触发启动 AD
 1: PWM 时基计数器与周期寄存器匹配时启动 AD
 (ADC 的 ADEN 和 ADCTRG3 位置 1 的情况下)
- BIT[4] **ZTRGAD** - PWM 计数器归 0 匹配启动 AD 控制
 0: PWM 模块归 0 时刻不会触发启动 AD
 1: PWM 时基计数器归 0 时刻启动 AD
 (ADC 的 ADEN 和 ADCTRG3 位置 1 的情况下)
- BIT[3] **PLDEN** - PWM 计数器周期匹配时重载占空比寄存器使能位: (只在中心对齐模式下有效)
 0: 周期匹配时刻不允许重载占空比寄存器
 1: 周期匹配时刻允许重载占空比寄存器

注:

- 1、无论寄存器如何设置, 周期匹配时不会重载 PWM 周期寄存器和死区寄存器。
- 2、只在中心对齐模式下周期匹配发生时, 由 PLDEN 控制, 占空比寄存器是否重载。
- 2: 归 0 匹配时会自动重载周期寄存器、占空比寄存器、死区寄存器以及死区方式选择 DTMOD[1:0], 此重载动作不受上述控制位影响, **PWM1RLEN = 0XAA** 条件依然有效。

- BIT[2] **OSYNC** - 手动修改 PWMx/PWMxN 输出同步位
 此位保留。

- BIT[1:0] **DTMOD[1:0]** - PWM 死区寄存器控制方式
 00: PWMA/B/C 的前死区由 PWMDT0 设置;
 PWMA/B/C 的后死区由 PWMDT1 设置
 其他: 无效

注: DTMOD[1:0] 只在互补模式下有效, 因为在独立模式下无死区。PWMA/B/C 的前死区只会影响 PWMA、PWMB、PWMC 的输出波形; PWMA/B/C 的后死区只会影响 PWMAN、PWMBN、PWMCN 的输出波形, 具体参见死区章节。

10.10.4 PWM1 周期寄存器高位 (PWM1PH, 0xFF4A/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PH	PWMP15	PWMP14	PWMP13	PWMP12	PWMP11	PWMP10	PWMP9	PWMP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- BIT[7:0] **PWMP[15:8]** - PWM1 周期寄存器高位

10.10.5 PWM1 周期寄存器低位 (PWM1PL, 0xFF4B/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	-------	-------	-------	-------	-------	-------	-------	-------



PWM1PL	PWMP7	PWMP6	PWMP5	PWMP4	PWMP3	PWMP2	PWMP1	PWMP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMP[7:0] - PWM1 周期寄存器低位

注：16 位周期数据载入和更新，必须先写低位再写高位。

10.10.6 PWM1A 占空比寄存器低位 (PWM1ADL, 0xD2/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1ADL	PWMAD7	PWMAD6	PWMAD5	PWMAD4	PWMAD3	PWMAD2	PWMAD1	PWMAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMAD[7:0] - PWM1A 占空比寄存器低位

注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.7 PWM1A 占空比寄存器高位 (PWM1ADH, 0xD3/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1ADH	PWMAD15	PWMAD14	PWMAD13	PWMAD12	PWMAD11	PWMAD10	PWMAD9	PWMAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMAD[15:8] - PWM1A 占空比寄存器高位

10.10.8 PWM1B 占空比寄存器低位 (PWM1BDL, 0xD4/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BDL	PWMBD7	PWMBD6	PWMBD5	PWMBD4	PWMBD3	PWMBD2	PWMBD1	PWMBD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMD[7:0] - PWM1B 占空比寄存器低位

注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.9 PWM1B 占空比寄存器高位 (PWM1BDH, 0xD5/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BDH	PWMBD15	PWMBD14	PWMBD13	PWMBD12	PWMBD11	PWMBD10	PWMBD9	PWMBD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMBD[15:8] - PWM1B 占空比寄存器高位

10.10.10 PWM1C 占空比寄存器低位 (PWM1CDL, 0xD6/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CDL	PWPCD7	PWPCD6	PWPCD5	PWPCD4	PWPCD3	PWPCD2	PWPCD1	PWPCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWPCD[7:0] - PWM1C 占空比寄存器低位

注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.11 PWM1C 占空比寄存器高位 (PWM1CDH, 0xD7/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CDH	PWPCD15	PWPCD14	PWPCD13	PWPCD12	PWPCD11	PWPCD10	PWPCD9	PWPCD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWPCD[15:8] - PWM1C 占空比寄存器高位

10.10.12 PWM1AN 占空比寄存器低位 (PWM1ANDL, 0xDA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1ANDL	PWMAND7	PWMAND6	PWMAND5	PWMAND4	PWMAND3	PWMAND2	PWMAND1	PWMAND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMAND[7:0] - PWM1AN 占空比寄存器低位



注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.13 PWM1AN 占空比寄存器高位 (PWM1ANDH, 0xDB/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1ANDH	PWMAND15	PWMAND14	PWMAND13	PWMAND12	PWMAND11	PWMAND10	PWMAND9	PWMAND8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMAND[15:8] - PWM1AN 占空比寄存器高位

10.10.14 PWM1BN 占空比寄存器低位 (PWM1BNDL, 0xDC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BNDL	PWMBND7	PWMBND6	PWMBND5	PWMBND4	PWMBND3	PWMBND2	PWMBND1	PWMBND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMBND[7:0] - PWM1BN 占空比寄存器低位

注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.15 PWM1BN 占空比寄存器高位 (PWM1BNDH, 0xDD/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BNDH	PWMBND15	PWMBND14	PWMBND13	PWMBND12	PWMBND11	PWMBND10	PWMBND9	PWMBND8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMBND[15:8] - PWM1BN 占空比寄存器高位

10.10.16 PWM1CN 占空比寄存器低位 (PWM1CNDL, 0xDE/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CNDL	PWMCND7	PWMCND6	PWMCND5	PWMCND4	PWMCND3	PWMCND2	PWMCND1	PWMCND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMCND[7:0] - PWM1CN 占空比寄存器低位

注：16 位占空比数据载入和更新，必须先写低位再写高位。

10.10.17 PWM1CN 占空比寄存器高位 (PWM1CNDH, 0xDF/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CNDH	PWMCND15	PWMCND14	PWMCND13	PWMCND12	PWMCND11	PWMCND10	PWMCND9	PWMCND8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMCND[15:8] - PWM1CN 占空比寄存器高位

10.10.18 PWM1 输出使能寄存器 (PWM1OE, 0xD1/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1OE	PWM1EN	-	PWMCNOE	PWMBNOE	PWMANOE	PWMCOE	PWMBOE	PWMAOE
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	-	0	0	0	0	0	0

BIT[7] PWM1EN - PWM1 总使能位

0: 关闭 PWMA/B/C/AN/BN/CN 时钟

1: 打开 PWMA/B/C/AN/BN/CN 时钟

BIT[5, 4, 3] PWMxNOE (x=A, B, C) - PWM1xN 输出使能位 (x=A, B, C)

0: PWM1xN 引脚作为普通 I/O 口 (x=A, B, C)

1: PWM1xN 引脚作为 PWM 输出口, 输出 PWM 波形或 PWMOxN 定义的电平 (x=A, B, C)

BIT[2, 1, 0] PWMxOE (x=A, B, C) - PWM1x 输出使能位 (x=A, B, C)

0: PWM1x 引脚作为普通 I/O 口 (x=A, B, C)

1: PWM1x 引脚作为 PWM 输出口, 输出 PWM 波形或 PWMOx 定义的电平 (x=A, B, C)

10.10.19 PWM1 手动输出控制寄存器 0 (PWM1MANCRO, 0xB6/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1MANCRO	-	-	MANCN	MANBN	MANAN	MANC	MANB	MANA
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W



复位值	-	-	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[5, 4, 3] **MANxN** (x=A, B, C) - PWM1xN 输出手动控制位, x=A, B, C

(只有当 PWMxNOE=1 时有效)

0: PWMxN 口输出 PWM 波形

1: PWMxN 口输出由 PWMMANCR2 寄存器中的 PWM0xN 位控制

BIT[2, 1, 0] **MANx** (x=A, B, C) - PWM1x 输出手动控制位, x=A, B, C

(只有当 PWMxOE=1 时有效)

0: PWMx 口输出 PWM 波形

1: PWMx 口输出由 PWMMANCR2 寄存器中的 PWM0x 位控制

10.10.20 PWM1 手动输出控制寄存器 1 (PWM1MANCR1, 0xB7/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1MANCR1	-	-	PWMOCN	PWMOBN	PWMOAN	PWMOC	PWMOB	PWMOA
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	0	0	0	0	0	0

BIT[5, 4, 3] **PWMOxN** (x=A, B, C) - MANxN = 1 时, 决定 PWMxN 口的输出电平, x=A, B, C

0: 输出 0

1: 输出 1

BIT[2, 1, 0] **PWMOx** (x=A, B, C) - MANxN = 1 时, 决定 PWMxN 口的输出电平, x=A, B, C

0: 输出 0

1: 输出 1

注: 必须先写 PWMMANCR1 寄存器, 后写 PWMMANCR0 寄存器。

10.10.21 PWM1 刹车控制寄存器 (PWM1FBCR, 0xBF/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1FBCR	FBOEN	FBOSEL	FB1EN	FB1S	FB1FLT1	FB1FLT0	FBMOD	FBSTA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	1

BIT[7] **FBOEN** - 刹车信号 0 使能控制位

0: 刹车信号 0 功能禁止

1: 刹车信号 0 功能打开, 输入源可以选择比较器 0 或比较器 1

BIT[6] **FBOSEL** - 刹车信号 0 输入源选择控制位

0: 比较器 0 滤波后的输出作为 PWM 刹车信号 0 输入源, 高电平有效

1: 比较器 1 滤波后的输出作为 PWM 刹车信号 0 输入源, 高电平有效

BIT[5] **FB1EN** - 刹车信号 1 使能控制位

0: 刹车信号 1 功能禁止

1: 刹车信号 1 功能打开, 输入源为 FB 引脚

BIT[4] **FB1S** - 刹车信号 1 有效电平选择位

0: 刹车信号 1 FB 输入高电平有效

1: 刹车信号 1 FB 输入低电平有效

BIT[3:2] **FB1FLT[1:0]** - 刹车信号 1 输入滤波选择

00: 无滤波

01: 滤波时间常数为 1us (24clocks@HIRC/2)

10: 滤波时间常数为 4us (96 clock@HIRC/2)

11: 滤波时间常数为 16us (384 clock@HIRC/2)

注: 上述滤波常数时间不是精确值, 仅供参考。

BIT[1] **FBMOD** - 刹车功能模式选择

0/1: 逐次模式, PWM 输出直接由刹车信号输入端来控制, 如果刹车信号输入有效, FBSTA 被硬件置 1, 立即关闭 PWM 输出。如果刹车信号输入变为无效, FBSTA 自动清 0, PWM1A/B/C 波形将在 PWM 时基计数器归 0 时刻自动恢复输出。

BIT[0] **FBSTA** - 刹车输入检测标志位

0: PWM1A/B/C 通道处于正常输出状态

1: 检测到刹车有效输入信号, PWM1A/B/C 通道处于停止输出状态, 如果在锁存模式, 可软件清 0, 恢复 PWM 输出

注: 刹车信号输入 0 和 1 可同时打开, 任一信号有效都会关闭 PWM1A/B/C 模块输出。

10.10.22 PWM1 中断使能控制寄存器 (PWM1IE, 0xFF49/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1IE	PWMP1E	PWMZ1E	PWMDDC1E	PWMDUC1E	PWMDDB1E	PWMDUB1E	PWMDDA1E	PWMDUA1E



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- BIT[7] **PWMP1E** - PWM 时基周期匹配中断允许位 (只在中心对齐模式下有效)
0: 禁止 PWM 时基周期匹配中断
1: 允许 PWM 时基周期匹配中断
- BIT[6] **PWMZ1E** - PWM 时基归零中断允许位
0: 禁止 PWM 时基归零中断
1: 允许 PWM 时基归零中断
- BIT[5] **PWMD1C1E** - PWM 时基计数器减计数时与占空比寄存器 PWM1CDL/H 匹配时中断使能位 (只在中心对齐模式下有效)
0: 禁止 PWM 计数器减计数时与 PWM1CDL/H 匹配时触发中断
1: 允许 PWM 计数器减计数时与 PWM1CDL/H 匹配时触发中断
- BIT[4] **PWMD1C2E** - PWM 时基计数器增计数时与占空比寄存器 PWM1CDL/H 匹配时中断使能位
0: 禁止 PWM 计数器增计数时与 PWM1CDL/H 匹配时触发中断
1: 允许 PWM 计数器增计数时与 PWM1CDL/H 匹配时触发中断
- BIT[3] **PWMD1B1E** - PWM 时基计数器减计数时与占空比寄存器 PWM1BDL/H 匹配时中断使能位 (只在中心对齐模式下有效)
0: 禁止 PWM 计数器减计数时与 PWM1BDL/H 匹配时触发中断
1: 允许 PWM 计数器减计数时与 PWM1BDL/H 匹配时触发中断
- BIT[2] **PWMD1B2E** - PWM 时基计数器增计数时与占空比寄存器 PWM1BDL/H 匹配时中断使能位
0: 禁止 PWM 计数器增计数时与 PWM1BDL/H 匹配时触发中断
1: 允许 PWM 计数器增计数时与 PWM1BDL/H 匹配时触发中断
- BIT[1] **PWMD1A1E** - PWM 时基计数器减计数时与占空比寄存器 PWM1ADL/H 匹配时中断使能位 (只在中心对齐模式下有效)
0: 禁止 PWM 计数器减计数时与 PWM1ADL/H 匹配时触发中断
1: 允许 PWM 计数器减计数时与 PWM1ADL/H 匹配时触发中断
- BIT[0] **PWMD1A2E** - PWM 时基计数器增计数时与占空比寄存器 PWM1ADL/H 匹配时中断使能位
0: 禁止 PWM 计数器增计数时与 PWM1ADL/H 匹配时触发中断
1: 允许 PWM 计数器增计数时与 PWM1ADL/H 匹配时触发中断

10.10.23 PWM1 中断标志寄存器 (PWM1IF, 0xB8/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1IF	PWMP1IF	PWMZ1IF	PWMD1C1IF	PWMD1C2IF	PWMD1B1IF	PWMD1B2IF	PWMD1A1IF	PWMD1A2IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- BIT[7] **PWMP1IF** - PWM 时基周期匹配中断标志位 (只在中心对齐模式下有效)
0: 无 PWM 时基周期匹配中断
1: 发生 PWM 时基周期匹配中断
- BIT[6] **PWMZ1IF** - PWM 时基归零中断标志位
0: 无 PWM 时基归零中断
1: 发生 PWM 时基归零中断
- BIT[5] **PWMD1C1IF** - PWM 时基计数器减计数时与占空比寄存器 PWM1CDL/H 匹配时中断标志位 (只在中心对齐模式下有效)
0: 未发生中断
1: 发生中断
- BIT[4] **PWMD1C2IF** - PWM 时基计数器增计数时与占空比寄存器 PWM1CDL/H 匹配时中断标志位
0: 未发生中断
1: 发生中断
- BIT[3] **PWMD1B1IF** - PWM 时基计数器减计数时与占空比寄存器 PWM1BDL/H 匹配时中断标志位 (只在中心对齐模式下有效)
0: 未发生中断
1: 发生中断
- BIT[2] **PWMD1B2IF** - PWM 时基计数器增计数时与占空比寄存器 PWM1BDL/H 匹配时中断标志位
0: 未发生中断
1: 发生中断
- BIT[1] **PWMD1A1IF** - PWM 时基计数器减计数时与占空比寄存器 PWM1ADL/H 匹配时中断标志位 (只在中心对齐模式下有效)
0: 未发生中断



1: 发生中断
 BIT[0] PWMUDAIF - PWM 时基计数器增计数时与占空比寄存器 PWM1ADL/H 匹配时中断标志位
 0: 未发生中断
 1: 发生中断

注 1: 即使 PWM 中断允许位为 0, 中断标志位也会置 1, 但中断不会响应。

10.10.24 PWM1 死区控制寄存器 0 低位 (PWM1DT0L, 0xFF4D/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DT0L	PWMDT07	PWMDT06	PWMDT05	PWMDT04	PWMDT03	PWMDT02	PWMDT01	PWMDT00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMDT0[7:0] - PWM1 死区控制寄存器 0 低位

10.10.25 PWM1 死区控制寄存器 1 低位 (PWM1DT1L, 0xFF4F/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DT1L	PWMDT17	PWMDT16	PWMDT15	PWMDT14	PWMDT13	PWMDT12	PWMDT11	PWMDT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMDT1[7:0] - PWM1 死区控制寄存器 1 低位

10.10.26 PWM1 寄存器修改重载解锁寄存器 (PWM1RLEN, 0xD9/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1RLEN	PWMRL7	PWMRL6	PWMRL5	PWMRL4	PWMRL3	PWMRL2	PWMRL1	PWMRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] PWMRL[7:0] - 寄存器修改和重载控制寄存器

0x55: 允许软件对模块寄存器的修改

0xAA: 允许模块带缓冲的寄存器的重载

注 1: PWM 模块的寄存器, 除了中断标志寄存器 PWM1IF 和手动控制寄存器 PWMMANCR1 外, 其它寄存器的修改有限制条件, 只有当 PWM1RLEN=0x55 时才允许软件修改, 否则修改无效。

注 2: PWM 模块中有一些寄存器带有缓冲寄存器, 包括占空比寄存器、周期寄存器。这些寄存器在软件修改后不会立即生效, 只有在归零或周期溢出时硬件重载, 才会真正生效。是否允许这些寄存器重载同样受控制, 只有当 PWM1RLEN=0xAA 时才允许重载。

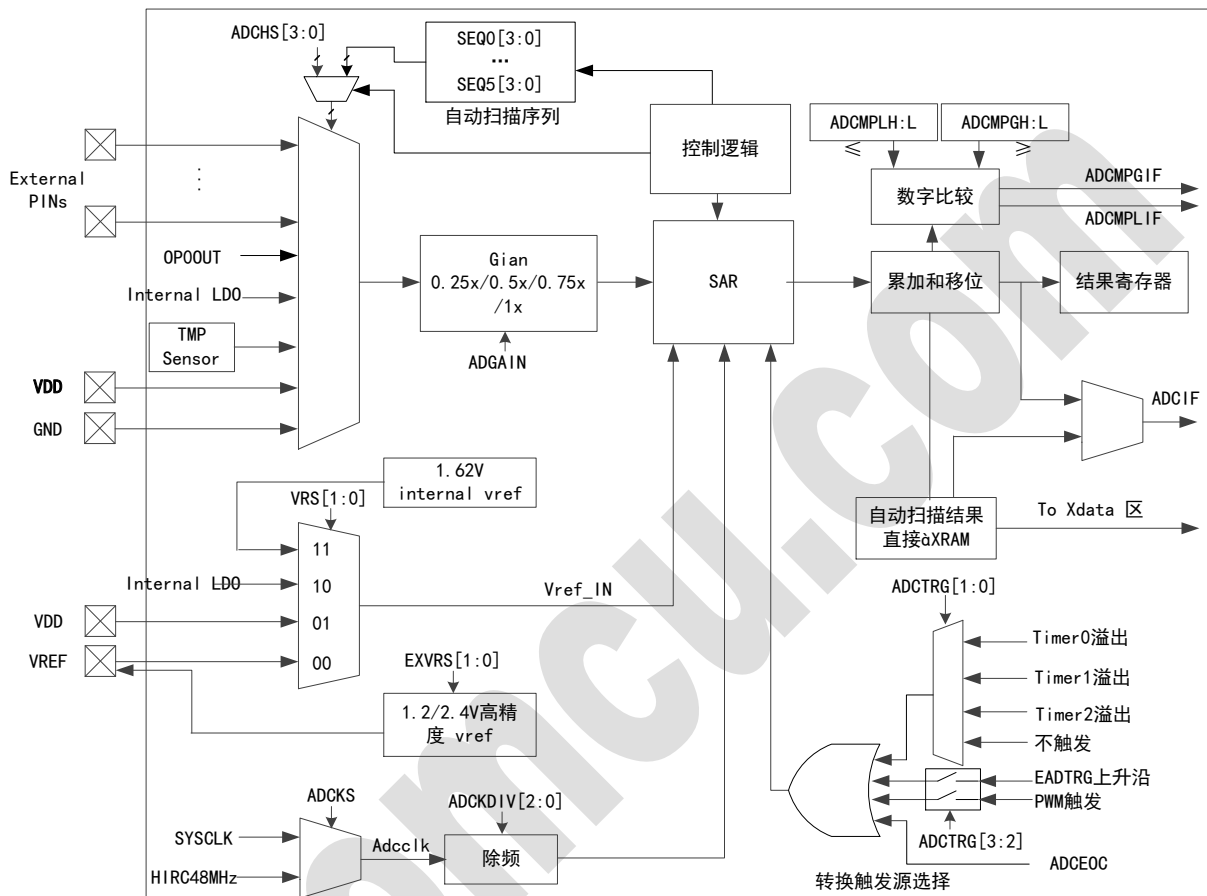


11 12-bit ADC

11.1 概述

ADC 模块为 12-bit 逐次逼近型 (SAR) ADC，具有 12 位、10 位、8 位和 6 位模式可选，集成采样保持和可编程数字比较器。通过配置模拟多路复用器可以测量不同通道信号。电压参考可选择内部和外部多个参考源。

ADC 模块框图



11.2 特性

- ✧ 多达 11 个外部输入通道
- ✧ 单端 12 位、10 位、8 位和 6 位模式
- ✧ 支持转换速率高达 1 Msps (12 位模式下)
- ✧ 通道转换序列直接与 xdata 区关联
- ✧ 异步硬件转换触发功能，可选择软件触发、外部 I/O、内部定时器
- ✧ 内建输出数据数字比较器，支持上下限比较，阈值独立配置
- ✧ 支持输出数据累加
- ✧ 支持转换完成中断和数字比较中断
- ✧ 灵活的输出数据格式
- ✧ 参考源包括：内部快速启动 1.62 V 参考、2.4/1.2 V 片内高精度参考、外部参考
- ✧ 集成出厂校准的温度传感器

11.3 功能描述

11.3.1 输入通道选择

ADC 内建一个模拟多路复用器，可以选择外部引脚通道、片上温度传感器、内部 LDO、VDD 电源或 GND。配置 ADCMUX 寄存器选择 ADC 输入通道。

注 1: 任何选择作为 ADC 输入的端口引脚都应在其关联的端口配置寄存器中配置为模拟输入。



注 2: 当自动扫描模式关闭时 ($ASEN=0$)，通道选择由 $ADCHS[3:0]$ 控制， $ADCHS[3:0]$ 可读/写；当自动扫描模式开启时 ($ASEN=1$)，通道选择由 $SEQn[3:0]$ ($n=0\sim5$) 控制，此时 $ADCHS[3:0]$ 为只读，实时表示当前自动扫描的通道号。

多路通道选择

$ADCHS[3:0]/SEQn[3:0], n=0\sim5$	模拟通道名	管脚名
0000	AIN0	P3.3
0001	AIN1	P3.2
0010	AIN2	P3.1
0011	AIN3	P2.7
0100	AIN4	P2.6
0101	AIN5	P2.5
0110	AIN6	P2.4
0111	AIN7	P2.3
1000	AIN8	P2.2
1001	AIN9	P3.0
1010	AIN10	P3.4
1011	OP0OUT	内部 OP0 输出
1100	Temp Sensor	内部温度传感器通道
1101	LDO_OUT	内部 LDO 输出
1110	VDD	VDD 电源
1111	GND	GND

11.3.2 增益配置

ADC 增益可设置为 $1\times$ 、 $0.75\times$ 、 $0.5\times$ 、 $0.25\times$ 。在 $1\times$ 模式下，ADC 的输入全范围电压由 $VREF$ 决定。在其他模式下，当输入电压等于 $[VREF/\text{增益}]$ 时，达到全范围电压。

例如，在 $0.5\times$ 模式下，全范围输入电压为 $VREF / 0.5 = 2\times VREF$ 。当选择较小的 $VREF$ 电压时，较低的增益设置可以用于获得较高的输入电压范围，或者用于测量 $VREF$ 和电源电压之间的输入电压。

ADC 的增益设置由 $ADCCR1$ 的 $ADGAIN$ 控制。

注：即使采用较低的增益设置，大于电源电压也不能由 ADC 直接测量。

11.3.3 电压参考选择

内建电压参考多路复用器，可配置不同的内部和外部参考源。

寄存器 $ADCCRO$ 的 VRS 位选择不同参考源，寄存器 $ADCCR4$ 的 $GNDS$ 位选择不同的参考地连接。

内部电压参考

内建独立稳定的高速 $1.62V$ 参考。不会引线到 $VREF$ pin，无需外部解耦。当被选择时，根据需要由 ADC 自动启用/禁用。参考电压值以电气规格章节为准。

VDD 或 LDO 电压参考

对于电源电压不变的应用，使用电源 VDD 作为电压参考可以为 ADC 提供附加的动态输入范围，但必须降低电源的噪声抑制。此外，内部 LDO 电压可以作为参考。

VDD 和 LDO 参考源都不会被引线到 $VREF$ pin，不需要额外的外部解耦。

外部电压参考

可以选择外部参考源，通过 $VREF$ 引脚接入。

旁路电容器应按照外接参考源制造商的建议添加。如果制造商不提供建议，建议使用一个 $4.7\mu F$ 的电容器和一个 $0.1\mu F$ 的电容器并联使用。

注： $VREF$ 引脚是多功能 $GPIO$ 引脚。当使用外部电压参考时， $VREF$ 应该配置为模拟输入。

高精度电压参考

内建 1 个稳定的高精度电压参考源，经过生产校准， $1.2V/2.4V$ 两档可选。

输出引线至 $VREF$ 引脚，需要外部旁路电容 ($\geq 0.1\mu F$)。ADC 选择此参考源，需要启用和建立。

11.3.4 时钟配置

ADC 时钟 ($ADCCLK$) 可以通过配置 $ADCKS$ 选择系统时钟 ($SYSCLK$) 或 $HIRC48MHz$ 时钟。默认选择是系统时钟 ($SYSCLK$)。设置 $ADDIV$ 选择不同的分频系数。分频后的时钟用于 ADC 转换，最高时钟不能高于 $18MHz$ 。

11.3.5 转换时序

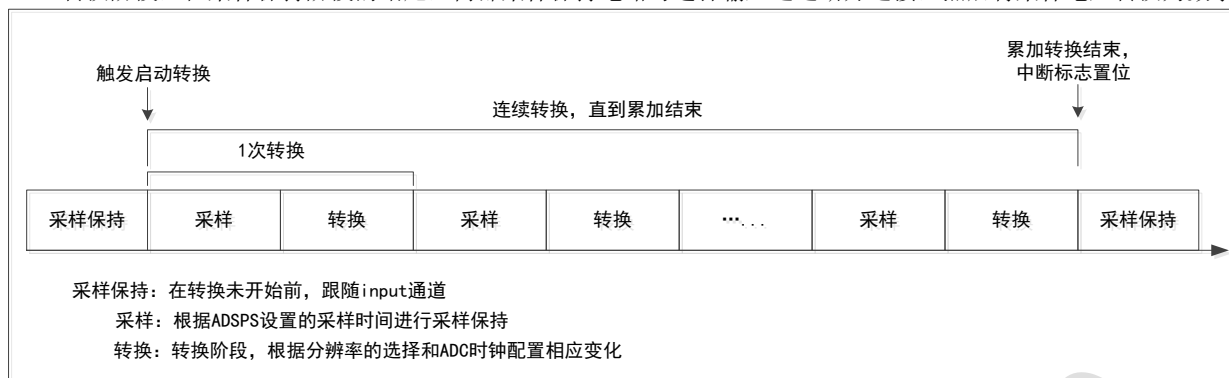
一次 ADC 转换包括：模块上电，采样保持和转换。

模块上电阶段，ADC 及内部参考电路建立时间段，为采样保持阶段做准备；

采样保持阶段，ADC 多路复用器连接到所选输入并采样的时间段，即转换尚未进行时或用户配置的特定采样时间。当连续多个转换时，需要对 $ADSPS$ 位进行配置，调整合适的采样保持时间。



转换阶段，在采样保持阶段的结尾，内部采样保持电路与选择输入通道断开连接，然后将采样电压转换为数字值。



输入采样保持

每次 ADC 转换前，都需要有足够的采样时间，保证内部的采样电容电压稳定，进而保证转换结果准确。举例，如果 ADC 输入具有较大的串联阻抗，则在采样阶段，采样电容建立待测电压值，需要更长的时间。

采样时间计算：

$$t = \ln\left(\frac{2^n}{\text{预期精度}}\right) \times R_{\text{total}} \times C_{\text{sample}}$$

t，采样建立时间，单位 s

n，为分辨率

预期精度，按照 LSB 为单位，例如，1/4LSB，预期精度=0.25

R_{total}，位 mux 及外部电阻总和，mux 电阻参考电气章节。

C_{sample}，ADC 采样电容，参考电气参数章节。

根据采样时间计算，通过配置 ADSPS 选择合适的挡位。

上电时序

ADC 上电内部电路及参考电路，需要一定的时间才能稳定建立。在 ADEN 使能后至触发启动转换前，需要用户延迟预热时间，详细参数参见电气章节。

转换分辨率和时序

转换时间计算：

$$T_{\text{cnv}} = [ADACC \times (ADSPS + ADBIT + 2) \times T_{\text{adclk}}]$$

注：如果外部 EADTRG 启动，需要额外的 1 个 sysclk 同步。

11.3.6 触发转换

配置 ADTRG 位可以选择多种方式触发启动 ADC 转换。

软件触发：ADEOC=1 触发

硬件触发：内部硬件功能模块自动触发，timer 溢出、PWM 触发信号等。

外部触发：EADTRG 引脚上升沿，需要对应引脚不被 DFM（IO 数字全映射）模块映射。

11.3.7 自动扫描模式

ADC 支持灵活的自动扫描 (autoscan) 模式。配置寄存器 ADASCRO/1/2 和 ADSEQ0/1/2 控制 Autoscan 模式相关功能。自动扫描模式支持从最多 6 个可选的 ADC 通道收集多次转换结果，转换的序列在 ADSEQ0/1/2 中配置。按转换序列的顺序被写入到指定的 XDATA 空间区域。最多收集 32 次转换结果 (32 个 words)。启动自动扫描模式，扫描器按照请求的收集次数工作；扫描序列结束，自动扫描硬件存储当前寄存器状态，并产生中断标志，也可以选择继续扫描循环。

触发

在自动扫描模式下，触发源选择同正常模式由 ADTRG 位配置。ADASCRO 中的 ASTRG 位控制一次扫描需要多次触发还是单次。当 ASTRG=0，每次转换（或累加转换）需要一次触发；当 ASTRG=1，一次触发执行一次扫描。

举例，如果选择定时器溢出为 ADC 触发源，当 ASTRG=0，并且 autoscan 硬件配置为累计 10 组 4 次转换，定时器需要溢出 10 次来为每个转换生成一个触发事件。当 ASTRG=1，Timer 溢出事件触发第一个转换，然后在每次转换完成时，自动扫描硬件将自动触发其余的转换。

注：在自动扫描模式下，为防止触发源在 ASEN 设置为 1 触发 ADC。最建议将 ASTRG 配置为软件触发器，直到 ASEN 设置为 1，然后选择所需的触发器源。

通道配置

扫描器硬件可以配置最多 6 个可选的 ADC 通道按扫描序列 SEQ0~5 配置的通道顺序收集数据。一旦启动自动扫描模式，ADCHS 位配置无效，SEQ0~5 配置了依次扫描的通道 (SEQ0 第一个通道)，ASCNUM 位定义转换的通道数数量 (1、2、3、4、5 或 6 可选)。

注：当自动扫描模式关闭时 (ASEN=0)，通道选择由 ADCHS[3:0] 控制，ADCHS[3:0] 可读/写；当自动扫描模式开启时 (ASEN=1)，通道选择由 SEQn[3:0] (n=0~5) 控制，此时 ADCHS[3:0] 为只读，实时表示当前自动扫描的通道号。

若通道数量设置为小于 6，那么剩余的序列配置通道不参与扫描（举例设置为 3，SEQ3/4/5 不参与扫描）；



举例，如果 SEGO 被配置为 AIN4，SEG1 被配置为 AIN3，SEG2 被配置为 AIN6，ASCNUM 配置为 3 个通道，并且 ASCNT 位配置为请求 9 次转换，autoscan 硬件将收集从 AIN4 开始转换，然后是 AIN3 转换，继而是（AIN6）转换，然后再 AIN4 重复，以此类推，直到 9 次转换完成（每个通道 3 次）。

ADACC 在自动扫描模式下是有效的，每组累加转换作为自动扫描收集的一次转换。如果 ADACC 配置为 8 次累加，自动扫描 ASCNT 位配置为请求 9 次转换，那么总共执行 $9 \times 8 = 72$ 次 AD 转换。当扫描配置多通道时，ADC 模块会在切换下个通道前完成此通道累加操作。

输出数据配置

自动扫描收集的数据由硬件直接写入 XDATA 空间，ADASCRI1/2 寄存器的 ADDR[10:0] 定义 XDATA 的起始地址，ADDR[10:0] 直接对应 XRAM 开始地址的 10:0 位，其中 bit0 无效，即 XRAM 的存储地址从偶数字节地址开始（16bit 对齐），地址 2 字节自增；每次转换存储的 2 个 bytes 固定为大端模式（高有效字节存储在偶数地址，低有效字节存储在奇数地址）。ADC 每次转换结果需要 2 个 bytes 的 XDATA 空间。

注：开发环境不会自动识别自动扫描写入 XDATA 的位置，用户在使用 autoscan 功能时，必须预留用于自动扫描的 XDATA 区域，以避免与其他变量冲突。

自动扫描操作

ADC 配置完成后，设置 ASEN 位为 1，ADC 进入扫描模式。请注意，设置 ASEN 位后，转换不会立即开始。ASEN 只是将 ADC 置于自动扫描模式，等待第一个触发后开始转换。

设置 ASEN 后，硬件会将 ADASCRO/1/2 寄存器的 ADDR 位、ASCNT 位、ASCNUM 位的内容备份到缓存寄存器中供扫描器使用，并将扫描序列的 SEGO 选择的通道加载至通道选择复用器。这样，用户可以立即为后续扫描设置参数。

扫描模式，一个通道转换结束后，SEGx 序列配置的通道选择依次加载到通道选择复用器，直至扫描结束。

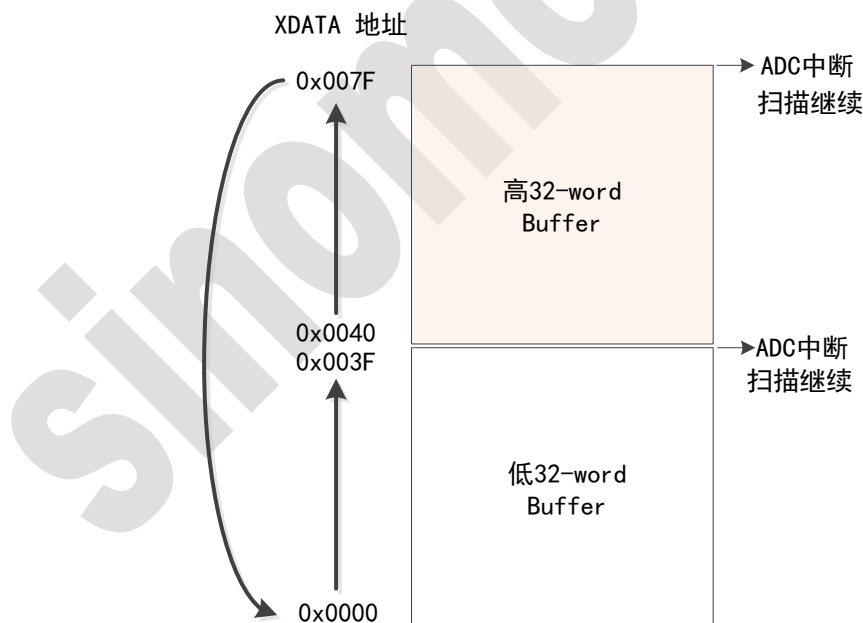
如果只需要一次扫描，使能 ASEN 后，可以立即清零。正如设置 ASEN 不会立即开始扫描一样，清除 ASEN 也不会立即使转换器脱离自动扫描模式。只有当扫描序列操作完成且 ASEN 为 0 时，自动扫描模式才会停止。要终止正在进行的扫描，必须完全关闭 ADC（ADEN 清零）。

当 ADC 首次启动自动扫描模式时，等待所选择的转换触发信号发生，自动扫描将按照配置选项进行，直到完成 ASCNT 设置收集数量。在扫描操作结束时，ADCIF 位设置为 1，并检查 ASEN 的状态。如果 ASEN 为 0，自动扫描模式终止，转换器将返回正常模式；如果 ASEN 为 1，一个新的扫描将立即开始，扫描相关设置被加载到扫描器的缓存寄存器中，并等待下一次触发发生。

自动扫描应用举例

示例 1：循环 buffer

此示例展示了使用自动扫描模式为单个 ADC 通道实现 64 words 乒乓缓冲（XDATA 区）所需的步骤。缓冲区将由 XDATA 中的两个 32-word（64 字节）区域组成，从 0x0000 和 0x0040 开始，固件负责以适当的间隔更改扫描仪硬件，以保持连续的数据流进入内存。本例假设 ADC 将从硬件源（如定时器）以多重触发模式触发。



初始化步骤：

- 1、配置 ADC 不累加：ADACC = 0
- 2、配置 ADC 通道：ASCNUM[1:0]=0，选择单个通道，SEQ0[3:0]配置预期的通道；
- 3、配置第一个 32-word 缓冲区地址：ADDR[10:0] = 0x0000；
- 4、配置收集样本的数量 32：ASCNT[4:0] = 31；
- 5、初始化 autoscan 模式：ASEN = 1；
- 6、配置第二个 32-word 缓冲区地址：ADDR[10:0] = 0x0040；
- 7、开始 ADC 转换：启动转换触发源，或者如果触发源已经运行，选择对应的触发源。



中断服务程序：

- 1、清除中断标志
- 2、切换缓存区地址：若 ADDR[10:0] = 0x0040，写 ADDR[10:0] = 0x0000；反之亦然；
- 3、处理最近缓冲区中的数据，或者可以选择向主线程发出信号，数据准备好可以处理。

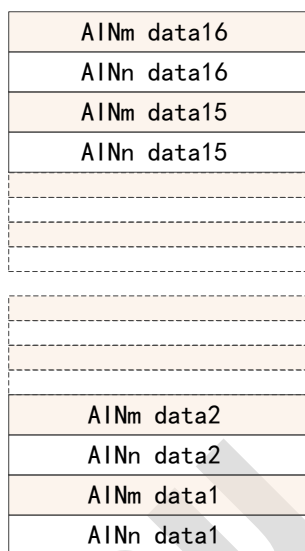
示例 2：单次扫描 2 个 通道

此示例展示了使用自动扫描模式将两个可选的通道扫描到 32 words 缓冲区(每个通道 16 次转换)所需的步骤。在本例中，单个软件触发器用于启动整个扫描序列。

XDATA 地址

0x003F

0x0000



→ ADC中断
扫描停止

初始化步骤：

- 1、配置 ADC 不累加：ADACC = 0；
- 2、配置 ADC 触发源：ADCEOC 软触发，不受 ADCTRG[3:0]控制影响，独立触发，若选择其他触发源，配置 ADCTRG[3:0]相关位；
- 3、配置 ADC 通道：ASCNUM[1:0]=1，选择两个通道，SEQ0[3:0]配置预期的通道 AINn，SEQ1[3:0]配置预期的通道 AINm；
- 4、配置缓冲区开始地址：ADDR[10:0] = 0x0000；
- 5、配置收集样本的数量 32：ASCNT[4:0] = 31；
- 6、初始化 autoscan 模式：ASEN = 1；
- 7、配置 ASEN=0，这将指示在此次扫描完成后停止。
- 8、开始 ADC 转换：启动转换触发源，或者如果触发源已经运行，选择对应的触发源。
- 9、开始 ADC 转换：写 ADEOC = 1；

中断服务程序：

- 4、清除中断标志
- 5、切换缓存区地址：若 ADDR[10:0] = 0x0040，写 ADDR[10:0] = 0x0000；反之亦然；
- 6、处理缓冲区中的数据，或者可以选择向主线程发出信号，数据准备好可以处理。

11.3.8 输出数据格式及累加

ADC 转换后的结果存储 ADRH:ADRL 寄存器中，包含高字节和低字节。数据支持多次累加，最终输出可选右移位数，组成一个“累加+平均”的函数。

ADACC 位控制累加次数，1/4/8/16/32 次可选。

ADRST 位控制右移位数，无移位/右移 1 位/2 位/3 位可选。

ADBIT 位控制 ADC 分辨率，6BIT/8BIT/10BIT/12BIT 可选。

当累加计数大于 1 时，输出结果寄存器为执行转换的累加结果，并在该累加序列中的最后一次转换完成后更新。累加结果以无符号整数格式表示。

ADPAC 位，使能此位，禁用累加器清除功能，ADC 总是会将最新的结果加到累加器的当前值上，硬件也不会将累加器复位到 0。使能 ADPAC 位，请设置 ADACC 位为 000b（不累加）。

ADACC=000b，ADRST=00b，增益为 1

输入电压	6-bit ADRH:L	8-bit ADRH:L	10-bit ADRH:L	12-bit ADRH:L
------	-----------------	-----------------	------------------	------------------



VREF - 1 LSBn	0x003F	0x00FF	0x03FF	0x0FFF
VREF / 2	0x0020	0x0080	0x0200	0x0800
VREF / 4	0x0010	0x0040	0x0100	0x0400
0	0x0000	0x0000	0x0000	0x0000

12bit 分辨率, ADRST=00b, 增益为 1

输入电压	累加次数 = 4	累加次数= 8	累加次数= 16
VREF - 1 LSB12	0x3FFC	0x7FF8	0xFF0
VREF / 2	0x2000	0x4000	0x8000
(VREF / 2) - 1 LSB12	0x1FFC	0x3FF8	0x7FF0
0	0x0000	0x0000	0x0000

12bit 分辨率, ADACC=010b (累加 8 次), 增益为 1

输入电压	ADRST = 0 (无移位)	ADRST = 1 (右移 1 位)	ADRST = 3 (右移 3 位)
VREF - 1 LSB12	0x7FF8	0x3FFC	0x0FFF
VREF / 2	0x4000	0x2000	0x0800
(VREF / 2) - 1 LSB12	0x3FF8	0x1FFC	0x07FF
0	0x0000	0x0000	0x0000

11.3.9 数字比较器

ADC 模块支持数字比较器功能, ADCMPGH:ADCMPGL 寄存器设置上限比较值, ADCMPLH:ADCMPLL 寄存器设置下限比较值, CMPEN 位控制数字比较器使能。

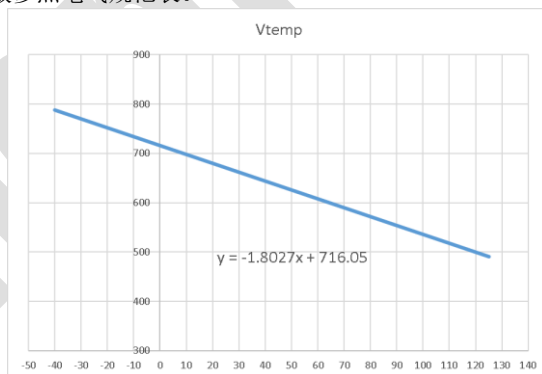
设置 CMPEN=1, 使能数字比较功能, 如果 ADC 转换结果大于等于(ADCMPGH:ADCMPGL)值, 则置位 ADCMPGIF; 如果 ADC 转换结果小于等于(ADCMPLH:ADCMPLL)值, 则置位 ADCMPLIF。

注: CMPEN 开启后, 启动一次数字比较, 此时若 ADC 未发生转换, 将当前结果寄存器与比较限值比较; 每次 ADC 转换进行一次比较, 比较发生在上次的结果数据。

11.3.10 温度传感器

温度传感器可以用来测量器件的结温(TJ)。

芯片内建温度传感器, 内部连接 ADC 多路复用器的输入。设置 ADCHS 可选择温度传感器通道接入 ADC, TMPEN 控制启动/关闭温度传感器, 当禁用时, 温度传感器默认为高阻抗状态, 在传感器上执行的任何 ADC 测量将导致无意义的数。温度传感器的斜率和偏置参数参照电气规范表。



读取温度流程

1、选择 ADC1_IN12 输入通道

选择适当的采样时间, 参考芯片数据手册 TS_temp 参数。

ADC_CCR 寄存器中设置 TSEN 位用来唤醒从断电模式下的温度传感器, 并等待其稳定时间(tSTART)。

2、置位 ADC_CR 寄存器中的 ADSTART 位 (也可用外部触发) 来启动 ADC 转换。

3、从 ADC_DR 寄存器中读取转换数据 VADC_Data

计算温度: $TADC_Data(^{\circ}C) = (VDDA / FULL_SCALE \times ADC_Data - b) / k$

4、从 sysrom 出厂区读出 VADC_Temp1、Sens_Temp1 为出厂校准值; 详细参考数据手册描述

计算校准点温度(芯片): $TADC_Temp1(^{\circ}C) = (VADC_Temp1 - b) / k$

读取校准点温度(环境): TSens_Temp1(^{\circ}C)

计算温度 offset: $Toffset = TSens_Temp1 - TADC_Temp1$

5、修正实际温度:

实际温度 $T(^{\circ}C) = TADC_DATA + Toffset$



其中,

--VDDA_3v3 为出厂校准时模拟电源, 电压= 3.3 v;

--VDDA 为实际应用中模拟电源, 由实际应用确定

--FULL_SCALE = 4095

--ADC_Data, 实际 ADC 转换值

--VADC_Data = VDDA / FULL_SCALE x ADC_DATA (选择 VDDA 当参考)

--VADC_Data = VREF/ FULL_SCALE x ADC_DATA (选择 VREF 当参考)

--VADC_Temp1 为出厂测试的温度传感器电压输出, 单位 mV, 参见数据手册描述;

--k 为温度传感器电压平均斜率 Avg_Slope = 2.82v/°C, 以实际数据手册为准;

--b 为温度传感器电压 vs 温度一次曲线常数项 (截距), b=0.7161 v

若实际应用 VDDA 为 3.3v, 公式可以简化为:

实际温度 T (°C) = ((VADC_Data-VADC_Temp1)/Avg_slope) +TSens_Temp1

注: 传感器从断电模式下唤醒时到能正确输出 VSENSE 的值要有一个启动时间, 另 ADC 从上电后启动也有一个启动时间, 若要最小化这个延时, 则需要在同一时间置位 ADEN 和 TSEN 位。

11.3.11 零点偏移修调

ADC 模块支持零偏修调功能, 包含硬件自动修调和用户软件修调 2 种方式。

硬件自动修调:

设置 ADASCRO→ADJSTR 为 1, 启动 ADC 校准流程, ADC 切换为校准模式, 开始自校准; 校准完成, 此位清零。通过 CALOS 寄存器 (SFR1, 只读) 可以读取校准后的修调值。

软件修调:

用户也可以软件操作 ADCALOS 寄存器进行修调, AD 模块零点偏移修调步骤如下:

- (1) 将 ADC 的输入通道选择为内部接地 (AN15), 设置 CALOSEN=1 打开修调功能; 设置一个基本的 ADC 时钟 (建议 ADC 工作频率在 1.5MHz 为宜)、采样时间 (建议 ADC 采样时间在 8 个 Clk 为宜) 等;
- (2) 设置 OS6 =1, OS[5:0]=00 0000B, 进行一次 ADC 转换;
若 ADC 转换结果不为 0, 则跳至步骤 (4), 进行负向修调;
若 ADC 转换结果为 0 (实际值≤0), 则执行步骤 (3);
- (3) 将 OS[5:0]的值, 加 1 后, 再次进行 ADC 转换;
若 ADC 转换结果不为 0, 则跳至步骤 (6);
若 ADC 转换结果为 0, 则循环执行步骤 (3), 直到 ADC 转换结果不为 0 或 OS[5:0]递增到最大值 11 1111B, 跳至步骤 (6);
- (4) 设置 OS6 =0, OSADJT[5:0]=11 1111B, 进行一次 ADC 转换;
若 ADC 转换结果为 0, 则跳至步骤 (6);
若 ADC 转换结果不为 0, 则执行步骤 (5);
- (5) 将 OS[5:0]的值, 减 1 后, 再次进行 ADC 转换;
若 ADC 转换结果为 0, 则跳至步骤 (6);
若 ADC 转换结果不为 0, 则循环执行步骤 (5), 直到 ADC 转换结果为 0 或 OSADJT[5:0]递减到 00 0000B, 跳至步骤 (6);
- (6) OS6 和 OS[5:0]的值, 即为零点偏移最佳修调结果, 修调流程结束, 后续 ADC 工作时可直接使用该值, 不需要再次进行修调 (除非该值丢失, 如断电等情况)。

注: 修调过程工作条件变化以及 ADC 有效位数, 可能影响修调的最佳效果, 用户可以通过增加单次步骤中的 ADC 采样次数或多次修调择优方式处理。

11.4 ADC 相关寄存器

ADC 寄存器汇总表

地址 (SFRn /H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR1/A1	CALOS	零点偏移校准结果寄存器	-	OS6	OS5	OS4	OS3	OS2	OS1	OS0	xxxx xxxx
XSFR/FF44	ADCALOS	ADC 零偏校准寄存器	CALOSEN	OS6	OS5	OS4	OS3	OS2	OS1	OS0	0xxx xxxx
XSFR/FF47	ADSEQR2	ADC 扫描序列寄存器 2	SEQ53	SEQ52	SEQ51	SEQ50	SEQ43	SEQ42	SEQ41	SEQ40	0000 0000
XSFR/FF46	ADSEQR1	ADC 扫描序列寄存器 1	SEQ33	SEQ32	SEQ31	SEQ30	SEQ23	SEQ22	SEQ21	SEQ20	0000 0000
XSFR/	ADSEQR0	ADC 扫描序列寄	SEQ13	SEQ12	SEQ11	SEQ10	SEQ03	SEQ02	SEQ01	SEQ00	0000



FF45		寄存器 0									0000
SFR0/ FC	ADASCR2	ADC 自动扫描控制寄存器 2	ASCNT4	ASCNT3	ASCNT2	ASCNT1	ASCNT0	ADDR10	ADDR9	ADDR8	0000 0000
SFR0/ FB	ADASCR1	ADC 自动扫描控制寄存器 1	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	-	0000 000-
SFR0/ FA	ADASCRO	ADC 自动扫描控制寄存器 0	ADJSTR	=	ASCNUM2	ASCNUM1	ASCNUM0	ASSTA	ASTRG	ASEN	---0 0000
XSFR/ FF42	ADCMPH	ADC 数字比较下限高位寄存器	CMPL15	CMPL14	CMPL13	CMPL12	CMPL11	CMPL10	CMPL9	CMPL8	0000 0000
XSFR/ FF43	ADCMPLL	ADC 数字比较下限低位寄存器	CMPL7	CMPL6	CMPL5	CMPL4	CMPL3	CMPL2	CMPL1	CMPL0	0000 0000
XSFR/ FF40	ADCMPLH	ADC 数字比较上限高位寄存器	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP9	CMP8	0000 0000
XSFR/ FF41	ADCMPLL	ADC 数字比较上限低位寄存器	CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0	0000 0000
SFR0/ FF	ADRH	ADC 转换结果高位寄存器	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8	xxxx xxxx
SFR0/ FE	ADRL	ADC 转换结果低位寄存器	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADRO	xxxx xxxx
SFR0/ F5	ADCCR4	ADC 控制寄存器 4	EXVRS1	EXVRS0	EXVREF_EN	=	ADCTRG3	ADCTRG2	ADCTRG1	ADCTRG0	000- 0000
SFR0/ F4	ADCCR3	ADC 控制寄存器 3	ADPAC	ADRST1	ADRST0	ADACC2	ADACC1	ADACC0	ADBIT1	ADBIT0	0000 0000
SFR0/ FD	ADCCR2	ADC 控制寄存器 2	ADCKS	ADCKDIV2	ADCKDIV1	ADCKDIV0	-	ADSPS2	ADSPS1	ADSPS0	0000 -111
SFR0/ F3	ADCCR1	ADC 控制寄存器 1	TMPEN	ADGAIN1	ADGAIN0	-	ADCHS3	ADCHS2	ADCHS1	ADCHS0	0000 1111
SFR0/ F2	ADCCRO	ADC 控制寄存器 0	CMPEN	ADCMPLI F	ADCMPI F	ADCIF	VRS1	VRS0	ADEOC	ADEN	0000 1100

注: CALOS 为 ADC 硬件校准结果寄存器, 只读 for debug。

11.4.1 ADC 控制寄存器 0 (ADCCRO, 0xF2/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCRO	CMPEN	ADCMPLI F	ADCMPI F	ADCIF	VRS1	VRS0	ADEOC	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	0	0

BIT[7] CMPEN - ADC 数字比较器使能

0: 数字比较器功能关闭

1: 数字比较器使能

BIT[6] ADCMPLI - ADC 下限比较中断标志位, 软件清 0, 硬件置 1

0: 未发生下限比较中断;

1: 发生下限比较中断, ADC 转换结果 ADRH:L 小于等于 ADCMPH:L 值。

BIT[5] ADCMPI - ADC 上限比较中断标志位, 软件清 0, 硬件置 1

0: 未发生上限比较中断;

1: 发生上限比较中断, ADC 转换结果 ADRH:L 大于等于 ADCMPH:L 值。

BIT[4] ADCIF - ADC 中断标志位

0: 未发生 ADC 中断, 由软件清 0;

1: 发生 ADC 中断, 硬件置 1。

BIT[3:2] VRS[1:0] - ADC 参考电压选择

VRS0	ADC 参考电压
00	VREF_PIN
01	VDD
10	内部 LDO 1.5V
11	内部参考 1.62V

注: 若选择 VREF_PIN, 需设置 ADCCR4 寄存器的 EXVREF_EN 位配置 VREF 端口有效, 再设置 EXVRS[1:0] 位选择内部高精度 2.4V/1.2V 参考或 VREF pin 外部输入参考。



BIT[1] ADEOC - ADC 启动位及转换结束标志位

0: AD 转换已结束, 同时作为 AD 转换完成的查询标志。

1: AD 启动转换结束后硬件自动清 0; 其他情况表示 AD 正在转换中

注: 该位写入 1 时, 启动一次 ADC 转换; 当未转换完成时, 再写入 1, 则上次转换中止, 开始一次新的转换; 0 可以被写入, 但其对 AD 模块无效, 即不会影响 AD 转换结果。

BIT[0] ADEN - ADC 功能使能位

0: 关闭 ADC 模块;

1: 使能 ADC 模块。

11.4.2 ADC 控制寄存器 1 (ADCCR1, 0xF3/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCR1	TMPEN	ADGAIN1	ADGAIN0	–	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	1	1

BIT[7] TMPEN - 温度传感器使能

0: 关闭

1: 使能

BIT[6:5] ADGAIN[1:0] - ADC 增益选择

00: PGA 增益 1x

01: PGA 增益 0.75x

10: PGA 增益 0.5x

11: PGA 增益 0.25x

BIT[3:0] ADCHS[3:0] - ADC 模拟通道选择

ADCHS[3:0]	模拟通道名
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
0100	AIN4
0101	AIN5
0110	AIN6
0111	AIN7
1000	AIN8
1001	AIN9
1010	AIN10
1011	OP0OUT
1100	Temp Sensor
1101	LDO_OUT
1110	保留
1111	GND

11.4.3 ADC 控制寄存器 2 (ADCCR2, 0xFD/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCR2	ADCKS	ADCKDIV2	ADCKDIV1	ADCKDIV0	–	ADSPS2	ADSPS1	ADSPS0
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W
复位值	0	0	0	0	–	1	1	1

BIT[7] ADCKS - ADC 时钟源选择

0: 选择系统时钟 F_{sys};

1: 选择内部高频 HIRC48MHz

BIT[6:4] ADCKDIV[2:0] - AD 转换时钟选择

ADCKDIV[2:0]	AD 转换时钟 F _{ADC}
000	F _{adcc1k}
001	F _{adcc1k} /2
010	F _{adcc1k} /4
011	F _{adcc1k} /8
100	F _{adcc1k} /16



	101	Fadccclk /32
	110	Fadccclk /64
	111	Fadccclk /3

BIT[2:0] ADSPS[2:0] - ADC 采样时间选择

ADSPS[2:0]	SAMPLE 时间
000	3 个 adccclk
001	4
010	8
011	16
100	32
101	64
110	128
111	256

注: ADSPS[2:0]表格中为采样时钟数, 实际有效的采样时间为上述时钟减去 0.5clk 周期。

11.4.4 ADC 控制寄存器 3 (ADCCR3, 0xF4/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCR3	ADPAC	ADRST1	ADRST0	ADACC2	ADACC1	ADACCO	ADBIT1	ADBIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] ADPAC - ADC 转换数据结果保持控制

0: 每次启动 ADC 转换, 新数据覆盖上次数据

1: 每次启动 ADC 转换, 新数据累加到上次数据, 如果 ADACC=0, 新数据为一次转换结果值; 若 ADACC≠0, 新数据为多次累加结果值

BIT[6:5] ADRST[1:0] - ADC 转换数据结果右移控制

00: ADC 转换结果右对齐, 无移位

01: ADC 转换结果右对齐, 右移 1 位

10: ADC 转换结果右对齐, 右移 2 位

11: ADC 转换结果右对齐, 右移 3 位

注: 此位控制[ADRH:ADRL]16 位结果寄存器, 配合累加功能 (ADACC[2:0]), 实现转换结果有效位扩展及减少。

BIT[4:2] ADACC[2:0] - ADC 转换数据结果累加控制

000: ADC 一次启动进行 1 次转换

001: ADC 一次启动进行 4 次转换, 结果累加

010: ADC 一次启动进行 8 次转换, 结果累加

011: ADC 一次启动进行 16 次转换, 结果累加

100: ADC 一次启动进行 32 次转换, 结果累加

其他: ADC 一次启动进行 32 次转换, 结果累加

注: ADACC[2:0]控制累加数, ADRST[1:0]控制移位数, AD 转换结果累加运算后进行移位操作, 然后传至 [ADRH:ADRL]16 位结果寄存器

BIT[1:0] ADBIT[1:0] - ADC 工作模式选择

00: ADC 12-bit mode

01: ADC 10-bit mode

10: ADC 8-bit mode

11: ADC 6-bit mode

11.4.5 ADC 控制寄存器 4 (ADCCR4, 0xF5/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCR4	EXVRS1	EXVRS0	EXVREF_EN	=	ADTRG3	ADTRG2	ADTRG1	ADTRG0
R/W	R/W	R/W	R/W	=	R/W	R/W	R/W	R/W
复位值	0	0	0	=	0	0	0	0

BIT[7:6] EXVRS[1:0] - 外部 VREF pin 选择控制

00: VREF pin 由片外参考提供, 内部高精度参考关闭, 并断开与 VREF pin 连接

01: 1.2V Internal VREF 输出到 VREF pin

10: 2.4V Internal VREF 输出到 VREF pin

11: 保留

注: VREF pin, 作为外部参考输入或内部高精度参考输出, 需要外加退耦电容 ($\geq 0.1\mu\text{F}$)

BIT[5] EXVREF_EN - 端口 VREF 使能位



0: 端口不作为 VREF 功能;

1: 端口用作 VREF 功能。

注: 当该位置 1 时且相应的 IO 设置为模拟输入, VREF 引脚输入或内部高精度 VREF 输出有效。

BIT[3] ADTRG3 - PWM1 触发 ADC 启动使能位

0: 无效;

1: 使能。

BIT[2] ADTRG2 - EADTRG 引脚上升沿触发 ADC 启动使能位

0: 无效;

1: 使能

BIT[1:0] ADTRG[1:0] - TIM0-2 触发 ADC 启动选择位

00: 任意 Timer 都不触发

01: Timer0 溢出时触发

10: Timer1 溢出时触发

11: Timer2 溢出时触发

注 1: ADTRG3 控制的 PWM1、ADTRG2 控制的 EADTRG 引脚、ADTRG[1:0]控制的 Timer (3 选 1 或不选择) 触发这三类触发可以同时使能并有效;

注 2: ADCEOC 软触发, 不受上述 ADCTRG[3:0]控制影响, 独立触发。

11.4.6 ADC 转换结果高位寄存器 (ADRH, 0xFF/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRH	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8
R/W	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

BIT[3:0] ADR[15:8] - ADC 转换结果高 8 位

11.4.7 ADC 转换结果低位寄存器 (ADRL, 0xFE/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
R/W	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

BIT[7:0] ADR[7:0] - ADC 转换结果低 8 位

注: 芯片复位后, ADC 结果寄存器中的值为随机值。

11.4.8 ADC 数字比较上限高位寄存器 (ADCMPGH, 0xFF40/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCMPGH	CMPG15	CMPG14	CMPG13	CMPG12	CMPG11	CMPG10	CMPG9	CMPG8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[3:0] CMPG[15:8] - ADC 上限比较高 8 位

11.4.9 ADC 数字比较上限低位寄存器 (ADCMPGL, 0xFF41/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCMPGL	CMPG7	CMPG6	CMPG5	CMPG4	CMPG3	CMPG2	CMPG1	CMPG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] CMPG[7:0] - ADC 上限比较低 8 位

11.4.10 ADC 数字比较下限高位寄存器 (ADCMPHL, 0xFF42/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCMPHL	CMPL15	CMPL14	CMPL13	CMPL12	CMPL11	CMPL10	CMPL9	CMPL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[3:0] CMPL[15:8] - ADC 下限比较高 8 位

11.4.11 ADC 数字比较下限低位寄存器 (ADCMPLL, 0xFF43/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCMPLL	CMPL7	CMPL6	CMPL5	CMPL4	CMPL3	CMPL2	CMPL1	CMPL0



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] CMPL[7:0] - ADC 下限比较低 8 位

11.4.12 ADC 自动扫描控制寄存器 0 (ADASCRO, 0xFA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADASCRO	ADJSTR	-	ASCNUM2	ASCNUM1	ASCNUM0	ASSTA	ASTRG	ASEN
R/W	R/W	-	R/W	R/W	R/W	R	R/W	R/W
复位值	0	-	0	0	0	0	0	0

BIT[7] ADJSTR - ADC 校准启动位

0: 校准流程结束, ADC 切换为正常模式。

1: 启动 ADC 校准流程, ADC 切换为校准模式, 开始自校准。

BIT[4:3] ASCNUM[1:0] - 自动扫描通道数设置

000: 自动扫描 1 个通道, SEQ0[3:0]对应的通道号;

001: 自动扫描 2 个通道, SEQ0[3:0]和 SEQ1[3:0]对应的通道号

010: 自动扫描 3 个通道, SEQ0[3:0] ~ SEQ2[3:0]对应的通道号

011: 自动扫描 4 个通道, SEQ0[3:0] ~ SEQ3[3:0]对应的通道号

100: 自动扫描 5 个通道, SEQ0[3:0] ~ SEQ4[3:0]对应的通道号

101: 自动扫描 6 个通道, SEQ0[3:0] ~ SEQ5[3:0]对应的通道号

其他: 保留

注: SEQx[3:0]对应的通道配置, 参见 ADSEQRO/1/2 寄存器描述

BIT[2] ASSTA - 自动扫描工作状态位, 只读

0: 自动扫描未进行或已完成, 一次扫描完成 ADCIF 会被置位。

1: 自动扫描进行中

BIT[1] ASTRG - 自动扫描触发控制

0: 每次转换需要一次触发启动

1: 每次扫描需要一次触发启动

BIT[0] ASEN - 自动扫描使能控制位

0: 清零此位, 完成当前扫描周期后, 停止自动扫描功能。

1: 启动自动扫描功能, 但需要触发启动信号, 才会开始转换。

11.4.13 ADC 自动扫描控制寄存器 1 (ADASCRI, 0xFB/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADASCRI	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:1] ADDR[7:0] - ADC 自动扫描存储起始地址 BIT[7:0]位

自动扫描模式, ADC 转换结果存储在 XRAM 区, ADDR[10:0]对应 XRAM 地址的 BIT[10:0]; bit0 无效, 即 XRAM 的存储地址从偶数字节地址开始 (16bit 对齐), 地址 2 字节自增; 每次转换存储的 2 个 bytes 固定为大端模式 (高有效字节存储在偶数地址, 低有效字节存储在奇数地址)。

注: 此字段控制位, 可以在扫描周期期间修改为新地址值, 下个扫描周期有效。请必须在下次扫描周期前更新此寄存器, 若设置值不变也要重新写入。

11.4.14 ADC 自动扫描控制寄存器 2 (ADASCRI2, 0xFC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADASCRI2	ASCNT4	ASCNT3	ASCNT2	ASCNT1	ASCNT0	ADDR10	ADDR9	ADDR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:3] ASCNT[4:0] - ADC 自动扫描输出计数设置

这个字段指定每个扫描周期收集多少 ADC 输出数据, 收集数据数量=(ASCNT[4:0]+1), 收集的 ADC 结果数据根据 ADDR[10:0]定义的起始地址进行存储, 地址自增; 每次存储 2 个 bytes XRAM, 故一个扫描周期存储 XRAM 的 bytes 数为 (ASCNT[4:0]+1)*2。

注: 此字段控制位, 可以在扫描周期期间修改为新值, 下个扫描周期有效。

BIT[2:0] ADDR[10:8] - ADC 自动扫描存储起始地址 BIT[10:8]位

注: 此字段控制位, 可以在扫描周期期间修改为新地址值, 下个扫描周期有效。请必须在下次扫描周期前更新此寄存器, 若设置值不变也要重新写入。



11.4.15 ADC 扫描序列寄存器 0 (ADSEQR0, 0xFF45/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADSEQR0	SEQ13	SEQ12	SEQ11	SEQ10	SEQ03	SEQ02	SEQ01	SEQ00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

11.4.16 ADC 扫描序列寄存器 1 (ADSEQR1, 0xFF46/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADSEQR1	SEQ33	SEQ32	SEQ31	SEQ30	SEQ23	SEQ22	SEQ21	SEQ20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

11.4.17 ADC 扫描序列寄存器 2 (ADSEQR2, 0xFF47/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADSEQR2	SEQ53	SEQ52	SEQ51	SEQ50	SEQ43	SEQ42	SEQ41	SEQ40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:4]/[3:0] SEQx[3:0] - ADC 自动扫描序列通道选择, x=0~5

0000: 通道 0 (AIN0)
 0001: 通道 1 (AIN1)

 1010: 通道 10 (AIN10)
 1011: 通道 11 (OP0OUT)
 1100: 通道 12 (Temp Sensor)
 1101: 通道 13 (LDO_OUT)
 1110: 通道 14 (VDD)
 1111: 通道 15 (GND)

注: 此寄存器序列通道配置仅在自动扫描模式开启且设置的扫描通道个数包含当前的序列时有效, 否则无效; 当自动扫描模式关闭, 转换的通道为 ADCHS[3:0]控制;

11.4.18 ADC 零偏校准寄存器 (ADCALOS, 0xFF44/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCALOS	CALOSEN	OS6	OS5	OS4	OS3	OS2	OS1	OS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] CALOSEN - ADC 零偏校准使能

0: OS[6:0]禁止修改

1: OS[6:0]允许修改

BIT[6:0] QS[6:0] - ADC 零偏校准位

QS[6]用于控制零偏修调方向, QS[5:0]控制修调量

OS[6:0]	修调电压 (典型值)
000 0000	-63 LSB
000 0001	-62 LSB
---	---
011 1101	-2 LSB
011 1110	-1 LSB
011 1111	0 LSB
100 0000	+1 LSB
100 0001	+2 LSB
---	---
111 1110	+62 LSB
111 1111	+63 LSB

注: 在 ADC 的修调结果中 OS[6:0]=100 0000B 或 011 1111B, 两者效果是一样的, 它们的偏移电压都是 0mV;

注: 这里的 VREF 可以是 VDD、内部参考、外部 Vref 输入电压三选一, 与用户的实际选用的 AD 参考电压有关。



12 增强型异步通讯UART

12.1 概述

片上包含 2 路 UART:

UART0 兼容 8051 标准的增强型通用异步收发器 UART, 波特率可选择为系统时钟分频或者定时器 T1/T2 的时钟溢出。

UART0 的增强功能包括帧出错检测以及自动地址识别, UART0 支持 8 位同步半双工 (方式 0)、8 位异步全双工 (方式 1)、9 位异步固定波特率 (方式 2)、9 位异步可变波特率 (方式 3) 4 种工作方式。

UART1 自带波特率发生器, 波特率可选择为系统时钟分频或者波特率发生器输出。

UART1 没有错误检测和自动硬件地址识别, 且 UART1 仅支持 8 位异步全双工 (方式 1)、9 位异步可变波特率 (方式 3) 2 种工作方式, 其寄存器功能和 UART0 相同。

12.2 UART0

UART0 有 4 种工作方式。在通讯之前必须先初始化串口控制寄存器 SCON, 选择 UART 的工作方式和波特率。如果使用方式 1 或方式 3 还应先初始化定时器 T1/T2。

在所有四种方式中, 任何将串口缓冲寄存器 SBUF 作为目标寄存器的写操作都会启动发送。在方式 0 中由条件 RI=0 和 REN=1 初始化接收, 将在 TxD 引脚上产生 1 个时钟信号, 然后在 RxD 引脚上串行移入/移出 8 位数据。在其他方式中则利用外部输入的起始位来初始化接收 (如果 REN=1), 外部发送器通过发送起始位开始通信。

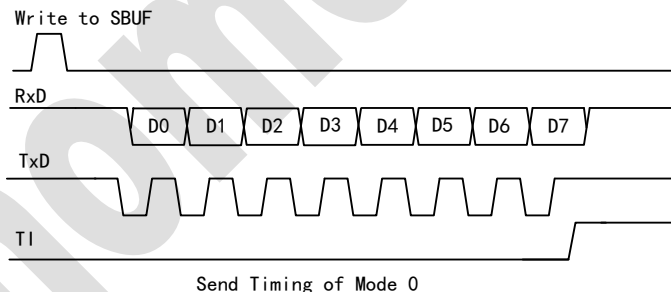
UART 方式列表

SM[0:1]	方式	类型	波特率	帧长度	起始位	停止位	第 9 位
00	0	同步	Fsys/4 或 Fsys/12	8 位	无	无	无
01	1	异步	T1/T2 溢出率/16	10 位	1	1	无
10	2	异步	Fsys/32 或 Fsys/64	11 位	1	1	0/1
11	3	异步	T1/T2 溢出率/16	11 位	1	1	0/1

12.2.1 方式 0: 8 位同步半双工

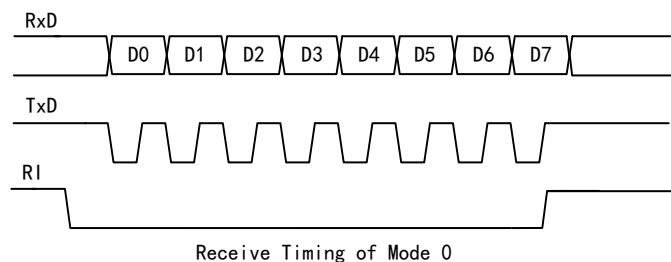
方式 0 支持与外部设备的同步通信。UART 通过 TxD 引脚发送移位时钟, 在 RxD 引脚上收发串行数据。因此这个方式是串行通信的半双工方式, 在此方式中, 每帧收发 8 位, 低位先接收/发送。

通过设置寄存器 SCON 中的 SM2 位为 0 或 1, 波特率固定为系统时钟的 1/12 或 1/4。当 SM2 位为 0 时, 串行端口以系统时钟的 1/12 运行。当 SM2 位置 1 时, 串行端口以系统时钟的 1/4 运行。与标准 8051 唯一不同的是, 芯片在方式 0 中有可变波特率。

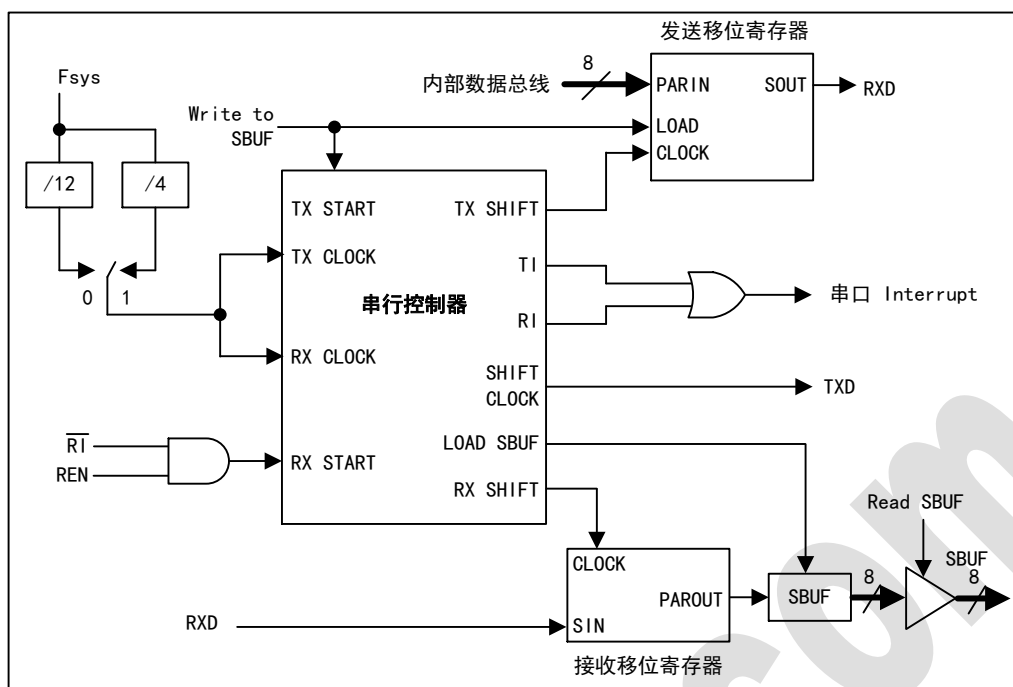


UART 模块通过 TxD 引脚输出同步时钟, 通过 RxD 引脚将数据读入或移出串行端口。任何将 SBUF 作为目标寄存器的写操作都会启动发送。下一个系统时钟 Tx 控制块开始发送。数据转换发生在移位时钟的上升沿, 移位寄存器的内容逐次从左往右移位, 空位置 0。当移位寄存器中的所有 8 位都发送后, TX 控制模块停止发送操作, 然后在下一个系统时钟的上升沿将 TI 置 1。

REN 位置 1 和 RI 位清 0 将初始化接收。下一个系统时钟启动接收, 在移位时钟的上升沿锁存数据, 接收转换寄存器的内容逐次向左移位。当所有 8 位都接收到接收移位寄存器中后, RX 控制模块停止接收, 然后在下一个系统时钟的上升沿上 RI 置 1, 直到被软件清 0 才允许接收。



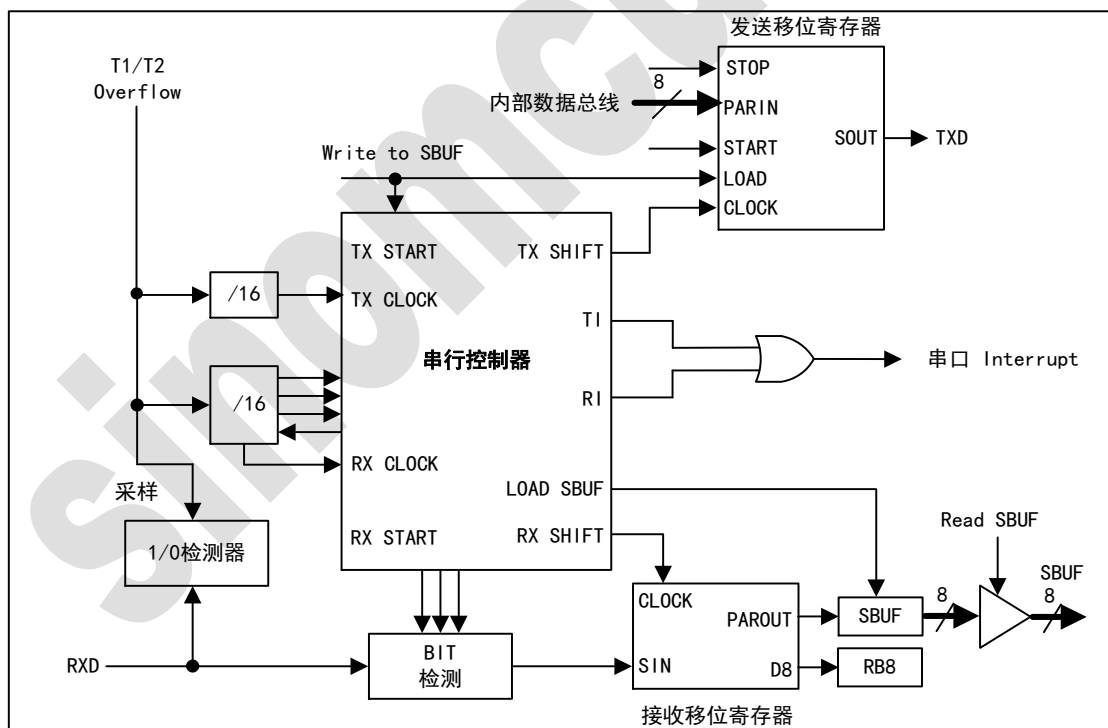
方式 0 功能块框图如下图所示:



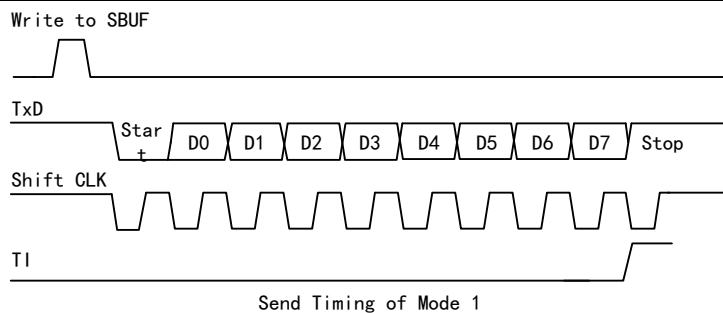
12.2.2 方式 1：8 位异步全双工，可变波特率

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0）、8 个数据位（低位在前）和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中而停止位储存在 RB8 位中。方式 1 中的波特率是可变的，串行收发波特率为定时器 T1/T2 溢出率的 1/16。

方式 1 功能块框图如下图所示:



任何将 SBUF 作为目标寄存器的写操作都会启动发送，实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TxD 引脚上移出，然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后，停止位在 TxD 引脚上移出，在停止位发出的同时 TI 标志置 1。

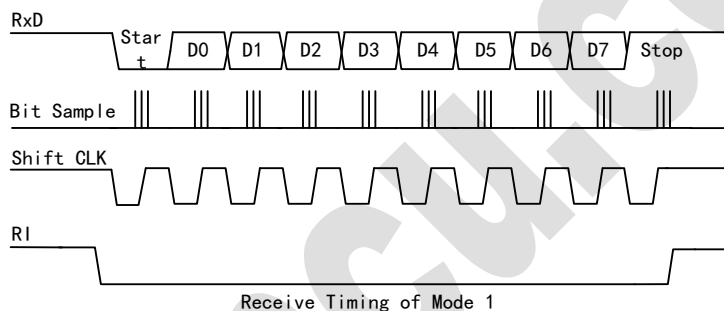


只有 REN 位置 1 时才允许接收。当 RxD 引脚检测到下降沿时串行口开始接收串行数据。为此，芯片对 RxD 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位，这有助于 16 分频计数器与 RxD 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RxD 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RxD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，RI 置 1，但必须满足下列条件：

(1) RI=0;

(2) SM2=0，或者接收的停止位=1;

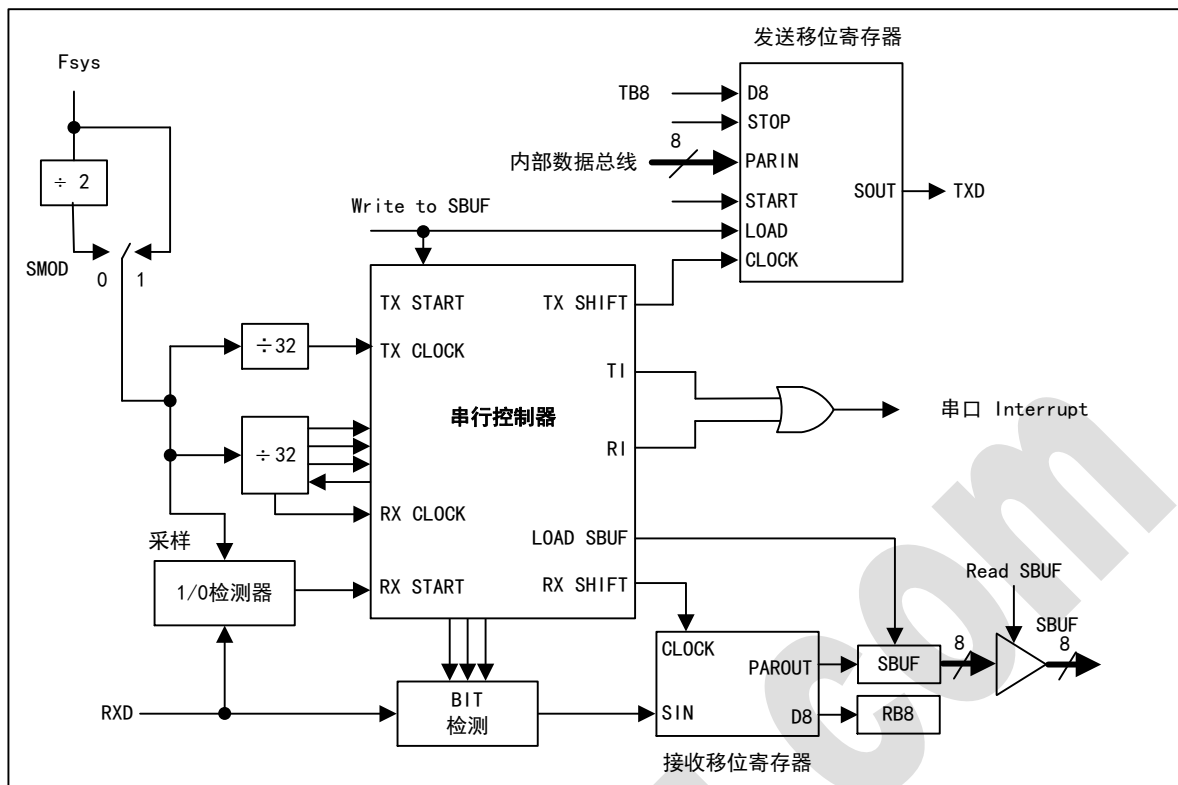
如果这些条件被满足，那么停止位移入 RB8，8 个数据位移入 SBUF，RI 被置 1。否则接收的帧会丢失。这时，接收器将重新去探测 RxD 端是否另一个下降沿。用户必须用软件清除 RI，然后才能再次接收。



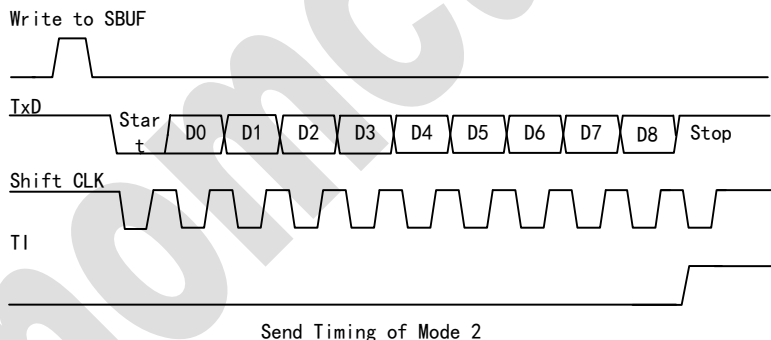
12.2.3 方式 2：9 位异步全双工，固定波特率

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位在前），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 2 支持多机通信和硬件地址识别（详见多机通讯章节）。在数据传送时，第 9 数据位（SCON 中的 TB8）可以写 0 或 1，例如，可写入 PSW 中的奇偶位 P，或用作多机通信中的数据/地址标志位。当接收到数据时，第 9 数据位进入 RB8 而停止位不保存。

方式 2 功能块框图如下所示：



任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TxD 引脚上移出，然后是第 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TxD 引脚上移出，在停止位开始发送时 TI 标志置 1。



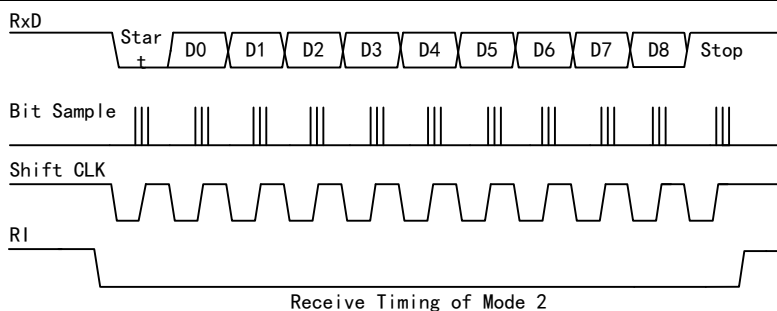
Send Timing of Mode 2

只有 REN 位置 1 时才允许接收。当 RxD 引脚检测到下降沿时串行口开始接收串行数据。为此，芯片对 RxD 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位。这有助于 16 分频计数器与 RxD 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RxD 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RxD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，RI 置 1，但必须满足下列条件：

(1) RI=0;

(2) SM2=0 或者接收的第 9 位=1，且接收的字节符合实际从机地址；

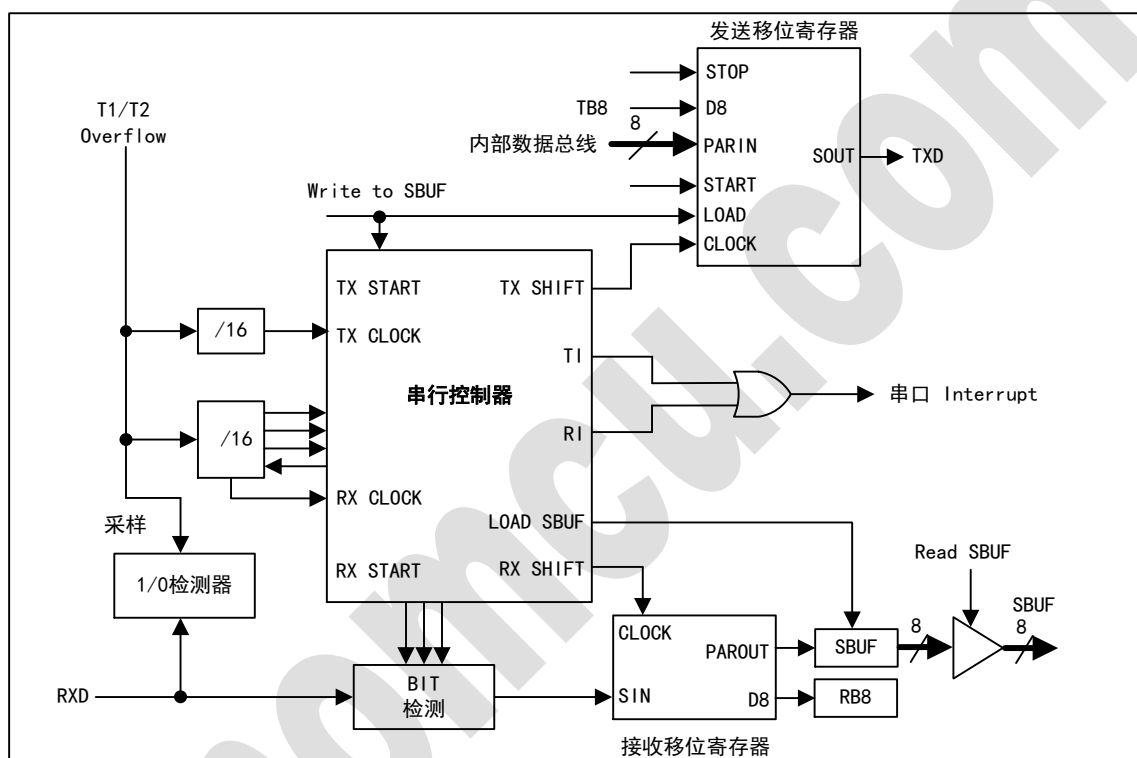
如果这些条件被满足，那么第 9 位移入 RB8，8 位数据移入 SBUF，RI 被置 1。否则接收的数据帧会丢失。在停止位的当中，接收器回到寻找 RxD 引脚上的另一个下降沿。用户必须用软件清除 RI，然后才能再次接收。



12.2.4 方式 3：9 位异步全双工，可变波特率

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式。

方式 3 功能块框图如下所示：



12.2.5 波特率计算

UART 方式 0

波特率可编程为系统时钟的 1/12 或 1/4，由 SM2 位决定。当 SM2 为 0 时，串行端口在系统时钟的 1/12 下运行；当 SM2 为 1 时，串行端口在系统时钟的 1/4 下运行。

$$\text{UART0 波特率} = \frac{F_{\text{sys}}}{4}$$

或者

$$\text{UART0 波特率} = \frac{F_{\text{sys}}}{12}$$

UART 方式 1 和方式 3

T1 在工作方式 2 下作为波特率发生器，在 UART0 方式 1 和方式 3 中的波特率计算公式为：

$$\text{T1 产生的波特率} = \frac{1}{16} \times \frac{F_{\text{t1}}}{[256 - (\text{TH1})]}$$

其中 F_{t1} 为 T1 实际的工作频率，TH1 为实际设定值。

注：当用定时器 T1 作为波特率发生器时，T1 的 TH1 值不可设置为 0xFF，并应该同时关闭相应 T1 的中断使能位。

T2 在工作方式 3 下作为波特率发生器，在 UART0 方式 1 和方式 3 中波特率计算公式：

$$\text{T2 产生的波特率} = \frac{1}{16} \times \frac{F_{\text{sys}}}{\{\text{Pre_scale} \times [65536 - (\text{RCAPH2}) \times 256 + (\text{RCAPL2})]\}}$$



Fsys 为系统时钟频率，预分频值 Pre_scale 为 T2 时钟选择位 T2CKS[2:0] 的分频系数，RCMPH2/RCMPL2 为实际设定值。

注：当用定时器 T2 作为波特率发生时，T2 的 RCAP2 值不可设置为 0xFFFF，并应该同时关闭相应 T2 的中断使能位。

UART 方式 2

UART0 的波特率为：

$$\text{UART0 波特率} = \frac{F_{\text{sys}}}{64} \times 2^{\text{SMOD0}}$$

其中 SMOD0 位根据用户设定值可以取 0 或者 1，起到倍频的作用。

12.2.6 多机通讯

软件地址识别

方式 2 和方式 3 有一个专门的适用于多机通讯的功能。在这两个方式下，接收的是 9 位数据，第 9 位移入 RB8 中，然后再来一位停止位。UART 可以这样设定：当接收到停止位时，只有在 RB8=1 的条件下，串行口中断才会有效（请求标志 RI 置 1）。可通过将 SCON 寄存器的 SM2 位置 1 使 UART 具有这个功能。

在多机通讯系统中，以如下所述来利用这一功能。当主机要发送一数据块给几个从机中的一个时，先送出一地址字节，以辨认目标从机。地址字节与数据字节可用第 9 数据位来区别，地址字节的第 9 位为 1，数据字节的第 9 位为 0。

如果从机 SM2 为 1，则不会响应数据字节中断。地址字节可以中断所有从机，这样，每一个从机都检查所接收到的地址字节，以判别自己是不是目标从机。被寻到的从机将 SM2 位清 0，并准备接收即将到来的数据字节，当接收完毕时，从机再一次将 SM2 置 1。没有被寻到的从机，则维持其 SM2 位为 1，忽略到来的数据字节，继续做自己的事情。

注：在方式 0 中，SM2 用来选择波特率加倍。在方式 1 中，SM2 用来检测停止位是否有效，如果 SM2 = 1，接收中断不会响应直到接收到一个有效的停止位。

自动（硬件）地址识别

在方式 2 和方式 3 中，SM2 置 1 将使 UART 在如下状态运行：当 1 个停止位被接收时，如果载入 RB8 的第 9 数据位为 1（地址字节）并且接收到的数据字节符合 UART 的从机地址，UART 产生一个中断。接着，从机应将 SM2 清零，以接收后续的数据字节。

在 9 位方式下要求第 9 位为 1 以表明该字节是地址而非数据。当主机要发送一组数据给几个从机中的一个时，必须先发送目标从机的地址。所有从机在等待接收地址字节时，为了确保仅在接收地址字节时产生中断，SM2 位必须置 1。自动地址识别的特点是只有地址匹配的从机才能产生中断，地址比较通过硬件完成而不是软件。

中断产生后，地址相匹配的从机清零 SM2，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。一旦全部信息接收完毕，地址匹配的从机应该再次把 SM2 置 1，忽略所有传送的非地址字节，直到接收到下一个地址字节。

使用自动地址识别功能时，主机可以通过调用给定的从机地址选择与一个或多个从机通信。使用广播地址可以联系所有的从机。有两个特殊功能寄存器用来定义从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个 8 位的字节，存于 SADDR 寄存器中。SADEN 用于定义 SADDR 内位的有效与否，如果 SADEN 中某一位为 0，则 SADDR 中相应位的被忽略，如果 SADEN 中某一位置 1，则 SADDR 中相应位的将用于得到给定的从机地址。这可以使用户在不改变 SADDR 寄存器中的从机地址的情况下灵活地寻址多个从机。使用给定地址可以识别多个从机而排除其他的从机。

	从机 1	从机 2
SADDR	10100100	10100111
SADEN（为 0 的位被忽略）	11111010	11111001
实际从机地址	10100x0x	10100xx1
广播地址（SADDR 或 SADEN）	1111111x	11111111

从机 1 和从机 2 给定地址的最低位是不同的。从机 1 忽略了最低位，而从机 2 的最低位是 1。因此只与从机 1 通讯时，主机必须发送最低位为 0 的地址（10100000）。类似地，从机 1 的第 1 位为 0，从机 2 的第 1 位被忽略。因此，只与从机 2 通讯时，主机必须发送第 1 位为 1 的地址（10100011）。如果主机希望同时与两从机通讯，则第 0 位为 1，第 1 位为 0，第 2 位被两从机都忽略，此时有两个不同的地址用于选定两个从机（1010 0001 和 1010 0101）。

主机可以通过广播地址与所有从机同时通讯。这个地址等于 SADDR 和 SADEN 的逻辑或，结果中的 0 表示该位被忽略。多数情况下，广播地址为 0xFFh，该地址可被所有从机应答。

系统复位后，SADDR 和 SADEN 两个寄存器初始化为 0，这两个结果设定了给定地址和广播地址为 XXXXXXXX（所有位都被忽略）。这有效地去除了多处机通讯的特性，禁止了自动寻址方式。这样的 UART 将对任何地址都产生应答，兼容了不支持自动地址识别的 8051 控制器。用户可以按照上面提到的方法实现软件识别地址的多机通讯。

12.2.7 帧出错检测

当寄存器 PCON 中的 SSTAT 位为逻辑 1 时，帧出错检测功能才有效。3 个错误标志位被置 1 后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

注：SSTAT=1 时是访问状态位（FE，RXOV 和 TXCOL），SSTAT=0 时是访问方式选择位（SM0，SM1 和 SM2）。

发送冲突

当一个发送正在进行，而此时软件写数据到 SBUF 寄存器，则发送冲突位 TXCOL 置 1。如果发生了冲突，新数据会被



忽略，不会被写入发送缓冲器。

接收溢出

在接收缓冲器中数据未被读取之前，RI 被清 0，此时如果又有新的数据存入接收缓冲器，则接收溢出位 RXOV 位置 1。如果发生了接收溢出，接收缓冲器中原来的数据将丢失。

帧出错

如果检测到一个无效（低）停止位，则帧出错位 FE 置 1。

暂停检测

当连续检测到 11 个位都为低电平位时，则认为检测到一个暂停。由于暂停条件同样满足帧错误条件，因此检测到暂停时也会报告帧错误。一旦检测到暂停条件，UART 将进入空闲状态并一直保持，直至接收到有效停止位（RXD 引脚上出现上升沿）。

12.3 UART1

UART1 自带波特率发生器，在方式 1 和方式 3 下提供波特率。

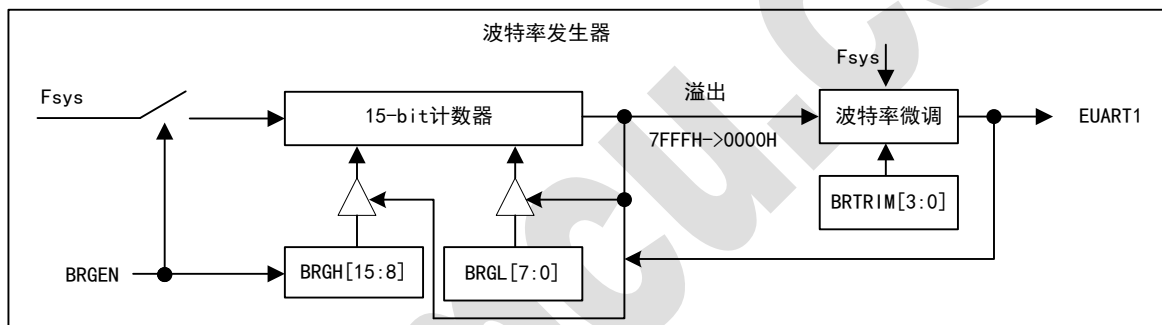
UART1 没有错误检测和自动硬件地址识别，且 UART1 仅支持 8 位异步全双工（方式 1）、9 位异步可变波特率（方式 3）2 种工作方式，除波特率发生器不同，其它功能和 UART0 相同。

注 1： UART1 没有方式 0，故 SM2_1 位在方式 0 中无效。

注 2： UART1 没有方式 2，故无对应的 SMOD0_1 位。

12.3.1 方式 1 和方式 3

UART1 的工作方式与 UART0 类似，自带波特率发生器且波特率可微调，精度为一个系统时钟。其他操作与 UART0 相同。



波特率公式如下：

$$\text{波特率} = \frac{F_{\text{sys}}}{16 \times (65536 - [BRGH, BRGL]) + BRTRIM}$$

举例：Fsys = 12MHz，计算 115200bps 波特率如下：

$$12000000/16/115200 = 6.51$$

$$[BRGH, BRGL] = 65536 - 6 = 65530$$

由波特率公式， $115200 = 12000000 / (16 \times 6 + BRTRIM)$ ，得 $BRTRIM = 8.2$ ，取整 8。

实际波特率 = $12000000 / (16 \times 6 + 8) = 115384.6\text{bps}$ ，误差 0.16%。





98		寄存器		V	OL						000
SFR0/ 87	PCON	电源控制寄存器	UART1EN	-	-	UART0EN	SMOD0	SSTAT0	SLEEP	STOP	0—0 0100

12.5.1 电源控制寄存器 (PCON, 0x87/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	UART1EN	-	-	UART0EN	SMOD0	SSTAT0	SLEEP	STOP
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
复位值	0	-	-	0	1	0	0	0

BIT[7、4] **UARTnEN** (n = 0、1) - UARTx (x = 0、1) 模块使能位

0: 关闭 UARTx 模块, 端口用于通用 IO 或其他复用模块;

1: 打开 UARTx 模块。

BIT[3] **SMOD0** - UART0 波特率倍增位

0: UART0 波特率保持现状;

1: UART0 波特率增加一倍。

注: 该控制位, 因模块本身的差异, 仅在 UART0 中存在, UART1 中并不存在; 且仅在 UART0 的方式 2 中有效, 其他方式无效。

BIT[2] **SSTAT0** - UART0 寄存器功能选择位

0: 寄存器 SCON[7:5] 功能为 SM[0:2];

1: 寄存器 SCON[7:5] 功能为 FE, RXOV, TXCOL。

注: 该控制位, 因模块本身的差异, 仅在 UART0 中存在, UART1 中并不存在。

特别注意: 上述控制位 (PCON[7:2]) 皆不受 SLPCR 保护寄存器的影响, 可以直接读/写。

12.5.2 串口 0 控制寄存器 (SCON, 0x98/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

12.5.3 串口 1 控制寄存器 (SCON_1, 0x9C/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON_1	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

UART0 在 SSTAT=0 时:

BIT[7:6] **SM[0:1]** 或者 **SM[0:1]_1** - UART 方式控制位 (SSTAT=0)

SM[0:1]	UART 工作方式	适用
00	方式 0: 同步方式, 固定波特率	UART0
01	方式 1: 8 位异步方式, 可变波特率	UART0、UART1
10	方式 2: 9 位异步方式, 固定波特率	UART0
11	方式 3: 9 位异步方式, 可变波特率	UART0、UART1

注: 当 UAET1 的 SM[1:0] 设置为 00 或者 10 时, UART1 不能正常工作。

BIT[5] **SM2** 或者 **SM2_1** - UART 功能设定位 (SSTAT=0)

SM2	方式 0	方式 1	方式 2/3
0	波特率=Fsys/12	禁止停止位确认检验, 停止位将置 RI 为 1 产生中断	任何字节均会置 RI 为 1 产生中断
1	波特率=Fsys/4	允许停止位确认检验, 只有有效的停止位 (1) 才能置 RI 为 1 产生中断	只有寻址字节 (第 9 位=1) 能置 RI 为 1 产生中断

注: 在 UART1 中没有方式 0 和方式 2。

UART0 在 SSTAT=1 时:

BIT[7] **FE** - 帧出错标志位 (SSTAT=1)

0: 无帧出错, 由软件清 0;

1: 发生帧出错, 由硬件置 1。

BIT[6] **RXOV** - 接收溢出标志位 (SSTAT=1)

0: 无接收溢出, 由软件清 0;

1: 接收溢出, 由硬件置 1。

BIT[5] **TXCOL** - 发送冲突标志位 (SSTAT=1)

0: 无发送冲突, 由软件清 0;



1: 有发送冲突, 由硬件置 1。

注: 以上 3 个控制位 (Bit[7-5]), 因模块本身的差异, 仅在 UART0 中存在, UART1 中并不存在。

BIT[4]	REN 或者 REN_1 - 接收器允许位 0: 禁止接收; 1: 允许接收。
BIT[3]	TB8 或者 TB8_1 - 方式 2/3 时发送数据的第 9 位 0: 方式 2/3 发送数据的第 9 位为 0; 1: 方式 2/3 发送数据的第 9 位为 1。
BIT[2]	RB8 或者 RB8_1 - 方式 1/2/3 时接收数据的第 9 位 (停止位或数据位) 0: 方式 1/2/3 时接收数据的第 9 位为 0; 1: 方式 1/2/3 时接收数据的第 9 位为 1。
BIT[1]	TI 或者 TI_1 - 发送中断标志位 0: 由软件清 0; 1: 在方式 0 的第 8 位最后, 或在其他方式的停止位开始, 由硬件置 1。
BIT[0]	RI 或者 RI_1 - 接收中断标志位 0: 由软件清 0; 1: 在方式 0 的第 8 位最后, 或在其他方式的停止位开始, 由硬件置 1。

12.5.4 串口 0 缓存寄存器 (SBUF, 0x99/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF	SBUF7	SBUF6	SBUF5	SBUF4	SBUF3	SBUF2	SBUF1	SBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

12.5.5 串口 1 缓存寄存器 (SBUF_1, 0x9D/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF_1	SBUF7_1	SBUF6_1	SBUF5_1	SBUF4_1	SBUF3_1	SBUF2_1	SBUF1_1	SBUF0_1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

BIT[7:0] SBUF[7:0] 或者 SBUF_1[7:0] - UART 数据缓存, SBUF 访问两个寄存器: 1 个移位寄存器和 1 个接收锁存寄存器。SBUF 的写入将发送字节到移位寄存器中, 然后开始从端口发送; SBUF 的读取将返回接收锁存寄存器中的内容

注: UART0 和 UART1 的发送和接收用缓存, 使用了同一个缓存寄存器, 写入时是一个通道, 读取时是另一个通道, 故用户编写软件中需要注意。

12.5.6 串口 0 从机地址寄存器 (SADDR, 0x9A/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADDR	SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] SADDR[7:0] - UART 的从机地址

注: UART1 没有该寄存器, 也不支持该功能。

12.5.7 串口 0 地址掩码寄存器 (SADEN, 0x9B/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADEN	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] SADENN - SADDRn 检验控制位 (n=7-0)

0: 忽略 SADDRn 位;

1: 检验 SADDRn 位是否对应接收地址。

注: UART1 没有该寄存器, 也不支持该功能。

12.5.8 串口 1 波特率控制寄存器 (BRGCR, 0x9E/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BRGCR	BRGEN	-	-	-	BRTRIM3	BRTRIM2	BRTRIM1	BRTRIM0
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W



复位值	0	-	-	-	0	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[7] BRGEN - 串口1波特率发生器使能控制位

0: 关闭;

1: 使能

BIT[3:0] BRTRIM[3:0] - 串口1波特率微调控制位

波特率微调, 单位为系统时钟 Fsys clock, 设置为0表示不做微调。

12.5.9 串口1波特率计数器低位寄存器 (BRGH, 0x95/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

12.5.10 串口1波特率计数器高位寄存器 (BRGL, 0x94/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BRGL	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] BRG[15:0] - 串口1波特率计数器控制寄存器



13 串行外设接口SPI

13.1 SPI 概述

片上 SPI 支持全双工，3 线/4 线同步传输，主从机操作模式可选，LSB/MSB 优先可选，带芯片中断主模式故障出错标志，带写入冲突标志保护等功能。

串行外部设备接口（SPI）是一种高速串行通信接口，允许芯片与外部设备（包括芯片）进行全双工同步串行信号通信。

13.2 SPI 信号描述

主输出从输入（MOSI）：该信号表示主设备、从设备之间，数据流经主设备串行发送到从设备，主设备输出，从设备输入。

主输入从输出（MISO）：该信号表示主设备、从设备之间，数据流经从设备串行发送到主设备，从设备输出，主设备输入。注意：在 4 线 SPI 通信中从设备必须被选中，方可发送数据；若未被选中 MISO 引脚将处于高阻状态。

串行时钟（SCLK）：该信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动，每 8 个时钟周期 MOSI 和 MISO 线上传送一个字节，如果从设备未被选中 SCLK 信号将被此设备忽略。注意：只有主设备才能产生 SCLK 信号。

从设备选择信号（SS）：当从设备使用时，每一个 SPI 从设备都带有一个片选信号，该信号低电平有效，当引脚信号由高到低反转时表明该从设备被选中，开始与 SPI 主设备进行通信。为了防止 SPI 通信冲突，同一时刻仅允许一个主设备与从设备通信。当做主设备使用时，该引脚可另作他用，从设备设备时交给 SPI 主设备控制。

13.3 工作模式

片上的 SPI 在工作中，仅可配置为主/从模式中的一种，对应于 3 线/4 线 SPI 模式，参考做法如下：

通信模式	本机 SPI 设备	本机 SPI 的 SS 脚	其他 SPI 设备	其他 SPI 设备 SS 脚
一主一从	主模式	可另作他用	从模式	拉低（片选信号始终有效）
一主多从				通过 GPIO 控制，通信时分别拉低（片选有效），切换不同从设备
一从一主	从模式	由其他 SPI 设备控制是否拉低（片选有效）	主模式	可另作他用

13.3.1 主机模式

当 SPCON 寄存器的 MSTR 位置 1 时，SPI 模块工作在主机模式。整个 SPI 系统中只允许一个主机启动传输，每次传输总是由主机发起。

在 SPI 作为主机模式中，向 SPI 的数据寄存器 SPDAT 写入数据后，数据将会写入发送移位缓冲器。如果发送移位寄存器已经存在一个数据，那么主 SPI 产生一个 WCOL 信号以表明写入太快。但是发送移位寄存器中的数据不会受到影响，发送也不会中断。另外如果发送移位寄存器为空，那么主设备立即按照 SCLK 的时钟频率串行地移出发送寄存器中的数据到 MOSI 上。当传送完毕，SPSTA 寄存器中的 SPIIF 位置 1。如果 SPI 中断被允许，当 SPIIF 位置 1，会产生一个中断请求。

当主设备通过 MOSI 传送数据给从设备时，相应的从设备同时也通过 MISO 将其发送移位寄存器的内容传送给主设备的接收移位寄存器，实现全双工。因此，SPIIF 标志位置 1 既表示数据发送完成，又表示接收完成。用户可以从 SPDAT 寄存器中读出接收的数据，并清除 SPIIF。

13.3.2 从机模式

当 SPCON 寄存器的 MSTR 位清 0 时，SPI 模块工作在从机模式。在数据传输之前，从设备的 SS 引脚必须拉低，且必须保持低电平直到一个字节数据传输完毕。当从机设备的 SS 管脚不为低时，主机设备不能与从机交换数据。在数据传输开始前和数据传输完成前，SS 管脚都需要保持低电平状态。如果 SS 变为高电平，SPI 模块将被迫进入闲置状态。如果 SS 管脚在传输的过程被置高，那么传输将被取消，接收移位缓存区里剩下的位数将变高，同时也将进入闲置状态。

在从机模式下，按照主机的 SCLK 的时钟信号，数据通过 MOSI 管脚由主机向从机传输，通过 MISO 管脚由从机向主机传输。一个位计数器记录 SCLK 的边沿数，当数据移位寄存器移入 8 位数据（一字节）；同时，在移位寄存器接收到 8 位数据（一字节），数据将移到接收数据缓存器，同时 SPIIF 置 1。若 SPI 中断被允许，当 SPIIF 置 1 时，会产生一个中断请求。此时接收移位寄存器将保持且 SPIIF 位置 1，这样 SPI 从设备将不会接收到任何数据，直到 SPIIF 位清 0。SPI 从设备必须在主设备开始一次新的数据传输前，将要传的数据写入发送移位寄存器。如果发送开始前未写入数据，从设备将传送“0x00”给主设备。如果写 SPDAT 操作发生在传输过程中，那么 SPI 从设备的 WCOL 标志位置 1，表示写 SPDAT 冲突。但是移位寄存器的数据不受影响，本次传输也不会中断。

13.4 数据传输格式

通过软件设置 SPCON 寄存器的 CPOL 位和 CPHA 位，用户可以选择 SPI 时钟 SCLK 的极性和相位的组合。

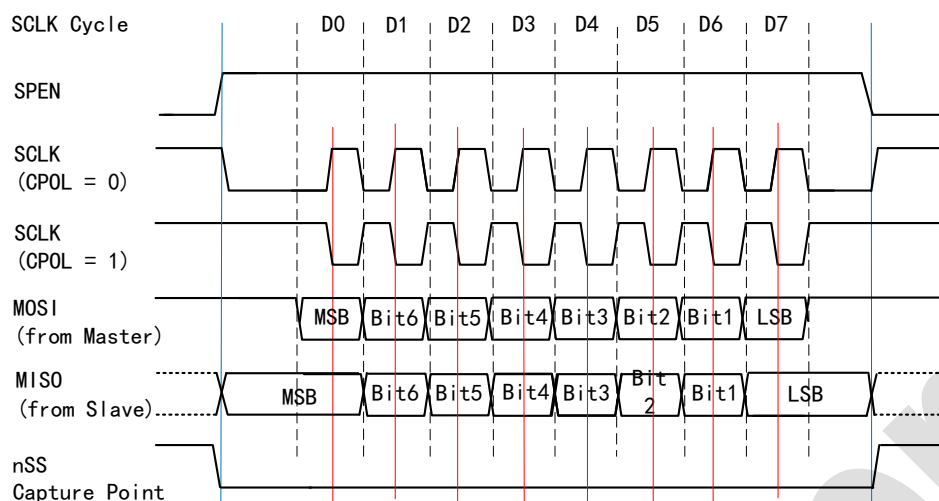
CPOL 位定义时钟的极性，即空闲时的电平状态。

CPHA 位定义时钟的相位，即定义了允许数据采样移位的时钟边沿。在 SPI 通信中主从通讯的两个设备，时钟极性相位的设置应一致。



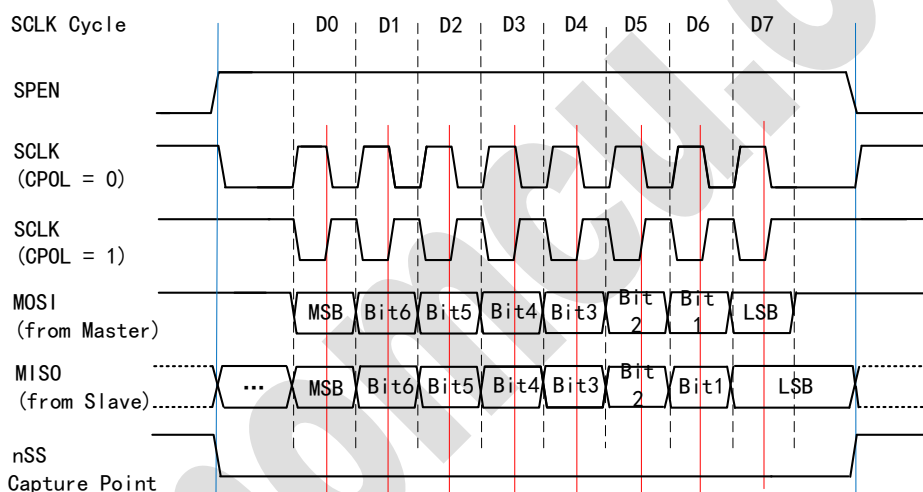
当 CPHA=0, SCLK 的第一个捕获数据, 从设备必须在 SCLK 的第一个沿之前将数据准备好。

CPHA = 0 时的数据传输示意图



当 CPHA=1, 主设备在 SCLK 的第一个沿将数据输出到 MOSI 上, 从设备把 SCLK 的第一沿作为开始发送信号, SCLK 的第二个沿开始捕获数据, 因此用户必须在第一个 SCLK 的两个沿内完成写 SPDAT 的操作。这种数据传输格式是一个主设备一个从设备之间通讯的首选。

CPHA = 1 时的数据传输示意图



13.5 出错检测

出错情形一：模式故障 (MODF 位)

SPI 主模式下的模式故障出错表明 SS 引脚上的电平状态与实际的设备模式不一致。SPSTA 寄存器中 MODF 位置 1 后, 表明系统控制存在多主设备冲突的问题。这种情况下, 由于 SPCON 寄存器的 SS 引脚禁止位 (SSDIS) 清 0, SS 引脚信号为低时, MODF 标志位置 1。然而, 对于只有一个主设备的系统来说, 主设备的 SS 引脚被拉低, 那是绝对不允许的。这种情况下, 为防止 MODF 置 1, 可使 SPCON 寄存器中的 SSDIS 位置 1, 让 SS 引脚作为普通 I/O 口或是其它功能引脚。

此时, 在重启 SPI 通信前, 用户必须将 MODF 位以软件方式清 0, 将 SPCON 寄存器中的 MSTR 位和 SPSTA 寄存器的 SPEN 位置 1, 重新启动主模式。

注: 当 SSDIS=0 且 MODF=1 时将产生 SPI 错误中断。

出错情形二：写冲突 (WCOL 位)

在发送数据序列期间写入 SPDAT 寄存器会引起写的冲突, SPSTA 寄存器中的 WCOL 标志位置 1。WCOL 位置 1 不会引起中断, 发送也不会中止。该标志位仅用于通知用户, SPI 数据写入的时机不对, WCOL 位需由软件清 0。

出错情形三：超限情况 (SPIOV 位)

当作为主设备或从设备时, 软件尚未清除 SPIIF 位, 此时主或从设备又试图发送/接受几个数据字节时, 发生超限情况。在这种情况下, 接收移位寄存器保持原有数据, SPIIF 置 1, 同样 SPI 设备直到 SPIIF 被清除后才会再接收新的数据。在 SPIIF 位被清除之前继续调用中断, SPIOV 位置 1 不会引起中断, 发送也不会中止。该标志位仅用于通知用户, 仍有接收的 SPI 数据未读取, SPIOV 位需由软件清 0。



13.6 SPI 相关寄存器

SPI 寄存器汇总表

地址 (SFRn/H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR0/AF	SPDAT	SPI 数据寄存器	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0	xxxx xxxx
SFR0/AE	SPSTA	SPI 状态寄存器	SPEN	SPIF	MODF	WCOL	SPIOV	-	-	-	00000 ---
SFR0/AD	SPCON	SPI 控制寄存器	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0	00000 000

13.6.1 SPI 控制寄存器 (SPCON, 0xAD/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCON	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] DIR - 传输方向选择位

0: 发送时, 由 MSB 开始依次发送;

1: 发送时, 由 LSB 开始依次发送。

BIT[6] MSTR - SPI 主从选择位

0: 片上 SPI 做从设备使用;

1: 片上 SPI 做主设备使用。

注: 如果要设置 MSTR=1, 务必先让 SSDIS=1; 或者在 SSDIS=0 的状态, 但请提前上拉外部 SS 引脚, 否则 SPI 保护功能生效, 并不能正常进入主模式, 该位无法置 1, 仍保持 0 (从模式状态)。

BIT[5] CPHA - SPI 时钟相位控制位

0: SPI 周期的第一个上升沿采集数据;

1: SPI 周期的第二个上升沿采集数据。

BIT[4] CPOL - SPI 时钟极性控制位

0: 在 Idle 状态下 SCK 处于低电平;

1: 在 Idle 状态下 SCK 处于高电平。

BIT[3] SSDIS - SPI 片选信号 SS 控制位

0: 在主/从模式下, 打开 SS 引脚;

1: 在主/从模式下, 关闭 SS 引脚, SS 脚位可另作他用。

注: 如果 SSDIS 置 1, 不产生 MODF 中断请求; 在从模式下, 若 CPHA 为 0, 该位无效。

BIT[2:0] SPR[2:0] - SPI 时钟速率选择位

SPR[2:0]	SPI 工作时钟 F _{SPI}
000	F _{sys} /2
001	F _{sys} /4
010	F _{sys} /8
011	F _{sys} /16
100	F _{sys} /32
101	F _{sys} /64
110	F _{sys} /128
111	F _{sys} /256

注: 作从模式时, SPI 的时钟由主机决定, 但不超过 8Mbit/s; 作主模式时, SPI 的时钟由上表决定, 最大设置也不超过 8Mbit/s。

13.6.2 SPI 状态寄存器 (SPSTA, 0xAE/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTA	SPEN	SPIF	MODF	WCOL	SPIOV	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-
复位值	0	0	0	0	0	-	-	-

BIT[7] SPEN - SPI 模块使能位

0: 关闭 SPI 模块, 端口用于通用 IO 或其他复用模块;

1: 打开 SPI 模块, MOSI、MISO、SCLK 端口有效, 开始数据的收发。

注: nSS 端口的使能与否, 直接由 SSDIS 位来控制。

BIT[6] SPIF - SPI 数据传送标志位



- 0: 由软件清 0;
1: 表明数据传输已完成, 由硬件置 1。
BIT[5] **MODF** - 模式故障标志位
0: 由软件清 0;
1: 表明 SS 引脚电平与 SPI 模式不匹配, 由硬件置 1。
BIT[4] **WCOL** - 写入冲突标志位
0: 表明没有写入冲突发生, 由软件清 0;
1: 表明已检测到写入冲突发生, 由硬件置 1。
BIT[3] **SPIOV** - 接收超限标志位
0: 表明以处理接收超限, 由软件清 0;
1: 表明已检测到接收超限, 由硬件置 1。

13.6.3 SPI 数据寄存器 (SPDAT, 0xAF/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

BIT[7:0] **SPDAT[7:0]** - SPI 数据缓存寄存器

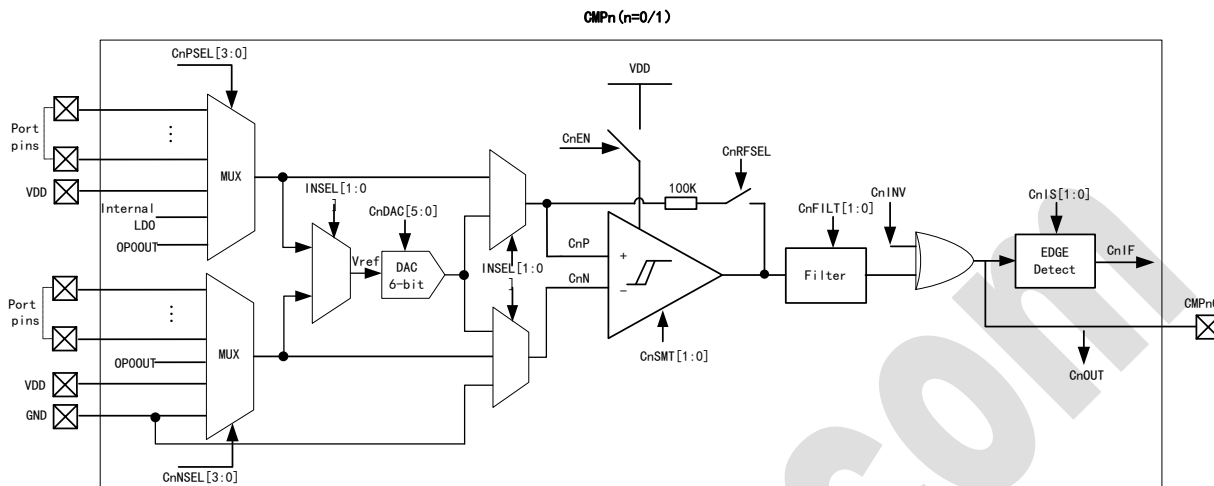
注 1: 写入 SPDAT 的数据被放置到发送移位寄存器中, 而读取 SPDAT 时将获得接收移位寄存器中的数据, 本质上对应了两个不同的寄存器, 因此用户无法直接读取到自己写入的值。

注 2: SPDAT 的复位值为随机值, 因此作主模式发送数据前必须设定其初值; 当关闭 SPI 功能后, 再读取 SPDAT 的内容无效。

14 电压比较器

14.1 概述

片上包含 2 个独立模拟电压比较器, 比较器反相输入端参考信号可选外部输入或内部电压基准, 内部电压基准多档可调, 内置施密特窗口电路和输出数字滤波电路, 输出触发方式可选。



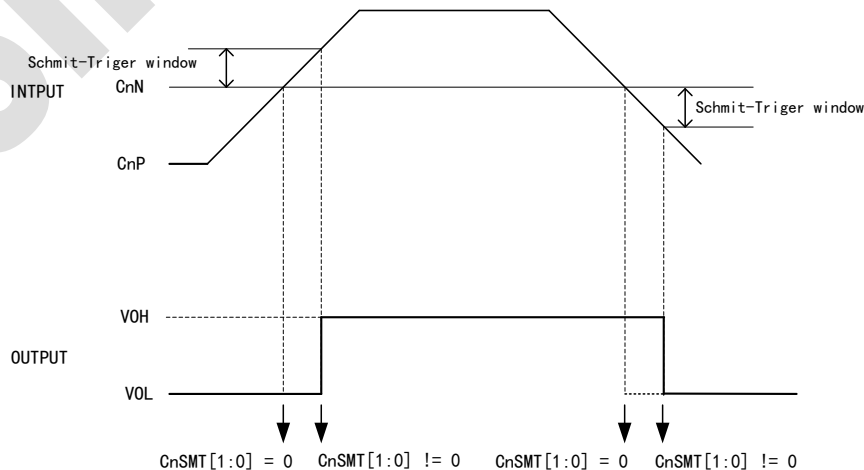
14.2 特性

- ✧ 高达 9 (CMP0) 或 9 (CMP1) 路外部正相输入
- ✧ 高达 9 (CMP0) 或 9 (CMP1) 路外部负相输入
- ✧ 可选择内部输入通道:
- ✧ 内部连接 LDO 输出
- ✧ 直接连接到 GND
- ✧ 直接连接到 VDD
- ✧ 专用的 6-bit DAC 参考
- ✧ 滤波后输出可以映射到所有引脚
- ✧ 可编程施密特窗口电压, 0~90mV 之间, 4 档可调节
- ✧ 输出中断沿可选择上升、下降或双沿
- ✧ 可连接 PWM1 刹车保护功能

14.3 功能描述

14.3.1 施密特窗口（迟滞）

比较器可以通过设置 CMPnCR0 寄存器 CnSMT[1:0] 位 (n=1, 2) 设置施密特比较器窗口(迟滞比较器), 如下图所示:



注: 施密特窗口为双边边窗口, CnN 输入电平为窗口中心值。



14.3.2 输入通道选择

比较器正反相输入内置 MUX 多路选择电路，可以灵活选择不同的通道和内部 DAC 参考。

注：选择作为比较器输入的任何端口引脚都应该配置为模拟输入。

CMP0 多路输入选择列表

CMPnCR2->CnPSEL	通道对应	管脚名称	CMPnCR2->CnNSEL	通道对应	管脚名称
0000	CMP00	P3.2	0000	CMP00	P3.2
0001	CMP01	P3.1	0001	CMP01	P3.1
0010	CMP02	P2.7	0010	CMP02	P2.7
0011	CMP03	P2.6	0011	CMP03	P2.6
0100	CMP04	P2.5	0100	CMP04	P2.5
0101	CMP05	P2.4	0101	CMP05	P2.4
0110	CMP06	P2.3	0110	CMP06	P2.3
0111	CMP07	P2.2	0111	CMP07	P2.2
1000	CMP08	P2.1	1000	CMP08	P2.1
1001	CMP09	P0.4	1001	CMP09	P0.4
1010	LDO_OUT	内部 LDO	1010	GND	GND
1011	VDD	VDD	1011	VDD	VDD
1100	OPOOUT	内部	1100	OPOOUT	内部
1101~1111	-	-	1101~1111	-	-

CMP1 多路输入选择列表

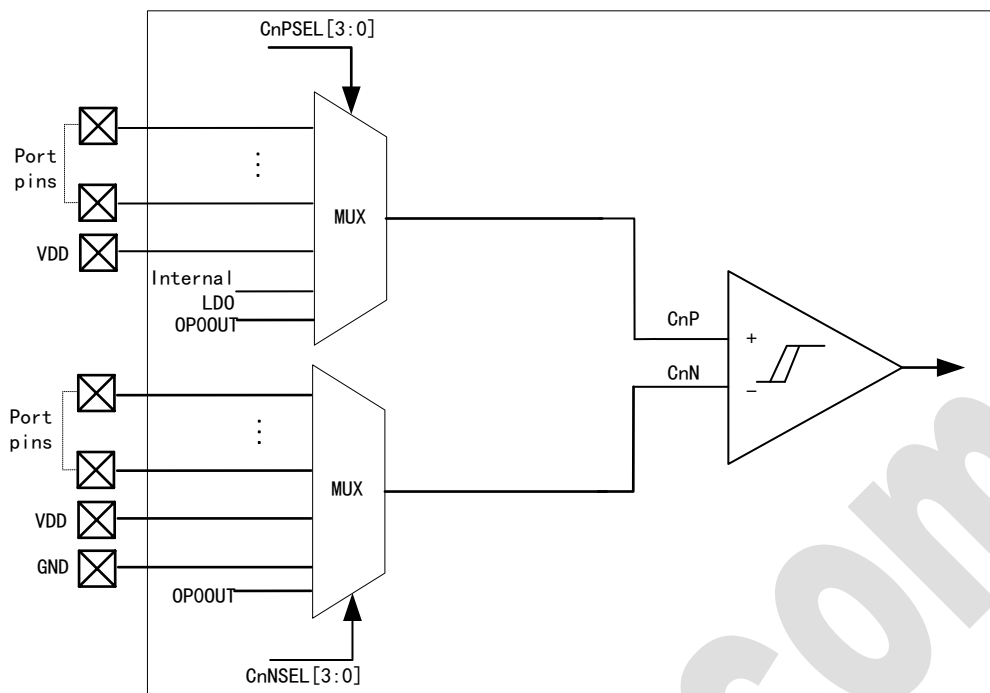
CMPnCR2->CnPSEL	通道对应	管脚名称	CMPnCR2->CnNSEL	通道对应	管脚名称
0000	CMP10	P2.5	0000	CMP10	P2.5
0001	CMP11	P2.4	0001	CMP11	P2.4
0010	CMP12	P2.0	0010	CMP12	P2.0
0011	CMP13	P0.7	0011	CMP13	P0.7
0100	CMP14	P0.6	0100	CMP14	P0.6
0101	CMP15	P0.5	0101	CMP15	P0.5
0110	CMP16	P0.3	0110	CMP16	P0.3
0111	CMP17	P0.2	0111	CMP17	P0.2
1000	CMP18	P0.1	1000	CMP18	P0.1
1001	-	-	1001	-	-
1010	LDO_OUT	内部 LDO	1010	GND	GND
1011	VDD	VDD	1011	VDD	VDD
1100	OPOOUT	内部	1100	OPOOUT	内部
1101~1111	-	-	1101~1111	-	-

14.3.3 内部参考 DAC 及输入模式选择

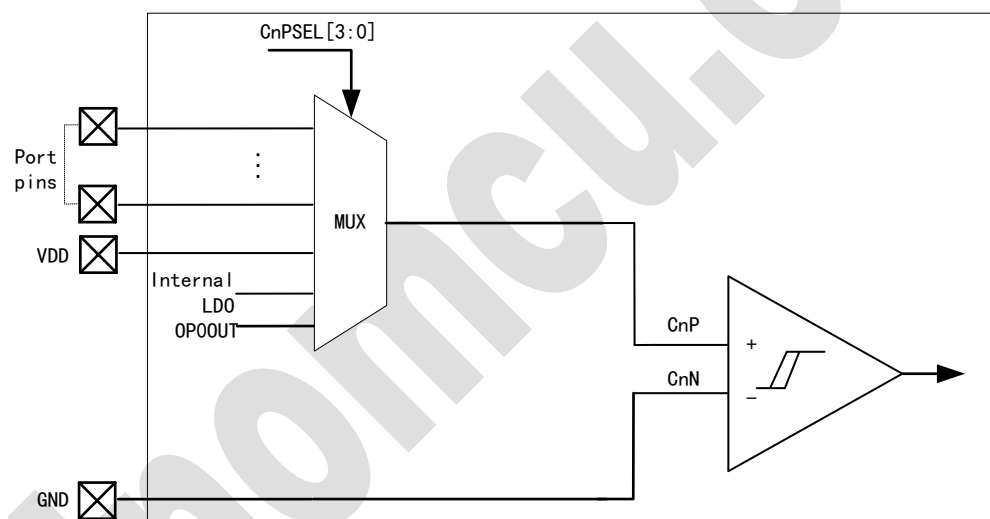
比较器模块包括一个专用的参考 DAC，它可以接入所选的 mux 通道和比较器输入之间（正相和反相）。

CMPnCR1 寄存器中的 CnINSEL 位选择输入模式，mux 输入、参考 DAC 和比较器输入之间的不同连接，组合为四种结构模式。

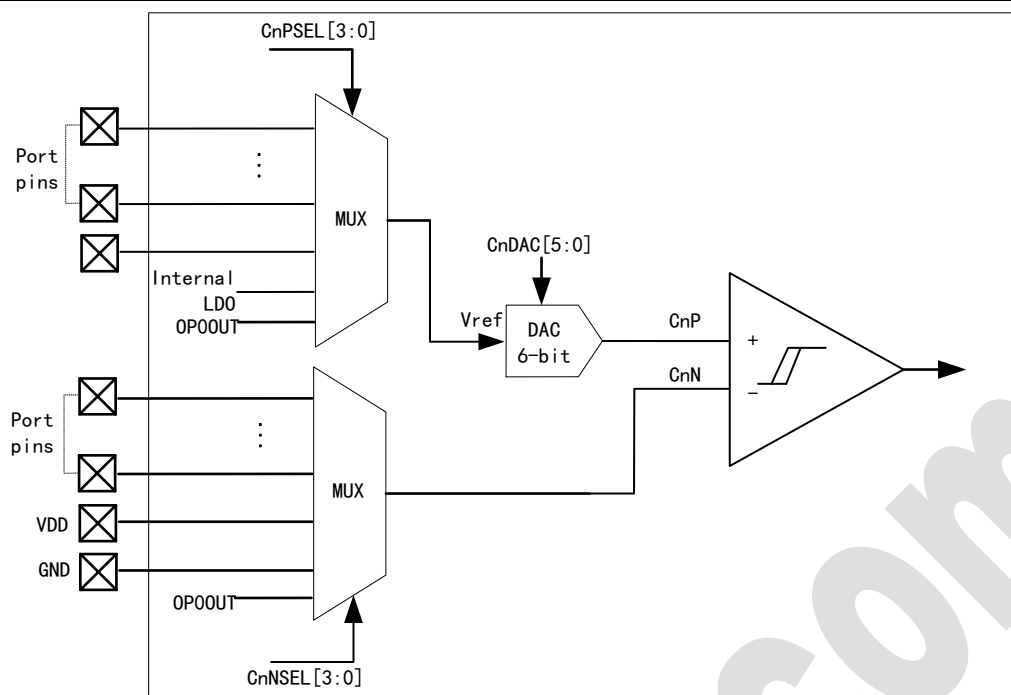
CnINSEL = 00b



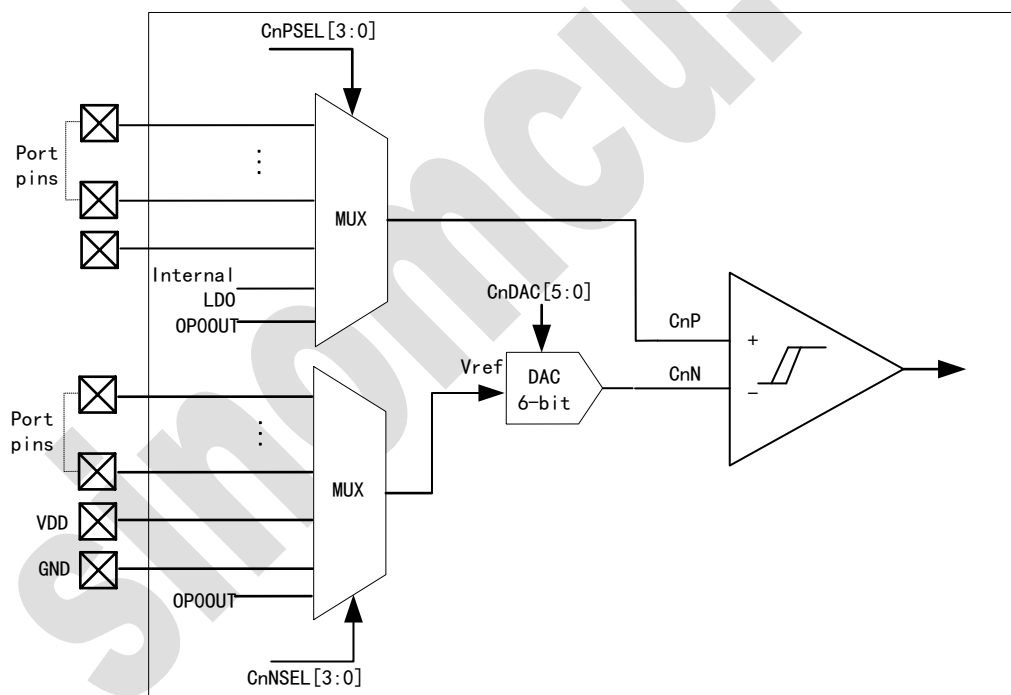
CnINSEL = 01b



CnINSEL = 10b



CnINSEL = 11b



14.3.4 输出控制

比较器输出可通过 I/O 全映射模块连接至任意 port 口，通过读取 CnOUT 位可以获得比较器的输出状态。

CnEN 设置为 1，启动比较器；设置为 0，关闭比较器。当比较器关闭状态，比较器输出 CnOUT 和输出引脚（如果被映射到 I/O）固定为输出低。

CnFILT[1:0] 可选择比较输出的数字滤波 0/1/4/16us 可选；

CnINV 控制比较器输出反向，如果 CnINV=0，CnP>CnN，CnOUT 输出 1，反之输出 0；如果 CnINV=1，CnP>CnN，CnOUT 输出 0，反之输出 1；

设置 CnIS[1:0] 选择中断触发沿，不触发/上升沿触发/下降沿触发/双沿触发可选。当被选择的触发条件发生时，中断标志 CnIF 位置位，若相应的中断使能 CnIE 和 CMPIE（IE1 寄存器）置 1，则发生比较器中断。

注：当比较器首次启动、施密特迟滞窗口修改可能产生错误的沿输出，建议启动后短时间内清零标志 CnIF，再开启



中断使能。

14.3.5 内部电阻反馈

配置 CnRFSEL 为 1 可选择内部 100k 电阻反馈，当内建的施密特迟滞电压无法满足需求，可以通过外接电阻和内部 100k 电阻组成正反馈电路，实现灵活的迟滞电压选择。

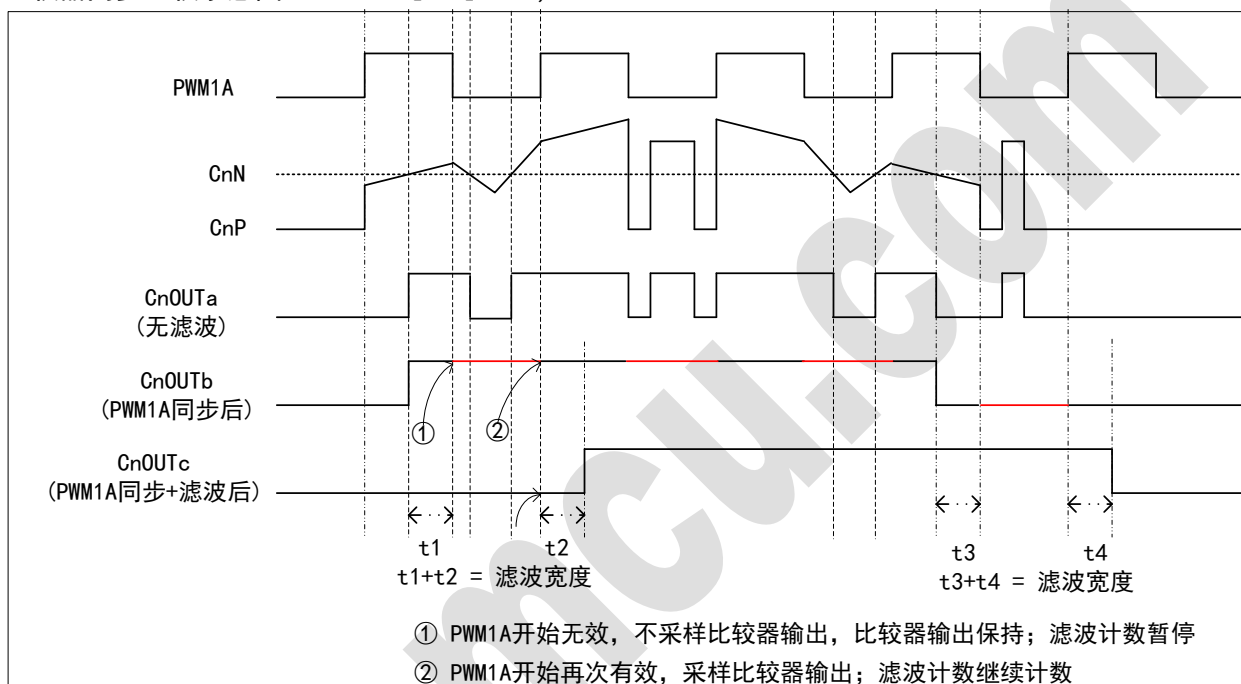
14.3.6 PWM 同步比较

配置 CxPPWMTRG 位，可开启比较器同步比较功能，四种模式：连续比较、PWM1A 同步比较、PWM1B 同步比较、PWM1C 同步比较；

配置 CxTRGPOL 位，选择对应的 PWMT 同步极性，“0”为 PWM 输出有效期间比较，“1”PWM 输出无效期间比较。

同步比较功能，可以和 PWM1A/PWM1B/PWM1C 输出同步，如果 CxTRGPOL 设为 0，那么只有当 PWM1 输出为有效电平期间，比较器的输出结果才被采样，给到后级滤波电路，在 PWM1 输出无效期间，不采样比较器输出，此时比较器的输出会保持，同时滤波器也停止计数，当 PWM1 输出再次有效时，恢复对比较器的采样和滤波计数。参见下图

比较器同步比较示意图 (PWMTRGS[1:0]= 01, TRGPOL = 0)



上图中的 CnOUTa 信号线为比较器不滤波也不和 PWM1A 信号同步时的 CnOUT；

上图中的 CnOUTb 信号线为比较器不滤波但是和 PWM1A 信号同步时的 CnOUT；

上图中的 CnOUTc 信号线为先和 PWM1A 同步再经滤波后的 CnOUT。

14.4 CMP 相关寄存器

CMP 寄存器汇总表

地址 (SFRn /H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
SFR0/ A4	CMP0CR0	CMP0 控制寄存 器 0	COEN	COTRGPO L	COFILT2	CORFSEL	COSMT1	COSMT0	COFILT1	COFILT0	0000 000
SFR0/ A5	CMP0CR1	CMP0 控制寄存 器 1	COINV	COIE	COINSEL 1	COINSEL 0	COIS1	COIS0	COOUT	COIF	0-00 0000
SFR0/ A6	CMP0CR2	CMP0 控制寄存 器 2	CONSEL3	CONSEL2	CONSEL1	CONSEL0	COPSEL3	COPSEL2	COPSEL1	COPSEL0	00000 000
SFR0/ A7	CMP0CR3	CMP0 控制寄存 器 3	COPWMTR GS1	COPWMTR GS0	CODAC5	CODAC4	CODAC3	CODAC2	CODAC1	CODAC0	0000 0000
SFR0/ C4	CMP1CR0	CMP1 控制寄存 器 0	C1EN	C1TRGPOL	C1FILT2	C1RFSEL	C1SMT1	C1SMT0	C1FILT1	C1FILT 0	00000 000
SFR0/ C5	CMP1CR1	CMP1 控制寄存 器 1	C1INV	C1IE	C1INSEL 1	C1INSEL 0	C1IS1	C1IS0	C1OUT	C1IF	0-00 0000
SFR0/ C6	CMP1CR2	CMP1 控制寄存 器 2	C1INSEL3	C1INSEL2	C1INSEL1	C1INSEL0	C1PSEL3	C1PSEL2	C1PSEL1	C1PSEL0	00000 000



SFR0/ C7	CMP1CR3	CMP1 控制寄存 器 3	C1PWMTR GS1	C1PWMTR GS0	C1DAC5	C1DAC4	C1DAC3	C1DAC2	C1DAC1	C1DAC0	0000 0000
-------------	---------	------------------	----------------	----------------	--------	--------	--------	--------	--------	--------	--------------

14.4.1 CMPn 控制寄存器 0 (CMPOCR0, 0xA4/SFR0, CMP1CR0, 0xC4/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPOCR0	COEN	COTRGPOL	COFILT2	CORFSEL	COSMT1	COSMT0	COFILT1	COFILT0
CMP1CR0	C1EN	C1TRGPOL	C1FILT2	C1RFSEL	C1SMT1	C1SMT0	C1FILT1	C1FILT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **CnEN** - CMPn 使能控制, n=0, 1

0: 比较器关闭; 比较器输出 CnOUT 固定输出低, 对应的复用端口输出, 由 IO 寄存器控制。

1: 比较器开启。

BIT[6] **CnTRGPOL** - CMPn 同步比较功能控制, n=0, 1, 此位仅在 CxPPWMTRG[1:0] != 00b 时有效

0: 仅在 PWM 输出有效期间比较。

1: 仅在 PWM 输出无效期间比较。

BIT[4] **CnRFSEL** - CMPn 反馈电阻选择, n=0, 1

0: 比较器 n 反馈电阻 100k 断开;

1: 比较器 n 反馈电阻 100k 接入电路。

BIT[3:2] **CnSMT[1:0]** - CMPn 施密特窗口选择, n=0, 1

00: 施密特窗口 1, 关闭施密特

01: 施密特窗口 2

10: 施密特窗口 3

11: 施密特窗口 4。

注: 详细窗口参数, 参见电气参数章节。

BIT[5&1:0] **CnFILT[2:0]** - CMPn 输出滤波选择, n=0, 1

若系统时钟选择高频 (CLKS=0), 滤波时钟源 HIRC/2, 如下

000: 无滤波

001: 滤波时间常数为 1us

010: 滤波时间常数为 4us

011: 滤波时间常数为 16us

100: 滤波时间常数为 250ns

101: 滤波时间常数为 500ns。

110~111: 保留 (不能设置)

若系统时钟选择为低频 (CLKS=1), 滤波时钟源为系统时钟

000: 无滤波

其他: 2 Fsys clk

注 1: 上述滤波常数时间不是精确值, 仅供参考

注 2: PWM 同步模式下, 必须开启滤波 (CnFILT[2:0] ≠ 000b), 否则同步无效。

14.4.2 CMPn 控制寄存器 1 (CMPOCR1, 0xA5/SFR0, CMP1CR1, 0xC5/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPOCR1	COINV	COIE	COINSEL1	COINSEL0	COIS1	COIS0	COOUT	COIF
CMP1CR1	C1INV	C1IE	C1INSEL1	C1INSEL0	C1IS1	C1IS0	C1OUT	C1IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **CnINV** - CMPn 输出反向控制, n=0, 1

0: 比较器输出正常, CnP>CnN, CnOUT 输出 1, 反之输出 0;

1: 比较器输出反向, CnP>CnN, CnOUT 输出 0, 反之输出 1。

BIT[6] **CnIE** - CMPn 中断使能控制, n=0, 1

0: 关闭;

1: 使能, CMPIE 同时为 1, 中断使能生效。

BIT[5:4] **CnINSEL[1:0]** - CMPn 输入模式选择, n=0, 1

00: 比较器 n 正相输入端由 CnPSEL[3:0] 选择, 比较器 n 反相输入端由 CnNSEL[3:0] 选择;

01: 比较器 n 正相输入端由 CnPSEL[3:0] 选择, 比较器 n 反相输入端接 GND, DAC 参考无效;

10: 比较器 n 正相输入端接 DAC 输出, 比较器 n 反相输入端由 CnNSEL[3:0] 选择, DAC 的参考源由由 CnPSEL[3:0] 选择

11: 比较器 n 正相输入端由 CnPSEL[3:0] 选择, 比较器 n 反相输入端接 DAC 输出, DAC 的参考源由由 CnNSEL[3:0] 选择

BIT[3:2] **CnIS [1:0]** - CMP0/1 中断模式选择位

00: 不触发中断标志



- 01: 下降沿触发, 比较器输出由高变低时触发中断标志
 10: 上升沿触发, 比较器输出由低变高时触发中断标志
 11: 双沿触发, 比较器输出由高变低时和由低变高时都触发中断标志。
- BIT[1] **CnOUT** - CMP0/1 输出状态标志位 (表示经过滤波器以后的状态, 只读)
 0: 比较器输出低;
 1: 比较器输出高。
- BIT[0] **CnIF** - CMP0/1 中断标志位, 硬件置 1, 软件清 0
 0: 无中断标志产生;
 1: 产生中断标志。

14.4.3 CMPn 控制寄存器 2 (CMP0CR2, 0xA6/SFR0, CMP1CR2, 0xC6/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP0CR2	CONSEL3	CONSEL2	CONSEL1	CONSEL0	COPSEL3	COPSEL2	COPSEL1	COPSEL0
CMP1CR2	C1NSEL3	C1NSEL2	C1NSEL1	C1NSEL0	C1PSEL3	C1PSEL2	C1PSEL1	C1PSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:4] **CnNSEL[3:0]** - CMPn 反相多路输入选择, n=0, 1

BIT[3:0] **CnPSEL[3:0]** - CMPn 正相多路输入选择, n=0, 1

详情参看<输入通道选择>章节的<CMP0 多路输入选择列表>和<CMP1 多路输入选择列表>

14.4.4 CMPn 控制寄存器 3 (CMP0CR3, 0xA7/SFR0, CMP1CR3, 0xC7/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP0CR3	COPWMTRGS1	COPWMTRGS0	CODAC5	CODAC4	CODAC3	CODAC2	CODAC1	CODAC0
CMP1CR3	C1PWMTRGS1	C1PWMTRGS0	C1DAC5	C1DAC4	C1DAC3	C1DAC2	C1DAC1	C1DAC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:6] **CnPWMTRGS1 [1:0]** - CMPn 同步比较控制位, n=0, 1

00: 连续比较模式, 无同步比较功能

01: 和 PWM1A 输出同步

10: 和 PWM1B 输出同步

11: 和 PWM1C 输出同步

注: 此为设置为同步模式 (01b/10b/11b) 时, 滤波控制 **CnFILT[2:0]** 不可以设置为不滤波 (000b)。

BIT[5:0] **CnDAC[5:0]** - CMPn 内建 DAC 输出电压选择, n=0, 1

$DACOUT = VREF * (CnDAC[5:0] + 1/64)$

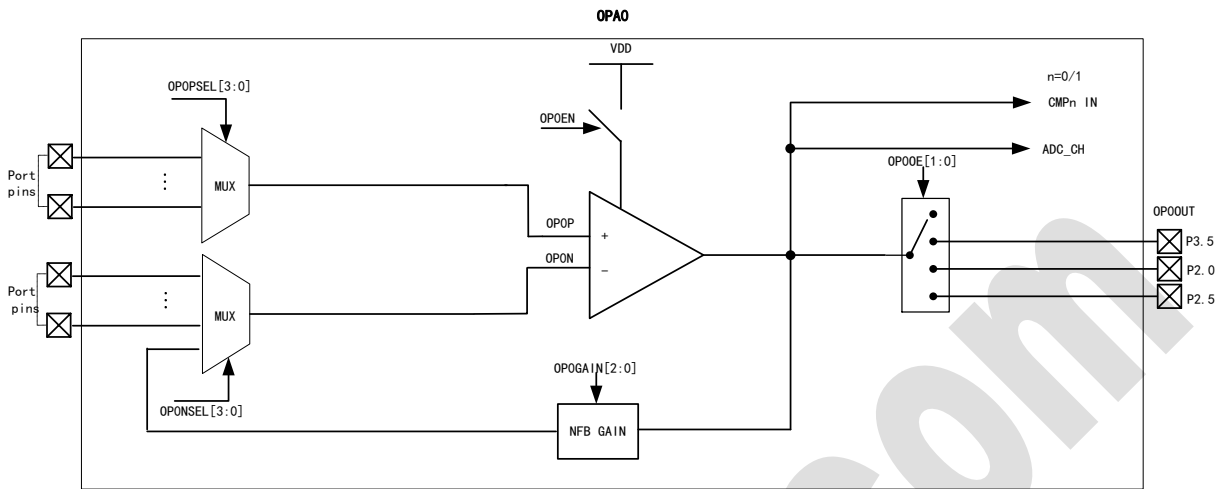
VREF 由 CnINSEL、CnNSEL[3:0] 和 CnPSEL[3:0] 组合控制决定



15 运算放大器

15.1 概述

片上包含 1 个独立模拟运算放大器，内部多档可调放大倍数（负反馈增益）。



15.2 特性

- ✧ 正/反向输入支持高达 7 路外部端口输入
- ✧ 内部固定多挡倍数放大可选
- ✧ 内部输出直接作为 ADC 内部通道进行采样

15.3 功能描述

15.3.1 输入通道选择

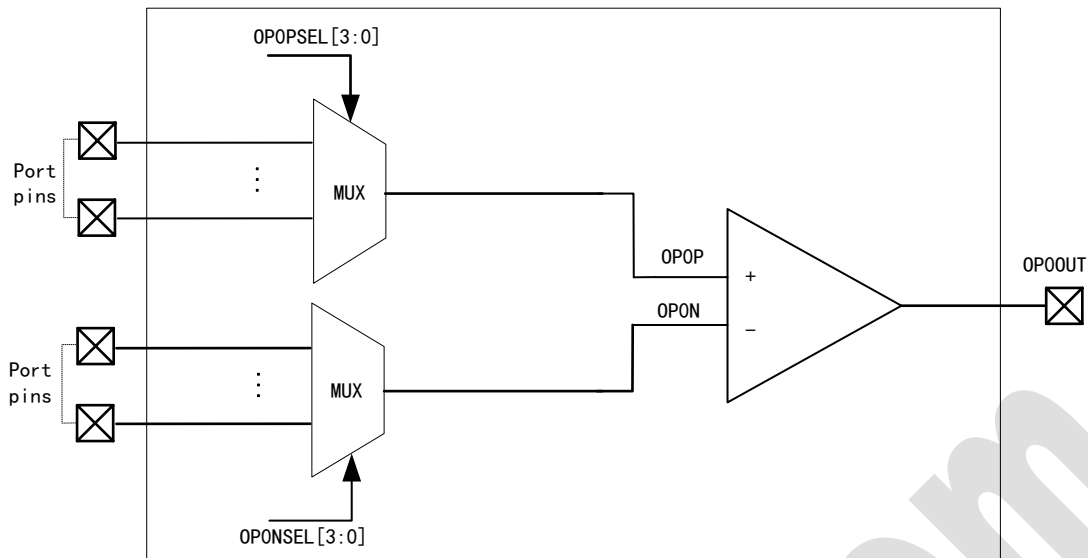
放大器正反相输入内置 MUX 多路选择电路，可以灵活选择不同的通道。
注：选择作为放大器输入的任何端口引脚都应该配置为模拟输入。

OP0 多路输入选择列表

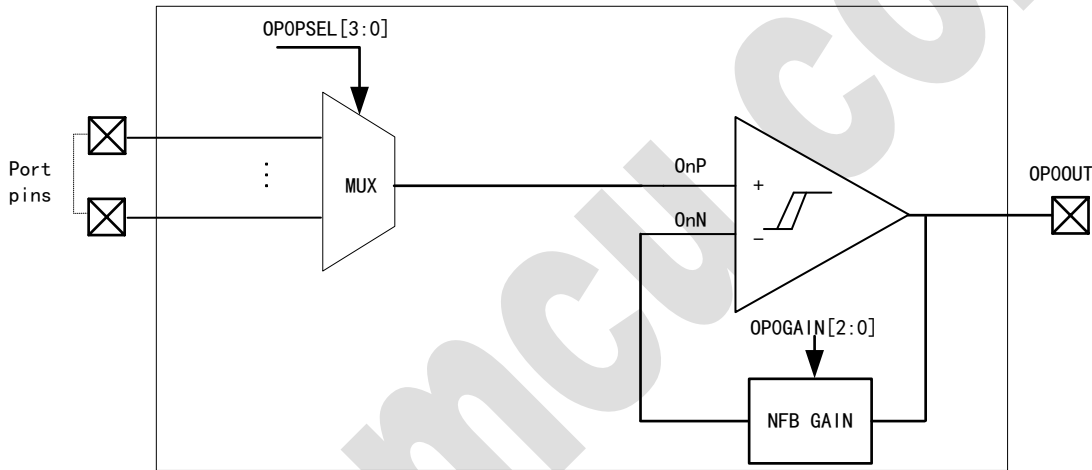
OPAOCR1->OPOPSEL	通道对应	管脚名称	OPAOCR1->OPONSEL	通道对应	管脚名称
0000	OP00	P3.2	0000	OP00	P3.2
0001	OP01	P3.1	0001	OP01	P3.1
0010	OP02	P2.7	0010	OP02	P2.7
0011	OP03	P2.6	0011	OP03	P2.6
0100	OP04	P2.4	0100	OP04	P2.4
0101	OP05	P2.3	0101	OP05	P2.3
0110	OP06	P2.2	0110	OP06	P2.2
0111	OP07	P2.1	0111	OP07	P2.1
1000	-	-	1000	GAIN（内部）	-
1001~1111	-	-	1001~1111	-	-

15.3.2 输入模式

OPAOCR1 寄存器中的 OPONSEL 位，可以选择输入通道连接至内部增益 GAIN 环路或外部通道，组合为 2 种结构模式。
OPONSEL = 0000b~0111b



OPONSEL = 1000b



15.3.3 输出控制

放大器输出可通过 OnOE[1:0]的设置连接到不同的 port 口，同时内部连接到 ADC 内部通道和比较器的输入通道多路复选器。

端口多路输出选择

OPAOCR0-> OnOE[1:0]	内部通道连接	管脚名称
00	不连接至 port 口	无
01	OP0OUT 连接至	P2.0
10	OP0OUT 连接至	P2.5
11	OP0OUT 连接至	P3.5

15.4 OP 相关寄存器

OP 寄存器汇总表

地址 (SFRn /H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
XSFR/ FF70	OPAOCR0	OP0 控制寄存器 0	OPOEN	-	OP0OE1	OP0OE0	-	OPOGAIN 2	OPOGAIN 1	OPOGAIN 0	0-00 -000
XSFR/ FF71	OPAOCR1	OP0 控制寄存器 1	OPONSEL 3	OPONSEL 2	OPONSEL 1	OPONSEL 0	OPOPSEL 3	OPOPSEL 2	OPOPSEL 1	OPOPSEL 0	0000 0000



15.4.1 OP0 控制寄存器 0 (OPAOCR0, 0xFF70/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPAOCR0	OPOEN	–	OP0OE1	OP0OE0	–	OPOGAIN2	OPOGAIN1	OPOGAIN0
R/W	R/W	–	R/W	R/W	–	R/W	R/W	R/W
初始值	0	–	0	0	–	0	0	0

BIT[7] **OPOEN** – OP0 使能控制

0: 放大器关闭; 放大器输出 OP0OUT 内部弱下拉。

1: 放大器开启。

BIT[5:4] **OP0OE[1:0]** – OP0OUT 输出至 PORT 口选择

00: OP0OUT 放大器输出不连接至 PORT 口;

01: OP0OUT 放大器输出连接至 P2.0

10: OP0OUT 放大器输出连接至 P2.5

11: OP0OUT 放大器输出连接至 P3.5

BIT[2:0] **OPOGAIN [2:0]** – OP0 内部增益放大倍数选择

000: 1 倍增益;

001: 2 倍增益;

010: 4 倍增益;

011: 8 倍增益;

100: 16 倍增益;

101: 32 倍增益;

110: 64 倍增益;

111: 64 倍增益。

15.4.2 OP0 控制寄存器 1 (OPAOCR1, 0xFF71/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPAOCR1	OPONSEL3	OPONSEL2	OPONSEL1	OPONSEL0	OPOPSEL3	OPOPSEL2	OPOPSEL1	OPOPSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:4] **OPONSEL[3:0]** – OP0 反相多路输入选择

BIT[3:0] **OPOPSEL[3:0]** – OP0 正相多路输入选择

详情参看<输入通道选择>章节的<OP0 多路输入选择列表>



16 中断

芯片的中断包括外部中断 INTOX（有 5 个通道共用该中断源）、INT1X（有 5 个通道共用该中断源）、定时器中断（T0、T1、T2、T3）、UART0 中断、ADC 转换中断、PWM0 中断、PWM1 中断、UART1 中断、SPI 中断、PWM1 刹车保护中断（PWM1FB）、CRC 中断共 15 个可用中断源，分为 4 个中断优先级。

每个中断均具有独立的中断标志位，中断使能位、中断向量和优先级设置位，还可以通过 1 个全局中断使能位将所有中断同时禁止。

16.1 中断源列表

芯片的中断源及其中断向量、相关控制位等列表如下：

中断源	触发条件	中断标志	中断使能位	优先级设置位	中断向量	中断号 (C51)	查询优先级	中断标志清除方式	停止模式唤醒	休眠模式唤醒
Reset	-	-	-	-	0000H	-	0(最高)	-	Yes	Yes
INTOX	上升沿/下降沿/双沿	EXT0IF0~EXT0IF7	EX0IE	EX0IP	0003H	0	1	软件清除	Yes	Yes
T0	T0 溢出	TF0	T0IE	T0IP	000BH	1	2	硬件自动	Yes	No
INT1X	上升沿/下降沿/双沿	EXT1IF0~EXT1IF7	EX1IE	EX1IP	0013H	2	3	软件清除	Yes	Yes
T1	T1 溢出	TF1	T1IE	T1IP	001BH	3	4	硬件自动	Yes	No
UART0	UART0 收/发完成	RI0/TI0	UART0IE	UART0IP	0023H	4	5	软件清除	Yes	No
SPI	SPI 收发完成/模式故障	SPIIF/SSDIS=0 且 MODF=1	SPIIE	SPIIP	002BH	5	6	软件清除	Yes	No
ADC	ADC 转换完成	ADCIF	ADCIE	ADCIP	0033H	6	7	软件清除	Yes	No
T2	T2 溢出、比较/T2 重载、捕获	TF2/ EXTf2	T2IE	T2IP	003BH	7	8	软件清除	Yes	No
T3	T3 溢出	TF3	T3IE	T3IP	0043H	8	9	硬件自动	Yes	Yes
PWM1	PWM1 溢出	PWMP1IF	PWM1IE+PWMP1IE	PWM1IP	004BH	9	10	软件清除	Yes	No
	PWM1 归零	PWMZ1IF	PWM1IE+PWMZ1IE							
	PWM1C 通道占向下计数匹配	PWMDDC1IF	PWM1IE+PWMDDC1IE							
	PWM1C 通道占向上计数匹配	PWMUDC1IF	PWM1IE+PWMUDC1IE							
	PWM1B 通道占向下计数匹配	PWMDDB1IF	PWM1IE+PWMDDB1IE							
	PWM1B 通道占向上计数匹配	PWMUDB1IF	PWM1IE+PWMUDB1IE							
	PWM1A 通道占向下计数匹配	PWMDDA1IF	PWM1IE+PWMDDA1IE							
	PWM1A 通道占向上计数匹配	PWMUDA1IF	PWM1IE+PWMUDA1IE							
UART1	UART1 收/发完成	RI1/TI1	UART1IE	UART1IP	0053H	10	11	软件清除	Yes	No
PWM0	PWM0 溢出	PWMO1IF	PWMO1IE + PWMO1IE	PWMO1IP	005BH	11	12	硬件自动	Yes	No
PWM1FB	PWM1FB 上的高/低电平	FBSTA	PWM1FBIE	PWM1FBIP	0063H	12	13	软件清除	Yes	No
CMP0/1	CMP 输出沿	CO1F/C11F	CMPIE	CMPIP	006BH	13	14	软件清除	Yes	No
CRC 注 2	CRC 校验完成	CRCIF	CSS_CRCIE+CRCIE	CRCIP	0073H	14	15(最低)	软件清除	No	No
CSS	CSS 检测时钟异常	CSS1F	CSS_CRCIE+CSSIE						YES 注 3	NO

注 1: CRCIF 和 CSS1F, 复用中断源。

注 2: CRC 中断不能唤醒停止模式, CSS 中断可以唤醒停止模式。

16.2 中断优先级

芯片的中断具有 4 个中断优先级，中断源的中断请求可设定为高优先级或低优先级，通过优先级的设定可实现中断服务程序的 4 级嵌套。



低优先级的中断服务程序能被高优先级请求所中断，反之则不行；中断服务程序不能被同一级的中断请求所中断。当多个同一优先级的中断同时产生中断请求时，根据预定的查询优先级顺序响应。

16.3 中断处理流程

当一个中断产生并且被芯片响应，则主程序运行被中断，将执行下述操作：

1. 当前正在执行的指令执行完；
2. PC 值被压入堆栈，保护现场；
3. 中断向量地址载入程序计数器 PC；
4. 执行相应的中断服务程序；
5. 中断服务程序结束并 RETI；
6. 将 PC 值出栈，并返回执行中断前的程序继续执行。

在此过程中，系统不会立即执行其它同一优先级的中断，但会保留所发生的中断请求，在当前中断处理结束后，再执行一条指令，之后转去执行新的中断请求。

当用户开启优先级中断，高优先级中断可以打断低优先级的中断，本芯片可以实现 2 级中断嵌套。

16.4 外部中断

本芯片有 2 个外部中断源 INT0 和 INT1，每个中断源分别有 8 路外部输入，对应端口 INT_{xy} ($xy = 0-1; y = 0-7$)。外部中断为边沿触发，配置控制位 ExnIS[1:0] ($n=0-1$) 选择不同的触发沿（不触发/上升沿/下降沿/双沿）。用户可以通过 EXTEN/ EXTEN1 寄存器灵活地在 5 个口选择一个或多个端口作为输入，端口 INT0_y ($y=0-7$) 复用同一个中断源 INT0，端口 INT1_y ($y=0-7$) 复用同一个中断源 INT1。

若设置触发沿为下降沿，一个周期内引脚输入连续采样为高电平，下个周期采样为低电平，则对应中断标志置起，发出中断请求；若多个中断请求同时发生，同一个中断源只响应一个中断请求，用户可以在中断处理程序中进行标志判断区分；中断标志需要软件清零，未清零标志，硬件将会在中断退出后再次发出中断请求。

同一个中断源的 8 路输入，当其中 1 路触发已进入中断处理程序，不影响新的中断标志建立，只是不会响应新的中断请求。

注：外部中断输入通道内建最小 100ns 模拟滤波，高温低压条件下滤波偏大（不超过 500ns）。外部中断沿信号的高电平必须大于（1 个系统时钟周期+模拟滤波时间），才能保证有效的触发。

EXTIS2 寄存器的 EXT0A 位为 INT0 的总使能开关，EXT1A 位为 INT1 的总使能开关。

16.4.1 外部中断使能寄存器 (EXTEN, 0xB5/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTEN	EXTOEN7	EXTOEN6	EXTOEN5	EXTOEN4	EXTOEN3	EXTOEN2	EXTOEN1	EXTOEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

16.4.2 外部中断使能寄存器 1 (EXTEN1, 0xB4/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTEN1	EXT1EN7	EXT1EN6	EXT1EN5	EXT1EN4	EXT1EN3	EXT1EN2	EXT1EN1	EXT1EN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7~0] EXTnENm ($n = 0-1; m = 0-7$) - 外部中断 INT_{xy} ($x = 0-1; y = 0-7$) 端口使能位

0: 禁止对应某一路 INT_{xy} ($x = 0-1; y = 0-7$) 端口的复用功能，端口用于通用 I/O 或其他复用模块；

1: 使能对应某一路 INT_{xy} ($x = 0-1; y = 0-7$) 端口的复用功能，端口作为外部中断输入用。

注 1: 要使能对应的 INT_{xy} ($x = 0-1; y = 0-7$) 端口，还必须先使能 EXTnA ($n = 0-1$) 位。中断源 INT_x ($x = 0-1$)，输入信号有效后将产生对应的 EXTnIFm ($n = 0-1; m = 0-7$) 标志位置 1。

注 2: 当使用 T0/T1 测量电平宽度时，T0/T1 不能用作他用，且只通过 INT04/ INT14，即这两个外设联动工作，资源同时被使用。此时 INT04 使用 T0，计算的方式可对应 T0 的方式 0-3，四种方式；而 INT14 使用 T1，但计算的方式可对应 T1 的方式 0-2，仅三种方式。

16.4.3 外部中断控制寄存器 2 (EXTIS2, 0xAA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTIS2	EXT0A	EXT1A	-	-	EXT1IS1	EXT1IS0	EXT0IS1	EXT0IS0
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
复位值	0	0	-	-	0	0	0	0

BIT[7], BIT[6] EXTnA ($n = 0-1$) - 外部中断 INT_{xy} ($x = 0, 1; y = 0-7$) 总使能位

0: 禁止所有物理 INT_{xy} ($x = 0-1; y = 0-4$) 端口上外部中断功能，即关闭 INT_x 模块；

1: 使能所有物理 INT_x ($x = 0-1; y = 0-4$) 端口上外部中断功能，即打开 INT_x 模块。

BIT[3:2] EXT1IS[1:0] - 外部中断 INT1_y ($y = 0-7$) 触发方式选择位



BIT[1:0] EXT0IS[1:0] - 外部中断 INT0y (y = 0~7) 触发方式选择位

EX0IS[1:0]/ EX1IS[1:0]	INTxy (x = 0, 1) 的端口触发方式
00	不触发
01	下降沿触发
10	上升沿触发
11	双沿触发

注：同一个中断源 INTx (x = 0~1) 而言，对 INTxy (x = 0, 1; y = 0~7) 这 8 个端口将使用共同的触发方式，不可单独设置。

EXnIS[1:0]	在使用 T0/T1 脉冲宽度测量方式下，INT04/INT14 的端口触发方式
XX	始终有效（用于测量高电平的脉冲宽度）

注：在使用 T0/T1 脉冲宽度测量方式下，仅 INT04 和 INT14 两个端口可用，且 EXnIS[1:0] 控制位不参与控制，此时可用于测量高电平的脉冲宽度；在 INT04 对应于 T0 的资源，INT14 则对应于 T1 的资源。

16.4.4 外部中断控制寄存器 1 (EXTIS1, 0xAB/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTIS1	EXT1IF7	EXT1IF6	EXT1IF5	EXT1IF4	EXT1IF3	EXT1IF2	EXT1IF1	EXT1IF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

16.4.5 外部中断控制寄存器 0 (EXTIS, 0xAC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTIS	EXT0IF7	EXT0IF6	EXT0IF5	EXT0IF4	EXT0IF3	EXT0IF2	EXT0IF1	EXT0IF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7~0] EXTnIFm (n = 0~1; m = 0~7) - 外部中断 INTxy (x = 0, 1; y = 0~7) 标志位

0: 在对应物理 INTxy (x = 0, 1; y = 0~7) 端口上没有外部中断产生，由软件清 0；

1: 在对应物理 INTxy (x = 0, 1; y = 0~7) 端口上有外部中断产生，由硬件置 1，若同时 EXnIE (n = 0, 1) 位置位，同时将产生 INTx (x = 0, 1) 中断，则芯片会响应该中断请求。

注 1: 对 EXT0IFm (m = 0~7)，对应于 INT0 中断源；EXT1IFm (m = 0~7)，对应于 INT1 中断源；每个物理端口标志独立建立，但每个中断源的 8 个子中断口并不会嵌套，同一个中断源不会同时响应多个中断请求。

注 2: 这些标志位置 1 后需要软件清 0。

16.5 中断相关寄存器

中断寄存器汇总表

地址 (SFR n/H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR0/BC	IPH1	中断优先级控制 1 高位寄存器	CRCIPH	CMPIPH	PWM1FBI PH	PWMOIPH	UART1IP H	PWM1IPH	T3IPH	T2IPH	0000 0000
SFR0/BB	IP1	中断优先级控制 1 低位寄存器	CRCIPL	CMPIPL	PWM1FBI IPL	PWMOIPL	UART1IP L	PWM1IPL	T3IPL	T2IPL	00000 000
SFR0/BA	IPH	中断优先级控制高位寄存器	-	ADCIPH	SPIIPH	UART0IP H	T1IPH	EXT1IPH	TOIPH	EXT0IPH	-000 0000
SFR0/B9	IP	中断优先级控制低位寄存器	-	ADCIPL	SPIIPL	UART0IP L	T1IPL	EXT1IPL	TOIPL	EXT0IPL	-000 0000
SFR0/B5	EXTEN	外部中断使能寄存器	EXT0EN7	EXT0EN6	EXT0EN5	EXT0EN4	EXT0EN3	EXT0EN2	EXT0EN1	EXT0EN0	-000 00-0
SFR0/B4	EXTEN1	外部中断使能寄存器 1	EXT1EN7	EXT1EN6	EXT1EN5	EXT1EN4	EXT1EN3	EXT1EN2	EXT1EN1	EXT1EN0	-000 00-0
SFR0/AC	EXTIS	外部中断控制寄存器	EXT0IF7	EXT0IF6	EXT0IF5	EXT0IF4	EXT0IF3	EXT0IF2	EXT0IF1	EXT0IF0	-000 00--
SFR0/AB	EXTIS1	外部中断控制寄存器 1	EXT1IF7	EXT1IF6	EXT1IF5	EXT1IF4	EXT1IF3	EXT1IF2	EXT1IF1	EXT1IF0	-000 00--
SFR0/AA	EXTIS2	外部中断控制寄存器 2	EXT1A	EXT0A	-	-	EXT1IS1	EXT1IS0	EXT0IS1	EXT0IS0	----- 0000
SFR0/	IE1	中断使能寄存	CSS_CRC	CMPIE	PWM1FBI	PWMOIE	UART1IE	PWM1IE	T3IE	T2IE	00000



A9		器 1	IE		E						000
SFR0/A8	IE	中断使能寄存器	EA	ADCIE	SPIIE	UART0IE	T1IE	EXT1IE	TOIE	EX0IE	00000 000

16.5.1 中断使能寄存器 (IE, 0xA8/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE	EA	ADCIE	SPIIE	UART0IE	T1IE	EXT1IE	TOIE	EXT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] EA - 总中断使能位

0: 关闭所有中断;

1: 打开中断, 但具体中断使能控制由相应的控制位决定。

BIT[6] ADCIE - ADC 中断使能位

0: 关闭 ADC 中断;

1: 打开 ADC 中断。

BIT[5] SPIIE - SPI 中断允许控制位

0: 禁止 SPI 中断;

1: 允许 SPI 中断, 当标志位 SPIIF=1 直接产生正常中断或者 SSDIS=0 且 MODF=1 时将产生 SPI 错误中断。

BIT[4] UART0IE - UART0 中断使能位

0: 关闭 UART0 中断;

1: 打开 UART0 中断。

BIT[3] T1IE - T1 中断使能位

0: 关闭 T1 中断;

1: 打开 T1 中断。

BIT[2] EXT1IE - INT1 中断使能位

0: 关闭 INT1 中断;

1: 打开 INT1 中断。

BIT[1] TOIE - TO 中断使能位

0: 关闭 TO 中断;

1: 打开 TO 中断。

BIT[0] EXT0IE - INTO 中断使能位

0: 关闭 INTO 中断;

1: 打开 INTO 中断。

16.5.2 中断使能寄存器 1 (IE1, 0xA9/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE1	CSS_CRCIE	CMP1IE	PWM1FBIE	PWMOIE	UART1IE	PWM1IE	T3IE	T2IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7] CSS_CRCIE - CRC 和 CSS 中断使能位

0: 禁止 CRC 中断, 禁止 CSS 中断;

1: 允许 CRC 中断, 允许 CSS 中断。

注: 此控制位与 IE2 的 CSSIE 和 CRCIE 联动控制, 若要使能 CRC 中断, 必须同时置位 CSS_CRCIE 和 CRCIE; 若要禁止 CRC 中断, 仅需清零 CSS_CRCIE 和 CRCIE 任意一位。CSS 同理。CSS 和 CRC 复用同一个中断入口。

BIT[6] CMP1IE - CMP0/1 中断使能位

0: 禁止 CMP 中断;

1: 允许 CMP 中断。

BIT[5] PWM1FBIE - PWM1FB 中断使能位

0: 禁止 PWM1FB 中断;

1: 打开 PWM1FB 中断。

BIT[4] PWMOIE - PWMO 中断使能位

0: 禁止 PWMO 中断;

1: 打开 PWMO 中断。

注: 必须与 PWMO 模块的中断使能同时使能, 中断使能有效

BIT[3] UART1IE - UART1 中断使能位

0: 关闭 UART1 中断;

1: 打开 UART1 中断。

BIT[2] PWM1IE - PWM1 总中断使能位

0: 禁止 PWM1 总中断;



1: 打开 PWM1 总中断。
BIT[1] **T3IE** - T3 中断使能位
0: 关闭 T3 中断;
1: 打开 T3 中断。
BIT[0] **T2IE** - T2 中断使能位
0: 关闭 T2 中断;
1: 打开 T2 中断。

注: 该中断源可同时受 TF2 和 EXTF2 两个中断标志位影响, 但原则上 T2 的这两个功能不可同时使用。

16.5.3 中断优先级控制低位寄存器 (IP, 0xB9/SFR0) 和控制高位寄存器 (IPH, 0xBA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP	-	ADCIPL	SPIIPL	UARTOIPL	T1IPL	EX1IPL	TOIPL	EXOIPL
IPH	-	ADCIPH	SPIIPH	UARTOIPH	T1IPH	EX1IPH	TOIPH	EXOIPH
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	0	0	0	0	0	0	0

16.5.4 中断优先级控制 1 低位寄存器 (IP1, 0xBB/SFR0) 和控制 1 高位寄存器 (IPH1, 0xBC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP1	CRCIPL	CMPIPL	PWM1FB IPL	PWM0IPL	UART1IPL	PWM1IPL	T3IPL	T2IPL
IPH1	CRCIPH	CMPIPH	PWM1FB IPH	PWM0IPH	UART1IPH	PWM1IPH	T3IPH	T2IPH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

优先级等级配置对照表

IPH/IPH1	IP/IP1	中断优先级等级
0	0	等级 0 (低)
0	1	等级 1
1	0	等级 2
1	1	等级 3 (高)

注: “高优先级中断”可以打断正在执行的“低优先级中断”, 实现多层中断嵌套; 在同一优先级情况下则按其“中断源列表”上的顺序排列。特别注意, 一个中断产生后, 若用户不进行处理, 相同优先级的任何新中断将不会再产生。



17 其他辅助功能

17.1 双 DPTR

传统 8051 架构仅有一组 DPTR（数据指针寄存器）。DPTR 本质是一个 16 位的特殊功能寄存器，其高位字节寄存器用 DPH 表示，低位字节寄存器用 DPL 表示。

本芯片重新设计了架构，升级了内核，增加了另一组数据指针 DPTR1，其高位字节寄存器用 DPH1 表示，低位字节寄存器用 DPL1 表示。这样的做法，既可以兼容了老的 51 代码，软件还能利用提供的两组数据指针 DPTR 及 DPTR1，进一步加快程序的执行速度同时缩减代码量。

17.1.1 数据指针低字节寄存器（DPL, 0x82/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

17.1.2 数据指针高字节寄存器（DPH, 0x83/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

17.1.3 数据指针 1 低字节寄存器（DPL1, 0x84/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

17.1.4 数据指针 1 高字节寄存器（DPH1, 0x85/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

17.1.5 数据指针选择寄存器（DPS, 0xD8/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPS	—	SFRSLO	—	—	DIV16	MUL16	—	SEL
R/W	—	R/W	—	—	R/W	R/W	—	R/W
复位值	—	0	—	—	0	0	—	0

BIT[0] SEL - DPTR/DPTR1 选择控制位

0: 指令“MOVC/MOVX @DPTR”中的操作对象为 DPTR;

1: 指令“MOVC/MOVX @DPTR”中的操作对象为 DPTR1。

CORE 切换 DPTR，汇编代码应用举例：

```
.....//假设当前使用的是 DPTR
INC DPS;          //切换至 DPTR1
.....
INC DPS;          //再切换回 DPTR
```

17.2 循环冗余校验（CRC）

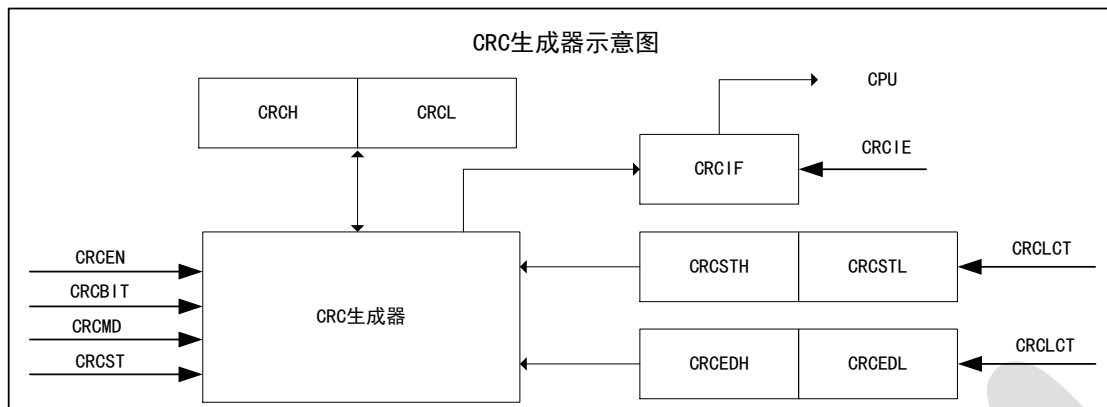
循环冗余校验（Cyclic Redundancy Check, CRC）是一种根据输入数据产生简短固定位数校验码的一种散列函数，主要用来提高系统的可靠性。

17.2.1 CRC 特性

集成硬件 16 位 CRC 单元，符合 CRC-16 标准，生成多项式为 $X_{16} + X_{15} + X_2 + 1$ ，即 0x8005，初值可设为任意值，



默认为 0xFFFF。



17.2.2 CRC 相关寄存器

CRC 寄存器汇总表

地址 (SFRn /H)	寄存器 名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位 值 (B)
XSFR/ FF60	CRCCR	CRC 控制寄存器	CRCEN	CRCIE	CRCIF	CRCST	CRCMD	CRCLCT	-	CRCENC	0000 00-0
XSFR/ FF62	CRCH	CRC 结果高字 节寄存器	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0	1111 1111
XSFR/ FF63	CRCL	CRC 结果低字 节寄存器	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0	1111 1111
XSFR/ FF64	CRCSTH	CRC 起始地址 高字节寄存器	CRCSTH7	CRCSTH6	CRCSTH5	CRCSTH4	CRCSTH3	CRCSTH2	CRCSTH1	CRCSTH0	0000 0000
XSFR/ FF65	CRCSTL	CRC 起始地址 低字节寄存器	CRCSTL7	CRCSTL6	CRCSTL5	CRCSTL4	CRCSTL3	CRCSTL2	CRCSTL1	CRCSTL0	0000 0000
XSFR/ FF66	CRCEDH	CRC 结束地址 高字节寄存器	CRCEDH7	CRCEDH6	CRCEDH5	CRCEDH4	CRCEDH3	CRCEDH2	CRCEDH1	CRCEDH0	1111 1111
XSFR/ FF67	CRCEDL	CRC 结束地址 低字节寄存器	CRCEDL7	CRCEDL6	CRCEDL5	CRCEDL4	CRCEDL3	CRCEDL2	CRCEDL1	CRCEDL0	1111 1111

17.2.3 CRC 控制寄存器 (CRCCR, 0xFF60/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCCR	CRCEN	CRCIE	CRCIF	CRCST	CRCMD	CRCLCT	-	CRCENC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
复位值	0	0	0	0	0	0	-	0

BIT[7] **CRCEN** - CRC 模块使能控制位

0: CRC 模块关闭;

1: CRC 模块打开。

BIT[6] **CRCIE** - CRC 中断使能位

0: 关闭;

1: 开启, 需同时置位 CSS_CRCIE 中断使能才生效。

注: 此控制位与 IE1 的 CSS_CRCIE 联动控制, 若要使能 CRC 中断, 必须同时置位 CSS_CRCIE 和 CRCIE; 若要禁止 CRC 中断, 仅需清零 CSS_CRCIE 和 CRCIE 任意一位。CSS 和 CRC 复用同一个中断入口。

BIT[5] **CRCIF** - CRC 中断标志位

0: 未启动或启动 CRC 校验后未完成, 由软件清 0;

1: 本次 CRC 校验计算完成, 由硬件置 1, 若同时 CRCIE 位置位, 将产生 CRC 中断, 则芯片会响应该中断请求。

BIT[4] **CRCST** - CRC 校验计算启动位

0: 本次 CRC 校验完成, 由硬件清 0;

1: 启动一次 CRC 校验或者处于 CRC 校验计算当中, 由软件置 1。

注: 写的话, 0 可以写入, 但无实际意义; 读的话, 该位根据 CRC 模块的实际状态返回 1/0。

BIT[3] **CRCMD** - CRC 校验模式选择位



0: 对整个 FLASH ROM 区域或者整个 EEPROM 区域做 CRC 校验;

1: 对指定起始地址和结束地址的区域做 CRC 校验。

注: 该位=0 时, 起始地址寄存器和结束地址寄存器设定无效; =1 时, 起始地址寄存器和结束地址寄存器设定有效。

BIT[2] CRCLCT - CRC 校验区域选择位

0: 对 FLASH ROM 区域做 CRC 校验;

1: 对 EEPROM 区域做 CRC 校验。

注: 这里 FLASH ROM 有效区域指 0x0000~0x3FFF 的地址范围; EEPROM 有效区域指 0x000~0x3FF 的地址范围。

BIT[0] CRCENC - CRC 加密相关控制位

0: 应用于 FLASH 无加密的状态, 对指定起始地址和结束地址的区域做 CRC 校验;

1: 应用于 FLASH 有加密的状态, 结束地址无效。

注: 对于 FLASH 不加密的区域 (CRCENC=0) 做 CRC 校验时, 用户可以指定任意的起始地址和结束地址; 对于 FLASH 加密的区域 (CRCENC=1) 做 CRC 校验时, 用户只要指定一个起始地址即可, 无需指定结束地址, 芯片会自动对当前地址所在的扇区做 CRC 校验, 并给出 CRC 计算结果, 此时结束地址并不参与本次 CRC 的运算。

17.2.4 CRC 结果低字节寄存器 (CRCL, 0xFF63/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCL	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

BIT[7:0] CRCL[7:0] - CRC 结果低字节

17.2.5 CRC 结果高字节寄存器 (CRCH, 0xFF62/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCH	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

BIT[7:0] CRCH[7:0] - CRC 结果高字节

注: CRC[15:0] 的内容可以在 CRC 校验启动前, 由用户写入任意值, 作为 CRC 初值, 默认为 0xFFFF; 其内容, 在 CRC 校验计算过程中, 不允许用户修改, 否则 CRC 结果将不可预知。

17.2.6 CRC 起始地址低字节寄存器 (CRCSTL, 0xFF65/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCSTL	CRCSTL7	CRCSTL6	CRCSTL5	CRCSTL4	CRCSTL3	CRCSTL2	CRCSTL1	CRCSTL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] CRCSTL[7:0] - CRC 起始地址低字节

17.2.7 CRC 起始地址高字节寄存器 (CRCSTH, 0xFF64/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCSTH	CRCSTH7	CRCSTH6	CRCSTH5	CRCSTH4	CRCSTH3	CRCSTH2	CRCSTH1	CRCSTH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] CRCSTH[7:0] - CRC 起始地址高字节

17.2.8 CRC 结束地址低字节寄存器 (CRCEDL, 0xFF67/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCEDL	CRCEDL7	CRCEDL6	CRCEDL5	CRCEDL4	CRCEDL3	CRCEDL2	CRCEDL1	CRCEDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

BIT[7:0] CRCEDL[7:0] - CRC 结束地址低字节

17.2.9 CRC 结束地址高字节寄存器 (CRCEDH, 0xFF66/XSFR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCEDH	CRCEDH7	CRCEDH6	CRCEDH5	CRCEDH4	CRCEDH3	CRCEDH2	CRCEDH1	CRCEDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1



BIT[7:0] CRCEDH[7:0] - CRC 结束地址高字节

注 1: 对起始地址和结束地址的设定, 当 $CRCLCT=1$ (EEPROM) 时必须满足 $0x000 \leq CRCST[15:0] \leq CRCED[15:0] \leq 0x3FF$ 的条件, 当 $CRCLCT=0$ (FLASH) 时必须满足 $0x0000 \leq CRCST[15:0] \leq CRCED[15:0] \leq 0x3FFF$ 的条件, 否则 CRC 结果将不可预知; 若 $CRCST[15:0]$ 和 $CRCED[15:0]$ 没有落在实际的物理地址上, CRC 结果也将不可预知。

注 2: 本芯片中, 对“指定加密 FLASH ROM 区域的地址范围”或者“包含加密 FLASH ROM 区域的地址范围”, 不能被 CRC 模块进行正确的 CRC 校验, 故最终得到的 CRC 结果将是一个异常值。因此, 当做 CRC 校验时, FLASH 区域当中全部或部分区域选择了“加密”时, 推荐做法有下面两种:

- ✧ 对整个 FLASH ROM 区域进行 CRC 校验, 即先设置 $CRCMD=0$, $CRCLCT=0$, $CRCENC=X$, 再做 CRC 运算。
- ✧ 对分别对扇区进行 CRC 校验, 通过指定 (起始地址落在的某个加密扇区的) 一个扇区进行 CRC 校验, 不用指定结束地址, 即先设置 $CRCMD=1$, $CRCLCT=0$, $CRCENC=1$, 再做 CRC 运算。

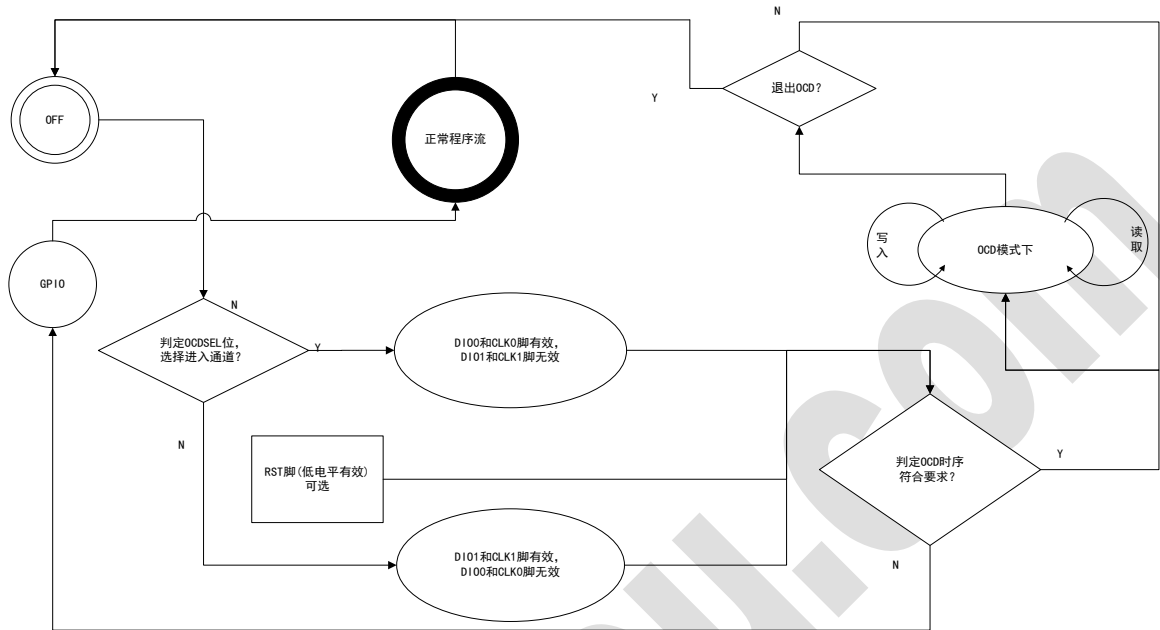
注 3: CRC 模块计算区域的大小有限制, 必须大于等于 4 个字节 (FLASH 是 4 字节; EEPROM 是 3 字节); 小于该字节数, CRC 计算结果将不可预知。



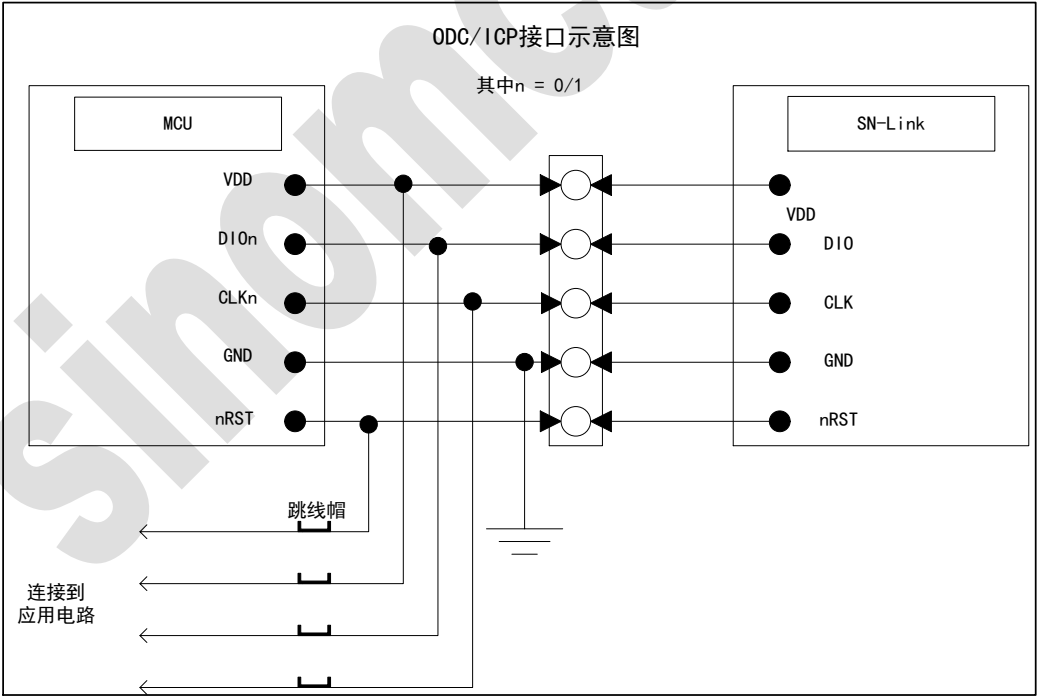
18 片上调试OCD和在电路编程ICP接口

本芯片芯片内嵌片上调试（OCD）功能，这为用户提供了低成本的调试方法，并且本芯片充分考虑了用户的需求，提供了精简的 2 线调试接口电路，以及自动二选一的两组调试实物接口（在用户配置字中设定，默认为第一组调试接口）。当一组连接后，另一组自动失效。在引脚复用的场合可以真正做到全仿真。

OCD状态机示意图



OCD/ICP接口示意图



片上调试 OCD 和在电路编程 ICP 接口整理

名称	对应 PIN	名称	对应 PIN	要求
电源正	VDD	电源正	VDD	必须
DIO1	P2.4	DIO0	P3.4	(二选一)
CLK1	P2.3	CLK0	P3.3	



				必须
nRST	P3.2	nRST	P3.2	可选
电源负	GND	电源负	GND	必须

注： DIO_n 和 CLK_n (n=0、1) 的选择与“用户”实际电路预留调试的接口有关，正常情况下选择其中一对使用即可。



19 IAP（片上编程）

片上 EEPROM-LIKE 数据存储器和 FLASH 存储器，支持用户程序在带电工作中实时地读出或写入数据。当然，用户可以通过相关的工具，由外部直接操作。在读数据时，利用 MOV_C 指令间接寻址，通过 EERD 标志位进行区分读的是 FLASH 区，还是 EEPROM-LIKE 区；而写入数据时，通过 EERD 标志位进行区分写的是 FLASH 区或 EEPROM-LIKE，通过操作地址寄存器 IAPAD、数据寄存器 IAPDB 进行。

EEPROM-LIKE 和 FLASH 存储器，执行擦除操作后，数据 bit 由 0 变为 1；若要修改数据 bit 由 1 变 0，必须执行擦除动作。

EEPROM-LIKE 有 128 字节共 1 页，独立 EEPROM 编址，支持页擦除/字节烧写操作，不支持字节擦除。

FLASH 空间大小 18K，128bytes/页，512bytes/块，支持页擦除/块擦除/字节烧写操作

数据寄存器 IAPDB 为数据锁存器，对 IAPDB 的“写”操作实际将数据写入数据锁存区的对应地址中。

19.1 IAP 相关寄存器

IAP 操作寄存器汇总表

地址 (SFR n/H)	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
SFR0/EF	IAPKEY	IAP 写保护寄存器	IAPKEY7	IAPKEY6	IAPKEY5	IAPKEY4	IAPKEY3	IAPKEY2	IAPKEY1	IAPKEY0	0000000
SFR0/EE	IAPMD	IAP 模式寄存器	IAPEN	-	-	BLOCKERS	-	PAGEERS	=	BYTEWR	0--0 -0-0
SFR0/ED	IAPCR1	IAP 控制寄存器 1	BOOT	MFREF	IAPERR	-	PCFEN	UCFEN	BLREN	MFREN	000-0000
SFR0/EC	IAPDB	IAP 数据寄存器	IAPDB7	IAPDB6	IAPDB5	IAPDB4	IAPDB3	IAPDB2	IAPDB1	IAPDB0	0000000
SFR0/EB	IAPADH	IAP 高地址寄存器	IAPAD15	IAPAD14	IAPAD13	IAPAD12	IAPAD11	IAPAD10	IAPAD9	IAPAD8	0000000
SFR0/EA	IAPADL	IAP 低地址寄存器	IAPAD7	IAPAD6	IAPAD5	IAPAD4	IAPAD3	IAPAD2	IAPAD1	IAPAD0	0000000
SFR0/E9	IAPCR	IAP 控制寄存器	IAPSTR	-	-	BLRS	IAPTRIG ₂	IAPTRIG ₁	IAPTRIG ₀	INFOS	0--0 0000

19.1.1 IAP 模式寄存器（IAPMD, 0xEE/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPMD	IAPEN	-	-	BLOCKERS	-	PAGEERS	-	BYTEWR
R/W	R/W	-	-	R/W	-	R/W	-	R/W
复位值	0	-	-	0	-	0	-	0

BIT[7] IAPEN - IAP 使能位

0: 关闭 IAP 功能（写一遍有效）；

1: 打开 IAP 功能（连续写两遍有效）。

注：只有对该位连续写 2 条相同的指令，才能使系统进行打开 IAP 使能位，否则下个时钟周期到来后，寄存器位中先前写入的值将被硬件自动清掉，保持为 0，这是一种简单的保护机制；关闭时写一次 0 即可。

BIT[4] BLOCKERS - 块擦除模式位

0: 无（写一遍有效）；

1: 块擦除模式（连续写两遍有效）。

注：仅对 flash 操作有效，对 EEPROM-LIKE 操作无效

BIT[2] PAGEERS - 页擦除模式位

0: 无（写一遍有效）；

1: 页擦除模式（连续写两遍有效）。

BIT[0] BYTEWR - 字节编程（写入）模式位

0: 无（写一遍有效）；

1: 字节编程模式（连续写两遍有效）。

注：特别注意，IAPMD 寄存器中的每一位置 1 的话，都需要连续写 2 次操作。另外，以上 3 种 EEPROM-LIKE/FLASH 的操作模式，在同一时刻，本芯片只允许进入一种模式执行，否则芯片会出现不可预知的情况。

19.1.2 IAP 写保护寄存器（IAPKEY, 0xEF/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	-------	-------	-------	-------	-------	-------	-------	-------



IAPKEY	IAPKEY7	IAPKEY6	IAPKEY5	IAPKEY4	IAPKEY3	IAPKEY2	IAPKEY1	IAPKEY0
R/W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] IAPKEY[7:0] - IAP 保护锁

IAP 保护用，当用户在启动 IAP 操作的时候，必须先向其写入 0x5A，紧接着写 0xA5，解除 IAP 保护，对 IAPCR 的操作才有效；否则本次操作视为无效。

注：IAPKEY 为只写寄存器，不可读取其值，其保护的保护对象为 IAPCR 寄存器中的 IAPSTR 位和 IAPTRIG[2:0] 位。

19.1.3 IAP 数据寄存器 (IAPDB, 0xEC/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPDB	IAPDB7	IAPDB6	IAPDB5	IAPDB4	IAPDB3	IAPDB2	IAPDB1	IAPDB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[7:0] IAPDB[7:0] - IAP 写入数据。

19.1.4 IAP 高地址寄存器 (IAPADH, 0xEB, SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPADH	IAPAD15	IAPAD14	IAPAD13	IAPAD12	IAPAD11	IAPAD10	IAPAD9	IAPAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[5:0] IAPAD[15:8] - IAP 写入地址的高 8 位。

19.1.5 IAP 低地址寄存器 (IAPADL, 0xEA/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPADL	IAPAD7	IAPAD6	IAPAD5	IAPAD4	IAPAD3	IAPAD2	IAPAD1	IAPAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

BIT[3:0] IAPAD[7:0] - IAP 写入地址的低 8 位。

注：EEPROM-LIKE 的容量为 128 bytes，可操作范围 0~0x007F 之间，对 IAPAD 寄存器 BIT[15:7] 位的操作是无效的。

注：FLASH 的容量 32K bytes，可操作范围 0~0x7FFF 之间，对 IAPAD 寄存器 BIT15 位的操作是无效的。

19.1.6 IAP 控制寄存器 (IAPCR, 0xE9/SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPCR	IAPSTR	-	-	BLRS	IAPTRIG2	IAPTRIG1	IAPTRIG0	INFOS
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
复位值	0	-	-	0	0	0	0	0

BIT[7] IAPSTR - IAP 操作启动控制位

0: 由硬件清 0，表示当前不在 IAP 流程中，或者 IAP 流程已结束；

1: 写 1 后启动一次 IAP 操作流程。

BIT[4] BLRS - IAP 读写区操作区域选择位，软件置位清零，需 INFOS=0 且 BLSIZE=1，此位才有效；

0: IAP 擦写及 MOVX 指令操作区域选择 main FLASH 区；

1: IAP 擦写及 MOVX 指令操作区域选择 bootloader 区；

注：当 INFOS=1，IAP 擦写及 MOVX 指令操作区域为 info block 区；若 INFOS=0 且 BLSIZE=0，此位无效，IAP 擦写及 MOVX 指令操作区域固定为 main FLASH 区。

BIT[3] IAPTRIG2 - IAP 操作执行控制位 2

0: 由硬件清 0，无 IAP 操作，或者当前 IAP 操作已结束；

1: 写 1 后，硬件开始执行一次 IAP 操作。

BIT[2] IAPTRIG1 - IAP 操作执行控制位 1

0: 由硬件清 0，无 IAP 操作，或者当前 IAP 操作已结束；

1: 写 1 后，硬件开始执行一次 IAP 操作。

BIT[1] IAPTRIG0 - IAP 操作执行控制位 0

0: 由硬件清 0，无 IAP 操作，或者当前 IAP 操作已结束；

1: 写 1 后，硬件开始执行一次 IAP 操作。

注：特别注意，为了防止芯片误动作，以上四个位的设置需要配合 IAPKEY 寄存器的设置联动操作，否则设置无效，详细可以参考“IAP 操作标准流程”。



BIT[0] **INFOS** - IAP 读写区操作区域选择位，软件置位清零
 0: IAP 擦写及 MOV C 指令操作区域选择 ALL FLASH ROM 区（包括 main flash rom 和 bootloader rom）；
 1: IAP 擦写及 MOV C 指令操作区域选择 info block 区（包括 EEPROM-LIKE、UID、用户配置字，分区加密配置字）。

注 1: 因 FLASH ROM 和 info block 地址重叠，芯片通过 **INFOS** 位识别 MOV C 指令访问的是 EEPROM-LIKE 区还是 FLASH ROM 区，擦写操作也通过此位识别 EEPROM-LIKE 区还是 FLASH 区。

注 2: Main FLASH 和 bootloader 地址重叠，通过 **BLRS** 区分 main flash 还是 bootloader

注 3: 当用 MOV C 指令读取“用户配置字”中设置了代码区加密功能时（ENCR 位=1），FLASH/bootloader 区读取到的内容将为零。

19.1.7 IAP 控制寄存器 1（IAPCR1, 0xED/SFR0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPCR1	BOOT	MFREF	IAPERR	–	PCFEN	UCFEN	BLREN	MFREN
R/W	R/W	R	R/*W	–	R/W	R/W	R/W	R/W
复位值	0	0	0	–	0	0	0	0

BIT[7] **BOOT** - 启动 ROM 区域选择
 0: 复位后 MCU 从 main flash rom 区 0000H 启动；
 1: 复位后 MCU 从 bootloader rom 区 0000H 启动；
 注：该位由复位（软件复位除外）后读取 **USER_OPT3→BOOT5**（BIT7）内容载入，软件复位（SWRST）此位保持不变。

注：CPU 复位后，硬件根据 BOOT 位（非 BOOT5 位）决定启动开始于 main flash rom 还是 bootloader rom。

BIT[6] **MFREF** - main flash 区全擦标志位，硬件置 1 清 0
 0: main flash 区未执行全部擦除；
 1: main flash 区执行了全部擦除操作（对全部 mainflash 空间执行 block 擦除，此位会置位）。

BIT[5] **IAPERR** - IAP 错误标志位，硬件置 1，软件清 0
 0: 无错误发生，或软件清零；
 1: 发生 IAP 错误。
 满足以下任意条件，硬件在 IAPSTR 置 1（IAP 启动）后置位 IAPERR。

- 1) 访问地址超限（不存在的区域）
- 2) 访问地址为禁止访问区
- 3) 擦除或烧写的区域使能未开启
- 4) IAP 的命令未正确执行（无效或执行错误）

注：指令未正确执行：IAPKEY 的 4 次解锁（IAPSRT、IAPTRIG0、IAPTRIG1、IAPTRIG3）未按照正确顺序操作。

注：此 IAPERR 标志，仅在 IAP 模块操作及 MOV C 指令操作满足上述条件才会置位；如果 PC 指针超限，此位不会置位，CPU 执行 nop 指令。

注：此位软件清零，或 POR 复位，其他复位不能复位此标志位

BIT[3] **PCFEN** - 加密配置区（password configuration block）擦/写操作使能
 0: 禁止擦/写加密配置区；
 1: 允许擦/写加密配置区。

BIT[2] **UCFEN** - 用户配置区（User configuration block）擦/写操作使能
 0: 禁止擦/写用户配置区；
 1: 允许擦/写用户配置区。

BIT[1] **BLREN** - Bootloader rom 擦/写操作使能
 0: 禁止擦/写 Bootloader rom 区；
 1: 允许擦/写 Bootloader rom 区。

注：特别注意，此 Bootloader rom 区域空间大小受 BLSIZE 控制。

BIT[0] **MFREN** - Main Flash rom 擦/写操作使能
 0: 禁止擦/写 Main Flash rom 区；
 1: 允许擦/写 Main Flash rom 区。

注：特别注意，此 Main Flash rom 区域空间大小受 BLSIZE 控制。

19.2 IAP 命令

IAP 命令汇总

IAP 操作	IAPCR1	IAPCR	IAPMD	IAPAD[H:L]	IAPDB
--------	--------	-------	-------	------------	-------



		区域擦写门控				访问区选择		IAP 使能&擦写模式选择					16 位地址	8 位数据
		POFEN	UCFEN	BLREN	MPREN	BLRS	INFOS	IAPEN	BLOCKERS	CHIPERS	PAGEERS	BYTEWR	IAPAD[16:0]	IAPDB[7:0]
96 位 UID	字节读	x	x	x	x	0	1	x	0	0	0	0	1868H~1873H	只读
Main flash rom	块擦	x	x	x	1	0	0	1	1	0	0	0	块地址, 低 9 位清 0	00H
	页擦	x	x	x	1	0	0	1	0	0	1	0	页地址, 低 7 位清 0	00H
	字节写	x	x	x	1	0	0	1	0	0	0	1	待写字节地址	待写字节数据
	字节读	x	x	x	x	0	0	x	0	0	0	0	Movc 指令读取	Movc 指令读取
Bootloader rom	块擦	x	x	1	x	1	0	1	1	0	0	0	块地址, 低 9 位清 0	00H
	页擦	x	x	1	x	1	0	1	0	0	1	0	页地址, 低 7 位清 0	00H
	字节写	x	x	1	x	1	0	1	0	0	0	1	待写字节地址	待写字节数据
	字节读	x	x	x	x	1	0	x	0	0	0	0	Movc 指令读取	Movc 指令读取
User cfg block	页擦	x	1	x	x	0	1	1	0	0	1	0	页地址, 1000H	00H
	字节写	x	1	x	x	0	1	1	0	0	0	1	待写字节地址	待写字节数据
	字节读	x	x	x	x	0	1	x	0	0	0	0	Movc 指令读取	Movc 指令读取
PSW cfg block	页擦	1	x	x	x	0	1	1	0	0	1	0	页地址, 2000H	00H
	字节写	1	x	x	x	0	1	1	0	0	0	1	待写字节地址	待写字节数据
	字节读	x	x	x	x	0	1	x	0	0	0	0	Movc 指令读取	Movc 指令读取
EEPROM-like block	页擦	x	x	x	x	0	1	1	0	0	1	0	页地址, 0000H	00H
	字节写	x	x	x	x	0	1	1	0	0	0	1	待写字节地址	待写字节数据
	字节读	x	x	x	x	0	1	x	0	0	0	0	Movc 指令读取	Movc 指令读取

注: main flash 地址区间 0000H~47FFH@BLSIZE=0, 0000H~3FFFH@BLSIZE=1;

Bootloader 地址区间 0000H~07FFH@BLSIZE=1;

96 位 UID 地址区间 1868H~1873H

User cfg block 地址区间 1000H~107FH

PSW cfg block 地址区间 2000H~207FH

EEPROM-like block 地址区间 0000H~007FH

19.3 IAP 操作标准流程

19.3.1 EEPROM-LIKE 读取

标准操作流程, 如下:

- (1) 首先, 关闭中断; 若使用看门狗的话, 必须将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=1;
- (3) 用 MOVc 指令读取对应的 EEPROM-LIKE 地址, 若需要可以重复操作多次;
- (4) 完成后, 恢复 INFOS=0, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

19.3.2 FLASH 读取

标准操作流程, 如下:

- (1) 首先, 关闭中断; 若使用看门狗的话, 必须将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=0, BLRS=0 (默认情况下);
- (3) 用 MOVc 指令读取对应的 FLASH 地址, 若需要可以重复操作多次;
- (4) 完成后, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

19.3.3 EEPROM-LIKE 页擦除

标准操作流程, 如下:

- (1) 首先, 关闭中断; 若使用看门狗的话, 必须将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=1;
- (3) 将 IAPEN 位重复执行 2 次置 1 操作, 将 PAGEERS 重复执行 2 次置 1 操作;
- (4) 对 IAPKEY[7:0]写入 0x5A, 再写入 0xA5, 解锁后对 IAPSRT 位置 1;
- (5) 向 IAPADH 中写入页擦除的 EEPROM-LIKE 字节的地址高 8 位;
- (6) 向 IAPADL 中写入页擦除的 EEPROM-LIKE 字节的地址低 8 位;



- (7) 向 IAPDB 中写入 0 值; (擦除完成数据变为 FFH)
- (8) 将 IAPADL 的低 7 位清 0; (擦除完成数据变为 FFH)
- (9) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG0 位置 1;
- (10) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG1 位置 1;
- (11) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG2 位置 1;
- (12) 若一切正常, 在 EEPROM-LIKE 页擦除的典型时间后, 原本置 1 的 IAPSTR 位将硬件清 0, 表示本次页擦除成功, 并退出本次 IAP 流程, 用户可以通过查询该标志位来确认本次操作的完成情况;
- (13) 完成后, 恢复 INFOS=0, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

注: 其他 INFO 区 (用户配置字、加密配置字) 操作同 EEPROM-like。

19.3.4 FLASH 页擦除

标准操作流程, 如下:

- (1) 首先, 关闭中断; 若使用看门狗的话, 必须将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=0, BLRS=0;
- (3) 将 IAPEN 位重复执行 2 次置 1 操作, 将 PAGEERS 重复执行 2 次置 1 操作;
- (4) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPSRT 位置 1;
- (5) 向 IAPADH 中写入页擦除的 FLASH 字节的地址高 8 位;
- (6) 向 IAPADL 中写入页擦除的 FLASH 字节的地址低 8 位;
- (7) 向 IAPDB 中写入 0 值; (擦除完成数据变为 FFH)
- (8) 将 IAPADL 的低 7 位清 0; (擦除完成数据变为 FFH)
- (9) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG0 位置 1;
- (10) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG1 位置 1;
- (11) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG2 位置 1;
- (12) 若一切正常, 在 FLASH 页擦除的典型时间后, 原本置 1 的 IAPSTR 位将硬件清 0, 表示本次页擦除成功, 并退出本次 IAP 流程, 用户可以通过查询该标志位来确认本次操作的完成情况;
- (13) 完成后, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

19.3.5 EEPROM-LIKE 字节写入 (1 字节)

标准操作流程, 如下:

- (1) 首先, 关闭中断, 将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=1;
- (3) 将 IAPEN 位重复执行 2 次置 1 操作, 将 BYTEWR 重复执行 2 次置 1 操作;
- (4) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPSRT 位置 1;
- (5) 向 IAPADH 中写入预备写入的 EEPROM-LIKE 字节的地址高 8 位;
- (6) 向 IAPADL 中写入预备写入的 EEPROM-LIKE 字节的地址低 8 位;
- (7) 向 IAPDB 中写入预备写入的新数值;
- (8) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG0 位置 1;
- (9) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG1 位置 1;
- (10) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG2 位置 1;
- (11) 若一切正常, 在 EEPROM-LIKE 字节写入的典型时间后, 原本置 1 的 IAPSTR 位将硬件清 0, 表示本次 EEPROM-LIKE 字节写入成功, 并退出本次 IAP 流程, 用户可以通过查询该标志位来确认本次操作的完成情况;
- (12) 完成后, 恢复 INFOS=0, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

19.3.6 FLASH 字节写入 (1 字节)

标准操作流程, 如下:

- (1) 首先, 关闭中断, 将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=0, BLRS=0;
- (3) 将 IAPEN 位重复执行 2 次置 1 操作, 将 BYTEWR 重复执行 2 次置 1 操作;
- (4) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPSRT 位置 1;
- (5) 向 IAPADH 中写入预备写入的 FLASH 字节的地址高 8 位;
- (6) 向 IAPADL 中写入预备写入的 FLASH 字节的地址低 8 位;
- (7) 向 IAPDB 中写入预备写入的新数值;
- (8) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG0 位置 1;
- (9) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG1 位置 1;
- (10) 对 IAPKEY[7:0] 写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG2 位置 1;
- (11) 若一切正常, 在 FLASH 字节写入的典型时间后, 原本置 1 的 IAPSTR 位将硬件清 0, 表示本次 FLASH 字节写入成功, 并退出本次 IAP 流程, 用户可以通过查询该标志位来确认本次操作的完成情况;
- (12) 完成后, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。



19.3.7 FLASH 块擦除

标准操作流程, (EEPROM-LIKE 不支持块擦除) 如下:

- (1) 首先, 关闭中断; 若使用看门狗的话, 必须将看门狗溢出时间调整到足够大;
- (2) 写 INFOS=0, BLRS=0;
- (3) 将 IAPEN 位重复执行 2 次置 1 操作, 将 BLOCKERS 重复执行 2 次置 1 操作;
- (4) 对 IAPKEY[7:0]写入 0x5A, 再写入 0xA5, 解锁后对 IAPSRT 位置 1;
- (5) 向 IAPADH 中写入页擦除的 FLASH 字节的地址高 8 位;
- (6) 向 IAPADL 中写入页擦除的 FLASH 字节的地址低 8 位;
- (7) 向 IAPDB 中写入 0 值;
- (8) 将 IAPADL 的低 9 位清 0;
- (9) 对 IAPKEY[7:0]写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG0 位置 1;
- (10) 对 IAPKEY[7:0]写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG1 位置 1;
- (11) 对 IAPKEY[7:0]写入 0x5A, 再写入 0xA5, 解锁后对 IAPTRIG2 位置 1;
- (12) 若一切正常, 在 FLASH 块擦除的典型时间后, 原本置 1 的 IAPSTR 位将硬件清 0, 表示本次块擦除成功, 并退出本次 IAP 流程, 用户可以通过查询该标志位来确认本次操作的完成情况;
- (13) 完成后, 退出本次 IAP 操作, 打开中断; 若使用看门狗的话, 复原看门狗溢出时间。

注 1: 芯片复位后, 需等待至少 20uS 后才能进行 EEPROM 以及 IAP 的正常操作;

注 2: 若 FLASH 地址处在加密区域中, “用户”是无法读取 FLASH 内容的, 即使强行读出则内容皆为 0。

注 3: 第一次使用 EEPROM-LIKE 前, 建议用户执行页擦一次。

19.4 Boot loader 区

bootloader 实质上借用主程序区最后 2k。通过 USER_OPT3->BLSIZE、USER_OPT3->BOOTS 和 IAPCR1->BOOT 控制最后 2k 是否作为 bootloader 区以及启动选择。

19.4.1 启动选择

本产品提供 2 种启动选择, 通过 IAPCR1->BOOT (bit7) 控制选择 cpu 从 main flash rom 0000H 启动或 bootloader rom 0000H 启动。

复位后 (软件复位除外), user_opt3->BOOTS (bit7) 加载到 IAPCR1->BOOT (bit7); 软件复位不执行此加载。

IAP 控制寄存器 1 (0xED, SFR0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPCR1	BOOT		IAPERR	-	PCFEN	UCFEN	BLREN	MFREN
R/W	R/W		R/*W	-	R/W	R/W	R/W	R/W
复位值	0		0	-	0	0	0	0

BIT[7] BOOT - 启动 ROM 区域选择

0: 复位后 MCU 从 main flash rom 区 0000H 启动;

1: 复位后 MCU 从 bootloader rom 区 0000H 启动;

注: 该位由复位 (软件复位除外) 后读取 USER_OPT3->BOOTS (BIT7) 内容载入, 软件复位 (SWRST) 此位保持不变。

注: CPU 复位后, 硬件根据 BOOT 位 (非 BOOTS 位) 决定启动开始于 main flash rom 还是 bootloader rom。

19.4.2 boot loader 空间大小选择

设置 USER_OPT3->BLSIZE, 可选择【BootLoader=0k, Main FLASH=18k】或【BootLoader=2k, Main FLASH=16k】;

ICP 进入烧写模式执行烧写, 不受启动控制及 bootloader 空间大小选择控制, 直接对 0000H~47FFH 操作。

19.4.3 Boot loader 区内的擦除/读写操作

在 bootloader 区内, 通过扇区擦除/块擦除功能, 可以擦除用户程序区和类 EEPROM 区的任意扇区, 以及 bootloader 区 (但对程序运行所在的 page 或 block 执行 page Erase 或 block Erase 无效), 可以擦除 INFO 区内用户/工具配置区 (用户配置字) 和加密配置区 (加密配置字)。

在 bootloader 区内, 不支持整体擦除功能;

在 bootloader 区内, 通过 IAP 功能, 可以向主程序存储区及 bootloader 区的空白区域写入数据, 也可以向 INFO 区内的类 EEPROM 区和用户/工具配置区的空白区域写入数据。

在 bootloader 区内, 可以用 MOVc 指令读取用户程序区、类 EEPROM 区以及 bootloader 区内的内容。在 bootloader 区内, 也可以读取用户/工具配置区、加密配置区和 96 位 UID。



必须指出，对 INFO 区内的用户配置字的写操作只针对有效地址有效，禁止对保留地址进行写操作。

以上操作仅限于目标区域没有进行代码保护的情况。当目标区域处于代码保护模式时，参见 FLASH 存储器部分关于代码保护方式的描述。

19.4.4 bootloader 应用

bootloader 区可实现 ISP 功能，用户可以把引导程序代码放在 bootloader 区（最大 2k），用户的引导代码可以通过烧录器或 ICP 下载至 bootloader 区。

一般应用中，ISP 通过 PC 端工具和 MCU 连接并通讯，PC 可以通过串口传输待更新的用户代码至 MCU。引导区代码接收并调用 IAP 模块烧录至指定用户代码区。

相关的引导代码例程及 PC 端软件，请参见<>。

19.5 INFO 区

EEPROM-LIKE 区（Addition Sector），支持 IAP 扇区擦除和字节写入/读取。

用户/工具配置区，支持 IAP 扇区擦除和字节写入/读取。

出厂配置区，不支持 IAP 扇区擦除和字节写入/读取；但 96 位 UID 支持只读操作。

加密配置区，仅在 bootloader 区执行 main flash 全擦后，支持 IAP 扇区擦除和字节写入；读取不受此条件限制；

在访问 INFO 区之前，应首先将寄存器 IAPCR->INFOS 控制位置 1；在访问 INFO 区之后，应将 IAPCR->INFOS 位清 0。

INFO 区内只能存放数据，不能存放用户程序。CPU 不会到此区域内取指令。

19.5.1 Flash 加密操作

被加密的 flash 区不支持扇区擦除和字节写入/读取；

被加密的 flash 区支持 block 擦除；

19.5.2 加密控制字解锁

加密控制字，在未执行所有的 main flash 区全部擦除动作前，不支持擦写操作（操作无效）；

满足两个条件，解锁加密控制字擦写权限：

- 1、程序在 bootloader 区执行
- 2、执行所有的 main flash 区 block 擦除

19.6 用户主程序区（main flash）

主程序区内，通过 IAP 扇区擦除/块擦除功能，可以擦除用户程序区（除正在执行代码所在的扇区/block 以及加密的区块）、bootloader 区（加密后不可擦除）、用户配置区和类 EEPROM 区的任意扇区；

注 1：主程序区内，对当前执行代码所在扇区或 block，执行扇区擦除或 block 擦除，操作将无效；

注 2：主程序区内，对被加密的区块执行擦除无效

注 3：主程序区内，禁止擦写分区加密控制字。

用户程序区内，对 info 区操作，必须符合对应条件，请参见《4.3 INFO 区》相关说明。

用户程序区内的擦除和读写操作参见 IAP 章节。

19.7 User option 控制

USER_OPT3 配置字

地址/位	符号	功能说明
1006H	USER_OPT3	代码加密/OTP 使能/ OCD 访问控制
BIT[7]	BOOTS	BOOT - 启动 ROM 区域选择 0: 复位后 MCU 从 main flash rom 区 0000H 启动; 1: 复位后 MCU 从 bootloader rom 区 0000H 启动; 注: 若 BLSIZE=0, 选择 BOOTS=1 无效, 复位后依旧从 main flash rom 区 0000H 启动 注: 除软复位 (SWRST) 外, 其他复位此位内容加载至 IAPCR1->BOOT (BIT7)。
BIT[6]	BLSIZE	Bootloader Rom 区空间大小选择 0: BootLoader=0k, Main FLASH=18k 1: BootLoader=2k, Main FLASH=16k 注: 若设置为 BootLoader=2k (BLSIZE=1), 烧写模式下 (ICP) 仍可以通过访问第 17/18k 空间 (4000H~47FFH) 访问。



19.8 复位说明

软件复位与其他复位区别：

- 1、 Warmup 时间不同，请参见用户手册《复位预热章节》；
- 2、 Option 加载不同，除软件复位的其他复位，会将 USER_OPT3->BOOTS 值加载至 IAPCR1->BOOT (BIT7)，BOOT 控制 CPU 启动位置：main flash rom 或 bootloader rom 的 0000H。软件复位的 option 加载相位，不加载 BOOTS 位，BOOT 位保持不变，其他 option 加载同其他复位。

19.9 ICP、IAP 和 ISP

ICP:

通过 JTAG 进入 ctrl 模式，再进入烧写模式

操作地址：0000H~47FFH，访问 18k 的 flash 空间，不受 BOOT 和 BLSIZE 影响；

实现方式：通过仿真器或烧写器，根据用户设置区域（main flash 或 bootloader）直接烧写 18k 空间。

IAP:

通过片上的 IAP 模块及对应的 IAP 命令操作方式，配合用户代码实现指定区域 rom 擦写；

操作地址：

main flash 启动，0000H~47FFF@ (BLSIZE=0&INFOS=0&BLRS=0 选择 main flash)；

0000H~3FFFF@ (BLSIZE=1& INFOS=0&BLRS=0 选择 main flash)；

0000H~07FFF@ (BLSIZE=1& INFOS=0&BLRS=1 选择 bootloader)；

bootloader 启动，0000H~3FFFF@ (BLSIZE=1& INFOS=0&BLRS=0 选择 main flash)；

bootloader 启动，不能对 bootloader 区擦写。

ISP:

通过启动控制，借助 pc 端软件与 MCU 通讯接口连接，并传输代码至 bootloader 区，bootloader 为用户开发的应用代码，调用 IAP 模块和通讯接口，实现 main flash 区代码升级。



20 附录

20.1 指令集总览表

助记符	说明	周期(T)	长度(B)
ADD A, Rn	寄存器 Rn 和累加器 ACC 相加, 结果存到 ACC	1	1
ADD A, direct	直接寻址字节和累加器 ACC 相加, 结果存到 ACC	2	2
ADD A, @Ri	内部 RAM Ri 和累加器 ACC 相加, 结果存到 ACC	2	1
ADD A, #data	立即数和累加器 ACC 相加, 结果存到 ACC	2	2
ADDC A, Rn	寄存器 Rn 和累加器 ACC 相加 (带 C 标志), 结果存到 ACC	1	1
ADDC A, direct	直接寻址字节和累加器 ACC 相加 (带 C 标志), 结果存到 ACC	2	2
ADDC A, @Ri	内部 RAM Ri 和累加器 ACC 相加 (带 C 标志), 结果存到 ACC	2	1
ADDC A, #data	立即数和累加器 ACC 相加 (带 C 标志), 结果存到 ACC	2	2
SUBB A, Rn	累加器 ACC 减寄存器 Rn (带 C 标志), 结果存到 ACC	1	1
SUBB A, direct	累加器 ACC 减直接寻址字节 (带 C 标志), 结果存到 ACC	2	2
SUBB A, @Ri	累加器 ACC 减内部 RAM Ri (带 C 标志), 结果存到 ACC	2	1
SUBB A, #data	累加器 ACC 减立即数 (带 C 标志), 结果存到 ACC	2	2
INC A	累加器 ACC 加 1, 结果存到 ACC	1	1
INC Rn	寄存器 Rn 加 1, 结果存到 Rn	2	1
INC direct	直接寻址字节加 1, 结果存到直接选址字节	3	2
INC @Ri	内部 RAM Ri 加 1, 结果存到 Ri	3	1
INC DPTR	数据指针 DPTR/DPRT1 加 1, 结果存到 DPTR/DPRT1	4	1
DEC A	累加器 ACC 自减 1, 结果存到 ACC	1	1
DEC Rn	寄存器 Rn 自减 1, 结果存到 Rn	2	1
DEC direct	直接寻址字节减 1, 结果存到直接选址字节	3	2
DEC @Ri	内部 RAM Ri 减 1, 结果存到 Ri	3	1
MUL AB 8 X 8	累加器 ACC 乘寄存器 B	11	1
16 X 8		20	
DIV AB 8 / 8	累加器 ACC 除寄存器 B	11	1
16 / 8		20	
DA A	BCD 码调整	1	1
ANL A, Rn	累加器 ACC 与寄存器 Rn, 结果存到 ACC	1	1
ANL A, direct	累加器 ACC 与直接寻址字节, 结果存到 ACC	2	2
ANL A, @Ri	累加器 ACC 与内部 RAM Ri, 结果存到 ACC	2	1
ANL A, #data	累加器 ACC 与立即数, 结果存到 ACC	2	2
ANL direct, A	直接寻址字节与累加器 ACC, 结果存到直接寻址字节	3	2
ANL direct, #data	直接寻址字节与立即数, 结果存到直接寻址字节	3	3
ORL A, Rn	累加器 ACC 或寄存器 Rn, 结果存到 ACC	1	1
ORL A, direct	累加器 ACC 直接寻址字节, 结果存到 ACC	2	2
ORL A, @Ri	累加器 ACC 或内部 RAM Ri, 结果存到 ACC	2	1
ORL A, #data	累加器 ACC 或立即数, 结果存到 ACC	2	2
ORL direct, A	直接寻址字节或累加器 ACC, 结果存到直接寻址字节	3	2



ORL direct, #data	直接寻址字节或立即数, 结果存到直接寻址字节	3	3
XRL A, Rn	累加器 ACC 异或寄存器 Rn, 结果存到 ACC	1	1
XRL A, direct	累加器 ACC 直接寻址字节, 结果存到 ACC	2	2
XRL A, @Ri	累加器 ACC 异或内部 RAM Ri, 结果存到 ACC	2	1
XRL A, #data	累加器 ACC 异或立即数, 结果存到 ACC	2	2
XRL direct, A	直接寻址字节异或累加器 ACC, 结果存到直接寻址字节	3	2
XRL direct, #data	直接寻址字节异或立即数, 结果存到直接寻址字节	3	3
CLR A	对 ACC 清零	1	1
CPL A	对 ACC 取反	1	1
RL A	累加器 ACC 循环左移	1	1
RLC A	累加器 ACC 循环左移 (带 C 标志)	1	1
RR A	累加器 ACC 循环右移	1	1
RRC A	累加器 ACC 循环右移 (带 C 标志)	1	1
SWAP A	交换累加器 ACC 的高低半字节, 结果存到 ACC	4	1
MOV A, Rn	将寄存器 Rn 送到累加器 ACC	1	1
MOV A, direct	将直接寻址字节送到累加器 ACC	2	2
MOV A, @Ri	将内部 RAM Ri 送到累加器 ACC	2	1
MOV A, #data	将立即数送到累加器 ACC	2	2
MOV Rn, A	将累加器 ACC 送到寄存器 Rn	2	1
MOV Rn, direct	将直接寻址直接送到寄存器 Rn	3	2
MOV Rn, #data	将立即数送到寄存器 Rn	2	2
MOV direct, A	将累加器 ACC 送到直接寻址字节	2	2
MOV direct, Rn	将寄存器 Rn 送到直接寻址字节	2	2
MOV direct1, direct2	将直接寻址字节 2 送到直接寻址字节 1	3	3
MOV direct, @Ri	将内部 RAM Ri 送到直接寻址字节	3	2
MOV direct, #data	将立即数送到直接寻址字节	3	3
MOV @Ri, A	将累加器 ACC 送到内部 RAM Ri	2	1
MOV @Ri, direct	将直接寻址字节送到内部 RAM Ri	3	2
MOV @Ri, #data	将立即数送到内部 RAM Ri	2	2
MOV DPTR, #data16	将 16 位立即数送到数据指针	3	3
MOVC A, @A+DPTR	将代码程序送到累加器 ACC (相对数据指针)	7	1
MOVC A, @A+PC	将代码程序送到累加器 ACC (相对程序计数器)	8	1
MOVB A, @Ri	通过内部 RAM Ri 将 8 位外存送到累加器 ACC	5	1
MOVB A, @DPTR	通过数据指针 DPTR 将 16 位外存送到累加器 ACC	6	1
MOVB @Ri, A	通过内部 RAM Ri 将累加器 ACC 送到 8 位外存	4	1
MOVB @DPTR, A	通过数据指针 DPTR 将累加器 ACC 送到 16 位外存	5	1
PUSH direct	直接寻址字节压入栈顶	5	2
POP direct	栈顶弹至直接寻址字节	4	2
XCH A, Rn	累加器 ACC 与寄存器 Rn 交换	3	1
XCH A, direct	累加器 ACC 与直接寻址字节交换	4	2
XCH A, @Ri	累加器 ACC 与内部 RAM Ri 交换	4	1
XCHD A, @Ri	累加器 ACC 低 4 位与内部 RAM Ri 低 4 位交换	4	1
ACALL addr11	2KB 内绝对调用	7	2
LCALL addr16	64KB 内长调用	7	3
RET	子程序返回	8	1
RETI	中断返回	8	1



AJMP addr11	2KB 内绝对转移	4	2
LJMP addr16	64KB 内长转移	5	3
SJMP rel	相对短转移	4	2
JMP @A+DPTR	相对长转移	6	1
JZ rel (不发生转移) (发生转移)	累加器为零转移	3	2
		5	
JNZ rel (不发生转移) (发生转移)	累加器为非零转移	3	2
		5	
JC rel (不发生转移) (发生转移)	C 置位转移	2	2
		4	
JNC rel (不发生转移) (发生转移)	C 清零转移	2	2
		4	
JB bit, rel (不发生转移) (发生转移)	直接寻址位置位转移	4	3
		6	
JNB bit, rel (不发生转移) (发生转移)	直接寻址位清零转移	4	3
		6	
JBC bit, rel (不发生转移) (发生转移)	直接寻址位置位转移并清该位	4	3
		6	
CJNE A, direct, rel (不发生转移) (发生转移)	累加器与直接寻址字节不等转移	4	3
		6	
CJNE A, #data, rel (不发生转移) (发生转移)	累加器与立即数不等转移	4	3
		6	
CJNE Rn, #data, rel (不发生转移) (发生转移)	寄存器与立即数不等转移	4	3
		6	
CJNE @Ri, #data, rel (不发生转移) (发生转移)	内部 RAM 与立即数不等转移	4	3
		6	
DJNZ Rn, rel (不发生转移) (发生转移)	寄存器减 1 不为零转移	3	2
		5	
DJNZ direct, rel (不发生转移) (发生转移)	直接寻址字节减 1 不为零转移	4	3
		6	
NOP	空操作	1	1
CLR C	C 清零	1	1
CLR bit	直接寻址位清零	3	2
SETB C	C 置位	1	1
SETB bit	直接寻址位置位	3	2
CPL C	C 取反	1	1
CPL bit	直接寻址位取反	3	2
ANL C, bit	C 逻辑与直接寻址位	2	2
ANL C, /bit	C 逻辑与直接寻址位的反	2	2
ORL C, bit	C 逻辑或直接寻址位	2	2
ORL C, /bit	C 逻辑或直接寻址位的反	2	2
MOV C, bit	直接寻址位送 C	2	2
MOV bit, C	C 送直接寻址位	3	2

注：对于条件跳转类指令，若跳转条件成立，则指令需 2 个周期，否则只需 1 个周期。



20.2 特殊功能寄存器总览表 (SFR0)

地址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
FF	SFR0	ADRH	ADC 转换结果高位寄存器	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8	xxxxxxx
FE	SFR0	ADRL	ADC 转换结果低位寄存器	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	xxxxxxx
FD	SFR0	ADCCR2	ADC 控制寄存器 2	ADCKS	ADCKDIV2	ADCKDIV1	ADCKDIV0	-	ADSPS2	ADSPS1	ADSPS0	0000-111
FC	SFR0	ADASCR2	ADC 自动扫描控制寄存器 2	ASCNT4	ASCNT3	ASCNT2	ASCNT1	ASCNT0	ADDR10	ADDR9	ADDR8	00000000
FB	SFR0	ADASCR1	ADC 自动扫描控制寄存器 1	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	-	0000000-
FA	SFR0	ADASCR0	ADC 自动扫描控制寄存器 0	ADJSTR	-	ASCNUM2	ASCNUM1	ASCNUM0	ASSTA	ASTRG	ASEN	---00000
F9	SFR0	PWRCR1	高级能耗控制寄存器 1	-	-	CRC_CLKEN	-	-	T3_CLKEN	T2_CLKEN	T01_CLKEN	11111111
F8	SFR0	PWRCR	高级能耗控制寄存器	-	-	PWM1_CLKEN	PWMO_CLKEN	SPI_CLKEN	ADC_CLKEN	-	OCD_CLKEN	11111111
F6	SFR0	OSCSS	时钟安全寄存器	CSEN	FHS2	FHS1	FHS0	CSSIE	HFSWS	LFSWS	CSSF	10100000
F5	SFR0	ADCCR4	ADC 控制寄存器 4	ADPDK	ADLPM	EXVREF_EN	-	ADCTRG3	ADCTRG2	ADCTRG1	ADCTRG0	---00000
F4	SFR0	ADCCR3	ADC 控制寄存器 3	ADPAC	ADRST1	ADRST0	ADACC2	ADACC1	ADACC0	ADBIT1	ADBIT0	00000000
F3	SFR0	ADCCR1	ADC 控制寄存器 1	TMPEN	ADGAIN1	ADGAIN0	-	ADCHS3	ADCHS2	ADCHS1	ADCHS0	00011111
F2	SFR0	ADCCR0	ADC 控制寄存器 0	CMPEM	ADCMPLIF	ADCMGIF	ADCIF	VRS1	VRS0	ADEOC	ADEN	00001100
F1	SFR0	AUXC	乘除法增强寄存器	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
F0	SFR0	B	B 寄存器	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
EF	SFR0	IAPKEY	IAP 写保护寄存器	IAPKEY7	IAPKEY6	IAPKEY5	IAPKEY4	IAPKEY3	IAPKEY2	IAPKEY1	IAPKEY0	00000000
EE	SFR0	IAPMD	IAP 模式寄存器	IAPEN	-	-	BLOCKERS	-	PAGEERS	=	BYTEWR	0--0-0-0
ED	SFR0	IAPCR1	IAP 控制寄存器寄存器 1	BOOT	MFREF	IAPERR	-	PCFEN	UCFEN	BLREN	MFREN	000-0000
EC	SFR0	IAPDB	IAP 数据寄存器	IAPDB7	IAPDB6	IAPDB5	IAPDB4	IAPDB3	IAPDB2	IAPDB1	IAPDB0	00000000
EB	SFR0	IAPADH	IAP 高地址寄存器	IAPAD15	IAPAD14	IAPAD13	IAPAD12	IAPAD11	IAPAD10	IAPAD9	IAPAD8	00000000
EA	SFR0	IAPADL	IAP 低地址寄存器	IAPAD7	IAPAD6	IAPAD5	IAPAD4	IAPAD3	IAPAD2	IAPAD1	IAPAD0	00000000
E9	SFR0	IAPCR	IAP 控制寄存器	IAPSTR	-	-	BLRS	IAPTRIG2	IAPTRIG1	IAPTRIG0	INFOS	0--00000
E8	SFR0	RSTFLAG	复位标志寄存器	PORF	LVRF	RSTF	WDTRF	SWRF	-	-	SWRST	00000--0
E7	SFR0	MDUDRB1	运算/结果寄存器 B 高位	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	00000000
E6	SFR0	MDUDRB0	运算/结果寄存器 B 低位	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0000000-
E5	SFR0	MDUDRA3	运算/结果寄存器 A 高 8 位	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24	00000000
E4	SFR0	MDUDRA2	运算/结果寄存器 A	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	00000000



地 址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
			次高 8 位									
E3	SFR0	MDUDRA1	运算/结果寄存器 A 次低 8 位	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	00000000
E2	SFR0	MDUDRA0	运算/结果寄存器 A 低 8 位	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	00000000
E1	SFR0	MDUCR	运算控制寄存器	MD1	MD0	SDIR	—	—	SIGN	DVERR	START	00000000
E0	SFR0	A	累加器	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
DF	SFR0	PWM1CNDH	PWM1CN 占空比寄 存器高位	PWMCND15	PWMCND14	PWMCND13	PWMCND12	PWMCND11	PWMCND10	PWMCND9	PWMCND8	00000000
DE	SFR0	PWM1CNDL	PWM1CN 占空比寄 存器低位	PWMCND7	PWMCND6	PWMCND5	PWMCND4	PWMCND3	PWMCND2	PWMCND1	PWMCND0	00000000
DD	SFR0	PWM1BNDH	PWM1BN 占空比寄 存器高位	PWMBND15	PWMBND14	PWMBND13	PWMBND12	PWMBND11	PWMBND10	PWMBND9	PWMBND8	00000000
DC	SFR0	PWM1BNDL	PWM1BN 占空比寄 存器低位	PWMBND7	PWMBND6	PWMBND5	PWMBND4	PWMBND3	PWMBND2	PWMBND1	PWMBND0	00000000
DB	SFR0	PWM1ANDH	PWM1AN 占空比寄 存器高位	PWMAND15	PWMAND14	PWMAND13	PWMAND12	PWMAND11	PWMAND10	PWMAND9	PWMAND8	00000000
DA	SFR0	PWM1ANDL	PWM1AN 占空比寄 存器低位	PWMAND7	PWMAND6	PWMAND5	PWMAND4	PWMAND3	PWMAND2	PWMAND1	PWMAND0	00000000
D9	SFR0	PWM1RLEN	PWM1 寄存器修改 重载控制寄存器	PWMRL7	PWMRL6	PWMRL5	PWMRL4	PWMRL3	PWMRL2	PWMRL1	PWMRL0	00000000
D8	SFR0	DPS	数据指针选择寄 存器	—	SFRSL0	—	—	DIV16	MUL16	—	SEL	00—00—0
D7	SFR0	PWM1CDH	PWM1C 占空比寄 存器高位	PWMCND15	PWMCND14	PWMCND13	PWMCND12	PWMCND11	PWMCND10	PWMCND9	PWMCND8	00000000
D6	SFR0	PWM1CDL	PWM1C 占空比寄 存器低位	PWMCND7	PWMCND6	PWMCND5	PWMCND4	PWMCND3	PWMCND2	PWMCND1	PWMCND0	00000000
D5	SFR0	PWM1BDH	PWM1B 占空比寄 存器高位	PWMBND15	PWMBND14	PWMBND13	PWMBND12	PWMBND11	PWMBND10	PWMBND9	PWMBND8	00000000
D4	SFR0	PWM1BDL	PWM1B 占空比寄 存器低位	PWMBND7	PWMBND6	PWMBND5	PWMBND4	PWMBND3	PWMBND2	PWMBND1	PWMBND0	00000000
D3	SFR0	PWM1ADH	PWM1A 占空比寄 存器高位	PWMAD15	PWMAD14	PWMAD13	PWMAD12	PWMAD11	PWMAD10	PWMAD9	PWMAD8	00000000
D2	SFR0	PWM1ADL	PWM1A 占空比寄 存器低位	PWMAD7	PWMAD6	PWMAD5	PWMAD4	PWMAD3	PWMAD2	PWMAD1	PWMAD0	00000000
D1	SFR0	PWM1OE	PWM1 输出使能寄 存器	PWM1EN	—	PWMCNOE	PWMBNOE	PWMANOE	PWMCOE	PWMBOE	PWMAOE	0—000000



地址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
D0	SFR0	PSW	程序状态寄存器	CY	AC	F0	RS1	RS0	OV	F1	P	00000000
CF	SFR0	WDTCR	看门狗定时器控制寄存器	-	-	-	-	-	WDTOT2	WDTOT1	WDTOT0	-----111
CE	SFR0	T2CON1	T2 控制寄存器 1	CM2ON	CM2	BR2EN	RCLKU1	TCLKU1	EX2SL	SIGTRG	T2EN	00000000
CD	SFR0	TH2	T2 计数高位寄存器	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20	00000000
CC	SFR0	TL2	T2 计数低位寄存器	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20	00000000
CB	SFR0	RCAPH2	T2 重载/捕获高位寄存器	RCAPH27	RCAPH26	RCAPH25	RCAPH24	RCAPH23	RCAPH22	RCAPH21	RCAPH20	00000000
CA	SFR0	RCAPL2	T2 重载/捕获低位寄存器	RCAPL27	RCAPL26	RCAPL25	RCAPL24	RCAPL23	RCAPL22	RCAPL21	RCAPL20	00000000
C9	SFR0	T2MOD	工作模式寄存器	LD2EN	T2CKS2	T2CKS1	T2CKS0	T2OE	T2SL1	T2SL0	-	00000000
C8	SFR0	T2CON	T2 控制寄存器	TF2	EXTF2	RCLKU0	TCLKU0	EX2EN	TR2	CT2	CPRL2	00000000
C7	SFR0	CMP1CR3	CMP1 控制寄存器 3	C1PWMTRGS1	C1PWMTRGS0	C1DAC5	C1DAC4	C1DAC3	C1DAC2	C1DAC1	C1DAC0	00000000
C6	SFR0	CMP1CR2	CMP1 控制寄存器 2	C1INSEL3	C1INSEL2	C1INSEL1	C1INSEL0	C1PSEL3	C1PSEL2	C1PSEL1	C1PSEL0	00000000
C5	SFR0	CMP1CR1	CMP1 控制寄存器 1	C1INV	C1IE	C1INSEL1	C1INSEL0	C1IS1	C1IS0	C1OUT	C1IF	0-000000
C4	SFR0	CMP1CR0	CMP1 控制寄存器 0	C1EN	C1TRGPOL	C1FILT2	C1RFSEL	C1SMT1	C1SMT0	C1FILT1	C1FILT 0	00000000
C3	SFR0	POPU	P0 端口上拉电阻控制寄存器	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU	00000000
C2	SFR0	POMOD1	P0 端口模式寄存器 1	P07M1	P07M0	P06M1	P06M0	P05M1	P05M0	P04M1	P04M0	10101010
C1	SFR0	POMOD0	P0 端口模式寄存器 0	P03M1	P03M0	P02M1	P02M0	P01M1	P01M0	P00M1	P00M0	10101010
C0	SFR0	OSCM	振荡模式寄存器	XTSPD	STBHXT	STBH	STBXT	-	CLKS	LFEN	HFEN	1000-x1x
BF	SFR0	PWM1FBCR	PWM1 刹车控制寄存器	FB0EN	FB0SEL	FB1EN	FB1S	FB1FLT1	FB1FLT0	FBMOD	FBSTA	00000001
BE	SFR0	PWM1CR2	PWM1 控制寄存器 2	TRGTIM1	TRGTIMO	PTRGAD	ZTRGAD	PLDEN	OSYNC	DTMOD1	DTMOD0	00000-00
BD	SFR0	PWM1CR1	PWM1 控制寄存器 1	OUTMOD	SYMC	PWMCNS	PWMBNS	PWMANS	PWMCNS	PWMBNS	PWMAS	00000000
BC	SFR0	IPH1	中断优先级控制 1 高位寄存器	CRCIPH	CMPIPH	PWM1FBIPH	PWMOIPH	UART1IPH	PWM1IPH	T3IPH	T2IPH	00000000
BB	SFR0	IP1	中断优先级控制 1 低位寄存器	CRCIPL	CMPIPL	PWM1FBIPL	PWMOIPL	UART1IPL	PWM1IPL	T3IPL	T2IPL	00000000
BA	SFR0	IPH	中断优先级控制高位寄存器	-	ADCIPH	SPIIPH	UARTOIPH	T1IPH	EXT1IPH	TOIPH	EXTOIPH	-0000000
B9	SFR0	IP	中断优先级控制低位寄存器	-	ADCIPL	SPIIPL	UARTOIPL	T1IPL	EXT1IPL	TOIPL	EXTOIPL	-0000000
B8	SFR0	PWM1IF	PWM1 中断标志寄存器	PWMPIF	PWMZIF	PWMDDCIF	PWMUDCIF	PWMDDBIF	PWMUDBIF	PWMDDAIF	PWMUDAIF	00000000
B7	SFR0	PWM1MANCR1	PWM1 手动输出控制	-	-	PWMOCN	PWMOBN	PWMOAN	PWMOC	PWMOB	PWMOA	--000000



地址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
			制寄存器 1									
B6	SFR0	PWM1MANCRO	PWM1 手动输出控制寄存器 0	-	-	MANCN	MANBN	MANAN	MANC	MANB	MANA	--000000
B5	SFR0	EXTEN	外部中断使能寄存器	-	EXTOEN4	EXTOEN3	EXTOEN2	EXTOEN1	EXTOEN0	-	EXTOA	-00000-0
B4	SFR0	EXTEN1	外部中断使能寄存器 1	-	EXT1EN4	EXT1EN3	EXT1EN2	EXT1EN1	EXT1EN0	-	EXT1A	-00000-0
B3	SFR0	P3PU	P3 端口上拉电阻控制寄存器	-	-	-	-	-	-	P31PU	P30PU	-----00
B2	SFR0	P3MOD1	P3 端口模式寄存器 1									
B1	SFR0	P3MOD0	P3 端口模式寄存器 0	-	-	-	-	P31M1	P31M0	P30M1	P30M0	----1010
B0	SFR0	P3	P3 端口数据寄存器	-	-	-	-	-	-	P31D	P30D	-----xx
AF	SFR0	SPDAT	SPI 数据寄存器	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0	xxxxxxxx
AE	SFR0	SPSTA	SPI 状态寄存器	SPEN	SPIF	MODF	WCOL	SPIOV	-	-	-	00000---
AD	SFR0	SPCON	SPI 控制寄存器	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0	00000000
AC	SFR0	EXTIS	外部中断控制寄存器	-	EXT0IF4	EXT0IF3	EXT0IF2	EXT0IF1	EXT0IF0	EXT0IS1	EXT0IS0	-0000000
AB	SFR0	EXTIS1	外部中断控制寄存器 1	-	EXT1IF4	EXT1IF3	EXT1IF2	EXT1IF1	EXT1IF0	EXT1IS1	EXT1IS0	-0000000
AA	SFR0	EXTIS2										
A9	SFR0	IE1	中断使能寄存器 1	CRCIE	TKIE	PWM1FBIE	PWMOIE	UART1IE	PWM1IE	T3IE	T2IE	00000000
A8	SFR0	IE	中断使能寄存器	EA	ADCIE	SPIIE	UART0IE	T1IE	EX1IE	TOIE	EX0IE	00000000
A7	SFR0	CMPOCR3	CMPO 控制寄存器 3	COPWMTRGS1	COPWMTRGS0	CODAC5	CODAC4	CODAC3	CODAC2	CODAC1	CODAC0	00000000
A6	SFR0	CMPOCR2	CMPO 控制寄存器 2	CONSEL3	CONSEL2	CONSEL1	CONSEL0	COPSEL3	COPSEL2	COPSEL1	COPSEL0	00000000
A5	SFR0	CMPOCR1	CMPO 控制寄存器 1	COINV	COIE	COINSEL1	COINSEL0	COIS1	COIS0	COOUT	COIF	0-000000
A4	SFR0	CMPOCR0	CMPO 控制寄存器 0	COEN	COTRGPOL	COFILT2	CORFSEL	COSMT1	COSMT0	COFILT1	COFILT0	00000000
A3	SFR0	P2PU	P2 端口上拉电阻控制寄存器	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU	00000000
A2	SFR0	P2MOD1	P2 端口模式寄存器 1	P27M1	P27M0	P26M1	P26M0	P25M1	P25M0	P24M1	P24M0	10101010
A1	SFR0	P2MOD0	P2 端口模式寄存器 0	P23M1	P23M0	P22M1	P22M0	P21M1	P21M0	P20M1	P20M0	10101010
A0	SFR0	P2	P2 端口数据寄存器	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D	xxxxxxxx
9F	SFR0	PXODM	端口输出驱动控制寄存器	-	-	-	-	P10DM1	P10DM0	POODM1	POODM0	----0000
9D	SFR0	SBUF_1	串口 1 缓存寄存器	SBUF7_1	SBUF6_1	SBUF5_1	SBUF4_1	SBUF3_1	SBUF2_1	SBUF1_1	SBUF0_1	xxxxxxxx
9C	SFR0	SCON_1	串口 1 控制寄存器	SMO_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	00000000
9B	SFR0	SADEN	串口 0 地址掩码寄存器	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0	00000000



地址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
			寄存器									
9A	SFR0	SADDR	串口 0 从机地址寄存器	SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0	00000000
99	SFR0	SBUF	串口 0 缓存寄存器	SBUF7	SBUF6	SBUF5	SBUF4	SBUF3	SBUF2	SBUF1	SBUF0	xxxxxxxx
98	SFR0	SCON	串口 0 控制寄存器	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI	00000000
97	SFR0	TH3	T3 计数高位寄存器	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30	00000000
96	SFR0	TL3	T3 计数低位寄存器	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30	00000000
95	SFR0	BRGH	串口 1 波特率计数器高位寄存器	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	00000000
94	SFR0	BRGL	串口 1 波特率计数器低位寄存器	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	00000000
93	SFR0	P1PU	P1 端口上拉电阻控制寄存器	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU	00000000
92	SFR0	P1MOD1	P1 端口模式寄存器 1	P17M1	P17M0	P16M1	P16M0	P15M1	P15M0	P14M1	P14M0	10101010
91	SFR0	P1MOD0	P1 端口模式寄存器 0	P13M1	P13M0	P12M1	P12M0	P11M1	P11M0	P10M1	P10M0	10101010
90	SFR0	P1	P1 端口数据寄存器	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D	xxxxxxxx
8F	SFR0	T3CR	T3 控制寄存器	TR3	CT3	T3F	-	T3EN	T3CKS2	T3CKS1	T3CKS0	000-0000
8E	SFR0	TCON1	定时器控制寄存器 1	T1FS	T10E	T1CKS1	T1CKS0	TOFS	T00E	TOCKS1	TOCKS0	00000000
8D	SFR0	TH1	定时器计数高位寄存器 1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10	00000000
8C	SFR0	TH0	定时器计数高位寄存器 0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00	00000000
8B	SFR0	TL1	定时器计数低位寄存器 1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10	00000000
8A	SFR0	TL0	定时器计数低位寄存器 0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00	00000000
89	SFR0	TMOD	定时器模式寄存器	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0	00000000
88	SFR0	TCON	定时器控制寄存器	TF1	TR1	TF0	TR0	-	-	-	-	0000----
87	SFR0	PCON	电源控制寄存器	UART1EN	-	-	UART0EN	SMOD0	SSTAT0	SLEEP	STOP	0-00000
86	SFR0	SLPCR	模式保护控制寄存器	SLPCR7	SLPCR6	SLPCR5	SLPCR4	SLPCR3	SLPCR2	SLPCR1	SLPCR0	00000000
85	SFR0	DPH1	数据指针 1 高字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
84	SFR0	DPL1	数据指针 1 低字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
83	SFR0	DPH	数据指针高字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
82	SFR0	DPL	数据指针低字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000000
81	SFR0	SP	堆栈指针	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	00000111
80	SFR0	P0	P0 端口数据寄存器	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D	xxxxxxxx



20.1 特殊功能寄存器总览表 (XSFR)

地 址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
FFB2	XSFR	CMP00_MAP	CMP00 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFB0	XSFR	CMP10_MAP	CMP10 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA8	XSFR	PWM0_MAP	PWM0 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA6	XSFR	PWM1FB_MAP	PWM1FB 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA5	XSFR	PWM1CN_MAP	PWM1CN 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA4	XSFR	PWM1C_MAP	PWM1C 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA3	XSFR	PWM1BN_MAP	PWM1BN 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA2	XSFR	PWM1B_MAP	PWM1B 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA1	XSFR	PWM1AN_MAP	PWM1AN 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FFA0	XSFR	PWM1A_MAP	PWM1A 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF9B	XSFR	MOSI_MAP	MOSI 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF9A	XSFR	MISO_MAP	MISO 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF99	XSFR	SCK_MAP	SCK 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF98	XSFR	nSS_MAP	nSS 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF8D	XSFR	RX1_MAP	RX1 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF8C	XSFR	TX1_MAP	TX1 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF89	XSFR	RX0_MAP	RX0 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF88	XSFR	TX0_MAP	TX0 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF87	XSFR	T2EX_MAP	T2EX 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF82	XSFR	T2_MAP	T2 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111



地 址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
			寄存器									
FF81	XSFR	T1_MAP	T1 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF80	XSFR	T0_MAP	T0 端口映射控制寄存器	-	-	PORTSEL1	PORTSEL0	-	PINSEL2	PINSEL1	PINSEL0	--11-111
FF71	XSFR	OPAOCR1	OP0 控制寄存器 1	OPONSEL3	OPONSEL2	OPONSEL1	OPONSEL0	OPOPSEL3	OPOPSEL2	OPOPSEL1	OPOPSEL0	00000000
FF70	XSFR	OPAOCR0	OP0 控制寄存器 0	OP0EN	-	OP0OE1	OP0OE0	-	OP0GAIN2	OP0GAIN1	OP0GAIN0	0-00-000
FF6A	XSFR	PWMOD	PWM0 占空比设置寄存器	PWMOD7	PWMOD6	PWMOD5	PWMOD4	PWMOD3	PWMOD2	PWMOD1	PWMOD0	00000000
FF69	XSFR	PWMOP	PWM0 周期寄存器	PWMOP7	PWMOP6	PWMOP5	PWMOP4	PWMOP3	PWMOP2	PWMOP1	PWMOP0	00000000
FF68	XSFR	PWMOCR	PWM0 控制寄存器	PWM0EN	-	PWM0IF	PWM0OE	PWM0S	PWM0C2	PWM0C1	PWM0C0	0-000000
FF67	XSFR	CRCEDL	CRC 结束地址低字节寄存器	CRCEDL7	CRCEDL6	CRCEDL5	CRCEDL4	CRCEDL3	CRCEDL2	CRCEDL1	CRCEDL0	11111111
FF66	XSFR	CRCEDH	CRC 结束地址高字节寄存器	CRCEDH7	CRCEDH6	CRCEDH5	CRCEDH4	CRCEDH3	CRCEDH2	CRCEDH1	CRCEDH0	11111111
FF65	XSFR	CRCSTL	CRC 起始地址低字节寄存器	CRCSTL7	CRCSTL6	CRCSTL5	CRCSTL4	CRCSTL3	CRCSTL2	CRCSTL1	CRCSTL0	00000000
FF64	XSFR	CRCSTH	CRC 起始地址高字节寄存器	CRCSTH7	CRCSTH6	CRCSTH5	CRCSTH4	CRCSTH3	CRCSTH2	CRCSTH1	CRCSTH0	00000000
FF63	XSFR	CRCL	CRC 结果低字节寄存器	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0	11111111
FF62	XSFR	CRCH	CRC 结果高字节寄存器	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0	00000000
FF60	XSFR	CRCCR	CRC 控制寄存器	CRCEN	-	CRCIF	CRCST	CRCMD	CRCLCT	-	CRCENC	0-0000-0
FF4F	XSFR	PWM1DT1L	PWM1 死区控制寄存器 1 低位	PWMDT17	PWMDT16	PWMDT15	PWMDT14	PWMDT13	PWMDT12	PWMDT11	PWMDT10	00000000
FF4D	XSFR	PWM1DT0L	PWM1 死区控制寄存器 0 低位	PWMDT07	PWMDT06	PWMDT05	PWMDT04	PWMDT03	PWMDT02	PWMDT01	PWMDT00	00000000
FF4B	XSFR	PWM1PL	PWM1 周期寄存器 低位	PWMP7	PWMP6	PWMP5	PWMP4	PWMP3	PWMP2	PWMP1	PWMP0	00000000
FF4A	XSFR	PWM1PH	PWM1 周期寄存器 高位	PWMP15	PWMP14	PWMP13	PWMP12	PWMP11	PWMP10	PWMP9	PWMP8	00000000
FF49	XSFR	PWM1IE	PWM1 中断使能控制寄存器	PWMP1E	PWMP1ZIE	PWMP1DCIE	PWMP1UDCIE	PWMP1DBIE	PWMP1UDBIE	PWMP1DAIE	PWMP1UDAIE	00000000
FF48	XSFR	PWM1CR0	PWM1 控制寄存器 0	CLKSEL	-	CTMOD1	CTMOD0	PREDIV1	PREDIV0	POSTDIV1	POSTDIV0	0-000000
FF47	XSFR	ADSEQR2	ADC 扫描序列寄存器 2	SEQ53	SEQ52	SEQ51	SEQ50	SEQ43	SEQ42	SEQ41	SEQ40	00000000

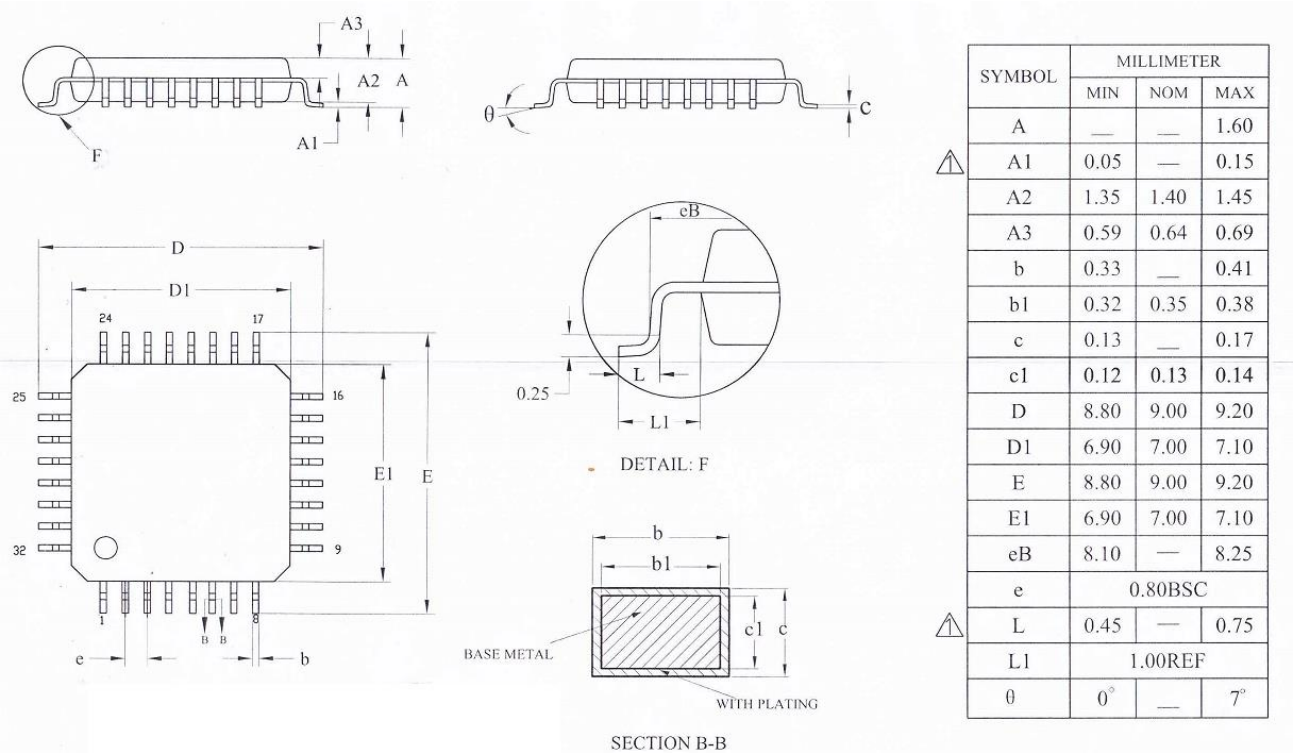


地 址 (H)	SFR 号	寄存器名	定义	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 (B)
FF46	XSFR	ADSEQR1	ADC 扫描序列寄存器 1	SEQ33	SEQ32	SEQ31	SEQ30	SEQ23	SEQ22	SEQ21	SEQ20	00000000
FF45	XSFR	ADSEQR0	ADC 扫描序列寄存器 0	SEQ13	SEQ12	SEQ11	SEQ10	SEQ03	SEQ02	SEQ01	SEQ00	00000000
FF43	XSFR	ADCMPLL	ADC 数字比较下限低位寄存器	CMPL7	CMPL6	CMPL5	CMPL4	CMPL3	CMPL2	CMPL1	CMPL0	00000000
FF42	XSFR	ADCMP LH	ADC 数字比较下限高位寄存器	CMPL15	CMPL14	CMPL13	CMPL12	CMPL11	CMPL10	CMPL9	CMPL8	00000000
FF41	XSFR	ADCMPGL	ADC 数字比较上限低位寄存器	CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0	00000000
FF40	XSFR	ADCMPGH	ADC 数字比较上限高位寄存器	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP9	CMP8	00000000



21 封装信息

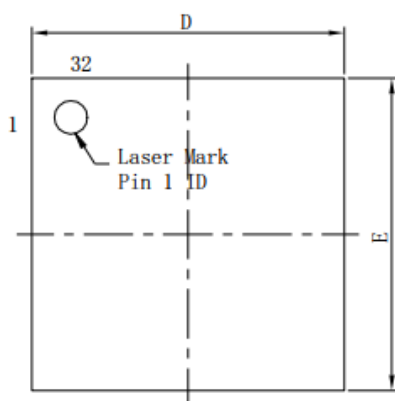
21.1 LQFP32 (0.8mm pitch)





21.2 QFN32 (4x4x0.75-0.40mm pitch)

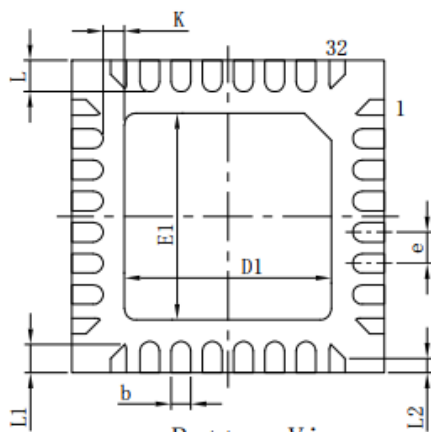
标注	尺寸	最小	标准	最大	标注	尺寸	最小	标准	最大
A		0.70	0.75	0.80	E1		2.55	2.65	2.75
A1		0.00	—	0.05	e		0.40TYP		
A3		0.203REF			K		0.20	—	—
b		0.15	0.20	0.25	L		0.30	0.40	0.50
D		3.90	4.00	4.10	L1		0.31	0.36	0.41
E		3.90	4.00	4.10	L2		0.13	0.18	0.23
D1		2.55	2.65	2.75					



Top View



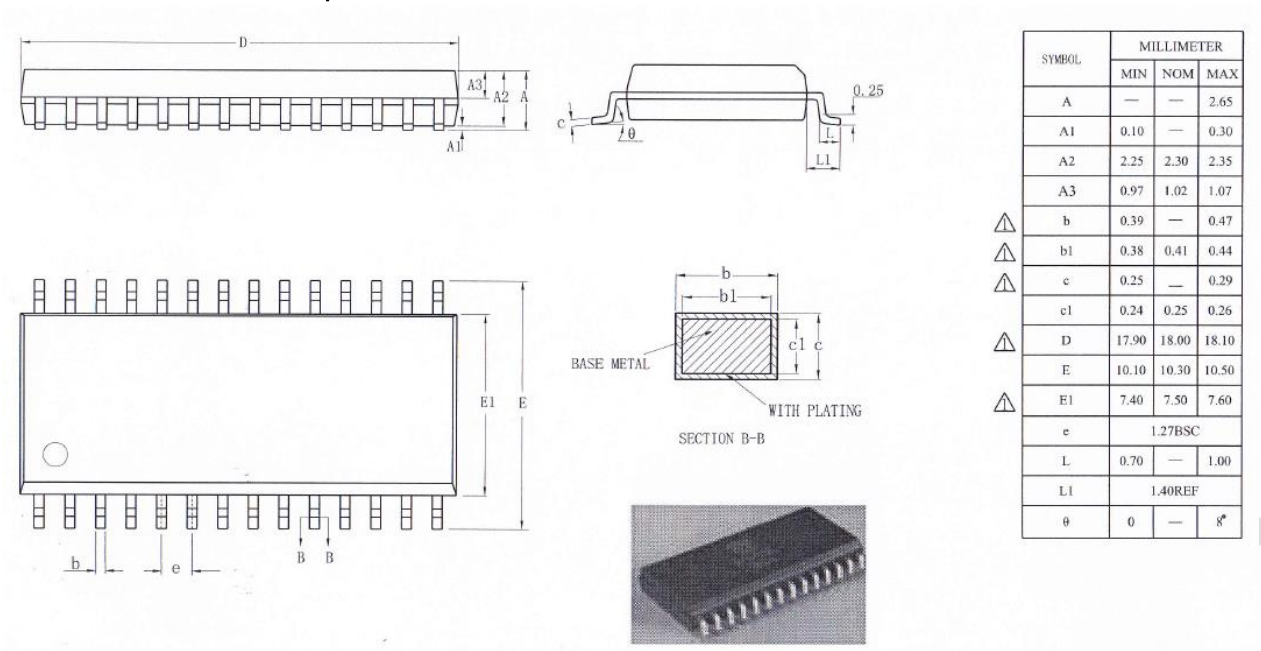
Side View



Bottom View



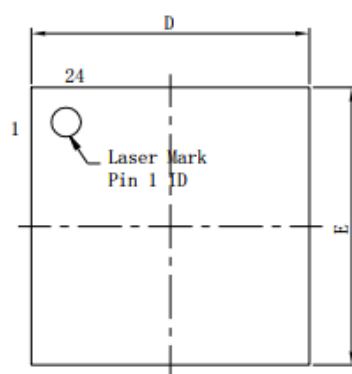
21.3 SOP28 (1.27mm pitch)



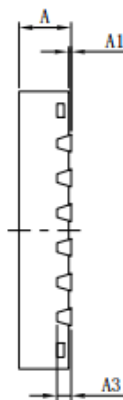


21.4 QFN24 (4x4x0.75 0.50mm pitch)

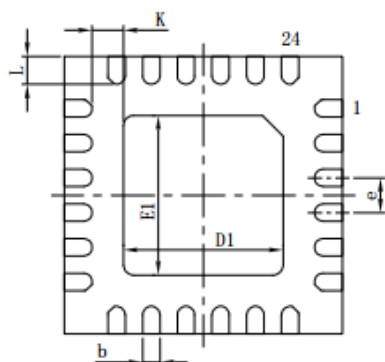
标注	尺寸	最小	标准	最大	标注	尺寸	最小	标准	最大
A		0.70	0.75	0.80	D1		2.20	2.30	2.40
A1		0.00	—	0.05	E1		2.20	2.30	2.40
A3		0.203REF			e		0.50TYP		
b		0.20	0.25	0.30	K		0.20	—	—
D		3.90	4.00	4.10	L		0.30	0.40	0.50
E		3.90	4.00	4.10					



Top View



Side View

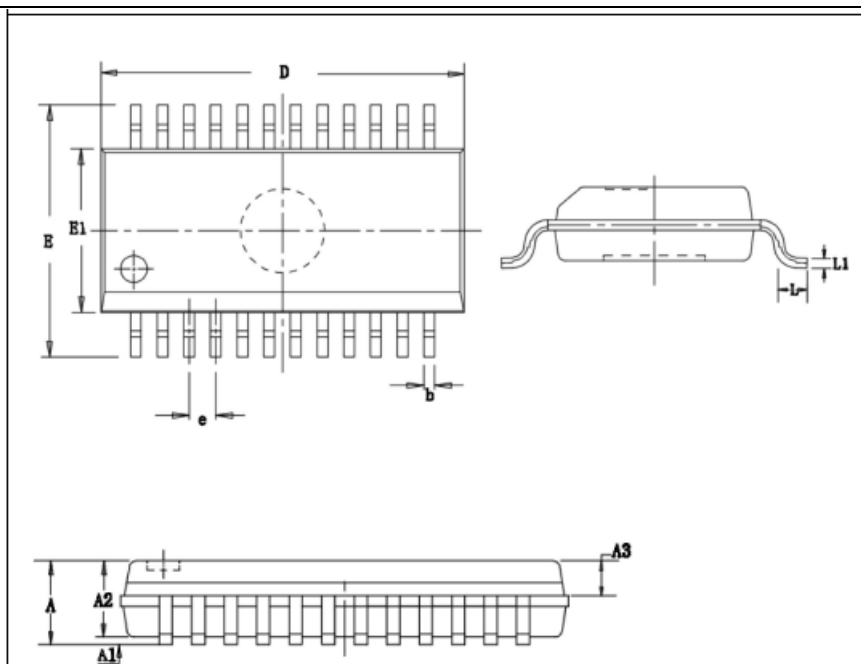
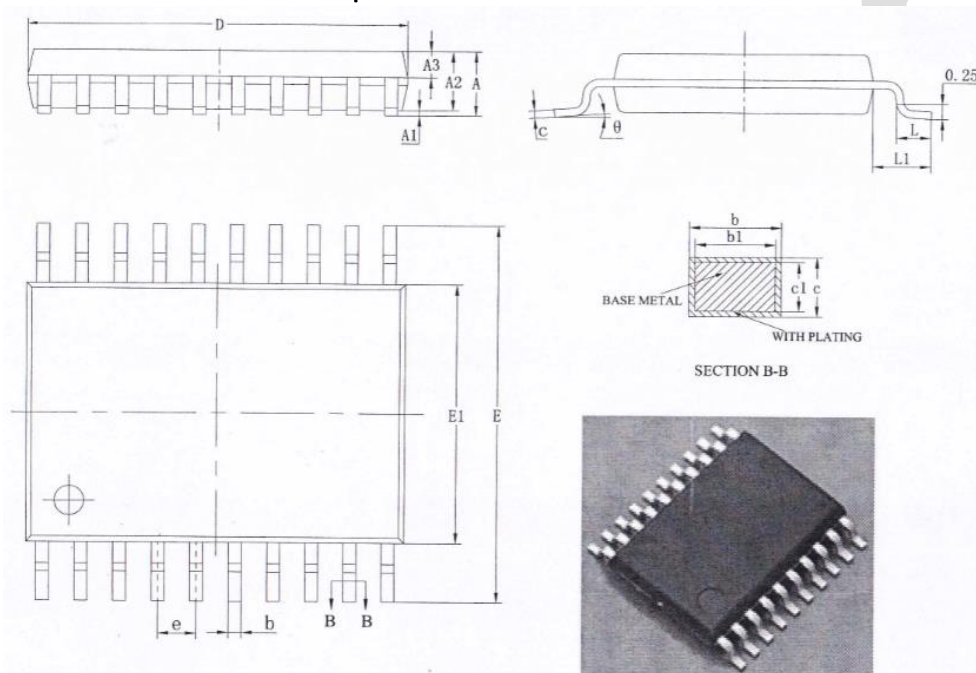


Bottom View

21.5 SS0P24 (0.635mm pitch)

单位:mm

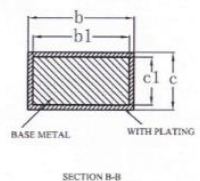
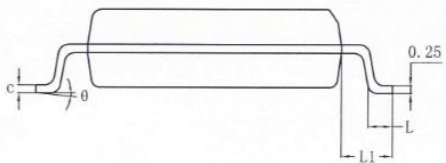
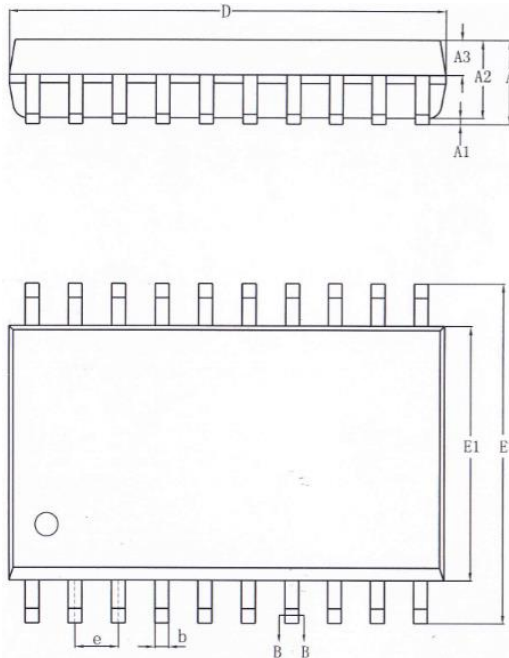
	MIN	NOM	MAX
A	—	—	1.700
A1(站高)	0.100	0.150	0.200
A2(厚度)	1.300	1.400	1.500
A3	0.600	0.650	0.700
b	0.229	—	0.279
e(脚间距)	0.605	0.635	0.665
D(长度)	8.500	8.600	8.700
E(跨度)	5.800	6.000	6.200
E1(宽度)	3.800	3.900	4.000
L(脚长)	0.500	0.600	0.700
L1	—	0.25BSC	—

**21.6 TSSOP20 (0.65mm pitch)**

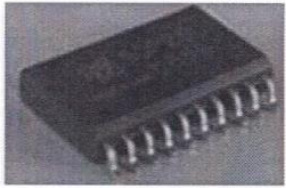
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.28
b1	0.19	0.22	0.25
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00REF		
θ	0	—	8°



21.7 SOP20 (1.27mm pitch)



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	2.65
A1	0.10	—	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.35	—	0.43
b1	0.34	0.37	0.40
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	12.70	12.80	12.90
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
θ	0	—	8°





22 版本修订记录

版本号	修订者	修订日期	修订内容
V1.0.0	sinomcu	2023-6-28	初始版本 7424B 增加了高低频晶振荡器的支持，新增 P3.0/P3.1 端口
V1.0.1	sinomcu	2023-6-30	1、引脚排列调整为 B0xx 系列，所有的封装都支持晶振功能；
V1.0.2	sinomcu	2023-7-10	1、用户配置字，更新高频晶体相关配置位
V1.0.3	sinomcu	2023-8-24	1、ADC 零点偏移修调章节，软件修调方法修正； 2、ADC 零偏校准寄存器控制位 QS[6:0]描述修正；
V1.0.4	sinomcu	2024-4-8	1、用户配置字，更新 EXOSC_FILS 为 FSYS_FILS，滤波对系统时钟 (Fsys) 有效，更新相关描述；
V1.0.5	sinomcu	2024-4-9	1、增加 SOP28 封装，代号 B0G；
V1.0.6	sinomcu	2025-4-8	1、增加 MC51F7424A1YK SSOP24 封装； 2、增加 MC51F7424A1Y TSSOP20/SOP20 封装； 3、删除 MC51F7424B0G SOP28 封装； 4、修改 PWM1FBCR 寄存器的 bit0 FBSTR 的初值为 1； 5、增加 MC51F7424B 与 MC51F7424 的对比表；
V1.0.7	sinomcu	2025-5-9	1、修改温度传感器章节内容；



23 免责声明

本资料内容为晟矽微电子（以下简称“我公司”）版权所有。

我公司将力求本资料的内容做到准确无误，但同时保留在不通知用户的情况下，对本资料内容的修改权。如您需要获得最新的资料，请及时联系我公司。

我公司将尽最大努力为您提供高品质、高稳定性的产品。尽管如此，由于一般半导体器件的电气敏感性及易受到外部物理损伤等固有特性存在，所以难免造成半导体器件出现故障或失效的可能。当您使用我公司产品时，有责任按照本资料以及相连资料中提到的“规则”来设计一个稳定及安全可靠的系统环境。另外，在遇到超规格（本资料中未描述到内容）的使用，请您提前咨询我公司，以免因我公司产品在一些特殊设备中或者特殊环境下的使用，导致财产损失、人员伤亡等严重后果的发生。因超规格的使用、未经咨询授权我公司产品的使用，我公司对此不承担任何法律责任。