

M9F951

**8BIT
USB+AD 型
FLASH MCU**

Version 1.05

2022 年 08 月



卓 芯 微 半 导 体

本公司不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，本公司的产品不是专门设计来应用于外科植入、生命维持和任何本公司产品的故障会对个体造成伤害甚至死亡的领域。如果将本公司的产品应用于上述领域，即使这些是由本公司在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证本公司及其雇员、子公司、分支机构和销售商与上述事宜无关。

目录

1 产品简述	9
1.1 特性	9
1.2 引脚图	11
1.2.1 LQFP48	11
1.2.2 LQFP64	12
1.3 引脚描述	13
2 中央处理器 (CPU)	18
2.1 程序存储器	18
2.1.1 复位向量 (0000h)	18
2.1.2 中断向量 (0x0008, 0x0018)	19
2.1.3 程序计数器	19
2.1.4 堆栈	20
2.1.5 SPUSH 和 SPOP 指令	21
2.1.6 快速寄存器堆栈	21
2.1.7 查表	22
2.2 数据存储器	23
2.2.1 数据存储器结构	23
2.2.2 BSR 页面选择寄存器	23
2.2.3 快速操作寄存器	24
2.2.4 数据存储器寻址模式	25
2.2.5 间接寻址 0/1/2	26
2.2.6 8x8 硬件乘法器	27
2.2.7 STATUS 状态寄存器	27
2.2.8 系统寄存器定义	28
2.3 FLASH 自编程 (IAP)	31
2.3.1 EECON1 寄存器	31
2.3.2 EECON2 寄存器	31
2.3.3 Flash 写入缓存器	32
2.3.4 Flash 操作示例	32
3 复位	33
3.1 复位方式	33
3.2 PCON 复位状态寄存器	34
4 系统时钟	35
4.1 概述	35
4.2 OSCM 寄存器	35
4.3 系统时钟的工作模式	36
4.3.1 普通模式	36
4.3.2 绿色模式	36
4.3.3 休眠模式	36

4.4 IRCCAL 寄存器.....	37
4.5 IRCCAH 寄存器.....	37
4.6 系统时钟结构框图	39
4.7 系统时钟高低频切换	40
5 中断	41
5.1 概述	41
5.2 中断向量分配	41
5.3 OPTION 配置寄存器.....	42
5.4 IO 端口变化中断使能寄存器	43
5.5 INTCR0 中断控制寄存器 0	44
5.6 INTF0 中断标志寄存器 0	45
5.7 INTP0 中断优先级寄存器 0	46
5.8 INTCR1 中断控制寄存器 1	47
5.9 INTF1 中断标志寄存器 1	48
5.10 INTP1 中断优先级寄存器 1	49
5.11 INTCR2 中断控制寄存器 2	50
5.12 INTF2 中断标志寄存器 2	51
5.13 INTP2 中断优先级寄存器 2	52
5.14 USB 中断控制寄存器.....	53
5.15 USB 中断优先级寄存器.....	54
6 端口	55
6.1 数据寄存器 IOx (x = A/B/C/D/E/F)	55
6.2 输出锁存寄存器 IOxOR (x = A/B/C/D/E/F)	55
6.3 输出方向寄存器 OEx (x = A/B/C/D/E/F)	56
6.4 上拉控制寄存器 PUX (x = A/B/C/D/E/F)	56
6.5 下拉控制寄存器 PDx (x = A/B/C/D/E/F)	57
6.6 模拟端口设置寄存器 ANSx (x = A/B/C/D)	57
6.7 驱动电流选择寄存器 IOxODS1/ IOxODS0 (x = A/B/C/D/E/F)	58
6.8 翻转电平设置寄存器 IOxIPS/FLIPCR (x = A/B/C/D/E/F)	61
7 定时器 0 (TC0/PWM0)	62
7.1 概述	62
7.2 TOCR 控制寄存器.....	63
7.3 TC0CL TC0 计数器低 8 位/周期寄存器	64
7.4 TC0CH TC0 计数器高位	64
7.5 PWMOCR 控制寄存器.....	65
7.6 PWMOD 数据位.....	65
7.7 PWM0 波形实例.....	66
8 定时器 1 (TC1/PWM1)	67
8.1 概述	67
8.2 T1CR 控制寄存器.....	68
8.3 TC1CL TC1 计数器低 8 位	69

8.4 TC1CH TC1 计数器高 8 位	69
8.5 TC1PRL TC1 周期寄存器低 8 位	69
8.6 TC1PRH TC1 周期寄存器高 8 位	69
8.7 PWM1xCR 控制寄存器 (x=0-21)	70
8.8 PWM1xH 数据寄存器 (x = 0-21)	70
8.9 PWM1x 波形示例 (x = 0-21)	71
9 定时器 2 (TC2)	72
9.1 概述	72
9.2 T2CR 控制寄存器	73
9.3 TC2CL 计数器低位	74
9.4 TC2CH 计数器高位	74
9.5 TC2PRL 周期寄存器低位	74
9.6 TC2PRH 周期寄存器高位	74
9.7 TC2GCR 门控控制寄存器	75
9.7.1 门控-TC0 溢出周期	76
9.7.2 门控-上升沿到下降沿模式	76
9.7.3 门控-下降沿到上升沿模式	76
9.7.4 门控-上升沿到上升沿模式	77
9.7.5 门控-下降沿到下降沿模式	77
10 通用串行通讯口 (USART)	78
10.1 概述	78
10.2 TXxCR 发送控制寄存器 (x=0/1)	78
10.3 TXxREG 发送数据寄存器 (x=0/1)	79
10.4 RXxCR 接收控制寄存器 (x=0/1)	79
10.5 RXxREG 接收数据寄存器 (x=0/1)	79
10.6 BRGDxH 波特率寄存器高位 (x=0/1)	80
10.7 BRGDxL 波特率寄存器低位 (x=0/1)	80
10.8 USART 使用说明	81
10.8.1 波特率设置	81
10.8.2 异步发送	82
10.8.3 异步接收	84
10.8.4 同步发送	85
10.8.5 同步接收	87
10.8.6 唤醒及休眠模式下通讯	88
11 串行通讯口 (I2C)	89
11.1 概述	89
11.1.1 I ² C	89
11.1.2 SPI	89
11.2 I ² C 功能描述	90
11.2.1 I ² C 通讯示意图	90
11.2.2 I ² C 从机模式	91
11.2.3 I ² C 主机模式	94

11.3 SPI 功能描述	100
11.3.1 SPI 通讯示意图	100
11.3.2 SPI 软件操作流程	102
11.4 寄存器	103
11.4.1 控制寄存器 1 (I2CxCR1, x=0/1)	103
11.4.2 控制寄存器 2 (I2CxCR2, x=0/1)	104
11.4.3 本机地址寄存器 (I2CxOAR, x=0/1)	105
11.4.4 预分频寄存器 (I2CxPRESC, x=0/1)	105
11.4.5 数据缓存寄存器 (I2CxDBUF, x=0/1)	105
11.4.6 状态寄存器 (I2CxSR, x=0/1)	106
11.5 唤醒及休眠模式下通讯	107
12 PWM2	108
12.1 概述	108
12.2 寄存器	108
PWM2CR0 控制寄存器 0	108
PWM2CR1 控制寄存器 1	109
PWM2CR2 控制寄存器 2	110
PWM2OE 端口使能极性控制寄存器	111
PMANUALCR0 端口控制寄存器 0	111
PMANUALCR1 端口控制寄存器 1	112
FLTCR 故障检测寄存器	113
PWM2INTCR 中断控制寄存器	114
PWM2INTF 中断标志寄存器	115
PWM2PRL 周期寄存器低位	116
PWM2PRH 周期寄存器高位	116
PWM2DTxL 死区寄存器低位 (x=0, 1)	116
PWM2DTxH 死区寄存器高位 (x=0, 1)	116
PWM2_xPL 占空比寄存器低位 (x=0, 1)	116
PWM2_xPH 占空比寄存器高位 (x=0, 1)	116
PWM2_xNL 占空比寄存器低位 (x=0, 1)	117
PWM2_xNH 占空比寄存器高位 (x=0, 1)	117
PWM2RLDEN 寄存器修改和重载控制寄存器	117
12.3 图示说明	118
12.4 使用步骤	119
13 模数转换器 (ADC)	120
13.1 概述	120
13.2 ADCON0 寄存器	120
13.3 ADCON1 寄存器	121
13.4 ADCON2 寄存器	122
13.5 ADH ADC 数据高位	122
13.6 ADL ADC 数据低位	122
14 看门狗 (WDT)	123

14.1 概述	123
14.2 WDTC 看门狗控制寄存器 0	123
14.3 WDTC 看门狗控制寄存器 1	124
14.4 软件配置看门狗	124
15 USB	125
15.1 概述	125
15.2 USB 时钟	125
15.3 总线连接	126
15.4 总线唤醒	126
15.5 相关寄存器	126
15.5.1 SPI UCLKCON USB 校准寄存器.....	126
15.5.2 USBCON USB 总线控制寄存器.....	127
15.5.3 UECON USB EP0 控制寄存器.....	128
15.5.4 UDADR USB 设备地址寄存器.....	128
15.5.5 UEXCON USB EPx 控制寄存器	129
15.5.6 UEXIOPS USB EPx 的 IN/OUT 包大小寄存器	130
15.5.7 LDO	130
15.5.8 USB 输出驱动.....	130
16 比较器(CMP)	131
16.1 概述	131
16.2 比较器框图	131
16.3 CMPC0 比较器控制寄存器 0	132
16.4 CMPC1 比较器控制寄存器 1	133
16.5 CMPC2 运放/比较器控制寄存器.....	134
17 芯片配置字 (OPTION BIT)	135
18 电性参数	137
18.1 极限参数.....	137
18.2 直流特性.....	137
18.3 IO 口拉灌电流特性.....	140
18.4 系统时钟特性	142
18.5 FLASH 特性.....	145
18.6 ADC 电气特性.....	146
19 封装信息	147
19.1 LQFP48	147
19.2 LQFP64	148
20 指令集简述	149
20.1 概述	149
20.2 符号说明	149
20.3 M9 指令集表.....	150

21 修改记录	153
---------------	-----

1 产品简述

M9F951 采用高速低功耗 CMOS 工艺设计开发的 8 位高性能精简指令单片机，内部有 32K×8 位 FLASH（可擦写次数 100,000 次）程序存储器（支持 IAP，页擦页写，每页 128 字节），2048×8 位 RAM，USB2.0 全速设备（支持 6 个端点），48 个双向 I/O 口，3 个 16 位 Timer 定时器/计数器，1 组 8 位 PWM，两组 16 位 PWM，2 路 UART，2 路 SPI/IIC，28+4 路 12 位 AD 转换器，支持多种系统工作模式和多个中断源。

1.1 特性

■ CPU 特性

- 高性能精简指令
- 32K×8位的FLASH程序存储器
- 2048×8位的数据存储器
- 高速CPU
- 31级堆栈缓存器

■ I/O 口

- 最多48个双向I/O口
- 所有端口可设置弱上拉(15K)下拉(30K)
- 两路外部中断
- 电平变化中断
- 6个大电流口（Sink电流 380mA@VOL=GND+1V）

■ 1 个 8/16 位定时器/计数器

- 可设8位/16位计数模式
- T0CR：定时器控制寄存器
- PWM0

■ 2 个 16 位定时器/计数器

- TCxPRH/TCxPRL：定时器周期寄存器
- TxCR：定时器控制寄存器

■ PWM1

- 16位分辨率
- 22个独立Duty PWM输出
- 以定时器1为时基

■ PWM2

- 16位分辨率
- 3个独立Duty PWM
- A组，B组输出可选

■ USART

- 宽范围波特率
- 支持半双工同步模式
- 支持全双工异步模式

■ 系统时钟

- 内部高速RC震荡器：16/24MHz（±3%）

USB(校准模式)模式精度0.25%

- 内部低速RC震荡器：64KHz/500KHz
- 外部高速晶体震荡器：4-24MHz
- 外部低速晶体振荡器：32768Hz

■ 系统工作模式

- 普通模式：高低速时钟同时工作
- 低速模式：仅低速时钟工作
- 休眠模式：高低速时钟都停止工作

■ 28+4 路 12 位 ADC

- 28路外部输入
- 内部通道（VDD/4，VREF，temp和 GND）

■ I2C

- 主从模式
- 7位和10位地址

■ SPI

- 全双工模式
- 主从模式

■ 多路唤醒源/中断

- 两级中断优先级
- 多个唤醒源

■ 比较器

- 4个正相输入源
- 4个反相输入源
- 1路内部电源电压检测VDD/2
- 1路内部GND电压检测
- 1路可设内部参考电压检测

■ USB

- 兼容USB2.0全速/低速
- 6个端点，0-5
- 共用256字节USB BUFFER
 - EP0 : 32字节
 - EP1 : 32字节

- EP2 : 32字节
- EP3 : 16字节
- EP4 : 64字节
- EP5 : 64字节

- 置LDO 输出 (3.3V)

■ 看门狗定时器

■ 特殊功能

- 可编程代码保护
- ISP功能
- IAP功能

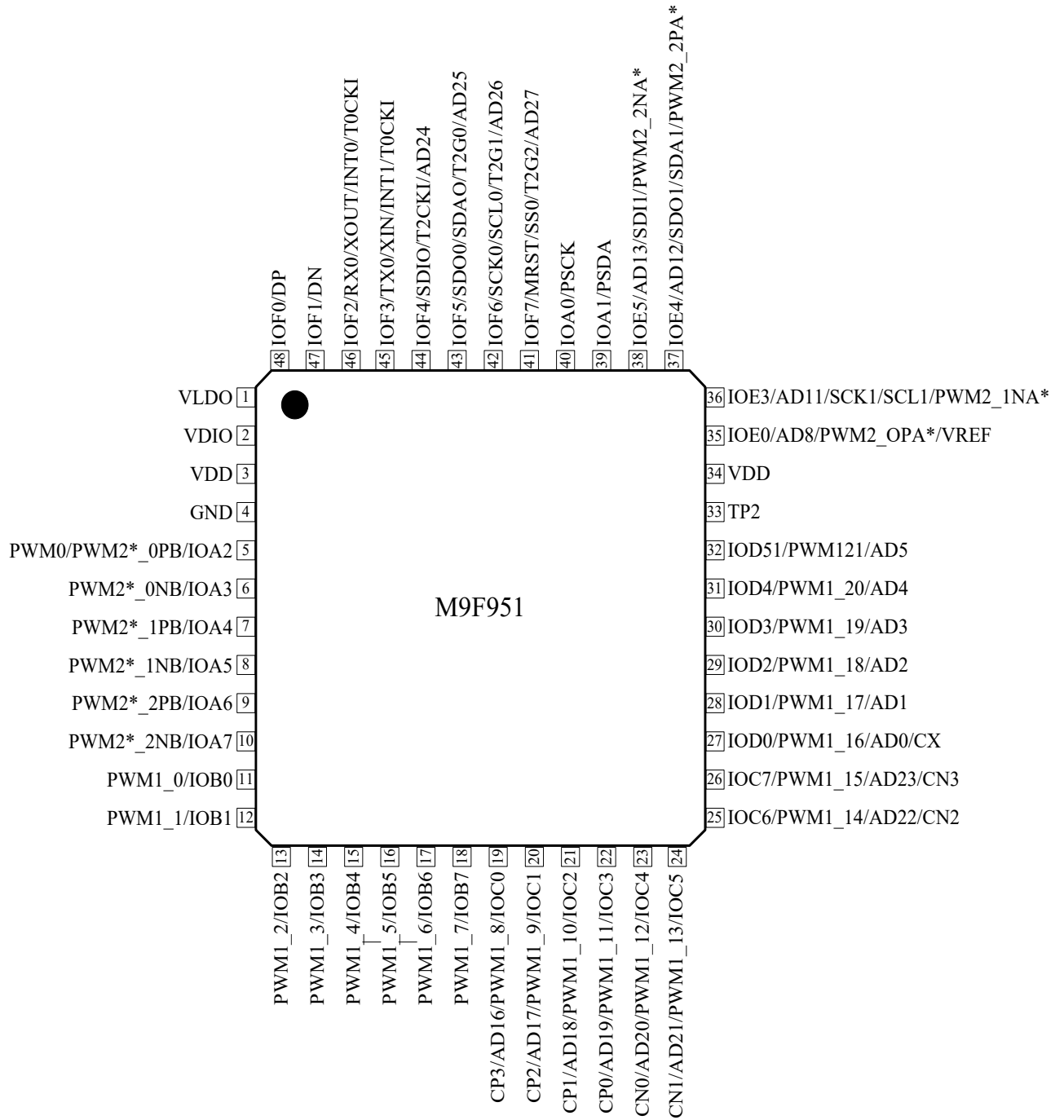
■ 封装形式

- LQFP48
- LQFP64

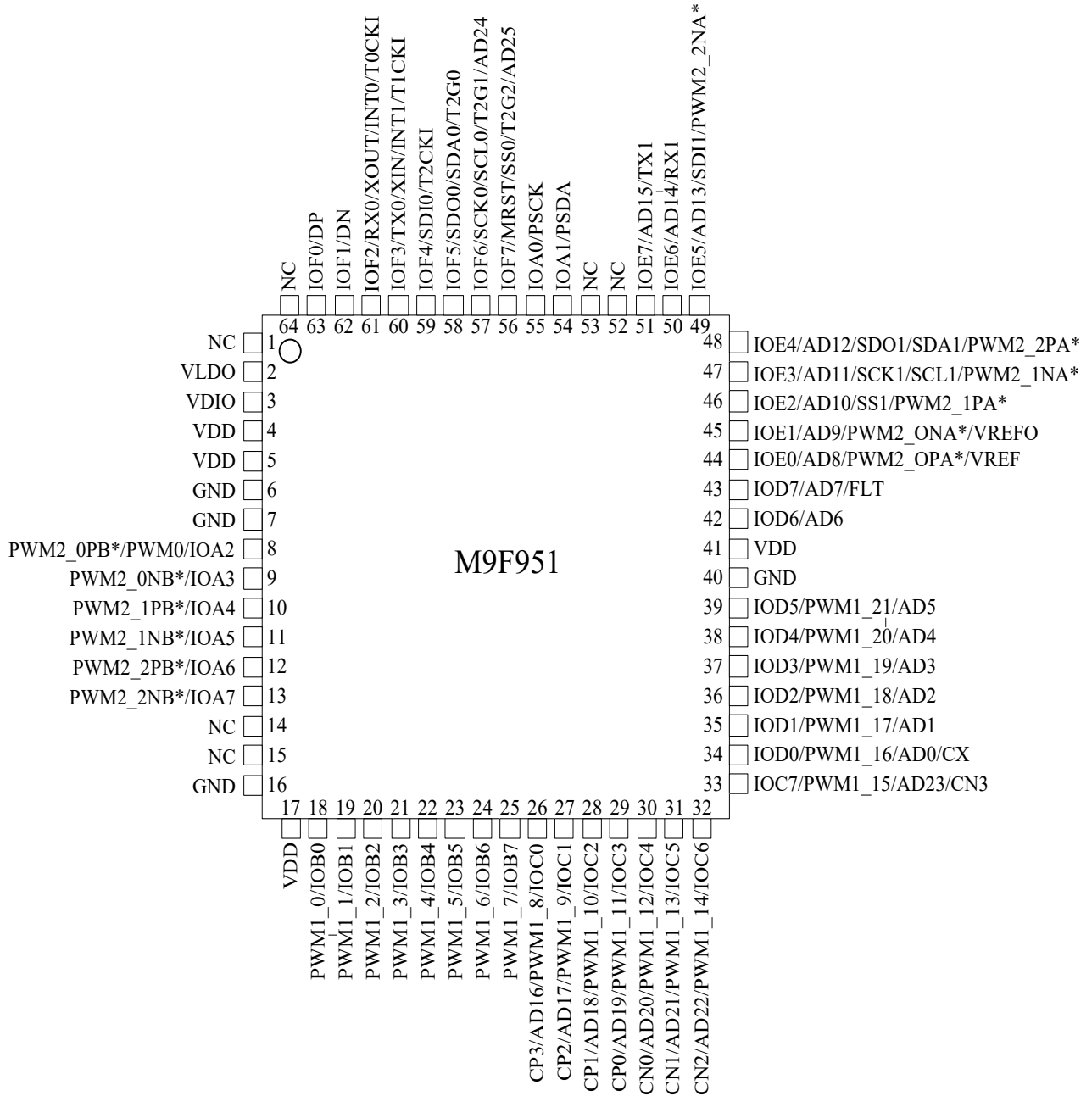
1.2 引脚图

注：芯片仿真烧录口分别是VDD、PSDA、PSCK、GND。

1.2.1 LQFP48



1.2.2 LQFP64



注：PWM2带*表示复用，具体见PWM2章节的PWM2OE寄存器描述

1.3 引脚描述

名称	类型	说明
VDD, GND	P	电源输入端
VLDO	P	3.3V 稳压输出
VDIO	P	IOF 端口电源输入
IOA[0]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
PSCK	I	编程/仿真串行时钟输入
IOA[1]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
PSDA	I/O	编程/仿真串行数据
IOA[2]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM0	O	PWM0 信号输出
PWM2_0PB*	O	PWM2 信号正相复用到 B 口输出
IOA[3]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM2_0NB*	O	PWM2 信号反相复用到 B 口输出
IOA[4]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM2_1PB*	O	PWM2 信号正相复用到 B 口输出
IOA[5]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM2_1NB*	O	PWM2 信号反相复用到 B 口输出
IOA[6]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM2_2PB*	O	PWM2 信号正相复用到 B 口输出
IOA[7]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、大 SINK 电流、变化中断
PWM2_2NB*	O	PWM2 信号反相复用到 B 口输出
IOB[0]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_0	O	PWM1 信号输出
IOB[1]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_1	O	PWM1 信号输出
IOB[2]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_2	O	PWM1 信号输出
IOB[3]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_3	O	PWM1 信号输出
IOB[4]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_4	O	PWM1 信号输出
IOB[5]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_5	O	PWM1 信号输出
IOB[6]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_6	O	PWM1 信号输出
IOB[7]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻，变化中断
PWM1_7	O	PWM1 信号输出

名称	类型	说明
IOC[0] PWM1_8 CP3 AD16	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器正端输入 ADC 外部模拟信号输入
IOC[1] PWM1_9 CP2 AD17	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器正端输入 ADC 外部模拟信号输入
IOC[2] PWM1_10 CP1 AD18	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器正端输入 ADC 外部模拟信号输入
IOC[3] PWM1_11 CP0 AD19	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器正端输入 ADC 外部模拟信号输入
IOC[4] PWM1_12 CN0 AD20	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器负端输入 ADC 外部模拟信号输入
IOC[5] PWM1_13 CN1 AD21	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器负端输入 ADC 外部模拟信号输入
IOC[6] PWM1_14 CN2 AD22	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器负端输入 ADC 外部模拟信号输入
IOC[7] PWM1_15 CN3 AD23	I/O O A A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 比较器负端输入 ADC 外部模拟信号输入
IOD[0] PWM1_16 AD0 CX	I/O O A O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入 比较器输出
IOD[1] PWM1_17 AD1	I/O O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入
IOD[2] PWM1_18 AD2	I/O O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入

名称	类型	说明
IOD[3] PWM1_19 AD3	I/O O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入
IOD[4] PWM1_20 AD4	I/O O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入
IOD[5] PWM1_21 AD5	I/O O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 PWM1 信号输出 ADC 外部模拟信号输入
IOD[6] AD6	I/O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入
IOD[7] AD7 FLT	I/O A I	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 故障检测引脚
TP2	I/O	FLASH 测试引脚
IOE[0] AD8 PWM2_0PA* VREF	I/O A O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号正相复用到 A 口输出 ADC 外部参考电压输入
IOE[1] AD9 PWM2_0NA* VREFO	I/O A O A	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号反相复用到 A 口输出 参考输出
IOE[2] AD10 PWM2_1PA* SS1	I/O A O I	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号正相复用到 A 口输出 SPI1 从机片选
IOE[3] AD11 PWM2_1NA* SCK1 SCL1	I/O A O I/O I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号反相复用到 A 口输出 SPI1 时钟 IIC 串行时钟
IOE[4] AD12 PWM2_2PA* SDO1 SDA1	I/O A O O I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号正相复用到 A 口输出 SPI1 串行数据输出 IIC 串行数据
IOE[5] AD13 PWM2_2NA* SDI1	I/O A O I	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断 ADC 外部模拟信号输入 PWM2 信号反相复用到 A 口输出 SPI1 串行数据输入

名称	类型	说明
IOE[6]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
AD14	A	ADC 外部模拟信号输入
RX1	I	UART RX
IOE[7]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
AD15	A	ADC 外部模拟信号输入
TX1	O	UART TX
IOF[0]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
DP	I/O	USB D+线
IOF[1]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
DN	I/O	USB D-线
IOF[2]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
RX0	I	UART0 RX
XOUT	A	外部晶体振荡器接口
INT0	I	外部中断 0
T0CKI	I	定时器 0 外部时钟输入
IOF[3]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
TX0	I	UART0 TX
XIN	A	外部晶体振荡器接口
INT1	I	外部中断 1
T1CKI	I	定时器 1 外部时钟输入
IOF[4]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
SDI0	I	SPI0 串行数据输入
T2CKI	I	定时器 2 外部时钟输入
AD24	A	ADC 外部模拟信号输入
IOF[5]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
SDO0	O	SPI0 串行数据输出
SDA0	I/O	IIC0 串行数据
T2G0	I	定时器 2 门控信号输入
AD25	A	ADC 外部模拟信号输入
IOF[6]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
SCK0	O	SPI0 时钟
SCL0	I/O	IIC0 串行时钟
T2G1	I	定时器 2 门控信号输入
AD26	A	ADC 外部模拟信号输入
IOF[7]	I/O	输入/输出 IO、SMT、上拉电阻、下拉电阻、变化中断
MRST	O	外部复位引脚
SS0	I/O	SPI0 从机片选
T2G2	I	定时器 2 门控信号输入
AD27	A	ADC 外部模拟信号输入

注： I = 输入， O = 输出， I/O = 输入/输出， P = 电源， A = 模拟端口

2 中央处理器（CPU）

2.1 程序存储器

地址	说明
0x0000	复位向量 0
0x0001 ~ 0x0007	用户区
0x0008 0x0018	中断向量
0x0019 ~ 0x7FFF	用户区

2.1.1 复位向量（0000h）

M9F951 有以下五种复位方式：

- 上电复位
- 看门狗复位
- 外部复位
- 欠压复位
- 软件复位(软复位指令：SRESET)

发生上述任一种复位后，程序将从 0000h 处重新开始执行，系统寄存器也将都恢复为初始默认值。

例：定义复位向量

ORG	0000H	
GOTO	Main_Program	;//跳转至用户程序开始
...		
Main_Program:		;//用户程序开始
...		
Main:		
...		
GOTO	Main	;//用户主程序循环

2.1.2 中断向量（0x0008,0x0018）

M9F951不同的中断源分配了不同的中断向量地址。一旦有中断响应，程序计数器PC的当前值就会存入堆栈缓存器并跳转到相应的中断向量地址开始执行中断服务程序。

例：中断服务程序

```

                ORG      0000h
                GOTO     Main_Program      ;//跳转到程序开始
                ORG      0008h
                GOTO     Interrupt_High    ;//发生高优先级中断后，跳转到中断子程序
                ORG      0018h
                GOTO     Interrupt_Low     ;//发生低优先级中断后，跳转到中断子程序
Main_Program:
    ...
Main:
    ...
    GOTO     Main                        ;//主程序循环

Interrupt_High:
    ...
    RETIE

Interrupt_Low:
    ...
    RETIE

    END
    
```

2.1.3 程序计数器

PCL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	PCL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **PCL[7:0]:** 程序计数器指针低字节

用户将该 PCL 作为目的操作数做加法运算时（ADDRA PCL、ADCRA PCL），15 位 PC 值参与运算，运算结果写入 PC，实现程序的相对跳转；加法运算外的其它运算时，仅 PCL 参与运算，PCH 保持不变。PCH 不可寻址。

PCLATH 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCLATH	-	PCLATH[6:0]						
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	0	0	0	0	0	0	0

Bit[5:0] **PCLATH[6:0]**: 程序计数器指针高位锁存

程序计数器（Program Counter, PC）指定欲取出执行的指令的地址。PC 为15 位宽，保存在两个不同的8 位寄存器中。存储低字节的寄存器称为PCL 寄存器，该寄存器可读写。存储高字节的寄存器，即PCH 寄存器，存储PC<14:8> 位；该寄存器不可直接读写。更新PCH寄存器的操作是通过PCLATH 寄存器实现的。PCLATH 的内容通过执行写PCL 的任何操作被传送到程序计数器。PC 是按字节寻址程序存储器的。为了防止PC 不能正确获取字指令，需要将PCL 的最低有效位固定取值为0。PC 每次加2 来寻址程序存储器中的顺序指令。CALL、RCALL、GOTO 和程序跳转指令直接写入程序计数器。对于这些指令，PCLATH 的内容不会传送到程序计数器。

2.1.4 堆栈

STKPTR 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STKPTR	STKUOV	STKDOV	-	STKPTR[4:0]				
读/写	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
复位后	0	0	-	0	0	0	0	0

Bit 7 **STKUOV**: 堆栈向上溢出

0 = 堆栈未向上溢出（软件清零或复位清零）

1 = 堆栈上溢出

Bit 6 **STKDOV**: 堆栈向下溢出

0 = 堆栈未向下溢出（软件清零或复位清零）

1 = 堆栈下溢出

Bit [4:0] **STKPTR[4:0]**: 堆栈指针

TOSL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOSL	TOS[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **TOS[7:0]**: 栈顶寄存器低七位

TOSH 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOSH	TOS[13:8]							
读/写			R/W	R/W	R/W	R/W	R/W	R/W
复位后			0	0	0	0	0	0

Bit[5:0] **TOS[13:8]**: 栈顶寄存器高六位

返回地址堆栈允许最多31 个程序调用和中断的任意组合。当执行CALL、RCALL 指令或响应中断时，PC 值会被压入该堆栈。而在执行RETURN、RETIA或RETIE指令时，PC 值会从堆栈弹出。PCLATH 不受RETURN 或CALL 指令的影响。通过16 位的RAM 和一个5 位的堆栈指针STKPTR 来实现31 字的堆栈操作。堆栈既不占用程序存储空间，也不占用数据存储空间。堆栈指针是可读写的，并且通过栈顶（TOS）特殊文件寄存器可以读写栈顶地址。也可以使用这些寄存器将数据压入堆栈或者从堆栈弹出。

执行CALL 类型指令进行压栈操作：首先堆栈指针加1，并且将PC（PC 已经指向CALL 之后的下一条指令）的内容写入堆栈指针所指向的地址单元。执行RETURN 类型指令进行出栈操作：STKPTR 寄存器所指向的地址单元的内容会被传送给PC，然后堆栈指针减1。所有复位后，堆栈指针均会初始化为00000。堆栈指针值00000 不指向任何RAM 单元；它仅仅是一个复位值。状态位指示堆栈是已满、上溢还是下溢。

有两个寄存器TOSH:TOSL 用于保存由STKPTR 寄存器所指向的堆栈单元的内容。这可以让用户在必要时实现软件堆栈。在CALL、RCALL 或中断后，软件可以通过读取TOSH:TOSL寄存器来读取压入堆栈的值。这些值可以被存放到由用户定义的软件堆栈。返回时，软件将这些值存回TOSH:TOSL并执行返回。为防止意外的堆栈操作，访问堆栈时用户必须禁止全局中断允许（GIE）位。

STKPTR 寄存器包含堆栈指针值、STKUOV（堆栈满）状态位和STKDOV（堆栈下溢）状态位。堆栈指针值可为0 至32范围内的任意值。向堆栈压入值前，堆栈指针加1；而从堆栈弹出值后，堆栈指针减1。复位时，堆栈指针值为零。用户可以读写堆栈指针的值。向堆栈压入PC 值32 次（且没有值从堆栈弹出）后，STKUOV 位置1。通过软件或POR 将STKUOV 位清零。通过软件或POR将STKUOV位清零。压栈操作会覆盖第31次压栈的值，并且STKPTR将保持为31。。当堆栈弹出次数足够卸空堆栈时，下一次出栈会向PC返回一个零值，并将STKUOV 位置1，而堆栈指针则保持为零。STKUOV 位将保持置1，直到由软件清零或发生上电复位为止。

2.1.5 SPUSH 和 SPOP 指令

由于栈顶是可以读写的，因此将值压入堆栈或从堆栈弹出而不影响程序的正常执行是非常理想的。指令集包含两条指令SPUSH 和SPOP，使用这两条指令可在软件控制下对TOS 执行操作。然后就可以修改TOSH 和TOSL，将数据或返回地址压入堆栈。SPUSH 指令将当前的PC 值压入堆栈。执行该指令会使堆栈指针加1 并将当前的PC 值装入堆栈。SPOP 指令通过将堆栈指针减1 来放弃当前的TOS 值。然后前一个入栈的值就成为了TOS 值。

2.1.6 快速寄存器堆栈

M9F951为STATUS、AREG 和BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器的堆栈只有一级且不可读写。当处理器转入中断向量处执行时，它装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用RETIE, FAST 指令从中断返回，这些寄存器中的值就会被装回相关的寄存器。如果同时允许低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储在堆栈寄存器中的值将被覆盖。在为低优先级中断提供服务时，用户必须用软件保存关键寄存器的值。如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复STATUS、AREG和BSR 寄存器。要在子程序调用中使用快速寄存器堆栈，必须执行CALL label, FAST 指令将STATUS、AREG 和BSR 寄存器的内容存入快速寄存器堆栈。然后执行RETURN, FAST 指令，从快速寄存器堆栈恢复这些寄存器。

2.1.7 查表

利用RDT指令可以读取程序区数据，TBLPTRH和TBLPTRL组成16位表地址，读取数据到TABLAT寄存器。

例 1: 查找 ROM 地址为“DTAB”的值

MOVIA	0	;//要查的数据在表中的位置
ADDIA	LOW(DTAB)	;//获取数据表地址低位
MOVAR	TBLPTRH	;//设置数据表低位指针
MOVIA	0	;//要查的数据在表中的位置
ADCIA	HIGH(DTAB)	;//获取数据表地址高位
MOVAR	TBLPTRL	;//设置数据表高位指针
		;//若需读取表的其它数据,修改指针
RDT		;//读取表的第一个数据0x01
MOVR	TABLAT,0	
...		
DTAB:		
DB	0x01,0x02,0x03,0x04,0x05,0x06,0x07.....	

2.2 数据存储器

2.2.1 数据存储器结构

数据存储器是用静态 RAM 实现的。在数据存储器中，每个寄存器都有 12 位地址，允许数据存储器实现为最大 4096 个字节。存储空间最多被分为 16 个存储区，每个存储区包含 256 个字节。数据存储器由特殊功能寄存器（Special Function Register, SFR）和通用寄存器（General Purpose Register, GPR）组成。SFR 用于单片机和外设功能模块的控制和状态指示，GPR 则用于用户应用程序的数据存储和中间结果暂存。任何未实现的存储单元均读为 0。指令集和架构支持跨所有存储区的操作。可以通过直接、间接寻址模式访问整个数据存储器。确保能在一个周期中访问常用寄存器（SFR 和某些 GPR），本器件实现了一个快速操作存储区。该存储区是一个 256 字节的存储空间，它可实现对 SFR 和 GPR Bank 0 的低地址单元的快速访问，而无需使用存储区选择寄存器（Bank Select Register, BSR）。

2.2.2 BSR 页面选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BSR	-	-	-	-	BSR [3:0]			
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit[3:0] **BSR[3:0]**：直接寻址的页面选择

容量较大的数据存储器需要高效的寻址机制，以便对所有地址进行快速访问。理想状况下，这意味着不必为每次读写操作提供完整地址。本器件是使用 RAM 存储区分区机制实现快速访问的。这种机制将存储空间分成连续的 16 个 256 字节的存储区。根据不同的指令，可以通过完整的 12 位地址，或通过 8 位的低字节地址和 4 位存储区指针直接寻址每个存储单元。指令集中的大部分指令都使用存储区指针，也就是存储区选择寄存器（BSR）。BSR 保存单元地址的高 4 位，而指令本身则包括单元地址的低 8 位。只使用 BSR 的低 4 位（BSR<3:0>），不使用高 4 位；高 4 位总是读为 0 且不能被写入。可以通过使用 BANKBSR 指令直接装入 BSR。BSR 的值代表数据存储器中的存储区；指令中的 8 位指向存储区中的存储单元，可以将它看作距离存储区下边界的偏移量。由于最多可有 16 个寄存器共享同一个低位地址，用户必须非常小心以确保在执行数据读或写之前选择了正确的存储区。

当选择存储区时，只有已实现的存储区才可以读写。对未实现的存储区进行的写操作将被忽略，而读这些存储区会返回 0。虽然是这样，这些操作仍然会对 STATUS 寄存器起作用，就好像操作成功了一样。指令中只有 MOVRR 指令指定源寄存器和目标寄存器的完整 12 位地址。该指令在执行时完全忽略 BSR。所有其他指令仅包含作为操作数的低位地址，而且必须使用 BSR 或快速操作存储区来寻址目标寄存器。

2.2.3 快速操作寄存器

使用BSR和嵌入的8位地址，用户可以寻址数据存储器的整个空间，但这同时也意味着用户必须始终确保选择了正确的存储区。否则，可能会从错误的单元读取数据或将数据写入错误的单元。如果本来是向GPR进行写操作，却将结果写入了SFR，后果是非常严重的。但是在每次对数据存储器进行读或写操作时验证和/或更改BSR会严重影响工作效率。为了提高访问大多数常用数据存储单元的效率，现为数据存储器配置了快速操作存储区，这样可以允许用户访问被映射的存储区而无需指定BSR。快速操作存储区由Bank 0的前96个字节（00h-5Fh）和Bank 15的后160个字节（60h-FFh）组成。地址较低的部分被称为“快速操作RAM”，由GPR组成。地址较高的部分则被映射为器件的SFR。这两个区域被连续地映射到快速操作存储区并且可以用一个8位地址进行线性寻址。包括快速操作RAM位（指令中的“a”参数）的指令使用快速操作存储区。当“a”等于1时，指令使用BSR和包含在操作码中的8位地址对数据存储器寻址。当“a”为0时，强制指令使用快速操作存储区地址映射，此时完全忽略BSR的当前值。此“强制”寻址模式可使指令在一个周期内对数据地址进行操作，而不需要首先更新BSR。这意味着用户可以更高效地对8位地址为60h或以上的SFR进行取值和操作。地址为60h以下的快速操作RAM非常适合于存储那些用户可能需要快速访问的数据值，如直接计算结果或常用程序变量。快速操作RAM也可实现更加快速和高效的现场保护和变量切换代码。

2.2.4 数据存储器寻址模式

本器件主要有三种寻址模式：立即数寻址，直接寻址，间接寻址。

立即数寻址主要是 MOVRR 指令时的寻址。

直接寻址：给出八位地址，根据 BSR 或者快速操作寄存器确定地址。

间接寻址有三组12位数据指针，FSR0，FSR1，FSR2。

寻址模式	地址组成
立即数寻址	INST[11:0]
直接寻址	BSR0[4:0],INST[7:0]
间接寻址 0	FSR0H,FSR0L
间接寻址 1	FSR1H,FSR1L
间接寻址 2	FSR2H,FSR2L

页	地址	用途	注
BANK0	0x000-0x007	GPR	快速操作寄存器
	0x008	保留	-
	0x009-0x05F	GPR	快速操作寄存器
	0x060-0x0FF	GPR	-
BANK1	0x100-0x1FF	GPR	-
BANK2	0x200-0x2FF	GPR	-
BANK3	0x300-0x3FF	GPR	-
BANK4	0x400-0x4FF	GPR	-
BANK5	0x500-0x5FF	GPR	-
BANK6	0x600-0x6FF	GPR	-
BANK7	0x700-0x77F	GPR	-
	0x780-0x7FF	FLASHBUFF	-
BANK8	0x800-0x8FF	USBBUF	-
BANK9	0x900-0x9FF	-	未实现，读为 0
BANK10	0xA00-0xAFF	-	未实现，读为 0
BANK11	0xB00-0xBFF	-	未实现，读为 0
BANK12	0xC00-0xCFF	-	未实现，读为 0
BANK13	0xD00-0xDFF	-	未实现，读为 0
BANK14	0xE00-0xEFF	SFR	-
BANK15	0xF00-0xF5F	SFR	-
	0xF60-0xFFFF	SFR	快速操作寄存器

2.2.5 间接寻址 0/1/2

FSRxL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSRxL	FSRxL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **FSRxL[7:0]:** FSRx 寄存器低七位地址 (x = 0/1/2)

FSRxH 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSRxH	-	-	-	-	FSRxH[3:0]			
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit [3:0] **FSRxH[3:0]:** FSR0 寄存器高四位地址 (x = 0/1/2)

注: INDFx, POSTINCx, POSTDECx, PREINCx, PLUSWx 不是实际存在的寄存器, 对这些寄存器的操作实际上是对以 FSRx 为地址的寄存器的操作。 (x = 0/1/2, 下同)

访问 INDFx 寄存器时, 实现间接寻址 x, 访问到的是 {FSRxH:FSRxL} 寄存器值作为数据指针所指向的寄存器内容, 间接寻址模式 x 可寻址所有页面空间。

访问 POSTINCx 寄存器时, 实现间接寻址模式 x, 访问到的是 {FSRxH:FSRxL} 寄存器值作为数据指针所指向的寄存器内容, 访问后, FSRx 的值自加一, 间接寻址模式 x 可寻址所有页面空间。

访问 POSTDECx 寄存器时, 实现间接寻址模式 x, 访问到的是 {FSRxH:FSRxL} 寄存器值作为数据指针所指向的寄存器内容, 访问后, FSRx 的值自减一, 间接寻址模式 x 可寻址所有页面空间。

访问 PREINCx 寄存器时, 实现间接寻址模式 x, 访问前, FSRx 的值自加一, 访问到的是 {FSRxH:FSRxL} 寄存器值作为数据指针所指向的寄存器内容, 间接寻址模式 x 可寻址所有页面空间。

访问 PLUSWx 寄存器时, 实现间接寻址模式 x, 访问到的是 {FSRxH:FSRxL}+AREG 寄存器值作为数据指针所指向的寄存器内容, 间接寻址模式可寻址所有页面空间。

2.2.6 8x8 硬件乘法器

本器件包含一个8*8 硬件乘法器（是ALU的一部分）。该乘法器可在一个指令周期内执行无符号运算并产生一个16位运算结果，该结果存储在—对乘积寄存器{PRODH:PRODL}中。该乘法器执行的运算不会影响STATUS 寄存器中的任何标志。

PRODL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRODL	PRODL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **PRODL[7:0]:** 乘法结果寄存器低八位

PRODH 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRODH	PRODH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **PRODH[7:0]:** 乘法结果寄存器高八位

2.2.7 STATUS 状态寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	-	-	-	N	OV	Z	DC	C
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	x	x	x	x	x

Bit 4 **N:** 负标志，此位用于有符号的算术运算（以二进制补码方式进行），它可以表示结果是否为负（ALU MSB = 1）

0 = 运算结果为正

1 = 运算结果为负

Bit 3 **OV:** 溢出标志，此位用于有符号的算术运算（以二进制补码方式进行），它表明运算结果溢出了7位二进制数的，范围溢出导致符号位（bit 7）发生改变。

0 = 运算未溢出

1 = 运算结果溢出

Bit 2 **Z:** 零标志

0 = 算术/逻辑运算的结果非零

1 = 算术/逻辑运算的结果为零

Bit 1 **DC:** 辅助进位标志

0 = 加法运算时低四位没有进位，减法时有向高四位借位

1 = 加法运算时低四位有进位，减法时没有向高四位借位

Bit 0 **C:** 进位标志

0 = 加法运算后没有进位，减法时有借位

1 = 加法运算后有进位，减法时没有借位

2.2.8 系统寄存器定义

数据寄存器映射表 (F60h – FFFh)									
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
FFFh	-	FDFh	INDF2	FBFh	OED	F9Fh	T0CR	F7Fh	I2C0CR1
FFEh	TOSH	FDEh	POSTINC2	FBEh	OEE	F9Eh	TC0CH	F7Eh	I2C0CR2
FFDh	TOSL	FDDh	POSTDEC2	FBDh	OEF	F9Dh	TC0CL	F7Dh	I2C0OAR
FFCh	STKPTR	FDCh	PREINC2	FBCh	-	F9Ch	PWM0CR	F7Ch	I2C0PRESC
FFBh	-	FDBh	PLUSW2	FBBh	-	F9Bh	PWM0D	F7Bh	I2C0DBUF
FFAh	PCLATH	FDAh	FSR2H	FBAh	-	F9Ah	T1CR	F7Ah	I2C0SR
FF9h	PCL	FD9h	FSR2L	FB9h	IOAOR	F99h	TC1PRH	F79h	USBCON
FF8h	-	FD8h	STATUS	FB8h	IOBOR	F98h	TC1PRL	F78h	UDADR
FF7h	TBLPTRH	FD7h	INTCR0	FB7h	IOCOR	F97h	TC1CH	F77h	UE0CON
FF6h	TBLPTL	FD6h	INTCR1	FB6h	IODOR	F96h	TC1CL	F76h	UE0IOPS
FF5h	TABLAT	FD5h	INTCR2	FB5h	IOEOR	F95h	T2CR	F75h	UE1CON
FF4h	PRODH	FD4h	UINTCR0	FB4h	IOFOR	F94h	TC2GCR	F74h	UE1IOPS
FF3h	PRODL	FD3h	INTF0	FB3h	-	F93h	TC2PRH	F73h	UE2CON
FF2h	OPTION	FD2h	INTF1	FB2h	-	F92h	TC2PRL	F72h	UE2IOPS
FF1h	EECON1	FD1h	INTF2	FB1h	-	F91h	TC2CH	F71h	UE3CON
FF0h	EECON2	FD0h	-	FB0h	ANSA	F90h	TC2CL	F70h	UE3IOPS
FEFh	INDF0	FCFh	INTP0	FAFh	ANSB	F8Fh	TX0CR	F6Fh	UE4CON
FEeh	POSTINC0	FCEh	INTP1	FAeh	ANSC	F8Eh	RX0CR	F6Eh	UE4IOPS
FEDh	POSTDEC0	FCDh	INTP2	FADh	ANSD	F8Dh	BRGD0H	F6Dh	UE5CON
FECh	PREINC0	FCCh	UINTP	FACH	ANSE	F8Ch	BRGD0L	F6Ch	UE5IOPS
FEbh	PLUSW0	FCBh	IOA	FABh	ANSF	F8Bh	TX0REG	F6Bh	UCLKCON
FEAh	FSR0H	FCAh	IOB	FAAh	-	F8Ah	RX0REG	F6Ah	-
FE9h	FSR0L	FC9h	IOC	FA9h	-	F89h	-	F69h	-
FE8h	AREG	FC8h	IOD	FA8h	-	F88h	-	F68h	-
FE7h	INDF1	FC7h	IOE	FA7h	ADCON0	F87h	-	F67h	-
FE6h	POSTINC1	FC6h	IOF	FA6h	ADCON1	F86h	-	F66h	-
FE5h	POSTDEC1	FC5h	-	FA5h	ADCON2	F85h	-	F65h	-
FE4h	PREINC1	FC4h	-	FA4h	-	F84h	-	F64h	-
FE3h	PLUSW1	FC3h	-	FA3h	ADH	F83h	-	F63h	-
FE2h	FSR1H	FC2h	OEA	FA2h	ADL	F82h	-	F62h	-
FE1h	FSR1L	FC1h	OEB	FA1h	WDTC	F81h	FLIPCR	F61h	-
FE0h	BSR	FC0h	OEC	FA0h	OSCM	F80h	PCON	F60h	-

数据寄存器映射表 (EC0h – F5Fh)									
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
F5Fh	PUA	F3Fh	IOAODS0	F1Fh	TX1CR	EFFh	PWM2CR1	EDFh	CMPC0
F5Eh	PUB	F3Eh	IOBODS0	F1Eh	RX1CR	EFEh	PWM2CR2	EDEh	CMPC1
F5Dh	PUC	F3Dh	IOCODS0	F1Dh	BRGD1H	EFDh	PMANUALCR0	EDDh	CMPC2
F5Ch	PUD	F3Ch	IODODS0	F1Ch	BRGD1L	EFCh	PMANUALCR1	EDCh	-
F5Bh	PUE	F3Bh	IOEODS0	F1Bh	TX1REG	EFBh	PWM2DT0H	EDBh	-
F5Ah	PUF	F3Ah	IOFODS0	F1Ah	RX1REG	EFAh	PWM2DT0L	EDAh	-
F59h	-	F39h	-	F19h	I2C1CR1	EF9h	PWM2DT1H	ED9h	-
F58h	-	F38h	-	F18h	I2C1CR2	EF8h	PWM2DT1L	ED8h	-
F57h	-	F37h	-	F17h	I2C1OAR	EF7h	FLTCR	ED7h	-
F56h	PDA	F36h	IOAODS1	F16h	I2C1PRESC	EF6h	PWM2INTCR	ED6h	-
F55h	PDB	F35h	IOBODS1	F15h	I2C1DBUF	EF5h	PWM2INTF	ED5h	-
F54h	PDC	F34h	IOCODS1	F14h	I2C1SR	EF4h	PWM2RLDEN	ED4h	-
F53h	PDD	F33h	IODODS1	F13h	-	EF3h	PWM2_0PH	ED3h	-
F52h	PDE	F32h	IOEODS1	F12h	-	EF2h	PWM2_0PL	ED2h	-
F51h	PDF	F31h	IOFODS1	F11h	-	EF1h	PWM2_1PH	ED1h	-
F50h	-	F30h	-	F10h	-	EF0h	PWM2_1PL	ED0h	-
F4Fh	-	F2Fh	-	F0Fh	IRCCAL	EEFh	PWM2_2PH	ECFh	-
F4Eh	-	F2Eh	-	F0Eh	IRCCAH	EEEh	PWM2_2PL	ECEh	-
F4Dh	IOAICR	F2Dh	IOAIPS1	F0Dh	-	EEDh	PWM2_0NH	ECDh	-
F4Ch	IOBICR	F2Ch	IOBIPS1	F0Ch	-	EECh	PWM2_0NL	ECCh	-
F4Bh	IOCICR	F2Bh	IOCIPS1	F0Bh	-	EEBh	PWM2_1NH	ECBh	-
F4Ah	IODICR	F2Ah	IODIPS1	F0Ah	-	EEAh	PWM2_1NL	ECAh	-
F49h	IOEICR	F29h	IOEIPS1	F09h	CALLOCK	EE9h	PWM2_2NH	EC9h	-
F48h	IOFICR	F28h	IOFIPS1	F08h	WDTCR	EE8h	PWM2_2NL	EC8h	-
F47h	-	F27h	-	F07h	-	EE7h	PWM2CR0	EC7h	-
F46h	-	F26h	-	F06h	-	EE6h	PWM2OE	EC6h	-
F45h	-	F25h	-	F05h	-	EE5h	PWM2PRH	EC5h	-
F44h	-	F24h	-	F04h	-	EE4h	PWM2PRL	EC4h	-
F43h	-	F23h	-	F03h	-	EE3h	-	EC3h	-
F42h	-	F22h	-	F02h	-	EE2h	-	EC2h	-
F41h	-	F21h	-	F01h	-	EE1h	-	EC1h	-
F40h	-	F20h	-	F00h	-	EE0h	-	EC0h	-

数据寄存器映射表 (E20h –EBFh)									
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
EBFh	PWM100CR	E9Fh	PWM105DH	E7Fh	PWM121DH	E5Fh	-	E3Fh	-
EBEh	PWM101CR	E9Eh	PWM105DL	E7Eh	PWM121DL	E5Eh	-	E3Eh	-
EBDh	PWM102CR	E9Dh	PWM106DH	E7Dh	-	E5Dh	-	E3Dh	-
EBCh	PWM103CR	E9Ch	PWM106DL	E7Ch	-	E5Ch	-	E3Ch	-
EBBh	PWM104CR	E9Bh	PWM107DH	E7Bh	-	E5Bh	-	E3Bh	-
EBAh	PWM105CR	E9Ah	PWM107DL	E7Ah	-	E5Ah	-	E3Ah	-
EB9h	PWM106CR	E99h	PWM108DH	E79h	-	E59h	-	E39h	-
EB8h	PWM107CR	E98h	PWM108DL	E78h	-	E58h	-	E38h	-
EB7h	PWM108CR	E97h	PWM109DH	E77h	-	E57h	-	E37h	-
EB6h	PWM109CR	E96h	PWM109DL	E76h	-	E56h	-	E36h	-
EB5h	PWM110CR	E95h	PWM110DH	E75h	-	E55h	-	E35h	-
EB4h	PWM111CR	E94h	PWM110DL	E74h	-	E54h	-	E34h	-
EB3h	PWM112CR	E93h	PWM111DH	E73h	-	E53h	-	E33h	-
EB2h	PWM113CR	E92h	PWM111DL	E72h	-	E52h	-	E32h	-
EB1h	PWM114CR	E91h	PWM112DH	E71h	-	E51h	-	E31h	-
EB0h	PWM115CR	E90h	PWM112DL	E70h	-	E50h	-	E30h	-
EAFh	PWM116CR	E8Fh	PWM113DH	E6Fh	-	E4Fh	-	E2Fh	-
EAEh	PWM117CR	E8Eh	PWM113DL	E6Eh	-	E4Eh	-	E2Eh	-
EADh	PWM118CR	E8Dh	PWM114DH	E6Dh	-	E4Dh	-	E2Dh	-
EACH	PWM119CR	E8Ch	PWM114DL	E6Ch	-	E4Ch	-	E2Ch	-
EABh	PWM120CR	E8Bh	PWM115DH	E6Bh	-	E4Bh	-	E2Bh	-
EAAh	PWM121CR	E8Ah	PWM115DL	E6Ah	-	E4Ah	-	E2Ah	-
EA9h	PWM100DH	E89h	PWM116DH	E69h	-	E49h	-	E29h	-
EA8h	PWM100DL	E88h	PWM116DL	E68h	-	E48h	-	E28h	-
EA7h	PWM101DH	E87h	PWM117DH	E67h	-	E47h	-	E27h	-
EA6h	PWM101DL	E86h	PWM117DL	E66h	-	E46h	-	E26h	-
EA5h	PWM102DH	E85h	PWM118DH	E65h	-	E45h	-	E25h	-
EA4h	PWM102DL	E84h	PWM118DL	E64h	-	E44h	-	E24h	-
EA3h	PWM103DH	E83h	PWM119DH	E63h	-	E43h	-	E23h	-
EA2h	PWM103DL	E82h	PWM119DL	E62h	-	E42h	-	E22h	-
EA1h	PWM104DH	E81h	PWM120DH	E61h	-	E41h	-	E21h	-
EA0h	PWM104DL	E80h	PWM120DL	E60h	-	E40h	-	E20h	-

数据寄存器映射表 (77Fh)									
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
77Fh	FLASHLOCK	-	-	-	-	-	-	-	-

注：空白处为未实现地址，读为 0，无法写。

2.3 Flash 自编程（IAP）

Flash 编程只能通过页写操作，即 Flash 写操作每次都是对一个页面的地址进行写操作，每一页的大小为 128 个字节。读为自由地址读取，可直接指针操作。

2.3.1 EECON1 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON1	PRGT4	PRGT3	PRGT2	PRGT1	PRGT0	CLRPL	ERASE	WRITE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	1	0	0	0	0	0	0

Bit [7:3] **PRGT[4:0]:** 编程、擦除时间控制

Bit 2 **CLRPL:** IAP清除Flash内部缓冲区操作，IAP写入操作必须置1

0 = 不清除Flash内部缓冲区

1 = 清除Flash内部缓冲区

Bit 1 **ERASE:** IAP擦除操作

0 = 不允许IAP擦除操作

1 = 允许IAP擦除操作

Bit 0 **WRITE:** IAP写入操作

0 = 不允许IAP写入操作

1 = 允许IAP写入操作

2.3.2 EECON2 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON2	WERR	-	-	-	EELOCK3	EELOCK2	EELOCK1	-
读/写	R	-	-	-	W	W	W	-
复位后	0	-	-	-	0	0	0	-

Bit 7 **WERR:** Flash写操作出错标志，写EECON1时该位自动置0。

Bit 3 **EELOCK3:** 解锁流程1

Bit 2 **EELOCK2:** 解锁流程2

Bit 1 **EELOCK1:** 解锁流程3

2.3.3 Flash 写入缓存器

数据存储器 BANK7 中的 0x780-0x7FF 为 Flash 写入缓存器，操作方式与普通 RAM 方式相同。Flash 写入缓存器暂用通用寄存器的 RAM。

2.3.4 Flash 操作示例

例：擦 Flash 操作

BCLR	OPTION,GIE	;操作前要关中断
BSET	EECON1,1	;擦除Flash使能
MOVR	TBLPTRHBUF,A	;写高位地址
MOVAR	TBLPTRH	
MOVR	TBLPTRLBUF,A	;写低位地址
MOVAR	TBLPTRL	
BSET	EECON2,3	;Flash解锁流程1
BSET	EECON2,2	;Flash解锁流程2
BSET	EECON2,1	;Flash解锁流程3
WDT*		;写入，CPU暂停，擦除结束后再运行下一条指令
NOP		
...		

例：Flash页写操作

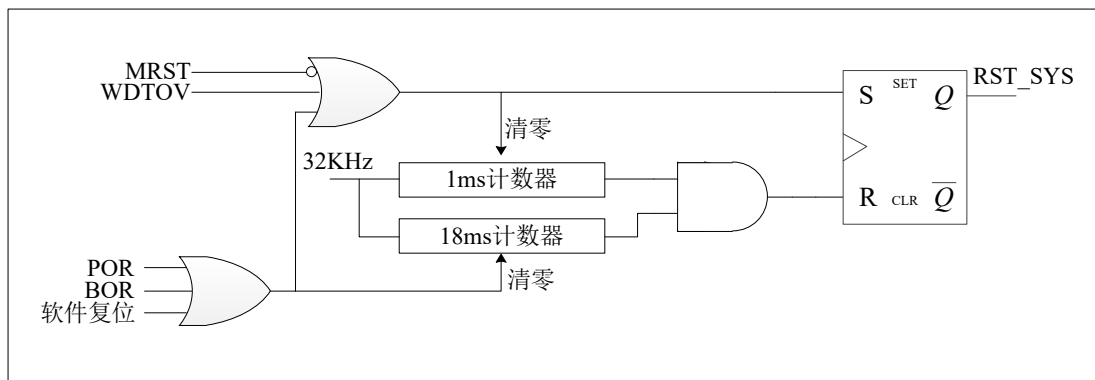
BCLR	OPTION,GIE	;操作前要关中断
MOVIA	DATA0	
MOVAR	FLASHBUF+0	
...		
MOVIA	DATA63	
MOVAR	FLASHBUF+63	;写Flash页面缓冲区64个字节
BSET	EECON1,2	;清缓存区
BSET	EECON1,0	;写Flash使能
MOVIA	HIGH(DTAB)	;写Flash页高位地址
MOVAR	TBLPTRH	
MOVIA	LOW(DTAB)	;写Flash页低位地址
MOVAR	TBLPTRL	
BSET	EECON2,3	;Flash解锁流程1
BSET	EECON2,2	;Flash解锁流程2
BSET	EECON2,1	;Flash解锁流程3
WDT*		;写入，CPU暂停，写入结束后再运行下一条指令
NOP		
...		;必要时软件校验是否写入成功

3 复位

3.1 复位方式

- 上电复位（POR）（复位时间 5V 供电下为 20ms 左右，3V 供电下 35ms 左右）
- 外部复位（MCLR Reset）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT Reset）
- 软件复位(软复位指令：SRESET)

M9F951有以上5种复位方式，任何一种复位都会使PC程序计数器清零，让程序从0000h处开始运行，并且使系统寄存器值复位。



3.2 PCON 复位状态寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	-	SOFRST	TO	PD	-	MCLR	POR	BOR
读/写	-	R/W	R	R	-	R/W	R/W	R/W
复位后	-	x	1	1	-	1	0	0

- Bit 6 **SOFRST:** 软复位
 0 = 上电复位、掉电复位，软件清零
 1 = 发生软复位
- Bit 5 **TO:** 超时位
 0 = WDT发生溢出
 1 = 上电复位或清除WDT
- Bit 4 **PD:** 掉电位
 0 = 进入休眠模式
 1 = 上电复位或清除WDT
- Bit 2 **MCLR:** 外部复位位
 0 = 发生外部复位（软件置一）
 1 = 未发生外部复位（硬件清零）
- Bit 1 **POR:** 上电复位位
 0 = 发生上电复位（软件置一）
 1 = 未发生上电复位（硬件清零）
- Bit 0 **BOR:** 欠压复位位
 0 = 发生欠压复位（软件置一）
 1 = 未发生欠压复位（硬件清零）

4 系统时钟

4.1 概述

M9F951支持双时钟系统：高速时钟和低速时钟。高速时钟由外部高速晶体振荡器（HXT）或内置高速RC振荡器（HIRC，24MHz/16MHz）提供；低速时钟由外部低速晶体振荡器（LXT，32768Hz）或内置的低速RC振荡器（LIRC，64KHz/500KHz）提供。两种时钟都可作为系统时钟源Fosc，系统工作在低速模式时，Fosc 2分频后为一个指令周期。低频系统时钟源和高频系统源可根据芯片配置字进行配置。

注：工作时勿在进行高低频切换同时 STOP CPU 操作，可能会造成系统紊乱。

4.2 OSCM 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCM	STBH	STBL	-	STOP	CLKM	STPH	LIRC500K	STPL
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	0	0	x	1	0	1

- Bit 7 **STBH**: 高频振荡器稳定标志
- Bit 6 **STBL**: 低频振荡器稳定标志
- Bit 5 **Reserved**: 保留，请保持为0
- Bit 4 **STOP**: CPU工作状态标志位
0 = CPU正常工作，所有复位唤醒
1 = CPU停止工作
- Bit 3 **CLKM**: 系统时钟状态标志位
0 = CPU运行于高频时钟
1 = CPU运行于低频时钟
- Bit 2 **STPH**: 高频振荡器控制
0 = 休眠状态或低速模式下高速振荡器仍然工作
1 = 休眠状态或低速模式下关闭高频振荡器
- Bit 1 **LIRC500K**: 低频振荡器频率选择
0 = 内部低频振荡器频率64KHz
1 = 内部低频振荡器频率500KHz
- Bit 0 **STPL**: 低频振荡器控制
0 = 休眠状态下低频振荡器仍然工作
1 = 休眠状态下低频振荡器停止工作

注：CLKM 的初始状态由配置字决定。

4.3 系统时钟的工作模式

4.3.1 普通模式

普通模式有两种分别是：

- (1) 系统时钟选择高频时钟源，STOP = 0。（电流特性参考电性参数表 I_{DD1}、I_{DD2}）
- (2) 系统时钟选择低频时钟源，STOP = 0。

4.3.2 绿色模式

绿色模式有两种分别是：

- (1) 高频时钟工作，低频时钟工作，STOP = 1。（电流特性参考电性参数表 I_{SP1}）
- (2) 高频时钟停止，低频时钟工作，STOP = 1。（电流特性参考电性参数表 I_{SP2}）

绿色模式可以由所有中断或 WDT 唤醒，此模式下只停止 CPU，外设依旧可以运行（外设所选时钟需开启）。

4.3.3 休眠模式

休眠模式只有一种是：

- (1) 高频时钟停止，低频时钟停止，STOP = 1。（电流特性参考电性参数表 I_{SP3}）

休眠模式可以由外部中断、IO 变化中断或 WDT 唤醒。

注：

- (1) 省电建议，程序运行时跑高频，快速跑完程序然后进休眠，此时休眠下需设置高频时钟停止工作。
- (2) 各工作模式的工作电流参考电性参数表。
- (3) 绿色和休眠模式下，如果总中断不开启，所有中断唤醒能唤醒芯片但是不会进中断。

4.4 IRCCAL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCCAL	-	IRCCAL[6:0]						
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	x	x	x	x	x	x	x

24MHz 高频内部 RC 振荡频率校准寄存器。复位后，初值为出厂校准值。

4.5 IRCCAH 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCCAH	-	IRCCAH[6:0]						
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	x	x	x	x	x	x	x

16MHz 高频内部 RC 振荡频率校准寄存器。复位后，初值为出厂校准值。

内置的双高频 RC 振荡器在芯片出厂后，频率将校准到 16MHz 和 24MHz，IRCCAH 和 IRCCAL 复位初值即为出厂校准值，但程序中可以通过特殊的写入流程来微调此频率以满足特定应用需求。

例： 调整 16MHz 高频内部 IRC 频率

TASK_IRCCAH:

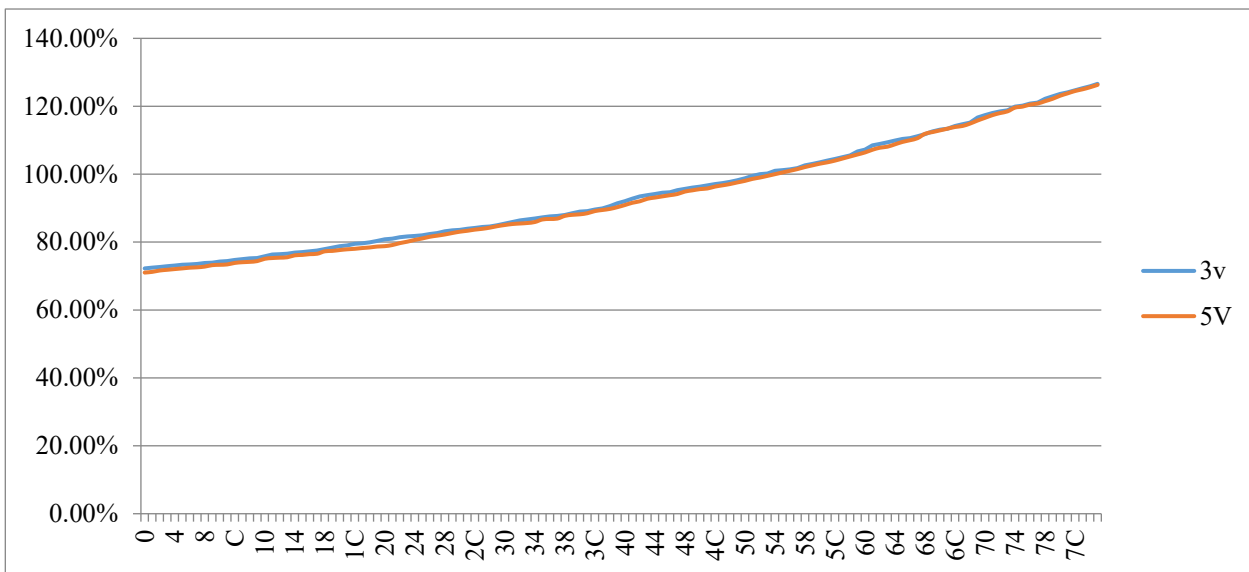
```

        BANKBSR    0x0F
        MOVIA      0x55
        MOVAR      CALLOCK          ;// CALLOCK写入0x55
        MOVIA      0xAA
        MOVAR      CALLOCK          ;// CALLOCK写入0xAA
        MOVIA      VALUE
        MOVAR      IRCCAH           ;//写入IRCCAH
        ...

```

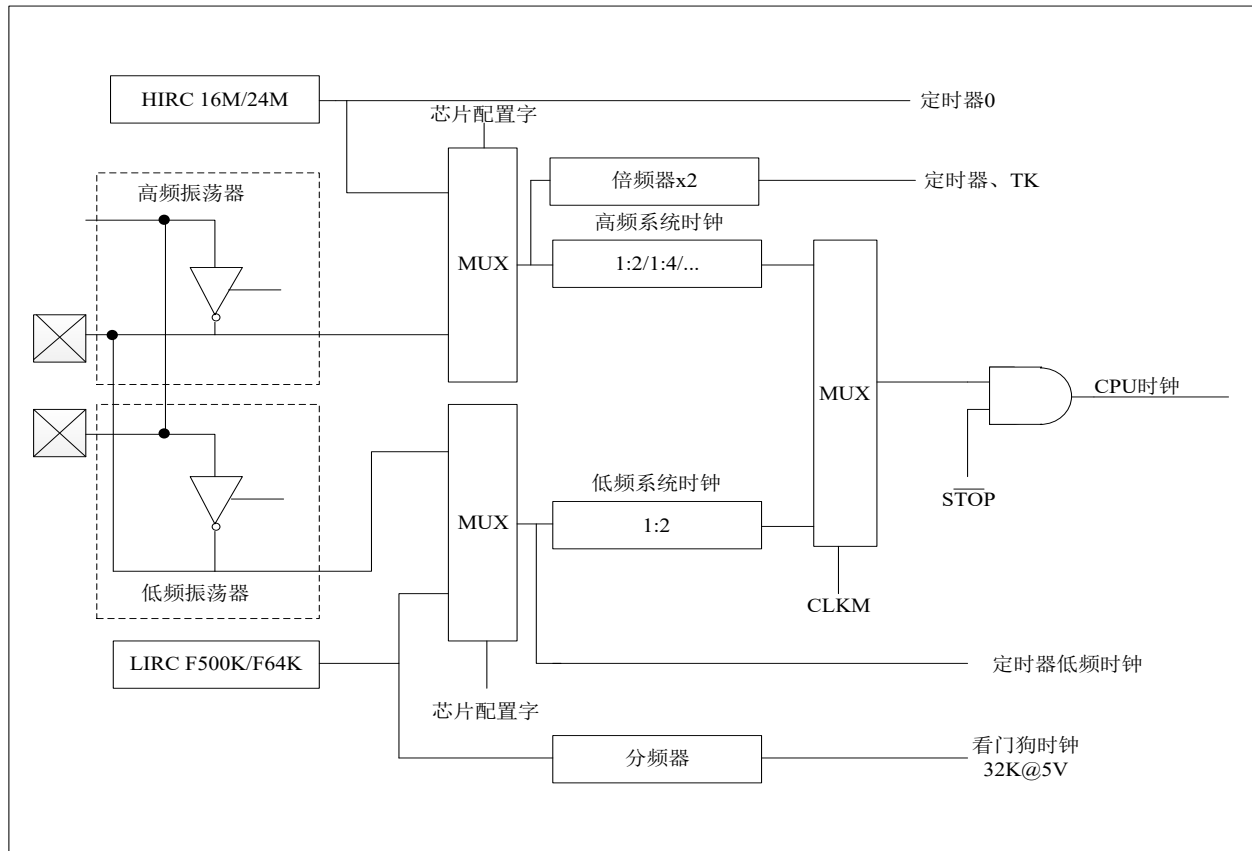
;//若需继续在IRCCAH寄存器内写入其他值需要重复以上所有步骤

HIRC 调整频率趋势图



注：具体值不做设计保证。

4.6 系统时钟结构框图



	高速运行模式 (CLKM = 0)	低速运行模式 (CLKM = 1)	休眠模式 (STOP = 1)
高频振荡器	运行	由 STPH 决定	由 STPH 决定
低频振荡器	运行	运行	由 STPL 决定
WDT	配置字决定	由配置字决定	由配置字决定
TC0/TC1	TXEN	若选择高速时钟， 需 STPH=0	高速时钟源&STPH=0 低速时钟源&STPL=0

4.7 系统时钟高低频切换

高频振荡器稳定计数器: 64 Clocks (内部 IRC 模式) / 1024 Clocks (外部高频振荡器模式)

低频振荡器稳定计数器: 16 Clocks (内部 RC 模式) / 1024 Clocks (外部低频振荡器模式)

高低频切换时间:

高频切低频: 1 个低频时钟周期+1 个高频时钟周期

低频切高频 (STBH = 0): 1 个低频时钟周期+高频振荡器起振时间+高频振荡器稳定时间

低频切高频 (STBH = 1): 1 个低频时钟周期+1 个高频时钟周期

唤醒时间:

CLKM = 0 & STBH = 0, 唤醒时间 = 高频振荡器起振时间+高频振荡器稳定时间

CLKM = 0 & STBH = 1, 唤醒时间 = 高频振荡器稳定时间

CLKM = 1 & STBL = 0, 唤醒时间 = 低频振荡器起振时间+低频振荡器稳定时间

CLKM = 1 & STBL = 1, 唤醒时间 = 低频振荡器稳定时间

5 中断

5.1 概述

M9F951可通过配置字选择位选择单优先级还是多优先级，单优先级是高优先级，所有中断默认都是高优先级（PWM2中断只有高优先级）。多优先级有两种优先级，高优先级和低优先级，通过INTPn寄存器进行控制。同级及高级优先级中断响应时不允许低级优先级中断嵌套，低级优先级中断允许高级优先级中断嵌套响应。

响应低优先级中断时，GIEL 会自动清零，同时将 GIEH/GIEL 状态压栈，执行中断时由于 GIEL 已被清零不允许响应同级中断，若 GIEH 的值是 1，则允许高优先级的中断嵌套；响应高优先级时，GIEH 和 GIEL 会被自动清零，同时将 GIEH/GIEL 状态压栈。其中 WAKE 信号可以唤醒 CPU。

所有中断信号都可以唤醒 CPU（在不使能全局中断的情况下也可以唤醒）。

注：使用外部中断 INT0、INT1 要注意，当中断触发和进休眠的操作（即 STOP 从 0 到 1）同时发生时，会导致外部中断 INT0、INT1 无效且无法唤醒休眠；
解决方法：如需使用外部中断尽量使用 IO 变化中断，如果必须使用外部中断 INT0、INT1，必须要开启 WDT 作为唤醒源。

5.2 中断向量分配

高优先级中断：0008h

低优先级中断：0018h

5.3 OPTION 配置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIEH	GIEL	-	-	MINT1[1:0]		MINT0[1:0]	
读/写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
复位后	0	0	-	-	0	0	0	0

Bit 7 **GIEH:** 全局中断高优先级中断控制位
0 = 屏蔽高优先级中断
1 = 使能高优先级中断，中断响应后自动清零

Bit 6 **GIEL:** 全局中断低优先级中断控制位
0 = 屏蔽低优先级中断
1 = 使能低优先级中断，中断响应后自动清零

Bit [3:2] **MINT1[1:0]:** INT1中断模式选择

MINT1[1:0]	INT1 中断模式选择
00	上升沿中断
01	下降沿中断
1x	变化中断

Bit [1:0] **MINT0[1:0]:** INT0中断模式选择

MINT0[1:0]	INT0 中断模式选择
00	上升沿中断
01	下降沿中断
1x	变化中断

注：同时使能 GIEL 和 GIEH，高优先中断可嵌套低优先级中断，但同级中断不可嵌套。

5.4 IO 端口变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAICR	IOAICR7	IOAICR6	IOAICR5	IOAICR4	IOAICR3	IOAICR2	IOAICR1	IOAICR0
IOBICR	IOBICR7	IOBICR6	IOBICR5	IOBICR4	IOBICR3	IOBICR2	IOBICR1	IOBICR0
IOCICR	IOCICR7	IOCICR6	IOCICR5	IOCICR4	IOCICR3	IOCICR2	IOCICR1	IOCICR0
IODICR	IODICR7	IODICR6	IODICR5	IODICR4	IODICR3	IODICR2	IODICR1	IODICR0
IOEICR	IOEICR7	IOEICR6	IOEICR5	IOEICR4	IOEICR3	IOEICR2	IOEICR1	IOEICR0
IOFICR	IOFICR7	IOFICR6	IOFICR5	IOFICR4	IOFICR3	IOFICR2	IOFICR1	IOFICR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **IOxICRy**: IOA端口变化中断使能 (x = A/B/C/D/E/F, y = 0-7)

0 = 屏蔽IOxy口电平变化中断

1 = 使能IOxy口电平变化中断

5.5 INTCR0 中断控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR0	-	-	-	CMPIE	TC2GIE	TC2IE	TC1IE	TC0IE
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

- Bit 4 **CMPIE:** 比较器中断使能位
0 = 屏蔽CMP中断
1 = 使能CMP中断
- Bit 3 **TC2GIE:** TC2门控中断使能位
0 = 屏蔽TC2门控中断
1 = 使能TC2门控中断
- Bit 2 **TC2IE:** TC2溢出中断使能位
0 = 屏蔽TC2溢出中断
1 = 使能TC2溢出中断
- Bit 1 **TC1IE:** TC1溢出中断使能位
0 = 屏蔽TC1溢出中断
1 = 使能TC1溢出中断
- Bit 0 **TC0IE:** TC0溢出中断使能位
0 = 屏蔽TC0溢出中断
1 = 使能TC0溢出中断

5.6 INTF0 中断标志寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	-	-	-	CMPIF	TC2GIF	TC2IF	TC1IF	TC0IF
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

注：除 USART 之外的所有中断标志位需软件清零。

- Bit 4 **CMPIF**: CMP中断标志
 0 = 未产生CMP中断
 1 = 产生CMP中断
- Bit 3 **TC2GIF**: TC2门控中断标志
 0 = 未产生TC2门控中断
 1 = 产生TC2门控中断
- Bit 2 **TC2IF**: TC2溢出中断标志
 0 = 未产生TC2溢出中断
 1 = 产生TC2溢出中断
- Bit 1 **TC1IF**: TC1溢出中断标志
 0 = 未产生TC1溢出中断
 1 = 产生TC1溢出中断
- Bit 0 **TC0IF**: TC0溢出中断标志
 0 = 未产生TC0溢出中断
 1 = 产生TC0溢出中断

5.7 INTP0 中断优先级寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTP0	-	-	-	CMPIP	TC2GIP	TC2IP	TC1IP	TC0IP
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

Bit 4 **CMPIP:** CMP中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 3 **TC2GIP:** TC2门控中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 2 **TC2IP:** TC2溢出中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 1 **TC1IP:** TC1溢出中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 0 **TC0IP:** TC0溢出中断优先级控制位

0 = 高优先级

1 = 低优先级

5.8 INTCR1 中断控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR1	INT1IE	INT0IE	IOFCHIE	IOECHIE	IODCHIE	IOCCHIE	IOBCHIE	IOACHIE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **INT1IE:** 外部端口中断1使能位

0 = 屏蔽外部端口中断1

1 = 使能外部端口中断1

Bit 6 **INT0IE:** 外部端口中断0使能位

0 = 屏蔽外部端口中断0

1 = 使能外部端口中断0

Bit 5 **IOFCHIE:** 端口F变化中断使能位

0 = 屏蔽端口F变化中断

1 = 使能端口F变化中断

Bit 4 **IOECHIE:** 端口E变化中断使能位

0 = 屏蔽端口E变化中断

1 = 使能端口E变化中断

Bit 3 **IODCHIE:** 端口D变化中断使能位

0 = 屏蔽端口D变化中断

1 = 使能端口D变化中断

Bit 2 **IOCCHIE:** 端口C变化中断使能位

0 = 屏蔽端口C变化中断

1 = 使能端口C变化中断

Bit 1 **IOBCHIE:** 端口B变化中断使能位

0 = 屏蔽端口B变化中断

1 = 使能端口B变化中断

Bit 0 **IOACHIE:** 端口A变化中断使能位

0 = 屏蔽端口A变化中断

1 = 使能端口A变化中断

5.9 INTF1 中断标志寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	INT1IF	INT0IF	IOFCHIF	IOECHIF	IODCHIF	IOCCHIF	IOBCHIF	IOACHIF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

注：所有中断标志位需软件清零。

- Bit 7 **INT1IF**：外部端口中断1标志
0 = 未产生外部端口中断1
1 = 产生外部端口中断1
- Bit 6 **INT0IF**：外部端口中断0标志
0 = 未产生外部端口中断0
1 = 产生外部端口中断0
- Bit 5 **IOFCHIF**：端口F变化中断标志
0 = 未产生端口F变化中断
1 = 产生端口F变化中断
- Bit 4 **IOECHIF**：端口E变化中断标志
0 = 未产生端口E变化中断
1 = 产生端口E变化中断
- Bit 3 **IODCHIF**：端口D变化中断标志
0 = 未产生端口D变化中断
1 = 产生端口D变化中断
- Bit 2 **IOCCHIF**：端口C变化中断标志
0 = 未产生端口C变化中断
1 = 产生端口C变化中断
- Bit 1 **IOBCHIF**：端口B变化中断标志
0 = 未产生端口B变化中断
1 = 产生端口B变化中断
- Bit 0 **IOACHIF**：端口A变化中断标志
0 = 未产生端口A变化中断
1 = 产生端口A变化中断

5.10 INTP1 中断优先级寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTP1	INT1IP	INT0IP	TKIP	ADIP	IODCHIP	IOCCHIP	IOBCHIP	IOACHIP
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

- Bit 7 **INT1IP:** 外部端口中断1优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 6 **INT0IP:** 外部端口中断0优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 5 **TKIP:** 触摸中断优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 4 **ADIP:** AD中断优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 3 **IODCHIP:** 端口D变化中断优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 2 **IOCCHIP:** 端口C变化中断优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 1 **IOBCHIP:** 端口B变化中断优先级控制位
0 = 高优先级
1 = 低优先级
- Bit 0 **IOACHIP:** 端口A变化中断优先级控制位
0 = 高优先级
1 = 低优先级

5.11 INTCR2 中断控制寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR1	RX1IE	TX1IE	RX0IE	TX0IE	-	ADIE	I2C1IE	I2C0IE
读/写	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
复位后	0	0	0	0	-	0	0	0

Bit 7 **RX1IE:** USART1接收中断使能位

0 = 屏蔽UART1接收中断

1 = 使能UART1接收中断

Bit 6 **TX1IE:** USART1发送中断使能位

0 = 屏蔽UART1发送中断

1 = 使能UART0发送中断

Bit 5 **RX0IE:** USART0接收中断使能位

0 = 屏蔽UART0接收中断

1 = 使能UART0接收中断

Bit 4 **TX0IE:** USART0发送中断使能位

0 = 屏蔽UART0发送中断

1 = 使能UART0发送中断

Bit 2 **ADIE:** AD转换中断使能位

0 = 屏蔽AD转换中断

1 = 使能AD转换中断

Bit 1 **I2C1IE:** I2C1或SPI1中断使能位

0 = 屏蔽I2C1或SPI1中断

1 = 使能I2C1或SPI1中断

Bit 0 **I2C0IE:** I2C0或SPI0中断使能位

0 = 屏蔽I2C0或SPI0中断

1 = 使能I2C0或SPI0中断

5.12 INTF2 中断标志寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF2	RX1IF	TX1IF	RX0IF	TX0IF	-	ADIF	I2C1IF	I2C0IF
读/写	R	R	R	R	-	R/W	R	R
复位后	0	1	0	1	-	0	0	0

注：除 USART 之外的所有中断标志位需软件清零。

- Bit 7 **RX1IF**: USART1接收中断标志
0 = 未产生USART1接收中断
1 = 产生USART1接收中断（对RX1REG进行读操作后自动清零）
- Bit 6 **TX1IF**: USART0发送中断标志
0 = 未产生USART1发送中断（数据不为空，还在发送中）
1 = 产生USART1发送中断（数据为空，可发送下一个数据）
- Bit 5 **RX0IF**: USART0接收中断标志
0 = 未产生USART0接收中断
1 = 产生USART0接收中断（对RX0REG进行读操作后自动清零）
- Bit 4 **TX0IF**: USART0发送中断标志
0 = 未产生USART0发送中断（数据不为空，还在发送中）
1 = 产生USART0发送中断（数据为空，可发送下一个数据）
- Bit 2 **ADIF**: AD转换中断标志
0 = 未产生AD转换中断
1 = 产生AD转换中
- Bit 1 **I2C1IF**: I2C1或SPI1中断标志
0 = 未产生I2C1或SPI1中断
1 = 产生I2C1或SPI1中断
- Bit 0 **I2C0IF**: I2C0或SPI0中断标志
0 = 未产生I2C0或SPI0中断
1 = 产生I2C0或SPI0中断

5.13 INTP2 中断优先级寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTP2	RX1IP	TX1IP	RX0IP	TX0IP	-	ADIP	I2C1IP	I2C0IP
读/写	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
复位后	0	0	0	0	-	0	0	0

Bit 7 **RX1IP:** UART1接收中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 6 **TX1IP:** UART1发送中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 5 **RX0IP:** UART0接收中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 4 **TX0IP:** UART0发送中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 2 **ADIP:** AD转换中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 1 **I2C1IP:** I2C1或SPI1中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 0 **I2C0IP:** I2C0或SPI0中断优先级控制位

0 = 高优先级

1 = 低优先级

5.14 USB 中断控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UINTCR0	-	UINTEP5	UINTEP 4	UINTEP 3	UINTEP 2	UINTEP 1	UINTEP 0	UINTBUS
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	0	0	0	0	0	0	0

- Bit 6 **UINTEP 5:** EP5中断使能位
0 = 屏蔽EP5中断
1 = 使能EP5中断
- Bit 5 **UINTEP 4:** EP4中断使能位
0 = 屏蔽EP4中断
1 = 使能EP4中断
- Bit 4 **UINTEP 3:** EP3中断使能位
0 = 屏蔽EP3中断
1 = 使能EP3中断
- Bit 3 **UINTEP 2:** EP2中断使能位
0 = 屏蔽EP2中断
1 = 使能EP2中断
- Bit 2 **UINTEP 1:** EP1中断使能位
0 = 屏蔽EP1中断
1 = 使能EP1中断
- Bit 1 **UINTEP 0:** EP0中断使能位
0 = 屏蔽EP0中断
1 = 使能EP0中断
- Bit 0 **UINTBUS:** BUS中断使能位
0 = 屏蔽BUS中断
1 = 使能BUS中断

注:

- (1) 使用 USB 时要把 IOF0、IOF1(即 DP、DN)设置为数字端口，否则无法进入总线中断。
- (2) 设置 IOPIS 寄存器时不能把 Bit[7:6]设置为 00（屏蔽输入功能），否则无法进入总线中断。
- (3) 使用 USB 时不要设置 IOF0、IOF1 的上、下拉。

5.15 USB 中断优先级寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UINTP	-	UINTP5	UINTP4	UINTP3	UINTP2	UINTP1	UINTP0	UINTBUSP
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	0	0	0	0	0	0	0

Bit 6 **UINTP5:** EP5中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 5 **UINTP4:** EP4中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 4 **UINTP3:** EP3中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 3 **UINTP2:** EP2中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 2 **UINTP1:** EP1中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 1 **UINTP0:** EP0中断优先级控制位

0 = 高优先级

1 = 低优先级

Bit 0 **UINTBUSP:** BUS中断优先级控制位

0 = 高优先级

1 = 低优先级

6 端口

6.1 数据寄存器 IO_x (x = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
IOB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
IOC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
IOD	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0
IOE	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	IOE0
IOF	IOF7	IOF6	IOF5	IOF4	IOF3	IOF2	IOF1	IOF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

注：如读 IOA 的值读取的是引脚电平（模拟端口设置寄存器必需关闭，默认端口是模拟使能），向 IOA 写值是写入输出寄存器。

6.2 输出锁存寄存器 IO_xOR (x = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAOR	IOAOR7	IOAOR6	IOAOR5	IOAOR4	IOAOR3	IOAOR2	IOAOR1	IOAOR0
IOBOR	IOBOR7	IOBOR6	IOBOR5	IOBOR4	IOBOR3	IOBOR2	IOBOR1	IOBOR0
IOCOR	IOCOR7	IOCOR6	IOCOR5	IOCOR4	IOCOR3	IOCOR2	IOCOR1	IOCOR0
IODOR	IODOR7	IODOR6	IODOR5	IODOR4	IODOR3	IODOR2	IODOR1	IODOR0
IOEOR	IOEOR7	IOEOR6	IOEOR5	IOEOR4	IOEOR3	IOEOR2	IOEOR1	IOEOR0
IOFOR	IOFOR7	IOFOR6	IOFOR5	IOFOR4	IOFOR3	IOFOR2	IOFOR1	IOFOR0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

Bit [7:0] IO_xOR_y: 输出锁存 (x = A/B/C/D/E/F, y = 0-7)

0 = 输出为0

1 = 输出为1

注：如对 IOAOR 寄存器的读操作是对 IOA 输出的读，对 IOAOR 寄存器的写操作是对 IOA 输出的写。

6.3 输出方向寄存器 OEx (x = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEA	OEA7	OEA6	OEA5	OEA4	OEA3	OEA2	OEA1	OEA0
OEB	OEB7	OEB6	OEB5	OEB4	OEB3	OEB2	OEB1	OEB0
OEC	OEC7	OEC6	OEC5	OEC4	OEC3	OEC2	OEC1	OEC0
OED	OED7	OED6	OED5	OED4	OED3	OED2	OED1	OED0
OEE	OEE7	OEE6	OEE5	OEE4	OEE3	OEE2	OEE1	OEE0
OEF	OEF7	OEF6	OEF5	OEF4	OEF3	OEF2	OEF1	OEF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

Bit [7:0] **OExy**: 输出/输出使能 (x = A/B/C/D/E/F, y = 0-7)

0 = 输入模式

1 = 输出模式

6.4 上拉控制寄存器 PUx (x = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUA	PUA7	PUA6	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
PUB	PUB7	PUB6	PUB5	PUB4	PUB3	PUB2	PUB1	PUB0
PUC	PUC7	PUC6	PUC5	PUC4	PUC3	PUC2	PUC1	PUC0
PUD	PUD7	PUD6	PUD5	PUD4	PUD3	PUD2	PUD1	PUD0
PUE	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
PUF	PUF7	PUF6	PUF5	PUF4	PUF3	PUF2	PUF1	PUF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

Bit [7:0] **PUxy**: 上拉使能 (x = A/B/C/D/E/F, y = 0-7)

0 = 上拉关闭

1 = 上拉使能

6.5 下拉控制寄存器 PD_x (x = A/B/C/D/E/F)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDA	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
PDB	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PDD	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0
PDE	PDE7	PDE6	PDE5	PDE4	PDE3	PDE2	PDE1	PDE0
PDF	PDF7	PDF6	PDF5	PDF4	PDF3	PDF2	PDF1	PDF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

Bit [7:0] **PD_{xy}**: 下拉使能 (x = A/B/C/D/E/F, y = 0-7)

0 = 下拉关闭

1 = 下拉使能

注：同一 IO 口上下拉电阻同时打开时，IO 口将自动屏蔽输入功能（读该端口状态为 0），此时端口电平接近于 VDD/2。

6.6 模拟端口设置寄存器 ANS_x (x = A/B/C/D)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSA	ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
ANSB	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
ANSC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0
ANSD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
ANSE	ANSE7	ANSE6	ANSE5	ANSE4	ANSE3	ANSE2	ANSE1	ANSE0
ANSF	ANSF7	ANSF6	ANSF5	ANSF4	ANSF3	ANSF2	ANSF1	ANSF0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	1	1	1	1	1	1	1	1

Bit [7:0] **ANS_{xy}**: 端口类型设置 (x = A/B/C/D/E/F, y = 0-7)

0 = 数字端口

1 = 模拟端口功能（数字输入功能被屏蔽，即端口状态MCU无法读取）

注：模拟端口模式，仅数字输入功能被屏蔽，但可输出。

6.7 驱动电流选择寄存器 IOxODS1/ IOxODS0 (x = A/B/C/D/E/F)

驱动电流选择寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAODS1	IOAODS17	IOAODS16	IOAODS15	IOAODS14	IOAODS13	IOAODS12	IOAODS11	IOAODS10
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

驱动电流选择寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAODS0	IOAODS07	IOAODS06	IOAODS05	IOAODS04	IOAODS03	IOAODS02	IOAODS01	IOAODS00
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[7:2] IOAODS1x, IOAODS0x驱动能力选择(x = 2-7)

IOAODS1x,IOAODS0x	IOAx SINK 电流选择 (VDD=5V)
00	标准驱动 A0(IOLA0\ IOHA0)
01	大驱动 A1(IOLA1\ IOHA1)
10	大驱动 A2(IOLA2\ IOHA2)
11	大驱动 A3(IOLA3\ IOHA3)

Bit[1:0] IOAODS1x, IOAODS0x驱动能力选择((x = 0-1)

IOAODS1x,IOAODS0x	IOAx 驱动电流选择 (VDD=5V)
00	标准驱动 X0(IOLX0\ IOHX0)
01	标准驱动 X1(IOLX1\ IOHX1)
10	标准驱动 X2(IOLX2\ IOHX2)
11	标准驱动 X3(IOLX3\ IOHX3)

注：各个 IO 口的驱动能力参考电性参数表。

驱动电流选择寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOBODS1	IOBODS17	IOBODS16	IOBODS15	IOBODS14	IOBODS13	IOBODS12	IOBODS11	IOBODS10
IOCODS1	IOCODS17	IOCODS16	IOCODS15	IOCODS14	IOCODS13	IOCODS12	IOCODS11	IOCODS10
IODODS1	IODODS17	IODODS16	IODODS15	IODODS14	IODODS13	IODODS12	IODODS11	IODODS10
IOEODS1	IOEODS17	IOEODS16	IOEODS15	IOEODS14	IOEODS13	IOEODS12	IOEODS11	IOEODS10
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

驱动电流选择寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOBODS0	IOBODS07	IOBODS06	IOBODS05	IOBODS04	IOBODS03	IOBODS02	IOBODS01	IOBODS00
IOCODS0	IOCODS07	IOCODS06	IOCODS05	IOCODS04	IOCODS03	IOCODS02	IOCODS01	IOCODS00
IODODS0	IODODS07	IODODS06	IODODS05	IODODS04	IODODS03	IODODS02	IODODS01	IODODS00
IOEODS0	IOEODS07	IOEODS06	IOEODS05	IOEODS04	IOEODS03	IOEODS02	IOEODS01	IOEODS00
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] IOxODS1y, IOxODS0y驱动能力选择 (x = A/B/C/D/E/F, y = 0-7)

IOxODS1y, IOxODS0y	IOxy 驱动电流选择(VDD=5V)
00	标准驱动 X0(IOLX0\ IOHX0)
01	标准驱动 X1(IOLX1\ IOHX1)
10	标准驱动 X2(IOLX2\ IOHX2)
11	标准驱动 X3(IOLX3\ IOHX3)

注：各个 IO 口的驱动能力参考电性参数表。

驱动电流选择寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOFODS1	IOFODS17	IOFODS16	IOFODS15	IOFODS14	IOFODS13	IOFODS12	IOFODS11	IOFODS10
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

驱动电流选择寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOFODS0	IOFODS07	IOFODS06	IOFODS05	IOFODS04	IOFODS03	IOFODS02	IOFODS01	IOFODS00
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit[2:7] IOFODS1x, IOFODS0x驱动能力选择(x = 2-7)

IOFODS1x,IOFODS0x	IOFx 驱动电流选择 (VDD=5V)
00	标准驱动 F0(IOLF0\ IOHF0)
01	标准驱动 F1(IOLF1\ IOHF1)
10	标准驱动 F2(IOLF2\ IOHF2)
11	标准驱动 F3(IOLF3\ IOHF3)

Bit[0:1] IOFODS1x, IOFODS0x驱动能力选择(x = 0-1)

IOFODS1x,IOFODS0x	IOFx 驱动电流选择 (VDD=5V)
00	标准驱动 F4(IOLF4\ IOHF4)
01	标准驱动 F5(IOLF5\ IOHF5)
10	标准驱动 F6(IOLF6\ IOHF6)
11	标准驱动 F7(IOLF7\ IOHF7)

注：各个 IO 口的驱动能力参考电性参数表。

6.8 翻转电平设置寄存器 IOxIPS/FLIPCR (x = A/B/C/D/E/F)

翻转控制寄存器 IOxIPS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAIPS1	IOAIPS17	IOAIPS16	IOAIPS15	IOAIPS14	IOAIPS13	IOAIPS12	IOAIPS11	IOAIPS10
IOBIPS1	IOBIPS17	IOBIPS16	IOBIPS15	IOBIPS14	IOBIPS13	IOBIPS12	IOBIPS11	IOBIPS10
IOCIPS1	IOCIPS17	IOCIPS16	IOCIPS15	IOCIPS14	IOCIPS13	IOCIPS12	IOCIPS11	IOCIPS10
IODIPS1	IODIPS17	IODIPS17	IODIPS15	IODIPS14	IODIPS13	IODIPS12	IODIPS11	IODIPS10
IOEIPS1	IOEIPS17	IOEIPS16	IOEIPS15	IOEIPS14	IOEIPS13	IOEIPS12	IOEIPS11	IOEIPS10
IOFIPS1	IOFIPS17	IOFIPS17	IOFIPS15	IOFIPS14	IOFIPS13	IOFIPS12	IOFIPS11	IOFIPS10
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	1	1	1	1	1	1	1	1

Bit [7:0] **IOxIPS1y**: 翻转电平选择 (x = A/B/C/D/E/F, y = 0-7)

0 = SMT

1 = 1.5 V 或 1/2VDD (由FLIPCR寄存器控制)

翻转控制寄存器 FLIPCR

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLIPCR	-	-	-	-	-	-	-	FLIPCR0
读/写	-	-	-	-	-	-	-	R/W
复位后	-	-	-	-	-	-	-	0

Bit 0 **FLIPCR0**: IO 口翻转控制, 在 IOxIPS1y 为 1 时有效 (x = A/B/C/D/E/F, y = 0-7)

0 = 翻转电平为 1/2VDD

1 = 翻转电平为 1.5V

7 定时器 0(TC0/PWM0)

7.1 概述

M9F951 TC0 为带有可设置 1:128 预分频器及周期寄存器的 8 位/16 位定时计数器，具有休眠状态下唤醒功能。

M9F951 有一个 8 位分辨率 PWM0。

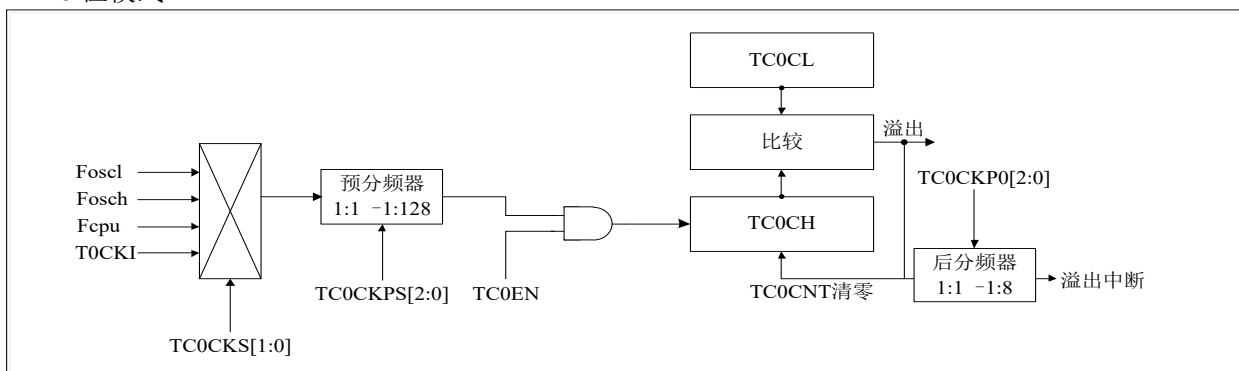
在 8 位模式下，TC0CL 作为 TC0 的周期寄存器，TC0 使能后，TC0CH 递加，当 TC0CH 与 TC0CL 数值相等时，TC0 溢出，将 TC0CH 清零重新开始计数，同时将中断标志位 T0IF 置 1。

在 16 位模式下，[TC0CH,TC0CL]作为 16 位的计数器，TC0 使能后，16 位计数器递加，当计数值等于 0xFFFF 时，16 位计数器将清零重新开始计数，同时将中断标志位 T0IF 置 1。

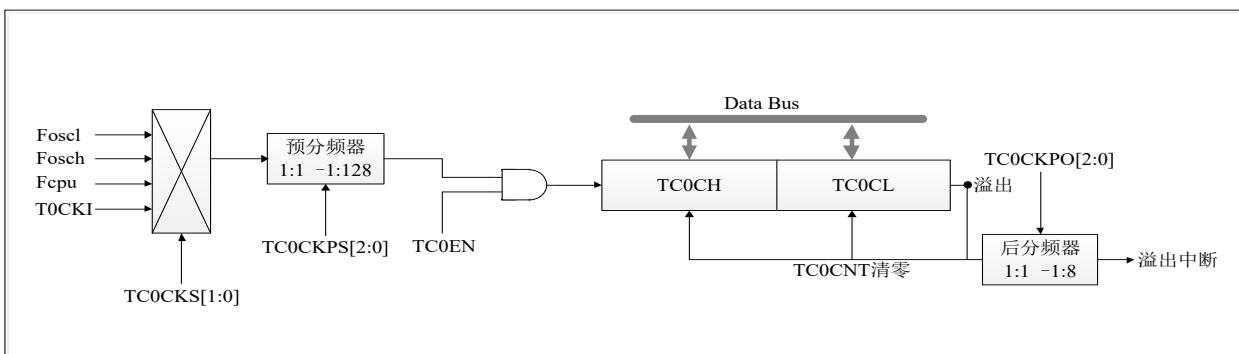
- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Foscl 和指令时钟 Fcpu 、外部时钟 T0CKI
- 可选择 8 位模式和 16 位模式
 - 8 位模式下，通过设置周期寄存器，可任意设置 TC0 的周期
- 预分频比多级可选，最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Foscl 或 Fosch 时，若所选择的时钟源振荡器一直工作，此时 TC0 在休眠状态下依然工作，溢出中断可唤醒 CPU）

TC0 框图

8 位模式



16 位模式



7.2 T0CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0CR	TC0EN	TC0MOD	T0HOSCS	TC0CKS[1:0]		TC0CKPS[2:0]		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TC0EN**: TC0模块使能位

0 = 关闭TC0

1 = 使能TC0

Bit 6 **TC0MOD**: TC0模式选择位

0 = 8位模式

1 = 16位模式

Bit 5 **TCHOSCS**: 定时器高频振荡器选择

0 = 选择与CPU相同的高频振荡器

1 = 选择与CPU不同的高频振荡器

Bit [4:3] **TC0CKS[1:0]**: TC0时钟源选择

TC0CKS[1:0]	TC0 时钟源选择
00	F _{OSCL} (低频系统时钟)
01	F _{OSCH} (高频系统时钟)
10	F _{CPU}
11	T0CKI

Bit [2:0] **TC0CKPS[2:0]**: TC0预分频比选择

TC0CKPS[2:0]	TC0 预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

7.3 TC0CL TC0 计数器低 8 位/周期寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CL	TC0CL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

7.4 TC0CH TC0 计数器高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CH	TC0CH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

7.5 PWM0CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0CR	PWM0EN	PWM0OE	-	-	-	-	PWM0M	-
读/写	R/W	R/W	-	-	-	-	R/W	-
复位后	0	0	-	-	-	-	0	-

Bit 7 **PWM0EN:** PWM0模块使能位

0 = 关闭PWM

1 = 使能PWM

Bit 6 **PWM0OE:** PWM0输出控制

0 = PWM信号不从管脚输出，管脚用做IO

1 = PWM信号从管脚输出(无需设置对应的方向寄存器位)

Bit 1 **PWM0M:** PWM0输出极性控制

0 = PWM高电平有效

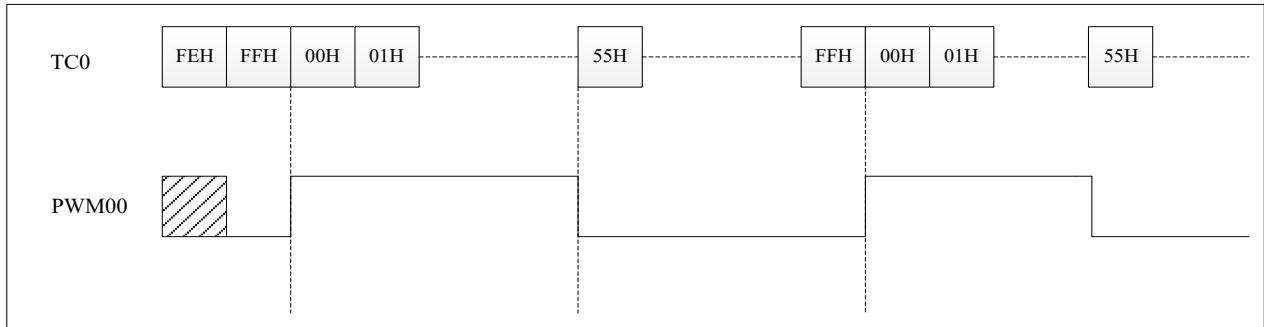
1 = PWM低电平有效

7.6 PWM0D 数据位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0D	PWM0D[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

7.7 PWM0 波形实例

例：PWM0CR = 11000000B，PWM0D = 55h，TC0CL = FFh，TC0CH = 00h。



8 定时器 1(TC1/PWM1)

8.1 概述

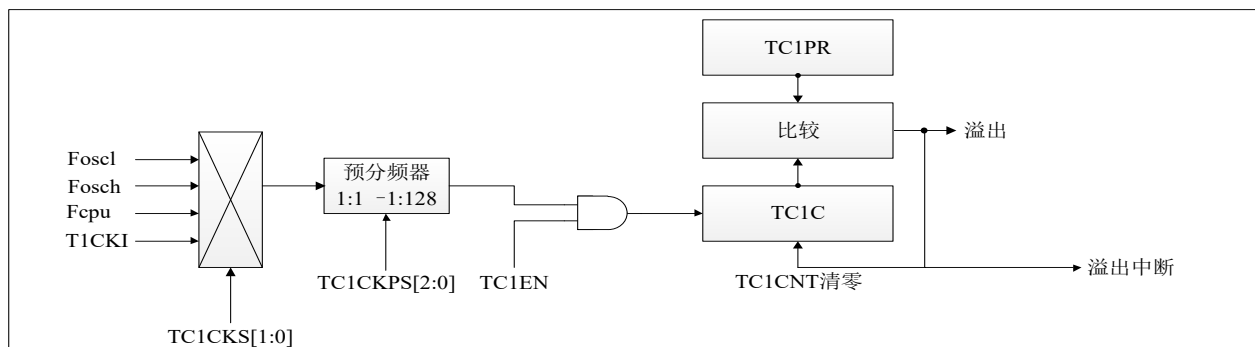
M9F951 TC1 为带有可设置 1:128 预分频器及周期寄存器的 16 位定时计数器，具有休眠状态下唤醒功能。

向上计数模式，TC1 使能后，TC1C 递增，当 TC1C 计数值与 TC1PR 相等时，TC1 溢出，将 TC1CH 清零重新开始计数，同时将中断标志位 TC1IF 置 1。

向下计数模式，TC1 使能后，TC1C 递减，当 TC1C 计数值为 0 时，TC1 溢出，TC1C 将重新载入 TC1PR 的值开始递减计数，同时将中断标志位 TC1IF 置 1。

中间对齐方式，TC1 使能后，TC1C 递增计数，当 TC1C 递增到 TC1PR 时，开始递减，递减到 0 时，产生溢出中断，同时开始递增。

- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Foscl、指令时钟 Fcpu 和外部时钟 TICKI
- 16 位周期寄存器
- 预分频比多级可选，最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Foscl，Fosch 或 TICKI 时，若所选择的时钟源振荡器一直工作，此时 TC1 在休眠状态下依然工作，溢出中断可唤醒 CPU）



M9F951 有 1 组 22 路 16 位 PWM。

- 5 位 PWM 有效起始寄存器
- 反相输出

8.2 T1CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR	TC1EN	TC1MOD[1:0]		TC1CKS[1:0]		TC1CKPS[2:0]		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TC1EN**: TC1模块使能位

0 = 关闭TC1

1 = 使能TC1

Bit [6:5] **TC1MOD[1:0]**: TC1模式选择位

TC1MOD[1:0]	TC1 模式选择
00	递增模式
01	递减模式
1x	中心对齐模式

注：中心对齐模式下，TC1MOD0 为只读模式，表示目前定时器的状态（增或减）。

Bit [4:3] **TC1CKS[1:0]**: TC1时钟源选择

TC1CKS[1:0]	TC1 时钟源选择
00	F _{OSCL} (低频系统时钟)
01	F _{OSCH} (高频系统时钟)
10	F _{CPU}
11	T1CKI

Bit [2:0] **TC1CKPS[2:0]**: TC1预分频比选择

TC1CKPS[2:0]	TC1 预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

8.3 TC1CL TC1 计数器低 8 位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1CL	TC1CL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

8.4 TC1CH TC1 计数器高 8 位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1CH	TC1CH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

8.5 TC1PRL TC1 周期寄存器低 8 位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRL	TC1PRL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

8.6 TC1PRH TC1 周期寄存器高 8 位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRH	TC1PRH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

8.7 PWM1xCR 控制寄存器(x=0-21)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1xCR	PWM1xEN	PWM1xOE	PWM1xM	PWM1xP[4:0]				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	RW	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **PWM1xEN:** PWM1x模块使能位(x=0-21)

0 = 关闭PWM1x

1 = 使能PWM1x

Bit 6 **PWM1xOE:** PWM1波形输出使能位(x=0-21)

0 = 端口用作IO

1 = 端口输出PWM1x波形(无需设置对应的方向寄存器位)

Bit 5 **PWM1xM:** PWM1x输出模式(x=0-21)

1 = 低电平有效

0 = 高电平有效

Bit [4:0] **PWM1xP:** PWM1xP波形有效起始时间(x=0-21)

有效电平起始控制，计数器1计到等于PWM1xP值时PWM1_x输出有效电平，详见PWM1x波形示例

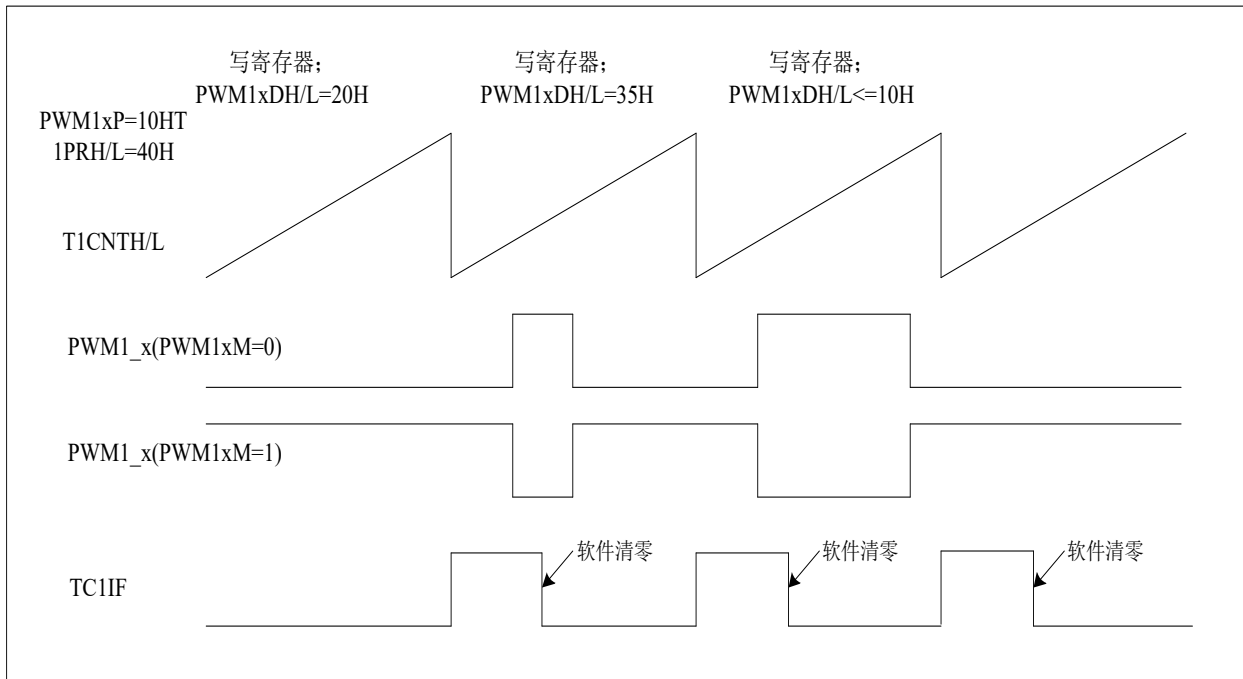
注：中心对齐模式不允许使用 PWM1xP。

8.8 PWM1xH 数据寄存器 (x = 0-21)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1xDH	PWM1xDH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1xDL	PWM1xDL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

8.9 PWM1x 波形示例 (x = 0-21)



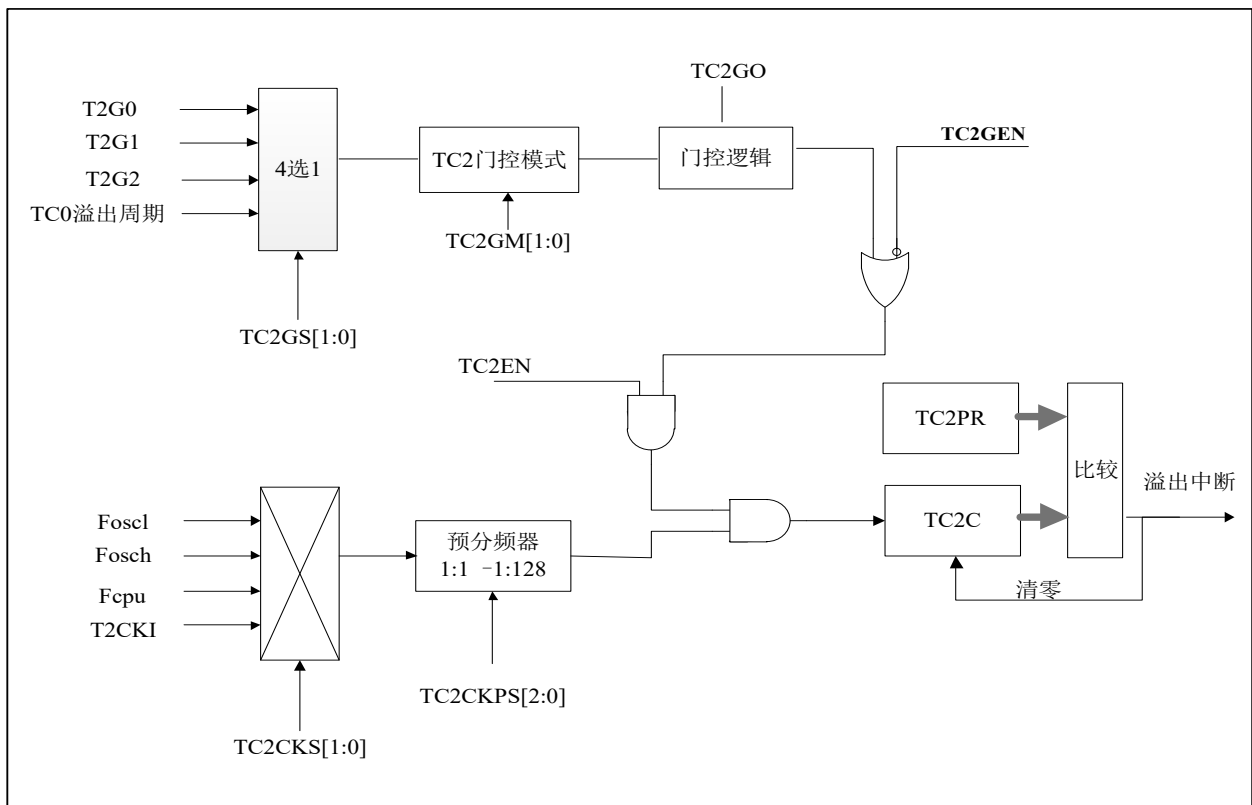
9 定时器 2 (TC2)

9.1 概述

M9F951 TC2 为带门控功能的可设置 1:128 预分频器及周期寄存器的 16 位定时计数器，具有休眠状态下唤醒功能。

- 可选择时钟源：高频系统时钟 Fosch，低频系统时钟 Fosc1、指令时钟 Fcpu 和外部输入 T2CKI
- 可选择预分频比，最大 1:128
- 门控模式可选择门控源：TC0 溢出信号、外部门控信号 T2G0/T2G1/T2G2
- 溢出中断功能
- 溢出中断唤醒功能（当输入频率选择 Fosc1，Fosch 或 T2CKI 时，若所选择的时钟源振荡器一直工作，此时 TC2 在休眠状态下依然工作，溢出中断可唤醒 CPU）

工作模式



9.2 T2CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR	TC2EN	TC2MOD[1:0]		TC2CKS[1:0]		TC2CKPS[2:0]		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TC2EN**: TC2模块使能位

0 = 关闭TC2

1 = 使能TC2

Bit [6:5] **TC2MOD[1:0]**: TC1模式选择位

TC2MOD[1:0]	TC2 模式选择
00	递增模式
01	递减模式
1x	中心对齐模式

注：中心对齐模式下，TC2MOD0 为只读模式，表示目前定时器的状态（增或减）。

Bit [4:3] **TC2CKS[1:0]**: TC2时钟源选择位

TC2CKS[1:0]	TC2 计数时钟源选择
00	F _{OSCL} (低频系统时钟)
01	F _{OSCH} (高频系统时钟)
10	F _{CPU}
11	T2CKI

Bit [2:0] **TC2CKPS[2:0]**: TC2预分频比选择

TCxCKPS[2:0]	TC2 预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

9.3 TC2CL 计数器低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CL	TC2CL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

9.4 TC2CH 计数器高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CH	TC2CH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

9.5 TC2PRL 周期寄存器低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRL	TC2PRL[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

9.6 TC2PRH 周期寄存器高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRH	TC2PRH[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

9.7 TC2GCR 门控控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2GCR	TC2GEN	TC2GO	-	-	TC2GS[1:0]		TC2GM[1:0]	
读/写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
复位后	0	0	-	-	0	0	0	0

Bit 7 **TC2GEN:** TC2门控模式使能位

0 = 关闭门控功能, TC2是否计数仅受TC2GEN控制

1 = 使能门控功能

Bit 6 **TC2GO:** 启动门控控制位

0 = 完成门控计数, 自动清零

1 = 启动门控

Bit [3:2] **TC2GS[1:0]:** TC2门控源选择位

TC2GS[1:0]	TC2 门控选择
00	T2G0
01	T2G1
10	T2G2
11	TC0 溢出周期 (只支持上升沿到上升沿模式)

Bit [1:0] **TC2GM[1:0]:** TC2门控模式选择位 (只有T2G0/T2G1/T2G2下可选择)

TC2GM[1:0]	TC2 门控模式选择
00	上升沿到下降沿
01	下降沿到上升沿
10	上升沿到上升沿
11	下降沿到下降沿

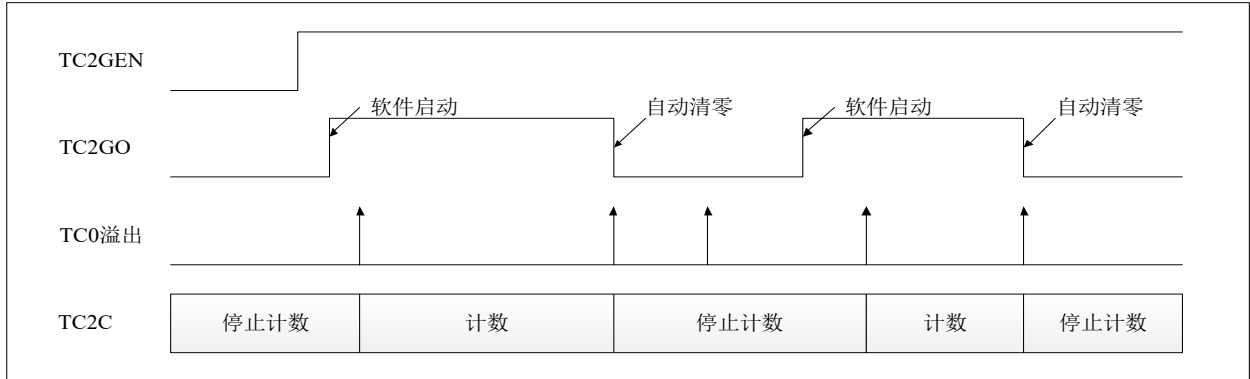
注:

(1) 启动门控前需先将门控使能, 不可同时置 1。

(2) 选择 TC0 溢出信号作为门控源时, 若 TC0 是 8 位模式, 则 TC0CL 不能为 0。

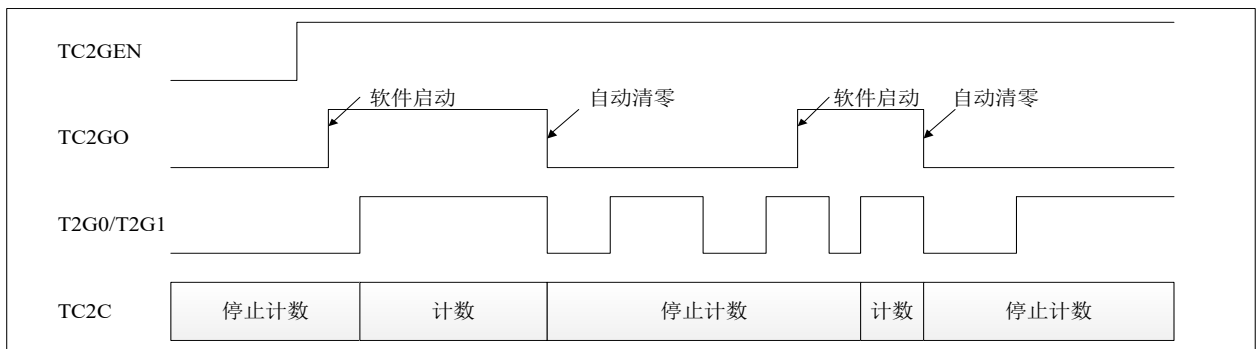
9.7.1 门控-TC0 溢出周期

上升沿到上升沿模式：启动门控计数后，门控逻辑从第一次 TC0 溢出开始计数，第二次 TC0 溢出停止计数，如下图。



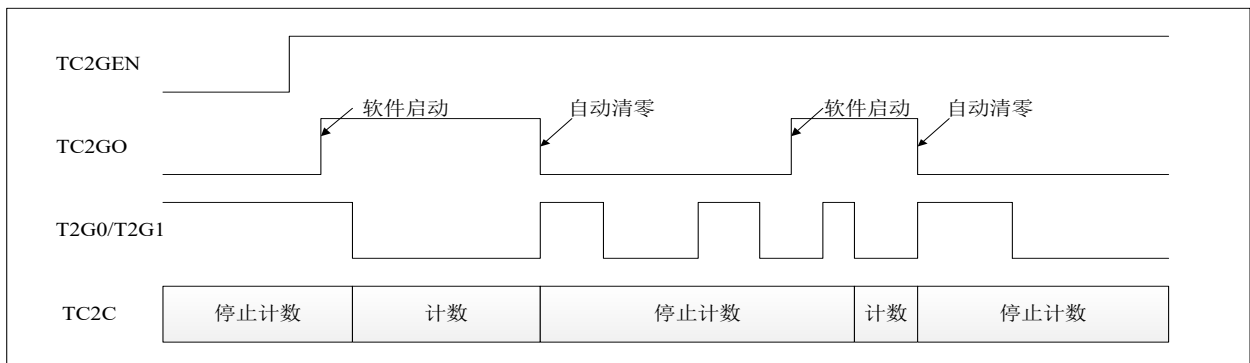
9.7.2 门控-上升沿到下降沿模式

上升沿到下降沿模式：启动门控计数后，门控逻辑从捕获到的第一个上升沿开始计数，然后捕获到下降沿停止计数，如下图。



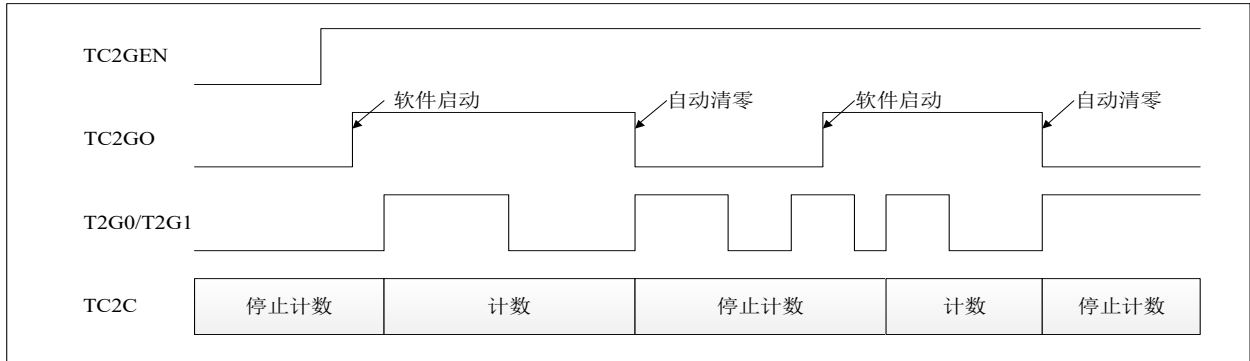
9.7.3 门控-下降沿到上升沿模式

下降沿到上升沿模式：启动门控计数后，门控逻辑从捕获到的第一个下降沿开始计数，然后捕获到上升沿停止计数，如下图。



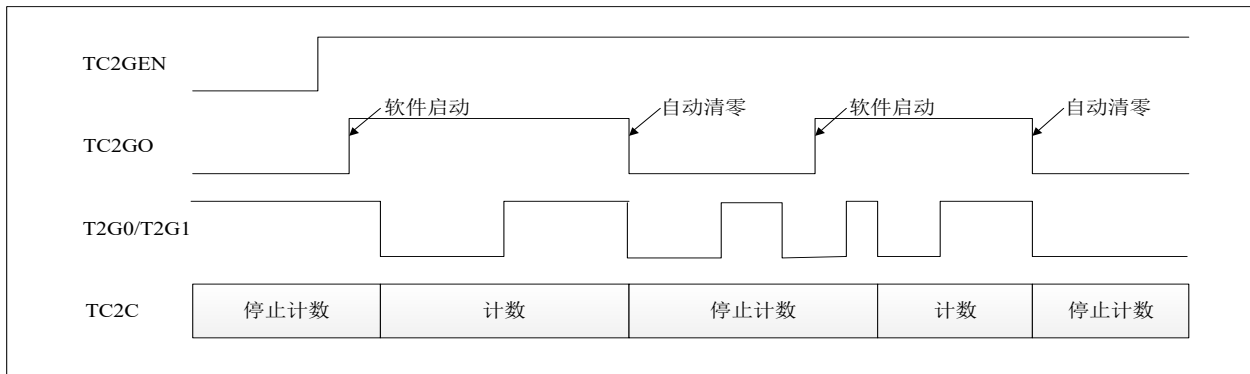
9.7.4 门控-上升沿到上升沿模式

上升沿到上升沿模式：启动门控计数后，门控逻辑从捕获到的第一个上升沿开始计数，然后捕获到第二个上升沿停止计数，如下图。



9.7.5 门控-下降沿到下降沿模式

下降沿到下降沿模式：启动门控计数后，门控逻辑从捕获到的第一个下降沿开始计数，然后捕获到第二个下降沿停止计数，如下图。



10 通用串行通讯口（USART）

10.1 概述

M9F951 有两个 USART，支持异步全双工模式和同步半双工模式。

10.2 TxxCR 发送控制寄存器(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxxCR	TXxEN	TxMCLR	TxSYNC	TxL9	TxSLAVE	TxSPD[1:0]		TxD9
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	1	0	0	0	0	0	0

Bit 7 **TXxEN**: 使能发送

0 = 屏蔽USART发送功能

1 = 使能USART发送功能

Bit 6 **TxMCLR**: 发送寄存器空标志

0 = 正在发送数据，移位寄存器不空

1 = 数据已发送，移位寄存器空

Bit 5 **TxSYNC**: 同步模式

0 = 选择异步模式

1 = 选择同步模式

Bit 4 **TxL9**: 数据长度选择

0 = 8位数据

1 = 9位数据

Bit 3 **TxSLAVE**: 同步发送/接收模式

0 = Master

1 = Slave

Bit [2:1] **TxSPD[1:0]**: 波特率时钟源分频比

SPD[1:0]	波特率分频比(n)
00	4
01	16
10	64
11	256

Bit 0 **TxD9**: 发送数据第9位数据

10.3 TXxREG 发送数据寄存器(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXxREG	TXxD[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

10.4 RXxCR 接收控制寄存器(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXxCR	RXxEN	RxCkPS	-	RxL9	SRxEN	RxOVF	FRxER	RxD9
读/写	R/W	R/W	-	R/W	R/W	R	R	R
复位后	0	0	-	0	0	0	0	x

- Bit 7 **RXxEN:** 使能接收
0 = 屏蔽USART接收功能
1 = 使能USART接收功能
- Bit 6 **RxCkPS:** 同步模式时钟模式选择
0 = 上升沿发送数据
1 = 下降沿发送数据
- Bit 4 **RxL9:** 数据长度选择
0 = 8位数据
1 = 9位数据
- Bit 3 **SRxEN:** 同步接收开始
0 = 停止同步接收
1 = 开始同步接收, 单字节接收模式下接收完一个字节自动清零
- Bit 2 **RxOVF:** 接收缓冲区溢出标志
0 = 接收缓冲区未发生溢出
1 = 接收缓冲区溢出, 读缓冲区自动清零
- Bit 1 **FRxER:** 接收数据格式错
0 = 当前接收数据未发生格式错
1 = 当前接收数据格式错 (未收到停止位)
- Bit 0 **RxD9:** 接收数据第9位数据

10.5 RXxREG 接收数据寄存器(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXxREG	RXxD[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	x	x	x	x	x	x	x	x

10.6 BRGDxH 波特率寄存器高位(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRGDxH	SBYTEx	-	-	-	-	-	BRGDx[9:8]	
读/写	R/W	-	-	-	-	-	R/W	R/W
复位后	0	-	-	-	-	-	0	0

Bit 7 **SBYTEx**: 同步接收模式选择
0 = 多字节接收
1 = 单字节接收, 接收完一个字节后自动清除SREN

Bit [1:0] **BRGDx[9:8]**: 10位波特率寄存器高2位

10.7 BRGDxL 波特率寄存器低位(x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRGDxL	BRGDx[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **BRGDx[7:0]**: 10位波特率寄存器低8位

10.8 USART 使用说明

10.8.1 波特率设置

通过设置 BRGD 和 SPD（分频比 n）来获得所需的波特率。

波特率计算公式：目标波特率 = $F_{OSC}/((BRGD+1) \times n)$

常用波特率设置（ $F_{OSC} = 16\text{MHz}$ ）

目标波特率	波特率时钟源分频比	BRGD	偏差
300	256	0xCF	0.17%
600	256	0x67	0.17%
1200	64	0xCF	0.16%
2400	64	0x67	0.16%
4800	16	0xCF	0.16%
9600	16	0x67	0.16%
19200	4	0xCF	0.16%
38400	4	0x67	0.16%
57600	4	0x44	0.64%
115200	4	0x21	2.12%

常用波特率设置（ $F_{OSC} = 24\text{MHz}$ ）

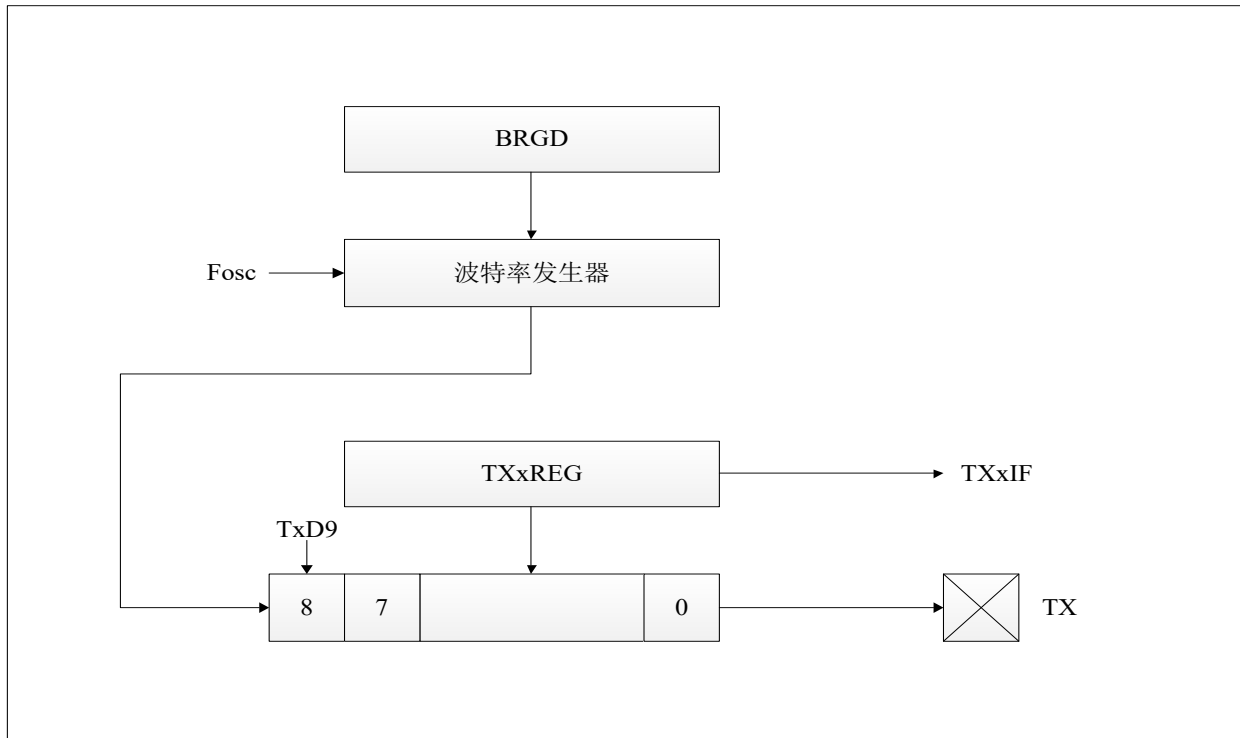
目标波特率	波特率时钟源分频比	BRGD	偏差
300	256	0x137	0.17%
600	256	0x9B	0.17%
1200	64	0x137	0.16%
2400	64	0x9B	0.16%
4800	16	0x137	0.16%
9600	16	0x9B	0.16%
19200	4	0x137	0.16%
38400	4	0x9B	0.16%
57600	4	0x67	0.16%
115200	4	0x33	0.16%

10.8.2 异步发送

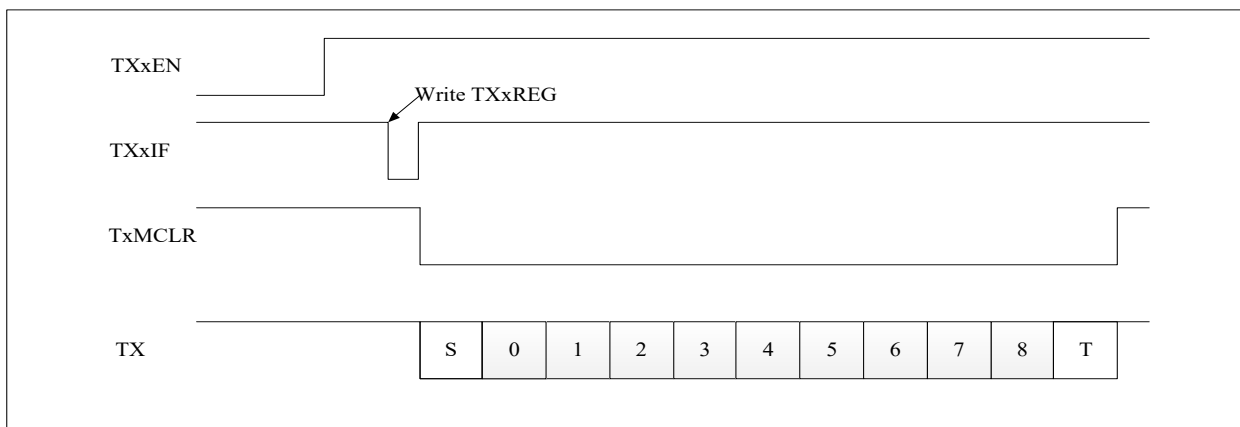
当 TXxEN 使能时, TXxIF 中断标志为 1 说明 TXxREG 发送寄存器为空, TxMCLR = 1 说明发送移位寄存器为空, 发送器处于空闲状态。

空闲状态时写入 TXxREG, 写入数据将立即装载到发送移位寄存器中, 此时, TXxIF 为 1, TxMCLR=0, 发送器进入发送状态。此时再次写入 TXxREG, TXxIF 将清零, 说明 TXxREG 有未发送数据, 发送移位寄存器发送完毕后, TXxREG 数据将自动载入发送移位寄存器继续发送, 且 TXxIF 为空。

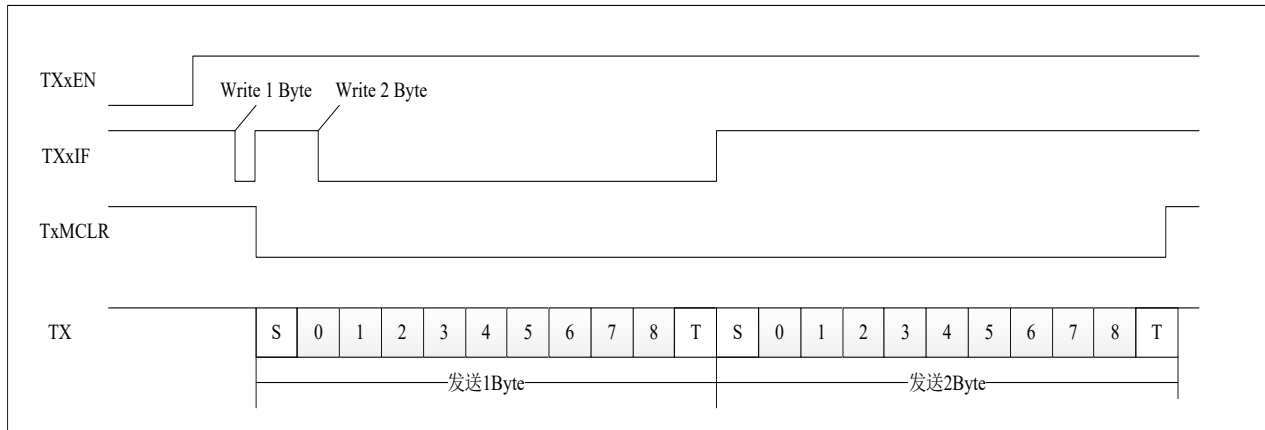
当 TXxIF 为 0 时写入 TXxREG, 将覆盖上次写入数据。



单字节发送:



多字节发送:



STEP1: 设置波特率 SSPD = X, BRGD = X, TxSYNC = 0

STEP2: 设置 TXxEN = 1, 设置数据模式 TxL9 = X

STEP3: 写入数据高位 TxD9

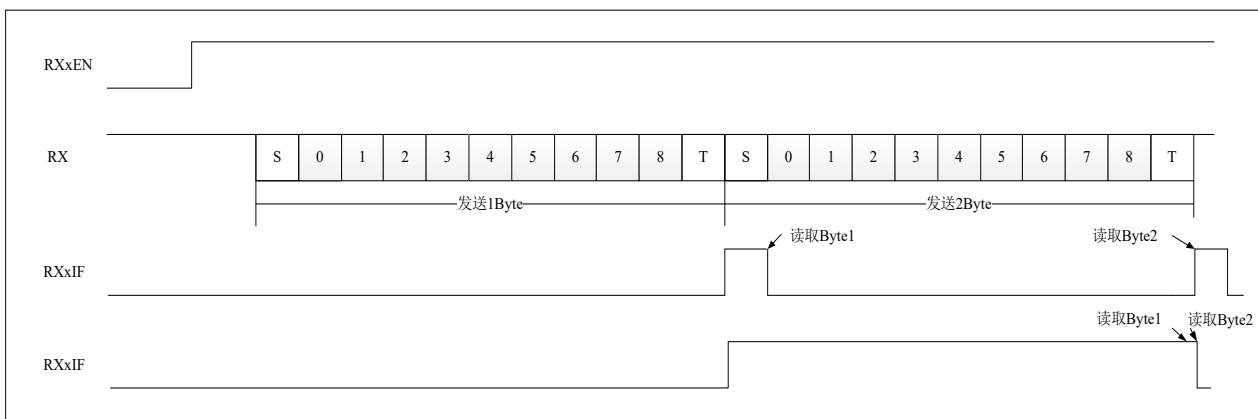
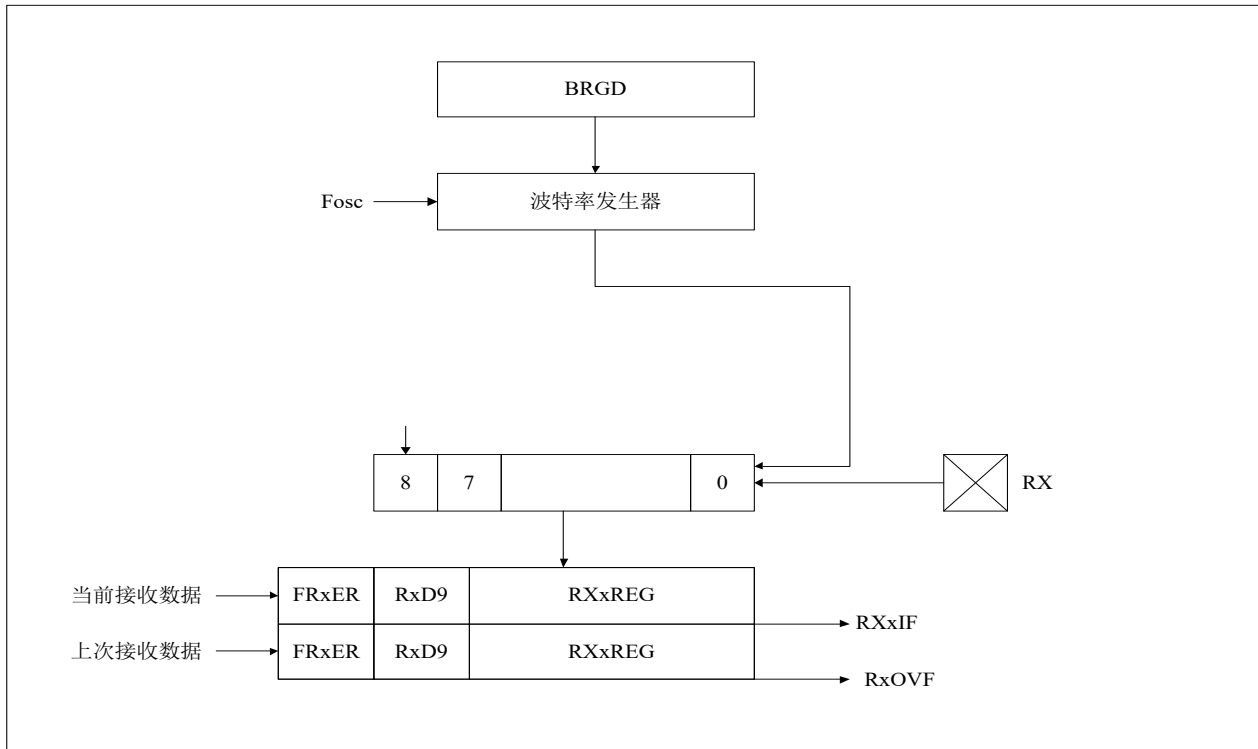
STEP4: 写入 TXxREG, 启动发送

STEP5: 当 TXxIF = 1 时, 写入 TXxREG 发送下一个字节

STEP6: 重复 STEP5, 直到该帧数据发送完成

10.8.3 异步接收

设置异步模式，使能 $RXxEN$ ，开始启动异步接收。 RX 管脚处于高电平时，接收器处于空闲状态，当检测到 RX 变为低电平，接收器检测该低电平是否有效起始位，若为有效起始位，则启动数据时钟恢复电路和数据恢复电路进行接收。1 个数据接收完成后， $RXxIF$ 置 1，当接收 3 个数据未读取， $RxOVF$ 置 1，同时舍弃第三个接收数据。完全读取 $RXxREG$ 后 $RXxIF$ 自动清零。



STEP1: 设置波特率 $SSPD = X$, $BRGD = X$, $TxSYNC = 0$

STEP2: 设置 $RXxEN = 1$

STEP3: 等待接收完成 $RXxIF = 1$

STEP4: 判断 $FRxER = 0$ ，若为 1，帧格式错误，舍弃数据

STEP5: 读取 $RxD9$

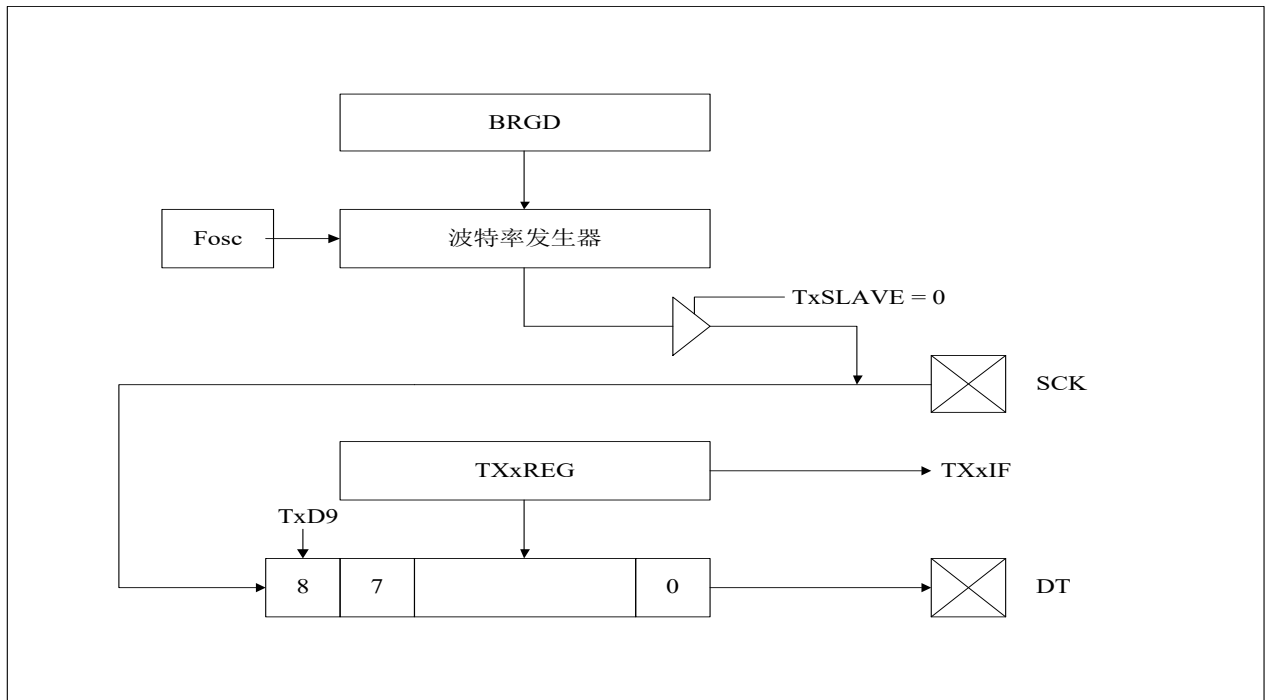
STEP6: 读取 $RXxREG$ ，重复 3-6

10.8.4 同步发送

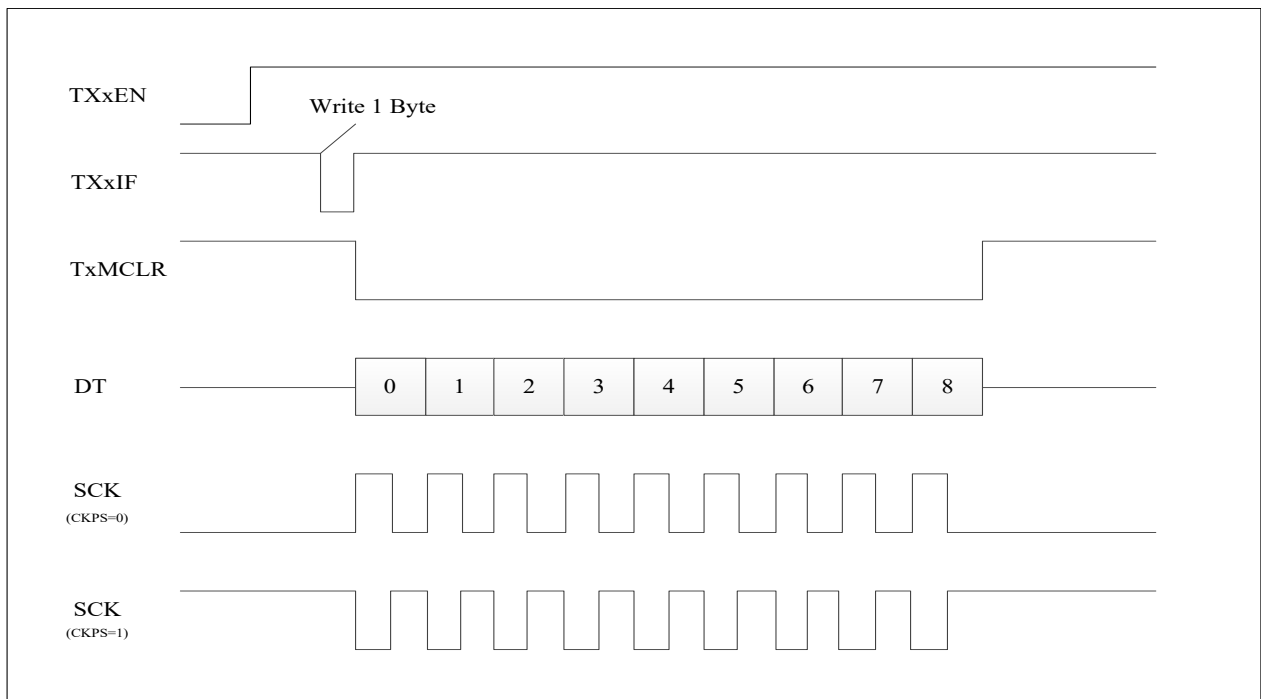
当 $TXxEN=1$, $TxSYNC=1$ 时, 使能同步发送功能。 $CKPS$ 选择发送时钟极性, $TXxIF$ 中断标志为 1 说明 $TXxREG$ 发送寄存器为空, $TxMCLR=1$ 说明发送移位寄存器为空, 发送器处于空闲状态。

空闲状态写入 $TXxREG$, 写入数据将立即装载到发送移位寄存器中, 此时, $TXxIF$ 为 1, $TxMCLR=0$, 发送器进入发送状态。此时再次写入 $TXxREG$, $TXxIF$ 将清零, 说明 $TXxREG$ 有未发送数据, 发送移位寄存器发送完毕后, $TXxREG$ 数据将自动载入发送移位寄存器继续发送, 且 $TXxIF$ 为空。

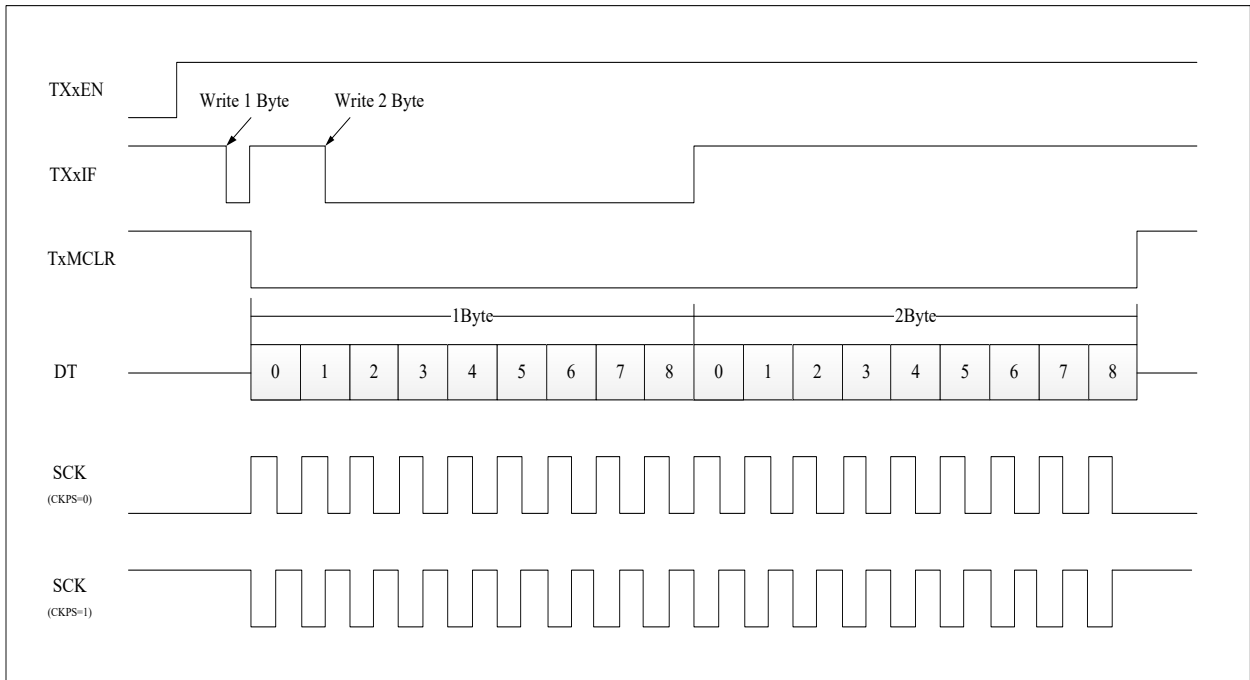
当 $TXxIF$ 为 0 时写入 $TXxREG$, 将覆盖上次写入数据。



单字节发送:



多字节发送:



参考操作步骤 TxSLAVE = 0:

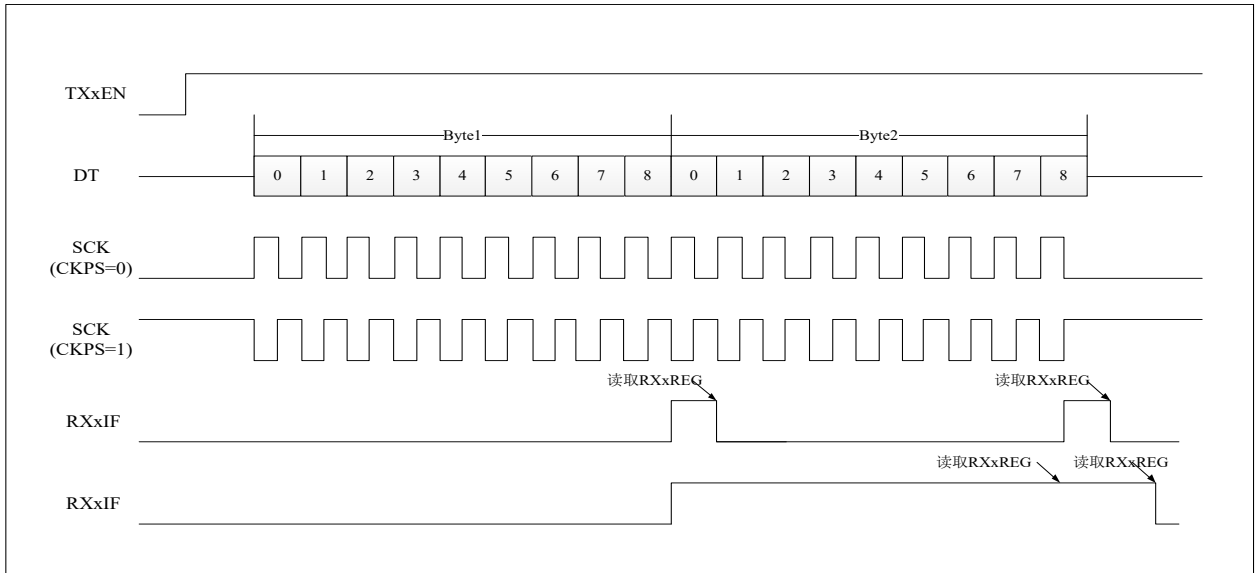
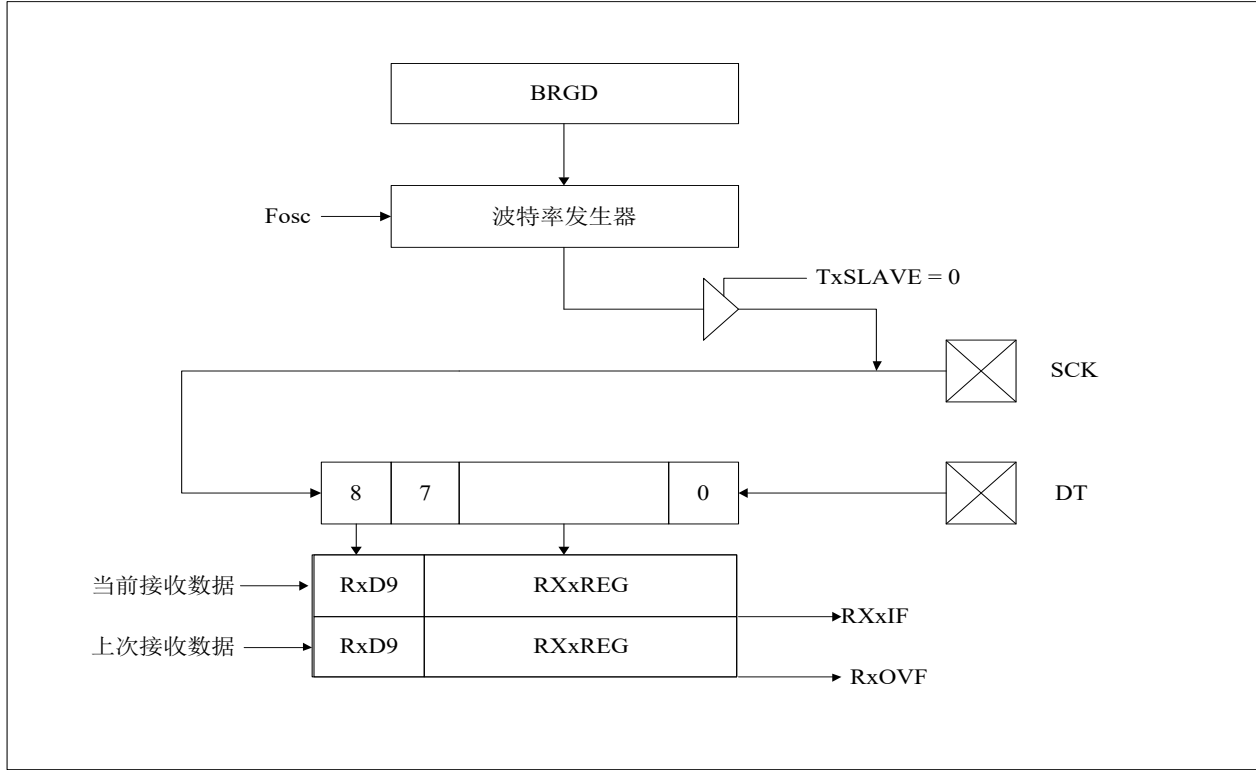
- STEP1: 设置波特率 SSPD = X, BRGD = X, TxSYNC = 1
- STEP2: 设置 TXxEN = 1, 设置数据模式 TxL9 = X
- STEP3: 写入数据高位 TxD9
- STEP4: 写入 TXxREG, 启动发送
- STEP5: 当 TXxIF = 1 时, 写入 TXxREG 发送下一个字节
- STEP6: 重复 STEP5, 直到该帧数据发送完成

参考操作步骤 TxSLAVE = 1:

- STEP1: 设置波特率 SSPD = X, BRGD = X, TxSYNC = 1
- STEP2: 设置 TXxEN = 1, 设置数据模式 TxL9 = X
- STEP3: 当 TXxIF = 1 时, 写入数据高位 TxD9
- STEP4: 写入 TXxREG 等待发送下一个字节
- STEP5: 重复 STEP3-4, 直到该帧数据发送完成

10.8.5 同步接收

设置同步 $TxSYNC=1$ 模式，使能 $RXxEN$ ，开始启动异步接收。 RX 管脚处于高电平时，接收器处于空闲状态，当检测到 RX 变为低电平，接收器检测该低电平是否有效起始位，若为有效起始位，则启动数据时钟恢复电路和数据恢复电路进行接收。1 个数据接收完成后， $RXxIF$ 置 1，当接收 3 个数据未读取， $RxOVF$ 置 1，同时舍弃第三个接收数据。完全读取 $RXxREG$ 后 $RXxIF$ 自动清零。



参考操作步骤 TxSLAVE = 0:

STEP1: 设置波特率 SSPD = X, BRGD = X, TxSYNC = 0

STEP2: 设置 RXxEN = 1

STEP3: 写 SRxEN 启动接收

STEP4: 等待接收完成 RXxIF = 1

STEP5: 读取 RxD9

STEP6: 读取 RXxREG, 单字节接收 (SBYTE_x = 1) 重复 3-6; 多字节接收 (SBYTE_x = 0) 重复

4-6

参考操作步骤 TxSLAVE = 1:

STEP1: 设置波特率 SSPD = X, BRGD = X, TxSYNC = 0

STEP2: 设置 RXxEN = 1

STEP3: 写 SRxEN 启动接收

STEP4: 等待接收完成 RXxIF = 1

STEP5: 读取 RxD9

STEP6: 读取 RXxREG, 单字节接收 (SBYTE_x = 1) 重复 3-6; 多字节接收 (SBYTE_x = 0) 重复

4-6

10.8.6 唤醒及休眠模式下通讯

TXxIE 置 1 时, TXxIF 中断标志可唤醒 CPU。

RXxIE 置 1 时, RXxIF 中断标志可唤醒 CPU。

异步接收时, 检测到 START 位将自动使能高频振荡器, 接收完成后唤醒 CPU。

同步接收时, 若作为主机, 则休眠状态下部工作; 作为从机, 则接收 1 个字节完成后唤醒 CPU。

11 串行通讯口（I2C）

11.1 概述

I²C 和 SPI 不可同时使用，M9F951 有 2 个功能相同的 I²C 模块，两个模块的寄存器地址分配具体请见 2.2.3 系统寄存器定义。

11.1.1 I²C

- I²C 总线规范兼容性：
 - 从机模式和主机模式
 - 标准模式（高达 100kHz）
 - 快速模式（高达 400kHz）
 - 超快速模式（高达 1MHz）
 - 7 位和 10 位地址模式
 - 广播呼叫
- 总线仲裁
- I2C 中断
- 独立的时钟：允许 I²C 选择一个独立的时钟源
- 有一个深度为 4 的 FIFO
- 高位数据先发。

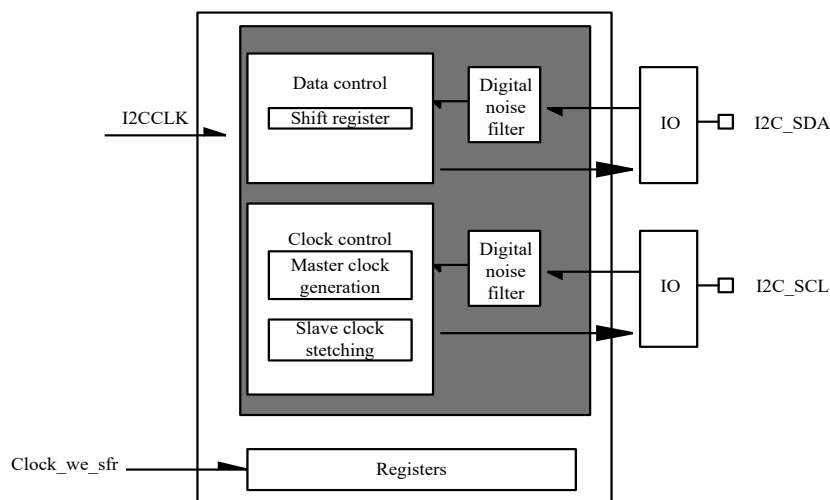


图 11.1 I²C 总体框图

11.1.2 SPI

- SPI 总线规范兼容性：

- 从机模式和主机模式
- 全双工模式
- 4 种工作模式
- 独立的时钟：允许 SPI 选择一个独立的时钟源
- 有一个深度为 4 的 FIFO
- 数据发送顺序可选。

11.2 I²C 功能描述

11.2.1 I²C 通讯示意图

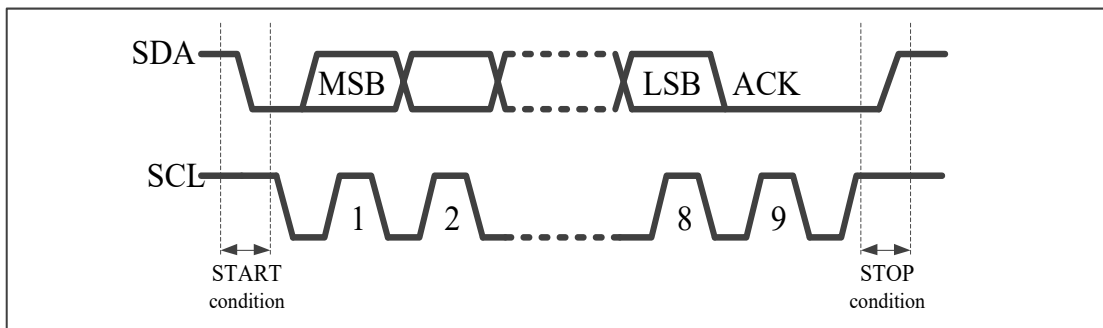


图 11.2 I²C 总线协议

在主模式下，I²C 接口启动数据传输，并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I²C 接口能识别它自己的地址(7 位或 10 位)和广播呼叫地址。

数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址(7 位模式为 1 个字节，10 位模式为 2 个字节)。地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位(ACK)给发送器。软件可以通过 CxGCEN 位设置 I²C 接口的地址(7 位、10 位地址或广播呼叫地址)。

11.2.2 I²C 从机模式

要工作在从机模式下，必须启用至少一个从机地址。寄存器 I2CxOAR 是用来写入从机的本机地址。

通过设置 I2CxCR2 寄存器的 CxADDR10 位，选择 7 位地址模式（默认）或 10 位地址模式，如选中 10 位地址，地址的高两位存放在 I2CxCR2 的 bit1-0。

I2CxCR1 寄存器的 CxGCEN 位被置 1 时，呼叫地址被启用。

当 I²C 有启用了的地址选中时，CxADDF 状态标志被置 1，当地址不匹配的时候，返回 NACK。

从机使用它的时钟延长的功能，这意味着，它可根据需要在 SCL 信号拉低期间执行软件动作。检查 CxDIR 标志，以了解数据传输的方向。在传输数据期间，当进行写操作 FIFO 为空时或者进行读操作 FIFO 为满时，从机会拉低总线。

从机片选信号可以通过寄存器 I2CxCR2 的 bti6-SxSDIS 配置。

11.2.2.1 从机发送

当发送 FIFO 为空时 CxBF 位置 1，如果 INTCR2 寄存器的 I2CxIE 为 1，会产生一个中断。向 I2CxDBUF 寄存器写入发送数据时，CxBF 位被清除，连续写 CxDBUF 寄存器 CxNT+1 次。

当收到一个 NACK，I2CxSR 寄存器中的 CxNACKF 位置 1，从机会自动释放 SCL 和 SDA 线，这是为了允许主机执行停止或重新启动条件。

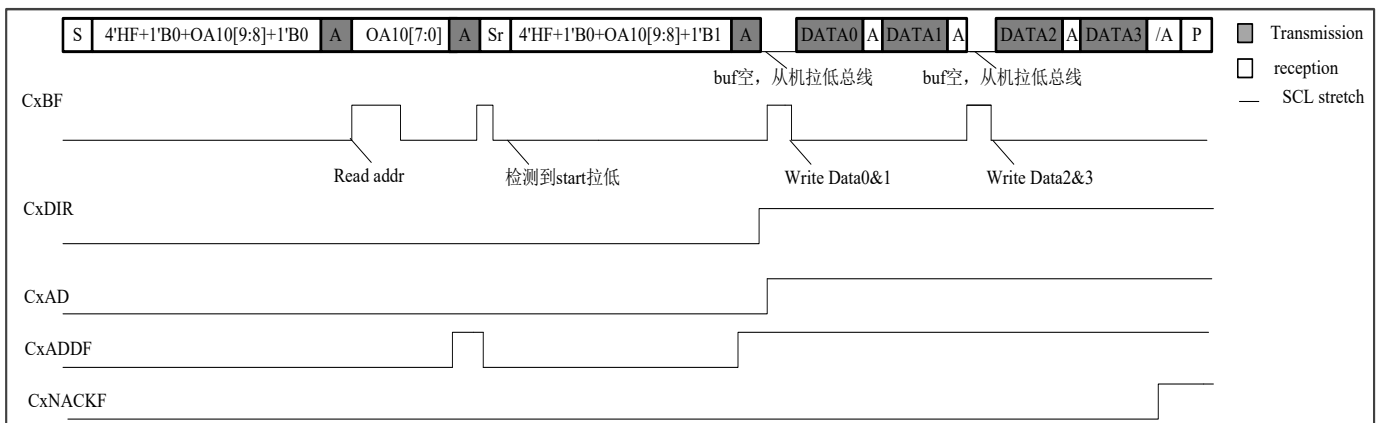


图 11.3 I²C 从机发送（以 10 位地址传输 2 个字节产生中断为例）

11.2.2.2 从机接收

在 FIFO 接收 $CxNT+1$ 个数据 I2CxSR 寄存器的 CxBF 被置 1，如果 INTCR2 寄存器的 I2CxIE 为 1，会产生一个中断。在接收 FIFO 为空时 CxBF 会被清除。

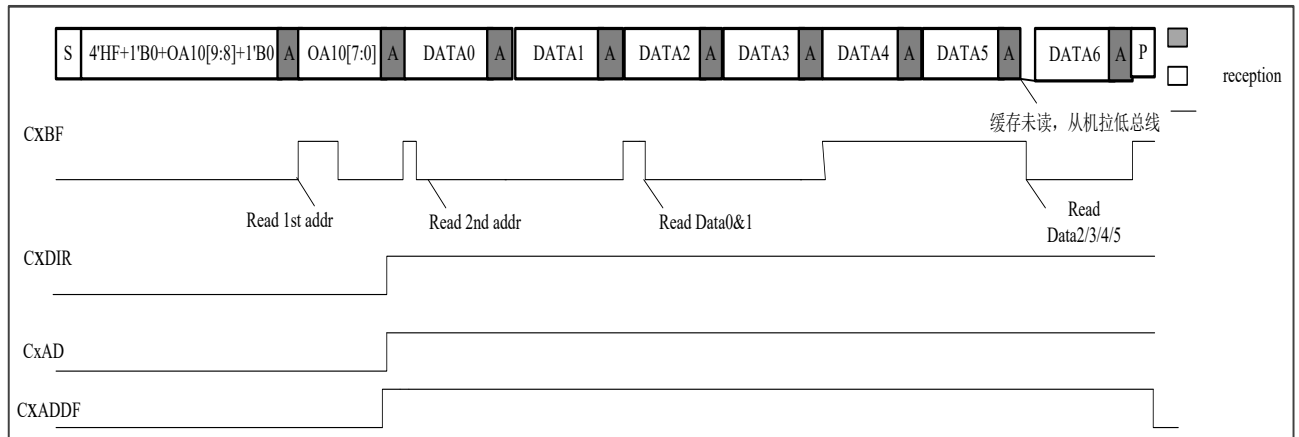


图 11.4 I²C 从机接收（以 10 位地址传输 2 个字节产生中断为例）

11.2.2.3 从机软件流程图

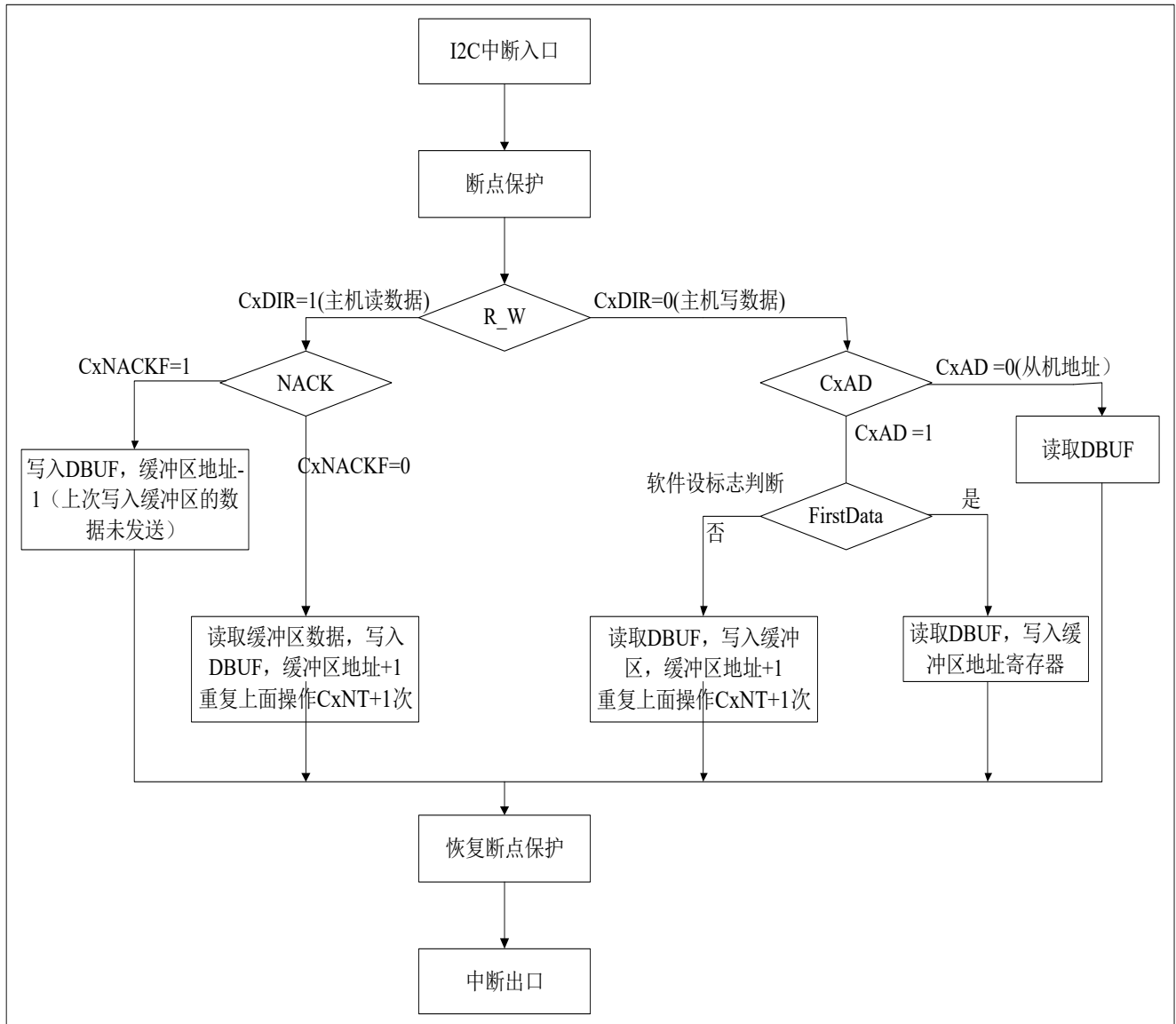


图 11.5 I²C 从机软件流程图

11.2.3 I²C 主机模式

11.2.3.1 I²C 主机初始化

在启动外设之前，必须设置 I2CxPRESC 寄存器来配置 I2C 主时钟。

为了启动通讯，必须设置从机地址的下列参数：

- 地址模式（7 位或 10 位）：CxADD10
- 设置 I2CxCR1 寄存器的 CxSR 位

主机在仲裁丢失 I2CxSR 寄存器 CxAL 位会置 1，如果 INTCR2 寄存器的 I2CxIE 为 1，会产生一个中断。

11.2.3.2 主机寻址一个 10 位地址的从机的初始化

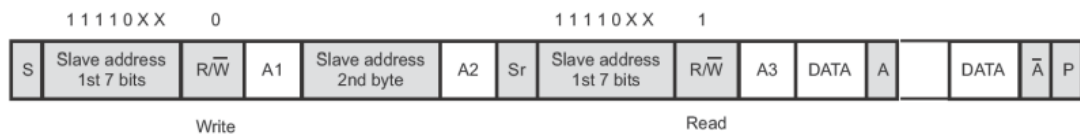


图 11.6 10-bit address 读操作

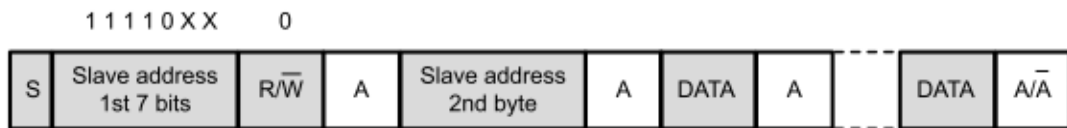


图 11.7 10-bit address 写操作

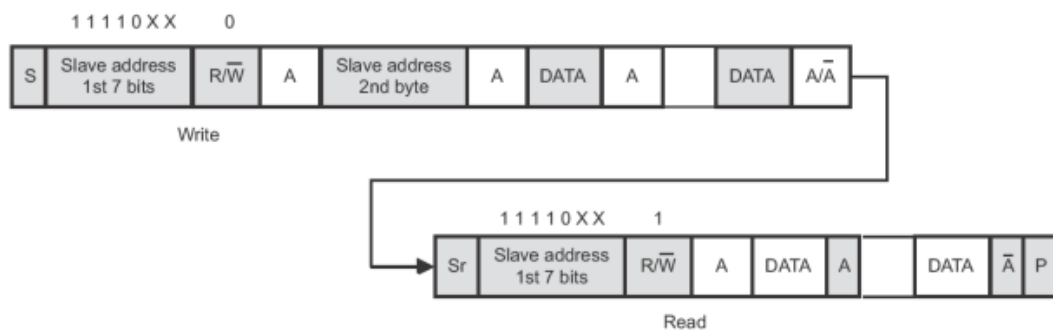


图 11.8 10-bit address 组合

11.2.3.3 总线仲裁

当所有 master 在 SDA 上都写 1 时，SDA 的数据才是 1，只要有一个 master 写 0，那此时 SDA 上的数据就是 0。一个 master 每发送一个 bit 数据，在 SCL 处于高电平时，就检查 SDA 的电平是否和发送的数据一致，如果不一致，这个 master 便仲裁失败，然后停止向 SDA 写数据。也就是说，如果 master 一直检测到总线上数据和自己发送的数据一致，则继续传输，这样在仲裁过程中就保证了赢得仲裁的 master 不会丢失数据。

Master 准备发送 start 条件，当检测到总线上已有 start 条件没有 stop 条件，master 仲裁失败，start 发不出去。

仲裁失败的 master 不再产生时钟脉冲，并且要在总线空闲时才能重新传输。主机在仲裁丢失 I2CxSR 寄存器 AL 位会置 1，如果 INTCR2 寄存器的 I2CxIE 为 1，会产生一个中断。

11.2.3.4 主机发送

在写传输的情况下，发送 FIFO 为空时(CxNT+1 个字节发送完成)，CxBF 标志会被置 1，如果 INTCR2 寄存器的 I2CxIE 位为 1，则会产生一个中断。向 DBUF 寄存器写入待发送数据时，CxBF 位被清除。连续写 DBUF 寄存器 CxNT+1 次。

软件将 I2CxCR1 中的 CxSR 位置 1，如果总线空闲发出一个 START 条件。I2CxCR1 寄存器的 CxSP 位置 1 来发出停止条件

如果收到一个 NACK，I2CxISR 寄存器的 CxNACKF 标志会被置 1，并在收到 NACK 之后自动停止发送。

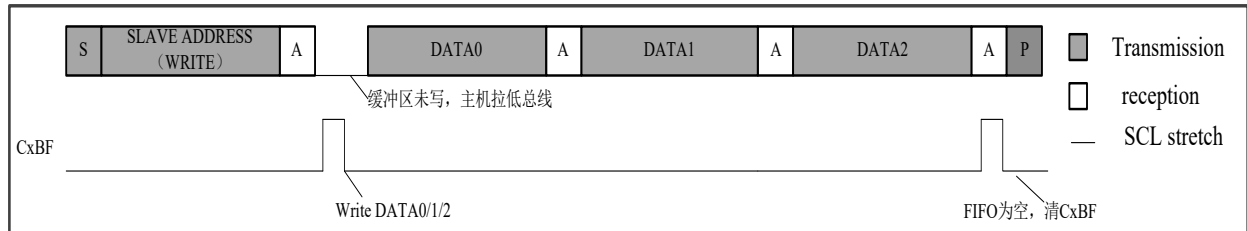


图 11.9 I²C 主机发送器传输时序图（以 7 位地址传输 3 个数据产生中断为例）

11.2.3.5 主机接收

在读传输时，CxNT+1 字节接收完后，CxBF 标志会置 1。如果 INTCR2 寄存器的 I2CxIE 位为 1，则产生一个中断。当接收 FIFO 为空时 BUF 会被自动清零。

软件将 I2CxCR2[5]置 1，停止接收数据。可以将 I2CxCR1 寄存器的 CxSP 位置 1 来发出停止条件。

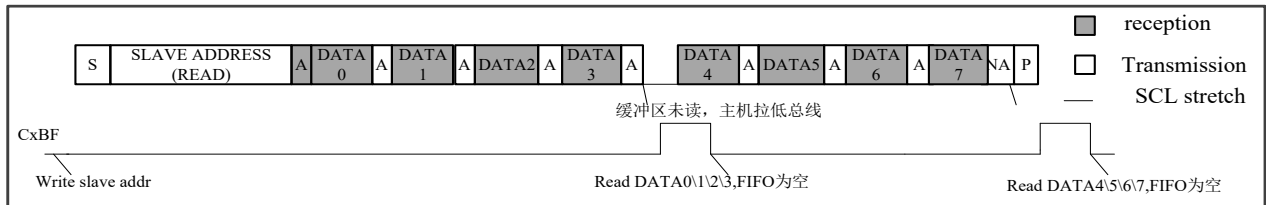


图 11.10 主机接收传输时序图

11.2.3.6 主机软件操作流程

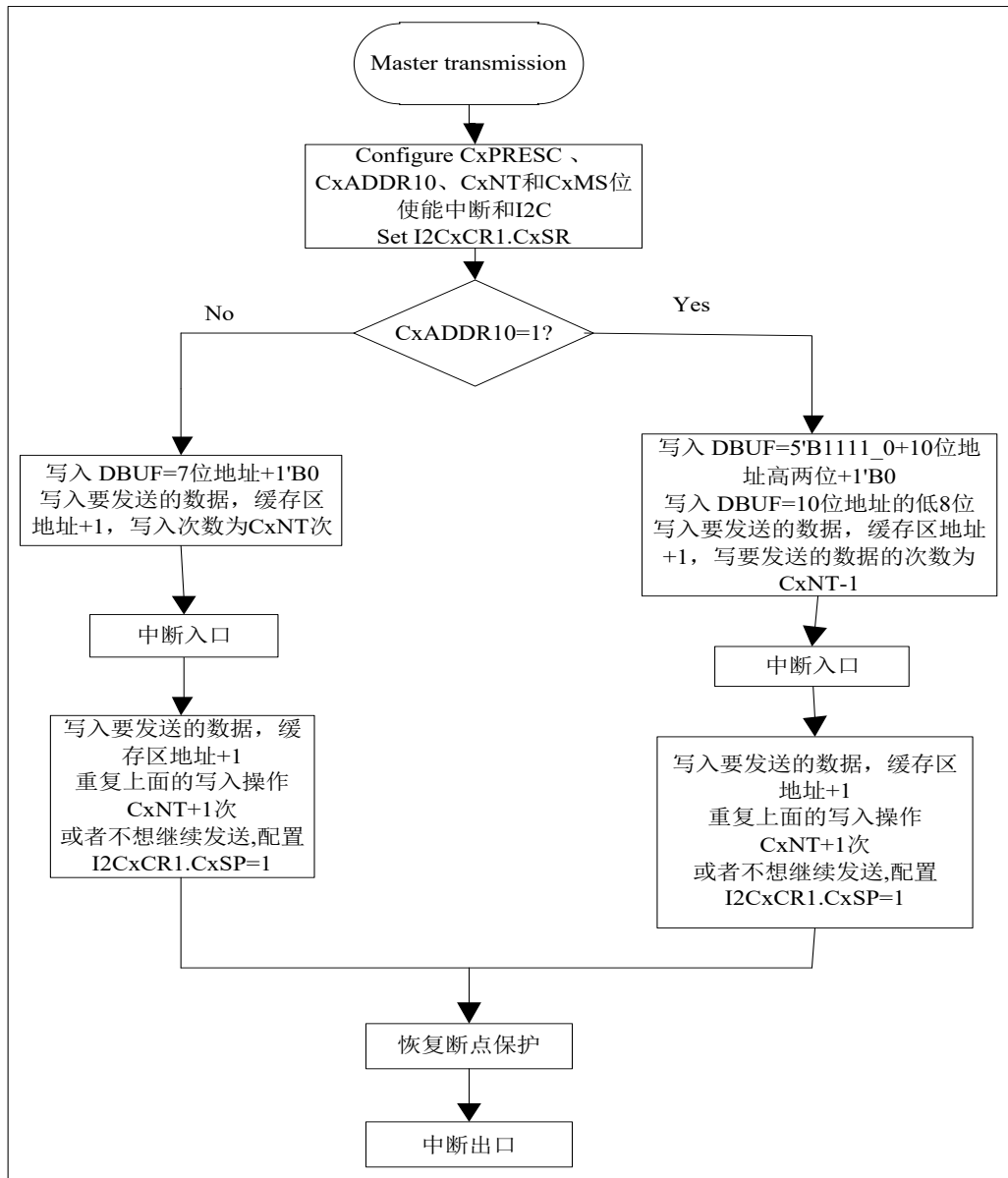


图 11.11 主机写操作软件流程图

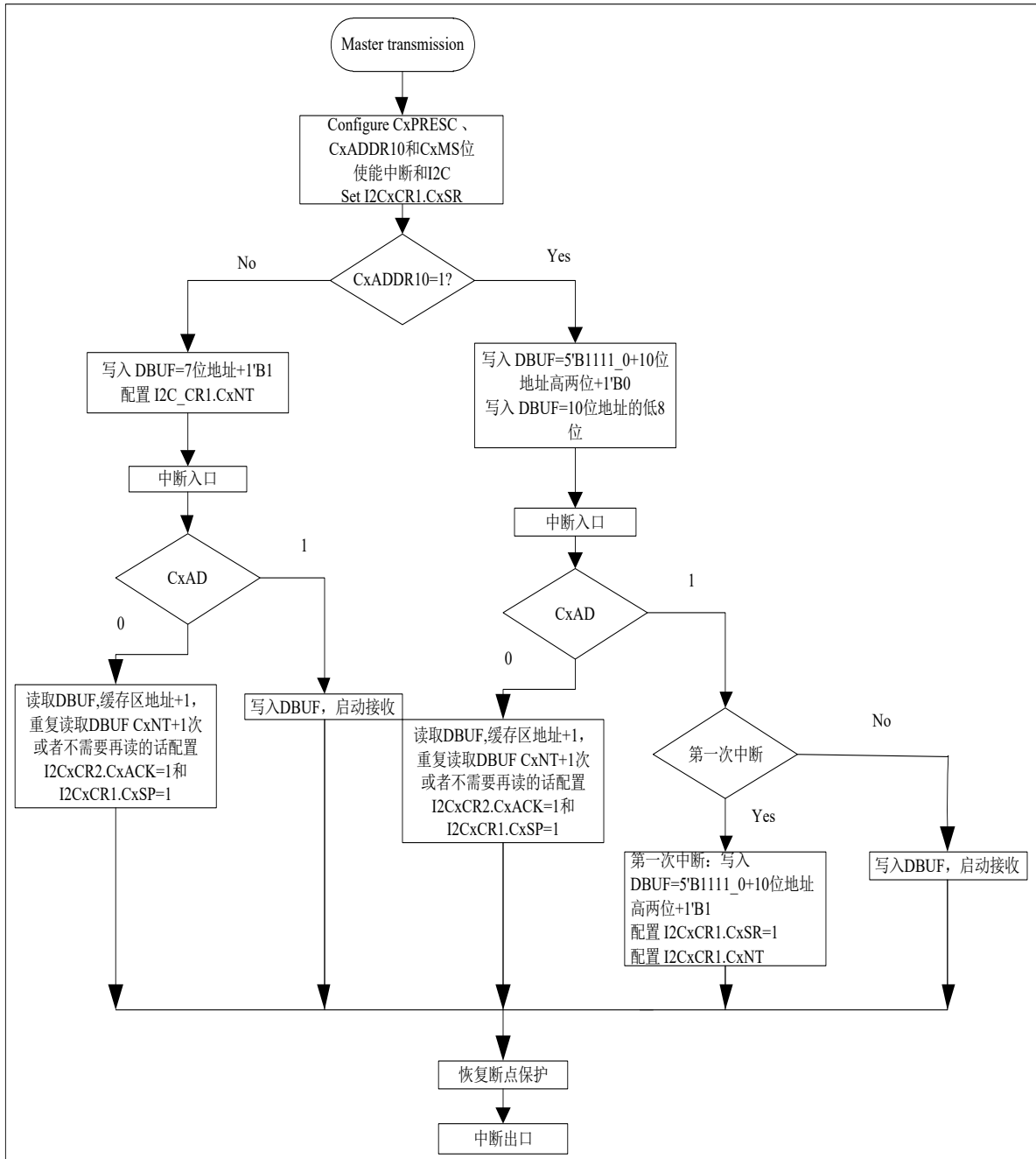


图 11.12 主机读操作软件流程图

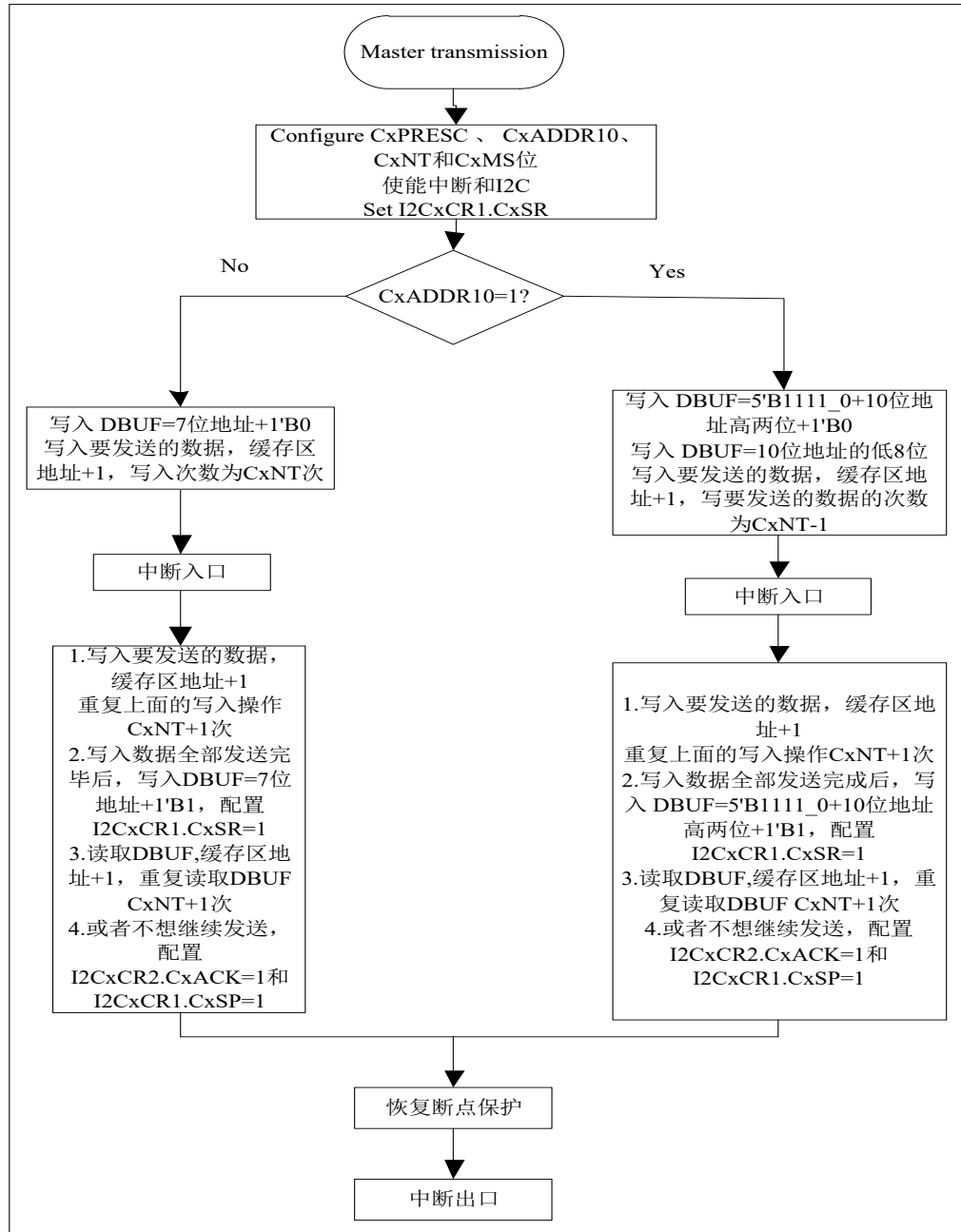


图 11.13 主机写完后读操作软件流程图

11.3 SPI 功能描述

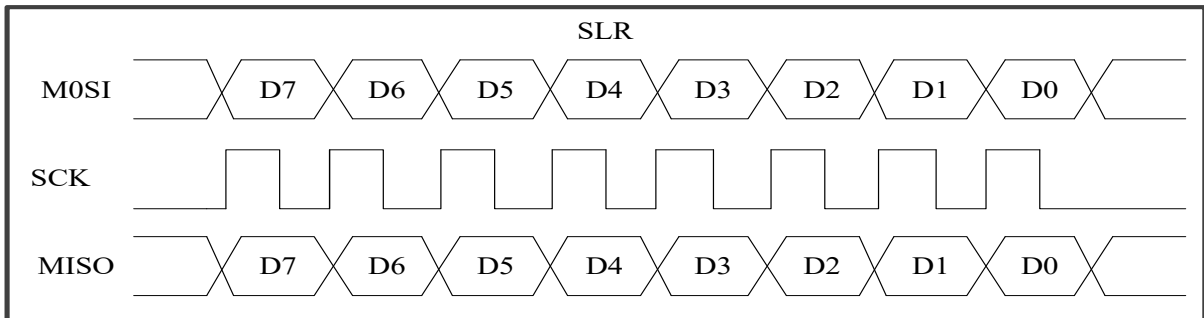
11.3.1 SPI 通讯示意图

SPI 在 DBUF 写入要传输的数据的时候开始传输,每传输 $CxNT+1$ 个字节, `spif` 会置 1, 如果 `INTCR2` 寄存器的 `I2CxIE` 为 1,则会产生中断(由于 SPI 和 I2C 不会同时使用, 故共用一个中断使能), 当接收 FIFO 为空时, `SxIF` 会由硬件清零。

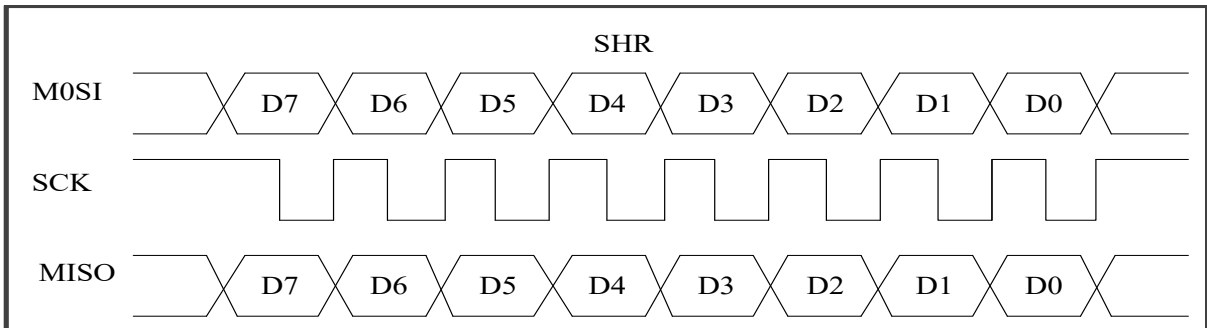
SPI 作为从机，是否需要片选信号（低有效）由 I2CxCR2[6]控制。

四种发送模式

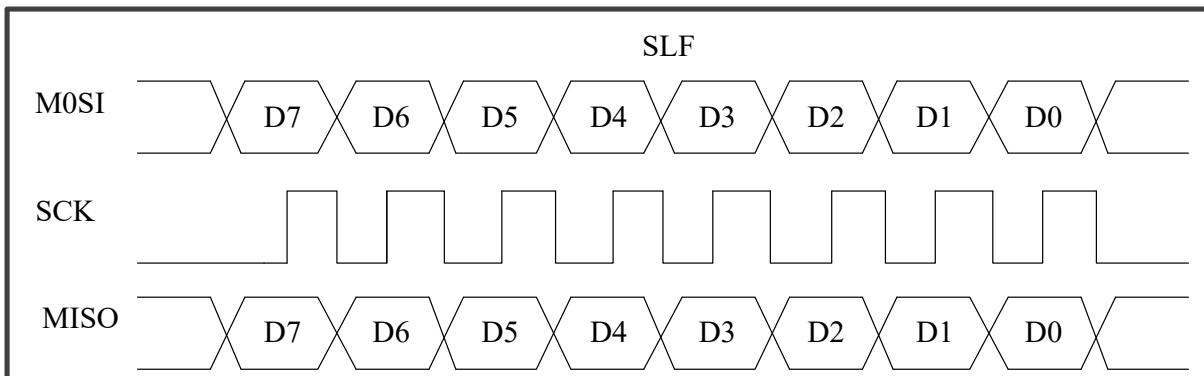
SLR：时钟空闲时为低电平，上升沿发送，下降沿接收；



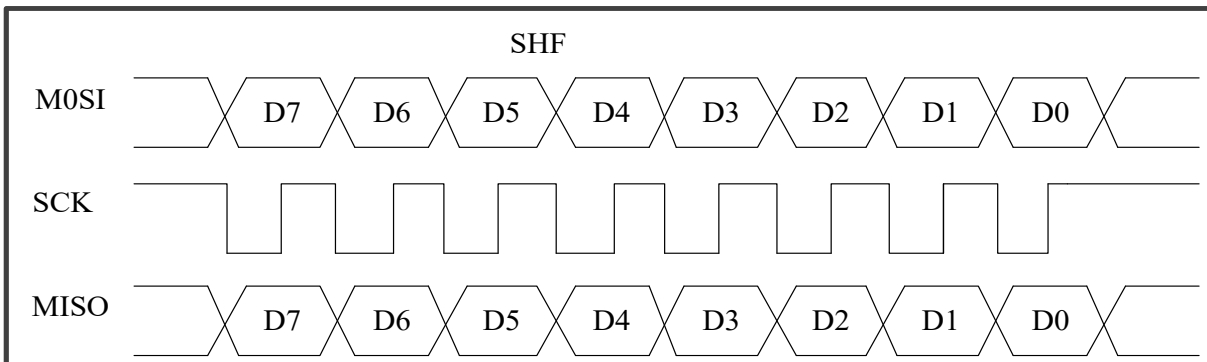
SHR：时钟空闲时为高电平，上升沿发送，下降沿接收；



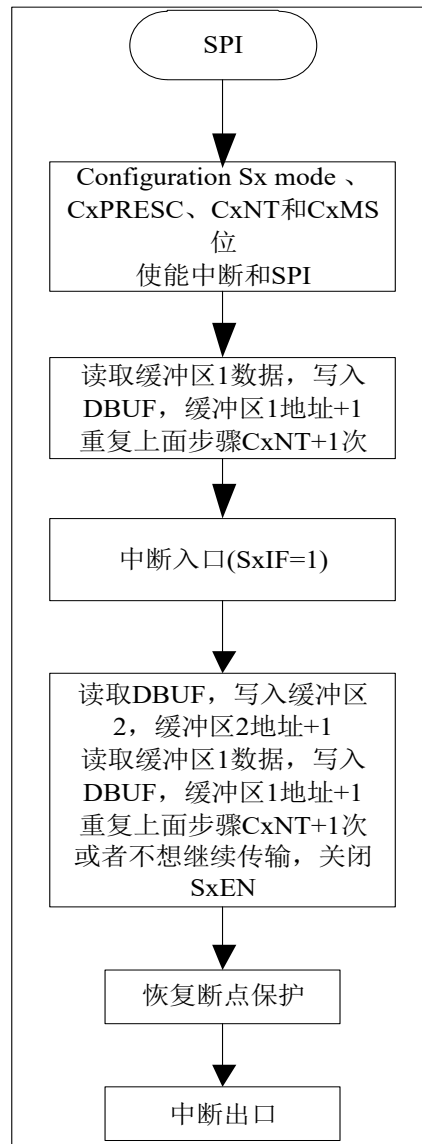
SLF：时钟空闲时为低电平，下降沿发送，上升沿接收；



SHF：时钟空闲时为高电平，下降沿发送，上升沿接收。



11.3.2 SPI 软件操作流程



注：SPI 作为 slaver 需要发送数据，如果不需要发送数据，硬件默认发送 0x00。

11.4 寄存器

11.4.1 控制寄存器 1 (I2CxCR1,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxCR1	CxEN	SxEN	CxSR	CxSP	CxMS	CxNT[1:0]		CxGCEN
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit 7 **CxEN**: 使能 I2Cx 通讯

- 0 = 禁用 I2Cx 通讯
- 1 = 启用 I2Cx 通讯。

Bit 6 **SxEN**: 使能 SPIx 通讯

- 0 = 禁用 SPIx 通讯
- 1 = 启用 SPIx 通讯。

注: CxEN 和 SxEN 不能同时使能。

Bit 5 **CxSR**: I2C 产生起始条件

- 0 = 不起始条件产生
- 1 = 产生起始条件

注: 软件置 1, 在检测到 start 条件后由硬件清零。

Bit 4 **CxSP**: I2C 产生停止条件

- 0 = 没有停止条件产生
- 1 = 产生停止条件

注: (1) 软件置 1, 在检测到 stop 条件后由硬件清零;

(2) 软件配置 CxSR 和 CxSP 位时, 如果使用 bset 指令, 两条 bset 指令不能连续发送, 中间要插入一条指令。

Bit 3 **CxMS**: master 和 slave 的选择位

- 0 = I2C/SPI 作为 slave
- 1 = I2C/SPI 作为 master

注: 该位在 I2C/SPI 使能时起作用。

Bit[2:1] **CxNT[1:0]**: 传输 N 个字节后产生中断标志

CxNT[1:0]	传输 N 个字节后产生中断标志
00	N=1
01	N=2
10	N=3
11	N=4

Bit 0 **CxGCEN**: 广播呼叫地址使能

0 = 禁止广播呼叫, 地址 8'b0000_0000 会被 NACK

1 = 启用广播呼叫, 地址 8'b0000_0000 会被 ACK

注: 该位仅在 I2C 作为 slaver 时起作用。

11.4.2 控制寄存器 2 (I2CxCR2,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxCR2	SxMLS	SxSDIS	CxACK	CxADDR10	SxMODE[1:0]		CxOAR[9:8]	
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit 7 **SxMLS**: SPI 数据移位顺序控制位

0 = MSB 优先

1 = LSB 优先。

Bit 6 **SxSDIS**: SPI slaver 片选信号选择

0 = 不需要片选信号, 此时 SSx 用作普通 IO, SPI slaver 应用于单主机的场景

1 = 片选信号来自于 SSx PIN, SPI slaver 应用于多主机的场景。

Bit 5 **CxACK**: I²C 主机是否继续读传输

0 = 主机继续读传输

1 = 主机结束读传输。

Bit 4 **CxADDR10**: I²C 地址模式

0 = 7 位地址模式操作

1 = 10 位地址模式操作。

Bit [3:2] **SxMODE[1:0]**: SPI 4 种模式选择

SxMODE[1:0]	SPI 4 种模式选择	Cpha /cpol
00	SLR: 时钟空闲时为低电平, 上升沿发送, 下降沿接收	1 0
01	SHR: 时钟空闲时为高电平, 上升沿发送, 下降沿接收	0 1
10	SLF: 时钟空闲时为低电平, 下降沿发送, 上升沿接收	0 0
11	SHF: 时钟空闲时为高电平, 下降沿发送, 上升沿接收	1 1

Bit [1:0] **CxOAR[9:8]**: 10 位地址高两位

11.4.3 本机地址寄存器 (I2CxOAR,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxOAR	CxOAR[7:1]							CxOAR0
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit [7:1] **CxOAR[7:1]**: 接口地址的 7:1

Bit 0 **CxOAR[0]**: 接口地址的 0 位

0 = 7 位地址模式: 无所谓

1 = 10 位地址模式: 地址的 0 位

11.4.4 预分频寄存器 (I2CxPRESC,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxPRESC	-	-	-	-	CxPRESC3	CxPRESC2	CxPRESC1	CxPRESC0
读/写	-	-	-	-	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit [7:4] 保留, 必须保持复位时的值。

Bit [3:0] **CxPRESC[3:0]**: 时序预分频器

此字段用于分频高频时钟产生 SCL 和 SCK 时钟

SCL 的频率为 $f_{SCL} = F_{I2CCLK} / (5 * 2^{PRESC})$, 如果高频时钟为 24MHz, SCL 的频率范围为: 146Hz-1MHz

SCK 的频率为 $f_{SCK} = F_{I2CCLK} / (2^{PRESC+1})$ 如果高频时钟为 24MHz, SCK 的频率范围为: 732.4Hz-3MHz。

注: PRESC [3:0]为 0 和 1 硬件会自动设为 2。

11.4.5 数据缓存寄存器 (I2CxDBUF,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxDBUF	I2CxDBUF[7:0]0							
读/写	RW	RW	RW	RW	RW	RW	RW	RW
复位后	0	0	0	0	0	0	0	0

Bit[7:0] **I2CxDBUF[7:0]**: 数据缓存寄存器

11.4.6 状态寄存器 (I2CxSR,x=0/1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CxSR	CxROV	CxAL	CxDIR	CxAD	CxBF	SxIF	CxNACKF	CxADDF
读/写	R	R	R	R	R	R	R	R
复位后	0	0	0	0	0	0	0	0

Bit 7 **CxROV**: 溢出标志

0 = 接收 FIFO 未溢出

1 = 接收 FIFO 溢出。

注: CxROV 为 1 后, 只要软件读了 FIFO, CxROV 由硬件清零。

Bit 6 **CxAL**: 主机仲裁失败标志

0 = 主机仲裁成功

1 = 主机仲裁失败。

Bit 5 **CxDIR**: 读写指示

0 = 从机接受到主机写操作

1 = 从机接受到主机读操作

注: 该位只在 I2C 作为 slave 有效。

Bit 4 **CxAD**: 数据地址标志

0 = 主机发送的是地址

1 = 主机发送的是数据。

Bit 3 **CxBF**: I²C 中断标志

I2C 发送状态

0 = 发送 FIFO 不为空

1 = 发送 FIFO 为空时 (CxNT+1 个字节发送完成), 写 DBUF 将 CxBF 清 0

I2C 接收状态

0 = 接收 FIFO 还未接收到 CxNT+1 个字节

1 = 接收 FIFO 接收到 CxNT+1 个字节, 接收 FIFO 为空将 CxBF 置 0

注: i2c master 读操作, 在发送完地址后 CxBF 会置 1, 需要写 dbuf 寄存器启动接收, 同时 CxBF 标志会被硬件清零。

Bit 2 **SxIF**: SPI 传输完成标志

该标志在 SPI 传输完成 CxNT 个字节时由硬件置 1。

Bit 1 **CxNACKF**: 收到 NACK 标志

该标志表示在一个字节传输后收到一个 NACK 的时候由硬件置 1。

Bit 0 **CxADDF**: 地址匹配 (从机模式)

该标志在收到的从机地址与其中一个有效的从机地址匹配的时候, 由硬件置 1。

11.5 唤醒及休眠模式下通讯

当 I2CxCR1.7 为 1 时，SCL、SDA 线的低电平会唤醒 CPU，并开始通讯，通讯期间 CPU 无法进入休眠模式。

当 I2CxCR1.6 为 1 时，数据缓存寄存器不为空，开始通讯，通讯期间 CPU 无法进入休眠模式。

12 PWM2

12.1 概述

M9F951 有 6 个单路 PWM,或者 3 路互补的 PWM。

- 重要寄存器受保护寄存器 PWM2RLDEN 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
- 提供 PWM 周期匹配、归零匹配和占空比比较匹配中断\
- 提供故障检测功能可紧急关闭 PWM 输出
- 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 ADC 转换

12.2 寄存器

PWM2CR0 控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2CR0	-	-	PTMOD[1:0]		PTCLK[1:0]		POSTPS[1:0]	
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

- Bit [5:4] **PTMOD:** PWM2 计数器工作模式选择位
- 00 = 向上计数模式，此模式只会发生归零匹配
 - 01 = 中心对齐模式，此模式会发生归零匹配和周期匹配
 - 1x = 单次计数模式，只发生归零匹配
- Bit [3:2] **PTCLK:** PWM2 时基预分频选择位
- 00 = 不分频
 - 01 = 2 分频
 - 10 = 4 分频
 - 11 = 8 分频
- Bit [1:0] **POSTPS:** 后分频系数选择位
- 00 = 无后分频
 - 01 = 中断、重载信号与事件触发信号 2 分频
 - 10 = 中断、重载信号与事件触发信号 4 分频
 - 11 = 中断、重载信号与事件触发信号 8 分频

注：PWM2 时基为 Fosch。

PWM2CR1 控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2CR1	POUT	-	PWM2_[2:0]NS			PWM2_[2:0]PS		
读/写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	-	0	0	0	0	0	0

Bit 7 **POUT**: PWM2 输出模式（独立/互补）控制位

0 = 3 通道互补输出

1 = 6 路独立输出

注：互补模式下 P2_xN 的占空比由与其互补的 P2_xP 的占空比寄存器控制。

Bit [5:3] **PWM2_xNS**: (x=0/1/2)

0 = 占空比时区低电平，非占空比时区高电平

1 = 占空比时区高电平，非占空比时区低电平

Bit [2:0] **PWM2_xPS**: (x=0/1/2)

0 = 占空比时期高电平，非占空比时期低电平

1 = 占空比时期低电平，非占空比时期高电平

PWM2CR2 控制寄存器 2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2CR2	-	-	PEAD	ZEAD	PDLDEN	-	DT[1:0]	
读/写	-	-	R/W	R/W	R/W	-	R/W	R/W
复位后	-	-	0	0	0	-	0	0

Bit 5 **PEAD:** PWM2 周期匹配时启动 AD 转换使能位（只在中心对齐模式）

0 = 不使能

1 = 使能

Bit 4 **ZEAD:** PWM2 归零匹配启动 AD 转换使能位

0 = 不使能

1 = 使能

Bit 3 **PDLDEN:** PWM2 计数器周期匹配重载占空比死区寄存器使能位

0 = 不使能

1 = 使能

Bit [1:0] **DT[1:0]:** 死区寄存器方式选择位

00 = PWM2 的前死区由 PWM2DT0 设置，PWM2 的后死区由 PWM2DT1 设置

01 = PWM2_1/PWM2_2 的前死区和后死区都由寄存器 PWM2DT0 设置，PWM2_0 的前死区和后死区都由寄存器 PWM2DT1 设置

10 = PWM2_0/PWM2_2 的前死区和后死区都由寄存器 PWM2DT0 设置，PWM2_1 的前死区和后死区都由寄存器 PWM2DT1 设置

11 = PWM2_0/PWM2_1 的前死区和后死区都由寄存器 PWM2DT0 设置，PWM2_2 的前死区和后死区都由寄存器 PWM2DT1 设置

注: DT[1:0]只在互补模式下有效，因为在独立模式下无死区。PWM2 的前死区只会影响 PWM2_0P、PWM2_1P、PWM2_2P 的上升沿；PWM2 的后死区只会影响 PWM2_0N、PWM2_1N、PWM2_2N 的上升沿。这里的上升沿是对应极性控制位为 0，若对应极性控制位为 1，只是将极性控制位为 0 取反，不包含死区。

PWM2OE 端口使能极性控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2OE	PWM2EN	PWM2S	PWM2_[2:0]NOE			PWM2_[2:0]POE		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **PWM2EN:** PWM2 时基使能位

0 = 关闭计数器

1 = 使能计数器

Bit 6 **PWM2S:** PWM2 输出选择

0 = A 组输出

1 = B 组输出

Bit [5:3] **PWM2_xNOE:** PWM2_xN 输出使能位(x=0/1/2)

0 = PWM2_xN 引脚作为普通 IO 口

1 = PWM2_xN 引脚作为 PWM 输出口(无需设置对应的方向寄存器位)

Bit [2:0] **PWM2_xPOE:** PWM2_xP 输出使能位(x=0/1/2)

0 = PWM2_xP 引脚作为普通 IO 口

1 = PWM2_xP 引脚作为 PWM 输出口(无需设置对应的方向寄存器位)

PMANUALCR0 端口控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PMANUALCR0	-	-	PMANUAL2_[2:0]N			PMANUAL2_[2:0]P		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [5:3] **PMANUAL2_xN:** PWM2_xN 口输出控制位 (只有在 PWM2_xNOE=1 有效, 修改立即生效)

0 = PWM2_xN 口输出 PWM 波形

1 = PWM2_xN 口输出由 PMANUALCR1 寄存器的 POUT2_xN 控制

Bit [2:0] **PMANUAL2_xP:** PWM2_xP 口输出控制位 (只有在 PWM2_xPOE=1 有效, 修改立即生效)

0 = PWM2_xP 口输出 PWM 波形

1 = PWM2_xP 口输出由 PMANUALCR1 寄存器的 POUT2_xP 控制

PMANUALCR1 端口控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PMANUALCR1	-	-	POUT2_[2:0]N			POUT2_[2:0]P		
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

Bit [5:3] **POUT2_xN**: 决定 PWM2_xN 口的输出电平 (x=0/1/2)

0 = 输出 0

1 = 输出 1

Bit [2:0] **POUT2_xP**: 决定 PWM2_xP 口的输出电平 (x=0/1/2)

0 = 输出 0

1 = 输出 1

注：必须先写 PMANUALCR1 寄存器，后写 PMANUALCR0 寄存器。

FLTCR 故障检测寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLTCR	FLT1EN	-	FLT2EN	FLT2S	FLT2DEB[1:0]		FLTM	FLTSTAT
读/写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	-	0	0	0	0	0	0

Bit 7 **FLT1EN:** 故障检测 1 功能使能位

0 = 关闭

1 = 使能(比较器 0 滤波后的输出作为 PWM2 故障检测 1 输入源, 高电平有效)

Bit 5 **FLT2EN:** 故障检测 2 功能使能位

0 = 关闭

1 = 使能, 输入源为 FLT 引脚

Bit 4 **FLT2S:** 故障检测 2 有效电平选择位

0 = 高电平有效

1 = 低电平有效

Bit [3:2] **FLT2DEB[1:0]:** 故障检测 2 输入滤波参数选择

00 = 无滤波

01 = 滤波时间为: 1us (时钟源为 24Mhz); 1.5us (时钟源为 16Mhz) (时间为约数)

10 = 滤波时间为: 4us (时钟源为 24Mhz); 6us (时钟源为 16Mhz) (时间为约数)

11 = 滤波时间为: 16us (时钟源为 24Mhz); 24us (时钟源为 16Mhz) (时间为约数)

Bit 1 **FLTM:** 检测功能模式选择

0 = 锁存模式, 当检测到故障输入有效, FLTSTAT 被硬件置 1, PWM2 将立即停止输出, 当故障输入变为无效时状态保持不变, 只有当 FLTSTAT 被软件清零后, PWM2 波形才会在 PWM2 时基计数器归 0 时刻恢复输出。(当故障输入一直有效时 FLTSTAT 无法被软件清除)

1 = 逐次模式, PWM2 输出直接由故障检测输入端来控制, 如果故障输入有效, FLTSTAT 被硬件置 1, 立即关闭 PWM2 输出。如果故障输入变为无效, FLTSTAT 自动清 0, PWM2 波形将在 PWM2 时基计数器归 0 时刻自动恢复输出

Bit 0 **FLTSTAT:** 故障检测标志位

0 = 模块正常输出

1 = 出现过流, PWM2 模块处于停止输出状态, 如果在锁存模式, 可软件清 0, 恢复 PWM2 输出

注: 故障检测输入 1 和 2 可同时打开, 任一信号有效都会关闭 PWM2 模块输出。

PWM2INTCR 中断控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2INTCR	PWMP1E	PWMZ1E	PTDD2_21E	PTUD2_21E	PTDD2_11E	PTUD2_11E	PTDD2_01E	PTUD2_01E
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 PWMP1E: PWM2 周期匹配中断使能位（在中心对齐模式，计数器的值等于周期寄存器的值，称为周期匹配）

0 = 关闭

1 = 使能

Bit 6 PWMZ1E: PWM2 归零中断使能位（当计数器的值变为 0 的时候，称为归零）

0 = 关闭

1 = 使能

Bit 5 PTDD2_21E: PWM2 时基计数器减计数时与占空比寄存器 PWM2_2PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

Bit 4 PTUD2_21E: PWM2 时基计数器增计数时与占空比寄存器 PWM2_2PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

Bit 3 PTDD2_11E: PWM2 时基计数器减计数时与占空比寄存器 PWM2_1PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

Bit 2 PTUD2_11E: PWM2 时基计数器增计数时与占空比寄存器 PWM2_1PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

Bit 1 PTDD2_01E: PWM2 时基计数器减计数时与占空比寄存器 PWM2_0PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

Bit 0 PTUD2_01E: PWM2 时基计数器增计数时与占空比寄存器 PWM2_0PDL/H 匹配时中断使能位

0 = 关闭

1 = 使能

PWM2INTF 中断标志寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2INTF	PWMPIF	PWMZIF	PTDD2_2IF	PTUD2_2IF	PTDD2_1IF	PTUD2_1IF	PTDD2_0IF	PTUD2_0IF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **PWMPIF**: PWM2 周期匹配中断标志位

0 = 无周期匹配中断

1 = 发生周期匹配中断

Bit 6 **PWMZIF**: PWM2 归零中断标志位

0 = 无归零中断

1 = 发生归零中断

Bit 5 **PTDD2_2IF**: PWM2 时基计数器减计数时与占空比寄存器 PWM2_2PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTUD2IF 信号清 0.

1 = 发生中断

Bit 4 **PTUD2_2IF**: PWM2 时基计数器增计数时与占空比寄存器 PWM2_2PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTDD2IF 信号清 0.

1 = 发生中断

Bit 3 **PTDD2_1IF**: PWM2 时基计数器减计数时与占空比寄存器 PWM2_1PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTUD1IF 信号清 0.

1 = 发生中断

Bit 2 **PTUD2_1IF**: PWM2 时基计数器增计数时与占空比寄存器 PWM2_1PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTDD1IF 信号清 0.

1 = 发生中断

Bit 1 **PTDD2_0IF**: PWM2 时基计数器减计数时与占空比寄存器 PWM2_0PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTUD0IF 信号清 0.

1 = 发生中断

Bit 0 **PTUD2_0IF**: PWM2 时基计数器增计数时与占空比寄存器 PWM2_0PDL/H 匹配时中断标志位

0 = 未发生中断或发生过但被 PTDD0IF 信号清 0.

1 = 发生中断

注: (1) 即使 PWM2 中断允许位为 0, 中断标志位也会置 1, 但中断不会响应。

(2) PTDD2_xIF 置 1 时, 硬件会自动将 PTUD2_xIF 清零; 同样, PTUD2_xIF 置 1 时, 硬件会自动将 PTDD2_xIF 清零; 因此通过判断这两个标志位可以判断目前 PWM2 波形处在占空比时区还是非占空比时区。

PWM2PRL 周期寄存器低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2PRL	PWM2PR[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

PWM2PRH 周期寄存器高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2PRH	PWM2PR[15:8]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

PWM2DTxL 死区寄存器低位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DTxL	PWM2DTx[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

PWM2DTxH 死区寄存器高位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DTxH	PWM2DTx[11:8]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2_xPL 占空比寄存器低位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2_xPL	PWM2_xP[7:0]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2_xPH 占空比寄存器高位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2_xPH	PWM2_xP[15:8]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2_xNL 占空比寄存器低位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2_xNL	PWM2_xN[7:0]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2_xNH 占空比寄存器高位(x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2_xNH	PWM2_xN[15:8]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2RLDEN 寄存器修改和重载控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2RLDEN	PWM2RLDEN[7:0]							
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

PWM2RLDEN = 0x55: 允许软件对模块寄存器的修改

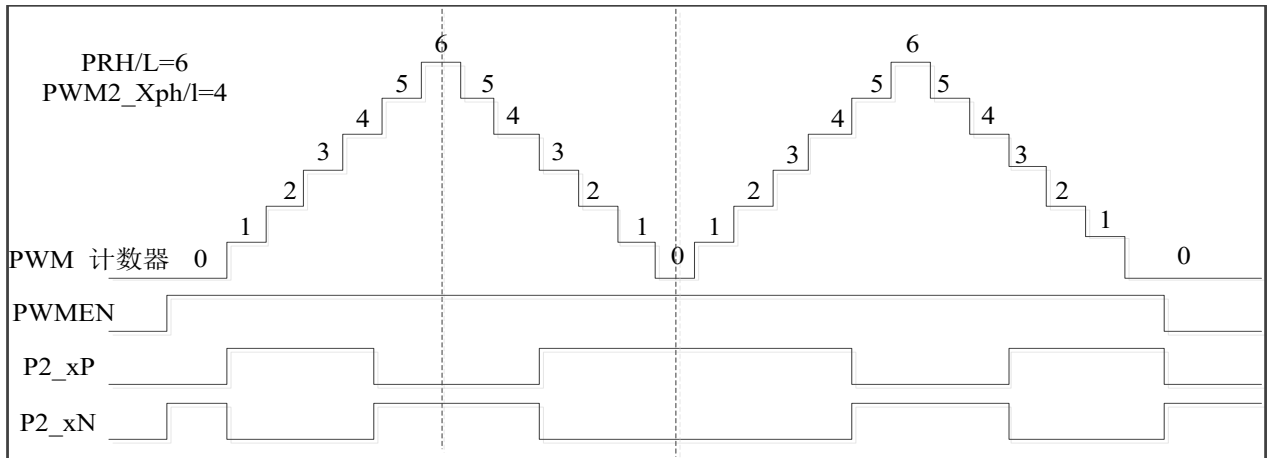
PWM2RLDEN = 0xAA: 允许模块带缓冲的寄存器的重载

注: (1) 本模块的寄存器, 除了中断标志寄存器 PWM2INTF 和手动控制寄存器 PMANUALCR1 外, 其它寄存器的修改有限制条件, 只有当 PWM2RLDEN = 0x55 时才允许软件修改, 否则修改无效。

(2) 本模块中有一些寄存器带有缓冲寄存器, 包括占空比寄存器、周期寄存器和 PWM 时钟预分频寄存器位段 PTCLK[1:0]、死区寄存器、以及死区模式选择位段 DT[1:0]。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 PWM2RLDEN = 0xAA 时才允许重载。

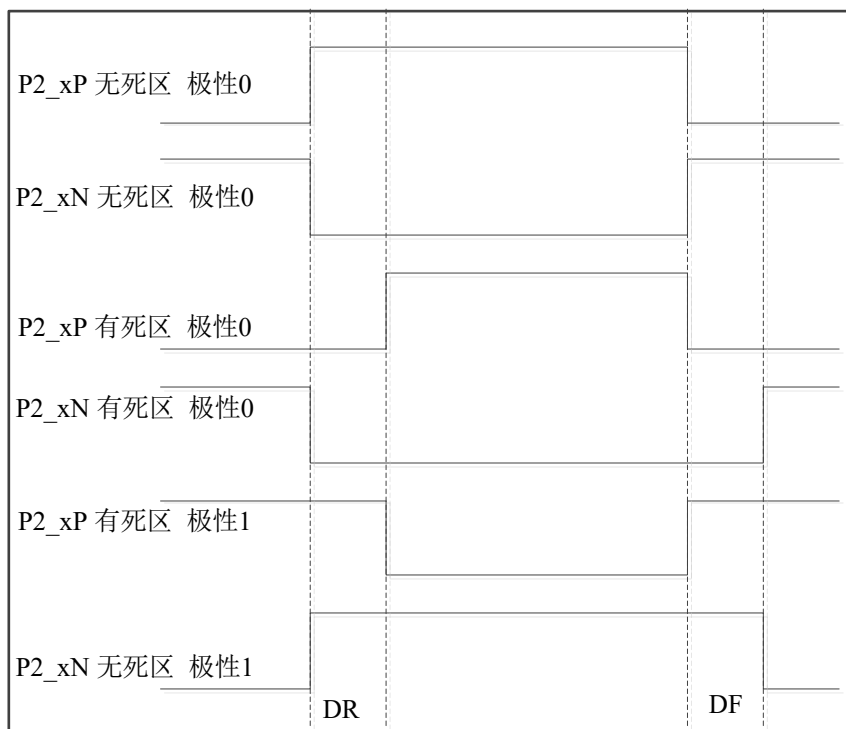
(3) 未进行说明的寄存器复位值均为 0。

12.3 图示说明



中心对齐互补模式示例图

如上图所示：占空比时区为 PWM2 计数器小于 PWM2_xPH/L，非占空比时区为 PWM2 计数器大于等于 PWM2_xPH/L（中心对齐模式输出以周期匹配为中心，在一个周期内中心对称）。



不同极性设置下的死区波形

如图所示，DR 为前死区，DF 为后死区。

12.4 使用步骤

- 1、将保护寄存器 PWM2RLDEN 设为 0x55
- 2、写 PWM2CR0 设置预分频，后分频及计数模式
写 PWM2CR1 设置输出模式和输出极性
写 PWM2PRH/L 设置周期
写 PWM2_xPH/L 设置占空比
如果需要死区：
 写 PWM2CR2 设置死区
 写死区寄存器
如果独立输出
 写 PWM2_xNH/L 设置占空比
如果需要中断
 开启中断配置寄存器 option 见 5.2
 写 PWM2INTCR 设置中断使能位
如果需要故障检测
 写 FLTCR 设置检测模式功能
- 3、写 PWM2OE 开启计数，使能输出
- 4、写 PWM2RLDEN=0xaa，将待缓存的寄存器的值载入

13 模数转换器(ADC)

13.1 概述

M9F951有一个28路外部通道(AIN0~AIN27)和4路内部通道(VDD_DIV, VREF, TEMP和GND) 12位分辨率的A/D 转换器, 可以将模拟信号转换成12位数字信号。进行AD转换时, 首先要选择输入通道, 然后启动AD转换。转换结束后, 系统自动将EOC设置为“1”, 并将转换结果存入寄存器ADH和寄存器ADL中。

13.2 ADCON0 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADEN	ADS	ADFM	CHS[4:0]				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	x	0	1	0	0	1	1

Bit 7 **ADEN:** ADC使能控制位

0 = 关闭ADC

1 = 使能ADC

Bit 6 **ADS:** ADC 启动位

0 = 停止, 转换完成自动清零

1 = 开始 (每次写入1将重新启动ADC)。

Bit 5 **ADFM:** 数据格式选择位

0 = ADRES = {ADH[7:0], ADL[7:4]}; ADL[3:0] = 0

1 = ADRES = {ADH[3:0], ADL[7:0]}; ADH[7:4] = 0

Bit [4:0] **CHS[4:0]:** ADC 输入通道选择位

[00000] ~ [11011] = AIN0 ~ AIN27

[11100] = VDD/4

[11101] = vref

[11110] = temp

[11111] = GND

注: 若 ADENB = 1, 用户应设置 IOA.n/AINn 为无上拉电阻的输入模式, 系统不会自动设置。若已经设置了 ANSEL.n, IOA.n/AINn 的数字 I/O 功能都是隔离开来的。

13.3 ADCON1 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	VHS2	ADCKS2	ADCKS1	ADCKS0	VREMS1	VREMS0	VHS1	VHS0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [6:4] **ADCKS[2:0]:** ADC 时钟源选择位

ADCKS[2:0]	ADC 时钟源选择
000	Fcpu
001	Fcpu/2
010	Fcpu/4
011	Fcpu/8
100	Fcpu/16
101	Fcpu/32
110	Fcpu/64
111	

Bit [3:2] **VREMS[1:0]:** ADC 参考电压模式选择位

VREMS[1:0]	ADC 参考电压模式
00	VDD
01	内部参考电压
10	外部参考电压
11	内部参考与外部参考连接

Bit [7,1:0] **VHS[2:0]:** ADC 内建基准电平选择位。

VHS[1:0]	内建 VREF 基准电平
000	关闭内部参考
001	2.0V
010	3.0V
011	4.0V
100	关闭内部参考
101	未定义
110	
111	

注：(1) 若由 VHS[1:0]控制选择的内部 VREF 电平高于 VDD，内部 VREF 为 VDD。

例：VHS[1:0] = 11（内部 VREF = 4.0V），VDD = 3.0V，则实际内部 VREF = 3.0V。

(2) 12 位 AD 转换时间 = 16 个 AD 时钟。

13.4 ADCON2 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2		TRGEN	VREFO	CMPHDEN	ADVOS[3:0]			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [6] **RTGEN:** 触发事件触发AD转换

1 = 允许

0 = 不允许

Bit [5] **VREFO:** 内部参考输出

1 = 参考电压从IOE1端口输出，可用作LDO

0 = 关闭内部参考输出

Bit [4] **CMPHDEN:** AD内部比较电路电流

1 = 高速大电流

0 = 低速小电流

Bit [3:0] **ADVOS[3:0]:** ADC失调补偿寄存器

13.5 ADH ADC 数据高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADH	-	-	-	-	-	-	-	-
读/写	R	R	R	R	R	R	R	R
复位后	X	X	X	X	X	X	X	X

13.6 ADL ADC 数据低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADL	-	-	-	-	-	-	-	-
读/写	R	R	R	R	R	R	R	R
复位后	X	X	X	X	X	X	X	X

注：ADH/ADL 的数据格式与 ADM 相关，当 ADFM=1 时，ADH[7:4]=0,ADH[3:0]存放高四位数据
ADL[7:0]存放低 8 位数据；当 ADFM=0 时，ADH[7:0]存放高 8 位数据，ADL[7:4]存放低 4 位
数据，AD[3:0] = 0。

14 看门狗（WDT）

14.1 概述

看门狗定时器的时钟为内部独立 RC 时钟。

配置字 WDTEN 设置看门狗定时器的三种工作状态：

- （1）始终使能：在 STOP 模式下仍然工作，溢出可唤醒 STOP
- （2）STOP 下关闭
- （3）始终关闭

配置字 TWDTEN 设置看门狗的八种溢出时间：4.5ms、18ms、72ms、288ms、576ms、1152ms、2304ms、4608ms。

注：看门狗正常溢出后，程序复位到 0000H，但是在休眠模式下看门狗溢出程序是继续往下运行。

14.2 WDTC 看门狗控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCR	-	-	-	WDTT[2:0]			WDTCR0[1:0]	
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

Bit [4:2] WDTT[2:0]: 溢出时间

WDTT[2:0]	WDT 溢出时间
000	4.5mS
001	18mS
010	72mS
011	288mS
100	576mS
101	1.152S
110	2.304S
111	4.608S

Bit [1:0] WDTCR[1:0]: 看门狗控制

- 00 = 始终关闭看门狗
- 01 = 休眠模式下关闭看门狗
- 1x = 始终开启看门狗

14.3 WDTC 看门狗控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTC	WDTC[7:0]							
读/写	W	W	W	W	W	W	W	W
复位后	-	-	-	-	-	-	-	-

注：CLRWDTC 指令也可清除 WDT 定时器。

14.4 软件配置看门狗

首先要按 0x55 和 0xAA 的顺序到 CALLOCK 寄存器后，才能向 WDTCSR 寄存器写值。

```

BANKBSR    0xF           ;//切页
MOVIA      0x55
MOVAR      CALLOCK       ;//写入55H
MOVIA      0xAA
MOVAR      CALLOCK       ;//写入AAH
MOVIA      VALUE
MOVAR      WDTCSR        ;//值

```

注：重新设置 WDT 最好先关闭看门狗。

15 USB

15.1 概述

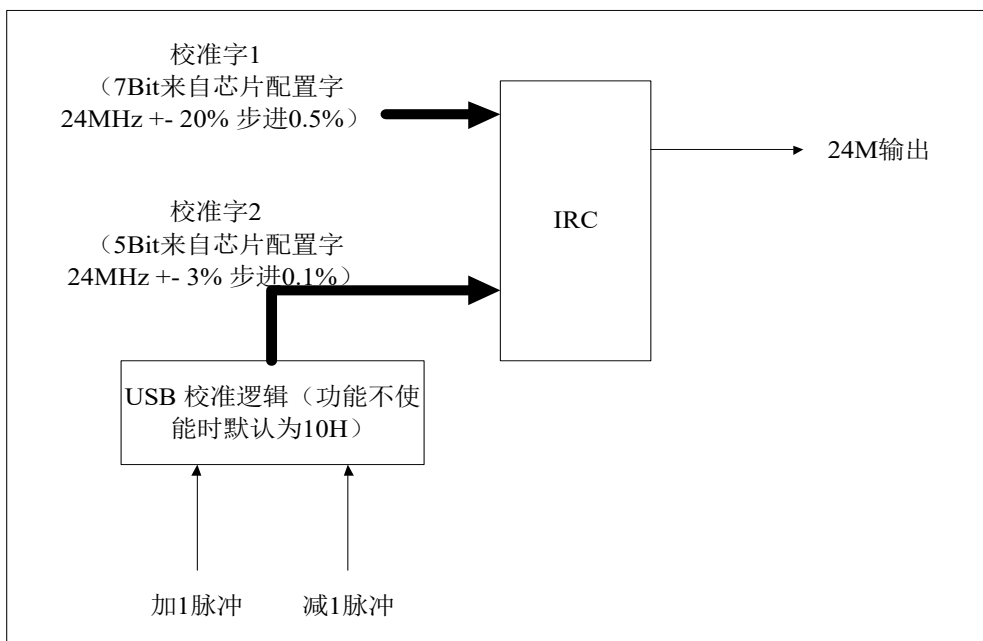
支持全速（FS）和低速（LS）。支持 EP0、EP1、EP2、EP3、EP4 和 EP5 一共 6 个端点，数据缓冲区被分配到第 8 个页面，每个端点的 IN 和 OUT 共用缓冲区，具体起始地址分配如下：

EP0	0x0800	； 32
EP1	0x0822	； 32
EP2	0x0844	； 32
EP3	0x0866	； 16
EP4	0x0878	； 64
EP5	0x08BA	； 64

为节省空间，每个端点都可以占用下一个端点的缓冲区，例如，枚举时 EP0 包长可以定义为 64 字节，这时占用了部分 EP1 的缓冲区，其它端点也一样，使用时要合理选用端点。

15.2 USB 时钟

USB FS模式下工作在48M时钟，LS模式下工作在6MHz，芯片内部高频RC振荡器已校准到24MHz，分别经过分频或倍频获得USB时钟，为了得到更精确的USB时钟，可通过使能USHC（USBCON[3]），使USB模块可利用HOST的数据速率对内部振荡器进行校准，经校准的时钟可以稳定在+0.25%的偏差范围之内。



15.3 总线连接

使能 USBCON0[0]位，DP 或 DN 上的上拉电阻将被使能（1.5K 欧），FS 模式，DP 上拉，LS 模式，DM 会上拉。HOST 通过上拉状态判断是 FS 还是 LP，然后对 USB 复位，复位完毕后开始枚举过程。

15.4 总线唤醒

使能 USBCON0[0]位，USBUSB 总线中断使能 DP 或 DN 上的上拉电阻将被使能（1.5K 欧），FS 模式，DP 上拉，LS 模式，DM 会上拉。HOST 通过上拉状态判断是 FS 还是 LP，然后对 USB 复位，复位完毕后开始枚举过程。

15.5 相关寄存器

15.5.1 SPI UCLKCON USB 校准寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UCLKCON	UCKS[1:0]		OVT[5:0]					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	1	0	0	0	0	0

Bit [7:6] UCKS [1:0]:
10 = 允许程序写入 OVT

15.5.2 USBCON USB 总线控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
USBCON	UBRST	UBSPD	UBRSM	-	USHC	UFS	USRSM	UE
读/写	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
复位后	0	0	0	-	0	0	0	0

Bit 7 **UBRST:** USB RESET状态

0 = 无RESET

1 = RESET

Bit 6 **UBSPD:** USB总线SUSPEND状态

0 = 未SUSPEND

1 = SUSPEND

Bit 5 **UBRSM:** USB总线RESUM状态（DEV发送的RESUME无该标志）

0 = 未RESUME

1 = RESUME

Bit 3 **USHC:** 内部振荡器校准功能

0 = 不使能

1 = 使能

Bit 2 **UFS:** 速度

0 = 低速（LS）

1 = 全速（FS）

Bit 1 **USRSM:** USB开始RESUME

0 = 无SUSPEND

1 = SUSPEND

Bit 0 **UE:** USB使能

0 = 不使能

1 = 使能

15.5.3 UE0CON USB EP0 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UE0CON	UE0SDF	-	-	UE0STL	UE0SOUTS	UE0SINS	UE0DOUTS	UE0DINS
读/写	R/W	-	-	R/W	R/W	R/W	R/W	R/W
复位后	0	-	-	0	0	0	1	0

- Bit 7 **UE0SDF:** USB EP0 STATUS/DATA标志
0 = 传输未完成
1 = 传输完成
- Bit 4 **UE0STL:** USB EP0的STALL操作
0 = 无操作
1 = STALL
- Bit 3 **UE0SOUTS:** USB EP0的STATUS OUT开始
0 = 无操作
1 = 开始
- Bit 2 **UE0SINS:** USB EP0的STATUS IN开始
0 = 无操作
1 = 开始
- Bit 1 **UE0DOUTS:** USB EP0的DATA/SETUP OUT开始
0 = 无操作
1 = 开始
- Bit 0 **UE0DINS:** USB EP0的DATA IN开始
0 = 无操作
1 = 开始

15.5.4 UDADR USB 设备地址寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UDADR	-	UDADR[6:0]						
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	0	0	0	0	0	0	0

- Bit [6:0] **UDADR:** USB设备地址

15.5.5 UExCON USB EPx 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UExCON	UExOUTF	UExINF	UExOUTSTL	UExINSTL	UExOUTD0	UExIND0	UExOUTS	UExINS
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	1	1	1	0

- Bit 7 **UExOUTF:** USB EPx 的OUT标志
0 = 传输未完成
1 = 传输完成
- Bit 6 **UExINF:** USB EPx 的IN标志
0 = 传输未完成
1 = 传输完成
- Bit 5 **UExOUTSTL:** USB EPx的OUT STALL
0 = 无操作
1 = STALL
- Bit 4 **UExINSTL:** USB EPx的IN STALL
0 = 无操作
1 = STALL
- Bit 3 **UExOUTD0:** USB EPx的OUT强制为DATA0
0 = 无操作
1 = 设为DATA0
- Bit 2 **UExIND0:** USB EPx的IN强制为DATA0
0 = 无操作
1 = 设为DATA0
- Bit 1 **UExOUTS:** USB EPx的 OUT开始
0 = 无操作
1 = 开始
- Bit 0 **UExINS:** USB EPx的 IN开始
0 = 无操作
1 = 开始

15.5.6 UExIOPS USB EPx 的 IN/OUT 包大小寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UExIOPS	UExIOPS[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **UExIOPS[7:0]:** 接收或发送数据包大小

15.5.7 LDO

LDO开关及电压由LDOS[3:0]设置（配置字CONFIG2[11:8]）

LDOS[1:0]: 选择LDO电压

LDOS[1:0] = 00 :3.35v

LDOS[1:0] = 01 : 3.40v

LDOS[1:0] = 10 :3.45v

LDOS[1:0] = 11 :3.50v

LDOS[2]: 始终开启LDO

LDOS[3]: USB开启时启动LDO

15.5.8 USB 输出驱动

可以选择数字IO或电流源驱动（配置字CONFIG2[12]）

TXSL=1: 数字IO

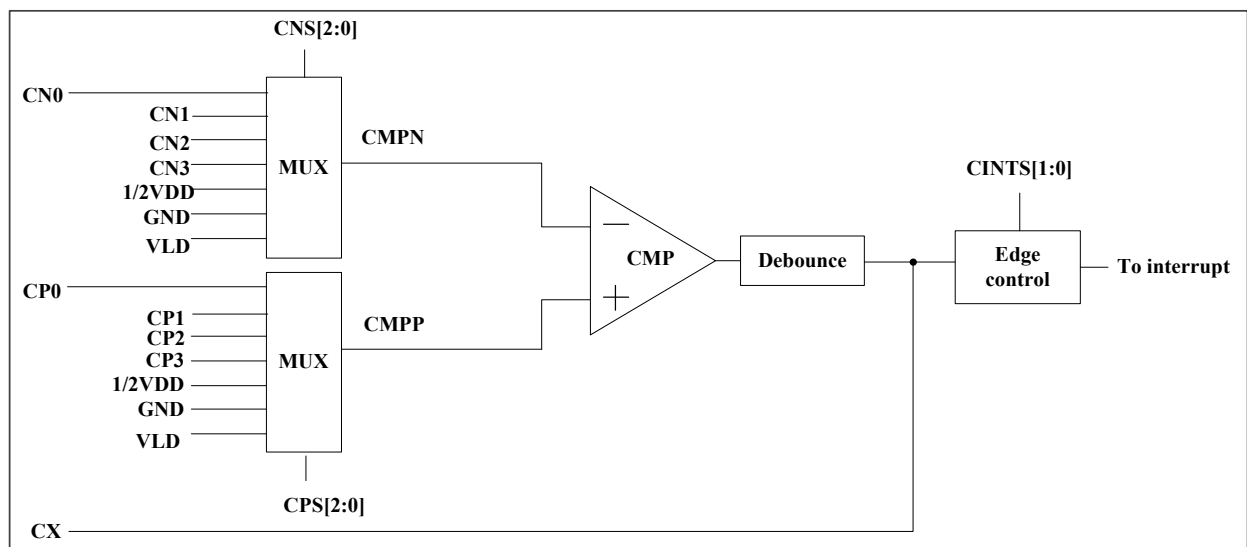
TXSL=0: 电流源

16 比较器(CMP)

16.1 概述

比较器具有多种输入源，多种参考电压选项，输出极性控制。还有多种输出中断触发和输出唤醒MCU 功能，增强了使用的灵活性，适应各种广泛的应用。

16.2 比较器框图



16.3 CMPC0 比较器控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC0	CMPEN	CMPOUT	CMPNS[2:0]			CMPPS[2:0]		
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CMPEN:** 比较器使能控制位

0 = 关闭比较器

1 = 使能比较器

Bit 6 **CMPOUT:** 比较器输出位

0 = CP脚输入电压小于CN脚

1 = CP脚输入电压大于CN脚

Bit [5:3] **CMPNS[2:0]:** 比较器反相输入信号选择位

CMPNS[2:0]	输入信号选择
000	CMPN0
001	CMPN1
010	CMPN2
011	CMPN3
100	1/2VDD
101	GND
110	VLD
111	未定义

Bit [2:0] **CMPPS[2:0]:** 比较器正相输入信号选择位

CMPPS[2:0]	输入信号选择
000	CMPP0
001	CMPP1
010	CMPP2
011	CMPP3
100	1/2VDD
101	GND
110	VLD
111	未定义

注：设计者必须在使能比较器中断之前将比较器使能，以避免未知的中断发生。

16.4 CMPC1 比较器控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC1	CMPOEN	CMPHIEN	-	CMPVLD[4:0]				
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	-	0	0	0	0	0

Bit 7 **CMPOEN:** 比较器输出使能

0 = 关闭比较器信号输出

1 = 比较器信号从端口输出

Bit 6 **CMPHIEN:** P端信号输入控制

0 = 无影响

1 = 不论CMPPS为何值，CP0通道导通

Bit [4:0] **CMPVLD[4:0]:** VLD电压选择位

CMPVLD [4:0]	电压 (V)	CMPVLD [4:0]	电压 (V)	CMPVLD [4:0]	电压 (V)	CMPVLD [4:0]	电压 (V)
00000	1.25	01000	1.65	10000	2.05	11000	2.45
00001	1.30	01001	1.70	10001	2.10	11001	2.50
00010	1.35	01010	1.75	10010	2.15	11010	2.55
00011	1.40	01011	1.80	10011	2.20	11011	2.60
00100	1.45	01100	1.85	10100	2.25	11100	2.65
00101	1.50	01101	1.90	10101	2.30	11101	2.70
00110	1.55	01110	1.95	10110	2.35	11110	2.75
00111	1.60	01111	2.00	10111	2.40	11111	2.80

16.5 CMPC2 运放/比较器控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPC2	CINTS[1:0]		-	-	-	DEB[2:0]		
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
POR	0	0	-	-	-	0	0	0

Bit [7:6] CINTS[1:0]: 比较器中断触发类型选择

CINTS[1:0]	触发类型
00	下降沿触发
01	上升沿触发
10	双边沿触发
11	无

Bit [2:0] DEB[2:0]: 比较器逻辑输出滤波设置

DEB[2:0]	滤波设置
000	关闭
001	4
010	8
011	16
100	32
101	64
110	128
111	256

注：（1）滤波时间为 $T_{deb} = T_{sys} * DEB * 3/4$ 。

（2）端口的 Cx 输出不经过滤波。

（3）滤波算法简介：系统时钟对比较器的输出进行采样，为 1 时滤波计数器加 1。为零时滤波计数器减 1，滤波器初值为设置值的一半，当滤波器计数器 $> DEB * 3/4$ 时，滤波结果为 1；滤波器计数器 $< DEB * 1/4$ 时，滤波结果为 0

（4）滤波器稳定时间 $T_{sys} * DEB/2$ 。

17 芯片配置字（OPTION BIT）

烧录选项	内容	说明
振荡器模式	HIRC(24M)+LIRC	双系统时钟
	HIRC(16M)+LIRC	
	HIRC(24M)+LXT	
	HIRC(16M)+LXT	
	HXT(24M)+LIRC	
	HXT(32M)+LIRC	
FCPU	4T (LVR \geq 2.4V)	高频模式下 CPU 速度选择； 低频模式下固定为 2T
	8T (LVR \geq 2.2V)	
	16T (LVR \geq 1.8V)	
	32T (LVR \geq 1.4V)	
	64T (LVR \geq 1.4V)	
	128T (LVR \geq 1.4V)	
	256T (LVR \geq 1.4V)	
外部复位 端口选择	使能外部复位端口	
	作为 IO 端口	
启动模式 选择	低速启动	
	高速启动	
复位向量	0x0000	
	0x7FF8	
复位电压 选择	LVR=1.4V	系统高速运行时，请选择相应较高的 LVR 电压，以保证系统的可靠性
	LVR=1.5V	
	LVR=1.6V	
	LVR=1.7V	
	LVR=1.8V	
	LVR=1.9V	
	LVR=2.0V	
	LVR=2.1V	
	LVR=2.2V	
	LVR=2.3V	
	LVR=2.4V	
	LVR=2.5V	
	LVR=3.5V	
	LVR=3.6V	
	LVR=3.7V	
	LVR=3.8V	

烧录选项	内容	说明
中断向量 选择	双中断向量	
	唯一中断向量 0x0008	
WDT 使能 选择	屏蔽 WDT 功能	
	使能，绿色或休眠模式下关闭	
	始终开启 WDT 功能	
WDT 溢出 时间	WDT 溢出时间=4.5mS	VDD=5V 典型值
	WDT 溢出时间=18mS	
	WDT 溢出时间=72mS	
	WDT 溢出时间=288mS	
	WDT 溢出时间=576mS	
	WDT 溢出时间=1152mS	
	WDT 溢出时间=2304mS	
	WDT 溢出时间=4608mS	
USB 校准 档间距	0.25%	
	0.75%	
	1.75%	
	3.75%	
LDO 电压 选择	3.35V	
	3.40V	
	3.45V	
	3.50V	
LDO 使能	关闭 LDO 功能	
	始终打开	
	USB 使能时打开	
USB 输出 驱动	电流源驱动	
	数字 IO 驱动	
0000-0FFF 写保护	不允许擦写	
	允许擦写	
1000-5FFF 写保护	不允许擦写	
	允许擦写	
6000-6FFF 写保护	不允许擦写	
	允许擦写	
7000-7FFF 写保护	不允许擦写	
	允许擦写	
仿真电压 选择	VDD 5.0V(<200mA)	
	VDD 3.3V(<300mA)	
	外供电源	

18 电性参数

18.1 极限参数

储存温度.....-50℃~125℃
 工作温度.....-40℃~85℃
 电源供应电压.....0V~5.5V
 端口输入电压.....VSS-0.3V~VDD+0.3V

18.2 直流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25℃）				
VDD	工作电压	—	Fosc = 16MHz，16T	1.8	-	5.5	V
IDD1	工作电流 1	3V	高频运行（HIRC=16M） 低频运行（LIRC=64K） FCPU=HIRC/16T	-	0.3	-	mA
		5V	全速工作	-	0.5	-	
IDD2	工作电流 2	3V	高频运行（HIRC=24M） 低频运行（LIRC=64K） FCPU=HIRC/16T	-	0.5	-	
		5V	全速工作	-	0.7	-	
IDD3	工作电流 3	3V	高频运行（HIRC=24M） 低频运行（LIRC=64K） FCPU=HIRC/4T		1.3		
		5V	全速工作		2.0		
ISP1	静态电流 1	3V	高频运行（HIRC=16M） 低频运行（LIRC=64K） STOP =1	-	0.2	-	
		5V	无唤醒源	-	0.3	-	
ISP2	静态电流 2	3V	高频停止 低频运行（LIRC=64K） STOP =1	-	2	-	uA
		5V	无唤醒源	-	6	-	
ISP3	静态电流 3	3V	高频停止 低频停止	-	0.7	-	

		5V	STOP =1 无唤醒源	-	0.9	-	
--	--	----	-----------------	---	-----	---	--

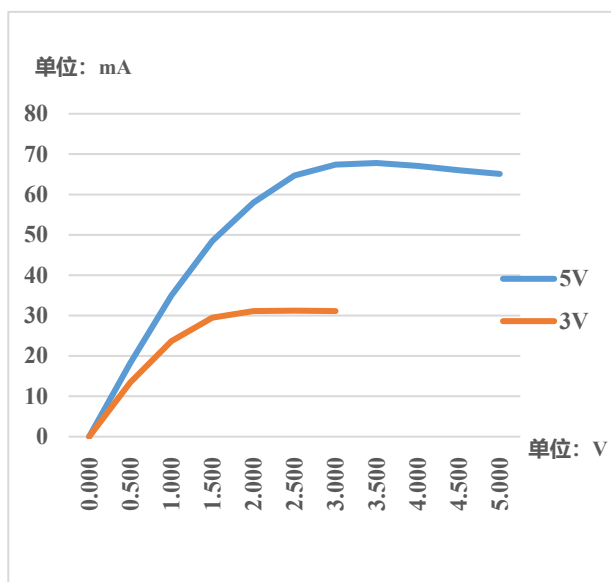
符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25℃）				
V _{IL1}	输入低电平	3V	SMT	0	-	0.3VDD	V
V _{IH1}	输入高电平	3V		0.6VDD	-	VDD	
V _{IL2}	输入低电平	5V		0	-	0.3VDD	
V _{IH2}	输入高电平	5V		0.6VDD	-	VDD	
V _{IL3}	输入低电平	3V	1.5V	0	-	1	
V _{IH3}	输入高电平	3V		1.2	-	VDD	
V _{IL4}	输入低电平	5V		0	-	1.5	
V _{IH4}	输入高电平	5V		1.8	-	VDD	
V _{IL5}	输入低电平	3V	1/2VDD	0	-	0.4VDD	
V _{IH5}	输入高电平	3V		0.5VDD	-	VDD	
V _{IL6}	输入低电平	5V		0	-	0.4VDD	
V _{IH6}	输入高电平	5V		0.5VDD	-	VDD	
I _{PH}	上拉电阻	5V	V _{IN} = GND	-	16	-	kΩ
		3V		-	50	-	
I _{PL}	下拉电阻	5V		-	40	-	
		3V		-	120	-	
I _{OLX0}	输出灌电流	5V	输出口，V _{out} =VSS+0.6V (IOA[1:0]/ IOB/IOC/IOD/IOE)	-	25	-	mA
		3V		-	17	-	
I _{OLX1}	输出灌电流	5V		-	19	-	
		3V		-	13	-	
I _{OLX2}	输出灌电流	5V		-	12	-	
		3V		-	8	-	
I _{OLX3}	输出灌电流	5V		-	4	-	
		3V		-	2.5	-	
I _{OHX0}	输出拉电流	5V	输出口，V _{out} =VDD-0.6V (IOA[1:0]/ IOB/IOC/IOD/IOE)	-	24	-	
		3V		-	15	-	
I _{OHX1}	输出拉电流	5V		-	18	-	
		3V		-	13	-	
I _{OHX2}	输出拉电流	5V		-	10	-	
		3V		-	7	-	
I _{OHX3}	输出拉电流	5V		-	5	-	
		3V		-	3	-	
I _{OLA0}	输出灌电流	5V	输出口，V _{out} =VSS+0.6V (IOA[7:2])	-	23	-	mA
		3V		-	15	-	
I _{OLA1}	输出灌电流	5V		-	180	-	
		3V		-	120	-	
I _{OLA2}	输出灌电流	5V		-	250	-	
		5V		-	250	-	

		3V		-	170	-	
IO LA3	输出灌电流	5V		-	270	-	
		3V		-	190	-	
符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25℃）				
IOHA0	输出拉电流	5V	输出口，Vout=VDD-0.6V ((IOA[7:2]))	-	25	-	mA
		3V		-	16	-	
IOHA1	输出拉电流	5V		-	38	-	
		3V		-	22	-	
IOHA2	输出拉电流	5V		-	39	-	
		3V		-	22	-	
IOHA3	输出拉电流	5V		-	38	-	
		3V		-	22	-	
IOLF0	输出灌电流	5V	输出口，Vout=VSS+0.6V (IOF[1:0])	-	65	-	mA
		3V		-	45	-	
IOLF1	输出灌电流	5V		-	55	-	
		3V		-	38	-	
IOLF2	输出灌电流	5V		-	50	-	
		3V		-	32	-	
IOLF3	输出灌电流	5V		-	35	-	
		3V		-	25	-	
IOHF0	输出拉电流	5V	输出口，Vout=VDD-0.6V ((IOF[1:0]))	-	40	-	mA
		3V		-	28	-	
IOHF1	输出拉电流	5V		-	36	-	
		3V		-	25	-	
IOHF2	输出拉电流	5V		-	30	-	
		3V		-	20	-	
IOHF3	输出拉电流	5V		-	26	-	
		3V		-	17	-	
IOLF4	输出灌电流	5V	输出口，Vout=VSS+0.6V (IOF[7:2])	-	25	-	mA
		3V		-	17	-	
IOLF5	输出灌电流	5V		-	18	-	
		3V		-	12	-	
IOLF6	输出灌电流	5V		-	12	-	
		3V		-	8	-	
IOLF7	输出灌电流	5V		-	4	-	
		3V		-	2.5	-	
IOHF4	输出拉电流	5V	输出口，Vout=VDD-0.6V ((IOF[7:2]))	-	18	-	mA
		3V		-	12	-	
IOHF5	输出拉电流	5V		-	16	-	
		3V		-	10	-	
IOHF6	输出拉电流	5V		-	9	-	
		3V		-	6	-	

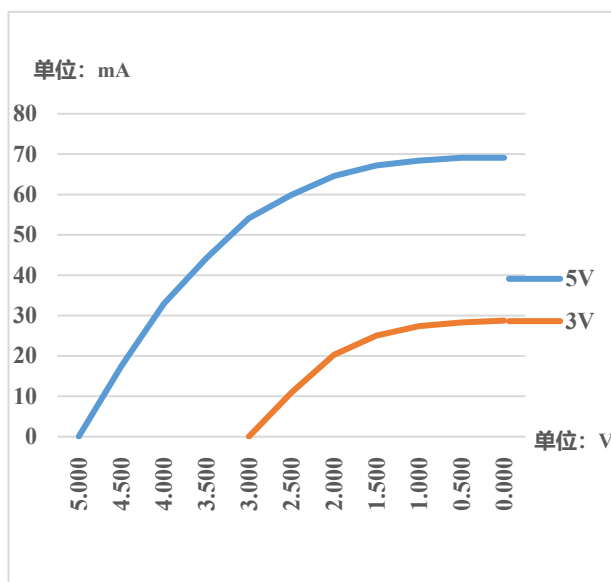
IOHF7	输出拉电流	5V		-	5	-	
		3V		-	3	-	

18.3 IO 口拉灌电流特性

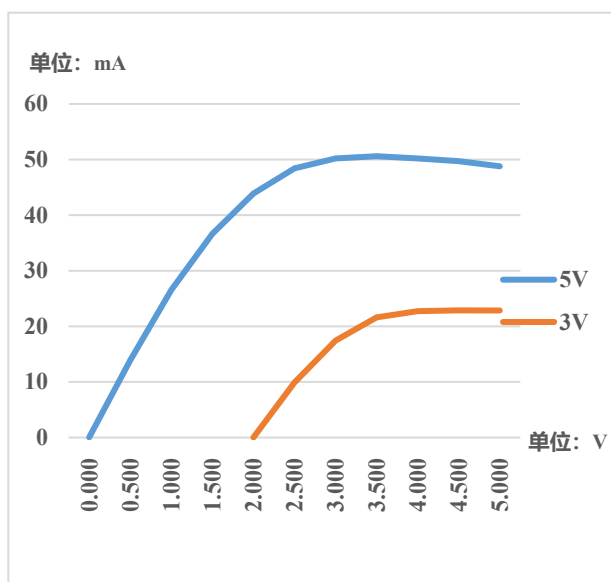
IOHX0



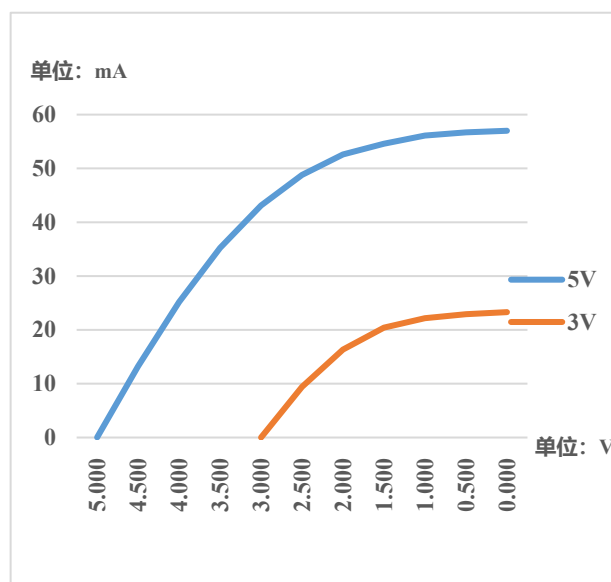
IOLX0



IOHX1

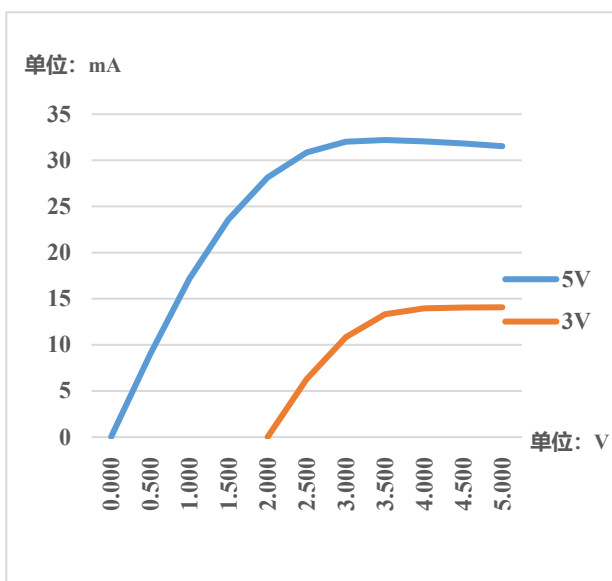


IOLX1

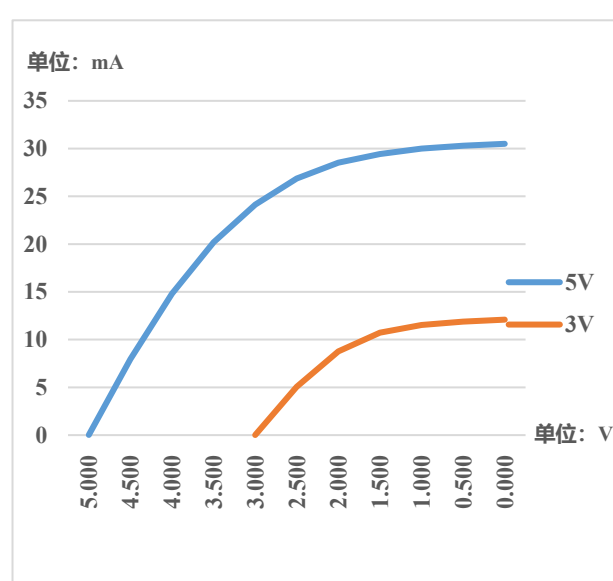


注: 具体值不做设计保证。

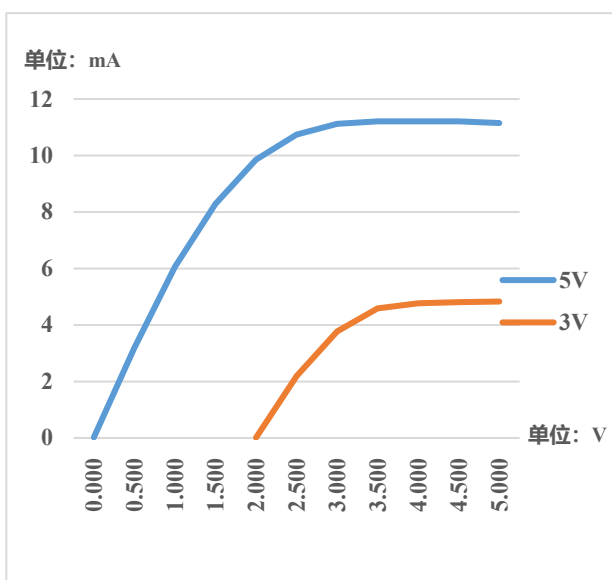
IOHX2



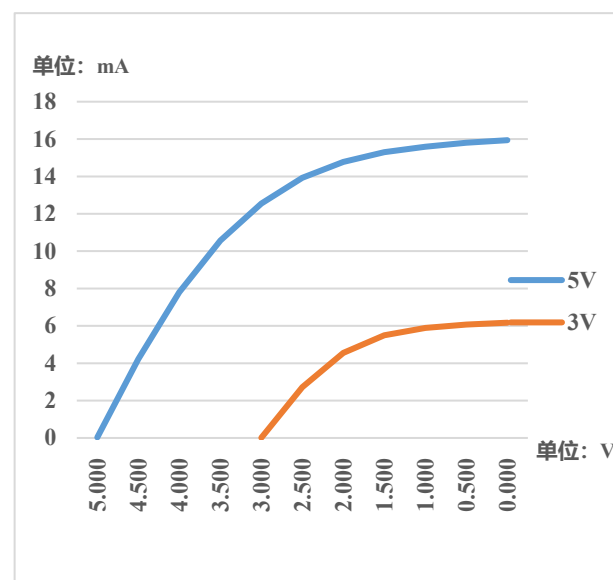
IOLX2



IOHX3



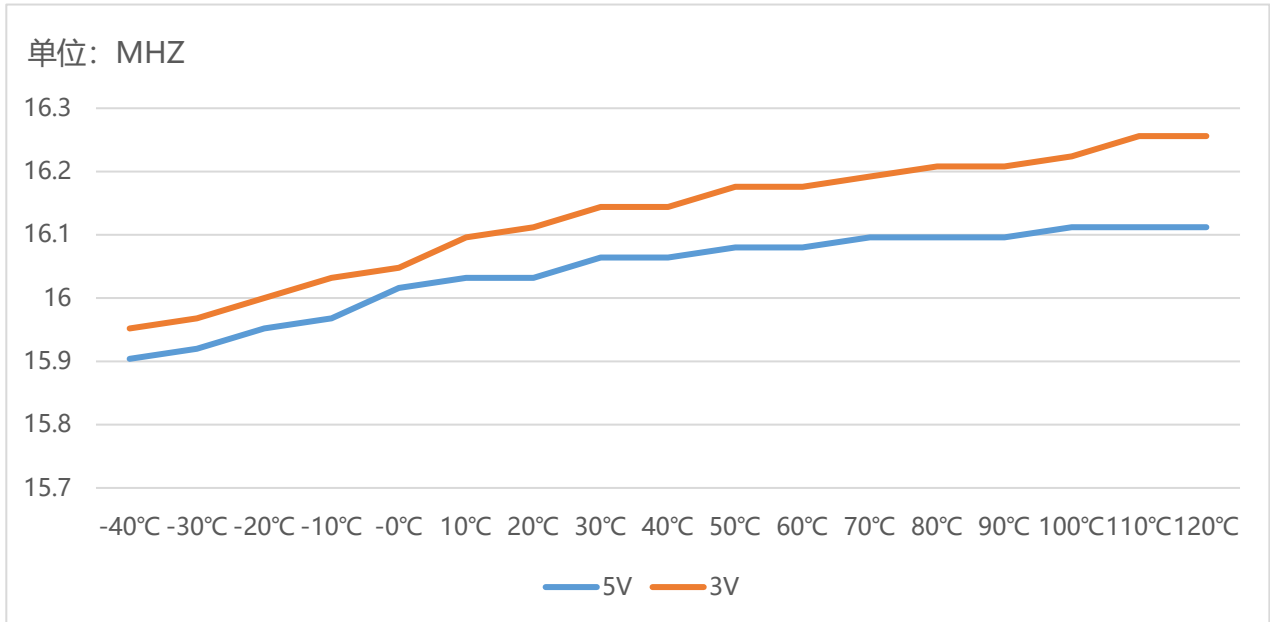
IOLX3



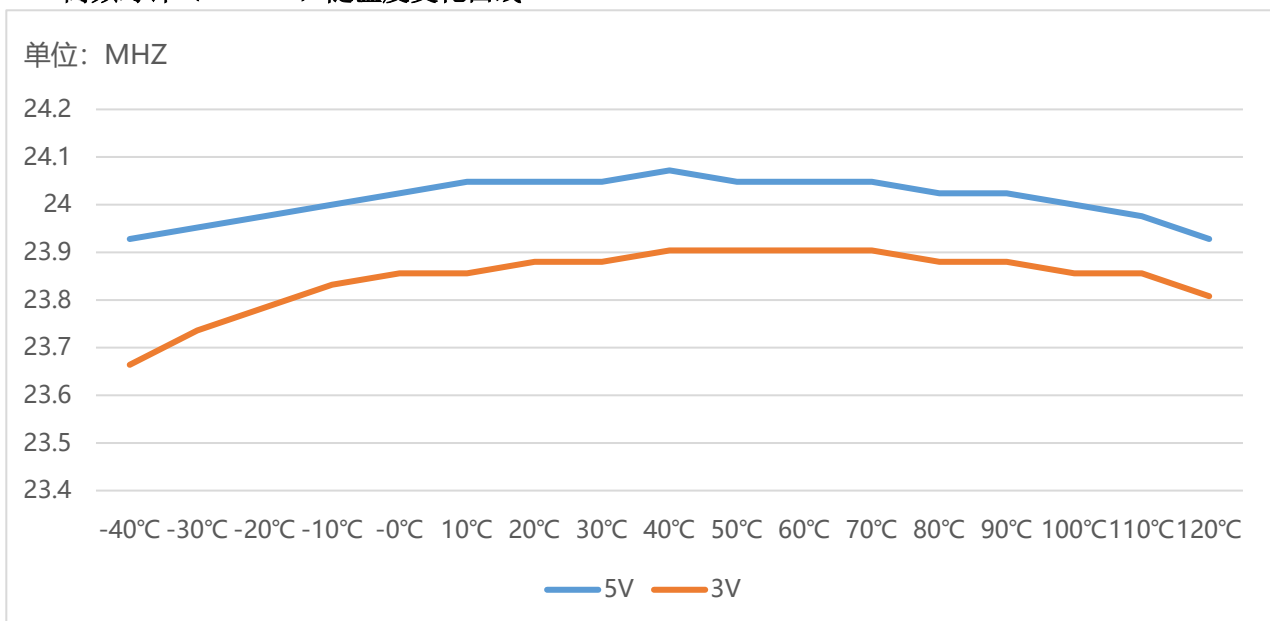
注: 具体值不做设计保证。

18.4 系统时钟特性

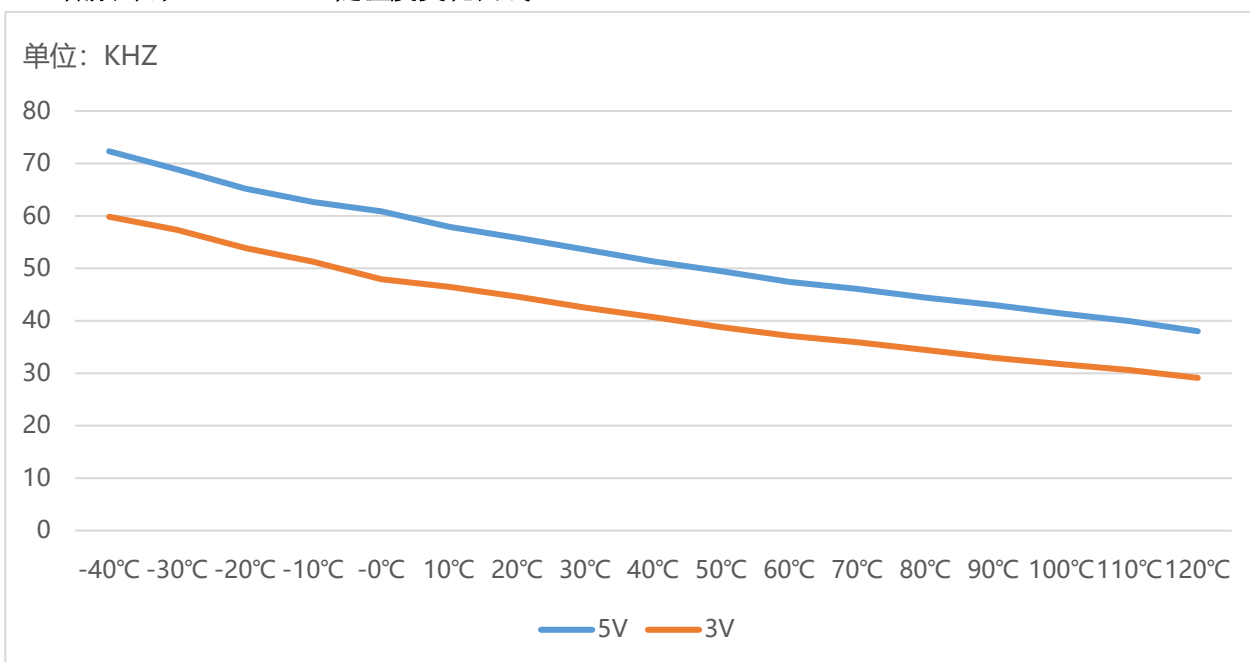
高频时钟（16MHZ）随温度变化曲线



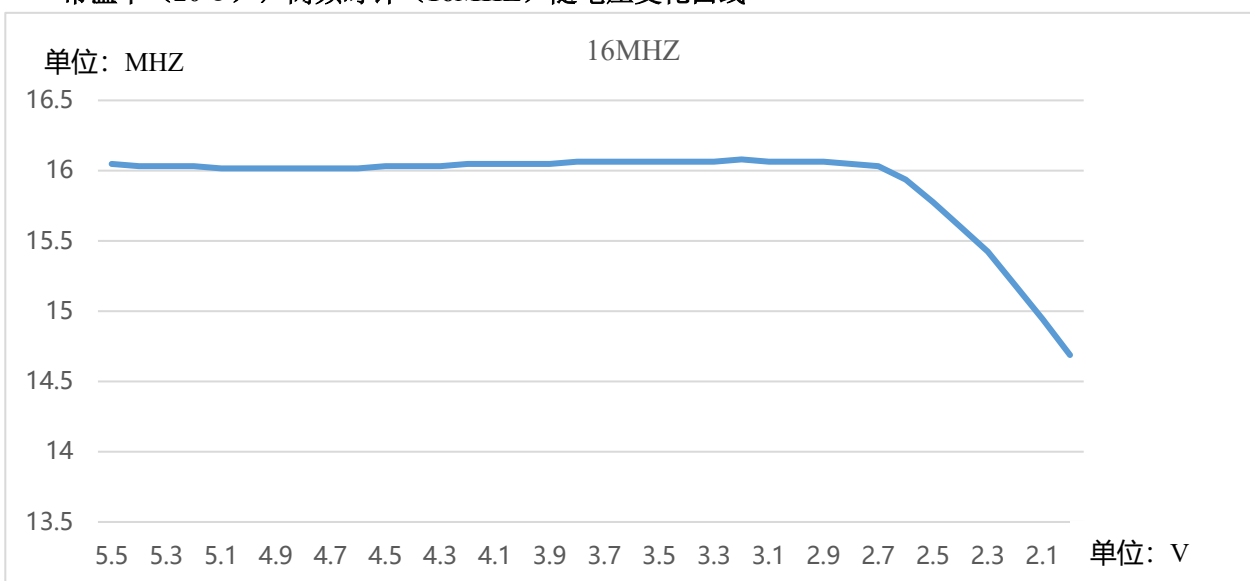
高频时钟（24MHZ）随温度变化曲线



低频时钟（64KHZ） 随温度变化曲线

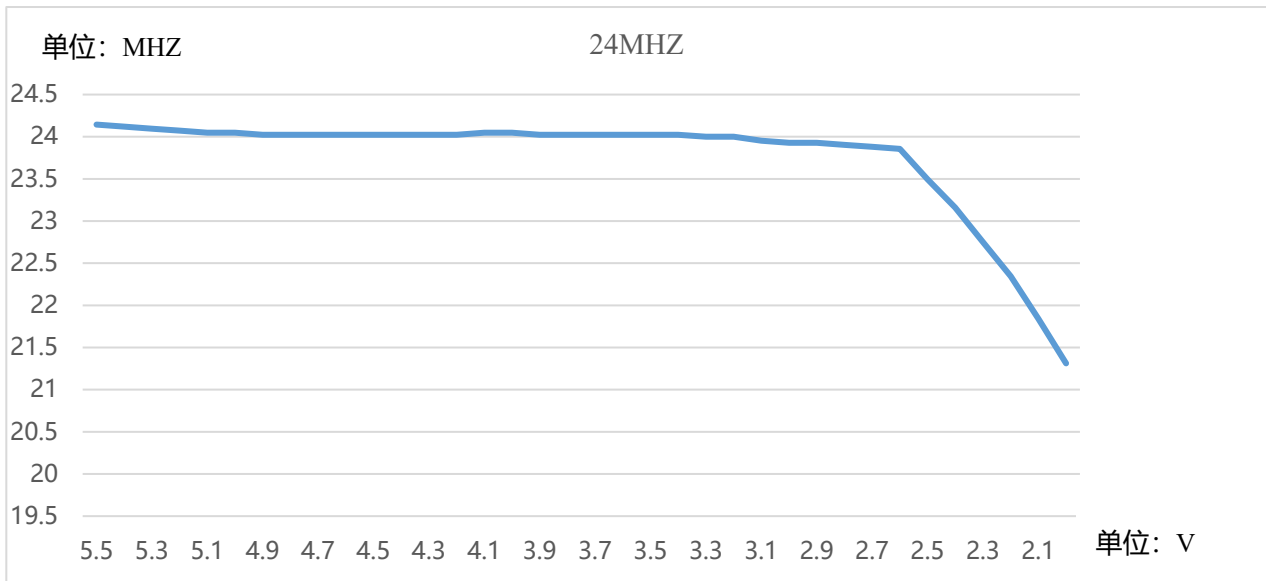


常温下（20℃），高频时钟（16MHZ）随电压变化曲线

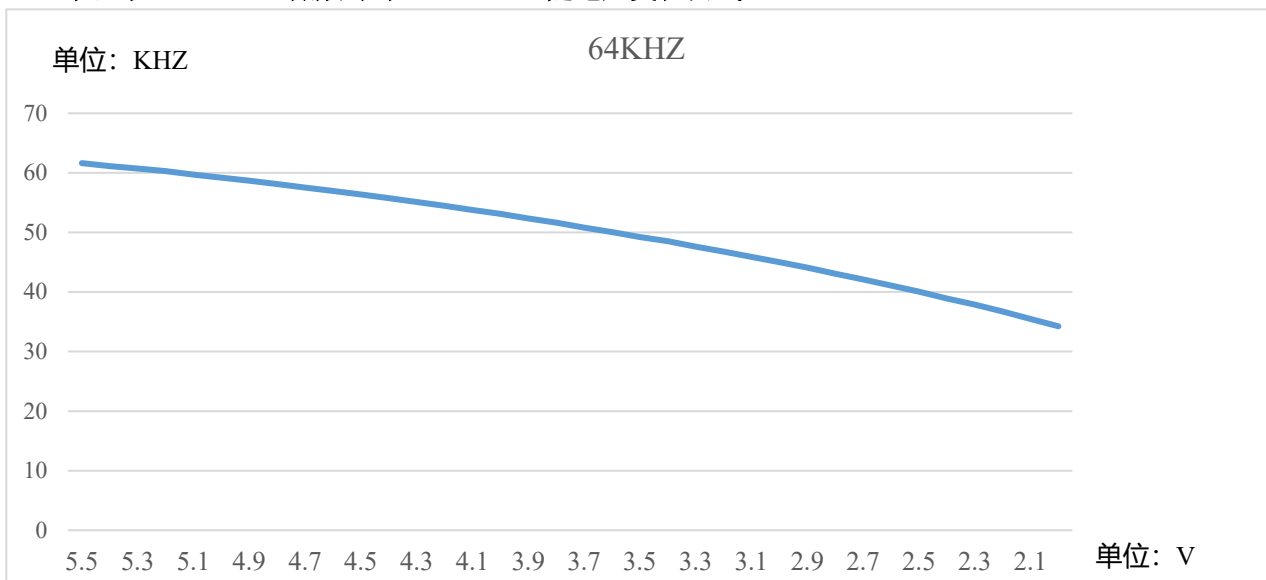


注：具体值不做设计保证。

常温下 (20℃)，高频时钟 (24MHZ) 随电压变化曲线



常温下 (20℃)，低频时钟 (64KHZ) 随电压变化曲线



18.5 Flash 特性

参数	符号	温度 (°C)	最小值	典型值	最大值	单位
编程电压	$V_F^{(1)}$	$T_A = 25$	-	2.4	-	V
编程擦写次数	N_{ENDUR}	$T_J = -40 \sim 85$	100,000	-	-	Cycle
数据保存时间	T_{RET}	$T_A = 25$	10	-	-	Year
页擦除时间	T_{PERASE}	$T_A = 25$	-	3	-	ms
页编程时间	T_{PROG}	$T_A = 25$	-	2	-	ms

注：除此之外，必需保证在最低工作电压之上。

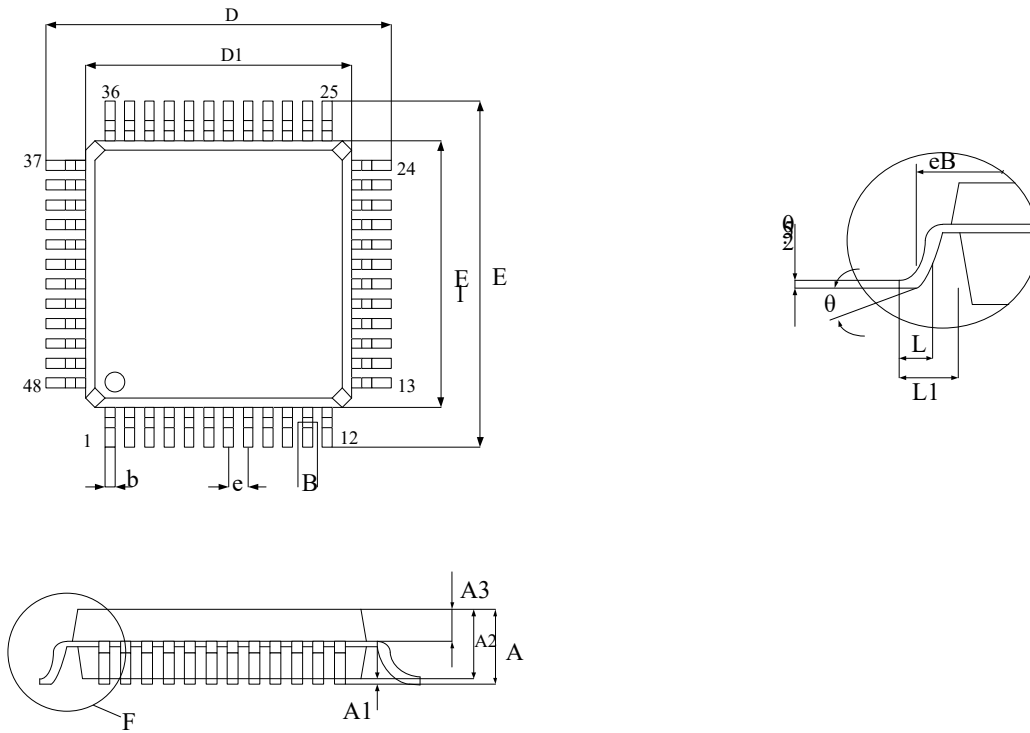
18.6 ADC 电气特性

符号	参数	测试条件	最小值	典型值	最大值	单位
		条件 (常温 25℃)				
V _{ADC}	ADC 工作电压	-	2.5	-	5.5	V
I _{ADC}	ADC 工作电流	VDD=5V	-	200	-	μA
V _{AIN}	ADC 输入电压	-	GND	-	V _{REF}	V
V _{iREF1}	内部参考电压 1	内部 2V 参考电压, VDD ≥ 2V+0.5V	-2%	2	+2%	V
V _{iREF2}	内部参考电压 2	内部 3V 参考电压, VDD ≥ 2V+0.5V	-2%	3	+2%	V
V _{iREF3}	内部参考电压 3	内部 4V 参考电压, VDD ≥ 2V+0.5V	-2%	4	+2%	V
T _{VREF}	参考稳定时间	VDD=5V, 参考电压选择和切换后	-	50	-	μs
F _{ADC}	ADC 时钟	V _{REF} = VDD =5V	-	-	2	MHZ
T _{CON}	ADC 转换时间	-	-	16	-	T _{ADC}
DNL	微分非线性误差	V _{REF} = VDD =5V, F _{ADC} =1MHZ	-2	-	+2	LSB
INL	积分非线性误差	V _{REF} = VDD =5V, F _{ADC} =1MHZ	-3	-	+3	LSB
E _Z	偏移误差	V _{REF} = VDD =5V, F _{ADC} =1MHZ	-5	-	+5	LSB
E _F	满刻度误差	V _{REF} = VDD =5V, F _{ADC} =1MHZ	-10	-	+10	LSB

注：具体值不做设计保证。

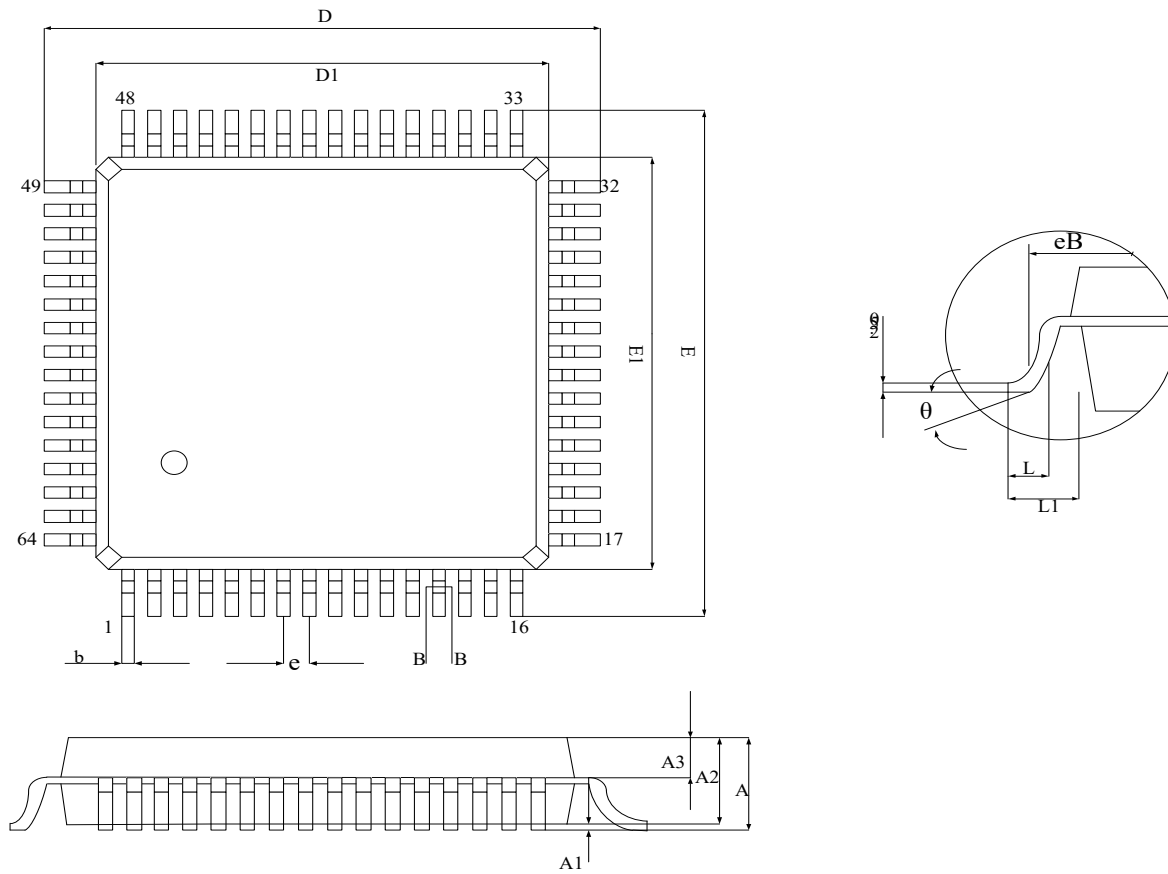
19 封装信息

19.1 LQFP48



符号	单位 (mm)		
	最小	正常	最大
A	1.500	---	1.600
A1	0.050	---	0.150
A2	1.350	1.400	1.450
A3	0.600	---	0.640
b	0.180	---	0.260
D	8.800	9.000	9.200
D1	6.900	7.000	7.100
e	0.500 (BSC)		
E	8.800	9.000	9.200
E1	6.900	7.000	7.100
L	0.400	---	0.650
L1	1.0 (REF)		
eB	8.100	---	8.250

19.2 LQFP64



符号	单位 (mm)		
	最小	正常	最大
A	1.500	---	1.600
A1	0.050	---	0.150
A2	1.3500	1.40	1.450
A3	0.600	---	0.640
b	0.160	---	0.240
D	8.800	9.000	9.200
D1	6.900	7.000	7.100
e	0.400 (BSC)		
E	8.800	9.000	9.200
E1	6.900	7.000	7.100
L	0.400	---	0.650
L1	1.0 (REF)		
eB	8.100	---	8.250

20 指令集简述

20.1 概述

M9指令集是一种精简指令集（RISC），指令宽度为16位，由操作码和0~2个操作数组成。指令按照功能可分为5类：字节操作类指令、位操作类指令、控制操作类指令、立即数操作类指令和存储器操作类指令。

一个指令周期由1个系统时钟周期组成，除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特征情况，指令执行需要两个指令周期。

任何一条指定文件寄存器，作为指令的一部分都会进行读-修改-写操作。读寄存器、修改数据并根据指令或目标寄存器标识符“d”存储结果。即使是写寄存器的指令也将先对改寄存器进行读操作。

20.2 符号说明

符号	范围	说明	符号	范围	说明
R/r	000h-fffh	8 位文件寄存器地址	PC	-	PC 指针
Rs	000h-fffh	12 位文件寄存器源地址	C	-	进位标志
Rd	000h-fffh	12 位文件寄存器 目标地址	DC	-	半进位标志
A	-	AREG 寄存器	DC	-	半进位标志
a		快速操作 RAM 位	Z	-	零标志
BSR		存储选择寄存器	OV	-	溢出标志
b	0-7	位地址	N	-	负标志
K/k		立即数、常数或标号	d	0-1	目的操作数定义
TOS	-	栈顶	TO	-	超时位
Label	-	标号名称	PD	-	掉电位
s	-	快速调用选择位	GIE	-	总中断使能位

20.3 M9 指令集表

指令集表中：d=1，目的操作数为 R；d=0，目的操作数为 A。

指令类型	助记符	指令说明	周期数	影响标志位	备注
字节操作类指令	ADDAR R, d, a	A 与 R 相加	1	C, DC, Z, OV, N	1, 2
	ADCAR R, d, a	A 带进位与 R 相加	1	C, DC, Z, OV, N	1, 2
	ANDAR R, d, a	A 和 R 作逻辑与运算	1	Z, N	1, 2
	CLRR R, a	将 R 清零	1	Z	2
	COMR R, d, a	对 R 取反	1	Z, N	1, 2
	JERA R, a	R 与 A 比较，相等跳过	1 (2 或 3)	-	3
	JGRA R, a	R 与 A 比较，大于跳过	1 (2 或 3)	-	3
	JLRA R, a	R 与 A 比较，小于跳过	1 (2 或 3)	-	1
	DECR R, d, a	R 递减 1	1	C, DC, Z, OV, N	1, 2, 3
	DJZR R, d, a	R 递减 1，为 0 跳过	1 (2 或 3)	-	1, 2, 3
	DJNZR R, d, a	R 递减 1，非 0 跳过	1 (2 或 3)	-	1
	INCR R, d, a	R 递增 1	1	C, DC, Z, OV, N	1, 2, 3
	JZR R, d, a	R 递增 1，为 0 跳过	1 (2 或 3)	-	3
	JNZR R, d, a	R 递增 1，非 0 跳过	1 (2 或 3)	-	1
	ORAR R, d, a	A 与 R 作逻辑或运算	1	Z, N	1
	MOVR R, d, a	传送 R	1	Z, N	1
	MOVRR Rs, Rd	从源 Rs 送到目标 Rd	2	-	
	MOVAR R, a	将 A 中的内容送入 R	1	-	
	MULAR R, a	A 与 R 相乘	1	-	1
	NEGR R, a	对 R 取补	1	C, DC, Z, OV, N	
	RLR R, d, a	R 带进位循环左移	1	C, Z, N	1
	RLNCR R, d, a	R 循环左移（无进位）	1	Z, N	
	RRR R, d, a	R 带进位循环右移	1	C, Z, N	
	RRNCR R, d, a	R 循环右移（无进位）	1	Z, N	
	SETR R, a	将 R 全置为 1	1	-	1
	SBCAR R, d, a	A 减去 R（带借位）	1	C, DC, Z, OV, N	
	SUBRA R, d, a	R 减去 A	1	C, DC, Z, OV, N	1
	SBCRA R, d, a	R 减去 A（带借位）	1	C, DC, Z, OV, N	
	SWAPR R, d, a	对 R 进行半字节交换	1	-	3
	JREZ R, a	测试 R，为 0 则跳过	1 (2 或 3)	-	1
	XORAR R, d, a	A 和 R 作逻辑异或运算	1	Z, N	
位操作指令	BCLR R, b, a	将 R 中的指定位清零	1	-	1
	BSET R, b, a	将 R 中的指定位置 1	1	-	1
	JBTS0 R, b, a	测试 R 的位，为 0 跳过	1 (2 或 3)	-	2, 3
	JBTS1 R, b, a	测试 R 的位，为 1 跳过	1 (2 或 3)	-	2, 3
	BNEG R, b, a	将 R 中的指定位取反	1	-	1

指令类型	助记符	指令说明	周期数	影响标志位	备注
控制操作类指令	RJBC Label	如果有进位则跳转	1 (2)	-	
	RJBN Label	如果为负则跳转	1 (2)	-	
	RJBNC Label	如果没有进位则跳转	1 (2)	-	
	RJBNN Label	如果不为负则跳转	1 (2)	-	
	RJBNOV Label	如果未溢出则跳转	1 (2)	-	
	RJBNZ Label	如果不为零则跳转	1 (2)	-	
	RJBOV Label	如果溢出则跳转	1 (2)	-	
	RGOTO Label	无条件跳转	2	-	
	RJBZ Label	如果为零则跳转	1 (2)	-	
	CALL k, s	调用子程序	2	-	
	DAA	对 A 进行十进制调整	1	C	
	GOTO k	跳转到地址	2	-	
	NOP	无操作	1	-	
	NOP	无操作	1	-	3
	SPOP	弹出堆栈栈顶 TOS	1	-	
	SPUSH	压入堆栈栈顶 TOS	1	-	
	RCALL Label	相对调用	2	-	
	SRESET	软件器件复位	1	所有	
	RETIE s	中断返回并允许中断	2	GIE/GIEH	
	RETIA k	返回并将 k 送入 A	2	-	
	RETURN s	从子程序返回	2	-	
立即数操作指令	ADDIA k	A 与立即数 k 相加	1	C, DC, Z, OV, N	
	ANDIA k	立即数 k 和 A 作逻辑与运算	1	Z, N	
	ORIA k	立即数 k 和 A 作逻辑或运算	1	Z, N	
	LDFSR R,k	传送立即数 k 到 FSR	2	-	
	BANKBSR k	将立即数 k 送入 BSR	1	-	
	MOVIA k	将立即数 k 送入 A	1	-	
	MULIA k	立即数 k 与 A 相乘	1	-	
	SUBIA k	立即数 k 减去 A	1	C, DC, Z, OV, N	
	XORIA k	立即数 k 与 A 作逻辑异或运算	1	Z, N	
存储器操作指令	RDT*	表读	2	-	
	RDT*+	表读, 后递增		-	
	RDT*-	表读, 后递减		-	
	RDT+*	表读, 预递增		-	
	WDT*	表写	2	-	

注:

- (1) 当端口寄存器修改自身时, 修改时使用的值是引脚上的当前值。例如, 如果将一引脚配置为输入, 其对应数据锁存器中的值将为 1, 但此时若有外部器件将该引脚驱动为低电平, 则被写回数据锁存器的数据值将是 0。
- (2) 如果程序计数器 (PC) 被修改或者条件测试为真, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- (3) 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息, 否则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元内存储的都是合法的指令。

21 修改记录

版本	日期	描述
V1.01	2021-08	初版
---	---	---
V1.04	2022-01	增加 ADC 电气特性，勘误
V1.05	2022-08	勘误