



**REALMAGIC**  
锐盟半导体

# RM1221A 数据手册

26 路触控按键和 PWM 的增强型 8051SOC

V1.21  
2023-10-07



## 目 录

1	产品简介.....	10
1.1	功能特性.....	10
1.2	系统框图.....	12
2	管脚配置.....	13
2.1	管脚排列图.....	13
2.2	脚位定义.....	18
2.3	数字外设功能引脚全映射模块 PTM.....	20
2.3.1	PTM 模块特性.....	20
2.3.2	PTM 可全映射外设功能口.....	20
3	中央处理器（CPU）.....	22
3.1	概述.....	22
3.2	中央处理器（CPU）相关寄存器.....	22
3.2.1	程序计数器（PC）.....	22
3.2.2	累加器（ACC）.....	22
3.2.3	寄存器（B）.....	22
3.2.4	程序状态字寄存器（PSW）.....	22
3.2.5	堆栈指针 SP.....	24
3.2.6	双数据指针 DPTR.....	24
4	MTP ROM 和 SRAM.....	25
4.1	程序存储器（ROM）.....	25
4.2	随机数据存储器（RAM）.....	26
4.3	特殊功能寄存器(SFR).....	28
4.3.1	直接寻址读写 SFR.....	28
4.3.2	外部扩展 XSFR.....	31
5	时钟系统和电源管理.....	34
5.1	概述.....	34
5.2	寄存器描述.....	35
5.2.1	系统时钟配置寄存器(CKCON)（受 WPT 保护）.....	35
5.2.2	电源管理寄存器(PCON).....	35
5.2.3	TRIM 控制寄存器(TRIMCON).....	36
5.2.4	内部高速时钟 TRIM 低 8 位(RCTRIML).....	37
5.2.5	内部高速时钟 TRIM 高 2 位(RCTRIMH).....	37
6	复位.....	38
6.1	上电复位.....	39
6.2	低电压复位.....	39
6.3	外部引脚复位.....	40
6.4	硬件故障复位.....	40

6.5	看门狗复位.....	40
6.6	软件复位.....	40
6.7	寄存器描述.....	41
6.7.1	IAP 使能寄存器 (CHPCON) (受 WPT 保护) .....	41
6.7.2	辅助功能寄存器 1 (AUXR1) .....	41
6.7.3	时控保护寄存器 (WPT) .....	42
7	中断.....	43
7.1	概述.....	43
7.2	中断源及中断向量.....	43
7.3	中断优先级.....	44
7.4	中断服务.....	47
7.5	中断延迟.....	47
7.6	外部中断.....	48
7.7	寄存器列表.....	50
7.8	寄存器描述.....	50
7.8.1	扩展中断使能寄存器 (EIE) .....	50
7.8.2	扩展中断使能寄存器 (EIE1) .....	51
7.8.3	中断使能寄存器 (IE) .....	52
7.8.4	中断优先级配置寄存器 0 (IPO) .....	52
7.8.5	中断优先级配置寄存器 1 (IP1) .....	53
8	低电压检测.....	54
8.1	寄存器描述.....	54
8.1.1	LVD 配置寄存器(LVDCON) (受 WPT 保护) .....	54
9	GPIO .....	56
9.1	概述.....	56
9.2	寄存器列表.....	56
9.3	寄存器描述.....	57
9.3.1	PA/PB/PC/PD 端口数据锁存寄存器 (PORTA / PORTB / PORTC / PORTD) .....	57
9.3.2	PA/PB/PC/PD 端口控制寄存器 (PAM/PBM/PCM/PDM) .....	58
9.3.3	PA/PB/PC/PD 上拉控制寄存器 (PAUR/PBUR/PCUR/PDUR) .....	58
9.3.4	PA/PB/PC/PD 下拉控制寄存器 (PADR/PBDR/PCDR/PDDR) .....	58
9.3.5	PA/PB/PC/PD 开漏功能控制寄存器 (PAOD /PLVD /PCOD /PDOD) .....	59
9.3.6	驱动电流配置寄存器 (PAIO/ PBIO/ PCIO/ PDIO) .....	59
9.3.7	PA/PB/PC/PD 翻转中断使能寄存器 (PAINTR/ PBINTR/ PCINTR/ PDINTR) .....	60
9.3.8	PA/PB/PC/PD 翻转中断标志寄存器 (PAINTRF/ PBINTRF/ PCINTRF/ PDINTRF) .....	61
9.4	数字外设功能引脚全映射控制 .....	61
9.4.1	外设功能引脚全映射控制寄存器.....	61
10	定时/计数器 0 和 1 .....	64
10.1	概述.....	64
10.2	功能描述.....	64
10.2.1	模式 0 (13 位定时器) .....	64
10.2.2	模式 1 (16 位定时器) .....	65

10.2.3	模式 2 (8 位自动重装载定时器) .....	65
10.2.4	模式 3 (两组独立 8 位定时器) .....	66
10.3	寄存器列表 .....	67
10.4	寄存器描述 .....	67
10.4.1	定时器 0 和 1 控制寄存器 (TCON) .....	67
10.4.2	定时器 0 和 1 模式寄存器 (TMOD) .....	68
10.4.3	定时器 0 低 8 位数据寄存器 (TL0) .....	69
10.4.4	定时器 1 低 8 位数据寄存器 (TL1) .....	69
10.4.5	定时器 0 高 8 位数据寄存器 (TH0) .....	69
10.4.6	定时器 1 高 8 位数据寄存器 (TH1) .....	69
10.4.7	辅助功能寄存器 1 (AUXR1) .....	70
<b>11</b>	<b>定时器 2 .....</b>	<b>71</b>
11.1	概述 .....	71
11.2	功能描述 .....	71
11.2.1	自动重装载功能模式 .....	71
11.2.2	比较功能模式 .....	72
11.2.3	输入捕获功能模块 .....	73
11.3	寄存器列表 .....	74
11.4	寄存器描述 .....	74
11.4.1	定时器 2 控制寄存器 (T2CON) .....	74
11.4.2	定时器 2 模式选择 (T2MOD) .....	75
11.4.3	定时器 2 重装载/比较数据低字节 (RCMP2L) .....	76
11.4.4	定时器 2 重装载/比较数据高字节 (RCMP2H) .....	76
11.4.5	定时器 2 低字节数据 (TL2) .....	76
11.4.6	定时器 2 高字节数据 (TH2) .....	77
11.4.7	输入捕获控制寄存器 0 (CAPCON0) .....	77
11.4.8	输入捕获控制寄存器 1 (CAPCON1) .....	78
11.4.9	输入捕获控制寄存器 2 (CAPCON2) .....	78
11.4.10	捕获通道 0 低字节数据 (T2C0L) .....	79
11.4.11	捕获通道 0 高字节数据 (T2C0H) .....	79
11.4.12	捕获通道 1 低字节数据 (T2C1L) .....	79
11.4.13	捕获通道 1 高字节数据 (T2C1H) .....	80
11.4.14	捕获通道 2 低字节数据 (T2C2L) .....	80
11.4.15	捕获通道 2 高字节数据 (T2C2H) .....	80
<b>12</b>	<b>定时器 3 .....</b>	<b>81</b>
12.1	概述 .....	81
12.2	寄存器列表 .....	81
12.3	寄存器描述 .....	82
12.3.1	定时器 3 控制寄存器 (T3CON) .....	82
12.3.2	定时器 3 重载值低字节数据 (T3RL) .....	82
12.3.3	定时器 3 重载值高字节数据 (T3RH) .....	83
<b>13</b>	<b>定时器 4/PWM .....</b>	<b>84</b>

13.1	概述.....	84
13.2	寄存器列表.....	84
13.3	寄存器描述.....	85
13.3.1	定时器 4 控制寄存器 (T4CON) .....	85
13.3.2	定时器 4 定时溢出低字节寄存器 (T4OVRL) .....	85
13.3.3	定时器 4 定时溢出高字节寄存器 (T4OVRH) .....	86
13.3.4	定时器 4 PWM0 占空比低字节寄存器 (T4D0L) .....	86
13.3.5	定时器 4 PWM0 占空比高字节寄存器 (T4D0H) .....	86
13.3.6	定时器 4 PWM1 占空比低字节 (T4D1L) .....	87
13.3.7	定时器 4 PWM1 占空比高字节 (T4D1H) .....	87
14	脉冲宽度调制 (PWM) .....	88
14.1	概述.....	88
14.2	功能描述.....	88
14.2.1	脉冲宽度调制发生器.....	88
14.2.2	PWM 类型.....	91
14.2.3	工作模式 .....	92
14.2.4	故障刹车 .....	94
14.2.5	极性控制 .....	94
14.2.6	PWM 中断.....	94
14.3	寄存器列表.....	95
14.4	寄存器描述.....	96
14.4.1	PWM 控制寄存器 0 (PWMCON0) .....	96
14.4.2	PWM 控制寄存器 1 ( PWMCON1) .....	97
14.4.3	PWM 周期低字节寄存器 (PWMOVRL) .....	98
14.4.4	PWM 周期高字节寄存器 (PWMOVRH) .....	98
14.4.5	占空比低字节寄存器 (PWMD0L /PWMD1L /PWMD2L /PWMD3L /PWMD4L /PWMD5L) .....	98
14.4.6	占空比高字节寄存器 (PWMD0H /PWMD1H /PWMD2H /PWMD3H /PWMD4H /PWMD5H) .....	99
14.4.7	PWM 死区控制寄存器 (PDTEN) .....	99
14.4.8	PWM 死区时间计数器 (PDTCNT) .....	100
14.4.9	PWM 输出掩码控制寄存器 (PMEN) .....	100
14.4.10	PWM 掩码数据寄存器 (PMD) .....	101
14.4.11	PWM 故障刹车数据寄存器 (FBD) .....	101
14.4.12	PWM 负极性寄存器 (PNP) .....	102
14.4.13	PWM 中断控制寄存器 (PWMINTC) .....	102
15	看门狗定时器 (WDT) .....	104
15.1	概述.....	104
15.2	功能描述.....	104
15.3	寄存器描述.....	104
15.3.1	WDT 配置寄存器(WDTCON).....	104
15.3.2	WDT 定时溢出寄存器(WDTR) .....	105

16	自唤醒定时器 (WKT) .....	106
16.1	寄存器列表.....	106
16.2	寄存器描述.....	106
16.2.1	WKT 控制寄存器 (WKCON) .....	106
16.2.2	WKT 重载数据寄存器 (RWK) .....	107
17	通用异步收发器 (UART0/1) .....	108
17.1	概述.....	108
17.2	功能描述.....	108
17.2.1	模式 1 .....	108
17.2.2	模式 2 .....	109
17.2.3	波特率 .....	110
17.3	寄存器列表.....	110
17.4	寄存器描述.....	110
17.4.1	串口 0 控制寄存器 (S0CON) .....	110
17.4.2	串口 0 数据缓存寄存器 (S0BUF) .....	111
17.4.3	串口 0 波特率低字节 (S0OVL) .....	111
17.4.4	串口 0 波特率高字节 (S0OVRH) .....	112
17.4.5	串口 1 控制寄存器 (S1CON) .....	112
17.4.6	串口 1 数据缓存寄存器 (S1BUF) .....	113
17.4.7	串口 1 波特率低字节 (S1OVL) .....	113
17.4.8	串口 1 波特率高字节 (S1OVRH) .....	113
18	串行外设接口 (SPI) .....	114
18.1	概述.....	114
18.2	功能描述.....	114
18.2.1	主机模式 .....	116
18.2.2	从机模式 .....	116
18.2.3	时钟格式和数据传输.....	116
18.2.4	从机选择引脚 NSS 配置 .....	118
18.2.5	写冲突错误 .....	119
18.2.6	SPI 中断.....	119
18.3	寄存器列表.....	119
18.4	寄存器描述.....	120
18.4.1	SPI 控制寄存器 (SPCR) .....	120
18.4.2	SPI 数据寄存器 (SPDR) .....	120
18.4.3	SPI 状态寄存器 (SPSR) .....	121
19	IIC 总线 .....	122
19.1	概述.....	122
19.2	功能描述.....	122
19.2.1	主发送器模式.....	123
19.2.2	主接收器模式.....	124
19.2.3	从发送器模式.....	125
19.2.4	从接收器模式.....	126

19.3	寄存器列表.....	128
19.4	寄存器描述.....	128
19.4.1	IIC 控制寄存器 (I2CCON) .....	128
19.4.2	IIC 地址寄存器 (I2CADDR) .....	129
19.4.3	IIC 数据寄存器 (I2CDAT) .....	129
19.4.4	IIC 时钟寄存器 (I2CCLK) .....	129
19.4.5	IIC 状态寄存器 (I2CSTAT) .....	129
20	<b>模数转换 ADC</b> .....	<b>131</b>
20.1	概述.....	131
20.2	功能描述.....	132
20.2.1	ADC 工作方式 .....	132
20.2.2	外部触发 ADC .....	133
20.2.3	ADC 转换结果比较器 .....	134
20.2.4	ADC 参考带隙电压应用 .....	134
20.3	寄存器列表.....	134
20.4	寄存器描述.....	135
20.4.1	ADC 控制寄存器 0 (ADCCON0) .....	135
20.4.2	ADC 控制寄存器 1 (ADCCON1) .....	135
20.4.3	ADC 控制寄存器 2 (ADCCON2) .....	137
20.4.4	ADC 控制寄存器 3 (ADCCON3) .....	137
20.4.5	ADC 控制寄存器 4 (ADCCON4) .....	138
20.4.6	ADC 控制寄存器 5 (ADCCON5) .....	139
20.4.7	ADC 控制寄存器 6 (ADCCON6) .....	140
20.4.8	ADC 转换结果低位寄存器 (ADCRL) .....	140
20.4.9	ADC 转换结果高位寄存器 (ADCRH) .....	141
20.4.10	ADC 比较值低位寄存器 (ADC MPL) .....	141
20.4.11	ADC 比较值高位寄存器 (ADC MPH) .....	141
20.4.12	ADC 比较值高位寄存器 (ADC DLY) .....	142
20.4.13	ADC 模拟输入通道 (PA) 功能寄存器 (PACHS) .....	142
20.4.14	ADC 模拟输入通道 (PB) 功能寄存器 (PBCHS) .....	142
20.4.15	ADC 模拟输入通道 (PC) 功能寄存器 (PCCHS) .....	143
20.4.16	ADC 模拟输入通道 (PD) 功能寄存器 (PDCHS) .....	143
20.5	ADC 应用注意事项 .....	143
21	<b>运算放大器 (OPA0)</b> .....	<b>145</b>
21.1	概述.....	145
21.2	寄存器列表.....	145
21.3	寄存器描述.....	145
21.3.1	运放控制寄存器 0(OPACON0).....	145
21.3.2	运放控制寄存器 1(OPACON1).....	146
21.3.1	运放控制寄存器 3 (LDO).....	146
21.4	运算放大器的使用.....	147
21.4.1	OPA0 使能 .....	147
21.4.2	OPA0 端口选择.....	147

21.4.3	OPA0 工作模式.....	148
22	触控按键(TOUCH KEY).....	149
23	LCD.....	150
23.1	概述.....	150
23.2	寄存器列表.....	150
23.3	寄存器描述.....	151
23.3.1	LCD 模块控制寄存器 (LCDCON) .....	151
23.3.2	PA 口输出选择寄存器 (PAVO) .....	152
23.3.3	PB 口输出选择寄存器 (PBVO) .....	152
23.3.4	PC 口输出选择寄存器 (PCVO) .....	152
23.3.5	PD 口输出选择寄存器 (PDVO) .....	153
23.4	LCD 使用说明.....	153
24	乘除法器.....	157
24.1	概述.....	157
24.2	寄存器列表.....	157
24.3	寄存器描述.....	157
24.3.1	乘除法运行控制寄存器 (MDCON) .....	157
24.3.2	扩展累加器 (EXAx) .....	158
24.3.3	扩展 B 寄存器 (EXBx) .....	158
25	IAP、EEPROM、BootLoader 及安全加密.....	159
25.1	概述.....	159
25.2	寄存器列表.....	159
25.3	寄存器描述.....	159
25.3.1	IAP 执行寄存器 (IAPTRG) (受 WPT 保护) .....	159
25.3.2	IAP 地址低字节 (IAPAL) .....	160
25.3.3	IAP 地址高字节 (IAPAH) .....	160
25.3.4	IAP 数据寄存器 (IAPFD) .....	161
25.3.5	IAP 控制寄存器 (IAPCON) .....	161
25.3.6	IAP 使能寄存器 (CHPCON) (受 WPT 保护) .....	162
25.4	BootLoader.....	162
25.5	安全加密.....	165
26	指令集.....	166
27	Code Option 代码选项.....	170
27.1	OPTION0 (0x1E0) .....	170
27.2	OPTION1 (0x1E1) .....	170
27.3	OPTION2 (0x1E2) .....	171
27.4	OPTION3 (0x1E3) .....	171
27.5	OPTION5 (0x1E5) .....	172
27.6	OPTION19 (0x1F8) .....	172
27.7	OPTION22 (0x1FD) .....	172
27.8	OPTION23 (0x1FE) .....	173



28	电气特性.....	174
28.1	最大绝对额定值.....	174
28.2	DC 电气特性.....	174
28.3	内部 16 MHz 高速时钟(HIRC).....	175
28.4	内部 128 KHz 低速时钟(LIRC).....	175
28.5	低压复位规格.....	175
28.6	欠压检测规格.....	176
28.7	ADC 电气特性.....	176
28.8	OPA 电气特性.....	177
28.9	交流电气特性.....	177
29	封装信息.....	178
29.1	SOP28L 封装示意图.....	178
29.2	TSSOP28L 封装示意图.....	179
29.3	SOP20L 封装示意图.....	180
29.4	SOP16L 封装示意图.....	181
29.5	QFN4*4-20L 封装示意图.....	182
29.6	QFN4*4-28L 封装示意图.....	183
30	修订记录.....	184
31	声明.....	185

## 1 产品简介

### 1.1 功能特性

- ◆ 基于 8051 指令的高速 1T 增强型 MTP SOC
- ◆ ROM
  - ROM 空间: 16K\*8 位 MTP, 可重复写入 10K 次, 105°C 下 10 年保存寿命
  - EEPROM 空间: 512 BYTE, 可重复写入 20K 次, 105°C 下 10 年保存寿命
  - BootLoader: 支持 1K BYTE LDR0M, Code Option 选择从 APROM 或 LDR0M 启动
  - 支持双线仿真、IAP 和 ICP 操作
  - 灵活的代码保护模式, 硬件加密保护
- ◆ RAM
  - 256 BYTE IRAM
  - 768 BYTE XRAM
- ◆ 工作条件
  - 2.3~5.5V@-40~85°C 应用环境
  - IC 系统时钟 (fsys) 对应的工作电压范围
    - >16MHz@2.9~5.5V
    - ≤8MHz@2.3~5.5V
- ◆ 时钟系统
  - 内建高频 16MHz 振荡器 (fHRC)
    - 作为系统时钟源时, fsys 可通过编程器选择设定为 16M/ 8M/ 4MHz/ 1M
    - 频率误差: 跨越 (2.5V~5.5V) 及 (-40~85°C) 应用环境, 不超过 ±4%
  - 内置高频晶体振荡器电路及低频 32.768KHz 晶振电路
    - 可外接 2~16MHz 晶振或 32.768KHz 晶振
    - 作为系统时钟源时, fsys 可选择使用外接晶振 1/2/4/16 这四种分频中的一种
  - 内建低频 128K/32KHz LRC 振荡器
    - 可作为看门狗定时器 (WDT) 和自唤醒定时器 (WKT) 的时钟源
    - 频率误差: 跨越 (4.0V~5.5V) 及 (-40~85°C) 应用环境, 频率误差不超过 ±15%
- ◆ 低电压复位 (LVR)
  - 复位电压有 (2.3~4.3V), 共 7 档可选
  - 缺省值为用户烧写 Code Option 所选值
- ◆ 16 档电平可选的低电压检测模块 (LVD)
- ◆ 中断源 (INT)
  - Timer0/1/2/3/4, WKT, ADC, PWM, UART, SPI, IIC, TK, LVD 共 27 个中断源
  - 外部中断, 所有 IO 口可作为外部中断共 26 个中断口, 可设下降沿, 低电平中断, 所有 PA、PB、PC 和 PD 都分别共用一个共 4 个入口地址
  - 四级中断优先级可设

**◆ GPIO**

- 最大 26 个双向可独立控制的 I/O 口，可独立设定上下拉电阻，可设置为开漏输出
- IO 口源驱动能力分四级控制
- 全部 IO 具有大灌电流驱动能力（80mA）
- 所有 IO 可作为 1/2 BIAS、1/3BIAS 软件 LCD 驱动
- 数字外设功能引脚全映射模块（PTM）

**◆ 数字外设**

- 8 位 WDT 和 WKT
- 2 个兼容 80C51 定时器 Timer0、Timer1（标准 51）
- 16 位 Timer2 带有 3 路输入捕获功能，输入管脚可任意配置
- 16 位 Timer3 可工作在普通模式和掉电模式
- 16 位 Timer4/PWM
- PWM
  - 1 组 16 位 3 对 6 通道，带不同工作模式和刹车故障功能的 PWM，时钟高达 16MHz
  - 1 组 16 位 1 对 2 通道 16 位，复用 TIMER4 时基的 PWM
- 2 个独立 UART 通信口
- 1 路 IIC
- 1 路 SPI
- 集成 16 X 16 位硬件乘除法器

**◆ 26 通道高灵敏度高可靠性触控电路**

- 高灵敏度可适应隔空按键触控、接近感应等对灵敏度要求较高的触控应用
- 高可靠性具有很强的抗干扰性，可通过 10V CS 测试
- 可实现 26 路触控按键及衍生功能如滚轮、滑条等多种应用
- 高灵活度开发软件库支持，低开发难度
- 自动化调试软件支持，智能化开发
- 低功耗触控模式，单触控按键唤醒时芯片整体功耗可低至 5uA

**◆ 12 位 SAR ADC**

- 支持最多 26+6 通道 AD 输入，包括 VDD/4 电压、IO/4 等
- ADC 的参考电压源可选 VDD、内部基准（1V、2V、3V、4V）、外部参考
- 支持硬件启动和转换结果比较
- 可设 ADC 转换完成中断

**◆ 运算放大器 0**

- 支持全引脚外接，手动校准，
- 支持内部 16 倍、64 倍放大，输出可选内部滤波后引至 ADC

**◆ 省电模式**

- 空闲（IDLE）模式：任何中断可唤醒
- 掉电（PD）模式：外部中断、WDT、WKT、TIMER3 和 TK 可唤醒

**◆ 高可靠性**

- ESD 优于 ±8KV、EFT 优于 ±4.5KV

**◆ 开发工具**

- 基于 KEIL 或 RM 集成开发环境（IDE），片上调试（OCD）

**◆ 封装**

- SOP28/TSSOP28、SOP20/TSSOP20、SOP16、QFN28、QFN20

## 1.2 系统框图

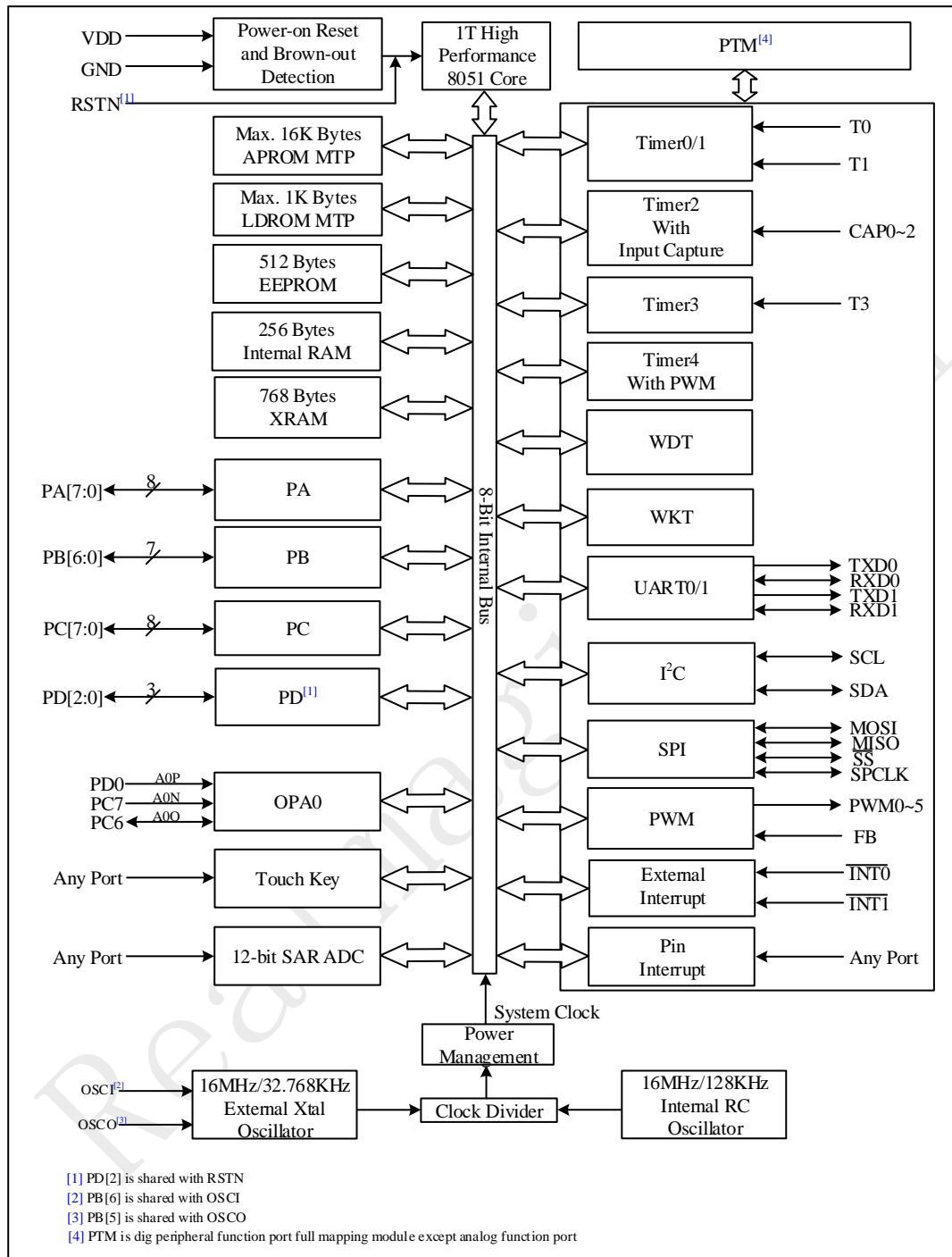
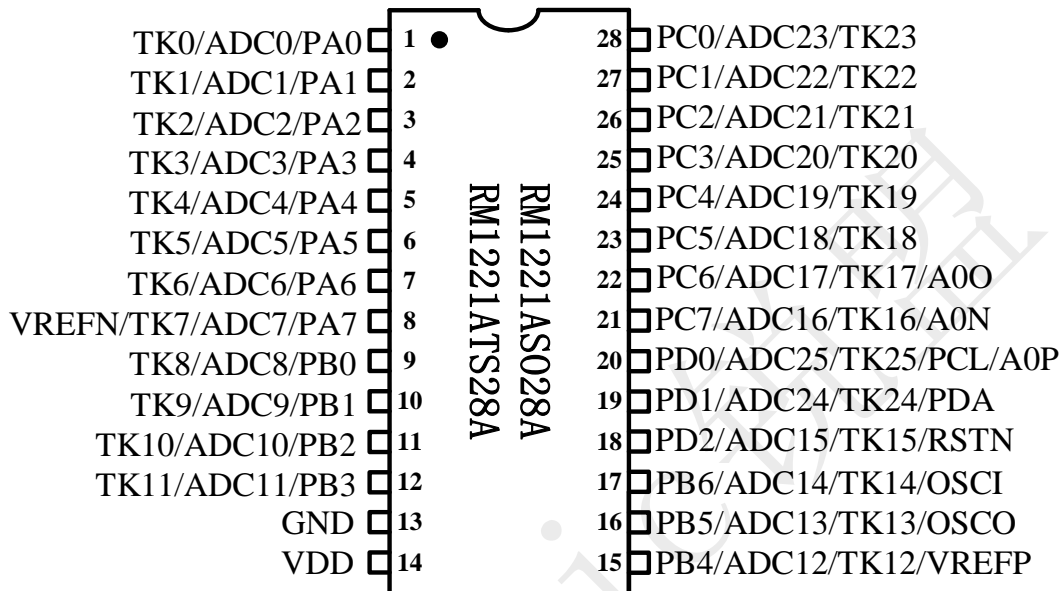


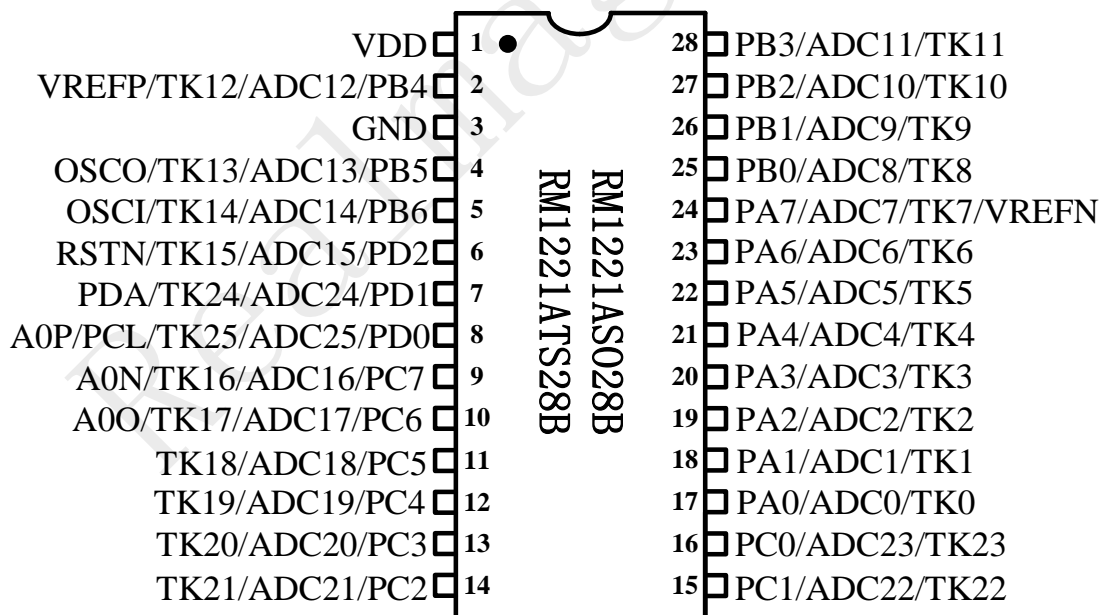
图 1-2-1 系统框图

## 2 管脚配置

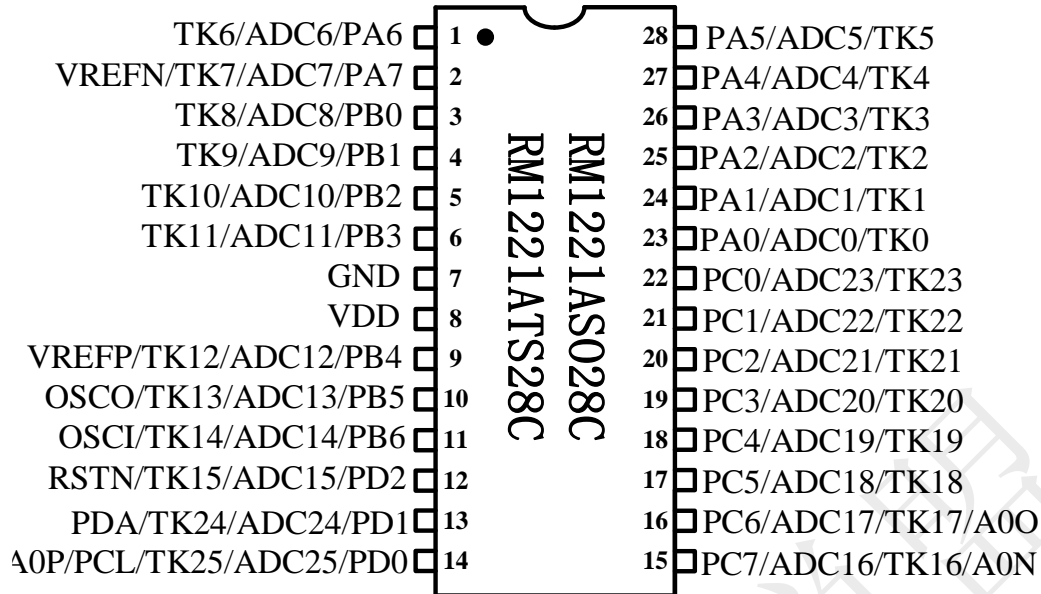
### 2.1 管脚排列图



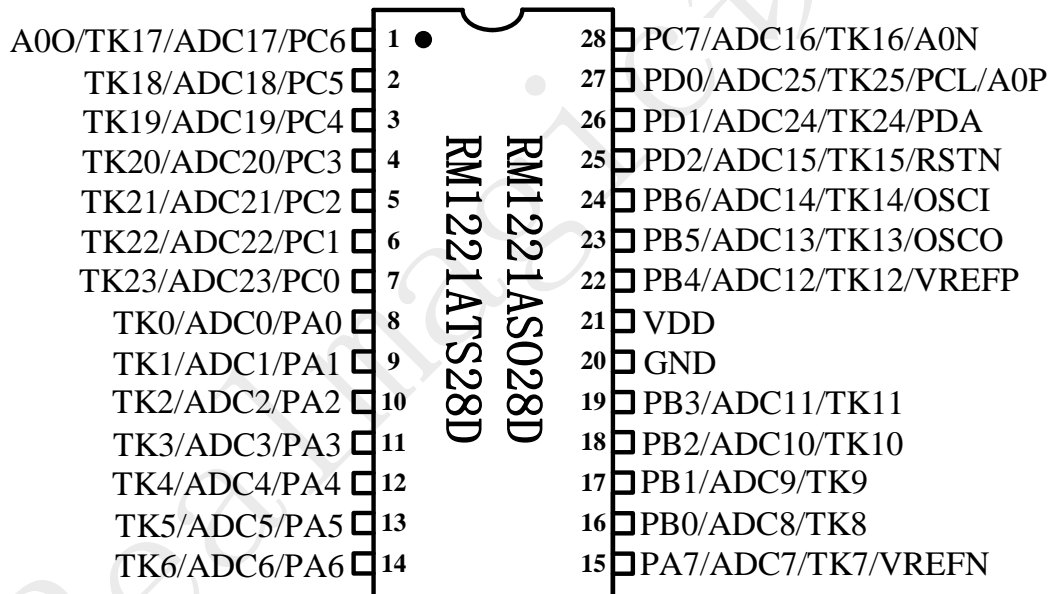
RM1221AS028A/RM1221ATS28A (SOP28/TSSOP28)



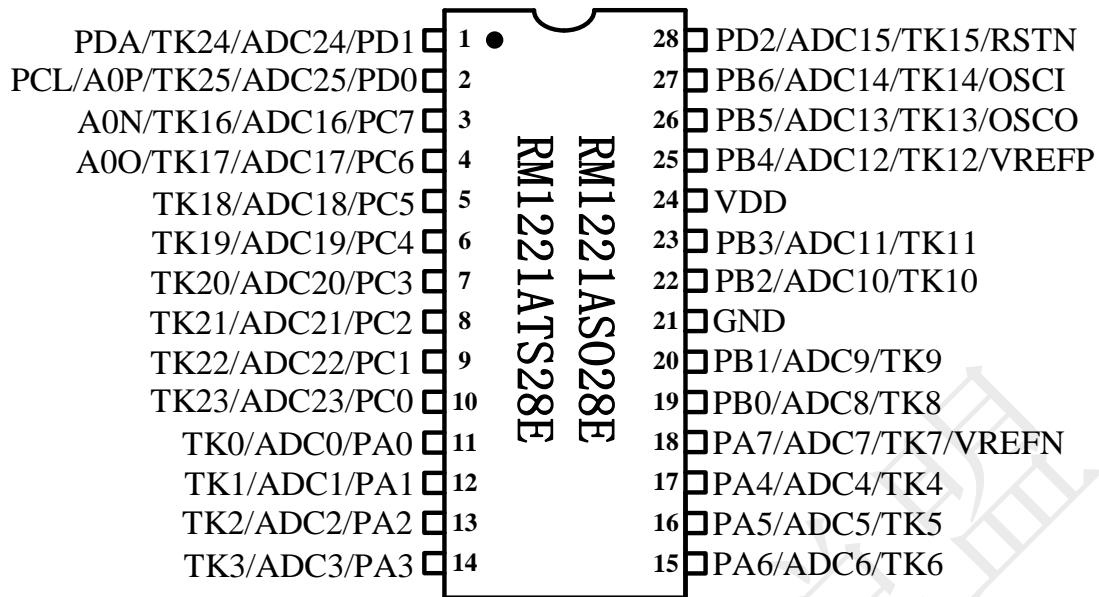
RM1221AS028B/RM1221ATS28B (SOP28/TSSOP28)



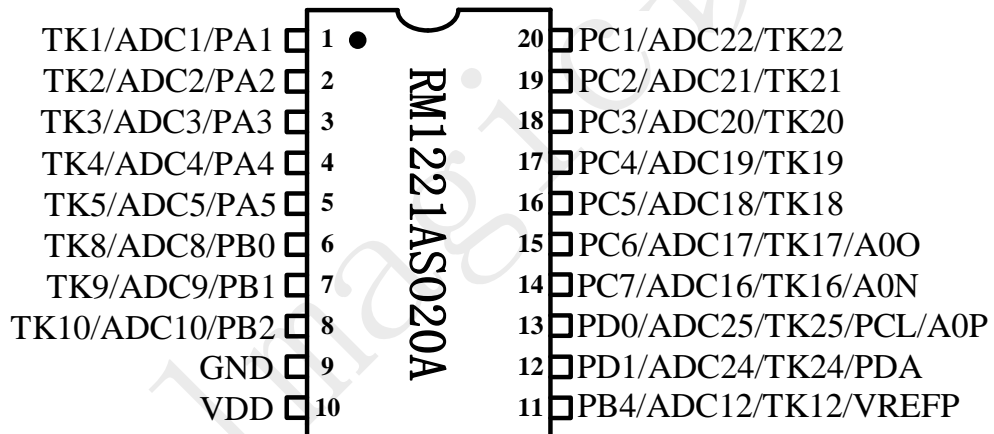
RM1221ASO28C/RM1221ATS28C (SOP28/TSSOP28)



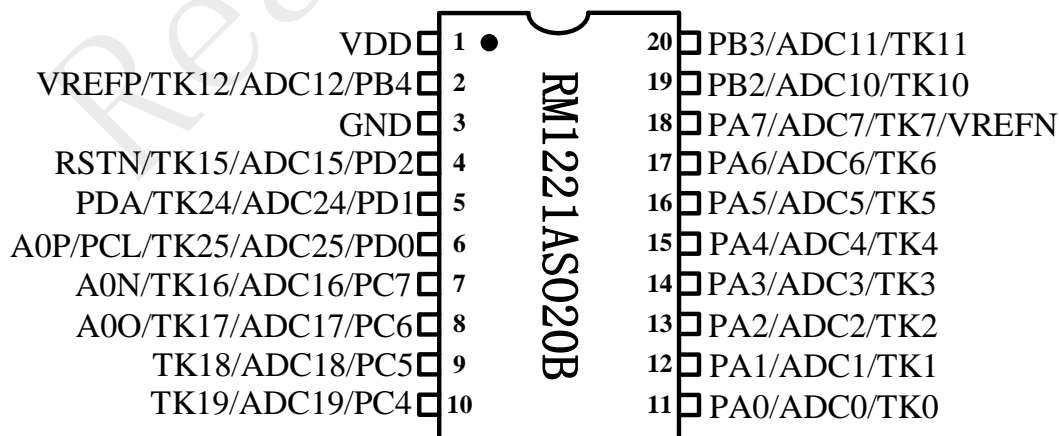
RM1221ASO28D/RM1221ATS28D (SOP28/TSSOP28)



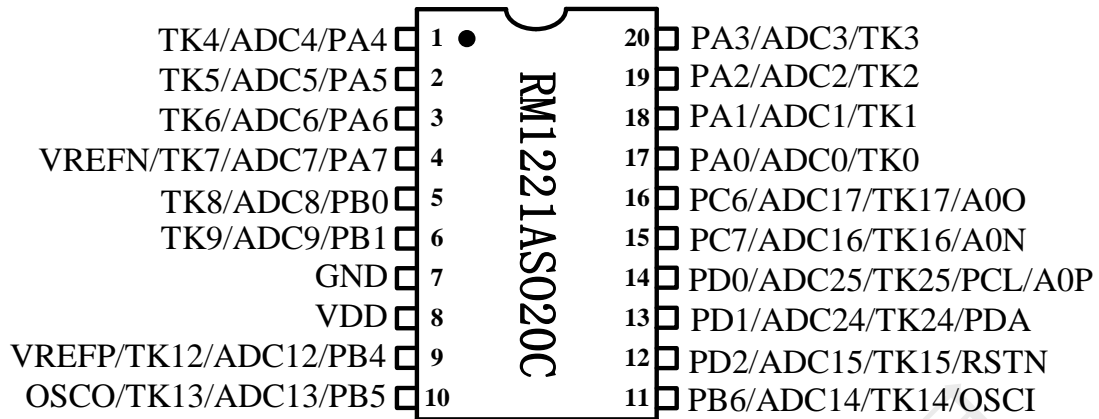
RM1221ASO28E/RM1221ATS28E (SOP28/TSSOP28)



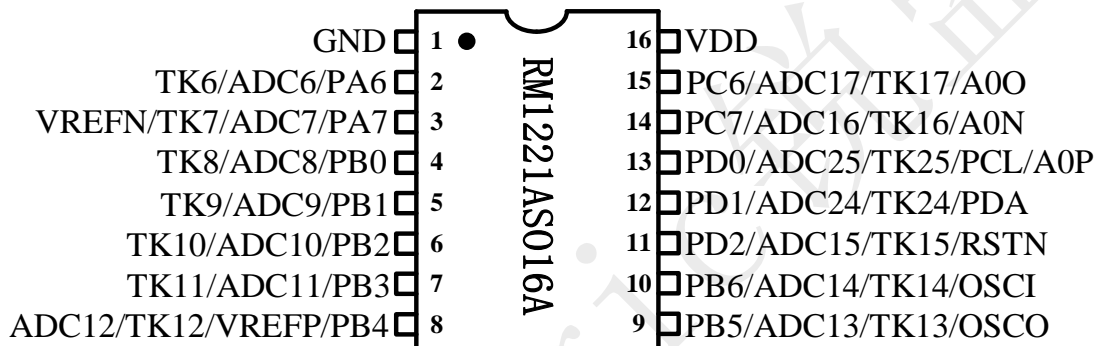
RM1221ASO20A (SOP20)



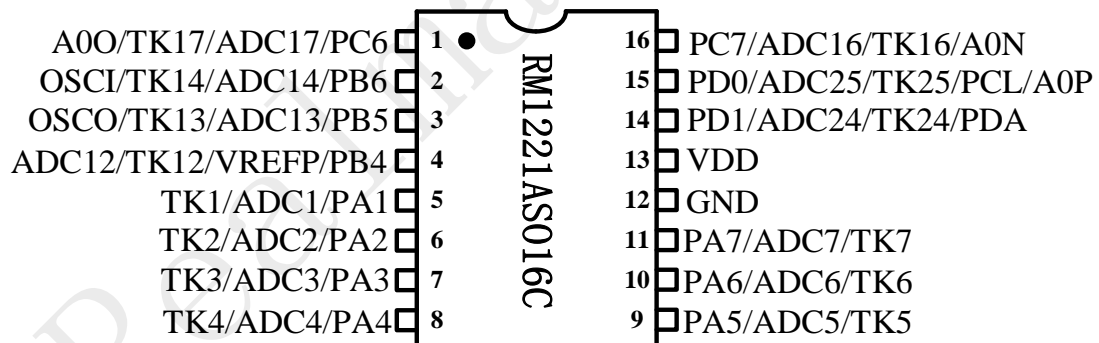
RM1221ASO20B (SOP20)



RM1221ASO20C (SOP20)

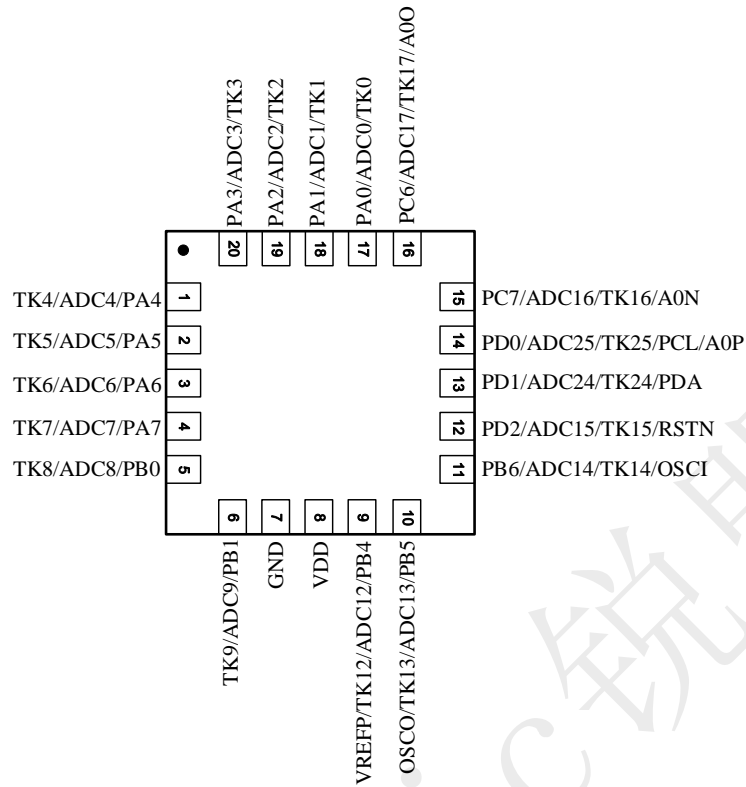


RM1221ASO16A (SOP16)

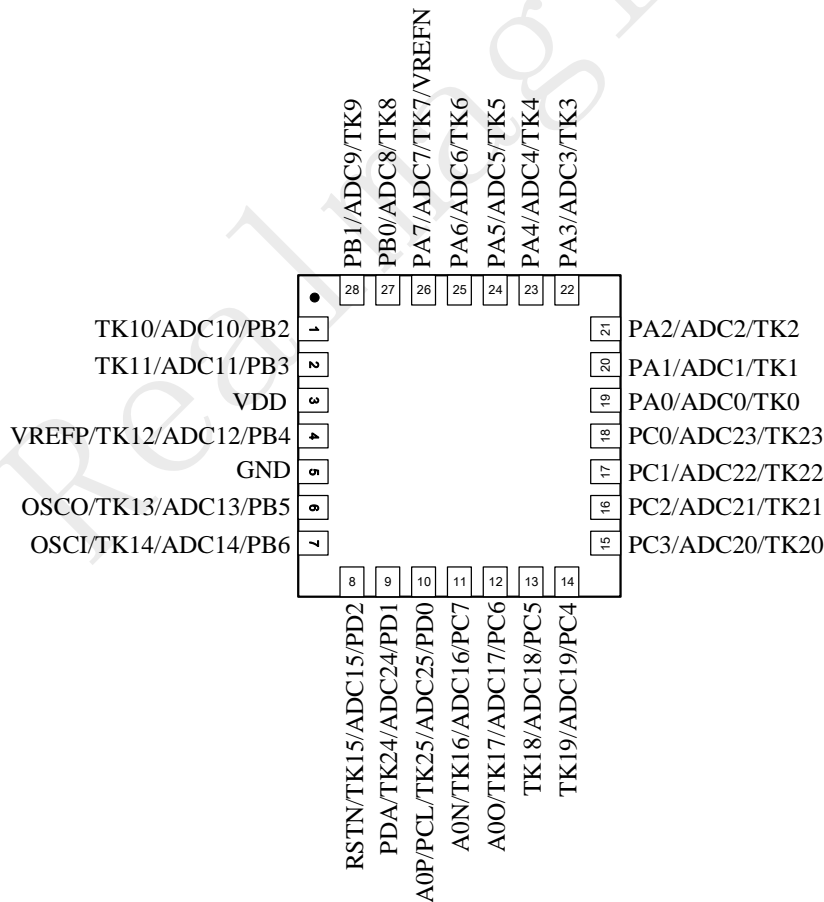


RM1221ASO16C (SOP16)





RM1221AQN20A (QFN4\*4-20L)



RM1221AQN28A (QFN4\*4-28L)

## 2.2 脚位定义

管脚	管脚名称	管脚类型	描述
PA3	PA3	I/O	双向 GPIO 口
	ADC3	AI	ADC 通道 3 输入
	TK3	AI	触摸按键通道 3 输入
PA4	PA4	I/O	双向 GPIO 口
	ADC4	AI	ADC 通道 4 输入
	TK4	AI	触摸按键通道 4 输入
PA5	PA5	I/O	双向 GPIO 口
	ADC5	AI	ADC 通道 5 输入
	TK5	AI	触摸按键通道 5 输入
PA6	PA6	I/O	双向 GPIO 口
	ADC6	AI	ADC 通道 6 输入
	TK6	AI	触摸按键通道 6 输入
PA7	PA7	I/O	双向 GPIO 口
	ADC7	AI	ADC 通道 7 输入
	TK7	AI	触摸按键通道 7 输入
	VREFN	AI	ADC 参考电压负端
PB0	PB0	I/O	双向 GPIO 口
	ADC8	AI	ADC 通道 8 输入
	TK8	AI	触摸按键通道 8 输入
PB1	PB1	I/O	双向 GPIO 口
	ADC9	AI	ADC 通道 9 输入
	TK9	AI	触摸按键通道 9 输入
PB2	PB2	I/O	双向 GPIO 口
	ADC10	AI	ADC 通道 10 输入
	TK10	AI	触摸按键通道 10 输入
PB3	PB3	I/O	双向 GPIO 口
	ADC11	AI	ADC 通道 11 输入
	TK11	AI	触摸按键通道 11 输入
PB4	PB4	I/O	双向 GPIO 口
	ADC12	AI	ADC 通道 12 输入
	TK12	AI	触摸按键通道 12 输入
	VREFP	AI	ADC 参考电压正端
PB5	PB5	I/O	双向 GPIO 口
	ADC13	AI	ADC 通道 13 输入
	TK13	AI	触摸按键通道 13 输入
	OSCO	O	外部振荡器输出口
PB6	PB6	I/O	双向 GPIO 口
	ADC14	AI	ADC 通道 14 输入

	TK14	AI	触摸按键通道 14 输入
	OSCO	AI	外部振荡器输入口
VSS	VSS	P	地
VDD	VDD	P	电源
PD2	PD2	I/O	双向 GPIO 口
	ADC15	AI	ADC 通道 15 输入
	TK15	AI	触摸按键通道 15 输入
	RSTN	I	外部复位输入
PD1	PD1	I/O	双向 GPIO 口
	ADC24	AI	ADC 通道 24 输入
	TK24	AI	触摸按键通道 24 输入
	PDA	I/O	串行编程和调试数据输入输出
PD0	PD0	I/O	双向 GPIO 口
	ADC25	AI	ADC 通道 25 输入
	TK25	AI	触摸按键通道 25 输入
	PCL	I/O	串行编程和调试时钟输入
	A0P	AI	运算放大 0 正端输入
PC7	PC7	I/O	双向 GPIO 口
	ADC16	AI	ADC 通道 16 输入
	TK16	AI	触摸按键通道 16 输入
	A0N	AI	运算放大 0 负端输入
PC6	PC6	I/O	双向 GPIO 口
	ADC17	AI	ADC 通道 17 输入
	TK17	AI	触摸按键通道 17 输入
	A0O	AI	运算放大输出
PC5	PC5	I/O	双向 GPIO 口
	ADC18	AI	ADC 通道 18 输入
	TK18	AI	触摸按键通道 18 输入
PC4	PC4	I/O	双向 GPIO 口
	ADC19	AI	ADC 通道 19 输入
	TK19	AI	触摸按键通道 19 输入
PC3	PC3	I/O	双向 GPIO 口
	ADC20	AI	ADC 通道 20 输入
	TK20	AI	触摸按键通道 20 输入
PC2	PC2	I/O	双向 GPIO 口
	ADC21	AI	ADC 通道 21 输入
	TK21	AI	触摸按键通道 21 输入
PC1	PC1	I/O	双向 GPIO 口
	ADC22	AI	ADC 通道 22 输入
	TK22	AI	触摸按键通道 22 输入
PC0	PC0	I/O	双向 GPIO 口
	ADC23	AI	ADC 通道 23 输入

	TK23	AI	触摸按键通道 23 输入
PA0	PA0	I/O	双向 GPIO 口
	ADC0	AI	ADC 通道 0 输入
	TK0	AI	触摸按键通道 0 输入
PA1	PA1	I/O	双向 GPIO 口
	ADC1	AI	ADC 通道 1 输入
	TK1	AI	触摸按键通道 1 输入
PA2	PA2	I/O	双向 GPIO 口
	ADC2	AI	ADC 通道 2 输入
	TK2	AI	触摸按键通道 2 输入

注：引脚类型 I = 数字输入, O = 数字输出; AI = 模拟输入; P = 电源; AP = 模拟电源; ST = 施密特触发

## 2.3 数字外设功能引脚全映射模块 PTM

RM1221A 内置数字外设功能引脚可配置模块，绝大多数的数字外设功能引脚可通过配置模块重新映射到任何一个非电源地 IO 引脚上，且保持数字外设与硬件端口一一对应的布局关系。

### 2.3.1 PTM 模块特性

- ◆ 数字外设功能引脚无默认 IO 口，要使用数字外设功能引脚的功能均需要先配置。
- ◆ 外设功能引脚为输入特性时（如 T0/1 外部输入、RXD 等），可配置模块允许其多对一映像，以此来优化用户系统。
- ◆ 外设功能引脚为输出特性时（如 T0/1 时钟输出、TXD 等），如果多个外设功能输出配置到同一 IO 口，可配置模块将按优先级表输出优先级最高的外设功能输出。
- ◆ 软件操作，使用灵活。用户在应用系统设计和开发过程中，可根据实际情形灵活分配数字外设功能口，来达到减少改动、降低成本、缩短开发周期的目的。

### 2.3.2 PTM 可全映射外设功能口

芯片仅绝大多数数字外设功能可映射，所有的模拟功能脚（如 ADC 输入、触摸按键输入、外部中断（INT0/INT1）、IO 唤醒功能口、晶振脚等）不能映射，且外部硬件复位 RSTN 也不能映射。

可配置的数字外设功能口列表如下：

外设	名称	类型	说明
PWM	PWM0	O	PWM0 输出
	PWM1	O	PWM1 输出
	PWM2	O	PWM2 输出
	PWM3	O	PWM3 输出
	PWM4	O	PWM4 输出
	PWM5	O	PWM5 输出
	FB	I	故障刹车输入脚
定时器 0/1	T0OUT	O	T0 时钟分频输出
	T0CNT	I	T0 的外部输入
	T1OUT	O	T1 时钟分频输出
	T1CNT	I	T1 的外部输入
定时器 2	CAP0	I	T2 的输入捕获通道 0
	CAP1	I	T2 的输入捕获通道 1
	CAP2	I	T2 的输入捕获通道 2
定时器 4	T4PWM0	O	T4PWM0 输出
	T4PWM1	O	T4PWM1 输出
CLK	CLKO	O	时钟输出口
UART	TXD0	O	UART0 数据传输口
	RXD0	I/O	UART0 接收脚
	TXD1	O	UART1 数据传输口
	RXD1	I/O	UART1 接收脚
SPI	SCK	I/O	SPI 的时钟口
	MOSI	I/O	SPI 的数据口, 主机的输出和从机的输入
	MISO	I/O	SPI 的数据口, 主机的输入和从机的输出
	NSS	I	SPI 的片选口
IIC	SCL	I/O	IIC 时钟口
	SDA	I/O	IIC 数据口
外部中断	$\overline{\text{INT0}}$	I	外部中断 0 输入
	$\overline{\text{INT1}}$	I	外部中断 1 输入
ADC	STADCPX	I	外部触发 ADC 管脚输入

## 3 中央处理器（CPU）

### 3.1 概述

RM1221A 的中央处理器（CPU）是一个增强型 1T 兼容 8051 的内核，在同样的系统时钟下，较之传统的 8051 芯片具有运行更快速，性能更优越的特性。

### 3.2 中央处理器（CPU）相关寄存器

#### 3.2.1 程序计数器（PC）

程序计数器 PC 在物理上是独立的，不属于 SFR 之列。PC 字长 16 位，是专门用来控制指令执行顺序的寄存器。芯片上电或复位后，PC 的值为 0000H，这样芯片从程序的零地址开始执行程序。

#### 3.2.2 累加器（ACC）

累加器（ACC）在指令系统中又记做 A，用于向算术逻辑单元（ALU）提供操作数和存放运算结果，它是 CPU 中工作最频繁的寄存器，大多数指令的执行都要通过累加器 ACC 进行。

#### 3.2.3 寄存器（B）

寄存器 B 是专门为乘法和除法运算设置的寄存器，用于存放乘法和除法运算的操作数和运算结果，在不进行乘除运算时，可以作为通用寄存器使用。

#### 3.2.4 程序状态字寄存器（PSW）

此寄存器用来保存算术逻辑单元（ALU）运算结果的特征和处理状态，这些特征和状态可以作为控制程序转移的条件，供程序判别和查询，PSW 描述如下所示。

寄存器	地址	R/W	描述	复位后的值
PSW	D0H	R/W	程序状态寄存器	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CY	AC	F0	RS1	RS0	OV	F1	P

位	描述
---	----

[7]	<b>CY</b>	<b>高位进位标志</b> 进行加法或减法操作时，当前运算需要向高位进位或借位时，CY 将置位，否则清零。 在进行 MUL 或 DIV 运算时，CY 始终为 0。 CY 受 DAA 指令影响，用来表示是否初始 BCD 数大于 100。 在 CJNE 指令中，如果第一个无符号数的值小于第二个，则 CY 置 1，否则清 0。
[6]	<b>AC</b>	<b>辅助进位标志</b> 当前运算导致从半字节的低序第 4 位进位或借位，该位置位，否则清零。
[5]	<b>F0</b>	<b>用户标志 0</b> 可由用户置位或清零的通用标志。
[4]	<b>RS[1:0]</b>	<b>寄存器页选择</b> 这两位用来选择 R0 到 R7 位于四页中的哪一页：. <ul style="list-style-type: none"> <li>00: 寄存器页 0, RAM 地址 00H 到 07H</li> <li>01: 寄存器页 1, RAM 地址 08H 到 0FH</li> <li>10: 寄存器页 2, RAM 地址 10H 到 17H</li> <li>11: 寄存器页 3, RAM 地址 18H 到 1FH</li> </ul>
[2]	<b>OV</b>	<b>溢出标志</b> OV 用于标示发生溢出。 对于加法指令 ADD 或 ADDC 指令中，如果位 6 有进位而位 7 没进位，或者位 7 有进位而位 6 没有进位，则溢出标志置“1”，反之清“0”。 OV 也用于标示有符号数累加结果，当两个正数相加，或两个负数相加结果为负数时 OV 为 1。 对于减法指令 SUBB，当位 6 发生借位而位 7 没有，或者位 7 发生借位而位 6 没有借位，则溢出标志置“1”，反之清“0”。 OV 也用于标示两个数相减时，当一个正数加一负数结果为负，或两个负数相减结果为负时。 对于 MUL 乘法指令，当结果大于 255 (00FFH)时，OV 置 1。反之清 0。 对于 DIV 除法指令，通常情况下 OV 为 0。除非当 B 设定值为 00H，则 A 和 B 的返回值为随机值，同时 OV 置 1。
[1]	<b>F1</b>	<b>用户标志 1</b> 可由用户置位或清零的通用标志。
[0]	<b>P</b>	<b>奇偶标志</b> 当累加结果为奇数时，该标志置 1，偶数时清 0。其执行奇偶校验。

指令运行对标志位 CY、OV、AC 影响如下表 3-2-1 所示：

表 3-2-1 影响标志位 CY、OV、AC 的指令

指令	CY	OV	AC	指令	CY	OV	AC
ADD	√ <sup>[1]</sup>	√	√	CLR C	0		
ADDC	√	√	√	CPL C	√		
SUBB	√	√	√	ANL C, bit	√		
MUL	0	√		ANL C, /bit	√		
DIV	0	√		ORL C, bit	√		
DA	√			ORL C, /bit	√		

RRC	√			MOV C, bit	√		
RLC	√			CJNE	√		
SETB C	1						

[1] √表示根据指令的结果变化

### 3.2.5 堆栈指针 SP

堆栈指针 SP 是一个 8 位专用寄存器，在执行 PUSH、各种子程序调用、中断响应等指令时，SP 先加 1，再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减 1。堆栈栈顶可以是片上内部 RAM (00H-FFH) 的任意地址，系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 地址开始。

### 3.2.6 双数据指针 DPTR

数据指针 DPTR 是一个 16 位专用寄存器，其高位字节寄存器用 DPH 表示，低位字节寄存器用 DPL 表示。它们既可以作为一个 16 位寄存器 DPTR 来处理，也可以作为 2 个独立的 8 位寄存器 DPH 和 DPL 来处理。

RM1221A 提供两组数据指针，这样程序可以分别同时定义源地址和目标地址，直接进行数据移动。程序通过 DPS(AUXR1.0)位切换 DPTR0 及 DTTR1。



## 4 MTP ROM 和 SRAM

### 4.1 程序存储器 (ROM)

程序存储用于存放用户程序、数据等信息, RM1221A 内部集成了 16K 字节的 MTP 程序存取空间, 其地址为 0000~3FFFH, 可对其进行 Sector size(1K Bits)或 Page size(1Kx8 Bits) 读写操作详情如下结构图。其中 0000H-00EBH 是单片机各种中断服务程序入口地址(详情请见: 7.中断章节)。

RM1221A 的 16KB MTP 可分为两个内部编程块 APROM 和 LDROM。虽然他们都和标准 8051 编程内存一样, 但是它们 ROM 的大小不一样, LDROM 大小为 1KB。当使用 LDROM 时, APROM 的最大值 15KB; 当不使用 LDROM 时, APROM 的最大值为 16KB。APROM 通常用来放用户代码, CPU 从 APROM 获取指令来执行, MOVC 指令也可以从这个区域读取。

LDROM 的功能通常是上电初始化、自检以及存储启动代码用于 ISP。它可以更新 APROM 空间和 Code Option 字节。APROM 中的代码也可以重新 LDROM。APROM 和 LDROM 关于 IAP 的细节和配置位设置, 请看[章节 25.IAP、EEPROM、BootLoader 及安全加密](#)。可以认为 APROM 和 LDROM 是硬件独立模块, 因此如果 CPU 从 LDROM 启动, CPU 会自动重映射 PC 指针 0000H 到 LDROM 开始的地址, CPU 认为 LDROM 是单独编程内存且所有中断向量独立于 APROM。

此外还集成了独立的 EEPROM 区域, 其地址为 000H~1FFH, 可对其进行 Sector size(32 Bits)或 Page size (32x8 Bits) 读写操作。EEPROM 区分为 Code 区 (000H~1BFH, IAP 可读写)、User ID 区 (1C0H~1CFH, IAP 可读写)、UID 区 (1D0H~1DFH, IAP 可读不可写)、CONFIG 区 (1E0H~1EFH, LDROM 程序 IAP 可读写)、Reserved 区 (1F0H~1FFH, 不可读写)。

程序存储器结构图:

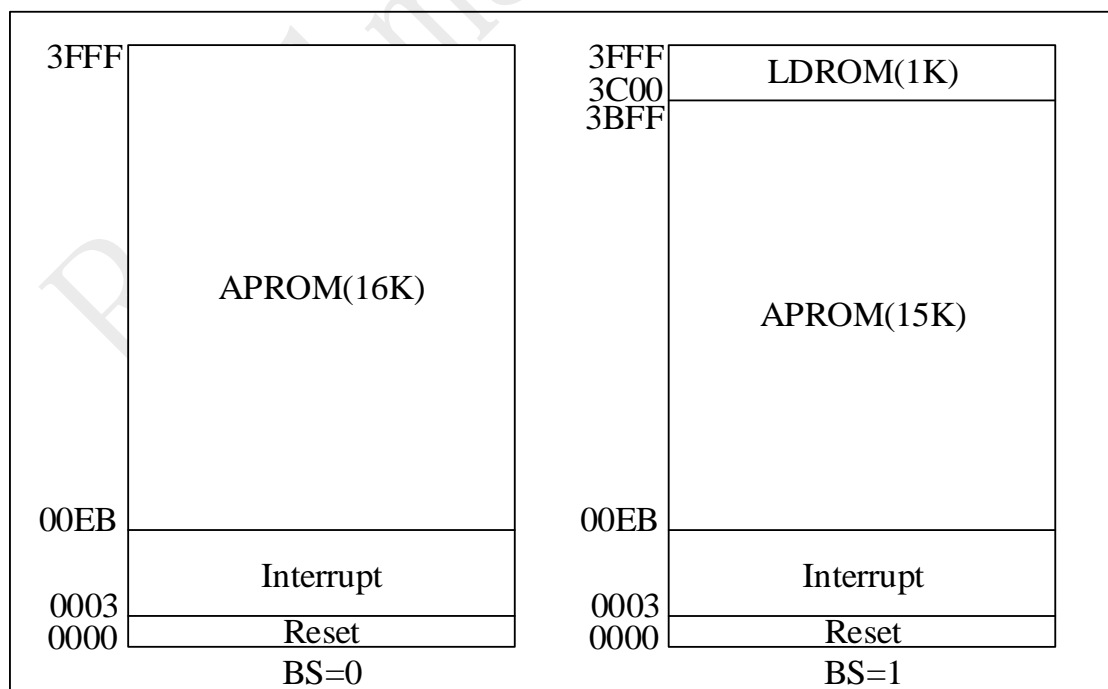


图 4-1-1 程序存储器结构图

EEPROM 结构图:

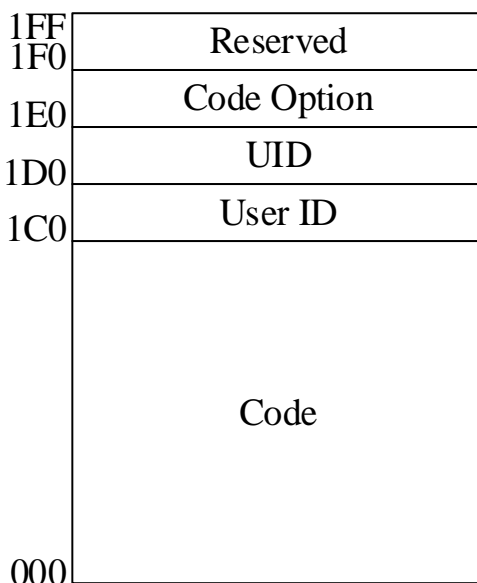


图 4-1-2 EEPROM 结构图

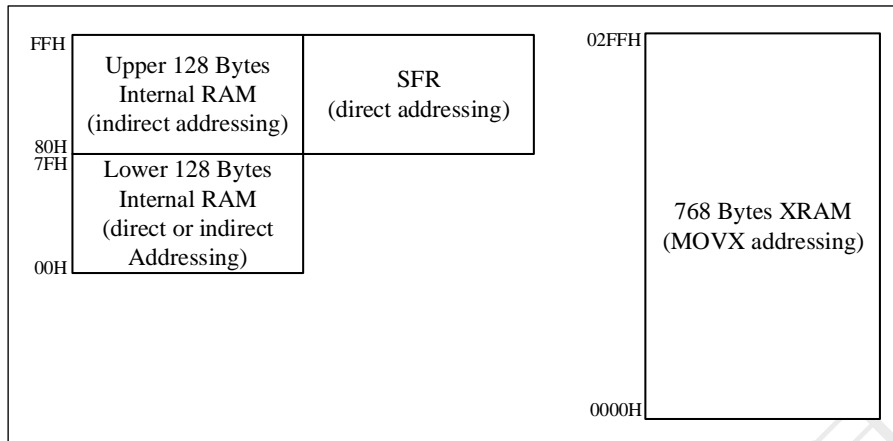
## 4.2 随机数据存储器 (RAM)

RM1221A 为数据存储提供了内部 RAM 和外部 RAM。下列为存储器空间分配:

- ◆ 低位 128 字节的 RAM (地址从 00H 到 7FH) 可直接或间接寻址。
- ◆ 高位 128 字节的 RAM (地址从 80H 到 FFH) 只能间接寻址。
- ◆ 特殊功能寄存器 (SFR, 地址从 80H 到 FFH) 只能直接寻址。
- ◆ 外部 RAM 可通过 MOVX 指令间接访问。

高位 128 字节的 RAM 占用的地址空间和 SFR 相同,但在物理上与 SFR 的空间是分离的。当一个指令访问高于地址 7FH 的内部位置时, CPU 可以根据访问的指令类型来区分是访问高位 128 字节数据 RAM 还是访问 SFR。

RM1221A 在外部数据空间额外提供了 768 字节 XRAM, 来扩大 RAM 空间。访问内部扩展 RAM 的方法和传统 8051 单片机访问外部扩展 RAM 的方法相同,但是不影响 I/O 口。在汇编语言中,内部扩展 RAM 通过 MOVX 指令访问,即 MOVX @DPTR 或者 MOVX @Ri。


**图 4-2-1 内部 256Bytes RAM 结构图**

内部 256Bytes RAM 地址如下：

FFH	Indirect Accessing RAM								
80H 7FH	Direct or Indirect Accessing RAM								
30H	7F	7E	7D	7C	7B	7A	79	78	} Bit-addressable
2FH	77	76	75	74	73	72	71	70	
2EH	6F	6E	6D	6C	6B	6A	69	68	
2DH	67	66	65	64	63	62	61	60	
2CH	5F	5E	5D	5C	5B	5A	59	58	
2BH	57	56	55	54	53	52	51	50	
2AH	4F	4E	4D	4C	4B	4A	49	48	
29H	47	46	45	44	43	42	41	40	
28H	3F	3E	3D	3C	3B	3A	39	38	
27H	37	36	35	34	33	32	31	30	
26H	2F	2E	2D	2C	2B	2A	29	28	
25H	27	26	25	24	23	22	21	20	
24H	1F	1E	1D	1C	1B	1A	19	18	
23H	17	16	15	14	13	12	11	10	
22H	0F	0E	0D	0C	0B	0A	09	08	
21H	07	06	05	04	03	02	01	00	
20H	Register Bank3								} General Purpose Registers
1FH	Register Bank2								
18H 17H	Register Bank1								
10H 0FH	Register Bank0								
08H 07H									
00H									

**图 4-2-2 内部 256Bytes RAM 地址图**

## 4.3 特殊功能寄存器(SFR)

特殊功能寄存器是 CPU 和外设模块用来控制所需的器件操作的寄存器。

### 4.3.1 直接寻址读写 SFR

Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
<b>F8</b>	TKCON	TKF	TK0ISL	TK0ISH	—	—	—	—
<b>F0</b>	B	TKCHS	RISEINTF	FALLINTF	TKRAWDL	TKRAWDH	DIFFDL	DIFFDH
<b>E8</b>	SPCR	SPDR	SPSR	I2CADDR	I2CDAT	I2CCLK	I2CSTAT	PWMINTC
<b>E0</b>	ACC	PMEN	PWMOVRL	PWMOVRH	PNP	FBD	PDTEN	PDTCNT
<b>D8</b>	PWMCON0	PWMCON1	PWMD3L	PWMD3H	PWMD4L	PWMD4H	PWMD5L	PWMD5H
<b>D0</b>	PSW	PMD	PWMD0L	PWMD0H	PWMD1L	PWMD1H	PWMD2L	PWMD2H
<b>C8</b>	I2CCON	T2MOD	T2C0L	T2C0H	T2C1L	T2C1H	T2C2L	T2C2H
<b>C0</b>	T2CON	CAPCON0	CAPCON1	CAPCON2	TL2	TH2	RCMP2L	RCMP2H
<b>B8</b>	IP0	IP1	IAPTRG	IAPAL	IAPAH	IAPDATA	IAPCON	CHPCON
<b>B0</b>	PORTD	EXA0	EXA1	EXA2	EXA3	EXB0	EXB1	MDCON
<b>A8</b>	IE	ADCCON0	ADCCON1	ADCCON2	ADCCON4	ADCRL	ADCRH	—
<b>A0</b>	PORTC	PAINTRF	PBINTRF	PCINTRF	PDINTRF	T3CON	T3RL	T3RH
<b>98</b>	SOCON	SOBUF	S0OVRL	S0OVRH	RCTRIML	RCTRIMH	AUXR1	WPT
<b>90</b>	PORTB	PAM	PBM	PCM	PDM	EIE	EIE1	METCH
<b>88</b>	TCON	TMOD	TL0	TL1	TH0	TH1	WDTCON	CKCON
<b>80</b>	PORTA	SP	DPL0	DPH0	DPL1	DPH1	—	PCON

#### 4.3.1.1 直接寻址读写 SFR 说明

直接寻址读写 SFR 的具体解释说明如下：

地址	符号	说明	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	复位值	
80	PORTA	PA口数据寄存器	PORTA[7:0]									0000 0000
81	SP	堆栈指针	SP[7:0]									0000 0000
82	DPL0	DPTR0数据指针低位	DPTR0[7:0]									0000 0000
83	DPH0	DPTR0数据指针高位	DPTR0[15:8]									0000 0000
84	DPL1	DPTR1数据指针低位	DPTR1[7:0]									0000 0000
85	DPH1	DPTR1数据指针高位	DPTR1[15:8]									0000 0000
87	PCON	电源管理寄存器	—	—	—	—	—	—	PD	IDL	0000 0000	
88	TCON	定时器0/1控制寄存器	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000	
89	TMOD	定时器0/1模式寄存器	GATE1	CT1	M1[1:0]	GATE0	CT0	M0[1:0]				0000 0000
8A	TL0	定时器0低8位数据寄存器	TL0[7:0]									0000 0000
8B	TL1	定时器1低8位数据寄存器	TL1[7:0]									0000 0000
8C	TH0	定时器0高8位数据寄存器	TH0[7:0]									0000 0000
8D	TH1	定时器1高8位数据寄存器	TH1[7:0]									0000 0000
8E	WDTCON	WDT配置寄存器	WDTRF	CLRWDT	WDTCKPS[3:0]				WDTFS	WDTEN		0000 0000
8F	CKCON	系统时钟配置寄存器	TRIMCLKEN	—	SYSCLOCKEN	WCKEN	SYSCCLKS	—	DEBUGEN	WCKS	0011 0011	
90	PORTB	PB口数据寄存器	PORTB[7:0]									0000 0000
91	PAM	PA口输入输出控制寄存器	PAM[7:0]									0000 0000
92	PBM	PB口输入输出控制寄存器	PBM[7:0]									0000 0000
93	PCM	PC口输入输出控制寄存器	PCM[7:0]									0000 0000
94	PDM	PD口输入输出控制寄存器	PDM[7:0]									0000 0000
95	EIE	扩展中断使能寄存器	ET2	ESPI	ETK	EATKMAX	EPWM	ECAP	EGPIO	EI2C	0000 0000	
96	EIE1	扩展中断使能寄存器1	ETRIMCLK	EECKPD	ET4PWM	EFB	EUART1	ET4	ET3	EWKT	0000 0000	
97	METCH	备用寄存器	METCH[7:0]									0000 0000
98	S0CON	串口0控制寄存器	—	—	SOB8EN	SOREN	SOTB8	SORB8	SOT1	SOR1	0000 0000	
99	S0BUF	串口0数据缓存寄存器	S0BUF[7:0]									0000 0000
9A	S0OVR1	串口0波特率高低字节	S0OVR[7:0]									0000 0000
9B	S0OVRH	串口0波特率高低字节	—	—	—	—	—	S0OVR[10:8]			0000 0000	
9C	RCTRIML	内部高速时钟TRIM低8位	RCTRIM[7:0]									0000 0000
9D	RCTRIMH	内部高速时钟TRIM高2位	—	—	—	—	—	RCTRIM[9:8]			0000 0010	
9E	AUXR1	辅助功能寄存器1	SWRF	RSTPINF	HARDF	IOVTH	TIM	T0M	—	DPS	0000 0000	
9F	WPT	时控保护寄存器	WPT[7:0]									0000 0000
A0	PORTC	PC口数据寄存器	PORTC[7:0]									0000 0000
A1	PAINTRF	PA翻转中断标志寄存器	PAINTRF[7:0]									0000 0000
A2	PBINTRF	PB翻转中断标志寄存器	PBINTRF[7:0]									0000 0000
A3	PCINTRF	PC翻转中断标志寄存器	PCINTRF[7:0]									0000 0000
A4	PDINTRF	PD翻转中断标志寄存器	PDINTRF[7:0]									0000 0000
A5	T3CON	定时器3控制寄存器	TF3	—	T3DIV[2:0]			TR3	—	T3MOD	0000 0000	
A6	T3RL	定时器3低字节数据寄存器	T3R[7:0]									0000 0000
A7	T3RH	定时器3高字节数据寄存器	T3R[15:8]									0000 0000
A8	IE	中断使能寄存器	EA	EADC	EBOD	EUART0	ET1	EXT1	ET0	EXT0	0000 0000	
A9	ADCCON0	ADC控制寄存器0	ADCF	ADCS	ETGSEL[1:0]		—	—	—	—	0000 0000	
AA	ADCCON1	ADC控制寄存器1	CHOP	ADVREFS[2:0]			ETGYYP[1:0]		ADCEX	ADCEN	0000 0000	
AB	ADCCON2	ADC控制寄存器2	ADFBEN	ADCMPOP	ADCM PEN	ADCMPO	—	ADCKS[2:0]			0000 0010	
AC	ADCCON4	ADC控制寄存器4	DFILEN	DFILNUM[1:0]			ADCHS[4:0]				0000 0000	
AD	ADCR1	ADC转换结果低位寄存器	ADCR[7:0]									0000 0000
AE	ADCRH	ADC转换结果高位寄存器	ADCR[15:8]									0000 0000
B0	PORTD	PD口数据寄存器	PORTD[7:0]									0000 0000
B1	EXA0	扩展累加器0	EXA0[7:0]									0000 0000
B2	EXA1	扩展累加器1	EXA1[7:0]									0000 0000
B3	EXA2	扩展累加器2	EXA2[7:0]									0000 0000
B4	EXA3	扩展累加器3	EXA3[7:0]									0000 0000
B5	EXB0	扩展B寄存器0	EXB0[7:0]									0000 0000
B6	EXB1	扩展B寄存器1	EXB1[7:0]									0000 0000
B7	MDCON	乘法运算控制寄存器	OPERS	MD	—	—	—	—	—	—	0000 0000	
B8	IP0	中断优先级配置寄存器0	IP0[7:0]									0000 0000
B9	IP1	中断优先级配置寄存器1	IP1[7:0]									0000 0000
BA	IAPTRG	IAP执行寄存器	—	—	—	—	—	—	—	IAPGO	0000 0000	
BB	IAPAL	IAP地址低字节	IAPAL[7:0]									0000 0000
BC	IAPAH	IAP地址高字节	IAPAH[7:0]									0000 0000
BD	IAPFD	IAP数据寄存器	IAPFD[7:0]									0000 0000
BE	IAPCN	IAP控制寄存器	—	—	—	—	IAPCN[3:0]				0000 0000	
BF	CHPCON	IAP使能寄存器	SWRST	—	—	—	—	—	BS	IAPEN	0000 0000	
C0	T2CON	定时器2控制寄存器	TF2	—	—	—	—	TR2	—	CM/RL2	0000 0000	
C1	CAPCON0	输入捕获控制寄存器0	—	CAPEN2	CAPEN1	CAPEN0	—	CAPF2	CAPF1	CAPF0	0000 0000	
C2	CAPCON1	输入捕获控制寄存器1	—	—	CAPZLS[1:0]		CAPILS[1:0]		CAPOLS[1:0]		0000 0000	
C3	CAPCON2	输入捕获控制寄存器2	—	ENF2	ENF1	ENF0	—	—	—	—	0000 0000	
C4	TL2	定时器2低字节数据	TL2[7:0]									0000 0000
C5	TH2	定时器2高字节数据	TH2[7:0]									0000 0000
C6	RCMP2L	定时器2重载/比较数据低字节	RCMP2[7:0]									0000 0000
C7	RCMP2H	定时器2重载/比较数据高字节	RCMP2[15:8]									0000 0000

C8	I2CON	IIC控制寄存器	-	I2CEN	STA	STO	I2CIF	AA	-	-	0000 0000	
C9	T2MOD	定时器2模式选择	L2DEN	T2DIV[2:0]			CAPCR	CMPCR	LDT[1:0]		0000 0000	
CA	T2C0L	捕获通道0低字节数据	T2C0[7:0]								0000 0000	
CB	T2C0H	捕获通道0高字节数据	T2C0[15:8]								0000 0000	
CC	T2C1L	捕获通道1低字节数据	T2C1[7:0]								0000 0000	
CD	T2C1H	捕获通道1高字节数据	T2C1[15:8]								0000 0000	
CE	T2C2L	捕获通道2低字节数据	T2C2[7:0]								0000 0000	
CF	T2C2H	捕获通道2高字节数据	T2C2[15:8]								0000 0000	
D0	PSW	程序状态寄存器	CY	AC	F0	RS1	RS0	OV	F1	P	0000 0000	
D1	PMD	PWM掩码数据寄存器	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000	
D2	PWMD0L	PWM0占空比低字节寄存器	PWMD0[7:0]								0000 0000	
D3	PWMD0H	PWM0占空比高字节寄存器	PWMD0[15:8]								0000 0000	
D4	PWMD1L	PWM1占空比低字节寄存器	PWMD1[7:0]								0000 0000	
D5	PWMD1H	PWM1占空比高字节寄存器	PWMD1[15:8]								0000 0000	
D6	PWMD2L	PWM2占空比低字节寄存器	PWMD2[7:0]								0000 0000	
D7	PWMD2H	PWM2占空比高字节寄存器	PWMD2[15:8]								0000 0000	
D8	PWMC0N0	PWM控制寄存器0	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000	
D9	PWMC0N1	PWM控制寄存器1	PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]			0000 0000	
DA	PWMD3L	PWM3占空比低字节寄存器	PWMD3[7:0]								0000 0000	
DB	PWMD3H	PWM3占空比高字节寄存器	PWMD3[15:8]								0000 0000	
DC	PWMD4L	PWM4占空比低字节寄存器	PWMD4[7:0]								0000 0000	
DD	PWMD4H	PWM4占空比高字节寄存器	PWMD4[15:8]								0000 0000	
DE	PWMD5L	PWM5占空比低字节寄存器	PWMD5[7:0]								0000 0000	
DF	PWMD5H	PWM5占空比高字节寄存器	PWMD5[15:8]								0000 0000	
E0	ACC	累加器	ACC[7:0]								0000 0000	
E1	PMEN	PWM输出掩码控制寄存器	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000	
E2	PWMOVRL	PWM周期低字节寄存器	PWMOVR[7:0]								0000 0000	
E3	PWMOVRH	PWM周期高字节寄存器	PWMOVR[15:8]								0000 0000	
E4	PNP	PWM负极性寄存器	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000	
E5	FBD	PWM故障刹车数据寄存器	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	FBD0	0000 0000	
E6	PDTEN	PWM死区控制寄存器	-	-	-	PDTCNT.8	-	PDT45EN	PDT23EN	PDT01EN	0000 0000	
E7	PDTCNT	PWM死区时间计数器	PDTCNT[7:0]								0000 0000	
E8	SPCR	SPI控制寄存器	SPR2	SPIEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0	0011 0100	
E9	SPDR	SPI状态寄存器	SPDR[7:0]								0000 0000	
EA	SPSR	SPI控制寄存器	SPIF	WCOL	SSERR	MODF	-	-	-	-	0000 0000	
EB	I2ADDR	IIC地址寄存器	I2ADDR[7:0]								0000 0000	
EC	I2DAT	IIC数据寄存器	I2DAT[7:0]								0000 0000	
ED	I2CLK	IIC时钟寄存器	I2CLK[7:0]								0000 0000	
EE	I2STAT	IIC状态寄存器	I2STAT[4:0]						-	-	-	1111 1000
EF	PWMINTC	PWM中断控制寄存器	-	-	INTTYP[1:0]		-	INTSEL[2:0]		0000 0000		
F0	B	寄存器B	B[7:0]								0000 0000	
F1	TKCHS	TK扫描通道配置寄存器	TKCHS[4:0]								0000 0000	
F2	RISEINTF	ATK上升事件中标志寄存器	RISE_FALL_INTF[3:0]			RISE_RISE_INTF[3:0]					0000 0000	
F3	FALLINTF	ATK下降事件中标志寄存器	FALL_FALL_INTF[3:0]			FALL_RISE_INTF[3:0]					0000 0000	
F4	TKRAWDL	CTK通道测量当前值低8位	TKRAWDL[7:0]								0000 0000	
F5	TKRAWDH	CTK通道测量当前值高8位	TKRAWDH[15:8]								0000 0000	
F6	DIFFDL	触摸按键DIFF低8位数据	DIFFDL[7:0]								1111 1111	
F7	DIFFDH	触摸按键DIFF高8位数据	DIFFDH[15:8]								1111 1111	
F8	TKCON	CTK模块控制寄存器	TKEOC	CPRECHG	TKCHS[1:0]		TKSOC	TKINTS	TKAUTO	TKEN	0000 0000	
F9	TKF	CTK标志寄存器	ATKIF	TKOVF	ATKMAXI F	ATKDT[3:0]					0000 0000	
FA	ATK0ISL	TK0补偿电流源低8位寄存器	ATK0ISL[7:0]								0000 0000	
FB	ATK0ISH	TK0补偿电流源高位寄存器	ATK0ISH[8:0]								1000 0000	

### 4.3.2 外部扩展 XSFR

扩展 XSFR 采用和 XRAM 同样的访问方式，使用 MOVX A, @DPTR 和 MOVX @DPTR, A 来进行读写。

例如写一个地址为 0xF010 的 XSFR，操作如下：

```
MOV A, #wdata
```

```
MOV DPTR, #0xF010
```

```
MOVX @DPTR, A
```

读地址为 0xF010 的 XSFR，操作如下：

```
MOV DPTR, #0xF010
```

```
MOVX A, @DPTR
```

## 扩展 XSFR（基地址 0xF000）

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x00	PAINTR	0x10	PAUR	0x20	ADCCON3	0x30	LCDCON
0x01	PBINTR	0x11	PADR	0x21	ADCCON5	0x31	PAVO
0x02	PCINTR	0x12	PAOD	0x22	ADCCON6	0x32	PBVO
0x03	PDINTR	0x13	—	0x23	—	0x33	PCVO
0x04	—	0x14	PBUR	0x24	PACHS	0x34	PDVO
0x05	—	0x15	PBDR	0x25	PBCHS	0x35	—
0x06	—	0x16	PLVD	0x26	PCCHS	0x36	—
0x07	—	0x17	—	0x27	PDCHS	0x37	—
0x08	PAIO	0x18	PCUR	0x28	ADCPLY	0x38	OPACON0
0x09	PBIO	0x19	PCDR	0x29	ADCMPL	0x39	OPACON1
0x0A	PCIO	0x1A	PCOD	0x2A	ADCMPLH	0x3A	LVDCON
0x0B	PDIO	0x1B	—	0x2B	—	0x3B	LDO
0x0C	—	0x1C	PDUR	0x2C	—	0x3C	—
0x0D	—	0x1D	PDDR	0x2D	—	0x3D	—
0x0E	—	0x1E	PDOD	0x2E	—	0x3E	—
0x0F	—	0x1F	—	0x2F	—	0x3F	—

## 扩展 XSFR（基地址 0xF040）

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x00	WKCON	0x10	T4CON	0x20	SETID0	0x30	TRIMCON
0x01	RWK	0x11	T4OVR	0x21	SETID1	0x31	—
0x02	—	0x12	T4OVRH	0x22	CHKIDFLAG	0x32	—
0x03	—	0x13	T4D0L	0x23	—	0x33	—
0x04	WDTR	0x14	T4D0H	0x24	—	0x34	—
0x05	—	0x15	T4D1L	0x25	—	0x35	—
0x06	—	0x16	T4D1H	0x26	—	0x36	—
0x07	—	0x17	—	0x27	—	0x37	—
0x08	S1CON	0x18	—	0x28	—	0x38	—
0x09	S1BUF	0x19	—	0x29	—	0x39	—
0x0A	S1OVR	0x1A	—	0x2A	—	0x3A	—
0x0B	S1OVRH	0x1B	—	0x2B	—	0x3B	—
0x0C	—	0x1C	—	0x2C	—	0x3C	—
0x0D	—	0x1D	—	0x2D	—	0x3D	—
0x0E	—	0x1E	—	0x2E	—	0x3E	—
0x0F	—	0x1F	—	0x2F	—	0x3F	—



## 扩展 XSFR（基地址 0xF080）

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x00	TKCONFIG0	0x10	—	0x20	PTHD0L	0x30	ATKBAS0L
0x01	TKCONFIG1	0x11	—	0x21	PTHD0H	0x31	ATKBAS0H
0x02	TKCONFIG2	0x12	TK1ISL	0x22	PTHD1L	0x32	ATKBAS1L
0x03	TKCONFIG3	0x13	TK1ISH	0x23	PTHD1H	0x33	ATKBAS1H
0x04	TKCONFIG4	0x14	TK2ISL	0x24	PTHD2L	0x34	ATKBAS2L
0x05	SHIELD	0x15	TK2ISH	0x25	PTHD2H	0x35	ATKBAS2H
0x06	—	0x16	TK3ISL	0x26	PTHD3L	0x36	ATKBAS3L
0x07	—	0x17	TK3ISH	0x27	PTHD3H	0x37	ATKBAS3H
0x08	ATKCHEN	0x18	RISEINTEN	0x28	NTHD0L	0x38	ATKCHS0
0x09	ATKNUM	0x19	FALLINTEN	0x29	NTHD0H	0x39	ATKCHS1
0x0A	FILTSEL	0x1A	HYSL	0x2A	NTHD1L	0x3A	ATKCHS2
0x0B	DEBOUNCER	0x1B	HYSH	0x2B	NTHD1H	0x3B	ATKCHS3
0x0C	—	0x1C	ATKMAX0	0x2C	NTHD2L	0x3C	KEYSEL0
0x0D	—	0x1D	ATKMAX1	0x2D	NTHD2H	0x3D	KEYSEL1
0x0E	—	0x1E	ATKTMR0	0x2E	NTHD3L	0x3E	KEYSEL2
0x0F	CTKTRIM	0x1F	ATKTMR1	0x2F	NTHD3H	0x3F	KEYSEL3

## 扩展 XSFR（基地址 0xFF80）

偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x00	PWM0_MAP	0x10	SCK_MAP
0x01	PWM1_MAP	0x11	MOSI_MAP
0x02	PWM2_MAP	0x12	MISO_MAP
0x03	PWM3_MAP	0x13	NSS_MAP
0x04	PWM4_MAP	0x14	EXT0_MAP
0x05	PWM5_MAP	0x15	EXT1_MAP
0x06	PWM_FB_MAP	0x16	T0CNT_MAP
0x07	CAPO_MAP	0x17	T0OUT_MAP
0x08	CAP1_MAP	0x18	T1CNT_MAP
0x09	CAP2_MAP	0x19	T1OUT_MAP
0x0A	TXD0_MAP	0x1A	CLKO_MAP
0x0B	RXD0_MAP	0x1B	T4PWM0_MAP
0x0C	TXD1_MAP	0x1C	T4PWM1_MAP
0x0D	RXD1_MAP	0x1D	STADC_MAP
0x0E	SCL_MAP	0x1E	—
0x0F	SDA_MAP	0x1F	—

## 5 时钟系统和电源管理

### 5.1 概述

芯片拥有多种时钟源可供选择, 这样在应用中可以有多种选择, 使系统性能发挥到最佳, 并且功耗降到最低。共有 5 种系统时钟源可供选择, 包括: 内部高速振荡器( $F_{HIRC}$ )、内部低速振荡器( $F_{LIRC}$ )、外部高速晶振( $F_{XT}$ )、外部低速晶振( $F_{LP}$ )、外部时钟输入( $F_{EC}$ )。另外 RM1221A 还可以有双时钟源模式 (外部 32.768KHz 为 RTC 时钟、内部 RC 振荡器为系统时钟)。

芯片内嵌 2 个内部 RC 振荡器: 一个 128K/32KHz 低速、一个 16MHz 高速 RC 振荡器, 高速 16MHz 误差在出厂时校准到  $\pm 2\%$ 。

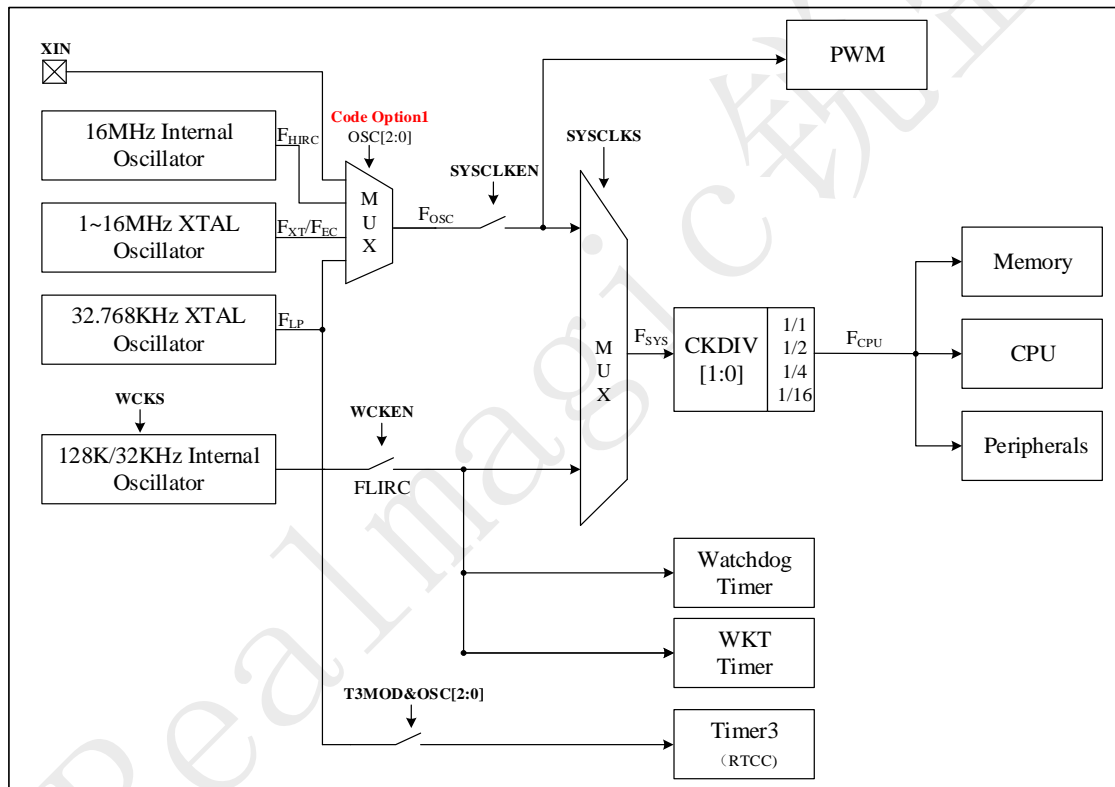


图5-1-1 RM1221A时钟结构图

RM1221A 内部高速 RC 时钟 HIRC 精度在晶圆测试时已进行校准。封装过程可能会导致 HIRC 频率漂移。烧录器软件可选择对 HIR 进行重新校准。HIRC 出厂校准值已存储到寄存器 RCTRIM 中, 用户可通过指令修改“RCTRIM”来微调 HIRC 频率(默认 16MHz), 微调 steps 是非线性的。

## 5.2 寄存器描述

### 5.2.1 系统时钟配置寄存器(CKCON) (受 WPT 保护)

寄存器	地址	R/W	描述	复位后的值
CKCON	0x8F	R/W	系统时钟配置寄存器	0011 0011

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TRIMCLKEN	—	SYSCLKEN	WCKEN	SYSCLKS	—	DEBUGEN	WCKS

位	描述	
[7]	TRIMCLKEN	<b>TRIM 时钟使能位</b> 0: 禁止 TRIM 时钟 1: 使能 TRIM 时钟
[6]	保留	保留
[3]	SYSCLKEN	<b>系统时钟使能位</b> 0: 禁止系统时钟 1: 使能系统时钟
[2]	WCKEN	<b>内部低速时钟 WCK 使能位</b> 0: 禁止内部低速时钟 WCK 1: 使能内部低速时钟 WCK
[3]	SYSCLKS	系统时钟选择位 0: 系统时钟为内部高速时钟 1: 系统时钟为内部低速时钟 WCK
[2]	保留	保留
[1]	DEBUGEN	调试模式使能位 0: 禁止调试模式 1: 使能调试模式
[0]	WCKS	<b>内部低速时钟 WCK 选择位</b> 0: 选择 WCK 为 32KHz 1: 选择 WCK 为 128KHz

**使用注意:** 内部低速时钟 32KHz 专为低功耗应用而设计, 如掉电模式下定时唤醒。可在程序中切换 32KHz/128KHz, 但此时 WDT/WKT 定时也会同时发生改变, 应用时需要十分注意这点, 且 32KHz 时钟唤醒 CPU 的时间也会相应变长。

### 5.2.2 电源管理寄存器(PCON)

寄存器	地址	R/W	描述	复位后的值
PCON	0x87	R/W	电源管理寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留						PD	IDLE

位	描述	
[7:2]	保留	保留
[1]	PD	<b>掉电模式</b> 设置该位使芯片进入掉电模式。 在此模式下，CPU 和外设时钟停止，程序计数器（PC）挂起。此时为最小功耗。 CPU 从掉电模式下唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从 ISR 返回后，设备继续执行让系统进入掉电模式的指令，使系统进入掉电模式。 注：如果 IDL 位和 PD 位同时置位，芯片进入掉电模式。从掉电模式退出后不会进入空闲模式。
[0]	IDLE	<b>空闲模式</b> 设置该位使芯片进入空闲模式。 在此模式下，CPU 时钟停止，且程序计数器（PC）挂起，但所有外设继续工作。 CPU 从空闲模式唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从 ISR 返回后，设备继续执行让系统进入空闲模式的指令，使系统进入空闲模式。

### 5.2.3 TRIM 控制寄存器(TRIMCON)

寄存器	地址	R/W	描述	复位后的值
TRIMCON	0xF070	R/W	TRIM 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TRIMCLKF	保留			SOC	MODE	NUM	TRIMCKS

位	描述	
[7]	TRIMCLKF	<b>校正完成标志位</b> 0: 正在校正 1: 此次校正完成 单次校正模式下，校正完置 1，由用户清零 连续校正模式下，校正完置 1；下次校正开始，自动清零
[6:0]	保留	保留

### 5.2.4 内部高速时钟 TRIM 低 8 位(RCTRIML)

寄存器	地址	R/W	描述	复位后的值
<b>RCTRIML</b>	0x9C	R/W	内部高速时钟 TRIM 低 8 位	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>RCTRIM[7:0]</b>							

位	描述	
[7:0]	<b>RCTRIM[7:0]</b>	内部高速时钟校正低 8 位

### 5.2.5 内部高速时钟 TRIM 高 2 位(RCTRIMH)

寄存器	地址	R/W	描述	复位后的值
<b>RCTRIMH</b>	0x9D	R/W	内部高速时钟 TRIM 高 2 位	00000010

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留						<b>RCTRIM[9:8]</b>	

位	描述	
[7:2]	保留	保留
[1:0]	<b>RCTRIM[9:8]</b>	内部高速时钟校正高 2 位

## 6 复位

RM1221A 有 6 种方式复位：上电复位、低电压复位、外部引脚复位（PD2）、硬件故障复位、看门狗复位和软件复位，前五种为硬件复位。

上述任意一种复位发生时，所有的系统寄存器将恢复默认状态，程序停止运行，同时程序计数器 PC 清零，复位结束后程序从复位向量 0000H 开始运行。

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。

RM1221A 的复位部分电路结构图如下：

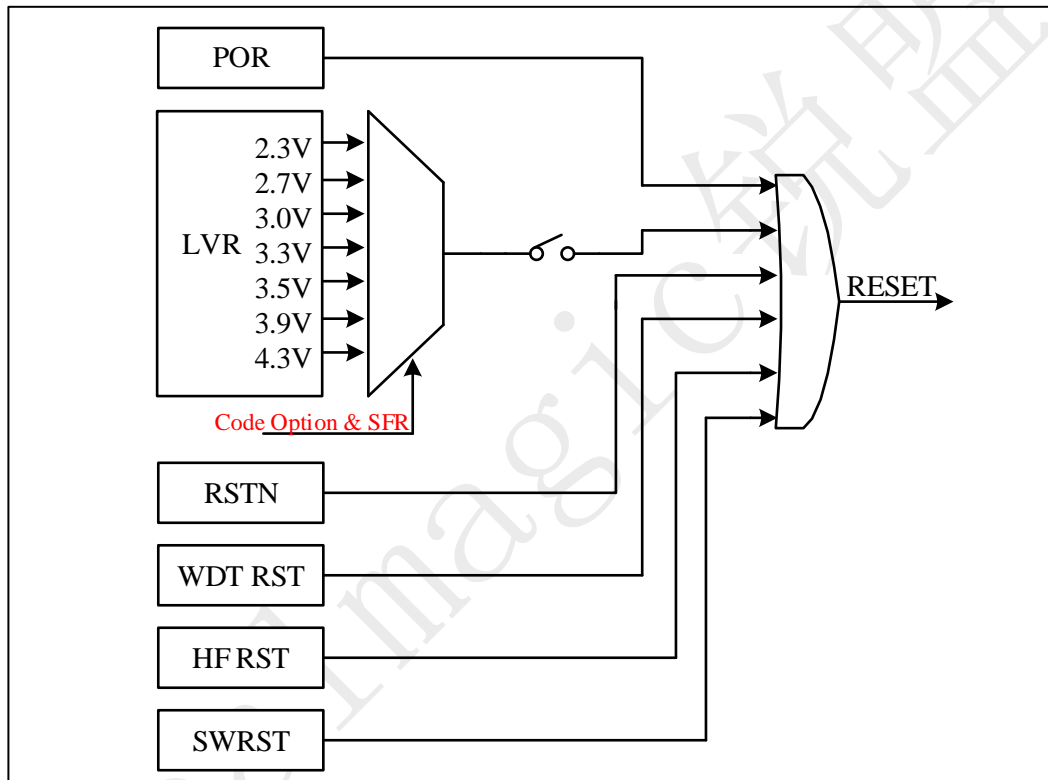
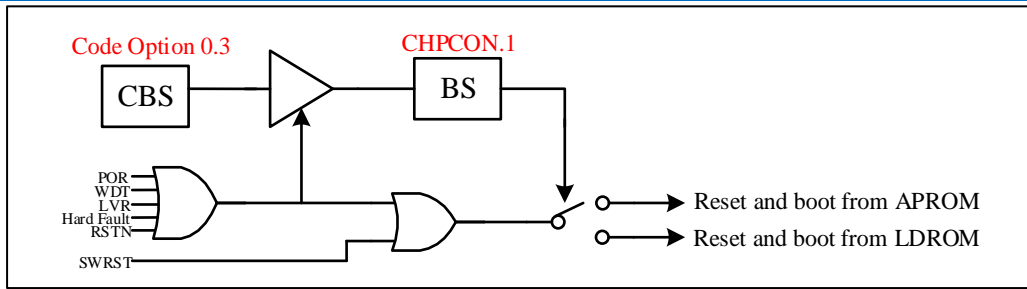


图 6-1 RM1221A 复位电路图

复位后的启动区域：

- 1、 外部 RSTN 复位、低电压复位 LVR、上电复位 POR、看门狗复位 WDT 这四种硬件复位后，芯片从用户 Code Option0 设定的启动区域（APROM/LDROM）启动。
- 2、 软件复位后，芯片从寄存器 CHIPCON.1（BS）设定的启动区域（APROM/LDROM）启动。


**图 6-2 RM1221A 复位后启动选择图**

## 6.1 上电复位

VDD 上升速度、芯片高速运行或要求一定模拟性能时，所需要的电压可能高于最低的 VDD，可以使用上电延时电路模块来延长启动时间，上电延时时间可根据需要烧录选择配置，以满足所有器件工作电压条件。

烧录配置值	上电延时时间
0	12ms
1	36ms

## 6.2 低电压复位

低电压复位电压有 8 档。低电压复位使能后，可选择不同档位复位电压：

烧录配置值	低电压复位档位
001	2.3V
010	2.7V
011	3.0V
100	3.3V
101	3.5V
110	3.9V
111	4.3V

LVR 可以配置为以下不同功能：

- 1、LVR 使能；
- 2、LVR 关闭
- 3、由软件控制使能或关闭 LVR（SLVREN）。

软件使能或关闭 LVR 需要在 [Code Option2.7\(LVRSOFEN\)](#)置 1，然后软件配置 [LVD 配置寄存器 LVDCON.6\(SLVREN\)](#)开关。

## 6.3 外部引脚复位

PD2 可烧录配置为外部引脚复位口，PD2 输入为低电平时芯片产生复位。在复位条件下，只要 RSTN 引脚电平从低到高，CPU 将退出复位状态，并从地址 0000H 处开始执行代码。

如果 CPU 在掉电模式下，外部 RSTN 引脚复位时，触发硬件复位的方法略有不同。因为掉电模式下系统时钟是停止的，复位信号将等待系统时钟恢复。在系统时钟稳定后，CPU 将进入复位状态，然后退出，并从地址 0000H 处开始执行程序。

RSTPINF(AUXR1.6)为复位标志位，用来标志发生了外部复位。当发生外部复位后，该位硬件置 1。除上电复位或外部复位引脚复位外，该位不会置 1，并通过软件清零。

## 6.4 硬件故障复位

程序计数器 PC 溢出 flash 地址空间，硬件故障将发生。硬件故障复位后辅助寄存器 1.HardF(AUXR1.5)被硬件置位，辅助寄存器 1.HardF(AUXR1.5)除了会被上电复位或硬件故障复位更改，不会被任何其他复位更改，这位能通过软件清零。

## 6.5 看门狗复位

看门狗定时器是一个自由运行的定时器，带可编程溢出时间间隔和专用内部时钟源。用户可以在任何时候清除看门狗定时器，使它重新开始计数。当选择的溢出时间间隔发生溢出后，看门狗定时器将直接复位系统。复位完成后，芯片从地址 0000H 开始运行。

如果看门狗定时器引起复位，看门狗定时器复位标志 WDTRF(WDCON0.7)将置位。除上电复位外该位保持不变，用户可以通过软件清 WDTRF。

看门狗时钟有两个频率可选 32KHz 和 128KHz，通过 [系统时钟配置寄存器 CKCON.1\(WCKS\)](#) 进行选择。需注意 32KHz 专为低功耗应用而设计，如掉电模式下定时唤醒。可在程序中切换 32KHz/128KHz，但此时 WDT/WKT 定时也会同时发生改变，应用时需要十分注意这点，且 32KHz 时钟唤醒 CPU 的时间也会相应变长。

看门狗复位相关寄存器详见 [章节 15 看门狗定时器 \(WDT\)](#)。

## 6.6 软件复位

RM1221A 提供软件复位功能，允许软件复位整个系统类似于外部复位，初使化 MCU 为复位状态。软件复位，在 ISP 动作结束后非常有用。例如，如果通过 ISP 启动代码更新用户代码完成，软件复位能重启 CPU 立即执行用户代码。写 1 到 SWRST(CHPCON.7)触发软件复位。注意，SWRST 时效访问控制受 WPT 保护，执行设置 SWRST 位是设备复位之前的最后指令。见下面例程。发生软件复位 SWRF(AUXR1.7)被硬件置 1，用户可通过读取该位，来确定复位发生原因。除上电复位或软件复位外，SWRF 不会被其它复位修改。通过软件清零。



## 6.7 寄存器描述

### 6.7.1 IAP 使能寄存器 (CHPCON) (受 WPT 保护)

寄存器	地址	R/W	描述	复位后的值
CHPCON	9FH	R/W	IAP 使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SWRST	保留					BS	IAPEN

位	描述	
[7]	SWRST	<b>软件复位</b> 对该位写 1, 芯片执行软件复位, 复位完成后该位自动清零。
[6:2]	保留	保留
[1]	BS	<b>启动选择位</b> 该位初始值为 option 的 CBS 值, 软件复位不会改变此值。 0: 由 APROM 启动 1: 由 LDROM 启动
[0]	IAPEN	<b>IAP 使能位</b> 0: 禁止 IAP 1: 使能 IAP

### 6.7.2 辅助功能寄存器 1 (AUXR1)

寄存器	地址	R/W	描述	复位后的值
AUXR1	9EH	R/W	辅助功能寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SWRF	RSTPINF	HARDF	IOVTH	T1M	T0M	—	DPS

位	描述	
[7]	SWRF	<b>软件复位标志位</b> 软件复位后, 该位将被硬件置 1, 建议复位发生后通过软件清零
[6]	RSTPINF	<b>外部复位标志位</b> 通过外部复位引脚 RSTN 复位后, 该位将被硬件置 1, 建议复位发生后通过软件清零
[5]	HARDF	<b>硬件故障复位标志</b> 一旦程序计数器 (PC) 溢出 FLASH 地址空间, MCU 将复位并且 HARDF 硬件置位。通过软件清零。

		注意：当 MCU 运行在调试模式并且 DEBUGEN=1，硬件故障复位被禁用，仅仅 HARDF (AUXR1.5)标志位置位。
[4]	<b>IOVTH</b>	<b>除 PD2 外引脚输入的电平阈值配置</b> 0: 输入阈值高/低电平为 0.7*VDD/0.3*VDD 1: 输入阈值高/低电平为 0.5*VDD/0.2*VDD 输入阈值配置对除 PD2 外所有 IO 同时生效。 P2 做复位脚时，输入阈值高/低电平强制为 0.7*VDD/0.3*VDD P2 做普通 IO 口时 0: 输入阈值高/低电平为 0.5*VDD/0.2*VDD 1: 输入阈值高/低电平为 0.7*VDD/0.3*VDD
[3]	<b>T1M</b>	<b>定时器 1 时钟选择</b> 0: 定时器 1 的时钟源选择为 1/12 系统时钟。与标准 8051 兼容 1: 定时器 1 的时钟源选择为系统时钟
[2]	<b>T0M</b>	<b>定时器 0 时钟选择</b> 0: 定时器 0 的时钟源选择为 1/12 系统时钟。与标准 8051 兼容 1: 定时器 0 的时钟源选择为系统时钟
[1]	<b>保留</b>	<b>保留</b>
[0]	<b>DPS</b>	<b>数据指针选择</b> 0: 数据指针 0 (DPTR) 默认有效。 1: 数据指针 1 (DPTR1) 有效。 当通过 DPS 切换当前有效 DPTR 后，之前有效 DPTR 寄存器内的值保持不变。

### 6.7.3 时控保护寄存器 (WPT)

寄存器	地址	R/W	描述	复位后的值
<b>WPT</b>	9FH	W	时控保护寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

位	描述
[7:0]	<b>WPT[7:0]</b> <b>时控保护</b> WPT 寄存器控制着对被保护的 SFRs 的访问权限。当需要写特殊规定的寄存器时，必须先对 WPT 寄存器写入 AAH，接着是 55H，当写完这两条后，才可以有 4 个周期的时间对具有时控保护的寄存器写入数据。

## 7 中断

### 7.1 概述

中断的目的是让软件处理非常规或异步的事件。RM1221A 有 4 个中断优先级、27 个中断源。每个中断源都有独立的优先级、标志位、中断向量和使能位。另外，中断可被全局使能或关闭。当中断发生时，CPU 将执行对应的中断服务子程序（ISR）。ISR 被分配到预先指定的地址如中断向量表 20-1.中断向量。如果中断使能，当中断发生时，CPU 将根据中断源跳转到相应的中断向量地址，执行此地址处的程序，保持中断服务状态直到执行中断服务程序 ISR。一旦 ISR 开始执行，仅能被更高优先级的中断抢占。ISR 通过指令 RETI 返回，该指令强迫 CPU 回到中断发生前所执行指令的下一条指令。

### 7.2 中断源及中断向量

中断号	中断入口地址(0x)	中断请求源	中断使能位	中断标志位
0	3	EXT0	IE[0]	TCON[1]
1	B	ET0	IE[1]	TCON[5]
2	13	EXT1	IE[2]	TCON[3]
3	1B	ET1	IE[3]	TCON[7]
4	23	EUART0	IE[4]	S0CON[1:0]
5	2B	ET2	EIE[7]	T2CON[7]
6	33	EI2C	EIE[0]	I2CCON[3]
7	3B	EGPIO	EIE[1]	PxINTRF
8	43	ELVD	IE[5]	LVDCON[5]
9	4B	ESPI	EIE[6]	SPIF
10	53	EWKT	EIE1[0]	WKCON[4]
11	5B	EADC	IE[6]	ADCCON0[7]
12	63	ECAP	EIE[2]	CAPCON0[2:0]
13	6B	EPWM	EIE[3]	PWMCON0[5]
14	73	ETK	EIE[5]	TKF[7:6]
15	7B	EUART1	EIE1[3]	S1CON[1:0]
16	83	ET3	EIE1[1]	T3CON[7]
17	8B	EATKMAX	EIE[4]	TKF[5]
18	93	EFB	EIE1[4]	FBD[7]
19	9B	ET4	EIE1[2]	T4CON[5]
20	A3	ET4PWM	EIE1[5]	T4CON[7: 6]
21	AB	RISE_RISE_INTEN <sub>x</sub> x=0~3	RISEINTEN[3:0]	RISEINTF[3:0]

22	B3	RISE_FALL_INTEN <sub>x</sub> x=0~3	RISEINTEN[7:4]	RISEINTF[7:4]
23	BB	FALL_RISE_INTEN <sub>x</sub> x=0~3	FALLINTEN[3:0]	FALLINTF[3:0]
24	C3	FALL_FALL_INTEN <sub>x</sub> x=0~3	FALLINTEN[7:4]	FALLINTF[7:4]
25	CB	EECKPD	EIE1[6]	AUXR1[4]
26	D3	ETRIMCLK	EIE1[7]	TRIMCON[7]

### 7.3 中断优先级

系统为所有中断源提供 4 种优先级：最高（3 级）、高（2 级）、低（1 级）、最低（0 级）。中断源可以单独设置各自的优先级位来配置其优先级。下表列举了 4 种优先级配置。相对来说，低优先级中断可以被高优先级中断打断，但不能被同等优先级或更低的优先级打断。默认优先级可以帮助中断控制器解决同等优先级同时请求中断的状况。

在多个中断时，遵循以下规则：

(1) 当一个低优先级中断正在运行，这时一个高优先级产生，该中断会被打断去执行高优先级中断。当高优先级中断执行完 RETI 后，低优先级中断恢复继续运行。当低优先级中断执行完 RETI 后，控制器把运行权利交还给主程序。

(2) 如果一个高优先级中断正在运行，不能被任何其他中断源打断——即使这高优先级中断，在默认优先级中比正在运行的中断优先级更高，也不能打断运行中的中断。

(3) 低优先级中断只有在其他中断没有执行的情况下才能被调用。然后同时，低优先级中断不能被另一个低优先级中断打断，即使这个低优先级中断，在默认优先级中比正在运行的中断优先级更高，也不能打断运行中的中断。

(4) 如果两个中断同时发生，优先级高的中断先执行。如果两个中断优先级相同，默认优先级高的中断先执行，这是符合默认优先级唯一的条件。

默认优先级如表 7-3 所示，总结了中断源、标志位、向量地址、使能位、优先级位和允许 CPU 可以从掉电模式中唤醒。

表 7-2. 中断优先线设置

中断优先级控制位		中断优先级
IP1	IP0	
0	0	等级0 (最低)
0	1	等级1
1	0	等级2
1	1	等级3(最高)

表 7-3. 各级中断源特性表

中断号	入口地址(0x)	中断请求源	中断使能位	中断标志位	中断标志清除方式	是否支持掉电模式唤醒
0	3	EXT0	IE[0]	TCON[1]	0	是

1	B	ET0	IE[1]	TCON[5]	1	否
2	13	EXT1	IE[2]	TCON[3]	2	是
3	1B	ET1	IE[3]	TCON[7]	3	否
4	23	EUART0	IE[4]	S0CON[1:0]	4	否
5	2B	ET2	EIE[7]	T2CON[7]	5	否
6	33	EI2C	EIE[0]	I2CCON[3]	6	否
7	3B	EGPIO	EIE[1]	PxINTRF	7	是
8	43	ELVD	IE[5]	LVDCON[5]	8	是
9	4B	ESPI	EIE[6]	SPIF	9	否
10	53	EWKT	EIE1[0]	WKCON[4]	10	是
11	5B	EADC	IE[6]	ADCCON0[7]	11	否
12	63	ECAP	EIE[2]	CAPCON0[2:0]	12	否
13	6B	EPWM	EIE[3]	PWMCON0[5]	13	否
14	73	ETK	EIE[5]	TKF[7:6]	14	否
15	7B	EUART1	EIE1[3]	S1CON[1:0]	15	否
16	83	ET3	EIE1[1]	T3CON[7]	16	否
17	8B	EATKMAX	EIE[4]	TKF[5]	17	是
18	93	EFB	EIE1[4]	FBD[7]	18	否
19	9B	ET4	EIE1[2]	T4CON[5]	19	否
20	A3	ET4PWM	EIE1[5]	T4CON[7: 6]	20	否
21	AB	RISE_RISE_INTENx x=0~3	RISEINTEN[3:0]	RISEINTF[3:0]	21	是
22	B3	RISE_FALL_INTENx x=0~3	RISEINTEN[7:4]	RISEINTF[7:4]	22	是
23	BB	FALL_RISE_INTENx x=0~3	FALLINTEN[3:0]	FALLINTF[3:0]	23	是
24	C3	FALL_FALL_INTENx x=0~3	FALLINTEN[7:4]	FALLINTF[7:4]	24	是
25	CB	EECKPD	EIE1[6]	AUXR1[4]	25	否
26	D3	ETRIMCLK	EIE1[7]	TRIMCON[7]	26	否

**IP1 中断优先级高位寄存器**

寄存器	地址	R/W	描述	复位后的值
IP1	B7H	R/W	中断优先级高位寄存器	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IP1[5:0]							

[5:0]	IP1[5:0]	6 组中断优先级配置高位
-------	----------	--------------

**IP0 中断优先级低位寄存器**

寄存器	地址	R/W	描述	复位后的值
IP0	B8H	R/W	中断优先级低位寄存器	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IP0[5:0]							

[5:0]	IP0[5:0]	6 组中断优先级配置低位
-------	----------	--------------

6 组的中断优先级可通过优先级控制寄存器 IP1 和 IP0 配置。若寄存器 IP1 和 IP0 均配置为全 0 时，各组与各组之间的优先级均相同，当有两组中断请求同时发生时，采用组 0 到组 5，优先级从最高（组 0）到最低（组 5）的方式处理。

每组内部的优先级相同，当组内有两个中断同时产生中断请求时，以查询的方式处理中断服务程序。查询的优先级顺序为左列（最高）到右列（最低），如下优先级结构表所示。

表 7-4. 中断优先级分组以及默认优先级结构表

组间 优先级	组号	配置	中断源				
			最高	组 0	IP1.0 IP0.0	03H	33H
...	组 1	IP1.1 IP0.1	0BH	3BH	8BH	CBH	4BH
...	组 2	IP1.2 IP0.2	13H	73H	93H	D3H	53H
...	组 3	IP1.3 IP0.3	1BH	7BH	9BH	DBH	5BH
...	组 4	IP1.4 IP0.4	23H	B3H	A3H	E3H	63H
最低	组 5	IP1.5 IP0.5	2BH	BBH	ABH	EBH	6BH
		组内 优先级	最高	...	...	...	最低

## 7.4 中断服务

中断标志在每个系统时钟周期都会被采样。在同一个周期内，被采样到的中断和优先级都会被解决。如果满足特定的条件硬件将执行内部产生的 LCALL 指令，目标地址是中断向量地址。能产生 LCALL 条件如下：

- (1) 没有相同或更高优先级中断服务程序在执行。
- (2) 当前查询中断标志周期正好是当前执行指令的最后一个周期。
- (3) 当前指令不是写任何中断使能位或优先级位且也不能是中断返回指令 RETI。

如果以上任何一个条件不满足，就不能产生 LCALL 指令。在每一个指令周期都会重新检测中断标志。当某个中断标志位被置起，但没有满足上述条件都不会被响应，即使后面满足上述条件，没有立即执行的中断仍然不会执行 LCALL 指令。这个中断标志生效，但没有进入中断服务程序，下一个指令周期需要重新检测中断标志。

处理器响应一个有效的中断，通过执行一个 LCALL 指令将程序转移到中断入口地址。对应的中断标志根据不同的中断源在执行中断服务程序时，可能被硬件清除，也可能不被清除。硬件 LCALL 与软件 LCALL 指令相同，执行 LCALL 指令，保存程序计数器 PC 内容到堆栈，但不保存程序状态字 PSW，PC 指针重新装载产生中断的中断向量地址，从向量地址继续执行程序直到执行 RETI 指令。在执行 RETI 指令时，处理器弹出堆栈，将栈顶内容加载到程序计数器 PC。用户必须注意堆栈的状态，如果堆栈的内容被修改，处理器不会被通知，将会从堆栈加载的地址继续执行。注：RET 指令与 RETI 指令表现相同，但它不会通知中断控制器中断服务已经完成，致使控制器认为中断服务仍在进行。使中断不可能再产生。

## 7.5 中断延迟

每一个中断源的响应时间取决于几个方面，如中断自身特点和指令的执行。在每个时钟周期，每一个中断标志和优先级都会被检测。如果有一个中断请求满足以上 3 个条件，硬件将自动产生 LCALL 指令，执行该指令需要 4 个机器周期。这样从中断标志置位，到执行中断服务程序最少需要 5 个机器周期。

如果三个条件不满足，很长的响应时间是可以预知的。如果高优先级和同等优先级中断正在执行，那么中断延迟时间很明显取决于正在执行的中断服务程序。最大的响应时间（如果没有其他中断正在执行或是也没有更高优先级中断产生）是执行 RETI 指令，然后下一条要执行的指令是最长的 6 个时钟周期的指令。从一个中断源被激活(没有检测到)，最长的反应时间是 16 个时钟周期。这些周期包括完成 RETI 指令的 5 个时钟周期、完成最长指令的 6 个时钟周期、侦测中断 1 个时钟周期和完成硬件 LCALL 跳转到中断地址的 4 个时钟周期。

因此一个简单的中断系统响应时间，总是大于 5 个时钟周期并且不超过 16 个时钟周期。

## 7.6 外部中断

外部中断 INT0 和 INT1 被作为中断源。它们可以根据 IT0(TCON.0)和 IT1(TCON.2)选择边沿触发，还是电平触发中断。检测 IE0 (TCON.1)和 IE1 (TCON.3)标志位判断是否产生中断。在边沿触发模式下，每个系统时钟周期都会采样 INT0 或 INT1 输入状态。如果在一个周期中采样是高电平，然后在下一个周期中为低电平，那么这个高到低的电平转换将会被检测到，并且置位中断请求标志 IE0 或 IE1。由于每个系统时钟周期都去采样外部中断，所以高电平或低电平至少保持一个系统时钟周期。当中断服务程序被执行时，IE0 和 IE1 会被硬件自动清除。如果选择电平触发模式，那么必须保持引脚为低电平，直到进入中断服务，在进入中断服务程序时 IE0 和 IE1 不会被硬件清除。在电平触发模式下，IE0 和 IE1 标志位与 INT0 和 INT1 引脚逻辑电平值相反。当中断服务程序结束后引脚依然保持低电平，处理器会响应另一个来自同一中断源的中断。INT0 和 INT1 均支持将芯片从掉电模式唤醒。

### TCON – 定时器 0 和 1 控制位 (可位寻址)

寄存器	地址	R/W	描述	复位后的值
TCON	88H	R/W	定时器 0 和 1 控制位	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

[7]	TF1	定时器 1 溢出标志。 在定时器 1 溢出时该位置 1。当程序响应定时器 1 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0。
[6]	TR1	定时器 1 启动控制。 0 = 定时器 1 中止。清该位将中止定时器 1 并且当前计数值将保存到 TH1 和 TL1 中 1 = 使能定时器 1。
[5]	TF0	定时器 0 溢出标志。 在定时器 0 溢出时该位置 1。当程序响应定时器 0 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0。
[4]	TR0	定时器 0 启动控制。 0 = 定时器 0 中止。清该位将中止定时器 0 并且当前计数值将保存到 TH0 和 TL0 中 1 = 使能定时器 0。
[3]	IE1	外部中断 1 边沿标志。 当检测到边沿/电平类型时，该标志由硬件置位。 如果 IT1 = 1 (下降沿触发)，该标志将保持置 1 直到软件清零或在外部中断 1 服务程序中硬件清零。 如果 IT1 = 0 (低电平触发)，该标志是 $\overline{INT1}$ 输入信号逻辑电平的反转。软件不可控制。
[2]	IT1	外部中断 1 类型选择。 该位选择 $\overline{INT1}$ 的中断触发类型是下降沿还是低电平。 0 = $\overline{INT1}$ 为低电平触发。



		1 = $\overline{\text{INT1}}$ 为下降沿触发。
[1]	IE0	<p>外部中断 0 边沿标志。</p> <p>当检测到边沿/电平类型时，该标志由硬件置位。</p> <p>如果 <math>\text{IT0} = 1</math>（下降沿触发），该位将保持置 1 直到软件清零或在外部中断 0 服务程序中硬件清零。</p> <p>如果 <math>\text{IT0} = 0</math>（低电平触发），该标志是 <math>\overline{\text{INT0}}</math> 输入信号逻辑电平的反转。软件不可控制。</p>
[0]	IT0	<p>外部中断 0 类型选择。</p> <p>该位选择 <math>\overline{\text{INT0}}</math> 的中断触发类型是下降沿还是低电平。</p> <p>0 = <math>\overline{\text{INT0}}</math> 为低电平触发。</p> <p>1 = <math>\overline{\text{INT0}}</math> 为下降沿触发。</p>

## 7.7 寄存器列表

寄存器	地址	R/W	描述	复位值
EIE	0x95	R/W	扩展中断使能寄存器	00000000
EIE1	0x96	R/W	扩展中断使能寄存器 1	00000000
IE	0xA8	R/W	中断使能寄存器	00000000
IPO	0xB8	R/W	中断优先级配置寄存器 0	00000000
IP1	0xB9	R/W	中断优先级配置寄存器 1	00000000

## 7.8 寄存器描述

### 7.8.1 扩展中断使能寄存器 (EIE)

寄存器	地址	R/W	描述	复位后的值
EIE	0x95	R/W	扩展中断使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ET2	ESPI	ETK	EATKMAX	EPWM	ECAP	EGPIO	EI2C

位	描述
[7]	<b>ET2</b> <b>定时器 2 中断使能位</b> 0: 禁止定时器 2 中断 1: 使能定时器 2 中断
[6]	<b>ESPI</b> <b>SPI 中断使能位</b> 0: 禁止 SPI 中断 1: 使能 SPI 中断
[5]	<b>ETK</b> <b>触摸按键中断使能位</b> 0: 禁止触摸按键中断 1: 使能触摸按键中断
[4]	<b>EATKMAX</b> <b>低电压检测中断使能位</b> 0: 禁止触摸按键硬件自动扫描溢出中断 1: 使能触摸按键硬件自动扫描溢出中断
[3]	<b>EPWM</b> <b>PWM 中断使能位</b> 0: 禁止 PWM 中断 1: 使能 PWM 中断
[2]	<b>ECAP</b> <b>捕获中断使能位</b> 0: 禁止捕获中断 1: 使能捕获中断
[1]	<b>EGPIO</b> <b>GPIO 管脚中断使能位</b> 0: 禁止 GPIO 管脚中断

		1: 使能 GPIO 管脚中断
[0]	<b>EI2C</b>	<b>I2C 中断使能位</b> 0: 禁止 I2C 中断 1: 使能 I2C 中断

## 7.8.2 扩展中断使能寄存器 (EIE1)

寄存器	地址	R/W	描述	复位后的值
<b>EIE1</b>	0x96	R/W	扩展中断使能寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>ETRIMCLK</b>	<b>EECKPD</b>	<b>ET4PWM</b>	<b>EFB</b>	<b>EUART1</b>	<b>ET4</b>	<b>ET3</b>	<b>EWKT</b>

位	描述	
[7]	<b>ETRIMCLK</b>	<b>TRIM 时钟中断</b> 0: 禁止 TRIM 时钟中断 1: 使能 TRIM 时钟中断
[6]	<b>EECKPD</b>	<b>外部时钟停止中断</b> 0: 禁止外部时钟停止中断 1: 使能外部时钟停止中断
[5]	<b>ET4PWM</b>	<b>定时器 4 PWM 中断使能位</b> 0: 禁止定时器 4 PWM 中断 1: 使能定时器 4 PWM 中断
[4]	<b>EFB</b>	<b>刹车中断使能位</b> 0: 禁止刹车中断 1: 使能刹车中断
[3]	<b>EUART1</b>	<b>串口 1 中断使能位</b> 0: 禁止串口 1 中断 1: 使能串口 1 中断
[2]	<b>ET4</b>	<b>定时器 4 溢出中断使能位</b> 0: 禁止定时器 4 溢出中断 1: 使能定时器 4 溢出中断
[1]	<b>ET3</b>	<b>定时器 3 溢出中断使能位</b> 0: 禁止定时器 3 溢出中断 1: 使能定时器 3 溢出中断
[0]	<b>EWKT</b>	<b>自唤醒溢出中断使能位</b> 0: 禁止自唤醒溢出中断 1: 使能自唤醒溢出中断

### 7.8.3 中断使能寄存器 (IE)

寄存器	地址	R/W	描述	复位后的值
IE	0xA8	R/W	中断使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
EA	EADC	ELVD	EUART0	ET1	EXT1	ET0	EXT0

位	描述
[7]	<b>EA</b> <b>总中断使能位</b> 0: 禁止总中断 1: 使能总中断
[6]	<b>EADC</b> <b>ADC 中断使能位</b> 0: 禁止 ADC 中断 1: 使能 ADC 中断
[5]	<b>ELVD</b> <b>低电压检测中断使能位</b> 0: 禁止低电压检测中断 1: 使能低电压检测中断
[4]	<b>EUART0</b> <b>串口 0 中断使能位</b> 0: 禁止串口 0 中断 1: 使能串口 0 中断
[3]	<b>ET1</b> <b>定时器 1 溢出中断使能位</b> 0: 禁止定时器 1 溢出中断 1: 使能定时器 1 溢出中断
[2]	<b>EXT1</b> <b>外部中断 1 使能位</b> 0: 禁止外部中断 1 1: 使能外部中断 1
[1]	<b>ET0</b> <b>定时器 0 溢出中断使能位</b> 0: 禁止定时器 0 溢出中断 1: 使能定时器 0 溢出中断
[0]	<b>EXT0</b> <b>外部中断 0 使能位</b> 0: 禁止外部中断 0 1: 使能外部中断 0

### 7.8.4 中断优先级配置寄存器 0 (IP0)

寄存器	地址	R/W	描述	复位后的值
IP0	0xB8	R/W	中断优先级配置寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

<b>IP0[7:0]</b>
-----------------

位	描述	
[7:0]	<b>IP0[7:0]</b>	中断优先级配置寄存器

### 7.8.5 中断优先级配置寄存器 1 (IP1)

寄存器	地址	R/W	描述	复位后的值
<b>IP1</b>	0xB9	R/W	中断优先级配置寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>IP1[7:0]</b>							

位	描述	
[7:0]	<b>IP1[7:0]</b>	中断优先级配置寄存器

## 8 低电压检测

低电压检测是用于监测 VDD 电压，当使能低电压检测 (LV DEN=1) 时，如果 VDD 电压低于设定的电压值，LVDCON 寄存器的 LVDOF 位将置 1。如果此时使能低电压检测中断 (LVDIEN=1)，那么会进中断，中断标志位 LV DIF 需软件清 0。

### 8.1 寄存器描述

#### 8.1.1 LVD 配置寄存器(LVDCON) (受 WPT 保护)

寄存器	地址	R/W	描述	复位后的值
LVDCON	0xF03A	R/W	低电压检测配置寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
LVDOF	SLVREN	LV DIF	LV DSEL[3:0]				LV DEN

位	描述
[7]	LVDOF LVD 状态标志位，1 表示芯片为低电压状态，只读
[6]	SLVREN 软件开启 LVR 使能位 0: 禁止软件开启 LVR 1: 使能软件开启 LVR
[5]	LV DIF LVD 中断标志位，软件清 0
[4:1]	LV DSEL[3:0] LVD 电压阈值选择 0000: 无效 0001: 无效 0010: 无效 0011: 无效 0100: 2.3V 0101: 2.4V 0110: 2.5V 0111: 2.7V 1000: 3.0V 1001: 3.3V 1010: 3.5V 1011: 3.7V 1100: 3.9V 1101: 4.1V 1110: 4.3V 1111: 4.5V

[0]	<b>LV DEN</b>	低电压检测使能位 0: 禁止低电压检测 1: 使能低电压检测
-----	---------------	--------------------------------------

## 9 GPIO

### 9.1 概述

RM1221A 独立方向控制的 26 个 I/O 引脚。所有端口独立的可编程弱上/下拉，所有口均可设定为伪开漏输出。所有端口可灵活配置驱动电流，能直接驱动 LED。所有端口可配置 IO 翻转中断，可唤醒 STOP 和 IDLE 低功耗模式。

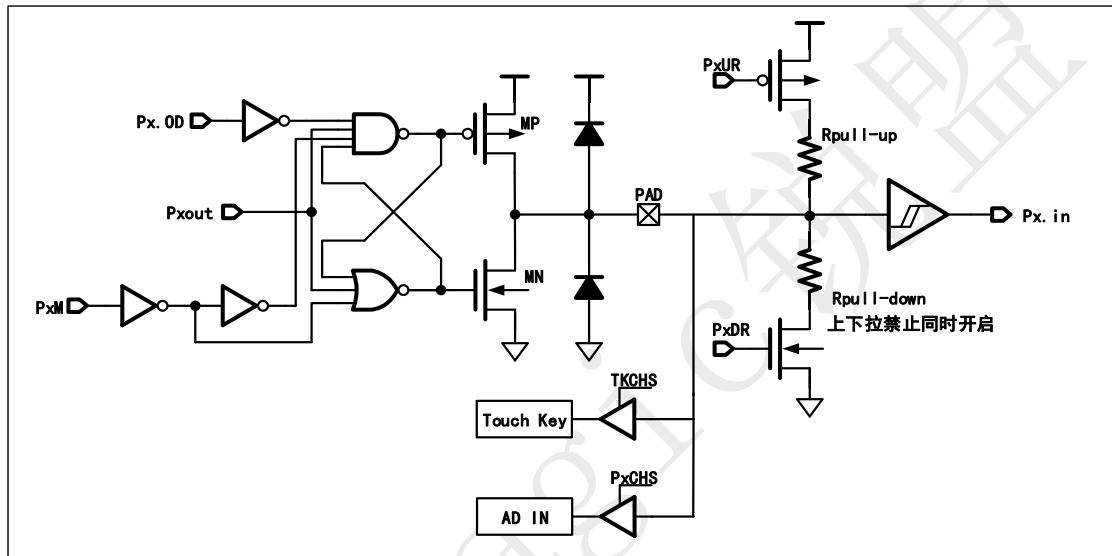


图 9-1-1 RM1221A IO 结构图

### 9.2 寄存器列表

寄存器	地址	R/W	描述	复位值
PORTA	0x80	R/W	PA 数据锁存寄存器	00000000
PAM	0x91	R/W	PA 控制寄存器	00000000
PAUR	0xF010	R/W	PA 上拉控制寄存器	00000000
PADR	0xF011	R/W	PA 下拉控制寄存器	00000000
PAOD	0xF012	R/W	PA 开漏功能控制寄存器	00000000
PORTB	0x90	R/W	PA 数据锁存寄存器	00000000
PBM	0x92	R/W	PA 控制寄存器	00000000
PBUR	0xF014	R/W	PA 上拉控制寄存器	00000000
PBDR	0xF015	R/W	PA 下拉控制寄存器	00000000
PBOD	0xF016	R/W	PA 开漏功能控制寄存器	00000000
PORTC	0xA0	R/W	PA 数据锁存寄存器	00000000
PCM	0x93	R/W	PA 控制寄存器	00000000



PCUR	0xF018	R/W	PA 上拉控制寄存器	00000000
PCDR	0xF019	R/W	PA 下拉控制寄存器	00000000
PCOD	0xF01A	R/W	PA 开漏功能控制寄存器	00000000
PORTD	0xB0	R/W	PA 数据锁存寄存器	00000000
PDM	0x94	R/W	PA 控制寄存器	00000000
PDUR	0xF01C	R/W	PA 上拉控制寄存器	00000000
PDDR	0xF01D	R/W	PA 下拉控制寄存器	00000000
PDOD	0xF01E	R/W	PA 开漏功能控制寄存器	00000000
PAIO	0xF008	R/W	PA 驱动电流配置寄存器	00000000
PBIO	0xF009	R/W	PB 驱动电流配置寄存器	00000000
PCIO	0xF00A	R/W	PC 驱动电流配置寄存器	00000000
PDIO	0xF00B	R/W	PD 驱动电流配置寄存器	00000000
PAINTR	0xF000	R/W	PA 翻转中断使能寄存器	00000000
PBINTR	0xF001	R/W	PB 翻转中断使能寄存器	00000000
PCINTR	0xF002	R/W	PC 翻转中断使能寄存器	00000000
PDINTR	0xF003	R/W	PD 翻转中断使能寄存器	00000000
PAINTRF	0xA1	R/W	PA 翻转中断标志寄存器	00000000
PBINTRF	0xA2	R/W	PB 翻转中断标志寄存器	00000000
PCINTRF	0xA3	R/W	PC 翻转中断标志寄存器	00000000
PDINTRF	0xA4	R/W	PD 翻转中断标志寄存器	00000000

### 9.3 寄存器描述

#### 9.3.1 PA/PB/PC/PD 端口数据锁存寄存器 (PORTA / PORTB / PORTC / PORTD)

寄存器	地址	R/W	描述	复位后的值
PORTA	0x80	R/W	PA 端口数据锁存寄存器	00000000
PORTB	0x90	R/W	PB 端口数据锁存寄存器	00000000
PORTC	0xA0	R/W	PC 端口数据锁存寄存器	00000000
PORTD	0xB0	R/W	PD 端口数据锁存寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PORTx[7:0] x=A,B,C,D							

位	描述	
[7:0]	PORTx x=A,B,C,D	Px 输入输出数据

### 9.3.2 PA/PB/PC/PD 端口控制寄存器 (PAM/PBM/PCM/PDM)

寄存器	地址	R/W	描述	复位后的值
<b>PAM</b>	0x91	R/W	PA 端口控制寄存器	00000000
<b>PBM</b>	0x92	R/W	PB 端口控制寄存器	00000000
<b>PCM</b>	0x93	R/W	PC 端口控制寄存器	00000000
<b>PDM</b>	0x94	R/W	PD 端口控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PxM[7:0] x=A,B,C,D							

位	描述	
[7:0]	<b>PxM</b> x=A,B,C,D	Px 输入输出控制位 0: 输入 1: 输出

### 9.3.3 PA/PB/PC/PD 上拉控制寄存器(PAUR/PBUR/PCUR/PDUR)

寄存器	地址	R/W	描述	复位后的值
<b>PAUR</b>	0xF010	R/W	PA 上拉控制寄存器	00000000
<b>PBUR</b>	0xF014	R/W	PB 上拉控制寄存器	00000000
<b>PCUR</b>	0xF018	R/W	PC 上拉控制寄存器	00000000
<b>PDUR</b>	0xF01C	R/W	PD 上拉控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PxUR[7:0] x=A,B,C,D							

位	描述	
[7:0]	<b>PxUR</b> x=A,B,C,D	Px 上拉电阻控制位 0: 禁止上拉电阻 1: 使能上拉电阻

### 9.3.4 PA/PB/PC/PD 下拉控制寄存器(PADR/PBDR/PCDR/PDDR)

寄存器	地址	R/W	描述	复位后的值
<b>PADR</b>	0xF011	R/W	PA 下拉控制寄存器	00000000
<b>PBDR</b>	0xF015	R/W	PB 下拉控制寄存器	00000000

<b>PCDR</b>	0xF019	R/W	PC 下拉控制寄存器	00000000
<b>PDDR</b>	0xF01D	R/W	PD 下拉控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PADR[7:0]</b> x=A,B,C,D							

位	描述	
[7:0]	<b>PxDR</b> x=A,B,C,D	Px 下拉电阻控制位 0: 禁止下拉电阻 1: 使能下拉电阻

### 9.3.5 PA/PB/PC/PD 开漏功能控制寄存器 (PAOD /PLVD /PCOD /PDOD)

寄存器	地址	R/W	描述	复位后的值
<b>PAOD</b>	0xF012	R/W	PA 开漏功能控制寄存器	00000000
<b>PBOD</b>	0xF016	R/W	PB 开漏功能控制寄存器	00000000
<b>PCOD</b>	0xF01A	R/W	PC 开漏功能控制寄存器	00000000
<b>PDOD</b>	0xF01E	R/W	PD 开漏功能控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PxOD[7:0]</b> x=A,B,C,D							

位	描述	
[7:0]	<b>PxOD</b> x=A,B,C,D	Px 开漏功能控制位 0: 禁止开漏 1: 使能开漏

### 9.3.6 驱动电流配置寄存器 (PAIO/ PBIO/ PCIO/ PDIO)

寄存器	地址	R/W	描述	复位后的值
<b>PAIO</b>	0xF008	R/W	PA 驱动电流配置寄存器	00000000
<b>PBIO</b>	0xF009	R/W	PB 驱动电流配置寄存器	00000000
<b>PCIO</b>	0xF00A	R/W	PC 驱动电流配置寄存器	00000000
<b>PDIO</b>	0xF00B	R/W	PD 驱动电流配置寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PxIOH1[1:0]		PxIOH0[1:0]		PxIOL1[1:0]		PxIOL0[1:0]	

位	描述	
[7:6]	<b>PxIOH1[1:0]</b> x=A,B,C,D	Px IOH 高 4 位驱动电流配置位 00: 16mA 01: 8mA 10: 4mA 11: 2mA
[5:4]	<b>PxIOH0[1:0]</b> x=A,B,C,D	Px IOH 低 4 位驱动电流配置位 00: 16mA 01: 8mA 10: 4mA 11: 2mA
[3:2]	<b>PxIOL1[1:0]</b> x=A,B,C,D	Px IOL 高 4 位驱动电流配置位 00: 80mA 01: 40mA 10: 20mA 11: 10mA
[1:0]	<b>PxIOL0[1:0]</b> x=A,B,C,D	Px IOL 低 4 位驱动电流配置位 00: 80mA 01: 40mA 10: 20mA 11: 10mA

### 9.3.7 PA/PB/PC/PD 翻转中断使能寄存器 (PAINTR/ PBINTR/ PCINTR/ PDINTR)

寄存器	地址	R/W	描述	复位后的值
<b>PAINTR</b>	0xF000	R/W	PA 翻转中断使能寄存器	00000000
<b>PBINTR</b>	0xF001	R/W	PB 翻转中断使能寄存器	00000000
<b>PCINTR</b>	0xF002	R/W	PC 翻转中断使能寄存器	00000000
<b>PDINTR</b>	0xF003	R/W	PD 翻转中断使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PxINTR[7:0]</b> x=A,B,C,D							

位	描述	
[7:0]	<b>PxINTR</b>	Px 翻转中断使能位

<b>x=A,B,C,D</b>	0: 禁止翻转中断 1: 使能翻转中断
------------------	------------------------

### 9.3.8 PA/PB/PC/PD 翻转中断标志寄存器 (PAINTRF/ PBINTRF/ PCINTRF/ PDINTRF)

寄存器	地址	R/W	描述	复位后的值
<b>PAINTRF</b>	0xA1	R/W	PA 翻转中断标志寄存器	--000000
<b>PBINTRF</b>	0xA2	R/W	PB 翻转中断标志寄存器	--000000
<b>PCINTRF</b>	0xA3	R/W	PC 翻转中断标志寄存器	--000000
<b>PDINTRF</b>	0xA4	R/W	PD 翻转中断标志寄存器	--000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PxINTRF[7:0]</b> <b>x=A,B,C,D</b>							

位	描述
[7:0]	<b>PxINTRF</b> <b>x=A,B,C,D</b> Px 翻转中断标志位, 1 表示产生翻转中断

## 9.4 数字外设功能引脚全映射控制

大多数外设功能口可以映射到任意 I/O 口上, 但 ADC 输入、触摸按键输入、外部中断 (INT0/INT1)、IO 唤醒功能口、晶振脚等除外, 且外部硬件复位 RSTN 也不能映射。

### 9.4.1 外设功能引脚全映射控制寄存器

扩展 SFR 地址	扩展 SFR 名称	扩展 SFR 地址	扩展 SFR 名称	扩展 SFR 地址	扩展 SFR 名称
FF80	PWM0_MAP	FF8A	TXD0_MAP	FF94	EXT0_MAP
FF81	PWM1_MAP	FF8B	RXD0_MAP	FF95	EXT1_MAP
FF82	PWM2_MAP	FF8C	TXD1_MAP	FF96	T0CNT_MAP
FF83	PWM3_MAP	FF8D	RXD1_MAP	FF97	T0OUT_MAP
FF84	PWM4_MAP	FF8E	SCL_MAP	FF98	T1CNT_MAP
FF85	PWM5_MAP	FF8F	SDA_MAP	FF99	T1OUT_MAP
FF86	PWM_FB_MAP	FF90	SCK_MAP	FF9A	CLKO_MAP
FF87	CAP0_MAP	FF91	MOSI_MAP	FF9B	T4PWM0_MAP
FF88	CAP1_MAP	FF92	MISO_MAP	FF9C	T4PWM1_MAP

FF89	CAP2_MAP	FF93	NSS_MAP	FF9D	STADC_MAP
------	----------	------	---------	------	-----------

注：以上 SFR 为外部扩展 XSFR，采用 MOVX 来进行读写

位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W
复位值	1	0	0	0	0	0	0	0
位符号	Map_disable	-	FPORT[1:0]		-	FPIN[2:0]		

位	描述	
[7]	Map_disable	映射 GPIO 端口选择位 0: 上述映射 GPIO 功能开启, IO 为外设映射功能, GPIO 不能使用。 1: 上述映射 GPIO 功能关闭, IO 为 GPIO 功能。
[6]	保留	保留
[5:4]	FPORT[1:0]	映射 GPIO 端口选择位 00: PA 01: PB 10: PC 11: PD
[3]	保留	保留
[2:0]	FPIN[2:0]	映射管脚选择位 FPIN[2:0] = x(x = 0...7), 表示选择对应端口名的 x(x = 0...7)脚

注：1、系统将禁止其多对一映射，如多个功能选择在同一端口输出，即会按优先级选择选择相应的功能。

2、如果需要取消某功能映射，将相应寄存器恢复为默认值 0x80 即可。

上面寄存器的复位值为 0x80，这样复位后 IO 都为 GPIO，用户在使用外设功能脚之前必须先配置上面的寄存器，否则外设功能将无法使用。

使用举例：

将 UART 的 TXD 和 RXD 分别映射到 PA1 和 PA2 上，用户在启动 UART 之前应该配置下面命令：

```
TXD_MAP = 0x01; //TXD-->PA1
```

```
RXD_MAP = 0x02; //RXD-->PA2
```

如果用户在下一轮的设计时，需要重新将 UART 的 TXD 和 RXD 分别映射到 PC2 和 PC3 上，那用户需要进行如下的配置：

```
TXD_MAP = 0x22; //TXD-->PC2
```

```
RXD_MAP = 0x23; //RXD-->PC3
```

多个输出映射到一个端口上时，只能有一个输出有效，下面是默认的优先级：

优先级顺序	扩展 SFR 地址	复用端口功能
1	FF80	PWM0
2	FF81	PWM1
3	FF82	PWM2

4	FF83	PWM3
5	FF84	PWM4
6	FF85	PWM5
7	FF86	PWM_FB
8	FF87	CAP0
9	FF88	CAP1
10	FF89	CAP2
11	FF8A	TXD0
12	FF8B	RXD0
13	FF8C	TXD1
14	FF8D	RXD1
15	FF8E	SCL
16	FF8F	SDA
17	FF90	SCK
18	FF91	MOSI
19	FF92	MISO
20	FF93	NSS
21	FF94	EXT0
22	FF95	EXT1
23	FF96	TOCNT
24	FF97	T0OUT
25	FF98	T1CNT
26	FF99	T1OUT
27	FF9A	CLKO
28	FF9B	T4PWM0
29	FF9C	T4PWM1
30	FF9D	STADC

比如：T0OUT\_MAP 配置为 0x01 选择 PA.1 口作为 T0 的输出口，CLKO\_MAP 也配置为 0x01，这个时候硬件会按上面的优先级，PA.1 将配置为 T1 的输出口，而 CLKO\_MAP 的配置无效。

输入可以配置为多个功能从一个 PAD 引脚进入，比如：

T0\_MAP 配置为 0x23，则选择 PC.3 作为 T0 的输入口，T1\_MAP 也配置为 0x23，这样从 PC.3 端口进入的信号同时作用于 T0 和 T1。

将 TXD 和 RXD 都配置到一个端口上时，并且此端口设置为输出，则 TXD 和 RXD 将内部连接起来。

在输入时，无论端口是什么功能，读端口数据寄存器都读芯片引脚上的值。

当所有的端口映射控制寄存器的最高位 BIT7 为 1 时，此时即使配置了其他位，相应端口仍然为普通 IO 口，外设映射功能只有将此寄存器 BIT7 清零后才有效。

## 10 定时/计数器 0 和 1

### 10.1 概述

定时器/计数器 0 和 1 是 2 个 16 位定时器/计数器。每个都是由两个 8 位的寄存器组成的 16 位计数寄存器。对于定时器/计数器 0，高 8 位寄存器是 TH0、低 8 位寄存器是 TL0。同样定时器/计数器 1 也有两个 8 位寄存器，TH1 和 TL1。TCON 和 TMOD 可以配置定时器/计数器 0 和 1 的工作模式。

通过 TMOD 中的 C/ $\bar{T}$  位来选择定时器或计数器功能。每个定时器/计数器都有选择位，TMOD 的 BIT2 选择定时器/计数器 0 功能，TMOD 的 BIT6 选择定时器/计数器 1 功能。

将它们设置为定时器后，定时器将对系统时钟周期计数。定时器 0 通过设置 T0M(AUXR1.2) 位，定时器 1 通过设置 T1M(AUXR1.3 位，来选择定时器时钟是系统时钟 (FSYS) 的 12 分频或直接是系统时钟。在计数器模式下，每当检测到外部输入脚 T0 上的下降沿，计数寄存器的内容就会加一。如果在一个时钟周期采样到高电平，在下一个时钟周期采样到低电平，那么 T0 或 T1 引脚就会确认为一个由高到低的跳变。

当有定时器溢出发生，定时器 0 和 1 能配置引脚 T0/T1 自动翻转输出。这个功能通过设 P2S 寄存器的 T0OE 和 T1OE 来设置，分别对应于定时器 0 和定时器 1。当打开这个功能，输出端口在第一个定时溢出之前输出逻辑 1。为确保此模式功能，C/ $\bar{T}$  位应该被清除并且选择系统时钟作为定时器的时钟源。

注意：TH0(TH1) 和 TL0(TL1) 是独立分开访问。需要特别注意，在模式 0 或模式 1 下时，当读/写 TH0(TH1) 和 TL0(TL1) 之前，必须清除 TR0(TR1) 来停止计时。否则将产生不可预料的结果。

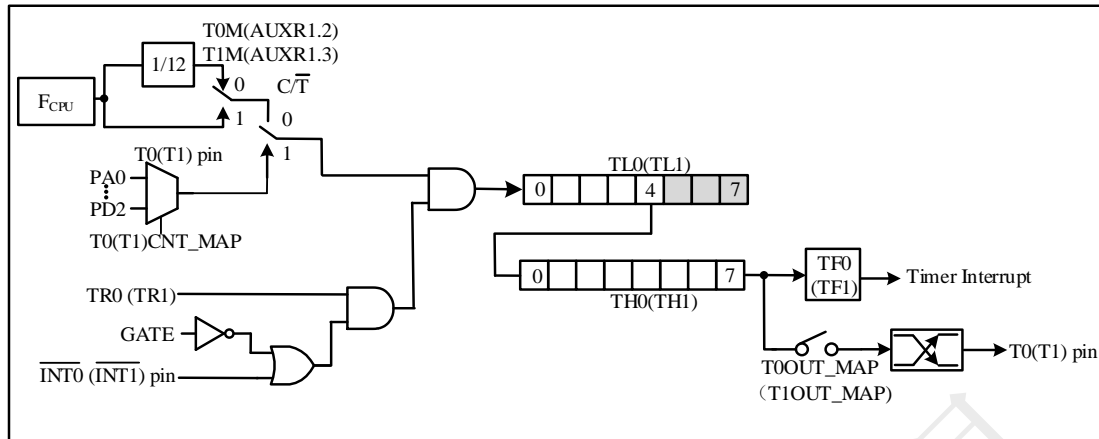
### 10.2 功能描述

#### 10.2.1 模式 0 (13 位定时器)

在模式 0，定时器/计数器是 13 位的计数器。13 位的计数器由 TH0(TH1) 和 TL0(TL1) 的低五位组成。TL0(TL1) 的高三位被忽略。当 TR0(TR1) 置位且 GATE 是 0 或 INT0(INT1) 是 1 时，定时器/计数器使能。

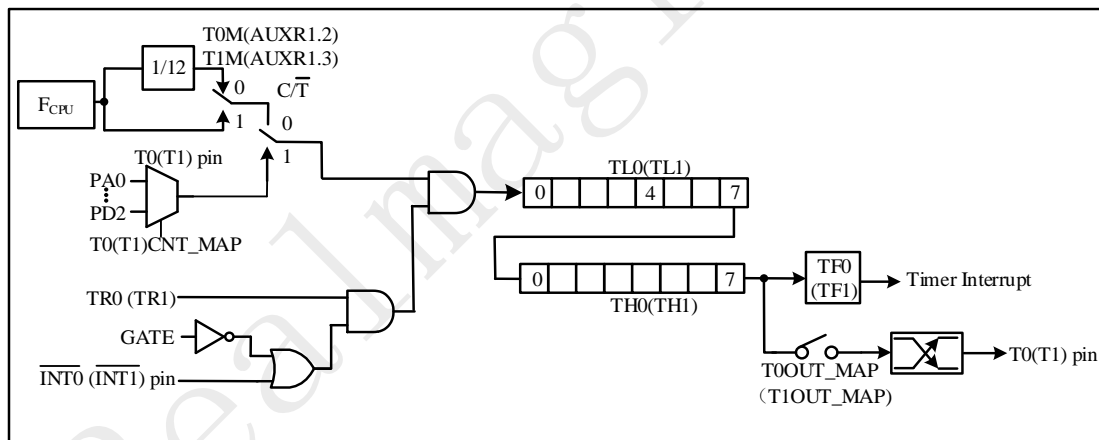
GATE 设置为 1 可以通过定时器来计算外部输入引脚  $\overline{INT0}$  ( $\overline{INT1}$ ) 上输入脉冲的宽度。当 13 位的定时器计数值从 1FFFH 变为 0000H 后，定时器溢出标志 TF0(TF1) 置位，如果中断打开，此时会产生一个定时器中断。




**图 10-2-1 定时器/计数器 0 和 1 的模式 0**

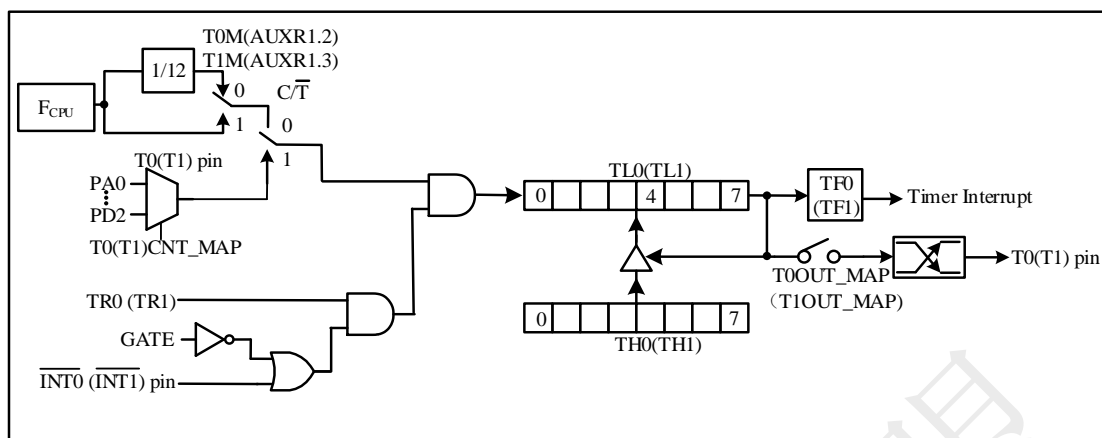
## 10.2.2 模式 1（16 位定时器）

模式 1 与模式 0 非常相似，只是模式 1 下定时器/计数器为 16 位的，就是说是用 THx 和 TLx 的全部 16 位用来计数。当计数值由 FFFFH 向 0000H 翻转后，定时器相应的溢出标志 TF0（TF1）置 1，如果中断使能则将产生中断。


**图 10-2-2 定时器/计数器 0 和 1 的模式 1**

## 10.2.3 模式 2（8 位自动重载定时器）

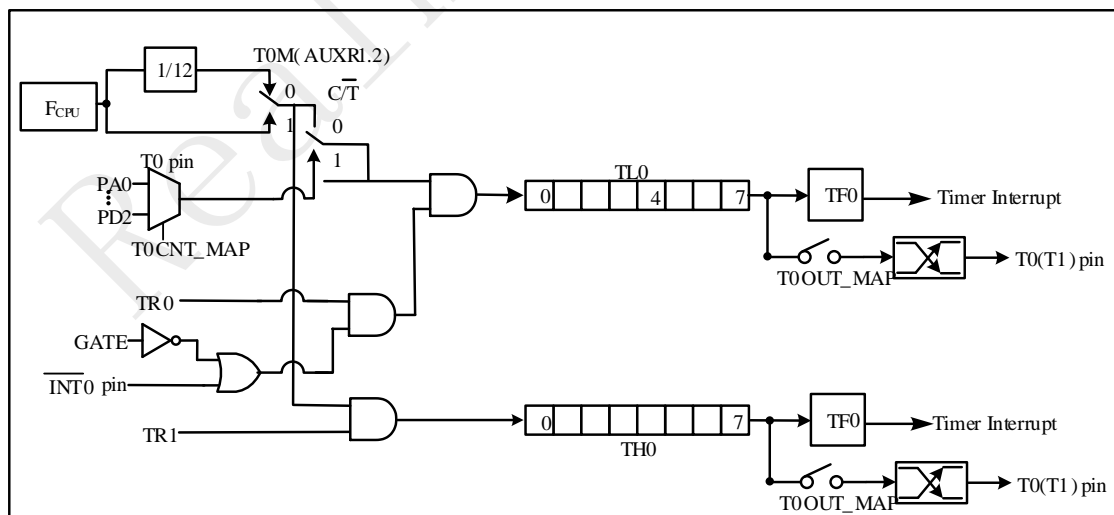
模式 2 下定时器/计数器为自动重载模式。此模式下 TL0（TL1）是一个 8 位的计数器，TH0(TH1)保存重载计数值。当 TL0(TL1)溢出后，TCON 中的 TF0(TF1)标志置位且 TH0(TH1)中内容重装至 TL0(TL1)，然后继续计数过程。重装过程中 TH0(TH1)内的值保持不变。正确设置 GATE 和  $\overline{\text{INT0}}$  ( $\overline{\text{INT1}}$ ) 引脚及 TR0(TR1)位，使能计数。GATE 和  $\overline{\text{INT0}}$  ( $\overline{\text{INT1}}$ ) 引脚的设置与模式 0 和 1 相同。


**图 10-2-3 定时器/计数器 0 和 1 的模式 2**

### 10.2.4 模式 3（两组独立 8 位定时器）

定时器 0 和定时器 1 的模式 3 有着不同的工作方式。对定时器/计数器 1 来说模式 3 会将其停用；对定时器/计数器 0 来说，模式 3 下 TLO 和 TH0 是 2 个独立的 8 位计数寄存器。模式 3 下 TLO 使用定时器 0 的控制位：如  $C/\bar{T}$ ，GATE，TR0， $\overline{INT0}$  和 TF0。TLO 也可以用来对 T0 脚上的 1 到 0 跳变计数，由  $C/\bar{T}$  (TMOD.2) 来决定。

TH0 只能对时钟周期计数，并使用定时器/计数器 1 的控制位 (TR1 和 TF1)。当需要额外的 8 位定时器时可以使用模式 3。当定时器 0 配置为模式 3 时，定时器 1 可以通过配置其进入或离开模式 3 的方式来打开或关闭自己。定时器 1 依然可以工作在模式 0、1、2 下，但它的灵活性受到限制。虽然基本功能得以维持，但已不能对 TF1 和 TR1 进行控制。此时定时器 1 依然可以使用 GATE、 $\overline{INT1}$  脚、T1M。


**图 10-2-4 定时器/计数器 0 的模式 3**

## 10.3 寄存器列表

寄存器	地址	R/W	描述	复位值
TCON	0x88	R/W	定时器 0 和 1 控制寄存器	00000000
TMOD	0x89	R/W	定时器 0 和 1 模式寄存器	00000000
TL0	0x8A	R/W	定时器 0 低 8 位数据寄存器	00000000
TL1	0x8B	R/W	定时器 1 低 8 位数据寄存器	00000000
TH0	0x8C	R/W	定时器 0 高 8 位数据寄存器	00000000
TH1	0x8D	R/W	定时器 1 高 8 位数据寄存器	00000000
AUXR1	0x9E	R/W	辅助功能寄存器 1	00000000

## 10.4 寄存器描述

### 10.4.1 定时器 0 和 1 控制寄存器 (TCON)

寄存器	地址	R/W	描述	复位后的值
TCON	0x88	R/W	定时器 0 和 1 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

位	描述	
[7]	TF1	<b>定时器 1 溢出标志.</b> 在定时器 1 溢出时该位置 1。当程序响应定时器 1 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0
[6]	TR1	<b>定时器 1 启动控制.</b> 0 = 定时器 1 中止。清该位将中止定时器 1 并且当前计数值将保存到 TH1 和 TL1 中 1 = 使能定时器 1
[5]	TF0	<b>定时器 0 溢出标志.</b> 在定时器 0 溢出时该位置 1。当程序响应定时器 0 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0
[4]	TR0	<b>定时器 0 启动控制.</b> 0 = 定时器 0 中止。清该位将中止定时器 0 并且当前计数值将保存到 TH0 和 TL0 中 1 = 使能定时器 0
[3]	IE1	<b>外部中断 1 边沿标志</b> 当检测到边沿/电平类型时，该标志由硬件置位 如果 IT1 = 1（下降沿触发），该位将保持置 1 直到软件清零或在外部中断 1 服务程序中硬件清零 如果 IT1 = 0（低电平触发），该标志是 INT1 输入信号逻辑电平的反转。软件不可控制
[2]	IT1	<b>外部中断 1 类型选择</b> 该位选择 INT1 的中断触发类型是下降沿还是低电平

		0 = $\overline{INT1}$ 为低电平触发 1 = $\overline{INT1}$ 为下降沿触发
[1]	<b>IE0</b>	<b>外部中断 0 边沿标志</b> 当检测到边沿/电平类型时, 该标志由硬件置位 如果 $IT0 = 1$ (下降沿触发), 该位将保持置 1 直到软件清零或在外部中断 0 服务程序中硬件清零 如果 $IT0 = 0$ (低电平触发), 该标志是 $\overline{INT0}$ 输入信号逻辑电平的反转。软件不可控制
[0]	<b>IT0</b>	<b>外部中断 0 类型选择</b> 该位选择 $\overline{INT0}$ 的中断触发类型是下降沿还是低电平 0 = $\overline{INT0}$ 为低电平触发 1 = $\overline{INT0}$ 为下降沿触发

### 10.4.2 定时器 0 和 1 模式寄存器 (TMOD)

寄存器	地址	R/W	描述	复位后的值
<b>TMOD</b>	0x89	R/W	定时器 0 和 1 模式寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>GATE1</b>	<b>C/T</b>	<b>M1[1:0]</b>		<b>GATE0</b>	<b>C/T0</b>	<b>M0[1:0]</b>	

位	描述	
[7]	<b>GATE1</b>	<b>定时器 1 门选择</b> 0 = 无论 $\overline{INT1}$ 的逻辑为何, 当 $TR1$ 为 1 时, 开始计数 1 = 仅当 $\overline{INT1}$ 为 1 时, $TR1$ 同时为 1, 才开始计数
[6]	<b>C/T</b>	<b>定时器 1 计数器/定时器选择.</b> 0 = 定时器 1 随内部时钟而递增 1 = 定时器 1 随外部引脚 T1 的下降沿递增
[5:4]	<b>M1[1:0]</b>	<b>定时器 1 模式选择</b> 00: 模式0 13 位定时器/计数器 01: 模式1 16 位定时器/计数器 10: 模式2 8 位定时器/计数器, 带自动 TH1 重载模式 11: 模式3 定时器 1 停止
[3]	<b>GATE0</b>	<b>定时器 0 门选择</b> 0 = 无论 $\overline{INT0}$ 的逻辑为何, 当 $TR0$ 为 1 时, 开始计数 1 = 仅当 $\overline{INT0}$ 为 1 时, $TR0$ 同时为 1, 才开始计数
[2]	<b>C/T0</b>	<b>定时器 0 计数器/定时器选择.</b> 0 = 定时器 0 随内部时钟而递增 1 = 定时器 0 随外部引脚 T0 的下降沿递增
[1:0]	<b>M0[1:0]</b>	<b>定时器 0 模式选择</b> 00: 模式0 13 位定时器/计数器 01: 模式1 16 位定时器/计数器 10: 模式2 8 位定时器/计数器, 带自动 TH0 重载模式 11: 模式3 T0 分成两个 (TL0/TH0) 独立的 8 位定时器/计数器

### 10.4.3 定时器 0 低 8 位数据寄存器 (TL0)

寄存器	地址	R/W	描述	复位后的值
<b>TL0</b>	0x8A	R/W	定时器 0 低 8 位数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TL0[7:0]							

位	描述	
[7:0]	<b>TL0[7:0]</b>	定时器 0 低 8 位数据寄存器

### 10.4.4 定时器 1 低 8 位数据寄存器 (TL1)

寄存器	地址	R/W	描述	复位后的值
<b>TL1</b>	0x8B	R/W	定时器 1 低 8 位数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TL1[7:0]							

位	描述	
[7:0]	<b>TL1[7:0]</b>	定时器 1 低 8 位数据寄存器

### 10.4.5 定时器 0 高 8 位数据寄存器 (TH0)

寄存器	地址	R/W	描述	复位后的值
<b>TH0</b>	0x8C	R/W	定时器 0 高 8 位数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TH0[7:0]							

位	描述	
[7:0]	<b>TH0[7:0]</b>	定时器 0 高 8 位数据寄存器

### 10.4.6 定时器 1 高 8 位数据寄存器 (TH1)

寄存器	地址	R/W	描述	复位后的值
<b>TH1</b>	0x8D	R/W	定时器 1 高 8 位数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TH1[7:0]							

位	描述	
[7:0]	TH1[7:0]	定时器 1 高 8 位数据寄存器

### 10.4.7 辅助功能寄存器 1 (AUXR1)

寄存器	地址	R/W	描述	复位后的值
AUXR1	0x9E	R/W	辅助功能寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SWRF	RSTPINF	HARDF	IOVTH	T1M	T0M	—	DPS

位	描述	
[7]	SWRF	软件复位标志位 软件复位后，该位将被硬件置 1，建议复位发生后通过软件清零
[6]	RSTPINF	外部复位标志位 通过外部复位引脚 RSTN 复位后，该位将被硬件置 1，建议复位发生后通过软件清零
[5]	HARDF	硬件故障复位标志 一旦程序计数器 (PC) 溢出 FLASH 地址空间，MCU 将复位并且 HARDF 硬件置位。通过软件清零。 注意：当 MCU 运行在调试模式并且 DEBUGEN=1，硬件故障复位被禁用，仅仅 HARDF (AUXR1.5) 标志位置位。
[4]	IOVTH	引脚输入的电平阈值配置 0: 输入阈值高/低电平为 0.7*VDD/0.3*VDD 1: 输入阈值高/低电平为 0.5*VDD/0.2*VDD 输入阈值配置对所有 IO 同时生效。
[3]	T1M	定时器 1 时钟选择 0: 定时器 1 的时钟源选择为 1/12 系统时钟。与标准 8051 兼容 1: 定时器 1 的时钟源选择为系统时钟
[2]	T0M	定时器 0 时钟选择 0: 定时器 0 的时钟源选择为 1/12 系统时钟。与标准 8051 兼容 1: 定时器 0 的时钟源选择为系统时钟
[1]	保留	保留
[0]	DPS	数据指针选择 0: 数据指针 0 (DPTR) 默认有效。 1: 数据指针 1 (DPTR1) 有效。 当通过 DPS 切换当前有效 DPTR 后，之前有效 DPTR 寄存器内的值保持不变。

## 11 定时器 2

### 11.1 概述

定时器 2 是一个 16 位的向上计数器,由高 8 位寄存器(TH2)和低 8 位寄存器(TL2)组成。通过配置寄存器 RCMP2H 和 RCMP2L,设置 CM/RL2(T2CON.0)后,定时器 2 能工作在比较模式和自动重载模式下。定时器 2 具有 3 通道输入捕获模块,可用于测量输入脉冲宽度或周期。3 通道捕获结果分别存放在寄存器 C0H 及 C0L, C1H 及 C1L, C2H 及 C2L 中。定时器 2 的时钟来自系统时钟的分频,总共具有 8 级分频,可适用于更多应用需求。当 TR2 (T2CON.2)置 1,定时器使能;TR2 置 0 时,定时器关闭。下列寄存器用于控制定时器 2 功能。

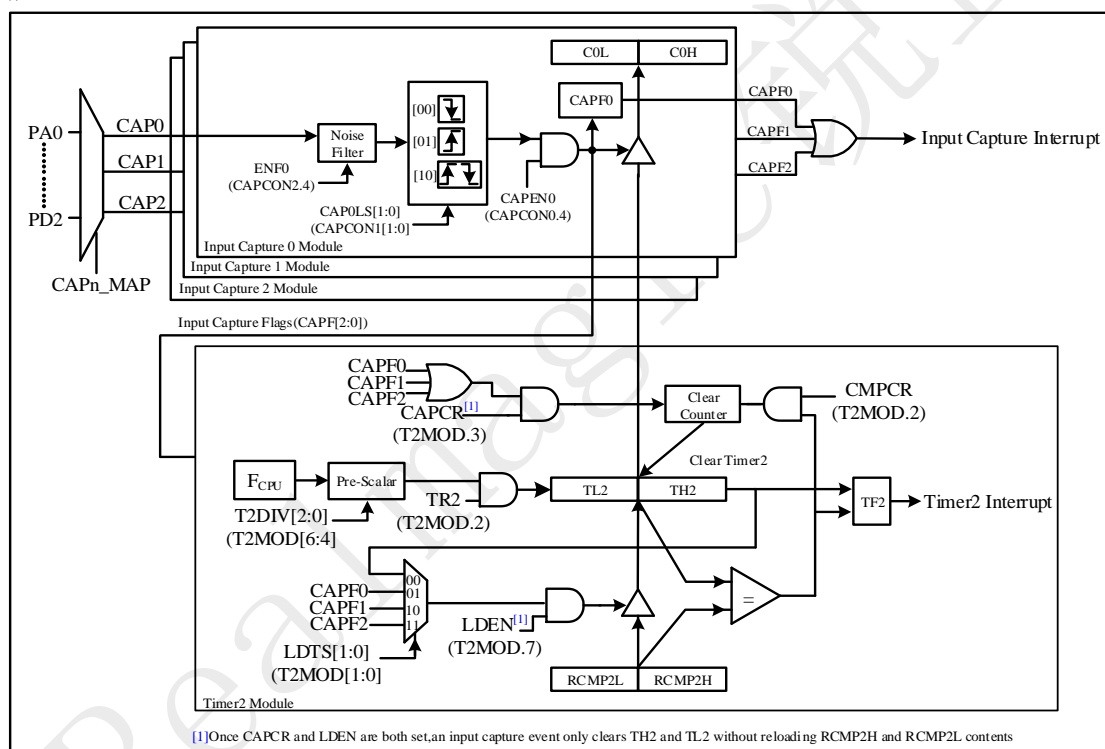


图 11-1-1 定时器 2 框图

### 11.2 功能描述

#### 11.2.1 自动重载功能模式

当 CM/RL2 清 0,定时器 2 配置为自动重载模式。在该模式下,RCMP2H 及 RCMP2L 保存重载的数值。当 LDEN 置位后,一旦有触发事件发生,硬件将自动把 RCMP2H 及 RCMP2L 寄存器内的值写入 TH2 及 TL2 中。触发事件可以是定时器 2 溢出或是一个所配置

的捕获信道有触发事件发生（根据 LDTS[1:0] (T2MOD[1:0])配置）。注意，一旦 CAPCR (T2MOD.3) 置 1, 如有一个捕获事件发生, 仅清除 TH2 及 TL2 内的值, 不会将 RCMP2H 及 RCMP2L 的值载入。

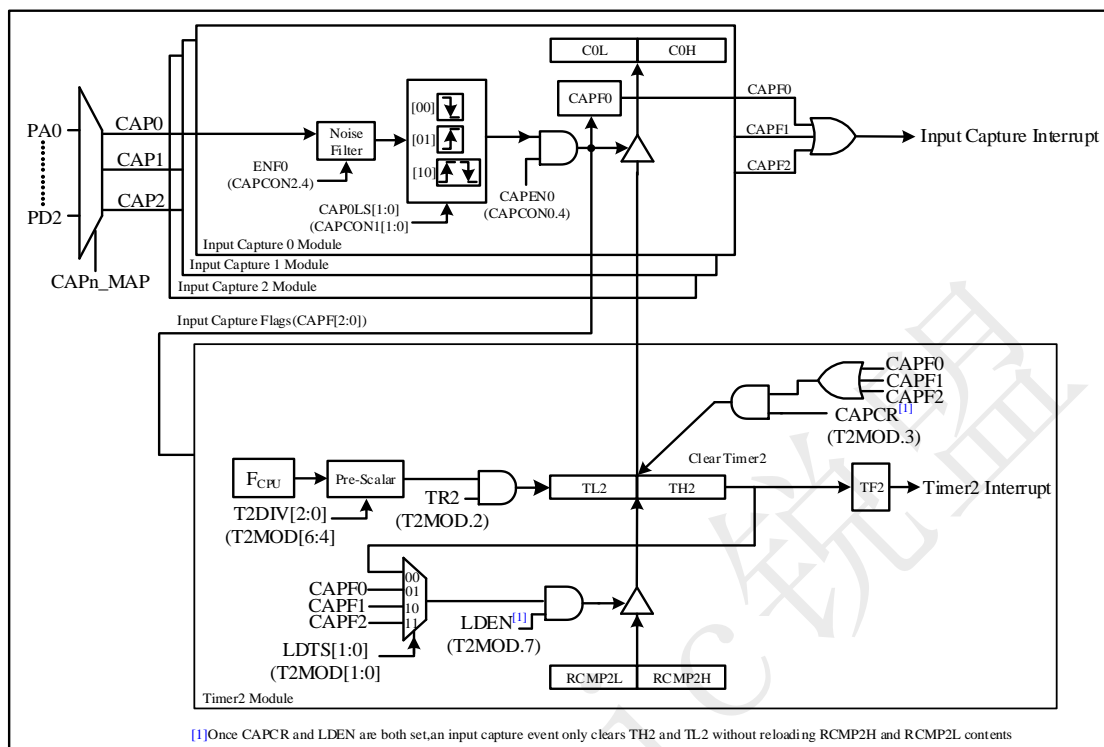


图 11-2-1 定时器 2 自动重载和捕获模式功能模块图

## 11.2.2 比较功能模式

当 CM/RL2 置 1, 定时器 2 配置为比较器模式。在该模式下, RCMP2H 及 RCMP2L 预存待比较数据。由于定时器 2 向上计数, 一旦 TH2 和 TL2 匹配 RCMP2H 和 RCMP2L 的设定值, TF2 (T2CON.7) 将会由硬件置 1, 用以标示发生了比较匹配事件。

如果 CMPCR (T2MOD.2) 置 1, 当发生比较匹配事件后, 定时器 2 计数器将硬件自动清 0。



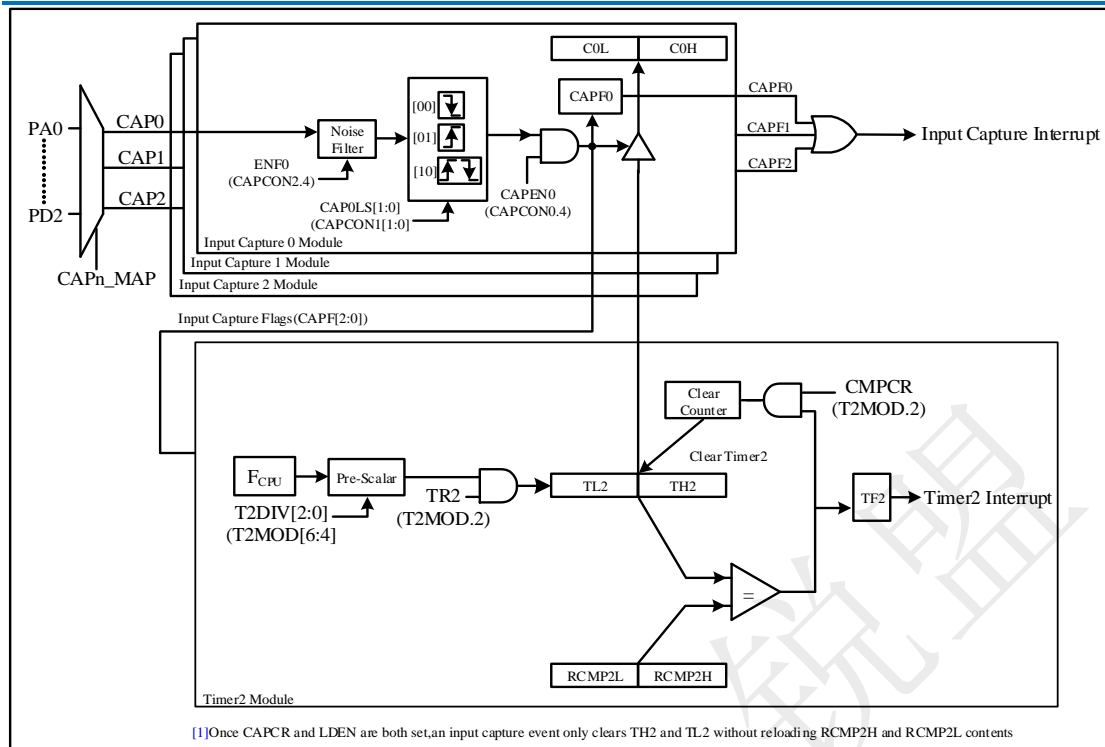


图 11-3-2 定时器 2 比较模式与输入捕获模式结构功能图

### 11.2.3 输入捕获功能模块

输入捕获模块依靠定时器 2 实现输入捕获功能。输入捕获模块通过寄存器 CAPCON0~2 配置来支持 3 组信道输入 (CAP0, CAP1 和 CAP2)。每个输入通道的噪声滤波器可通过设置 ENF0~2 (CAPCON2[6:4]) 使能, 可滤除小于 4 个系统时钟的输入毛刺。每组输入捕获通道共享定时器 2 计数, 但有自己独立的边沿检测。每个触发边沿检测可由寄存器 CAPCON1 的相关位独立配置, 支持正边沿捕获, 负边沿捕获, 或双边沿捕获。在使用前, 必须设置通道使能位 CAPEN0~2 (CAPCON0[6:4])。

当输入捕获通道使能且所选择的边沿触发发生时, 定时器 2 的计数值 TH2 和 TL2 将被捕获、传输并存储到捕获寄存器 CnH 和 CnL。边沿触发也可硬件置位 CAPF<sub>n</sub>(CAPCON0.n), 如果 ECAP (EIE.2) 和 EA 都打开, 将产生中断。三组输入捕获共享一个中断向量, 用户可通过检查 CAPF<sub>n</sub> 来确定具体哪个通道有输入捕获。这些标志必须由软件清零。

CAPCR (CAPCON2.3) 用于周期计算。当设置 CAPCR 为 1, 一个捕获边沿事件发生后, TH2 和 TL2 的值将被保存, 然后硬件将自动清除定时器 2 的值为 0000H。这样可以避免常规软件写 16 位计数或者算法开销。

## 11.3 寄存器列表

寄存器	地址	R/W	描述	复位值
T2CON	0xC0	R/W	定时器 2 控制寄存器	00000000
T2MOD	0xC9	R/W	定时器 2 模式选择	00000000
RCMP2L	0xC6	R/W	定时器 2 重装载/比较数据低字节	00000000
RCMP2H	0xC7	R/W	定时器 2 重装载/比较数据高字节	00000000
TL2	0xC4	R/W	定时器 2 低字节数据寄存器	00000000
TH2	0xC5	R/W	定时器 2 高字节数据寄存器	00000000
CAPCON0	0xC1	R/W	输入捕获控制寄存器 0	00000000
CAPCON1	0xC2	R/W	输入捕获控制寄存器 1	00000000
CAPCON2	0xC3	R/W	输入捕获控制寄存器 2	00000000
T2C0L	0xCA	R/W	捕获通道 0 输入结果低字节	00000000
T2C0H	0xCB	R/W	捕获通道 0 输入结果高字节	00000000
T2C1L	0xCC	R/W	捕获通道 1 输入结果低字节	00000000
T2C1H	0xCD	R/W	捕获通道 1 输入结果高字节	00000000
T2C2L	0xCE	R/W	捕获通道 2 输入结果低字节	00000000
T2C2H	0xCF	R/W	捕获通道 2 输入结果高字节	00000000

## 11.4 寄存器描述

### 11.4.1 定时器 2 控制寄存器 (T2CON)

寄存器	地址	R/W	描述	复位后的值
<b>T2CON</b>	0xC0	R/W	定时器 2 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>TF2</b>	-	-	-	-	<b>TR2</b>	-	<b>CM/RL2</b>

位	描述	
[7]	<b>TF2</b>	<b>定时器 2 溢出标志</b> 当定时器 2 溢出或者比较数据符合, 该位置 1。如果已使能定时器 2 中断, 该位置 1 会让芯片进入定时器 2 中断运行。该位硬件置 1, 但需要软件清 0。
[6:3]	<b>保留</b>	保留
[2]	<b>TR2</b>	<b>定时器 2 运行控制</b> 0 = 定时器计数关闭。清除该位关闭定时器计数, 目前已计数值会保留在 TH2 及 TL2 中 1 = 定时器 2 计数使能。
[1]	<b>保留</b>	保留
[0]	<b>CM/RL2</b>	<b>定时器 2 比较及自动重装载功能选择</b>

	0 = 自动重载模式 1 = 比较器模式.
--	--------------------------

### 11.4.2 定时器 2 模式选择 (T2MOD)

寄存器	地址	R/W	描述	复位后的值
T2MOD	0xC9	R/W	定时器 2 模式选择	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDT[1:0]	

位	描述	
[7]	LDEN	<b>自动重载使能位</b> 0 = 自动重载功能关闭 1 = 将 RCMP2H 及 RCMP2L 自动重载至 TH2 及 TL2 功能使能
[6:4]	T2DIV[2:0]	<b>定时器 2 时钟分频</b> 000: 定时器 2 时钟分频为 1/1. 001: 定时器 2 时钟分频为 1/4. 010: 定时器 2 时钟分频为 1/16. 011: 定时器 2 时钟分频为 1/32. 100: 定时器 2 时钟分频为 1/64. 101: 定时器 2 时钟分频为 1/128. 110: 定时器 2 时钟分频为 1/256. 111: 定时器 2 时钟分频为 1/512.
[3]	CAPCR	<b>捕获模式自动清除</b> 该位仅当定时器 2 设定为自动重载模式下有效。该位使能, 当捕获完成, TH2 及 TL2 内数据移入 RCMP2H 及 RCMP2L 后, 硬件自动清除 TH2 及 TL2 计数器功能。 0 = 捕获完成后定时器 2 计数按之前计数值继续累加。 1 = 捕获完成后定时器 2 数据自动清 0
[2]	CMPCR	<b>比较完成自动清除</b> 该位仅当定时器 2 比较功能模式下有效。当比较符合后, 自动清除 TH2 及 TL2 计数器值 0 = 比较符合之后, 定时器 2 内数据按之前继续计数。 1 = 比较符合之后, 定时器 2 内数据自动清 0。
[1:0]	LDT[1:0]	<b>自动重载触发选择</b> 00: 当定时器 2 溢出自动重载 01: 当捕获 0 通道事件完成, 自动重载 10: 当捕获 1 通道事件完成, 自动重载 11: 当捕获 2 通道事件完成, 自动重载

### 11.4.3 定时器 2 重载/ 比较数据低字节 (RCMP2L)

寄存器	地址	R/W	描述	复位后的值
<b>RCMP2L</b>	0xC6	R/W	定时器 2 重载/比较数据低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>RCMP2L[7:0]</b>							

位	描述	
[7:0]	<b>RCMP2L[7:0]</b>	<b>定时器 2 重载/比较器低字节</b> 当定时器 2 设定为比较模式，预存放低字节待比较数据。当设定为自动重载，预存放低字节数据。

### 11.4.4 定时器 2 重载/ 比较数据高字节 (RCMP2H)

寄存器	地址	R/W	描述	复位后的值
<b>RCMP2H</b>	0xC7	R/W	定时器 2 重载/比较数据高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>RCMP2H[7:0]</b>							

位	描述	
[7:0]	<b>RCMP2H[7:0]</b>	<b>定时器 2 重载/比较器高字节</b> 当定时器 2 设定为比较模式，预存放低字节待比较数据。当设定为自动重载，预存放低字节数据。

### 11.4.5 定时器 2 低字节数据 (TL2)

寄存器	地址	R/W	描述	复位后的值
<b>TL2</b>	0xC4	R/W	定时器 2 低字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>TL2[7:0]</b>							

位	描述	
[7:0]	<b>TL2[7:0]</b>	<b>定时器 2 低字节数据</b> 该寄存器存放 16 位定时器 2 实际计数的低 8 位字节数据。

### 11.4.6 定时器 2 高字节数据 (TH2)

寄存器	地址	R/W	描述	复位后的值
TH2	0xC5	R/W	定时器 2 高字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TH2[7:0]							

位	描述	
[7:0]	TH2[7:0]	<b>定时器 2 高字节数据</b> 该寄存器存放 16 位定时器 2 实际计数的高 8 位字节数据。

### 11.4.7 输入捕获控制寄存器 0 (CAPCON0)

寄存器	地址	R/W	描述	复位后的值
CAPCON0	0xC1	R/W	输入捕获控制寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0

位	描述	
[7]	保留	保留
[6]	CAPEN2	<b>输入捕获通道 2 使能位</b> 0: 关闭输入捕获通道 2 1: 打开输入捕获通道 2
[5]	CAPEN1	<b>输入捕获通道 1 使能位</b> 0: 关闭输入捕获通道 1 1: 打开输入捕获通道 1
[4]	CAPEN0	<b>输入捕获通道 0 使能位</b> 0: 关闭输入捕获通道 0 1: 打开输入捕获通道 0
[3]	保留	保留
[2]	CAPF2	<b>输入捕获通道 2 标志位</b> 如果输入捕获 2 边沿发生, 该位由硬件置位, 由软件清零
[1]	CAPF1	<b>输入捕获通道 1 标志位</b> 如果输入捕获 1 边沿发生, 该位由硬件置位, 由软件清零
[0]	CAPF0	<b>输入捕获通道 0 标志位</b> 如果输入捕获 0 边沿发生, 该位由硬件置位, 由软件清零

### 11.4.8 输入捕获控制寄存器 1 (CAPCON1)

寄存器	地址	R/W	描述	复位后的值
CAPCON1	0xC2	R/W	输入捕获控制寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	

位	描述	
[7:6]	保留	保留
[5:4]	CAP2LS[1:0]	输入捕获通道 2 条件选择 00: 下降沿 01: 上升沿 10: 上升沿或下降沿 11: 保留
[3:2]	CAP1LS[1:0]	输入捕获通道 1 条件选择 00: 下降沿 01: 上升沿 10: 上升沿或下降沿 11: 保留
[1:0]	CAP0LS[1:0]	输入捕获通道 0 条件选择 00: 下降沿 01: 上升沿 10: 上升沿或下降沿 11: 保留

### 11.4.9 输入捕获控制寄存器 2 (CAPCON2)

寄存器	地址	R/W	描述	复位后的值
CAPCON2	0xC3	R/W	输入捕获控制寄存器 2	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	ENF2	ENF1	ENF0	-	-	-	-

位	描述	
[7]	保留	保留
[6]	ENF2	输入捕获通道 2 噪声滤波使能位 0: 关闭输入捕获通道 2 的噪声滤波 1: 打开输入捕获通道 2 的噪声滤波
[5]	ENF1	输入捕获通道 1 噪声滤波使能位 0: 关闭输入捕获通道 1 的噪声滤波 1: 打开输入捕获通道 1 的噪声滤波
[4]	ENF0	输入捕获通道 0 噪声滤波使能位

		0: 关闭输入捕获通道 0 的噪声滤波 1: 打开输入捕获通道 0 的噪声滤波
[3:0]	保留	保留

### 11.4.10 捕获通道 0 低字节数据 (T2C0L)

寄存器	地址	R/W	描述	复位后的值
<b>T2C0L</b>	0xCA	R/W	捕获通道 0 低字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C0L[7:0]							

位	描述	
[7:0]	<b>T2C0L[7:0]</b>	捕获通道 0 输入结果低字节 寄存器 T2C0L 是 16 位捕获通道 0 输入结果的低字节值

### 11.4.11 捕获通道 0 高字节数据 (T2C0H)

寄存器	地址	R/W	描述	复位后的值
<b>T2C0H</b>	0xCB	R/W	捕获通道 0 高字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C0H[7:0]							

位	描述	
[7:0]	<b>T2C0H[7:0]</b>	捕获通道 0 输入结果高字节 寄存器 T2C0H 是 16 位捕获通道 0 输入结果的高字节值

### 11.4.12 捕获通道 1 低字节数据 (T2C1L)

寄存器	地址	R/W	描述	复位后的值
<b>T2C1L</b>	0xCC	R/W	捕获通道 1 低字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C1L[7:0]							

位	描述	
[7:0]	<b>T2C1L[7:0]</b>	捕获通道 1 输入结果低字节

	寄存器 T2C1L 是 16 位捕获通道 1 输入结果的低字节值
--	----------------------------------

### 11.4.13 捕获通道 1 高字节数据 (T2C1H)

寄存器	地址	R/W	描述	复位后的值
<b>T2C1H</b>	0xCD	R/W	捕获通道 1 高字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C1H[7:0]							

位	描述	
[7:0]	<b>T2C1H[7:0]</b>	捕获通道 1 输入结果高字节 寄存器 T2C1H 是 16 位捕获通道 1 输入结果的高字节值

### 11.4.14 捕获通道 2 低字节数据 (T2C2L)

寄存器	地址	R/W	描述	复位后的值
<b>T2C2L</b>	0xCE	R/W	捕获通道 2 低字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C2L[7:0]							

位	描述	
[7:0]	<b>T2C2L[7:0]</b>	捕获通道 2 输入结果低字节 寄存器 T2C2L 是 16 位捕获通道 2 输入结果的低字节值

### 11.4.15 捕获通道 2 高字节数据 (T2C2H)

寄存器	地址	R/W	描述	复位后的值
<b>T2C2H</b>	0xCF	R/W	捕获通道 2 高字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T2C2H[7:0]							

位	描述	
[7:0]	<b>T2C2H[7:0]</b>	捕获通道 2 输入结果高字节 寄存器 T2C2H 是 16 位捕获通道 2 输入结果的高字节值



## 12 定时器 3

### 12.1 概述

定时器 3 是一个 16 位自动重载，向上计数定时器。用户可以通过配置 T3DIV[2:0] (T3CON[5:4])选择预分频，并写入重载值到 T3RH 和 T3RL 寄存器来决定它的溢出速率。用户可以设置 TR3(T3CON.2)来开始计数。当计数跨过 FFFFH, TF3(T3CON.7)置为 1, 且 T3RH 和 T3RL 寄存器的内容重载到内部 16 位计数器。如果 ET3 (EIE1.1) 置为 1, 定时器 3 中断服务程序被执行。当进入中断服务程序, TF3 会被硬件自动清零。

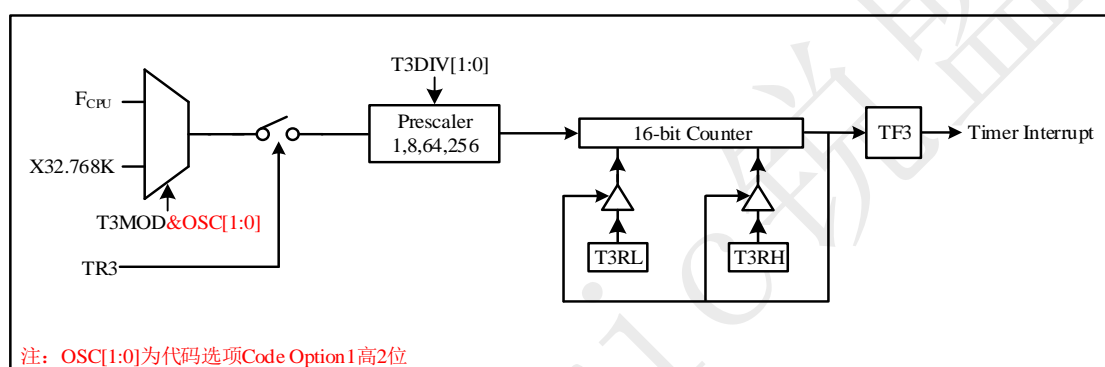


图 12-1-1 定时器 3 框图

当 T3MOD=1 且代码选项 Code Option1 的高两位 OSC[1:0]=11 时, 定时器 3 工作在 RTC 模式, 即定时器 3 的计数时钟源为外部 32.768KHz 低频晶振 (外部低频晶振在掉电模式下不会关闭)。此时, 如果 T3RH 和 T3RL 都为 0, RTC 定时时间将根据 T3DIV[1:0]的值选择 0.25s/0.125s/0.5s/1s 中的一个; 如果 T3RH 和 T3RL 不为 0, 定时时间为 T3R 到 0xFFFF 的溢出时间。

### 12.2 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>T3CON</b>	0xA5	R/W	定时器 3 控制寄存器	00000000
<b>T3RL</b>	0xA6	R/W	定时器 3 低字节数据寄存器	00000000
<b>T3RH</b>	0xA7	R/W	定时器 3 高字节数据寄存器	00000000

## 12.3 寄存器描述

### 12.3.1 定时器 3 控制寄存器 (T3CON)

寄存器	地址	R/W	描述	复位后的值
T3CON	0xA5	R/W	定时器 3 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
TF3	保留	T3DIV[2:0]		保留	TR3	保留	T3MOD

位	描述	
[7]	TF3	<b>定时器 3 溢出标志</b> 当定时器 3 溢出或者比较数据符合, 该位置 1。如果已使能定时器 3 中断, 该位置 1 会让芯片进入定时器 3 中断运行。该位硬件置 1, 进中断后硬件自动清 0。
[6]	保留	保留
[5:4]	T3DIV[1:0]	<b>定时器时钟分频</b> 00: 定时器 3 时钟分频为 1/1 01: 定时器 3 时钟分频为 1/8 10: 定时器 3 时钟分频为 1/64 11: 定时器 3 时钟分频为 1/256  <b>TH3:TL3 不为 0 时, 此两位不起作用, RTC 根据 TH3:TL3 设置的值为溢出值来计数定时; TH3:TL3 为 0 时, RTC 定时时间选择</b> 00: RTC 定时时间为 0.25s 01: RTC 定时时间为 0.125s 10: RTC 定时时间为 0.5s 11: RTC 定时时间为 1s
[3]	保留	保留
[2]	TR3	<b>定时器 3 运行控制</b> 0 = 定时器计数关闭。清除该位关闭定时器计数, 目前已计数值会保留在 TH3 及 TL3 中 1 = 定时器 3 计数使能。
[1]	保留	保留
[0]	T3MOD	<b>定时器 3 工作模式选择</b> 0: 定时器模式, 定时器 3 时钟为系统时钟 FSYS。 1: 当 OSC[1:0]=11 时为 RTC 模式, 否则为定时器模式, RTC 模式下定时器 3 时钟为外部 32768 时钟。

### 12.3.2 定时器 3 重载值低字节数据 (T3RL)

寄存器	地址	R/W	描述	复位后的值
-----	----	-----	----	-------

<b>T3RL</b>	0xA6	R/W	定时器 3 重载值低字节数据	00000000
-------------	------	-----	----------------	----------

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T3RL[7:0]</b>							

位	描述	
[7:0]	<b>T3RL[7:0]</b>	<b>定时器 3 重载值低字节数据</b> 该寄存器存放 16 位定时器 3 重载值低 8 位字节数据。

### 12.3.3 定时器 3 重载值高字节数据 (T3RH)

寄存器	地址	R/W	描述	复位后的值
<b>T3RH</b>	0xA7	R/W	定时器 3 重载值高字节数据	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T3RH[7:0]</b>							

位	描述	
[7:0]	<b>T3RH[7:0]</b>	<b>定时器 3 高字节数据</b> 该寄存器存放 16 位定时器 3 重载值高 8 位字节数据。

## 13 定时器 4/PWM

### 13.1 概述

定时器 4 是一个 16 位定时器，可配置两路 PWM 输出。用户可以通过配置 T4DIV[2:0] (T4CON[3:1]) 选择定时器分频，并写入 T4OVR 寄存器来决定它的溢出速率。用户可以设置 T4EN (T4CON.0) 来开始计数。当内部计数器计数等于 T4OVR 时，TF4 (T3CON.5) 置为 1，且内部计数器会从 0 开始计数。如果 ET4 (EIE1.3) 置为 1，定时器 4 中断服务程序被执行。当进入中断服务程序，TF4 需软件清零。如果 ET4PWM (EIE1.5) 置为 1，定时器 4PWM 中断服务程序被执行。当进入中断服务程序，T4PWM0F 和 T4PWM1F 需软件清零。

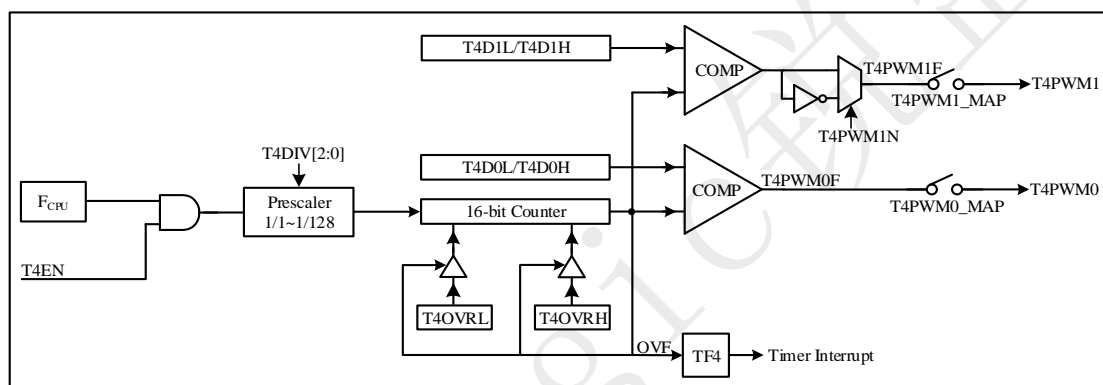


图 13-1-1 定时器 4 框图

### 13.2 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>T4CON</b>	0xF050	R/W	定时器 4 控制寄存器	00000000
<b>T4OVRH</b>	0xF052	R/W	定时器 4 定时溢出高字节寄存器	00000000
<b>T4OVRH</b>	0xF052	R/W	定时器 4 定时溢出高字节寄存器	00000000
<b>T4D0L</b>	0xF053	R/W	定时器 4 PWM0 占空比配置寄存器低字节	00000000
<b>T4D0H</b>	0xF054	R/W	定时器 4 PWM0 占空比配置寄存器高字节	00000000
<b>T4D1L</b>	0xF055	R/W	定时器 4 PWM1 占空比配置寄存器低字节	00000000
<b>T4D1H</b>	0xF056	R/W	定时器 4 PWM1 占空比配置寄存器高字节	00000000

## 13.3 寄存器描述

### 13.3.1 定时器 4 控制寄存器 (T4CON)

寄存器	地址	R/W	描述	复位后的值
<b>T4CON</b>	0xF050	R/W	定时器 4 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T4PWM1F</b>	<b>T4PWM0F</b>	<b>TF4</b>	<b>T4PWM1N</b>	<b>T4DIV[2:0]</b>			<b>T4EN</b>

位	描述	
[7]	<b>T4PWM1F</b>	定时器 4 PWM1 中断标志位, 软件清 0。
[6]	<b>T4PWM0F</b>	定时器 4 PWM0 中断标志位, 软件清 0。
[5]	<b>TF4</b>	定时器 4 中断标志位, 软件清 0。
[4]	<b>T4PWM1N</b>	定时器 4 PWM1 负极性输出位 0: T4PWM1 直接输出 1: T4PWM1 取反输出, 如 T4D0 和 T4D1 配置成一样的值, 那么 T4PWM0 和 T4PWM1 为一对互补输出。
[3:1]	<b>T4DIV[2:0]</b>	定时器 4 时钟分频 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[0]	<b>T4EN</b>	定时器 4 使能位 0: 禁止定时器 4 1: 使能定时器 4

### 13.3.2 定时器 4 定时溢出低字节寄存器 (T4OVR)

寄存器	地址	R/W	描述	复位后的值
<b>T4OVR</b>	0xF051	R/W	定时器 4 定时溢出低字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T4OVR[7:0]</b>							

位	描述	
[7:0]	<b>T4OVR[7:0]</b>	定时器 4 定时溢出和 T4 PWM 周期低字节寄存器

	该寄存器定时器 4 定时溢出时间和 T4 PWM 周期。
--	------------------------------

### 13.3.3 定时器 4 定时溢出高字节寄存器 (T4OVRH)

寄存器	地址	R/W	描述	复位后的值
<b>T4OVRH</b>	0xF052	R/W	定时器 4 定时溢出高字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T4OVR[15:8]							

位	描述	
[7:0]	<b>T4OVR[15:8]</b>	定时器 4 定时溢出和 T4 PWM 周期高字节寄存器 该寄存器定时器 4 定时溢出时间和 T4 PWM 周期。

### 13.3.4 定时器 4 PWM0 占空比低字节寄存器 (T4D0L)

寄存器	地址	R/W	描述	复位后的值
<b>T4D0L</b>	0xF053	R/W	定时器 4 PWM0 占空比低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T4D0[7:0]							

位	描述	
[7:0]	<b>T4D0[7:0]</b>	定时器 4 PWM0 占空比低字节 该寄存器配置 T4PWM0 占空比低字节

### 13.3.5 定时器 4 PWM0 占空比高字节寄存器 (T4D0H)

寄存器	地址	R/W	描述	复位后的值
<b>T4D0H</b>	0xF054	R/W	定时器 4 PWM0 占空比高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
T4D0[15:8]							

位	描述	
[7:0]	<b>T4D0[15:8]</b>	定时器 4 PWM0 占空比高字节 该寄存器配置 T4PWM0 占空比

### 13.3.6 定时器 4 PWM1 占空比低字节 (T4D1L)

寄存器	地址	R/W	描述	复位后的值
<b>T4D1L</b>	0xF055	R/W	定时器 4 PWM1 占空比低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T4D1[7:0]</b>							

位	描述	
[7:0]	<b>T4D1[7:0]</b>	定时器 4 PWM1 占空比低字节 该寄存器配置 T4PWM1 占空比低字节

### 13.3.7 定时器 4 PWM1 占空比高字节 (T4D1H)

寄存器	地址	R/W	描述	复位后的值
<b>T4D1H</b>	0xF056	R/W	定时器 4 PWM1 占空比高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>T4D1[15:8]</b>							

位	描述	
[7:0]	<b>T4D1[15:8]</b>	定时器 4 PWM1 占空比高字节 该寄存器配置 T4PWM1 占空比高字节

## 14 脉冲宽度调制 (PWM)

### 14.1 概述

PWM(脉冲宽度调制) 信号在控制方案中应用非常广泛。可用于电机驱动、风扇控制、背光调节、LED 光源调光或通过低通滤波器电路模拟一个简单的数模转换模块的输出。

RM1221A 包含三对 (6 个 PWM 通道) 16 位精度、可调周期和占空比的 PWM 输出, 非常适合用于电机控制。该模块架构适用于驱动单相或三相无刷直流电机(BLDC), 或三相交流感应电机。每个信道 PWM 输出可配置为独立输出模式、互补模式或同步模式。当设定为互补模式时, 通过插入可配置的死区时间, 保护 MOS 管同时导通。PWM 波形可配置边沿对齐或中心对齐来选择中断响应位置。

### 14.2 功能描述

#### 14.2.1 脉冲宽度调制发生器

脉冲宽度调制发生器时钟由系统时钟或定时器 1 计数溢出产生, 可通过 PWM 时钟预分频调整 1/1~1/128 除频。PWM 周期由 16 位周期寄存器 {PWMOVRL, PWMOVRL} 组合预先设置。所有 PWM 通道共享同一个 16 位周期计数器, 周期是相同的。每个 PWM 通道都有独立占空比寄存器分别为 {PWM0H, PWM0L}、{PWM1H, PWM1L}、{PWM2H, PWM2L}、{PWM3H, PWM3L}、{PWM4H, PWM4L} 及 {PWM5H, PWM5L}。有 6 个占空比寄存器, 6 个通道都能产生独立占空比的 PWM 信号, PWM 信号的周期和占空比可通过 16 位计数器跟周期和占空比寄存器比较产生。

为了更好适用于三相电机控制, 通过设置 GP(PWMCON1.5)位来使用组群模式, {PWM0H, PWM0L}和{PWM1H, PWM1L}占空比寄存器决定 PWM 输出的占空比。在三相电机控制应用中, 另二对 PWM 输出产生完全相同的占空比信号, 一旦组群模式启用 {PWM2H, PWM2L}、{PWM3H, PWM3L}、{PWM4H, PWM4L} 和 {PWM5H, PWM5L} 寄存器失效。意味着寄存器 {PWM2H, PWM2L} 和 {PWM4H, PWM4L} 与 {PWM0H, PWM0L} 相同, {PWM3H, PWM3L} 和 {PWM5H, PWM5L} 与 {PWM1H, PWM1L} 相同。

注: 启用 PWM 输出, 芯片不会自动配置管脚为“输出模式”, 用户需要通过软件配置, 参见[章节 9.4 数字外设功能引脚全映射控制](#)。



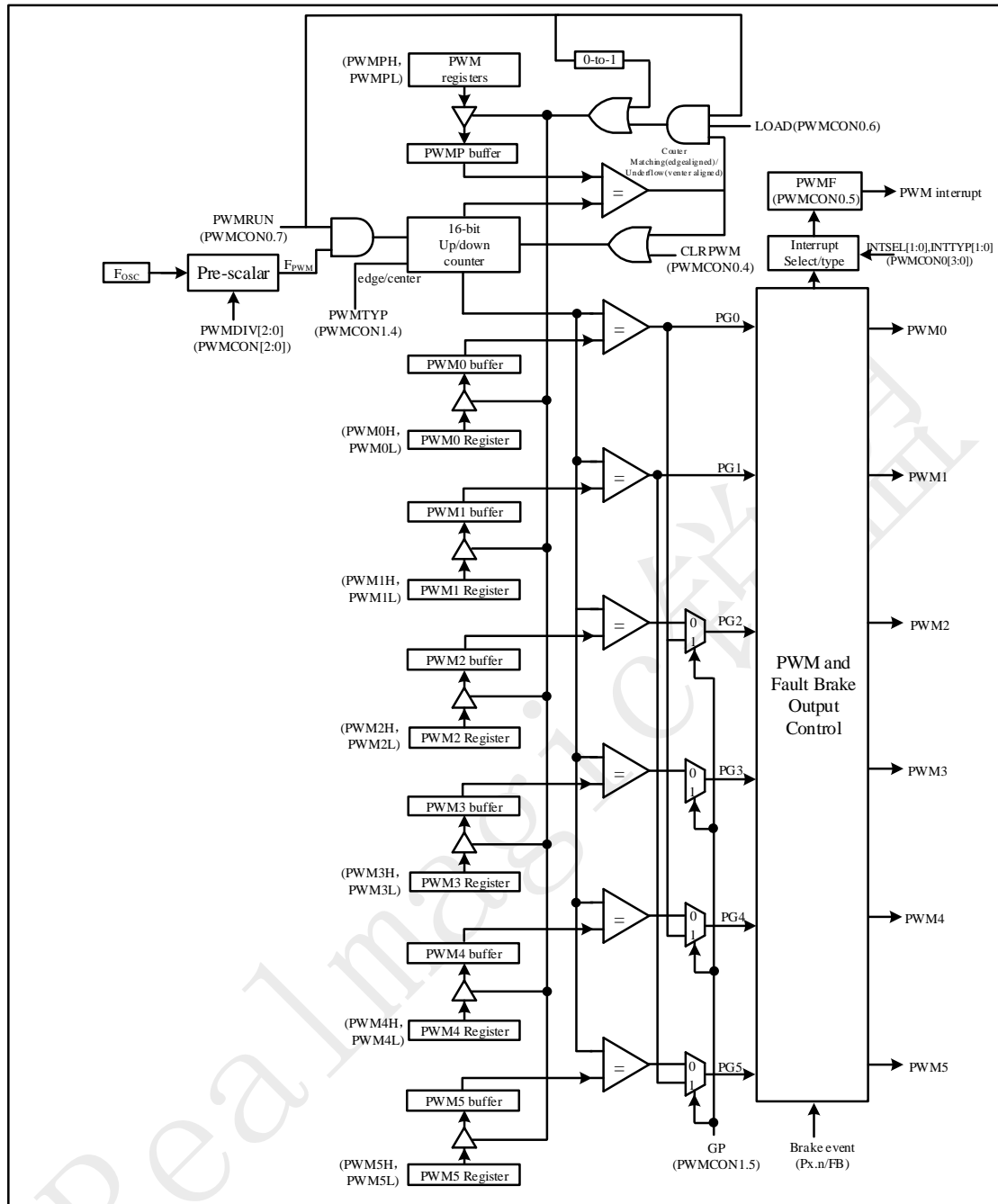
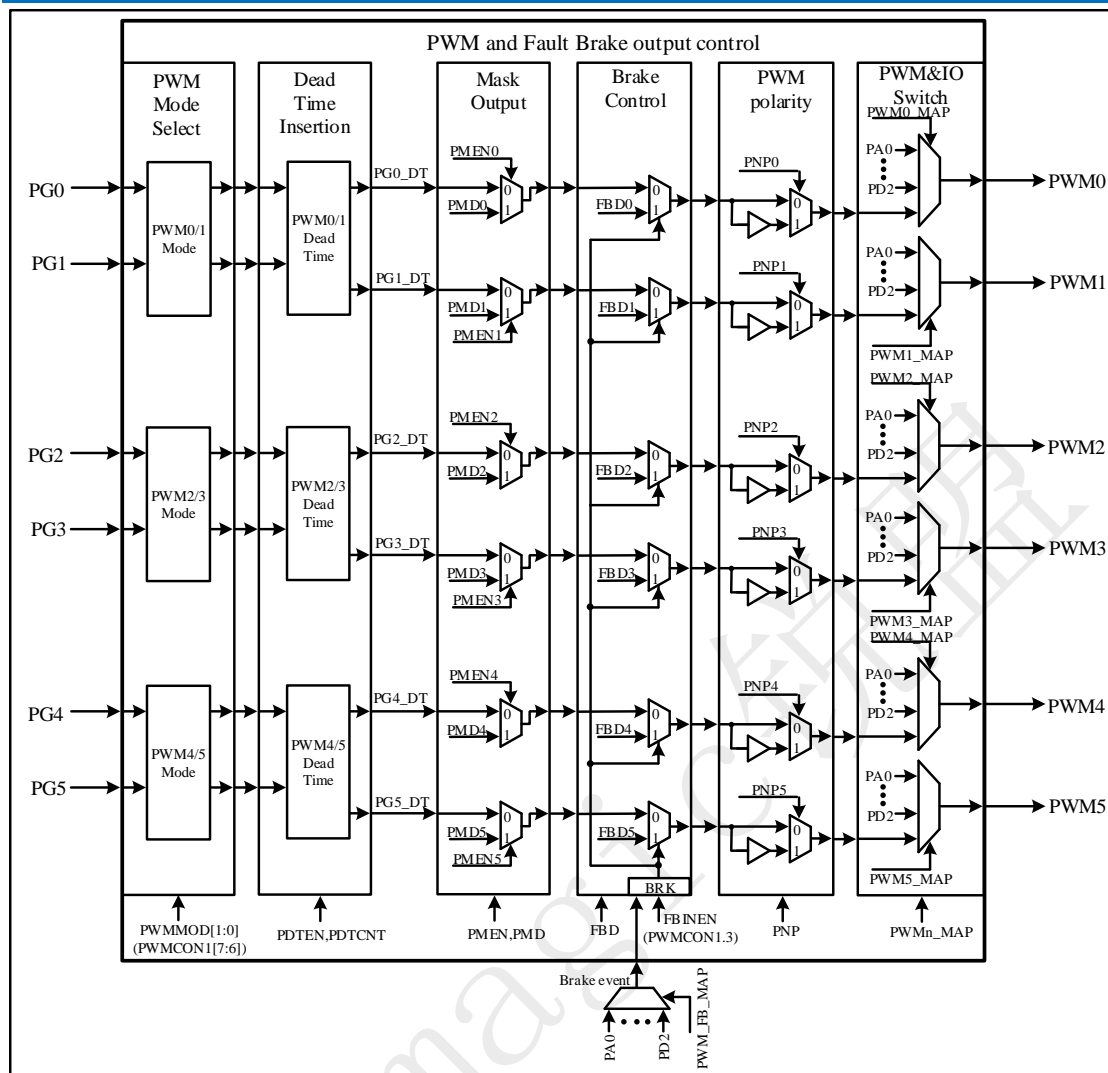


图 14-2-1-1 脉冲宽度调制模块框图

脉冲宽度调制计数器产生六个 PWM 信号，称为 PG0、PG1、PG2、PG3、PG4 及 PG5。这些信号通过 PWM 及故障刹车输出控制电路，输出至 I/O 管脚。输出控制电路决定输出 PWM 信号的模式、死区时间、输出掩码、故障刹车以及 PWM 的极性。


**图 14-2-1-2 脉冲宽度调制以及故障刹车控制模块图**

请按照如下初始化步骤来产生 PWM 信号。第一步，设定 CLRPWM(PWMCON0.4)位，清零计数器以确保 16 位向上计数器计数正确。然后设定 {PWMOVRL, PWMOVRL} 及所有 {PWMnH, PWMnL} 寄存器。对 PWMRUN(PWMCON0.7)置 1，开始 16 位向上计数器计数。PWM 信号开始产生，对应管脚输出 PWM 信号。所有的周期及占空比寄存器，具有硬件双缓存设计，因此 {PWMOVRL, PWMOVRL} 及 {PWMnH, PWMnL} 寄存器可随时被改写，但不会立即更新 PWM 周期和占空比，直到寄存器 LOAD(PWMCON0.6)置 1 后和当前周期完成。用于防止产生非完整周期或占空比的 PWM 波形。

**设置 LOAD 更新 PWM 周期及占空比寄存器值，通过监视 LOAD 位硬件自动清零确保完成，任何当 LOAD 还在保持 1 的时候，对周期或占空比寄存器内容的更改，可能引发无法预测的结果。**

## 14.2.2 PWM 类型

PWM 发生器包含两种类型，边沿对齐或中心对齐，设置 PWMTYP(PWMCON1.4) 位决定。

### 14.2.2.1 边沿对齐模式

边沿对齐模式，16 位计数器设定为单周期模式，从 0000H 向上计数，直到与 {PWMOVRH, PWMOVRL} 匹配，然后重新从 0000H 开始向上计数。PWM 信号 (PGn 信号在 PWM 和故障刹车输出控制模块之前) 在 16 位计数器与周期寄存器和占空比寄存器 {PWMnH, PWMnL} 设置值匹配时停止 PGn 信号，并且设置 16 位计数器为 0000H。PWM 输出波形为左边沿对齐方式。

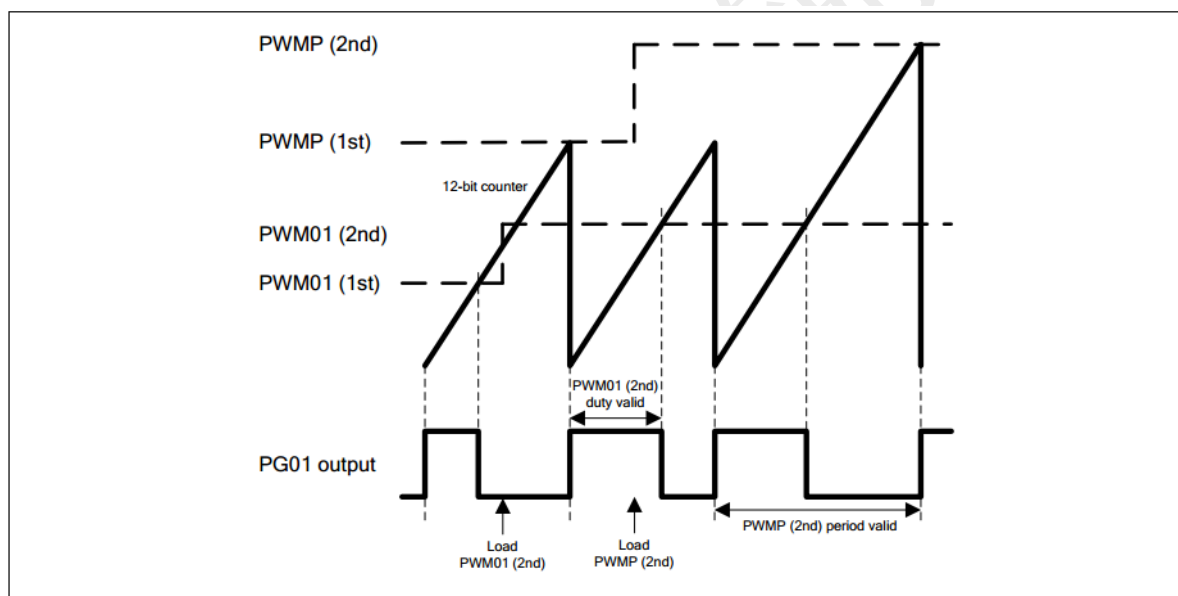


图 14-2-2-1 PWM 边沿对齐方式波形图

PWM 边沿对齐方式输出频率及占空比算式如下：PWM 频率 =  $\frac{F_{PWM}}{\{PWMPH, PWMP\} + 1}$  ( $F_{PWM}$  为时钟源除以 PWMDIV)

$$\text{PWM 占空比高电平} = \frac{\{PWMnH, PWMnL\}}{\{PWMPH, PWMP\} + 1}$$

### 14.2.2.2 中心对齐模式

中心对齐模式，16 位计数器采用双周期模式，从 0000H 开始向上计数至 {PWMOVRH,

PWMOVRL}, 然后由{PWMOVRLH, PWMOVRL}向下计数至 0000H。PGn 信号在 16 位计数器向上计数, 与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配, 然后 16 位计数器向下计数, 与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配, 计数至 0000H 时停止 PGn 信号。中心对齐型 PWM 用于产生非重叠波形。

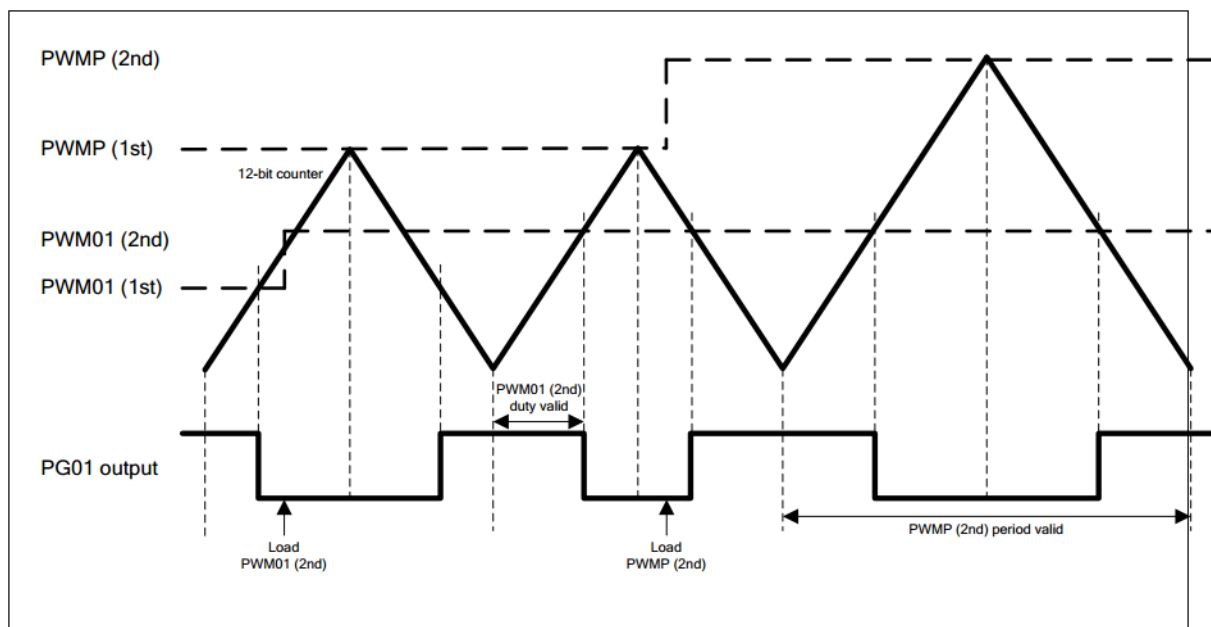


图 14-2-2 PWM 中心对齐模式波形图

中心对齐模式输出频率及占空比算式如下 PWM 频率 =  $\frac{F_{PWM}}{2 \times \{PWMPH, PWMPL\}}$  ( $F_{PWM}$

为时钟源除以 PWMDIV)

$$\text{PWM 占空比高电平} = \frac{\{PWMnH, PWMnL\}}{\{PWMPH, PWMPL\}}$$

### 14.2.3 工作模式

PGn 信号通过 PWM 和故障刹车输出控制电路后, PWM 模式选择电路将产生不同类型的 PWM 输出模式, 总共 6 个通道、三组, 分别是 PG0~PG5。支持独立输出模式, 互补模式及同步模式。

#### 14.2.3.1 独立输出模式

当 PWMMOD[1:0](PWMCON1[7:6])设定为[0:0], PWM 为独立输出模式。该模式为默认输出模式。PG0, PG1, PG2, PG3, PG4 及 PG5 独立输出 PWM 信号。

### 14.2.3.2 带死区插入的互补模式

当  $PWMMOD[1:0] = [0:1]$ ，设定为互补模式。在该模式中 PG0/2/4 输出信号与独立模式下输出信号相同，但 PG1/3/5 输出与 PG0/2/4 输出的信号互补。同时忽略 PG1/3/5 占空比寄存器  $\{PWMnH, PWMnL\}(n:1/3/5)$ 。该模式可使 PG0/PG1 形成一对互补的 PWM 输出。同样 PG2/PG3, PG4/PG5 也可以用于互补输出。

在实际的电机应用中，互补模式 PWM 输出需要插入“死区时间”用来防止损坏电源开关器件，像用于控制半桥连续开关的 GPIBS，用来控制不能同时开关的电源器件。RM1221A 每组 PWM 共享一个 9 位“死区时间”计数器 PDTCNT，用于产生在同组中两通道之间关闭时间，对同组两个 PWM 信号中插入“关闭时间”，同样，在 PDTCNT 定时器溢出，电平从 0 到 1 转换的边沿会加入一段延迟。下图：互补模式 PG0/PG1 组插入“死区时间”的时序图，PG2/PG3 组和 PG4/PG5 组也是同样的“死区时间”时序图。每组是否加入“死区时间”都可以通过 PDTEN[3:0]寄存器来配置。

*注：仅当 PWM 配置为互补模式，“死区时间”控制才有效。*

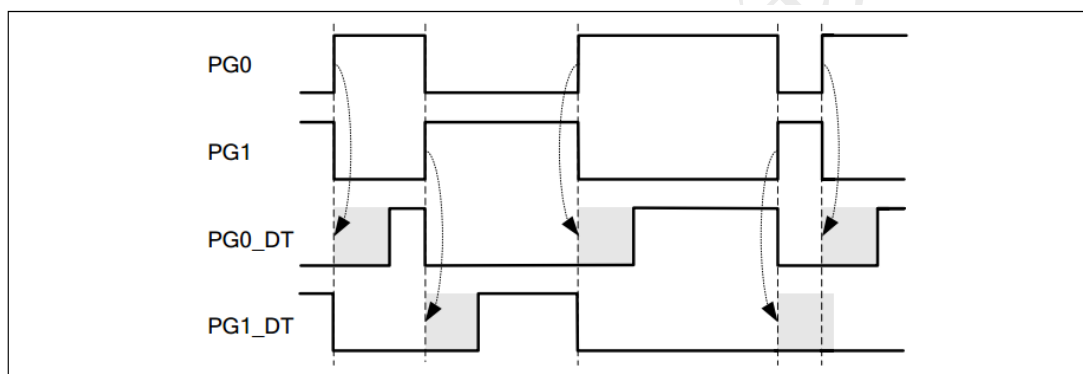


图14-2-3-1 PWM 互补模式死区控制

### 14.2.3.3 同步模式

当  $PWMMOD[1:0] = [1:0]$ ，PWM 选择同步模式。在该模式下 PG0/2/4 信号输出与独立输出模式相同。PG1/3/5 信号与 PG0/2/4 信号也完全相同。

### 14.2.3.4 输出掩码控制

通过软件设置 PWM 的掩码寄存器可以屏蔽每个通道的 PWM 信号，PWM 掩码输出功能广泛应用于电子换向电机如直流无刷电机 BLDC。PMEN 寄存器包含 6 位掩码使能位，每个掩码使能位掩码使能各自的 PWM 通道。PMEN 的默认值为 00H，即所有 PWM 通道输出都不掩码。

注：掩码电位按照 PMD 设定值决定，并不受 PNP 寄存器影响。

## 14.2.4 故障刹车

故障刹车功能应用在增强型 PWM 电路中，配置为输入故障侦测，保护电机系统防止损坏。当 FBINEN(PWMCON1.3)置 1，故障刹车输入脚（FB）生效。当故障发生，PWM 相对应的管脚的 FBD 值会被更改，PWMEN(PWMCON0.7)位自动被硬件清除，PWM 输出停止，PWM16 位计数器复位清 0，标志位 FBF 通过硬件置 1，如果中断使能了将会产生故障刹车中断。即便软件清除 FBF 值，FBD 数据内容仍然保持不变。用户需要重新设置 PWMEN 的值来重新启动输出 PWM 信号。此时，故障刹车状态才会被释放，PWM 按设定值正常输出。故障刹车具有极性选择位 FBINLS(FBD.6)。注意，FB 管脚内部有固定的 8/FSYS 响应过滤结构，FB 脚信号发生需要超过 8 个系统时钟故障刹车才会响应，以避免管脚干扰信号引发误操作。另一个触发故障刹车的方式是 ADC 比较事件，与 FB 脚输入效果相同。

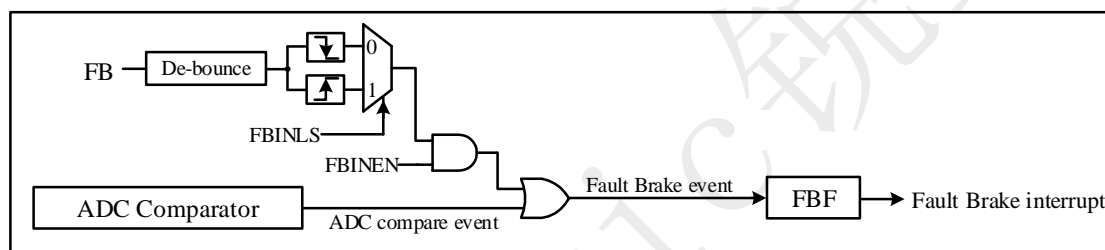


图 14-2-4-1 故障刹车功能模块图

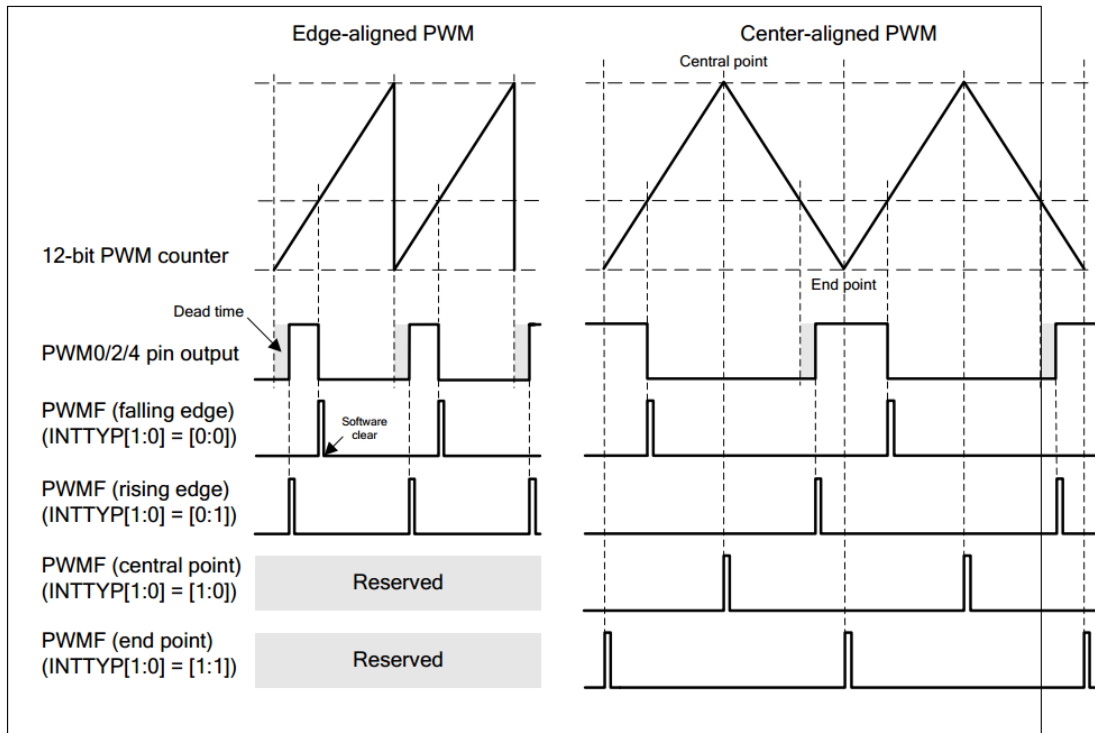
## 14.2.5 极性控制

每路 PWM 带有独立的极性控制位 PNP0~PNP5。默认正逻辑为高电平有效，即 PWM 输出高电平电源切换开，低电平电源切换关。用户可通过设置 PNP 位来改变 PWM 输出极性，产生相反的信号。

## 14.2.6 PWM 中断

PWM 模块带有标志位 PWMF(PWMCON0.5)用来标志当前 PWM 周期完成状态。响应条件根据 INTSEL[2:0]和 INTTYP[1:0](PWMINTC[2:0]和[5:4])设置。注：中心点触发或终点触发中断仅适用于中心对齐模式。PWMF 通过软件清零。

PWM 中断波形如下图：


**图14-2-6 PWM 中断类型**

故障刹车事件请求另一个中断，故障刹车中断。具有独立的中断向量，不同于 PWM 中断向量。无论故障刹车管脚输入事件或是 ADC 比较事件发生，FBF(FBD.7)都会通过硬件置 1，如果故障刹车中断 EFB(EIE.5)使能位使能了，将产生故障刹车中断，FBF 需通过软件清 0。

### 14.3 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>PWMCON0</b>	0xD8	R/W	PWM 控制寄存器 0	00000000
<b>PWMCON1</b>	0xD9	R/W	PWM 控制寄存器 1	00000000
<b>PWMOVRL</b>	0xE2	R/W	PWM 周期低字节寄存器	00000000
<b>PWMOVRLH</b>	0xE3	R/W	PWM 周期高字节寄存器	00000000
<b>PWMD0L</b>	0xD2	R/W	PWM0 占空比低字节寄存器	00000000
<b>PWMD0H</b>	0xD3	R/W	PWM0 占空比高字节寄存器	00000000
<b>PWMD1L</b>	0xD4	R/W	PWM1 占空比低字节寄存器	00000000
<b>PWMD1H</b>	0xD5	R/W	PWM1 占空比高字节寄存器	00000000
<b>PWMD2L</b>	0xD6	R/W	PWM2 占空比低字节寄存器	00000000
<b>PWMD2H</b>	0xD7	R/W	PWM2 占空比高字节寄存器	00000000
<b>PWMD3L</b>	0xDA	R/W	PWM3 占空比低字节寄存器	00000000
<b>PWMD3H</b>	0xDB	R/W	PWM3 占空比高字节寄存器	00000000
<b>PWMD4L</b>	0xDC	R/W	PWM4 占空比低字节寄存器	00000000
<b>PWMD4H</b>	0xDD	R/W	PWM4 占空比高字节寄存器	00000000
<b>PWMD5L</b>	0xDE	R/W	PWM5 占空比低字节寄存器	00000000

<b>PWMD5H</b>	0xDF	R/W	PWM5 占空比高字节寄存器	00000000
<b>PDTEN</b>	0xE6	R/W	PWM 死区控制寄存器	00000000
<b>PDTCNT</b>	0xE7	R/W	PWM 死区时间计数器	00000000
<b>PMEN</b>	0xE1	R/W	PWM 输出掩码控制寄存器	00000000
<b>PMD</b>	0xD1	R/W	PWM 掩码数据寄存器	00000000
<b>FBD</b>	0xE5	R/W	PWM 故障刹车数据寄存器	00000000
<b>PNP</b>	0xE4	R/W	PWM 负极性寄存器	00000000
<b>PWMINTC</b>	0xEF	R/W	PWM 中断控制寄存器	00000000

## 14.4 寄存器描述

### 14.4.1 PWM 控制寄存器 0 (PWMCON0)

寄存器	地址	R/W	描述	复位后的值
<b>PWMCON0</b>	0xD8	R/W	PWM 控制寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PWMRUN</b>	<b>LOAD</b>	<b>PWMF</b>	<b>CLRPWM</b>	-	-	-	-

位	描述	
[7]	<b>PWMRUN</b>	<b>PWM 运行使能位</b> 0: PWM 模块空闲 1: PWM 开始运行
[6]	<b>LOAD</b>	<b>PWM 载入新周期及占空比</b> 该位用于载入周期及占空比所新设定的值。当前一个 PWM 周期输出结束，载入动作才会开始。更改的周期及占空比值将在下一个周期展现。当载入动作结束，硬件自动将 LOAD 位清 0。这一特性会使得 LOAD 位写入及读出值可能不一致，意义也不相同。 <b>写</b> 0: 不动作。 1: 载入之前在缓存内存入的周期及占空比值。 <b>读</b> 0: 载入动作已完成 1: 正在执行载入动作
[5]	<b>PWMF</b>	<b>PWM 标志位</b> 该位根据 PWMINTC 的 INTSEL[2:0] 及 INTTYP[1:0] 位来设定。由软件清 0。
[4]	<b>CLRPWM</b>	<b>清除 PWM 计数器</b> 该位置 1，会直接清零 PWM16 位计数器至 0000H。当清除计数器动作完成，硬件会自动将该位清 0。这特性会使得 CLRPWM 位写入和读出值不一致，意义也不相同。 <b>写</b>



		0: 无动作。 1: 清除 16 位 PWM 计数器。 读 0: PWM16 位计数器已清除。 1: 16 位计数器内还有数值, 未清除。
[3:0]	保留	保留

#### 14.4.2 PWM 控制寄存器 1 ( PWMCON1)

寄存器	地址	R/W	描述	复位后的值
PWMCON1	0xD9	R/W	PWM 控制寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		

位	描述	
[7:6]	PWMMOD[1:0]	<b>PWM 模式选择</b> 00: 独立输出模式 01: 互补模式 10: 同步模式 11: 保留位
[5]	GP	<b>群组模式使能位</b> 该位使能 PWM 群组模式。一旦使能, 三对 PWM 的占空比由 PWM01H 和 PWM01L 决定, 原本配置的数据失效。 0: 未组成群组模式。 1: 群组模式使能
[4]	PWMTYP	<b>PWM 类型选择</b> 0: 边沿对齐型 PWM 1: 中心对齐型 PWM
[3]	FBINEN	<b>FB 管脚输入使能位</b> 0: 禁止 1: 通过 FB 管脚输入使能 PWM 故障刹车功能。一旦在 FB 管脚输入的边沿信号与 FBINLS (FBD.6)选择位相同。PWM0~5 输出 FBD 寄存器所设置的信号电平, PWMRUN (PWMCON0.7) 位硬件清除。当 PWMRUN 信号重置 1, PWM 信号重新输出。
[2:0]	PWMDIV[2:0]	<b>PWM 时钟除频</b> 该寄存器段用于配置 PWM 时钟频率预分频。 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64

### 14.4.3 PWM 周期低字节寄存器 (PWMOVRL)

寄存器	地址	R/W	描述	复位后的值
PWMOVRL	0xE2	R/W	PWM 周期低字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWMOVRL[7:0]							

位	描述	
[7:0]	PWMOVRL[7:0]	PWM 周期值低字节 该寄存器与 PWMOVRH 搭配 PWM 的周期值，该位段为低字节

### 14.4.4 PWM 周期高字节寄存器 (PWMOVRH)

寄存器	地址	R/W	描述	复位后的值
PWMOVRH	0xE3	R/W	PWM 周期高字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWMOVRH[7:0]							

位	描述	
[7:0]	PWMOVRH[7:0]	PWM 周期值高字节 该寄存器与 PWMOVRL 搭配 PWM 的周期值，该位段为高字节

### 14.4.5 占空比低字节寄存器 (PWMD0L /PWMD1L /PWMD2L /PWMD3L /PWMD4L /PWMD5L)

寄存器	地址	R/W	描述	复位后的值
PWMD0L	0xD2	R/W	PWM0 占空比低字节寄存器	00000000
PWMD1L	0xD4	R/W	PWM1 占空比低字节寄存器	00000000
PWMD2L	0xD6	R/W	PWM2 占空比低字节寄存器	00000000
PWMD3L	0xDA	R/W	PWM3 占空比低字节寄存器	00000000
PWMD4L	0xDC	R/W	PWM4 占空比低字节寄存器	00000000
PWMD5L	0xDE	R/W	PWM5 占空比低字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWMDxL[7:0] x=0~5							

位	描述	
[7:0]	PWMDxL[7:0] x=0~5	PWMDx 占空比低字节 该寄存器存储与 PWMDxH 搭配存储 PWM 的占空比值，该位段为低字节

#### 14.4.6 占空比高字节寄存器 (PWMD0H /PWMD1H /PWMD2H /PWMD3H /PWMD4H /PWMD5H)

寄存器	地址	R/W	描述	复位后的值
PWMD0H	0xD3	R/W	PWM0 占空比高字节寄存器	00000000
PWMD1H	0xD5	R/W	PWM1 占空比高字节寄存器	00000000
PWMD2H	0xD7	R/W	PWM2 占空比高字节寄存器	00000000
PWMD3H	0xDB	R/W	PWM3 占空比高字节寄存器	00000000
PWMD4H	0xDD	R/W	PWM4 占空比高字节寄存器	00000000
PWMD5H	0xDF	R/W	PWM5 占空比高字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWMDxH[7:0] x=0~5							

位	描述	
[7:0]	PWMDxH[7:0] x=0~5	PWMDx 占空比高字节 该寄存器存储与 PWMDxL 搭配存储 PWM 的占空比值，该位段为高字节

#### 14.4.7 PWM 死区控制寄存器 (PD TEN)

寄存器	地址	R/W	描述	复位后的值
PD TEN	0xE6	R/W	PWM 死区控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	-	PDTCNT.8	-	PDT45EN	PDT23EN	PDT01EN

位	描述	
[7:5]	保留	保留

[4]	<b>PDTCNT.8</b>	<b>PWM 8 位死区时间计数器</b> 详见 PDTCNT 寄存器
[3]	保留	保留
[2]	<b>PDT45EN</b>	<b>PWM4/5 死区时间使能位</b> 仅当 PWM4/5 配置位互补模式，死区功能才会生效。 0: GP4/GP5 信号无延时 1: 在 GP4/GP5 信号上升沿加入死区时间延时
[1]	<b>PDT23EN</b>	<b>PWM2/3 死区时间使能位</b> 仅当 PWM2/3 配置位互补模式，死区功能才会生效。 0: GP2/GP3 信号无延时 1: 在 GP2/GP3 信号上升沿加入死区时间延时
[0]	<b>PDT01EN</b>	<b>PWM0/1 死区时间使能位</b> 仅当 PWM0/1 配置位互补模式，死区功能才会生效。 0: GP0/GP1 信号无延时 1: 在 GP0/GP1 信号上升沿加入死区时间延时

#### 14.4.8 PWM 死区时间计数器 (PDTCNT)

寄存器	地址	R/W	描述	复位后的值
<b>PDTCNT</b>	0xE7	R/W	PWM 死区低字节寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PDTCNT[7:0]</b>							

位	描述
[7:0]	<p><b>PWM 死区时间计数器低字节</b></p> <p>该 8 位寄存器与 PDTEN.4 组成 9 位 PWM 死区时间计数器 PDTCNT。该计数器仅当 PWM 设定位互补模式，且有效死区使能位已设置时有效。</p> $\text{PWM 死区时间} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ <p>注：在 PWM 运行过程中，请勿更改 PDTCNT 的值</p>

#### 14.4.9 PWM 输出掩码控制寄存器 (PMEN)

寄存器	地址	R/W	描述	复位后的值
<b>PMEN</b>	0xE1	R/W	PWM 输出掩码控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	<b>PMEN5</b>	<b>PMEN4</b>	<b>PMEN3</b>	<b>PMEN2</b>	<b>PMEN1</b>	<b>PMEN0</b>

位	描述
[7:6]	保留

[x] x=0~5	<b>PMENx</b> x=0~5	<b>PWMx 输出掩码使能位</b> 0: PWMx 信号输出 1: PWMx 根据 PMDx 设定的电位值掩码
--------------	-----------------------	---

#### 14.4.10 PWM 掩码数据寄存器 (PMD)

寄存器	地址	R/W	描述	复位后的值
<b>PMD</b>	0xD1	R/W	PWM 掩码数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0

位	描述	
[7:6]	保留	保留
[x] x=0~5	<b>PMDx</b> x=0~5	<b>PWMx 掩码数据寄存器</b> 一旦相应的 PMENx 被设置, PWMx 信号输出掩码电平值 0: PWMx 掩码, 输出低电平 1: PWMx 掩码, 输出高电平

#### 14.4.11 PWM 故障刹车数据寄存器 (FBD)

寄存器	地址	R/W	描述	复位后的值
<b>FBD</b>	0xE5	R/W	PWM 故障刹车数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>FBF</b>	<b>FBINLS</b>	<b>FBD5</b>	<b>FBD4</b>	<b>FBD3</b>	<b>FBD2</b>	<b>FBD1</b>	<b>FBD0</b>

位	描述	
[7]	<b>FBF</b>	<b>故障刹车标志位</b> 当 FBINEN 设置为 1, FB 管脚上检测到符合 FBINLS (FBD.6) 设定的边沿信号后, 该位置 1。该位需要通过软件清 0。当 FBF 清 0 后, 故障刹车仍然不会释放 PWM 输出, 需要重新输出, 除非设置 PWMRUN (PWMCON0.7) 为 1, 重新启动 PWM 输出。
[6]	<b>FBINLS</b>	<b>FB 管脚输入选择</b> 0: 下降沿 1: 上升沿
[x] x=0~5	<b>FBDx</b> x=0~5	<b>PWMx 故障刹车数据寄存器</b> 0: 当故障发生时 PWMx 信号输出为低电平 1: 当故障发生时 PWMx 信号输出为高电平

### 14.4.12 PWM 负极性寄存器 (PNP)

寄存器	地址	R/W	描述	复位后的值
PNP	0xE4	R/W	PWM 负极性寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0

位	描述	
[7:6]	保留	保留
[x] x=0~5	PNP <sub>x</sub> x=0~5	<b>PWM<sub>x</sub> 负极性输出使能</b> 0: PWM <sub>x</sub> 输出按照设定直接输出到 PWM <sub>x</sub> 管脚 1: PWM <sub>x</sub> 输出按照设定取反输出到 PWM <sub>x</sub> 管脚

### 14.4.13 PWM 中断控制寄存器 (PWMINTC)

寄存器	地址	R/W	描述	复位后的值
PWMINTC	0xEF	R/W	PWM 中断控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	INTTYP[1:0]		-	INTSEL[2:0]		

位	描述	
[7:6]	保留	保留
[5:4]	INTTYP[1:0]	<b>PWM 中断类型选择</b> 通过该位选择 PWM 的中断类型 00: PWM0/1/2/3/4/5 脚下降沿. 01: PWM0/1/2/3/4/5 脚上升沿. 10: 每个 PWM 周期的中心点. 11: 每个 PWM 周期的终点. 注: 中心点中断方式或终点中断方式仅适用于 PWM 中心对齐模式
[3]	保留	保留
[2:0]	INTSEL[2:0]	<b>PWM 中断对选择</b> 在 PWM0/1/2/3/4/5 脚, 当 PWM 已选择中断类型为上升沿或下降沿时, 该位段用以选择中断响应所相对的 PWM 脚。 000: PWM0 001: PWM1 010: PWM2 011: PWM3 100: PWM4

	101: PWM5 其它: PWM0
--	-----------------------

## 15 看门狗定时器 (WDT)

### 15.1 概述

设计看门狗定时器的目的是，当系统运行到一个未知状态时，通过它来使系统复位。这种做法可以预防系统进入到无限期的死循环。此外，看门狗定时器还支持系统从空闲/休眠模式唤醒功能。RM1221A 内置定时溢出寄存器 WDTR 和分频配置位 WDTCKPS，可以灵活配置不同的溢出时间，来满足用户需求。

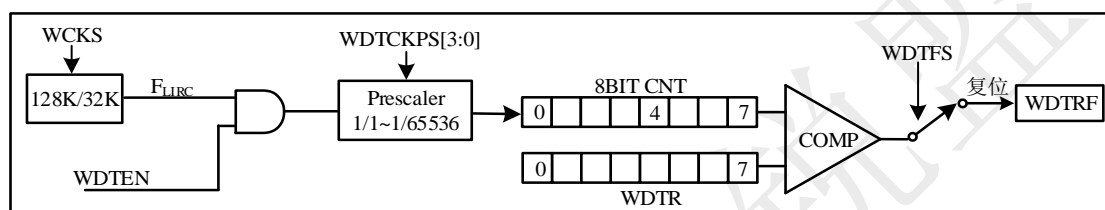


图 15-1 看门狗定时器结构图

### 15.2 功能描述

WDTFS 配置为 0 时，看门狗为复位功能，WDTCKPS 配置定时分频，WDTR 配置定时溢出值，WDT 使能开启后，定时到溢出时间，WDTRF 会置 1，芯片会复位。

### 15.3 寄存器描述

#### 15.3.1 WDT 配置寄存器(WDTCON)

寄存器	地址	R/W	描述	复位后的值
WDTCON	0x8E	R/W	WDT 配置寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
WDTRF	CLRWDT	WDTCKPS[3:0]				WDTFS	WDTEN

位	描述
[7]	<b>WDTRF</b> 看门狗复位标志位，看门狗产生复位时，此位置 1，软件清 0
[6]	<b>CLRWDT</b> 置 1，清看门狗计数，硬件自动清 0
[5:2]	<b>WDTCKPS[3:0]</b> 看门狗定时分频 0000: $F_{LIRC}/65536$ 0001: $F_{LIRC}/32768$



		0010: FLIRC/16384 0011: FLIRC/8192 0100: FLIRC/4096 0101: FLIRC/2048 0110: FLIRC/1024 0111: FLIRC/512 1000: FLIRC/256 1001: FLIRC/128 1010: FLIRC/64 1011: FLIRC/32 1100: FLIRC/16 1101: FLIRC/8 1110: FLIRC/4 1111: FLIRC/2
[1]	<b>WDTFS</b>	看门狗功能选择位 0: 复位功能 1: 无效
[0]	<b>WDTEN</b>	WDT 使能位 0: 禁止看门狗 1: 使能看门狗

### 15.3.2 WDT 定时溢出寄存器(WDTR)

寄存器	地址	R/W	描述	复位后的值
<b>WDTR</b>	0xF044	R/W	WDT 定时溢出寄存器	11111111

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>WDTR[7:0]</b>							

位	描述
[7:0]	<b>WDTR[7:0]</b> 看门狗定时溢出时间配置寄存器

## 16 自唤醒定时器 (WKT)

RM1221A 有一个专用的自唤醒定时器 (WKT)，用于低功耗模式下的周期唤醒芯片，也可用作通用定时器。WKT 保持计数在空闲或掉电模式。当 WKT 用作唤醒定时器时，WKT 要在进入省电模式之前开启。WKT 只能配置片内 128KHz/32KHz 时钟源 LIRC。注意系统时钟频率必须大于 WKT 时钟两倍以上。如果 WKT 开始计数，在设备进入空闲或掉电模式下，选择的时钟源会也要保持工作。注意选择的 WKT 时钟源不会连同 WKT 的配置自动使能，用户应该手动使能选择的时钟源并等待它稳定确保操作的成功。

WKT 配备了一个简单的 8 位自动重载向上计数定时器。它的预分频可选择从 1/1 到 1/2048，通过 WKPS[2:0] (WKCON[2:0])来设置。用户填重装载值到 RWK 寄存器来决定它的溢出速率。WKTR(WKCON.3)置位开始计数。当计数器溢出，WKTIF(WKCON.4)置为 1，并重载 RWK 寄存器的值到内部 8 为计数器。如果 EWKT(EIE1.2)置为 1，WKT 中断服务程序将被执行。

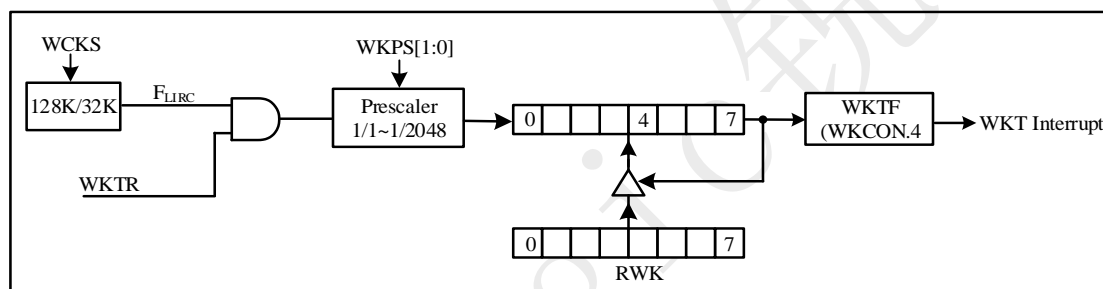


图16-1 自唤醒定时器结构图

### 16.1 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>WKCON</b>	0xF040	R/W	WKT 控制寄存器	00000000
<b>RWK</b>	0xF041	R/W	WKT 重装载数据寄存器	00000000

### 16.2 寄存器描述

#### 16.2.1 WKT 控制寄存器 (WKCON)

寄存器	地址	R/W	描述	复位后的值
<b>WKCON</b>	0xF040	R/W	WKT 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留			<b>WKTIF</b>	<b>WKTR</b>	<b>WKPS[2:0]</b>		

位	描述	
[7:5]	保留	保留
[4]	<b>WKTF</b>	<b>WKT 溢出标志</b> 当 WKT 溢出，该位置位。如果 WKT 中断和全局中断使能，置位该位会使 CPU 执行 WKT 中断服务程序。该位不会被硬件自动清零，应该通过软件清零。
[3]	<b>WKTR</b>	<b>WKT 运行控制</b> 0: WKT 停止 1: WKT 开始运行 注意重载寄存器 RWK 仅在 WKT 停止的时候可以写入（WKTR 位为 0）。否则结果是不可预知的。
[2:0]	<b>WKPS[2:0]</b>	<b>WKT 预分频</b> 这些位决定 WKT 时钟的预分频 000: 1/1 001: 1/4 010: 1/16 011: 1/64 100: 1/256 101: 1/512 110: 1/1024 111: 1/2048

## 16.2.2 WKT 重载数据寄存器 (RWK)

寄存器	地址	R/W	描述	复位后的值
<b>RWK</b>	0xF041	R/W	WKT 重载数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>RWK [7:0]</b>							

位	描述	
[7:0]	<b>RWK [7:0]</b>	<b>WKT 重载字节</b> 用以保存 WKT 的 8 位重载值。注意如果预分频是 1/1，RWK 限制不能是 FFH。

## 17 通用异步收发器 (UART0/1)

### 17.1 概述

RM1221A 提供两个可编程全双工串行通信接口,该接口能同时进行数据的发送和接收。

串口接收带有接收缓存,意味着在接收的前一个数据在被读取之前,串口就能接收第二个数据。接收和发送都是对 SxBUF 进行操作访问,写入 SxBUF 数据将直接传到发送寄存器,而读取 SxBUF 是访问一个具有独立物理地址的接收寄存器。

### 17.2 功能描述

#### 17.2.1 模式 1

模式 1 (8 位 UART) 为异步全双工的工作方式。异步通讯模式通常用于 PC 间,调制解调器和其它类似接口间通讯。模式 1 下,10 位数据通过 TXD 发送,通过 RXD 接收。10 位数据组成如下:起始位(逻辑 0),8 位数据(最低位在前),停止位(逻辑 1)。波特率由 SxOVR[10:0]决定。图 17-2-1 为串口模式 1 发送和接收的时序图。

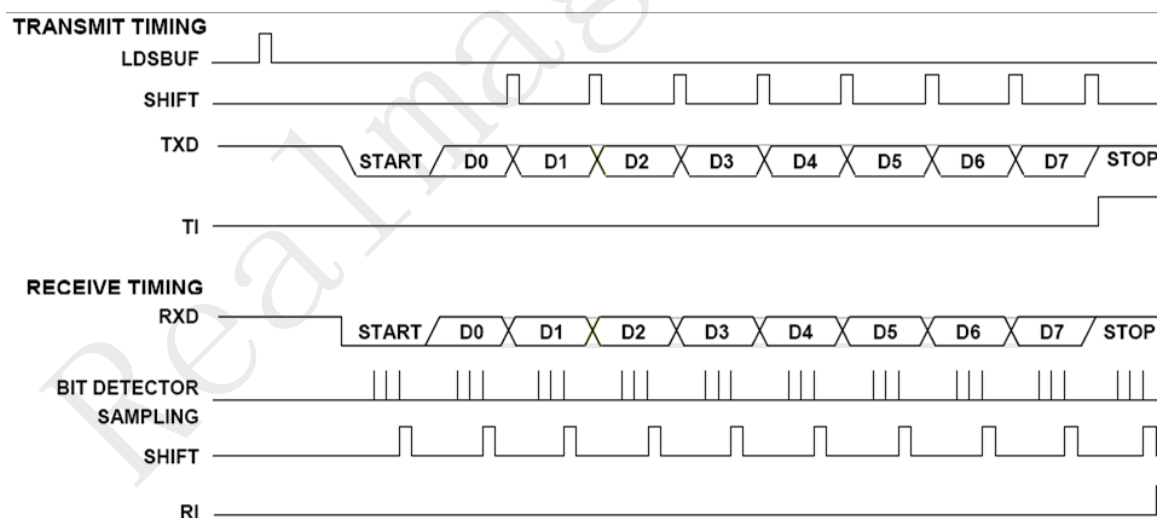


图17-2-1 串口模式1的时序图

任何将 SxBUF 作为目标寄存器的写操作都会启动发送,数据位时间与波特率移位时钟是同步的,与对 SxBUF 的写操作不同步。起始位首先在 TXD 引脚上移出,然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后,停止位在 TXD 引脚上移出,在停止位发出的同时 SxTI (SxCON.1) 标志置 1。SxTI (SxCON.1) 可以软件清掉。

当 RXD 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RXD 数据位的正中进行采样收。如果所接收的第一位不是 0,说明这位不是一帧数据的起始位,该位被忽略,接

收电路被复位，等待 RXD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被装入 SxBUF，SxRI (SxCON.0) 置 1。

在接收停止位后，接收器将重新去探测 RXD 端的另一个下降沿。用户必须用软件清 SxRI (SxCON.0)，然后才能再次接收。

## 17.2.2 模式 2

模式 2 (9 位 UART) 为全双工异步通信，与模式 1 不同的是，模式 2 是 11 位收发。数据由起始位 (逻辑 0)，8 位数据 (最低位在前)，第 9 位数据 (SxTB8 或 SxRB8) 和停止位 (逻辑 1) 组成。第 9 位做奇偶校验位或多机通信时用来区分数据和地址。波特率由 SxOVR[10:0] 决定。图 17-2-2 指示串口模式 2 的传输时序。

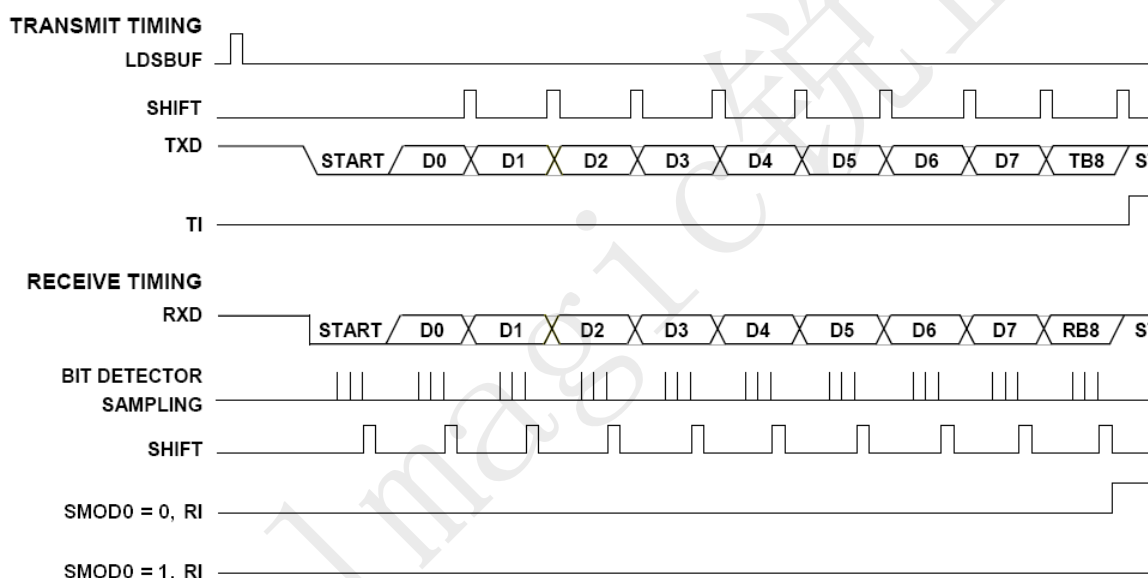


图17-2-2 串口模式2的传输时序

在数据传送时，第 9 数据位 (寄存器 SxCON 中的 SxTB9) 可以写 0 或 1，例如，可写入奇偶位 P，或用作多机通讯中的数据/地址标志位。当接收到数据时，第 9 数据位移入 SxRB9 而停止位不保存。

任何将 SxBUF 作为目标寄存器的写操作都会启动发送，同时也将 SxTB9 载入到发送移位寄存器的第 9 位中。数据位时间与波特率移位时钟是同步的，与对 SxBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TXD 引脚上移出，在停止位发出的同时 TI (SxCON.1) 标志置 1。TI (SxCON.1) 可以软件清掉。

当 RXD 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RXD 数据位的正中进行采样接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RXD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SxBUF 和 SxRB8 中，SxRI (SxCON.0) 置 1。

在接收停止位后，接收器将重新去探测 RXD 端的另一个下降沿。用户必须用软件清 SxRI (SxCON.0)，然后才能再次接收。

### 17.2.3 波特率

UART 实质上是一个 16 位递增计数器，其计数时钟为 FSYS。用户软件配置寄存器 SxOVRH 和 SxOVRL 来软件配置计数器的溢出值，从而实现需要的波特率。

$$\text{波特率计算公式: 波特率} = \frac{FSYS}{SxOVR[10:0]}。$$

## 17.3 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>S0CON</b>	0x98	R/W	串口 0 控制寄存器	00000000
<b>S0BUF</b>	0x99	R/W	串口 0 数据缓存选择	00000000
<b>S0OVRH</b>	0x9A	R/W	串口 0 波特率低字节	00000000
<b>S0OVRH</b>	0x9B	R/W	串口 0 波特率高字节	00000000
<b>S1CON</b>	0xF048	R/W	串口 1 控制寄存器	00000000
<b>S1BUF</b>	0xF049	R/W	串口 1 数据缓存选择	00000000
<b>S1OVRH</b>	0xF04A	R/W	串口 1 波特率低字节	00000000
<b>S1OVRH</b>	0xF04B	R/W	串口 1 波特率高字节	00000000

## 17.4 寄存器描述

### 17.4.1 串口 0 控制寄存器 (S0CON)

寄存器	地址	R/W	描述	复位后的值
<b>S0CON</b>	0x98	R/W	串口 0 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	<b>S0B8EN</b>	<b>S0REN</b>	<b>S0TB8</b>	<b>S0RB8</b>	<b>S0TI</b>	<b>S0RI</b>

位	描述	
[7:6]	保留	保留
[5]	<b>S0B8EN</b>	串口第 9 位使能位 0: 关闭串口第 9 位 1: 使能串口第 9 位
[4]	<b>S0REN</b>	串口使能位 0: 关闭串口

		1: 使能串口
[3]	<b>S0TB8</b>	<b>串口第 9 位发送位</b> 串口要被发送的第九位数据
[2]	<b>S0RB8</b>	<b>串口第 9 位接收位</b> 串口接收到的第九位数据
[1]	<b>S0TI</b>	<b>串口发送中断标志位</b> 发送中断标志: 该标志由硬件在发送完 8 位数据后置 1。当该位中断使能, 发生中断后会转至中断子程序。该位必须由软件来清除。
[0]	<b>S0RI</b>	<b>串口接收中断标志</b> 该标志由硬件置位。当串口中断使能, 该位置位会转跳到中断服务子程序运行。该位必须由软件来清除。

### 17.4.2 串口 0 数据缓存寄存器 (S0BUF)

寄存器	地址	R/W	描述	复位后的值
<b>S0BUF</b>	0x99	R/W	串口 0 数据缓存寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>S0BUF[7:0]</b>							

位	描述	
[7:0]	<b>S0BUF[7:0]</b>	<b>串口 0 数据缓存</b> 串口接收或发送的数据都放在这个寄存器中。实际上该地址上有 2 个独立的 8 位寄存器。一个用于接收数据, 一个用于发送数据。对它进行读操作将会接收串行数据, 对它进行写操作则发送串行数据。 每次向 S0BUF 写入一字节数据, 启动一次发送。

### 17.4.3 串口 0 波特率低字节 (S0OVR)

寄存器	地址	R/W	描述	复位后的值
<b>S0OVR</b>	0x9A	R/W	串口 0 波特率低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>S0OVR[7:0]</b>							

位	描述	
[7:0]	<b>S0OVR[7:0]</b>	<b>串口 0 波特率低字节</b>

### 17.4.4 串口 0 波特率高字节 (S0OVRH)

寄存器	地址	R/W	描述	复位后的值
<b>S0OVRH</b>	0x9B	R/W	串口 0 波特率高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-					<b>S0OVR[10:8]</b>		

位	描述	
[7:3]	保留	保留
[2:0]	<b>S0OVR[10:8]</b>	串口 0 波特率高字节

### 17.4.5 串口 1 控制寄存器 (S1CON)

寄存器	地址	R/W	描述	复位后的值
<b>S1CON</b>	0xF048	R/W	串口 1 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	<b>S1B8EN</b>	<b>S1REN</b>	<b>S1TB8</b>	<b>S1RB8</b>	<b>S1TI</b>	<b>S1RI</b>

位	描述	
[7:6]	保留	保留
[5]	<b>S1B8EN</b>	串口 1 第 9 位使能位 0: 关闭串口第 9 位 1: 使能串口第 9 位
[4]	<b>S1REN</b>	串口 1 使能位 0: 关闭串口 1: 使能串口
[3]	<b>S1TB8</b>	串口 1 第 9 位发送位 串口要被发送的第九位数据
[2]	<b>S1RB8</b>	串口 1 第 9 位接收位 串口接收到的第九位数据
[1]	<b>S1TI</b>	串口 1 发送中断标志位 发送中断标志: 该标志由硬件在发送完 8 位数据后置 1。当该位中断使能, 发生中断后会转至中断子程序。该位必须由软件来清除。
[0]	<b>S1RI</b>	串口 1 接收中断标志 该标志由硬件置位。当串口中断使能, 该位置位会转跳到中断服务子程序运行。该位必须由软件来清除。



### 17.4.6 串口 1 数据缓存寄存器 (S1BUF)

寄存器	地址	R/W	描述	复位后的值
<b>S1BUF</b>	0xF049	R/W	串口 1 数据缓存寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>S1BUF[7:0]</b>							

位	描述	
[7:0]	<b>S1BUF[7:0]</b>	<b>串口 1 数据缓存</b> 串口接收或发送的数据都放在这个寄存器中。实际上该地址上有 2 个独立的 8 位寄存器。一个用于接收数据，一个用于发送数据。对它进行读操作将会接收串行数据，对它进行写操作则发送串行数据。 每次向 S1BUF 写入一字节数据，启动一次发送。

### 17.4.7 串口 1 波特率低字节 (S1OVR)

寄存器	地址	R/W	描述	复位后的值
<b>S1OVR</b>	0xF04A	R/W	串口 1 波特率低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>S1OVR[7:0]</b>							

位	描述	
[7:0]	<b>S1OVR[7:0]</b>	串口 1 波特率低字节

### 17.4.8 串口 1 波特率高字节 (S1OVRH)

寄存器	地址	R/W	描述	复位后的值
<b>S1OVRH</b>	0xF04B	R/W	串口 1 波特率高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-					<b>S1OVR[10:8]</b>		

位	描述	
[7:3]	保留	保留
[2:0]	<b>S1OVR[10:8]</b>	串口 1 波特率高字节

## 18 串行外设接口 (SPI)

### 18.1 概述

SPI 为微控制与外设 EEPROM, LCD 驱动, D/A 转换之间提供全双工、高速、同步传输的总线。可提供主机从机模式传输, 速度可达到时钟频率  $F_{CPU}/2$ , 支持传输完成标志位和“写”冲突标志位。在多主机系统中, SPI 支持主机模式错误用以防止主机冲突。

### 18.2 功能描述

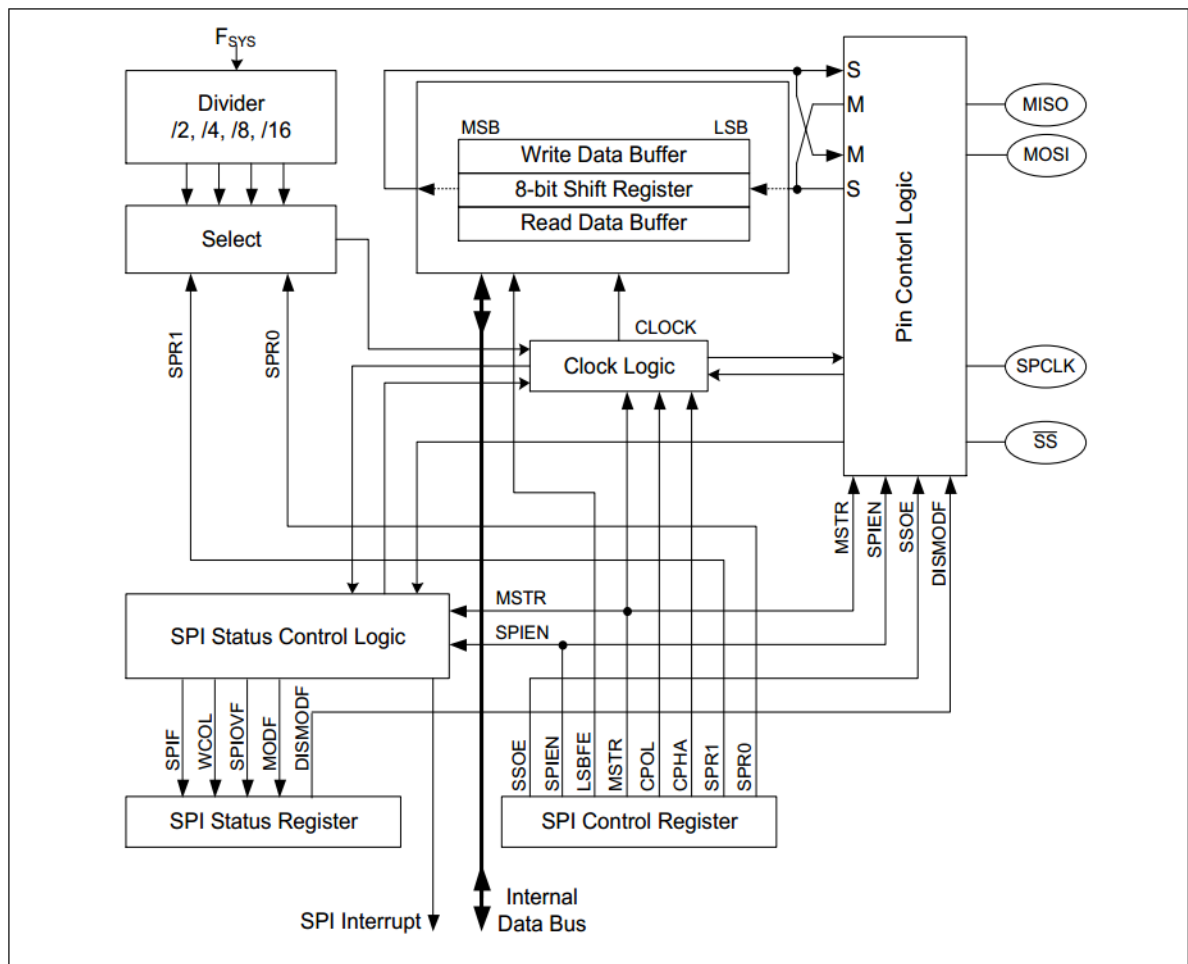


图 18-2-1. SPI 结构功能图

SPI 的主要模块有 SPI 控制寄存器, SPI 状态逻辑, 波特率控制和管脚逻辑控制。为了传送数据和接收数据, SPI 提供了移位寄存器和读数据缓冲器。因为无论传送数据或是接收数据都是双缓存, 所以传输端时, 在前一个数据发送完成前, 也可以写入下一笔把数据。接

收端能读取读数据缓存，在移位寄存器接收第二个数据时，同时前一个接收的数据将被传送到读数据缓存。

SPI 接口有四个管脚，分别是主进/从出(MISO)，主出/从进(MOSI)，移位时钟(SPCLK)，和从机选择 NSS。MOSI 脚用于传输主机到从机的 8 位数据，所以 MOSI 是一个主机设备的输出引脚，从机设备的输入引脚。相应的，MISO 用于接收从机到主机的串行数据。

SPCLK 引脚为主机模式下的时钟输出，从机的输入时钟。移位时钟用于 MOSI 和 MISO 脚之间数据传输的同步时钟。主机模式发送 8 个移位时钟周期，在总线上交换一个字节数据。移位时钟由主机输出，所以一组 SPI 传输系统上只能有一个主机以避免设备冲突。

从机设备通过设定从机选择脚 NSS 选择。当需要访问任何从机时，该从机的此信号脚必须保持低。当 NSS 为高，该从机访问将被禁止。若为多从机模式，在同一时刻必须保持只有一个从机被选定。对于主机，NSS 脚不做任何用途，可配置为普通端口另做他用。NSS 可用于多主机模式下主机模式错误侦测功能。

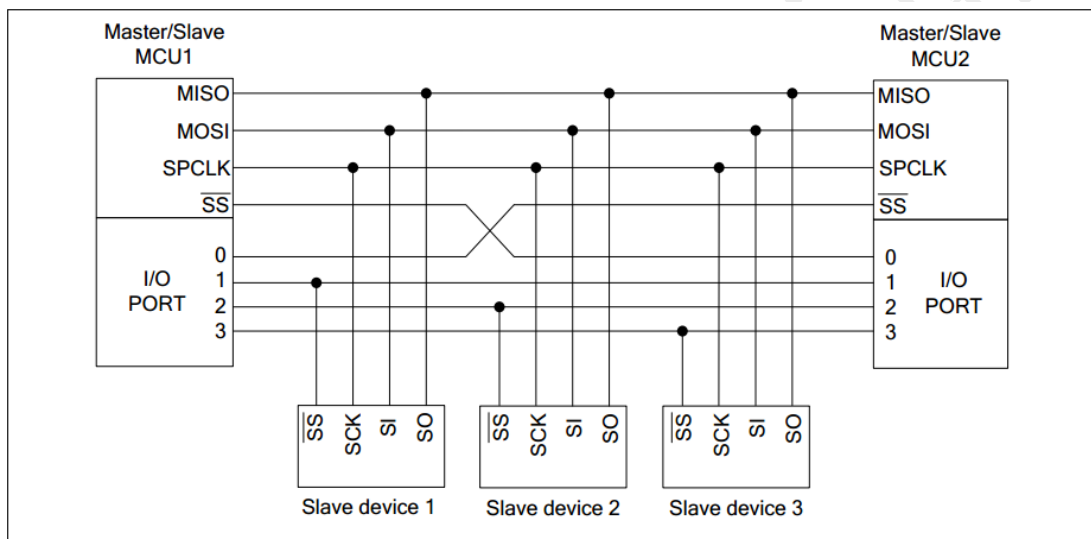


图 18-2-2 SPI 多主机，多从机连接图

通信总线通过 3 根信号线相连，MOSI ~ MOSI，MISO ~ MISO，和 SPCLK~SPCLK。主机通过一个并口的 4 个管脚来控制 4 个 NSS 脚，从而实现每个 NSS 线分别控制每个从机。MCU1 和 MCU2 可以任意定义为主机或从机模式。NSS 需配置为主机模式侦测功能避免多主机冲突。

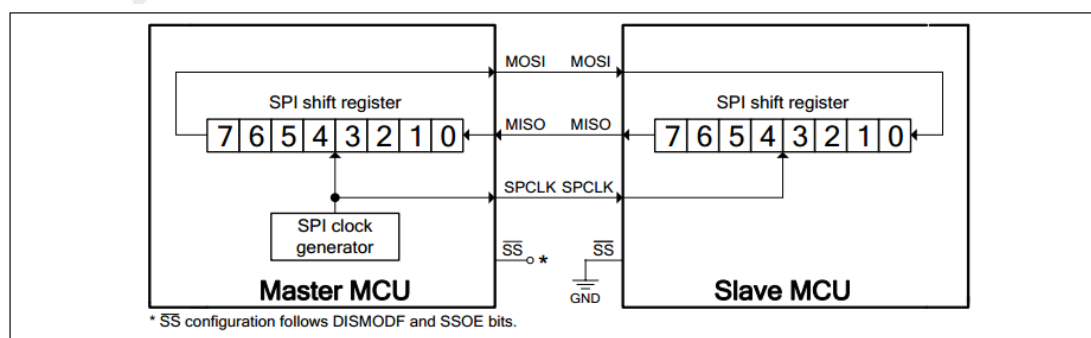


图 18-2-3 SPI 单主机，单从机连接图

在传输时，主机通过 MOSI 线向从机发送数据。同时，主机也通过 MISO 线由从机接收数据。此时主机和从机的两个移位寄存器可被视为一个 16 位的循环移位寄存器。因此，当主机向从机发送数据时，从机数据也同时推向主机。这样通过两个 MCU 的 SPI 移位寄存器，就完成了交换数据。

控制寄存器（SPCR），SPI 状态寄存器（SPSR），SPI 数据寄存器（SPDR）这三个寄存器用于 SPI 传输。这些寄存器提供控制，状态检测，数据存储以及时钟发生设置。

### 18.2.1 主机模式

MSTR (SPSR.4)位置 1，SPI 工作在主机模式。整个 SPI 系统中只允许一个主机启动传输。每次传输总是由主机发起，对主机 SPDR 寄存器的写开始传送。在 SPCLK 控制下在 MOSI 管脚传送数据。同时，MISO 管脚接收数据。在 8 位数据传输完毕后，SPIF(SPSR.7)由硬件自动置位以示完成一个字节数据传输。同时接收到的数据也会传送到 SPDR。用户可以从 SPDR 读出数据，并清除 SPIF。

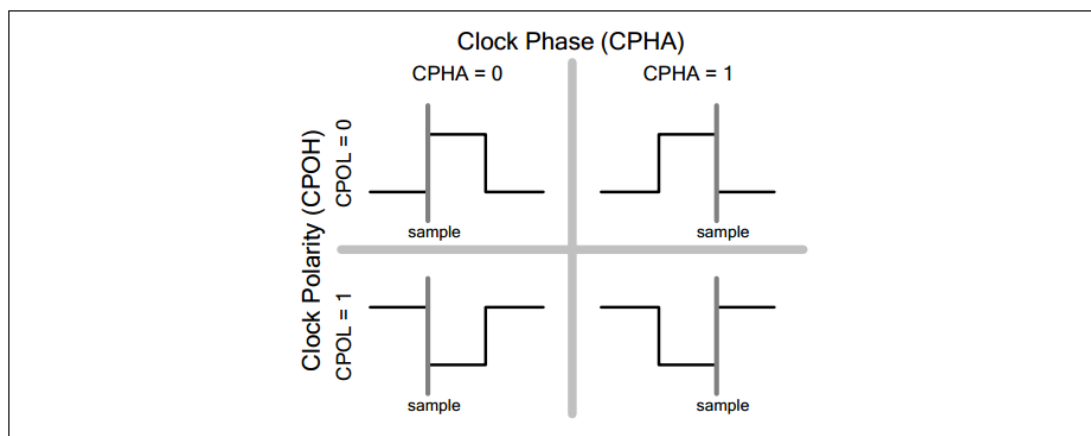
### 18.2.2 从机模式

设定 MSTR 为 0，SPI 将工作在从机模式。当作为从机模式时，SPCLK 管脚变为输入脚，它将作为时钟输入被另外一个主机 SPI 设备的输出时钟控制，NSS 管脚也变为输入脚。当从机设备的 NSS 管脚不为低时，主机设备不能与从机交换数据。在数据传输开始前和数据传输完成前，NSS 管脚都需要保持低电平状态。如果 NSS 变为高电平，SPI 将被迫进入闲置状态。如果 NSS 管脚在传输的过程被置高，那么传输将被取消，接收移位缓存区里剩下的位数将变高，同时也将进入闲置状态。

在从机模式下，数据通过 MOSI 管脚从主机向从机传输，通过 MISO 管脚从从机向主机传输。通过主机 SPCLK 的时钟控制，数据进入位移寄存器。在移位寄存器接收到一个字节后，数据将移到读数据缓存，同时 SPIF 置 1。对 SPDR 的读操作实际上就是对读缓冲器的一次读操作。为了防止缓冲器溢出或因溢出导致数据丢失，从设备必须在数据第二次从移位寄存器向读缓冲器传送前，把数据从 SPDR 读出和把 SPIF 清零。

### 18.2.3 时钟格式和数据传输

为了适应各种各样的同步串行外设，SPI 提供时钟极性位 CPOL (SPCR.3) 和时钟相位 CPHA (SPCR.2) 寄存器用以控制。如图 18-2-4 所示，CPOL 和 CPHA 组合出四种不同的时钟格式。CPOL 位表示空闲状态时 SPCLK 脚电平。CPHA 位定义表示是 MOSI 和 MISO 上时钟的哪个边沿用来采样。在同一系统上的主从设备中，CPOL 和 CPHA 的配置应该是相同的。传输不同的数据格式，将产生随机错误结果。

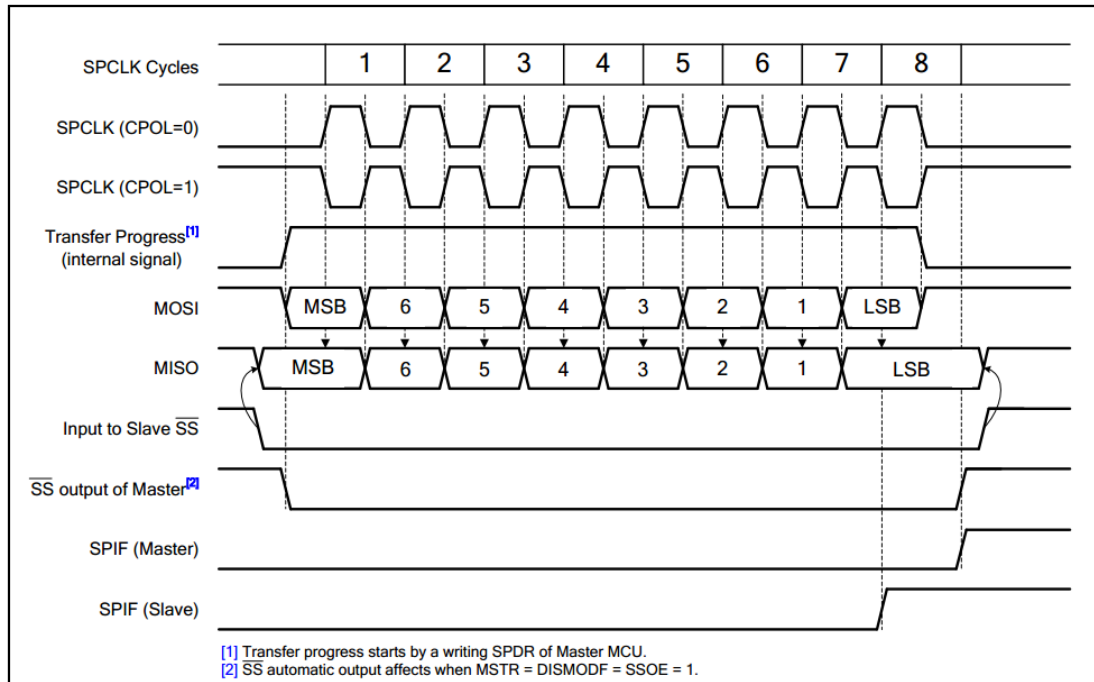
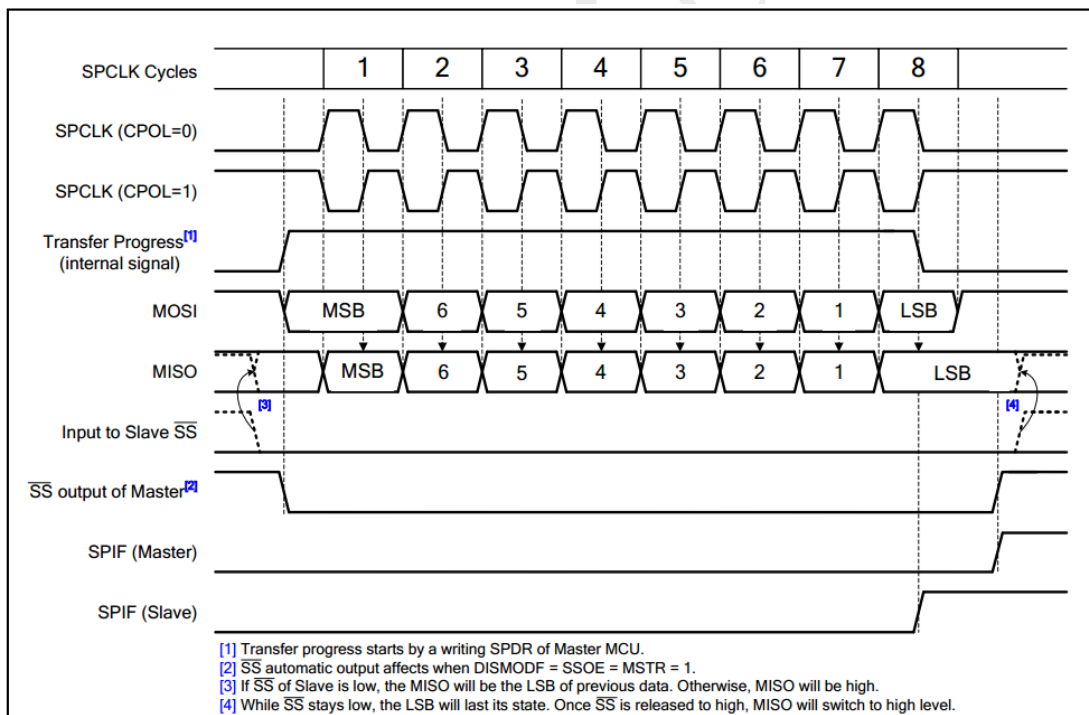

**图 18-2-4 SPI 时钟格式**

在 SPI 传输中，总是由主机启动传输。如果 SPI 被选定作为主模机式 ( $MSTR = 1$ ) 并且打开传输 ( $SPIEN = 1$ )，对主机的 SPI 数据寄存器 (SPDR) 写入数据将启动 SPI 时钟和数据传输。传出一个字节的会同时接收一个字节的的内容，此后 SPI 时钟停止，主机和从机的 SPIF (SPSR.7) 同时被置 1。如果 SPI 中断使能位 ESPI (EIE.0) 设置为 1，全局中断使能 ( $EA = 1$ )，将执行 SPI 的 (ISR) 中断服务程序。

关于从机模式下，NSS 信号需要注意。如图 18-2-5 所示， $CPHA=0$  时，SPCLK 第一个边沿为 MSB 的采样点。因此，从机必须在 SPCLK 第一个采样边沿出现之前先把 MSB 传出。NSS 的下降沿可用于准备 MISO 的 MSB。因此，每次成功串行传输一个字节后，NSS 引脚必须切换先高然后低。此外，如果从机将数据写入 SPI 数据寄存器 (SPDR) 时，如果 NSS 为低电位，则会发生写冲突错误。

当  $CPHA = 1$ ，采样边沿位于 SPCLK 时钟的第二个边沿。从机使用的第一个 SPCLK 时钟转移的 MSB，而不是 NSS 的下降沿。因此，在每次成功传输时 NSS 可以始终保持低电位保持低之间的转移。此格式更适合单主机单从机的结构使用。 $CPHA=1$  模式，从机的 NSS 可以不连接在 SPI 系统中，直接接地。

在 SPI 传输使能 ( $SPIEN=1$ ) 前，必须先对 SPI 传输进行配置，否则传输过程中对  $MSTR$ ， $CPOL$ ， $CPHA$  及  $SPR[1:0]$  的任一更改，将会停止 SPI 传输并强迫总线进入空闲模式。所以在任何配置位更改前，请先关闭  $SPIEN$  使能位。


**图 18-2-5 SPI 时钟和数据格式 (CPHA = 0)**

**图 18-2-6 SPI 时钟和数据格式 (CPHA = 1)**

## 18.2.4 从机选择引脚 NSS 配置

SPI 每个从属外围设备由一个从选择引脚 (NSS 引脚) 选择, 当引脚信号为低电平时,

表明该从设备被选中。主设备可以通过软件控制连接于从设备 NSS 引脚的端口电平选择每个从设备，很明显，只有一个主设备可以驱动通讯网络。为了防止 MISO 总线冲突，同一时间只允许一个从设备与主设备通讯。在主设备模式中，NSS 引脚状态关联 SPI 状态寄存器 SPSR 的 MODF 标志位以防止多个主设备驱动 MOSI 和 SCK。

下列情况，NSS 引脚可以作为普通 IO 或其它功能使用：

设备作为主设备，寄存器 SPCR 寄存器的 SSDIS 位置 1。这种软件配置仅仅存在于通讯网络中只有一个主设备的情况，因此，SPI 状态寄存器 SPSR 的 MODF 标志位不会被置 1。

设备软件配置为从设备，SPI 控制寄存器 SPCR 的 CPHA 位和 SSDIS 位置 1。这种软件配置情况存在于只有一个主设备一个从设备的通讯网络中，因此，设备总是被选中的，主设备也不需要控制从设备的 NSS 引脚选择其作为通讯目标。

### 18.2.5 写冲突错误

SPI 在发送方向上是单缓存，发送完成才允许写入下一个数据。即前一个数据传输完，新的数据才能写入移位寄存器。当正在进行一次传送时，如果设备同时又写数据到 SPDR，将发生写冲突错误。发送数据时由于 SPDR 不是一个双缓存，任何写入 SPDR 数据将直接写入 SPI 的移位寄存器。一旦发生一个写冲突错误，WCOL(SPSR.6)会被硬件置 1 指示发生一个写冲突。这种情况下，当前的传输继续不停，然而引起写冲突的新数据将丢失。尽管 SPI 逻辑可以在主机和从机之间进行写冲突检测，但写冲突通常会是一个从机错误，原因是当主机开始一次传送时，从机是无法预知。在从机接收过程中，写 SPDR 也将产生写冲突错误。WCOL 标志用软件清除。

### 18.2.6 SPI 中断

SPI 中断状态标志包括 SPIF 和 MODF，用于产生 SPI 事件中断请求。这些位都放在 SPSR 寄存器中。当有外部数据传入 SPDR 或自身完成数据传输后，SPIF 标志将被置位。MODF 置 1 时，表示 NSS 进入模式错误状态。当 SPI 中断打开时（ESPI(EIE.6)和 EA 置 1），当这 2 个标志中的任意一个置 1，CPU 会执行 SPI 中断服务程序。用户若需要了解是由何种标志引起中断，必须检查相应的标志位。这 2 个标志必须由用户软件清除。

## 18.3 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>SPCR</b>	0xE8	R/W	SPI 控制寄存器	00000000
<b>SPDR</b>	0xE9	R/W	SPI 数据寄存器	00000000
<b>SPSR</b>	0xEA	R/W	SPI 状态寄存器	00000000

## 18.4 寄存器描述

### 18.4.1 SPI 控制寄存器 (SPCR)

寄存器	地址	R/W	描述	复位后的值
<b>SPCR</b>	E8H	R/W	SPI 控制寄存器	0011 0100

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>SPR2</b>	<b>SPIEN</b>	<b>SSDIS</b>	<b>MSTR</b>	<b>CPOL</b>	<b>CPHA</b>	<b>SPR1</b>	<b>SPR0</b>

位	描述
[7]	<b>SPR2</b> SPI 时钟速率选择位，配合 <b>SPR1</b> 和 <b>SPR0</b> 使用
[6]	<b>SPIEN</b> SPI 使能位 0: 禁止 SPI 1: 使能 SPI
[5]	保留
[4]	<b>MSTR</b> SPI 模式选择位 0: 从模式 1: 主模式
[3]	<b>CPOL</b> 时钟极性控制位 0: 在空闲状态下 SCK 处于低电平 1: 在空闲状态下 SCK 处于高电平
[2]	<b>CPHA</b> SPI 时钟相位控制 0: 在 SCK 奇数边沿采样数据 1: 在 SCK 偶数边沿采样数据
[7 1:0]	<b>SPR[2:0]</b> SPI 时钟 (SPCLK) 速率选择位 000: FSYS/2 001: FSYS/4 010: FSYS/8 011: FSYS/16 100: FSYS/32 101: FSYS/64 110: FSYS/128 111: 无时钟产生

### 18.4.2 SPI 数据寄存器 (SPDR)

寄存器	地址	R/W	描述	复位后的值
<b>SPDR</b>	0xE9	R/W	SPI 数据寄存器	00000000



BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SPDR[7:0]							

位	描述	
[7:0]	SPDR[7:0]	<b>串行外设数据寄存器</b> 该字节为 SPI 总线上传输或接收的数据。对该字节执行写入，是对移位寄存器写操作。读取这个字节，实际上是从一个缓冲区读取数据。在主机模式，读该寄存器，初始化同时传输和接收一个字节。

### 18.4.3 SPI 状态寄存器 (SPSR)

寄存器	地址	R/W	描述	复位后的值
SPSR	0xEA	R/W	SPI 状态寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SPIF	WCOL	SSERR	MODEF	-	-	-	-

位	描述	
[7]	SPIF	<b>SPI 数据传输标志位</b> 0: 传输过程中由硬件清除。或者当 SPIF 置位后读取 SPSR 后再读 SPDR 清除 1: 已完成数据传输，由硬件置 1
[6]	WCOL	<b>写入冲突标志位</b> 0: 已处理写入冲突，由软件清 0 1: 检测到一个冲突，由硬件置 1
[5]	SSERR	同步串行从属错误标志，当 NSS 输入低电平在接收序列结束前被释放，将由硬件置 1。通过将寄存器 SPCR 的 SPIEN 位写 0 来清除标志
[4]	MODEF	<b>模式错误标志位</b> 0: 未发生模式错误，由软件清 0 1: 表明 NSS 引脚电平与 SPI 模式不一致，由硬件置 1
[3:0]	保留	保留

## 19 IIC 总线

### 19.1 概述

RM1221 的 I2C 总线提供了一种串行通信方式，用在 MCU 与外设 EEPROM, LCD 模块，温度传感器等等之间控制。I2C 用两条线（数据线 SDA 和时钟线 SCL）在设备间传输数据。

I2C 总线用作主机与从机之间双向数据传输。可以用于多主机系统，支持无中央主机及多主机系统，主机与主机之间的总线仲裁传输，同步时钟 SCL 的存在，允许设备间使用不同比特率的数据传输。支持四种传输模式：主发，主收，从发，从收。I2C 总线仅支持 7 位地址。支持广播呼叫，支持标准速率传输（100kbps）和快速传输（400kbps）。

### 19.2 功能描述

I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。可以开启和禁止中断。接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。

I2C 接口可以下述 4 种模式中的一种运行：

- (1) 主发送器模式
- (2) 主接收器模式
- (3) 从发送器模式
- (4) 从接收器模式

该模块默认工作于从模式。接口在生成起始条件后自动地由从模式切换到主模式；当仲裁丢失或产生停止信号时，则由主模式切换到从模式。

允许多主机功能，即如果两个或更多主机同时初始化数据传输，可以通过冲突检测与仲裁来防止数据被破坏。任何能够进行发送和接收的设备都可以成为主机，但在任何时间点上只能有一个主机。

主模式时，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（I2CON 寄存器中的 STA 和 STO 位）。从模式时，I2C 接口能识别它自己的 7 位地址和广播呼叫地址。

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 个字节是地址。地址只在主模式下发送。在 1 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 I2CON 寄存器中的 AA 位开启或禁止应答(ACK)，并可以在 I2ADDR 寄存器中设置 I2C 接口的地址。

根据 CPU 时钟和 I2C 时钟软件配置 I2CLK[7:0]寄存器，寄存器最小软件配置大于等于 4。

I2C 时钟计算公式为：
$$I2CCLK = \frac{CPUCLK}{(I2CLK[7:0]+1) \times 4}$$

### 19.2.1 主发送器模式

在主发送器模式下，当 I2C 的状态（I2STA 寄存器）发生转变时，中断请求位（I2CON 寄存器中的 I2CIF 位）将被置 1，若 EI2C（EIE.0）和 EA 被使能，I2C 将发出中断请求。表 17-2-1 为主发送器模式下的 I2C 状态列表。

表 19-2-1 主发送器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2DAT	写 I2CON				
			STA	STO	I2CIF	AA	
08H	1 个起始位已发送	写 SLA+W	X	0	0	X	SLA+W 将被发送，并收到 ACK
10H	1 个重新起始位已发送	写 SLA+W	X	0	0	X	SLA+W 将被发送，并收到 ACK
		写 SLA+R	X	0	0	X	SLA+R 将被发送，I2C 被切换到主接收器模式
18H	“SLA+W”已发送，且 ACK 已收到	写数据字节	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
20H	“SLA+W”已发送，且 ACK 未收到	无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
		写数据字节*	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
28H	I2DAT 中的数据字节已发送，且 ACK 已收到	无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
		写数据字节	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
30H	I2DAT 中的数据字节已发送，且 ACK 未收到	无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		写数据字节*	0	0	0	X	数据字节将被发送，并收到 ACK
		无操作	1	0	0	X	重新起始位将被发送

[1] “SLA”表示 7 位的通信目标从机地址，“R”表示将从目标从机读数据（R/W=1），此位将与“SLA”一起发送，“W”表示将向目标从机写数据（R/W=0），此位也将与“SLA”一起发送。

[2] \*仅适用于 NACK 并不表示通信结束的那些应用。

## 19.2.2 主接收器模式

在主接收器模式下，当 I2C 的状态（I2STA 寄存器）发生转变时，中断请求位（I2CON 寄存器中的 I2CIF 位）将被置 1，若 EI2C（EIE.0）和 EA 位被使能，I2C 将发出中断请求。  
 表 19-2-2 主接收器模式下的 I2C 状态列表。

表 19-2-2 主接收器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作				I2C 硬件的下一步动作	
		写/读 I2DAT	写 I2CON				
			STA	STO	I2CIF		AA
08H	1 个起始位已发送	写 SLA+R	X	0	0	X	SLA+R 将被发送，并收到 ACK
10H	1 个重新起始位已发送	写 SLA+R	X	0	0	X	SLA+R 将被发送，并收到 ACK
		写 SLA+W	X	0	0	X	SLA+W 将被发送，I2C 被切换到主发送器模式
40H	“SLA+R”已发送，且 ACK 已收到	无操作	0	0	0	0	数据字节将被接收，并未收到 ACK
		无操作	0	0	0	1	数据字节将被接收，并收到 ACK
48H	“SLA+R”已发送，且 ACK 未收到	无操作	1	0	0	X	重新起始位将被发送
		无操作	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		无操作	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0
50H	接收到的数据字节已存入 I2DAT 中，且 ACK 已收到	读数据字节	0	0	0	0	数据字节将被接收，并未收到 ACK
		读数据字节	0	0	0	1	数据字节将被接收，并收到 ACK
58H	接收到的数据字节已存入 I2DAT 中，且 ACK 未收到	读数据字节	1	0	0	X	重新起始位将被发送
		读数据字节	0	1	0	X	结束位将被发送；STO 标志位将被清 0
		读数据字节	1	1	0	X	起始位与结束位将被发送；STO 标志位将被清 0

### 19.2.3 从发送器模式

在从发送器模式下，当 I2C 的状态（I2STA 寄存器）发生转变时，中断请求位（I2CON 寄存器中的 I2CIF 位）将被置 1，若 EI2C（EIE.0）和 EA 位被使能，I2C 将发出中断请求。表 19-2-3 为从发送器模式下的 I2C 状态列表。

表 19-2-3 从发送器模式下的 I2C 状态列表

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2DAT	写 I2CON				
			STA	STO	I2CIF	AA	
A8H	自己作为从机的地址“SLA+R”已收到，且 ACK 已发送	写数据字节	X	0	0	0	最后一个数据字节将被发送，并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送，并收到 ACK
B0H	多主机下通过仲裁，主机控制权丢失；自己作为从机的地址“SLA+R”已收到，且 ACK 已发送	写数据字节	X	0	0	0	最后一个数据字节将被发送，并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送，并收到 ACK
B8H	数据字节已发送，且 ACK 已收到	写数据字节	X	0	0	0	最后一个数据字节将被发送，并收到 ACK
		写数据字节	X	0	0	1	一个数据字节将被发送，并收到 ACK
C0H	数据字节已发送，且 ACK 未收到	无操作	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位
C8H	最后一个数据字节已发送，且 ACK 已收到	无操作	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位

## 19.2.4 从接收器模式

在从接收器模式下，当 I2C 的状态（I2STA 寄存器）发生转变时，中断请求位（I2CON 寄存器中的 I2CIF 位）将被置 1，若 EI2C（EIE.0）和 EA 位被使能，I2C 将发出中断请求。表 19-2-4 为从接收器模式下的 I2C 状态列表。

**表 19-2-4 从接收器模式下的 I2C 状态列表**

状态码	I2C 状态	应用软件操作					I2C 硬件的下一步动作
		写/读 I2DAT	写 I2CON				
			STA	STO	I2CIF	AA	
60H	自己作为从机的地址“SLA+W”已收到，且ACK已发送	无操作	X	0	0	0	数据字节将被接收，且无ACK返回
		无操作	X	0	0	1	数据字节将被接收，且有ACK返回
68H	多主机下通过仲裁，主机控制权丢失；自己作为从机的地址“SLA+W”已收到，且ACK已发送	无操作	X	0	0	0	数据字节将被接收，且无ACK返回
		无操作	X	0	0	1	数据字节将被接收，且有ACK返回
70H	通用广播呼叫地址(00H)已收到，且ACK已发送	无操作	X	0	0	0	数据字节将被接收，且无ACK返回
		无操作	X	0	0	1	数据字节将被接收，且有ACK返回
78H	多主机下通过仲裁，主机控制权丢失；通用广播呼叫地址(00H)已收到，且ACK已发送	无操作	X	0	0	0	数据字节将被接收，且无ACK返回
		无操作	X	0	0	1	数据字节将被接收，且有ACK返回
80H	此前已被寻址为从机；数据字节已接收，且ACK已发送	读数据字节	X	0	0	0	数据字节将被接收，且无ACK返回
		读数据字节	X	0	0	1	数据字节将被接收，且有ACK返回
88H	此前已被寻址为从机；数据字节已接收，且ACK未发送	读数据字节	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		读数据字节	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		读数据字节	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		读数据字节	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位
90H	此前已被通用广播呼叫寻址；数据字节已接收，且ACK已发送	读数据字节	X	0	0	0	数据字节将被接收，且无ACK返回
		读数据字节	X	0	0	1	数据字节将被接收，且有ACK返回
98H	此前已被通用广播呼叫寻址；数据字节已接收，且ACK未发送	读数据字节	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		读数据字节	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		读数据字节	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		读数据字节	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位
A0H	当被寻址为从接收器和从发送器模式时，停止位或重新起始位被接收到	无操作	0	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应
		无操作	0	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应
		无操作	1	0	0	0	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫不响应；当总线空闲时发送起始位
		无操作	1	0	0	1	切换到不可寻址的从机模式；对自己的从机地址和通用广播呼叫响应；当总线空闲时发送起始位

## 19.3 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>I2CCON</b>	0xC8	R/W	IIC 控制寄存器	00000000
<b>I2CADDR</b>	0xEB	R/W	IIC 地址寄存器	00000000
<b>I2CDAT</b>	0xEC	R/W	IIC 数据寄存器	00000000
<b>I2CCLK</b>	0xED	R/W	IIC 时钟寄存器	00000000
<b>I2CSTAT</b>	0xEE	R/W	IIC 状态寄存器	00000000

## 19.4 寄存器描述

### 19.4.1 IIC 控制寄存器 (I2CCON)

寄存器	地址	R/W	描述	复位后的值
<b>I2CCON</b>	0xC8	R/W	IIC 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	<b>I2CEN</b>	<b>STA</b>	<b>STO</b>	<b>I2CIF</b>	<b>AA</b>	-	-

位	描述	
[7]	保留	保留
[6]	<b>I2CEN</b>	<b>I2C 使能位</b> 0: 禁止 I2C 1: 使能 I2C
[5]	<b>STA</b>	<b>IIC 起始标志位</b> 当 STA = 1,时, 如果总线是空闲的, 起始信号会发出
[4]	<b>STO</b>	<b>IIC 起始标志位</b> 当 STO = 1,且处于主模式, STOP 信号会被发出
[3]	<b>I2CIF</b>	<b>I2C 中断标志位, 需软件清 0</b>
[2]	<b>AA</b>	<b>I2C 应答标志位</b> 当 AA=1, 满足下面条件下将会返回应答 从模式下, 收到自己的从地址 从模式下, 当 GC 位被设置并且接收到自己的广播地址时 主模式下, 接收到数据 从模式接收到数据 当 AA=0, 任何情况下都不会产生应答信号
[1:0]	保留	保留



### 19.4.2 IIC 地址寄存器 (I2CADDR)

寄存器	地址	R/W	描述	复位后的值
I2CADDR	0xEB	R/W	IIC 地址寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
I2ADDR[7:0]							

位	描述	
[7:0]	I2ADDR[7:0]	I2C 地址寄存器

### 19.4.3 IIC 数据寄存器 (I2CDAT)

寄存器	地址	R/W	描述	复位后的值
I2CDAT	0xEC	R/W	IIC 数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
I2DAT[7:0]							

位	描述	
[7:0]	I2DAT[7:0]	I2C 数据发送接收寄存器

### 19.4.4 IIC 时钟寄存器 (I2CCLK)

寄存器	地址	R/W	描述	复位后的值
I2CCLK	0xED	R/W	IIC 时钟寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
I2CLK[7:0]							

位	描述	
[7:0]	I2CLK[7:0]	I2C 时钟寄存器

### 19.4.5 IIC 状态寄存器 (I2CSTAT)

寄存器	地址	R/W	描述	复位后的值
I2CSTAT	0xEE	R/W	IIC 状态寄存器	11111000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>I2CSTAT[4:0]</b>					-	-	-

位	描述	
[7:3]	<b>I2CSTAT[4:0]</b>	<b>I2C 状态寄存器</b> I2C 状态列表，详情请查阅主发送器模式、主接收器模式、从发送器模式、从接收器模式下的 I2C 状态列表
[2:0]	保留	保留

## 20 模数转换 ADC

### 20.1 概述

RM1221A 内嵌 12 位逐次逼近寄存器型(SAR)的模拟数字转换器(ADC)。模数转换模块负责将管脚上的模拟信号转换为 12 位二进制数据。RM1221A 支持 12 信道单端输入模式。内部带隙电压(band-gap voltage)为 4/3/2/1V，同时也可用作内部 ADC 输入端。所有模拟电路复用同一组采样电路和同一组采样保持电容。该组采样保持电容为转换电路的输入端。然后转换器通过逐次逼近的方式得到有效结果并存放在 ADC 结果寄存器中。

## 20.2 功能描述

### 20.2.1 ADC 工作方式

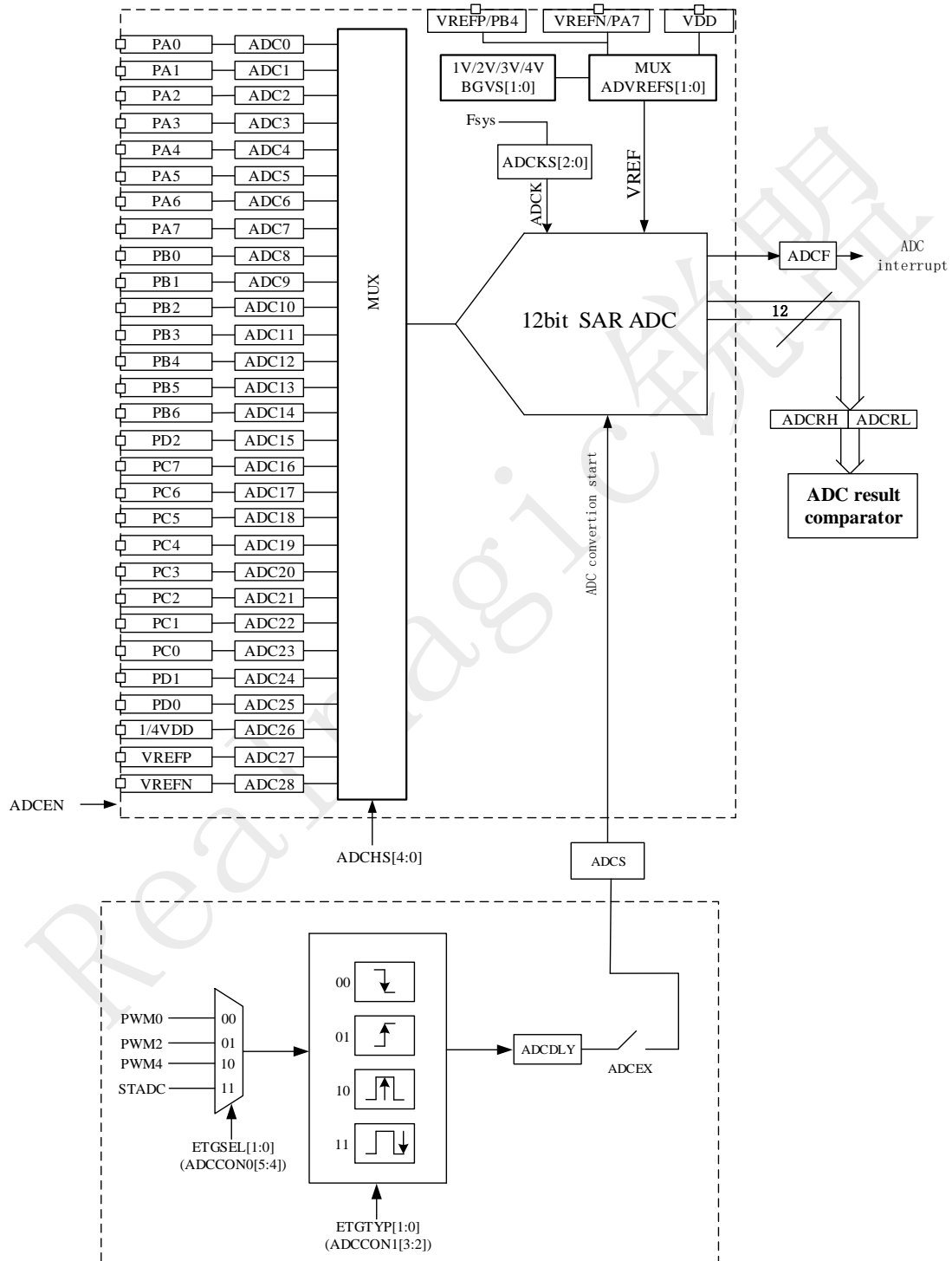


图 20-2-1 12-位 ADC 模块结构图

在开始 ADC 转换前,通过设置 ADCEN(ADCCON1.0)位使能 ADC 电路,从而激活 ADC 电路,由于 ADC 模块需要额外功耗,一旦 ADC 转换模块不再使用,建议清零 ADCEN 位关闭 ADC 模块电路以节省功耗。

ADC 转换输入管脚需要特别配置,通过 ADCHS[3:0]来选择采样所需要的 ADC 通道连接到采样电路上。同时,用户需要通过配置 AINCHS 寄存器相应位来将 ADC 输入脚变成纯模拟输入电路。同样 ADC 采样时钟也需要认真考虑。ADC 最高时钟频率参考表表 31-9。当采样时钟设置超过最大值时,采样结果数据为不可预测。

通过置 ADCS 位(ADCCON0.6)开启 AD 转换。当转换完成后,硬件会自动清除该位,同时置 ADCF(ADCCON0.7)位,如果之前 ADC 中断已使能,则会产生 ADC 中断。转换结果存放在 ADCRH (高 8 位)及 ADCRL (低 4 位)中。12 位转换结果值为  $4095 \times \frac{V_{AIN}}{V_{REF}}$

内部及外部数字电路,可能影响采样结果的准确度。所以如果需要高精度的转换结果,请参考如下应用,以降低噪声电平干扰。

- ◆ 模拟输入脚尽量离芯片越近越好,避免管脚附近有高速数字电路经过,并离高速数字电路越远越好。
- ◆ 在转换过程中,将芯片进入空闲模式。
- ◆ 如果模拟输入脚 AIN 在系统中同时需要切换做数字管脚,请确保在转换过程中不要做数字/模拟切换动作。

## 20.2.2 外部触发 ADC

除了通过软件启动 AD 转换外, RM1221A 提供硬件触发方式启动 AD 转换。一旦 ADCEX (ADCCON1.1)置 1,选择 PWM 通道的边沿或周期, STADC 管脚边沿自动触发启动 AD 转换(由硬件设置自动 ADCS 信号)。通过 STADCPX(CONFIG0.6)可以灵活配置 STADC 的输入管脚,通过 ETGSEL (ADCCON0[5:4])和 ETGTYP (ADCCON1[3:2])设置来选择触发源和触发类型。同时,还可以在外部触发信号与启动 AD 转换之间插入触发延时(触发延时计数器)。RM1221A 该功能将非常适用于高精度电机控制。注意,在 AD 模块转换过程中(ADCS = 1),任何软件或硬件触发信号都是无效的。

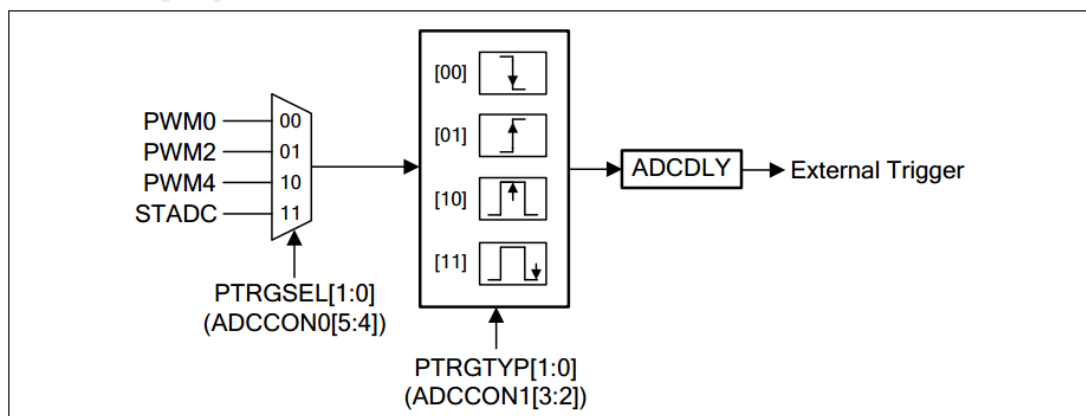


图 20-2-2 外部触发 ADC 电路结构

### 20.2.3 ADC 转换结果比较器

RM1221A ADC 提供一组数字比较器，用于比较 AD 12 位转换结果与预先填入寄存器 ACMPH 及 ACMPL 的内容是否一致。ADC 比较器使能位为 ADCMPEN(ADCCON2.5)一旦设定，每次 AD 转换结束都会进行比较。ADCMP0(ADCCON2.4)显示根据 ADCMPOP (ADCCON2.6)设定的比较结果。当 ADFBEN(ADCCON2.7)设置后，ADC 比较结果可触发 PWM 故障刹车。

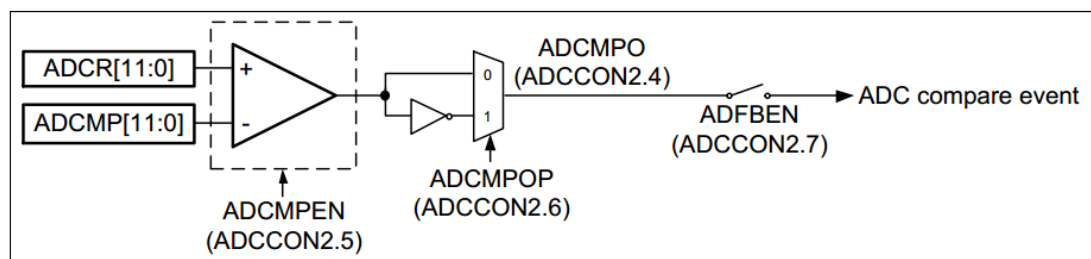


图 20-2-3 ADC 结果比较器

### 20.2.4 ADC 参考带隙电压应用

RM1221A 内嵌带隙电压(band-gap voltage)为 4/3/2/1V 可选。可选择内部 VDD、内部带隙电压、外部 VREFP 从 PB4 引脚输入作为 ADC 的正参考电压。可选择内部 VSS、外部 VREFN 从 PA.7 引脚输入作为 ADC 的负参考电压。

## 20.3 寄存器列表

寄存器	地址	R/W	描述	复位值
ADCCON0	0xA9	R/W	ADC 控制寄存器 0	00000000
ADCCON1	0xAA	R/W	ADC 控制寄存器 1	00000000
ADCCON2	0xAB	R/W	ADC 控制寄存器 2	00000000
ADCCON3	0xF020	R/W	ADC 控制寄存器 3	00000000
ADCCON4	0xAC	R/W	ADC 控制寄存器 4	00000000
ADCCON5	0xF021	R/W	ADC 控制寄存器 5	00000000
ADCCON6	0xF022	R/W	ADC 控制寄存器 6	00000000
PACHS	0xF024	R/W	ADC 模拟输入通道功能寄存器	00000000
PBCHS	0xF025	R/W	ADC 模拟输入通道功能寄存器	00000000
PCCHS	0xF026	R/W	ADC 模拟输入通道功能寄存器	00000000
PDCHS	0xF027	R/W	ADC 模拟输入通道功能寄存器	00000000
ADCDLY	0xF028	R/W	ADC 外部触发延迟计数器	00000000
ADCRL	0xAD	R/W	ADC 转换结果低位寄存器	00000000
ADCRH	0xAE	R/W	ADC 转换结果高位寄存器	00000000

<b>ADCMPL</b>	0xF029	R/W	ADC 比较值低位寄存器	00000000
<b>ADCMPLH</b>	0xF02A	R/W	ADC 比较值高位寄存器	00000000

## 20.4 寄存器描述

### 20.4.1 ADC 控制寄存器 0 (ADCCON0)

寄存器	地址	R/W	描述	复位后的值
<b>ADCCON0</b>	0xA9	R/W	ADC 控制寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>ADCF</b>	<b>ADCS</b>	<b>ETGSEL[1:0]</b>		保留			

位	描述	
[7]	<b>ADCF</b>	<b>ADC 标志</b> 当 AD 转换完成, 该位置 1。可读取到当前 AD 转换结果。该位为 1 时无法开始新一轮转换, 需要软件清零。
[6]	<b>ADCS</b>	<b>A/D 转换软件启动位</b> 该位置 1 启动 AD 转换。在 AD 转换过程中该位保持为 1, 当转换结束硬件自动清 0。这意味着写入 ADCS 的值和读出的不一定相符 写: 0: 无动作 1: 开始 AD 转换 读: 0: ADC 模块空闲状态 1: ADC 模块工作中
[5:4]	<b>ETGSEL[1:0]</b>	<b>外部触发源选择</b> 当 ADCEX (ADCCON1.1) 为 1, 该位选择外部触发 ADC 的来源 00: PWM0 01: PWM2 10: PWM4 11: STADC 脚
[3:0]	保留	保留

### 20.4.2 ADC 控制寄存器 1 (ADCCON1)

寄存器	地址	R/W	描述	复位后的值
<b>ADCCON1</b>	0xAA	R/W	ADC 控制寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

<b>CHOP</b>	<b>ADVREFS[2:0]</b>	<b>ETGTYP[1:0]</b>	<b>ADCEX</b>	<b>ADCEN</b>
-------------	---------------------	--------------------	--------------	--------------

位	描述	
[7]	<b>CHOP</b>	<p><b>OFFSET 交换</b></p> <p>0: 两端信号不交换 (正端为信号, 负端为参考电压)</p> <p>1: 器两端信号交换</p> <p>在 AD 转换过程中通过不断变换 ADCCON1 寄存器中的 CHOP 的值。如第一次 AD 转换 CHOP 置 0, 第二次 AD 转换 CHOP 置 1, 然后将第一次和第二次测试的 AD 值求平均值。两次转换得到的平均值就是去掉失调电压的正确结果。</p>
[6:4]	<b>ADVREFS[2:0]</b>	<p><b>ADC 参考电压 VREF 选择</b></p> <p>000: 选择 VDD 为参考电压正端, VSS 为参考电压负端(默认)</p> <p>001: 选择 VDD 为参考电压正端, VREFN (PA.7) 为参考电压负端</p> <p>010: 选择内部带隙电压为参考电压正端, VSS 为参考电压负端</p> <p>011: 选择内部带隙电压为参考电压正端, VREFN 为参考电压负端</p> <p>100: 选择内部带隙电压为参考电压正端可输出接滤波电容 PB.4, VSS 为参考电压负端</p> <p>101: 选择内部带隙电压为参考电压正端可输出接滤波电容 PB.4, VREFN 为参考电压负端</p> <p>110: 选择外部 VREFP 端口 PB.4 输入作为参考电压正端, VSS 为参考电压负端</p> <p>111: 选择外部 VREFP 端口 PB.4 输入作为参考电压正端, VREFN (PA.7) 为参考电压负端</p>
[3:2]	<b>ETGTYP[1:0]</b>	<p><b>外部触发信号类型选择</b></p> <p>当 ADCEX (ADCCON1.1)置 1, 该位决定响应外部触发的类型。</p> <p>00: PWM0/2/4 或 STADC 脚的下降沿</p> <p>01: PWM0/2/4 或 STADC 脚的上升沿</p> <p>10: 一个 PWM 周期的中点</p> <p>11: 一个 PWM 周期的终点</p> <p>注 PWM 周期中点或终点触发仅适用于中心对齐模式的 PWM 输出。</p>
[1]	<b>ADCEX</b>	<p><b>ADC 触发启动信号选择位</b></p> <p>该位决定启动 ADC 的触发条件</p> <p>0 = 当软件设定 ADCS 位, 启动 AD 转换</p> <p>1 = 当软件设定 ADCS 位后, 还需要外部触发信号触发才启动。外部触发信号条件由寄存器 ETGSEL[1:0]及 ETGTYP[1:0]决定。</p> <p>注: 当 ADCS 为 1 时 (正在转换), 外部触发信号不会影响 ADC 直到 ADC 转换结束, ADCS 被硬件清 0。</p>
[0]	<b>ADCEN</b>	<p><b>ADC 使能位</b></p> <p>0: ADC 转换电路关闭</p> <p>1: ADC 转换电路打开</p>



### 20.4.3 ADC 控制寄存器 2 (ADCCON2)

寄存器	地址	R/W	描述	复位后的值
ADCCON2	0xAB	R/W	ADC 控制寄存器 2	00000010

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	保留	ADCKS[2:0]		

位	描述	
[7]	ADFBEN	<b>ADC 比较结果响应故障刹车使能</b> 0: 禁止 1: ADC 触发故障刹车功能使能。一旦比较结果 ADCMPO 为 1, 触发故障刹车模块。即符合 PWM 故障刹车输出值后, 硬件将清除 PWMRUN (PWMCON0.7), 并终止 PWM 输出。当 PWMRUN 置 1, PWM 重新输出。
[6]	ADCMPOP	<b>ADC 比较输出极性选择位</b> 0: 若 ADCR[11:0]大于或等于 ADCMP[11:0], ADCMPO 为 1 1: 若 ADCR[11:0]小于 ADCMP[11:0], ADCMPO 为 1
[5]	ADCMPEN	<b>ADC 结果比较使能位</b> 0: ADC 结果比较功能关闭 1: ADC 结果比较功能打开
[4]	ADCMPO	<b>ADC 比较结果输出位</b> 该位输出 ACMPOP 设定比较输出的结果。每次 AD 转换结束都会更新输出。
[3]	保留	保留
[2:0]	ADCKS[2:0]	<b>ADC 转换时钟选择位</b> 000: Fcpu/2 001: Fcpu/4 010: Fcpu/8 (默认) 011: Fcpu/16 100: Fcpu/32 101: Fcpu/64 110: Fcpu/128 111: Fcpu/256

### 20.4.4 ADC 控制寄存器 3 (ADCCON3)

寄存器	地址	R/W	描述	复位后的值
ADCCON3	0xF020	R/W	ADC 控制寄存器 3	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ADTS[1:0]		SWVCMS	ADT[4:0]				

位	描述	
[7:6]	<b>ADTS[1:0]</b>	<b>ADC 引入 OFFSET</b> 00: 不引入 OFFSET 01: ADC 引入 8 个码的正 OFFSET 10: ADC 引入 8 个码的负 OFFSET 11: 不引入 OFFSET
[5]	<b>SWVCMS</b>	
[4:0]	<b>ADT[4:0]</b>	<b>ADC 的 OFFSET 校准控制位 (默认为 10000)</b> 00000 为负的最大 OFFSET 11111 为正的最大 OFFSET

### 20.4.5 ADC 控制寄存器 4 (ADCCON4)

寄存器	地址	R/W	描述	复位后的值
<b>ADCCON4</b>	0xAC	R/W	ADC 控制寄存器 4	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>DFILEN</b>	<b>DFILNUM[1:0]</b>		<b>ADCHS[4:0]</b>				

位	描述																																					
[7]	<b>DFILEN</b>	<b>SAR ADC 数字滤波器使能信号</b> 0: 关闭 1: 开启																																				
[6:5]	<b>DFILNUM[1:0]</b>	<b>SAR ADC 数字滤波器滤波次数</b> 00: 2 次, 01: 4 次 10: 8 次 11: 16 次 此时 ADR 的结果为 2 次或 4 次 ADC 转换的平均值, 开启数字滤波后 ADC 的转换速率会下降到原来的 1/2 或 1/4																																				
[4:0]	<b>ADCHS[4:0]</b>	<b>A/D 转换通道选择</b> 该位用于选择 ADC 转换通道。当 ADCEN 为 0 所有输入无效 <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>ADCHS[4:0]</th> <th>ADC 转换通道</th> <th>ADCHS[4:0]</th> <th>ADC 转换通道</th> </tr> </thead> <tbody> <tr> <td>00000</td> <td>AIN0</td> <td>10000</td> <td>AIN16</td> </tr> <tr> <td>00001</td> <td>AIN1</td> <td>10001</td> <td>AIN17</td> </tr> <tr> <td>00010</td> <td>AIN2</td> <td>10010</td> <td>AIN18</td> </tr> <tr> <td>00011</td> <td>AIN3</td> <td>10011</td> <td>AIN19</td> </tr> <tr> <td>00100</td> <td>AIN4</td> <td>10100</td> <td>AIN20</td> </tr> <tr> <td>00101</td> <td>AIN5</td> <td>10101</td> <td>AIN21</td> </tr> <tr> <td>00110</td> <td>AIN6</td> <td>10110</td> <td>AIN22</td> </tr> <tr> <td>00111</td> <td>AIN7</td> <td>10111</td> <td>AIN23</td> </tr> </tbody> </table>	ADCHS[4:0]	ADC 转换通道	ADCHS[4:0]	ADC 转换通道	00000	AIN0	10000	AIN16	00001	AIN1	10001	AIN17	00010	AIN2	10010	AIN18	00011	AIN3	10011	AIN19	00100	AIN4	10100	AIN20	00101	AIN5	10101	AIN21	00110	AIN6	10110	AIN22	00111	AIN7	10111	AIN23
ADCHS[4:0]	ADC 转换通道	ADCHS[4:0]	ADC 转换通道																																			
00000	AIN0	10000	AIN16																																			
00001	AIN1	10001	AIN17																																			
00010	AIN2	10010	AIN18																																			
00011	AIN3	10011	AIN19																																			
00100	AIN4	10100	AIN20																																			
00101	AIN5	10101	AIN21																																			
00110	AIN6	10110	AIN22																																			
00111	AIN7	10111	AIN23																																			

		01000	AIN8	11000	AIN24
		01001	AIN9	11001	AIN25
		01010	AIN10	11010	1/4 VDD
		01011	AIN11	11011	内部带隙电压 VREFP(1V/2V/3V/4V) 根据 BGV5[1:0]选择
		01100	AIN12	11100	VREFN
		01101	AIN13	11101	1/4 PB0
		01110	AIN14	11110	1/4 PC0
		01111	AIN15	11111	内部参考地 VREFN 或 ADC 的 VSS (VREFNINS 选择)

### 20.4.6 ADC 控制寄存器 5 (ADCCON5)

寄存器	地址	R/W	描述	复位后的值
ADCCON5	0xF021	R/W	ADC 控制寄存器 5	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ADREFNS	ADLDOPD	CALI	VCMS	BUFI[1:0]		ADCMPI[1:0]	

位	描述	
[7]	ADREFNS	内部 VREFN 选择 0: VSSA PAD 接入 1: ADC 本地 VSS
[6]	ADLDOPD	参考 LDO 关闭配置 0: 开启 1: 关掉
[5]	CALI	OFFSET 校正模式 0: 自校正 1: 手动校正
[4]	VCMS	VCM 参考电压选择 0: VCC 分压 1: REFP 分压
[3:2]	BUFI[1:0]	ADC BUFFER 电流控制 00: ×1 01: ×0.5 10: ×2 11: ×3
[1:0]	ADCMPI[1:0]	ADC 比较器电流控制位, 代码选项映射 SFR 00: ×1

		01: $\times 0.5$ 10: $\times 1.5$ 11: $\times 2$
--	--	--

### 20.4.7 ADC 控制寄存器 6 (ADCCON6)

寄存器	地址	R/W	描述	复位后的值
ADCCON6	0xF022	R/W	ADC 控制寄存器 6	10000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
RNADT[3:0]				VREFNINS	TRICKTMP[2:0]		

位	描述	
[7:4]	RNADT[3:0]	ADC OFFSET 校准控制位, 此位校正的 OFFSET 跟参考电压有关, 代码选项映射 SFR
[3]	VREFNINS	ADC 输入地电压选择位, 需配合 ADCHS[4:0] 0: PAD 接入 VREFN 1: VSS
[2:0]	TRTMP[2:0]	AD 参考电压温度特性 TRIMMING 位, 代码选项映射 SFR

### 20.4.8 ADC 转换结果低位寄存器 (ADCRL)

寄存器	地址	R/W	描述	复位后的值
ADCRL	0xAD	R/W	ADC 转换结果低位寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
BGVS[1:0]		TSAMPS[1:0]		ADCR[3:0]			

位	描述	
[7:6]	BGVS[1:0]	ADC 内部带隙基准电压选择 00: 1.024V 01: 2.048V (默认) 10: 3.072V 11: 4.096V
[5:4]	TSAMPS[1:0]	ADC 输入采样时间选择 00: $2 * T_{ADCK}$ 01: $4 * T_{ADCK}$ (默认) 10: $8 * T_{ADCK}$ 11: $16 * T_{ADCK}$
[3:0]	ADCR[3:0]	ADC 转换结果低位

	ADC 转换结果低 4 位
--	---------------

### 20.4.9 ADC 转换结果高位寄存器 (ADCRH)

寄存器	地址	R/W	描述	复位后的值
<b>ADCRH</b>	0xAE	R/W	ADC 转换结果高位寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ADCR[11:4]							

位	描述	
[7:0]	ADCR[11:4]	ADC 转换结果高位 ADC 转换结果高 8 位

### 20.4.10 ADC 比较值低位寄存器 (ADCMPPL)

寄存器	地址	R/W	描述	复位后的值
<b>ADCMPPL</b>	0xF029	R/W	ADC 比较值低位寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留				ADCMP[3:0]			

位	描述	
[7:4]	保留	保留
[3:0]	ADCMP[3:0]	ADC 比较值低位 ADC 比较值低 4 位

### 20.4.11 ADC 比较值高位寄存器 (ADCMPH)

寄存器	地址	R/W	描述	复位后的值
<b>ADCMPH</b>	0xF02A	R/W	ADC 比较值高位寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ADCMP[11:4]							

位	描述	
[7:0]	ADCMP[11:4]	ADC 比较值高位 ADC 比较值高 8 位

### 20.4.12 ADC 比较值高位寄存器 (ADC DLY)

寄存器	地址	R/W	描述	复位后的值
ADC DLY	0xF028	R/W	ADC 比较值高位寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留				ADC DLY[3:0]			

位	描述	
[7:4]	保留	保留
[3:0]	ADC DLY[3:0]	<p><b>ADC 外部触发启动延迟计数器位，9 位计数器</b></p> <p>用于在外部触发启动 ADC 之前加入一段延迟。延迟计数结束在开始 ADC 转换 ADC DLY</p> <p>外部延迟时间 = ADC DLY/FADC</p> <p>注，该延迟仅当 ADCEX (ADCCON1.1) 置 1 时有效。如果启用 PWM 输出触发 ADC 功能，在 PWM 运行过程中不得更改 ADC DLY 计数值。</p>

### 20.4.13 ADC 模拟输入通道 (PA) 功能寄存器 (PACHS)

寄存器	地址	R/W	描述	复位后的值
PACHS	0xF024	R/W	ADC 模拟输入通道 (PA) 功能寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PACHS[7:0]							

位	描述	
[7:0]	PACHS[7:0]	<p><b>ADC 模拟输入通道使能位</b></p> <p>0: PA<sub>x</sub> 为数字通道 (x=0~7)</p> <p>1: PA<sub>x</sub> 为 ADC 模拟输入通道 (x=0~7)</p>

### 20.4.14 ADC 模拟输入通道 (PB) 功能寄存器 (PBCHS)

寄存器	地址	R/W	描述	复位后的值
PBCHS	0xF025	R/W	ADC 模拟输入通道 (PB) 功能寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留	PBCHS[6:0]						

位	描述

[7]	保留	保留
[6:0]	<b>PBCHS[6:0]</b>	<b>ADC 模拟输入通道使能位</b> 0: PBx 为数字通道 (x=0~7) 1: PBx 为 ADC 模拟输入通道 (x=0~7)

### 20.4.15 ADC 模拟输入通道 (PC) 功能寄存器 (PCCHS)

寄存器	地址	R/W	描述	复位后的值
<b>PCCHS</b>	0xF026	R/W	ADC 模拟输入通道 (PC) 功能寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PCCHS[7:0]</b>							

位	描述	
[7:0]	<b>PCCHS[7:0]</b>	<b>ADC 模拟输入通道使能位</b> 0: PCx 为数字通道 (x=0~7) 1: PCx 为 ADC 模拟输入通道 (x=0~7)

### 20.4.16 ADC 模拟输入通道 (PD) 功能寄存器 (PDCHS)

寄存器	地址	R/W	描述	复位后的值
<b>PDCHS</b>	0xF027	R/W	ADC 模拟输入通道 (PD) 功能寄存器 0	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留					<b>PDCHS[6:0]</b>		

位	描述	
[7:3]	保留	保留
[2:0]	<b>PDCHS[2:0]</b>	<b>ADC 模拟输入通道使能位</b> 0: PDx 为数字通道 (x=0~7) 1: PDx 为 ADC 模拟输入通道 (x=0~7)

## 20.5 ADC 应用注意事项

在 ADC 使用过程中需要注意以下事项:

- 1、ADEN 使能后需要等待至少 5uS, 等待 ADC 内部基准建立后方能进行转换;
- 2、使用 ADC 前需要先进行 ADC OFFSET 校正, RM1221A 提供两种校正方式见 ADCCON5.CALI。如果使用手动校正, 需要软件预处理, 参见 Demo 程序。

3、ADC 最快速度可以接近 1MSPS，此时需要改变 ADC BUFFER 和 ADC 比较器电流档位，这时候功耗会增加，性能会稍微下降，最优的性能速度为 250KSPS 及以下。

4、为了小信号性能提升，实际 ADC 示范 Demo 程序可以打开 SAR ADC 数字滤波器功能，开启此功能后 ADC 转换速率下降。

5、对 AD 转换数据进行处理，可以采用以下几种滤波算法：限幅滤波、中位值滤波、算术平均滤波、递推平均滤波、中位值平均滤波、限幅平均滤波、一阶滞后滤波、加权递推平均滤波、消抖滤波、限幅消抖滤波。

ADC 转换步骤如下：

- 1、配置 ADC 时钟和采样建立时间；
- 2、配置引脚功能和选择 ADC 通道；
- 3、配置 ADC 参考电压，内部参考或外部参考；
- 4、使能 ADC，等待至少 5uS 等待 ADC 内部基准稳定；
- 5、启动 ADC 转换，判断 ADCF 是否置 1，根据应用是否进入中断；
- 6、清除 ADCF，完成一次 ADC 转换。

XSFR 中的寄存器 ADCFG 为 Code Option。



## 21 运算放大器 (OPA0)

### 21.1 概述

芯片内置 1 组运算放大器 OPA0。

OPA0 具有以下功能：

- 1、内部集成调零电路；
- 2、正端可以接至 I/O 或者内部参考电压；
- 3、负端可以接至 I/O 或者内部内部电压；
- 4、内置滤波电路。

### 21.2 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>OPACON0</b>	0xF038	R/W	运放控制寄存器 0	0000 0000
<b>OPACON1</b>	0xF039	R/W	运放控制寄存器 1	0000 0000

### 21.3 寄存器描述

#### 21.3.1 运放控制寄存器 0(OPACON0)

寄存器	地址	R/W	描述	复位后的值
OPACON0	0xF038	R/W	运放控制寄存器 0	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>OPA0OUT</b>	<b>OPA0FT</b>	<b>OPA0N</b>	<b>OPA0OEN</b>	<b>OPA0P[2:0]</b>			<b>OPA0EN</b>

位	描述
[7]	<b>OPA0OUT</b> <b>OPA0 输出结果</b> （仅在调节模式下有效），相当于此时为比较器模式 0：运放输出为低，正端电压低于负端电压 1：运放输出为高，正端电压高于负端电压
[6]	<b>OPA0FT</b> <b>运放内部输出滤波选择</b> 0：运放输出内部不接滤波电路 1：运放输出内部接滤波电路
[5]	<b>OPA0N</b> <b>运放负端输入选择</b> 0：运放负端内部接地，PC7 为普通 IO 口

		1: 运放负端接 OPA0-管脚 (PC7)
[4]	<b>OPA0OEN</b>	<b>运放输出选择</b> 0: OPA0 输出不接至 IO PC6 1: OPA0 输出接至 IO 口 (OPA0O 管脚 PC6)
[3:1]	<b>OPA0P[2:0]</b>	<b>运放正端输入选择</b> 000: 运放正端接地, PD0 为普通 IO 口 001: 运放正端接 1/10VDD, PD0 为普通 IO 口 010: 运放正端接 5/10VDD, PD0 为普通 IO 口 011: 运放正端接 7/10VDD, PD0 为普通 IO 口 100: 运放正端接 OPA0+, PD0 为运放输入口 111: 运放正端接 OPA0+, PD0 为运放输入口, 负端跟 opa0n 的配置无关, 且为内部 PGA 放大, 可配置滤波后接入 ADC, PGA 放大的输出根据 OPA0O 的配置 其他: 无效

### 21.3.2 运放控制寄存器 1 (OPACON1)

寄存器	地址	R/W	描述	复位后的值
<b>OPACON1</b>	0xF039	R/W	运放控制寄存器 1	0001 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PGA_GAIN</b>	<b>OPA0CRS</b>	<b>OPA0COFM</b>	<b>OPA0ADJ[4:0]</b>				

位	描述
[7]	<b>PGA_GAIN</b> 内部 PGA 放大倍数选择: 0: x16 1: x64
[6]	<b>OPA0CRS</b> <b>OPA0 调节模式输入端选择位</b> 0: OPA0 调节模式负端输入 1: OPA0 调节模式正端输入
[5]	<b>OPA0COFM</b> <b>OPA0 工作模式选择</b> 0: OPA0 工作在正常模式 1: OPA0 工作在调节模式
[4:0]	<b>OPA0ADJ[4:0]</b> OPA0 失调电压调节位, 默认为 10000,00000 为负失调最大, 11111 为正失调最大

### 21.3.1 运放控制寄存器 3 (LDO)

寄存器	地址	R/W	描述	复位后的值
<b>LDO</b>	0xF03B	R/W	运放控制寄存器 1	0000 0111

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

保留	TR_TMP_ADC[2:0]	SF_EN	LDO_OPT2	LDO_OPT1	LDO_OPT0
----	-----------------	-------	----------	----------	----------

位	描述	
[7]	保留	保留
[6: 4]	TR_TMP[2:0]	AD 参考电压温度特性 TRIMMING 位, 代码选项映射 SFR
[3]	SF_EN	运放小信号输入性能提升使能开关, 代码选项映射 SFR 0: 关闭 1: 打开
[2]	LDO_OPT2	触摸按键模块电流源工作电源选择 0: 工作在 VCC 电压域 1: 工作在 2.3 电压域 (此时 ti_vth 不能被配置为 11, 硬件强制为 10), 默认
[1]	LDO_OPT1	触摸按键模块电流源工作电源选择 0: 工作在 VCC 电压域 1: 工作在 2.3V 电压域
[0]	LDO_OPT0	高速时钟模块工作电源选择 0: 工作在 VCC 电压域 1: 工作在 2.3V 电压域

## 21.4 运算放大器的使用

### 21.4.1 OPA0 使能

将寄存器 OPACON0 的第 0 位置 1, 使能运算放大器。将 OPA0EN 置 0, 禁止运算放大器。

### 21.4.2 OPA0 端口选择

#### 21.4.2.1 OPA0 正端输入

通过设置 OPACON0 寄存器的第 3~1 位, 运放的正端输入可以有以下几种连接方式:

- 1、正端输入接至 I/O 口 (芯片 OPA0+);
- 2、正端内部接到地线;
- 3、正端内部接电阻分压, 可以选择 1/10、5/10、7/10 电压;
- 4、正端作为内部 PGA 放大的正端输入, 负端跟 opa0n 的配置无关, 且为内部 PGA 放大, PGA 放大的输出根据 OPA00 的配置。

#### 21.4.2.2 OPA0 负端输入

通过设置 OPACON0 寄存器的第 5 位, 运放的负端可以有以下几种连接方式:

- 1、运放负端输入接至 I/O 口（芯片的 OPA0-管脚）；
- 2、运放负端内部接地。

### 21.4.2.3 OPA0 输出

运放的输出可以从 OPA0O 引脚输出，这是通过设置 OPACON0 的第 4 位 OPA0OEN 来实现的。

### 21.4.2.4 OPA0 使用时端口方向设置

OPA0 使用相关的 I/O 口时，将会自动将相应 I/O 变为模拟口。

## 21.4.3 OPA0 工作模式

OPA0 正常工作时，用户预判 OPA0 正端、负端信号范围，如果在 0~300mV 之间，可以打开运放小信号性能提升使能开关，它位于 LDO 寄存器的第 3 位 SF\_EN。

运放 OPA0 具有 3 种工作模式：正常模式、内部 PGA 放大模式和调节模式。

寄存器 OPACON1 的第 5 位 OPA0COFM 置 0 且寄存器 OPACON0 的第 1~3 位 OPA0P[2:0] 不等于 111，运放进入正常工作模式。

寄存器 OPACON1 的第 5 位 OPA0COFM 置 0 且寄存器 OPACON0 的第 1~3 位 OPA0P[2:0] 等于 111，运放进入内部 PGA 放大模式。

寄存器 OPACON1 的第 5 位 OPA0COFM 置 1，运放进入调节模式。在调节模式下，运放的正负端内部短路在一起，并连接至运放的正端或者负端（通过 OPACON1 的第 6 位 OPA0CRS 来选择）。调节模式的作用是将运放的失调电压调至最小。

调节模式工作流程：

1. 使能运放功能；
2. 设置运放进入调节模式；
3. 设置运放调节模式从正端输入或者负端输入，输入端不能悬空；
4. 将调节位 OPA0ADJ<4:0> 设置成初始值，最大(1FH)或最小(00H)；
5. 延时一段时间，该时间和外部电容参数有关。
6. 读取运放输出；
7. 将调节位自减 1（初始值设置成最大 1FH）或者自加 1（初始值设置成 00H）；
8. 延时；
9. 读取运放输出，是否发生改变，如果没有改变，则继续执行步骤 7；
10. 读取值发生改变，调零结束，将 OPA0COFM 清零，进入正常工作模式。

## 22 触控按键(TOUCH KEY)

触摸按键提供了方便简单而可靠的方法来实现手指触摸的检测。在触摸应用中，并不需要一个外接电容组件。RM1221A 集成了硬件数字滤波器，可以实现任意 4 键自动检测，最多支持 26 通道触摸按键检测。RM1221A 也可以实现水位检测功能。

## 23 LCD

### 23.1 概述

RM1221A 所有 IO 口通过配置寄存器都可以作为 LCD 的 COM 或 SEG，每个 IO 除了正常 IO 的功能外，还可以输出两组可选电压：1/3VDD，2/3VDD 和 1/2VDD。用户可根据使用情况，随意选择任意 IO 作为 LCD 驱动的 COM 或者 SEG，LCD 的驱动能力可配置，LCD 控制信号（COM 和 SEG）由软件程序实现。

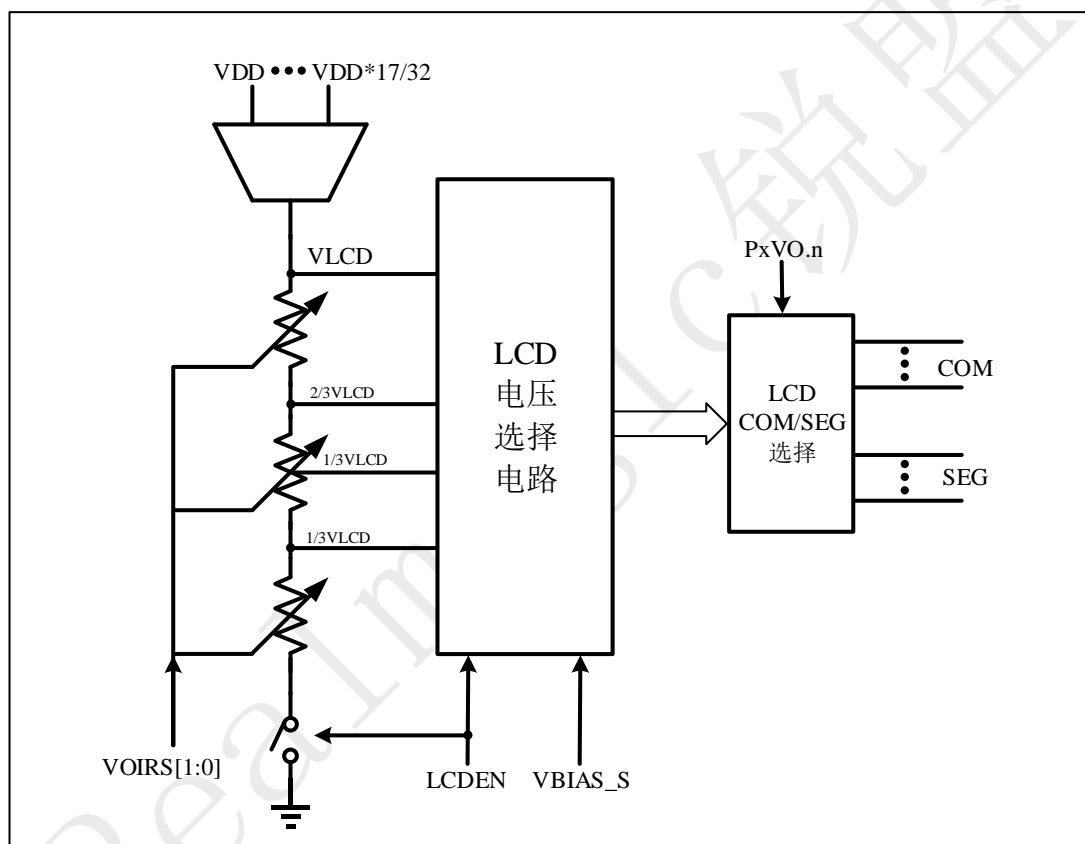


图 23-1 LCD 系统框图

### 23.2 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>LCDCON</b>	0xF030	R/W	LCD 模块控制寄存器	0000 0000
<b>PAVO</b>	0xF031	R/W	PA 口输出选择寄存器	0000 0000
<b>PBVO</b>	0xF032	R/W	PB 口输出选择寄存器	0000 0000
<b>PCVO</b>	0xF033	R/W	PC 口输出选择寄存器	0000 0000

PDVO	0xF034	R/W	PD 口输出选择寄存器	0000 0000
------	--------	-----	-------------	-----------

## 23.3 寄存器描述

### 23.3.1 LCD 模块控制寄存器 (LCDCON)

寄存器	地址	R/W	描述	复位后的值
LCDCON	0xF030	R/W	LCD 模块控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
LCDEN	保留		VBIAS	VLCDREF[3:0]			

位	描述																																				
[7]	LCDEN	<b>LCD 使能位</b> 0: 关闭 LCD 1: 使能 LCD																																			
[6:5]	保留	保留																																			
[4]	VBIAS	<b>LCD 电压输出口 BIAS 选择位</b> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>VBIASS</th> <th>PORTx</th> <th>PxUR</th> <th>PxDR</th> <th>LCD 输出口电压</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>0</td> <td>X</td> <td>0</td> <td>GND</td> </tr> <tr> <td>0</td> <td>0</td> <td>X</td> <td>1</td> <td>1/2VLCD</td> </tr> <tr> <td>0</td> <td>1</td> <td>X</td> <td>X</td> <td>VLCD</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1/3VLCD</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>2/3VLCD</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>VLCD</td> </tr> </tbody> </table>	VBIASS	PORTx	PxUR	PxDR	LCD 输出口电压	X	0	X	0	GND	0	0	X	1	1/2VLCD	0	1	X	X	VLCD	1	0	0	1	1/3VLCD	1	0	1	1	2/3VLCD	1	1	X	X	VLCD
		VBIASS	PORTx	PxUR	PxDR	LCD 输出口电压																															
		X	0	X	0	GND																															
		0	0	X	1	1/2VLCD																															
		0	1	X	X	VLCD																															
		1	0	0	1	1/3VLCD																															
		1	0	1	1	2/3VLCD																															
1	1	X	X	VLCD																																	
[3:0]	VLCDREF[3:0]	<b>LCD 参考电压选择</b> 0000: VLCD=17/32VDD 0001: VLCD=18/32VDD 0010: VLCD=19/32VDD 0011: VLCD=20/32VDD 0100: VLCD=21/32VDD 0101: VLCD=22/32VDD 0110: VLCD=23/32VDD 0111: VLCD=24/32VDD 1000: VLCD=25/32VDD 1001: VLCD=26/32VDD 1010: VLCD=27/32VDD 1011: VLCD=28/32VDD 1100: VLCD=29/32VDD 1101: VLCD=30/32VDD																																			

	1110: VLCD=31/32VDD 1111: VLCD=VDD
--	---------------------------------------

### 23.3.2 PA 口输出选择寄存器 (PAVO)

寄存器	地址	R/W	描述	复位后的值
<b>PAVO</b>	0xF031	R/W	PA 口输出选择寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PAVO[7:0]</b>							

D

位	描述	
[7:0]	<b>PAVO[7:0]</b>	<b>PA 口输出选择位</b> 0: PA.x 为普通 IO 1: PA.x 输出 LCD 电压 (1/2VLCD、1/3VLCD、2/3VLCD、VLCD、GND)

### 23.3.3 PB 口输出选择寄存器 (PBVO)

寄存器	地址	R/W	描述	复位后的值
<b>PBVO</b>	0xF032	R/W	PB 口输出选择寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留		<b>PBVO[6:0]</b>					

位	描述	
[7]	保留	保留
[6:0]	<b>PBVO[6:0]</b>	<b>PB 口输出选择位</b> 0: PB.x 为普通 IO 1: PB.x 输出 LCD 电压 (1/2VLCD、1/3VLCD、2/3VLCD、VLCD、GND)

### 23.3.4 PC 口输出选择寄存器 (PCVO)

寄存器	地址	R/W	描述	复位后的值
<b>PCVO</b>	0xF033	R/W	PC 口输出选择寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>PCVO[7:0]</b>							



位	描述	
[7:0]	PCVO[7:0]	<b>PC 口输出选择位</b> 0: PC.x 为普通 IO 1: PC.x 输出 LCD 电压 (1/2VLCD、1/3VLCD、2/3VLCD、VLCD、GND)

### 23.3.5 PD 口输出选择寄存器 (PDVO)

寄存器	地址	R/W	描述	复位后的值
PDVO	0xF034	R/W	PD 口输出选择寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留		VOIRS[1:0]		保留	PDVO[2:0]		

位	描述	
[7:6]	保留	保留
[5:4]	VOIRS[1:0]	<b>LCD 电压输出口分压电阻选择</b> 00: 关闭内部分压电阻 01: 内部分压电阻为 12.5K 10: 内部分压电阻为 37.5K 11: 内部分压电阻为 87.5K
[3]	保留	保留
[2:0]	PDVO[2:0]	<b>PD 口输出选择位</b> 0: PD.x 为普通 IO 1: PD.x 输出 LCD 电压 (1/2VLCD、1/3VLCD、2/3VLCD、VLCD、GND)

## 23.4 LCD 使用说明

LCD 的 COM/SEG 的驱动信号为交流模拟信号, LCD 像素点的显示对比度取决于此显示点上的 COM 与 SEG 电压( $V_{com}-V_{seg}$  /  $V_{seg}-V_{com}$ ), 当此电压差大于 LCD 的饱和电压就能打开此点的显示, 小于 LCD 的饱和电压就能关闭此 LCD 点的显示。这需要每个 COM/SEG 在一个扫描周期内根据用户需要在选通和非选通电压之间切换。

一个完整的 LCD 波形周期包含两个 Frame, 即 Frame0 和 Frame1, 通过 Frame0 和 Frame1 实现交流驱动信号。每个 Frame 时间可由软件用定时器控制实现。

#### Frame0

在 Frame0 中, COM 信号输出可以是 VDD, 或是  $V_{BIAS}=1/3VDD$  ( $1/2VDD$ );  
 在 Frame0 中, SEG 信号输出可以是 GND, 或是  $V_{BIAS}=2/3VDD$  ( $1/2VDD$ )。

#### Frame1

在 Frame1 中, COM 信号输出可以是 GND, 或是  $V_{BIAS}=2/3VDD$  ( $1/2VDD$ );

在 Frame1 中，SEG 信号输出可以是 VDD，或是 VBIAS=1/3VDD (1/2VDD)。

通过软件在定时器定时到期后设定 FRAME 位及相应的 I/O 寄存器来决定 COM 口目前输出的是 VDD，GND 或 VBIAS。

通过软件在定时器定时到期后设定 FRAME 位及相应的 I/O 寄存器来决定 SEG 口目前输出是 VDD，GND 或 VBIAS (在 1/2bias 时，SEG 只输出 VDD 或 GND)。

下面的波形图显示了一个利用应用程序产生的典型 1/2Bias LCD 波形。COMn 和 SEGm 引脚上所产生的 COM 和 SEG 信号极性通过相应的 I/O 相关寄存器位来产生。

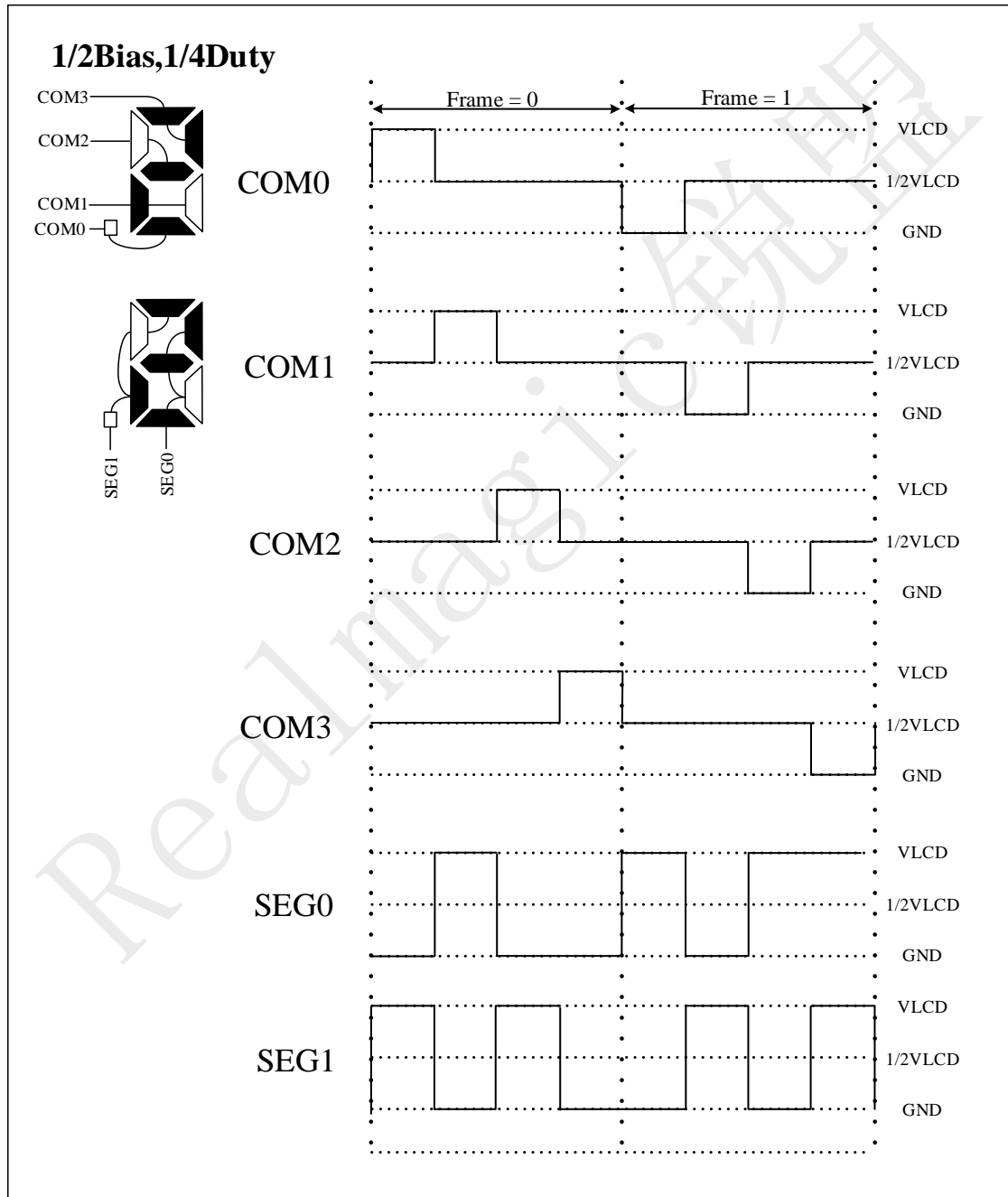


图 23-4-1 1/2LCD 波形图

1/2Bias, 1/4Duty LCD COM0~COM3 一轮扫描的 COM&SEG 电平和显示关系如下表:

Frame	Time	COM0	COM1	COM2	COM3	SEG0 点亮	SEG1 不点亮
Frame0	t0	VLCD	1/2 VLCD	1/2 VLCD	1/2 VLCD	GND	VLCD
	t1	1/2VLCD	VLCD	1/2 VLCD	1/2 VLCD	GND	VLCD
	t2	1/2 VLCD	1/2 VLCD	V VLCD	1/2 VLCD	GND	VLCD
	t3	1/2 VLCD	1/2 VLCD	1/2 VLCD	VLCD	GND	VLCD
Frame1	t4	GND	1/2 VLCD	1/2 VLCD	1/2 VLCD	VLCD	GND
	t5	1/2 VLCD	GND	1/2 VLCD	1/2 VLCD	VLCD	GND
	t6	1/2 VLCD	1/2 VLCD	GND	1/2 VLCD	VLCD	GND
	t7	1/2 VLCD	1/2 VLCD	1/2 VLCD	GND	VLCD	GND

此时如果选择的 1/2BIAS LCD 电压输出口输出的具体电压跟寄存器 Px\_OUT、Px\_PU、PA\_PD 具体 bit 的值有关，即

Px\_OUT.y=0, Px\_PD=0, Px\_PU=X 时，设定 LCD 电压输出口输出电压为 GND；

Px\_OUT.y=0, Px\_PD=1, Px\_PU=X 时，设定 LCD 电压输出口输出电压为 1/2VLCD；

Px\_OUT.y=1, Px\_PD=X, Px\_PU=X 时，设定 LCD 电压输出口输出电压为 VLCD。

下面的波形图显示了一个利用应用程序产生的典型 1/3Bias LCD 波形。COMn 和 SEGm 引脚上所产生的 COM 和 SEG 信号极性通过相应的相应的 I/O 寄存器位来产生。

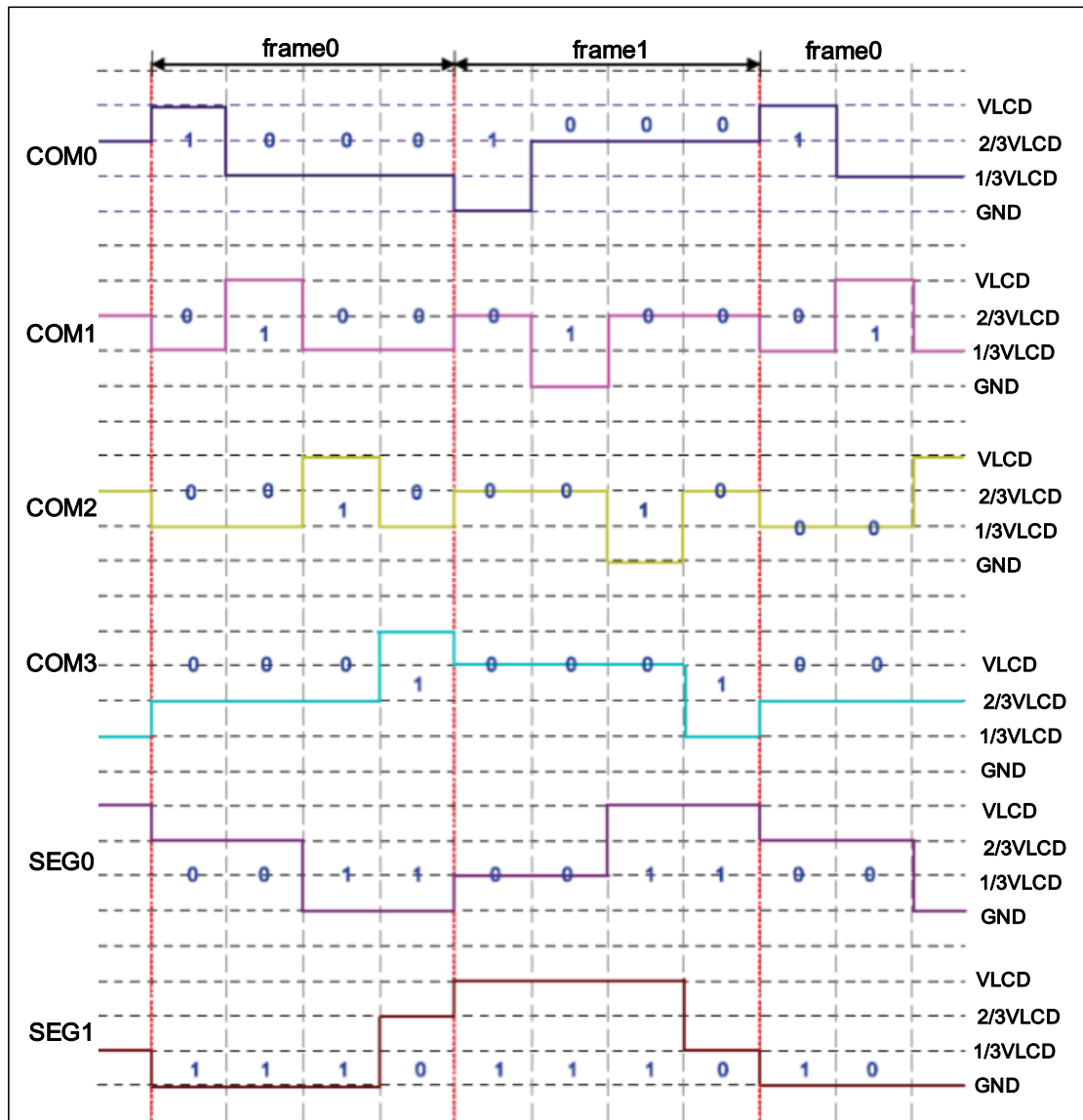


图 23-4-2 1/3LCD 波形图

此时如果选择的 1/2BIAS LCD 电压输出输出的具体电压跟寄存器 Px\_OUT、Px\_PU、PA\_PD 具体 bit 的值有关，即

- Px\_OUT.y=0, Px\_PD=0, Px\_PU=X 时，设定 LCD 电压输出输出电压为 GND；
- Px\_OUT.y=0, Px\_PD=1, Px\_PU=0 时，设定 LCD 电压输出输出电压为 1/3VLCD；
- Px\_OUT.y=0, Px\_PD=1, Px\_PU=1 时，设定 LCD 电压输出输出电压为 2/3VLCD；
- Px\_OUT.y=1, Px\_PD=X, Px\_PU=X 时，设定 LCD 电压输出输出电压为 VLCD。

## 24 乘除法器

### 24.1 概述

RM1221A 提供了 1 个 16 位的乘除法器，由扩展累加器 EXA0~EXA3、扩展 B 寄存器 EXB0~EXB1 和运算控制寄存器 MDCON 组成。可取代软件进行 16 位×16 位乘法运算和 32 位÷16 位除法运算。

### 24.2 寄存器列表

寄存器	地址	R/W	描述	复位值
MDCON	0xB7	R/W	乘除法运行控制寄存器	0000 0000
EXA0	0xB1	R/W	扩展累加器 0	0000 0000
EXA1	0xB2	R/W	扩展累加器 1	0000 0000
EXA2	0xB3	R/W	扩展累加器 2	0000 0000
EXA3	0xB4	R/W	扩展累加器 3	0000 0000
EXB0	0xB5	R/W	扩展 B 寄存器 0	0000 0000
EXB1	0xB6	R/W	扩展 B 寄存器 1	0000 0000

### 24.3 寄存器描述

#### 24.3.1 乘除法运行控制寄存器（MDCON）

寄存器	地址	R/W	描述	复位后的值
MDCON	0xB7	R/W	乘除法运行控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
OPERS	MD	保留					

位	描述																
[7]	OPERS	乘法器运算开始位，置 1 后，开始做一次乘除法运算，运算完后，硬件自动清 0。															
[6]	MD	乘除法选择位															
		0: 乘法运算，被乘数和乘数的写入，乘积读取如下：															
		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>运算数</th> <th>字节 3</th> <th>字节 2</th> <th>字节 1</th> <th>字节 0</th> </tr> </thead> <tbody> <tr> <td>被乘数 16 位</td> <td></td> <td></td> <td>EXA1</td> <td>EXA0</td> </tr> <tr> <td>乘数 16 位</td> <td></td> <td></td> <td>EXB1</td> <td>EXB0</td> </tr> </tbody> </table>	运算数	字节 3	字节 2	字节 1	字节 0	被乘数 16 位			EXA1	EXA0	乘数 16 位			EXB1	EXB0
		运算数	字节 3	字节 2	字节 1	字节 0											
被乘数 16 位			EXA1	EXA0													
乘数 16 位			EXB1	EXB0													

		乘积 32 位	EXA3	EXA2	EXA1	EXA0
		1: 除法运算, 被除数和除数的写入, 商和余数读取如下:				
		运算数	字节 3	字节 2	字节 1	字节 0
		被除数 32 位	EXA3	EXA2	EXA1	EXA0
		除数 16 位			EXB1	EXB0
		商 32 位	EXA3	EXA2	EXA1	EXA0
		余数 16 位			EXB1	EXB0
[5:0]	保留	保留				

### 24.3.2 扩展累加器 (EXAx)

寄存器	地址	R/W	描述	复位后的值
EXA0	0xB1	R/W	扩展累加器 0	0000 0000
EXA1	0xB2	R/W	扩展累加器 1	0000 0000
EXA2	0xB3	R/W	扩展累加器 2	0000 0000
EXA3	0xB4	R/W	扩展累加器 3	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
EXAx[7:0] x=0~3							

位	描述	
[7:0]	EXAx[7:0] x=0~3	扩展累加器

### 24.3.3 扩展 B 寄存器 (EXBx)

寄存器	地址	R/W	描述	复位后的值
EXB0	0xB5	R/W	扩展 B 寄存器 0	0000 0000
EXB1	0xB6	R/W	扩展 B 寄存器 1	0000 0000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
EXBx[7:0] x=0~3							

位	描述	
[7:0]	EXBx[7:0] x=0~3	扩展 B 寄存器

## 25 IAP、EEPROM、BootLoader 及安全加密

### 25.1 概述

修改 MTP 数据通常需要很长时间，不像 RAM 那样可以实时操作。而且擦除、编程或读取 MTP 数据需要遵循相当复杂的时序步骤。RM1221A 提供方便 MTP 编程方式，可以帮助用户通过 IAP 方式，重新编程 MTP 内容。IAP 就是通过软件实现在线电擦除和编程的方法。

通过设置 IAPEN(CHPCON.0 受 WPT 保护)使能 IAP, 用户将 16 位操作地址写入 IAPAH 和 IAPAL, 数据写入 IAPFD, 命令写入 IAPCN。然后通过设置触发位 IAPGO(IAPTRG.0)。注意: IAPTRG 也受 WPT 保护。此时, CPU 保持程序计数器, 内嵌 IAP 自动控制内部充电泵提高电压和信号时序。IAP 动作完成后, 程序计数器继续运行之后的指令, IAPGO 位将自动清零。通过这些纯软件的设置, 用户可以很方便对 FLASH 存储器进行擦除、编程和校验。

**注意:**

在进行 IAP 时, 不要开 IO 唤醒或中断, 如果中断打开应该临时清除 EA 位。

擦除或编程的区域不能是当前代码执行的区域。否则会出现不可预计程序动作, 甚至破坏存储的数据。

### 25.2 寄存器列表

寄存器	地址	R/W	描述	复位值
<b>IAPTRG</b>	0xBA	R/W	IAP 执行寄存器	00000000
<b>IAPAL</b>	0xBB	R/W	IAP 地址低字节	00000000
<b>IAPAH</b>	0xBC	R/W	IAP 地址高字节	00000000
<b>IAPFD</b>	0xBD	R/W	IAP 数据寄存器	00000000
<b>IAPCN</b>	0xBE	R/W	IAP 控制寄存器	00000000
<b>CHPCON</b>	0xBF	R/W	IAP 使能寄存器	00000000

### 25.3 寄存器描述

#### 25.3.1 IAP 执行寄存器 (IAPTRG) (受 WPT 保护)

寄存器	地址	R/W	描述	复位后的值
<b>IAPTRG</b>	0xBA	R/W	IAP 执行寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留							IAPGO

位	描述	
[7:1]	保留	保留
[0]	<b>IAPGO</b>	<p><b>IAP 执行位</b></p> <p>设置该位为 1 开始执行 IAP 操作。</p> <p>该指令后，CPU 保持程序计数器(PC)，IAP 硬件自动管理控制该过程。IAP 完成后，程序计数器继续执行。IAPGO 位自动清零，保持为 0。</p> <p>在触发 IAP 动作前，如果中断打开应该临时关闭，程序过程如下：</p> <pre>CLR    EA MOV    WPT,#AAH MOV    WPT,#55H ORL    IAPTRG, #01H NOP NOP NOP NOP (SETB  EA)</pre> <p><b>注：IAPGO 后需发 4 个 NOP 指令，以保证 IAP 动作能顺利完成。</b></p>

### 25.3.2 IAP 地址低字节 (IAPAL)

寄存器	地址	R/W	描述	复位后的值
<b>IAPAL</b>	0xBB	R/W	IAP 地址低字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IAPA[7:0]							

位	描述	
[7:0]	<b>IAPA[7:0]</b>	IAP 地址低字节

### 25.3.3 IAP 地址高字节 (IAPAH)

寄存器	地址	R/W	描述	复位后的值
<b>IAPAH</b>	0xBC	R/W	IAP 地址高字节	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0



<b>IAPA[15:8]</b>
-------------------

位	描述	
[7:0]	<b>IAPA[15:8]</b>	IAP 地址高字节

### 25.3.4 IAP 数据寄存器 (IAPFD)

寄存器	地址	R/W	描述	复位后的值
<b>IAPFD</b>	0xBD	R/W	IAP 数据寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>IAPFD[7:0]</b>							

位	描述	
[7:0]	<b>IAPFD[7:0]</b>	<b>IAP 数据</b> 该字节包含将要读或写进内存空间的数据。编程模式下，用户需要在触发 IAP 之前写数据到 IAPFD 里，读/校验模式下，在 IAP 完成后从 IAPFD 读出数据

### 25.3.5 IAP 控制寄存器 (IAPCON)

寄存器	地址	R/W	描述	复位后的值
<b>IAPCON</b>	0xBE	R/W	IAP 控制寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
保留		<b>CHIPES[1:0]</b>		<b>IAPCN[3:0]</b>			

位	描述	
[7:6]	保留	保留
[5:4]	<b>CHIPES[1:0]</b>	当对 MTP 选择 Chip 擦除时，指定可擦除的区域 00: 禁止对 MTP 擦除 01: 选择 MTP 的 Page0-14 区域 10: 选择 MTP 的 Page15 区域（需要在 Page15 开放的情况下才可执行擦除） 11: 选择 MTP 的 Page0-15 区域
[3:0]	<b>IAPCN[3:0]</b>	<b>IAP 控制位</b> 0000: 保留 0001: 读 MTP 0010: 高速写 MTP 0011: 低速写 MTP 0100: Sector 擦除 MTP (1K Bits)

	0101: Page 擦除 MTP (1K*8 Bits)
	0110: Chip 擦除 MTP (16K Bits)
	0111: 保留
	1000: 保留
	1001: 读 EEPROM
	1010: 高速写 EEPROM
	1011: 低速写 EEPROM
	1100: Page 擦除 EEPROM (32*8 Bits)
	1101: Sector 擦除 EEPROM (32 Bits)
	1110: 保留
	1111: Byte 擦除 EEPROM (1*8 Bits)

### 25.3.6 IAP 使能寄存器 (CHPCON) (受 WPT 保护)

寄存器	地址	R/W	描述	复位后的值
CHPCON	0xBF	R/W	IAP 使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SWRST	保留			BS		IAPEN	

位	描述	
[7]	SWRST	<b>软件复位</b> 对该位写 1, 芯片执行软件复位, 复位完成后该位自动清零。
[6:2]	保留	保留
[1]	BS	<b>启动选择位</b> 该位初始值为 option 的 CBS 值, 软件复位不会改变此值。 0: 由 APROM 启动 1: 由 LDROM 启动
[0]	IAPEN	<b>IAP 使能位</b> 0: 禁止 IAP 1: 使能 IAP

## 25.4 BootLoader

用户可以通过 1K BYTES LDROM 的空间来实现 ISP (In System Programing) 功能: ISP 运行时, IC 运行的是 LDROM 区的引导代码, 引导代码执行时会通过串口接收新的程序代码, 再将接收到的代码通过 IAP 命令编程到用户代码区域。整个过程不需将芯片从系统办事拆下来, 也不需要烧录器。BootLoader 程序不能通过 KEIL 直接下载, 需要专用下载软件, 配合 Link 来下载。

**BootLoader 操作相关 SFR 位定义如下：**

寄存器	地址	R/W	描述	复位后的值
<b>AUXR1</b>	9EH	R/W	辅助功能寄存器 1	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>SWRF</b>							

位	描述	
[7]	<b>SWRF</b>	<b>软件复位标志位</b> 软件复位后，该位将被硬件置 1，建议复位发生后通过软件清零
[6:0]		

寄存器	地址	R/W	描述	复位后的值
<b>CHPCON</b>	9FH	R/W	IAP 使能寄存器	00000000

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<b>SWRST</b>	保留				<b>BS</b>		<b>IAPEN</b>

位	描述	
[7]	<b>SWRST</b>	<b>软件复位</b> 对该位写 1，芯片执行软件复位，复位完成后该位自动清零。
[6:2]	保留	保留
[1]	<b>BS</b>	<b>启动选择位</b> 该位初始值为 Code Option 的 CBS 值，软件复位不会改变此值。 0：由 APROM 启动 1：由 LDROM 启动
[0]	<b>IAPEN</b>	<b>IAP 使能位</b> 0：禁止 IAP 1：使能 IAP

**注：**BS 位由复位后（除软件复位外）读取代码选项 **OPTION0.3 CBS** 位内容的值并写入，软件复位后保持不变。CPU 从所有复位状态释放后，硬件将检查 BS 位（非 CBS）以决定是由 APROM 还是 LDROM 启动。CHPCON 寄存器受 WPT 保护。

**代码选项区 BootLoader 相关位定义**

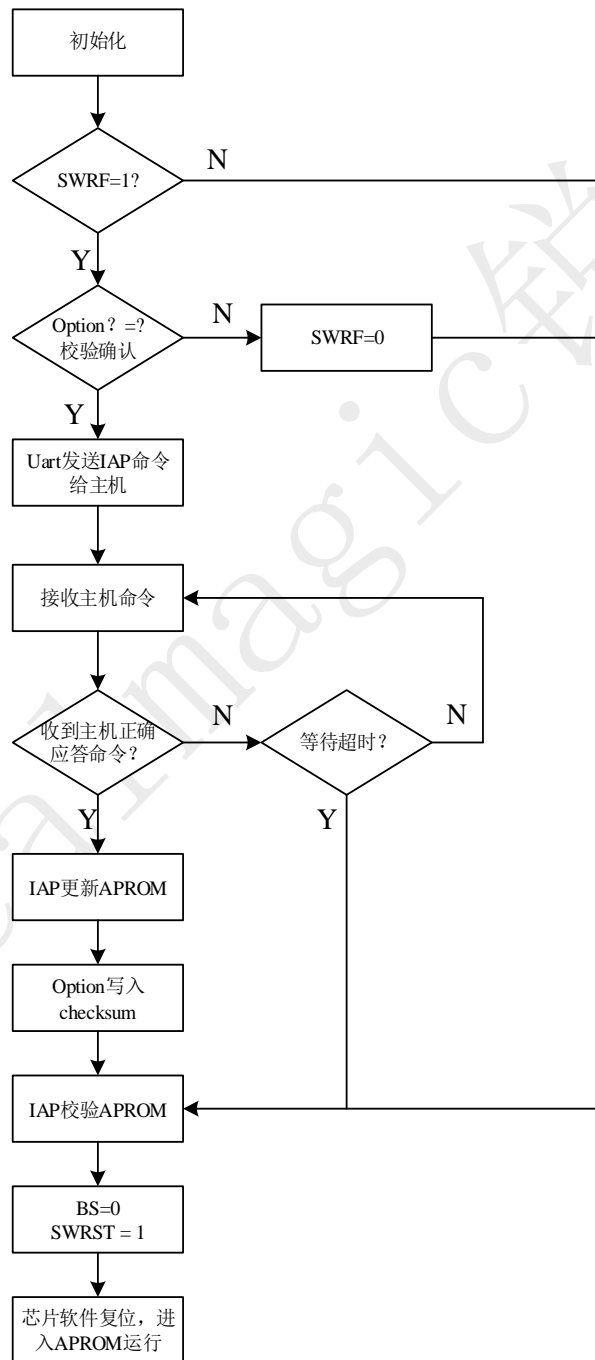
#### OPTION0

位	描述	
[7:4]	保留	保留
[3]	<b>CBS</b>	<b>CBS 配置启动选择位</b> 该位值会在芯片上电后加载到寄存器 BS 里，软件复位不会加载此值。 0：除软件复位外的所有复位后，MCU 从 APROM 启动 1：除软件复位外的所有复位后，MCU 从 LDROM 启动
[2]	<b>BOOTEN</b>	<b>BootLoader 使能位</b>

		0: 不使能 BootLoader 功能, code 区共有 16Kbyte 1: 使能 BootLoader 功能, code 区共有 15Kbyte
[1:0]		

应该过程描述:

Code Option 的 CBS 配置为 1, 上电后芯片会先进入 LDROM 区, 运行 BootLoader 程序, 执行完初始化、自检和更新后, 程序会将 BS 置 0, SWRST 置 1, 芯片软件复位, 然后进入 APROM 区运行。主程序也可以通过将 BS 置 1, SWRST 置 1, 软件复位进入 BootLoader 程序。



BootLoader程序流程图

## 25.5 安全加密

MTP 加密分烧录加密和应用加密。烧录加密可以实现用户保护区域不能被读取，也可以使得这部分区域不被再次烧录。应用加密则是指被保护的区域不能被 IAP 读写和 MOV 读取。加密区域如果要解密则必须对该区域从头连续写到尾后才可更改加密位，解密发生在对该区域写完值后。

用户可通过烧录上位机界面的“加密”设置项来选择是否开启 RM1221A 的 ROM 安全加密功能，具体操作方法参考锐盟开发量产相关资料。

## 26 指令集

指令	功能描述	代码	字节	时钟周期
NOP	无操作	00	1	1
<b>算术操作指令</b>				
ADD A, Rn	累加器加寄存器	28~2F	1	2
ADD A, direct	累加器加直接寻址字节	25	2	3
ADD A, @Ri	累加器加内部RAM	26, 27	1	4
ADD A, #data	累加器加立即数	24	2	2
ADDC A, Rn	累加器加寄存器和进位位	38~3F	1	2
ADDC A, direct	累加器加直接寻址字节和进位位	35	2	3
ADDC A, @Ri	累加器加内部RAM和进位位	36, 37	1	4
ADDC A, #data	累加器加立即数和进位位	34	2	2
SUBB A, Rn	累加器减寄存器和借位位	98~9F	1	2
SUBB A, direct	累加器减直接寻址字节和借位位	95	2	3
SUBB A, @Ri	累加器减内部RAM和借位位	96, 97	1	4
SUBB A, #data	累加器减立即数和借位位	94	2	2
INC A	累加器加1	04	1	1
INC Rn	寄存器加1	08~0F	1	3
INC direct	直接寻址字节加1	05	2	4
INC @Ri	内部RAM加1	06, 07	1	5
INC DPTR	数据指针加1	A3	1	1
DEC A	累加器减1	14	1	1
DEC Rn	寄存器减1	18~1F	1	3
DEC direct	直接寻址字节减1	15	2	4
DEC @Ri	内部RAM减1	16, 17	1	5
MUL AB	累加器乘寄存器B	A4	1	4
DIV AB	累加器除以寄存器B	84	1	4
DA A	十进制调整	D4	1	1
<b>逻辑操作指令</b>				
ANL A, Rn	累加器与寄存器	58~5F	1	2
ANL A, direct	累加器与直接寻址字节	55	2	3
ANL A, @Ri	累加器与内部RAM	56, 57	1	4
ANL A, #data	累加器与立即数	54	2	2
ANL direct, A	直接寻址字节与累加器	52	2	4
ANL direct, #data	直接寻址字节与立即数	53	3	4
ORL A, Rn	累加器或寄存器	48~4F	1	2
ORL A, direct	累加器或直接寻址字节	45	2	3

ORL	A, @Ri	累加器或内部RAM	46, 47	1	4
ORL	A, #data	累加器或立即数	44	2	2
ORL	direct, A	直接寻址字节或累加器	42	2	4
ORL	direct, #data	直接寻址字节或立即数	43	3	4
XRL	A, Rn	累加器异或寄存器	68~6F	1	2
XRL	A, direct	累加器异或直接寻址字节	65	2	3
XRL	A, @Ri	累加器异或内部RAM	66, 67	1	4
XRL	A, #data	累加器异或立即数	64	2	2
XRL	direct, A	直接寻址字节异或累加器	62	2	4
XRL	direct, #data	直接寻址字节异或立即数	63	3	4
CLR	A	累加器清零	E4	1	1
CPL	A	累加器取反	F4	1	1
RL	A	累加器左环移位	23	1	1
RLC	A	累加器连进位标志左环移位	33	1	1
RR	A	累加器右环移位	03	1	1
RRC	A	累加器连进位标志右环移位	13	1	1
SWAP	A	累加器高4位与低4位交换	C4	1	1
<b>数据传送指令</b>					
MOV	A, Rn	寄存器送累加器	E8~EF	1	1
MOV	A, direct	直接寻址字节送累加器	E5	2	3
MOV	A, @Ri	内部RAM送累加器	E6, E7	1	4
MOV	A, #data	立即数送累加器	74	2	2
MOV	Rn, A	累加器送寄存器	F8~FF	1	1
MOV	Rn, direct	直接寻址字节送寄存器	A8~AF	2	4
MOV	Rn, #data	立即数送寄存器	78~7F	2	2
MOV	direct, A	累加器送直接寻址字节	F5	2	2
MOV	direct, Rn	寄存器送直接寻址字节	88~8F	2	3
MOV	direct, direct	直接寻址字节送直接寻址字节	85	3	4
MOV	direct, @Ri	内部RAM送直接寻址字节	86, 87	2	5
MOV	direct, #data	立即数送直接寻址字节	75	3	3
MOV	@Ri, A	累加器送内部RAM	F6, F7	1	3
MOV	@Ri, direct	直接寻址字节送内部RAM	A6, A7	2	4
MOV	@Ri, #data	立即数送内部RAM	76, 77	2	3
MOV	DPTR, #data16	16位立即数送数据指针	90	3	3
MOVC	A, @A+DPTR	程序代码送累加器（相对数据指针）	93	1	4
MOVC	A, @A+PC	程序代码送累加器（相对程序计数器）	83	1	4
MOVX	A, @Ri	外部RAM送累加器（8位地址）	E2, E3	1	5
MOVX	A, @DPTR	外部RAM送累加器（16位地址）	E0	1	4
MOVX	@Ri, A	累加器送外部RAM（8位地址）	F2, F3	1	6
MOVX	@DPTR, A	累加器送外部RAM（16位地址）	F0	1	5
PUSH	direct	直接寻址字节压入栈顶	C0	2	4

POP	direct	栈顶弹至直接寻址字节	D0	2	3
XCH	A, Rn	累加器与寄存器交换	C8~CF	1	2
XCH	A, direct	累加器与直接寻址字节交换	C5	2	3
XCH	A, @Ri	累加器与内部RAM交换	C6, C7	1	4
XCHD	A, @Ri	累加器低4位与内部RAM低4位交换	D6, D7	1	5
<b>位操作指令</b>					
CLR	C	C清零	C3	1	1
CLR	bit	直接寻址位清零	C2	2	4
SETB	C	C置位	D3	1	1
SETB	bit	直接寻址位置位	D2	2	4
CPL	C	C取反	B3	1	1
CPL	bit	直接寻址位取反	B2	2	4
ANL	C, bit	C逻辑与直接寻址位	82	2	3
ANL	C, /bit	C逻辑与直接寻址位的反	B0	2	3
ORL	C, bit	C逻辑或直接寻址位	72	2	3
ORL	C, /bit	C逻辑或直接寻址位的反	A0	2	3
MOV	C, bit	直接寻址位送C	A2	2	3
MOV	bit, C	C送直接寻址位	92	2	4
<b>控制程序转移指令</b>					
ACALL	addr11	2KB内绝对调用	11, 31, 51, 71, 91, B1, D1, F1	2	4
LCALL	addr16	64KB内长调用	12	3	4
RET		子程序返回	22	1	5
RETI		中断返回	32	1	5
AJMP	addr11	2KB内绝对转移	01, 21, 41, 61, 81, A1, C1, E1	2	3
LJMP	addr16	64KB内长转移	02	3	4
SJMP	rel	相对短转移	80	2	3
JMP	@A+DPTR	相对长转移	73	1	3
JZ	rel	累加器为零转移	60	2	3
JNZ	rel	累加器为非零转移	70	2	3
JC	rel	C置位转移	40	2	3
JNC	rel	C清零转移	50	2	3
JB	bit, rel	直接寻址位置位转移	20	3	5
JNB	bit, rel	直接寻址位清零转移	30	3	5
JBC	bit, rel	直接寻址位置位转移并清该位	10	3	5
CJNE	A, direct, rel	累加器与直接寻址字节不等转移	B5	3	5
CJNE	A, #data, rel	累加器与立即数不等转移	B4	3	4
CJNE	Rn, #data, rel	寄存器与立即数不等转移	B8~BF	3	4
CJNE	@Ri, #data, rel	内部RAM与立即数不等转移	B6, B7	3	6
DJNZ	Rn, rel	寄存器减1不为零转移	D8~DF	2	4



DJNZ	direct, rel	直接寻址字节减1不为零转移	D5	3	5
------	-------------	---------------	----	---	---

## 27 Code Option 代码选项

代码选项是在烧录时候，通过烧录器烧录到 MTP 中。芯片每次上电都需要从此区域读出代码选项的值，来完成相关的配置。以下代码选项寄存器的地址指的是 MTP 程序区域地址，而不是 SFR 的地址。

### 27.1 OPTION0 (0x1E0)

位	描述	
[7:4]	保留	保留
[3]	<b>CBS</b>	<b>CBS 配置启动选择位</b> 该位值会在芯片上电后加载到寄存器 BS 里，软件复位不会加载此值。 0: 除软件复位外的所有复位后，MCU 从 APROM 启动 1: 除软件复位外的所有复位后，MCU 从 LDROM 启动
[2]	<b>BOOTEN</b>	BootLoader 使能位 0: 不使能 BootLoader 功能，code 区共有 16Kbyte 1: 使能 BootLoader 功能，code 区共有 15Kbyte
[1]	<b>MTPLOCK</b>	<b>MTP 加密使能位</b> 0: 芯片不加密。 1: 芯片加密。
[0]	<b>INFOLOCK</b>	<b>INFO 加密使能位</b> 0: 芯片不加密。 1: 芯片加密。

### 27.2 OPTION1 (0x1E1)

位	描述	
[7:6]	<b>OSC[1:0]</b> 和 <b>OSC[2]</b> — 一起使用	<b>FOSC 时钟源选择</b> 000: 系统时钟选择内部高速振荡器。PB.5 和 PB.6 均不复用时钟功能。 <b>STOP 模式下，硬件自动关闭内部时钟</b> 001: 系统时钟选择外部时钟。PB.6 作为外部时钟源接灌入时钟，PB.5 不复用时钟功能。 <b>STOP 模式下，硬件自动关闭内部时钟</b> 010: 系统时钟选择外部时钟。PB.5 和 PB.6 接外部晶振 4M~16MHz。 <b>STOP 模式下，硬件自动关闭内部时钟</b> 011: 系统时钟选择外部时钟。PB.5 和 PB.6 接外部晶振 32768Hz， <b>STOP 模式下，硬件自动关闭内部时钟</b> 100~111: 系统时钟选择内部时钟。PB.5 和 PB.6 接外部晶振 32768Hz，外部晶振 32768Hz 为 RTC 的时钟。 <b>STOP 模式下，硬件自动关闭内部时钟，外部时钟不关闭</b>

[5:4]	<b>CKDIV[1:0]</b>	<b>CPU 时钟 ICK 除频选择</b> 00: 1/2 01: 1/4 10: 1/1 11: 1/16
[3]	<b>OSC[2]</b>	与 <b>OSC[1:0]</b> 合并起来使用
[2:1]	保留	保留
[0]	<b>RSTSEL</b>	<b>芯片上电复位时间选择</b> 0: 12ms 1: 36ms

### 27.3 OPTION2 (0x1E2)

位	描述	
[7]	<b>LVRSOFEN</b>	<b>LVR 软件使能开关</b> 0: 关闭 LVREN 到 SFR 的映射, 软件不能控制 LVREN 1: 将 LVREN 映射到 SFR 中, 允许软件控制 LVREN
[6]	<b>LVREN</b>	<b>欠压复位使能</b> 0: 禁用欠压复位 1: 使能欠压复位
[5:4]	保留	保留
[3]	<b>RSTS</b>	<b>外部硬件复位引脚配置</b> 0: PD2 作为普通 IO 口 1: PD2 作为外部硬件复位口
[2:0]	<b>LVRS[2:0]</b>	<b>上电复位电压选择</b> 001: 2.3V 010: 2.7V 011: 3.0V 100: 3.3V 101: 3.6V 110: 3.9V 111: 4.3V

### 27.4 OPTION3 (0x1E3)

位	描述	
[7:2]	保留	保留
[1]	<b>IOVTH</b>	<b>除 PD2 外引脚输入的电平阈值配置</b> 0: 输入阈值高/低电平为 $0.7*VDD/0.3*VDD$

		1: 输入阈值高/低电平为 $0.5*VDD/0.2*VDD$ 输入阈值配置对除 PD2 外所有 IO 同时生效。 PD2 做复位脚时, 输入阈值高/低电平强制为 $0.7*VDD/0.3*VDD$ PD2 做普通 IO 口时 0: 输入阈值高/低电平为 $0.5*VDD/0.2*VDD$ 1: 输入阈值高/低电平为 $0.7*VDD/0.3*VDD$
[0]	保留	保留

## 27.5 OPTION5 (0x1E5)

位	描述	
[7:6]	<b>ADTS</b>	<b>ADC offset 控制位,代码选项映射 SFR</b>
[4:0]	<b>ADT[4:0]</b>	<b>ADC OFFSET 校准控制位</b> 00000: 最小负 offset 11111: 最大正 offset, 代码选项映射 SFR

## 27.6 OPTION19 (0x1F8)

位	描述	
[7]	保留	保留
[6:4]	<b>PAGE_VALID[2:0]</b>	<b>烧录/IAP 允许操作区域</b> 000: 0x0000~0x3FFF (默认) 001: 0x0000~0x37FF (14K) 010: 0x0000~0x2FFF (12K) 011: 0x0000~0x27FF (10K) 100: 0x0000~0x1FFF (8K) 101: 0x0000~0x17FF (6K) 110: 0x0000~0x0FFF (4K) 111: 0x0000~0x07FF (2K) 说明: Page_valid 用于限定编程区域, 不同于加密位, 在允许的区域可进行读写, 也可被加密位加密读不出来。
[3:1]	保留	保留
[0]	<b>MOVCINCEN</b>	<b>Movc 指令功能允许位</b> 0: movc 指令功能被禁止, 相当于 nop (设计目标限制区和 IAP 一致) 1: movc 指令功能正常, 不做限制

## 27.7 OPTION22 (0x1FD)

位	描述
---	----

[7:0]	CHECKSUM0[7:0]	CHECKSUM0[7:0]
-------	----------------	----------------

## 27.8 OPTION23 (0x1FE)

位	描述
[7:0]	CHECKSUM1[7:0]    CHECKSUM1[7:0]

## 28 电气特性

### 28.1 最大绝对额定值

参数	符号	条件	范围	单位
工作温度	TON	-	-40 to + 85	°C
储藏温度	TOFF	-	- 40 to + 125	°C
工作电压	VDD	-	- 0.3 to + 5.5	V
I/O 输入电压	VI	所有 I/O 口	- 0.3 to VDD + 0.3	V
I/O 输出电压	VO	所有输出口	- 0.3 to VDD + 0.3	V
I/O 口输出电流	IOH	单个 I/O 口工作时	25	mA
		所有 I/O 口工作时	100	mA
I/O 口输入电流	IOL	单个 I/O 口工作时	25	mA
		所有 I/O 口工作时	100	mA

注：如果芯片工作条件超过上表所列最大极限值,可能会对芯片负载能力和稳定性造成伤害。

### 28.2 DC 电气特性

(如无特殊说明, 测试环境为: V<sub>DD</sub> -V<sub>SS</sub> =5 V, TA = 25°C)

参数	符号	条件	最小值	典型值	最大值	单位	
工作电压	VDD	系统频率 <= 4MHz	2.2	-	5.5	V	
		系统频率 <= 16MHz	3.5	-	5.5		
IO 输入高	VIH	VDD = 1.8 to 5.5V	0.7VDD	-	VDD	V	
IO 输入低	VIL	VDD= 1.8 to5.5V	-	-	0.3VDD	V	
IO 高漏电流	ILIH	所有输入口 VIO = VDD	-	-	1	uA	
IO 低漏电流	ILIL	所有输入口 VIO = 0V	-	-	1	uA	
IO 拉电流	IOH1	所有输入口 VDD = 5V,VIO=4.3V		2		mA	
	IOH2			4			
	IOH3			8			
	IOH4			16			
IO 灌电流	IOL1	所有输入口 VDD =5V,VIO=0.8V		10		mA	
	IOL2			20			
	IOL3			40			
	IOL4			80			
IO 上拉电阻	RPU	VDD =5V,VIO = 0V		43		kΩ	
IO 下拉电阻	RPD	所有 IO VDD =5V,VIO = 0V		43		kΩ	
工作电流	IDD1 工作	CPUCLK=16MHz				mA	
		CPUCLK=4MHz				uA	
		CPUCLK=128KHz				uA	
	IDD2 空闲	CPUCLK=16MHz					uA
		CPUCLK=4MHz					uA
		CPUCLK=128KHz					uA

参数	符号	条件	最小值	典型值	最大值	单位
	IDD3 休眠	VDD = 3.3V (禁止 LVR)		0.4		uA
		VDD = 3.3V (使能 LVR)		34		
		VDD = 5V (禁止 LVR)		0.7		
		VDD = 5V (使能 LVR)		38		

### 28.3 内部 16 MHz 高速时钟(HIRC)

符号	参数	条件	最小值	典型值	最大值	单位
F <sub>HIRC</sub>	中心频率	-	-	16	-	MHz
	校准后内部时钟频率	TA = 25 °C, VDD = 5V	-1	-	+1	%
		TA = -40~85 °C, VDD=2.5V~5.5V	-4	-	+4	%

### 28.4 内部 128 KHz 低速时钟(LIRC)

符号	参数	条件	最小值	典型值	最大值	单位
F <sub>LIRC</sub>	中心频率	-	-	128	-	KHz
	校准后内部时钟频率	TA = 25 °C, VDD = 2.5V~5.5V	-4	-	+4	%
		TA = -40~85 °C, VDD=2.5V~5.5V	-15	-	+15	%

### 28.5 低压复位规格

参数	符号	复位值档位	条件	最小值	典型值	最大值
低电压复位 电压值	V <sub>LVR</sub>	2.3V		2.2V	2.3V	2.4V
		2.7V		2.6V	2.7V	2.8V
		3.0V		2.9V	3.0V	3.1V
		3.3V		3.2V	3.3V	3.4V
		3.5V		3.4V	3.5V	3.6V
		3.9V		3.8V	3.9V	4.0V
		4.3V		4.2V	4.3V	4.4V

## 28.6 欠压检测规格

参数	符号	复位值档位	条件	最小值	典型值	最大值
低电压检测 电压值	V <sub>LVD</sub>	2.3V		2.2V	2.3V	2.4V
		2.4V		2.3V	2.4V	2.5V
		2.5V		2.4V	2.5V	2.6V
		2.7V		2.6V	2.7V	2.8V
		3.0V		2.9V	3.0V	3.1V
		3.3V		3.2V	3.3V	3.4V
		3.5V		3.4V	3.5V	3.6V
		3.7V		3.6V	3.7V	3.8V
		3.9V		3.8V	3.9V	4.0V
		4.1V		4.0V	4.1V	4.2V
		4.3V		4.2V	4.3V	4.4V
4.5V		4.4V	4.5V	4.6V		

## 28.7 ADC 电气特性

(T<sub>A</sub>=25°C, 除非另有说明)

参数	符号	条件	最小值	典型值	最大值	单位
供电电压	V <sub>DD</sub>					V
精度	N <sub>R</sub>	GND ≤ V <sub>AIN</sub> ≤ V <sub>DD</sub>				bit
ADC 输入电压	V <sub>AIN</sub>		GND		V <sub>DD</sub>	V
ADC 输入电阻	R <sub>AIN</sub>	V <sub>AIN</sub> = 5V		10		Ω
ADC 转换电流 1	I <sub>ADC1</sub>	ADC 模块打开 V <sub>DD</sub> = 5 V, 电流 X1		1		mA
ADC 转换电流 2	I <sub>ADC2</sub>	ADC 模块打开 V <sub>DD</sub> = 5 V, 电流 X2		1.3		mA
微分非线性	DNL	V <sub>DD</sub> = 5V V <sub>REF</sub> = 5V ADC CLOCK = 2M ADC 采样周期 = 4		2		LSB
积分非线性	INL			4		LSB
偏移量误差	E <sub>Z</sub>					LSB
满刻度误差	E <sub>F</sub>					LSB
总绝对误差	E <sub>AD</sub>					LSB
ADC 转换时间 1	T <sub>ADC1</sub>	ADC CLOCK = 2M ADC 采样周期 = 4				μS
ADC 转换时间 2	T <sub>ADC2</sub>	ADC CLOCK = 1M ADC 采样周期 = 4				μS



## 28.8 OPA 电气特性

( $T_A=25^{\circ}\text{C}$ , 除非另有说明)

参数	符号	条件	最小值	典型值	最大值	单位
供电电压	$V_{DD}$		3.0		5.5	V
工作电流	$I_{OPA}$	$V_{DD}=5\text{V}$ , 无负载 $A0P=A0N=1\text{V}$				$\mu\text{A}$
共模输入范围	$V_{IN}$	$V_{DD}=5\text{V}$	GND		$V_{DD}$	V
输出电压范围	$V_{OUT}$	$V_{DD}=5\text{V}$ $I_{LOAD}=2\text{mA}$				V
失调电压	$V_{OFFSET}$	校准后				mV
增益带宽	GBW	$C_{LOAD}=100\text{pF}$				MHz
转换速率	SR	$V_{DD}=5\text{V}$ $C_{LOAD}=100\text{pF}$				V/ $\mu\text{s}$
开环增益	$A_{OL}$		60	90		dB
电源抑制比	PSRR		60	80		dB
共模抑制比	CMRR		60	90		dB

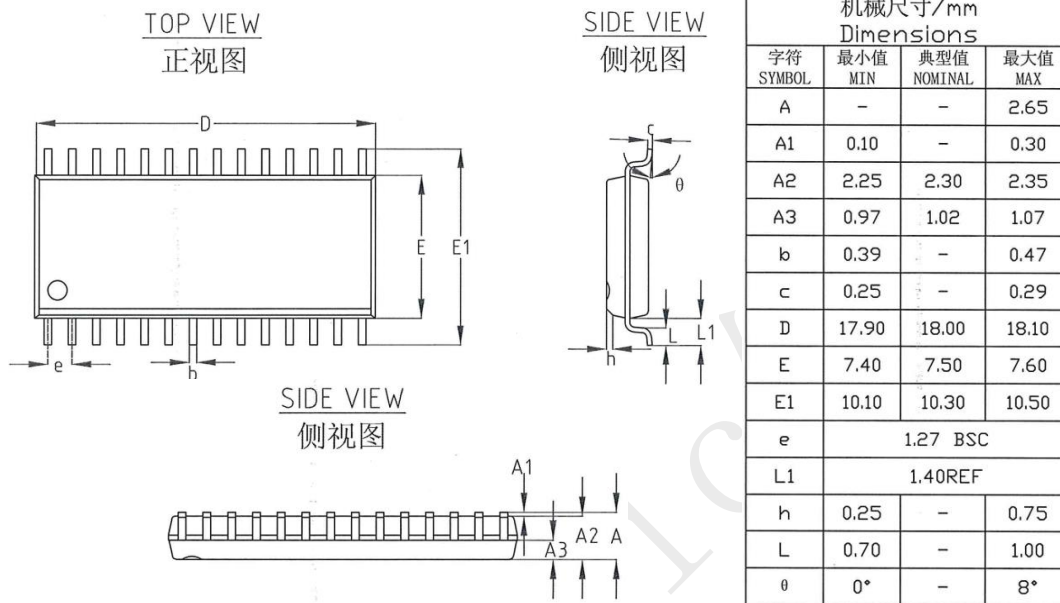
## 28.9 交流电气特性

( $T_A=25^{\circ}\text{C}$ , 除非另有说明)

参数	符号	条件	最小值	典型值	最大值	单位
外置高频振荡器起振时间	$T_{OSC1}$	外置 16MHz 晶体				ms
外置高频振荡器起振时间	$T_{OSC1}$	外置 32KHz 晶体				ms
POWER ON RESET 时间	$T_{POR}$					ms
PD 模式唤醒时间	$T_{PDW}$					ms
复位脉冲宽度	$T_{RESET}$					$\mu\text{s}$
内置 RC 振荡起振时间	$f_{HRC}$	$V_{DD}=3.3\text{V}\sim 5.5\text{V}$ $T_A=-40\sim 85^{\circ}\text{C}$				$\mu\text{s}$
MTP 编程时间	$T_{MTP}$	$V_{DD}=3\text{V}$				
		$V_{DD}=5\text{V}$				
EEPROM 编程时间	$T_{EEPROM}$	$V_{DD}=3\text{V}$				ms
		$V_{DD}=5\text{V}$				ms

## 29 封装信息

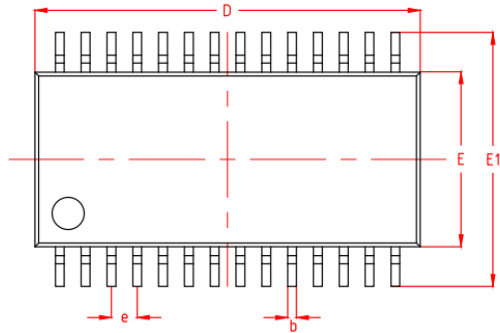
### 29.1 SOP28L 封装示意图



## 29.2 TSSOP28L 封装示意图

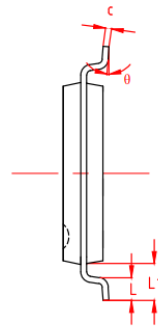
TOP VIEW

正视图



SIDE VIEW

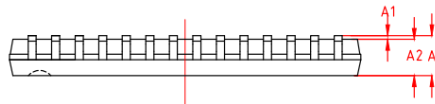
侧视图



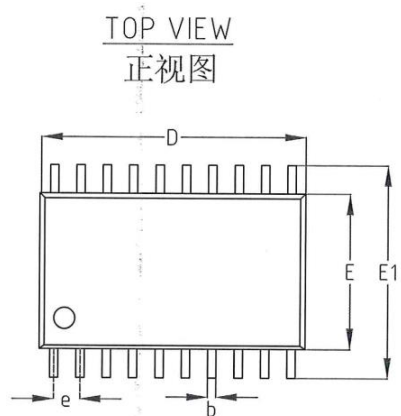
机械尺寸/mm Dimensions			
字符 SYMBOL	最小值 MIN	典型值 NOMINAL	最大值 MAX
A	-	-	1.20
A1	0.05	-	0.15
A2	0.80	0.90	1.00
b	0.19	-	0.30
c	0.14	-	0.18
D	9.60	9.70	9.80
E	4.30	4.40	4.50
E1	6.25	6.40	6.55
e	0.65 BSC		
L1	1.00REF		
L	0.45	0.60	0.75
$\theta$	0°	-	8°

SIDE VIEW

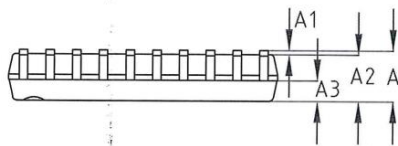
侧视图



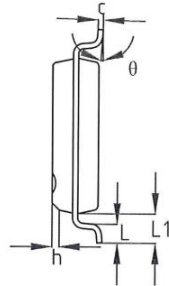
### 29.3 SOP20L 封装示意图



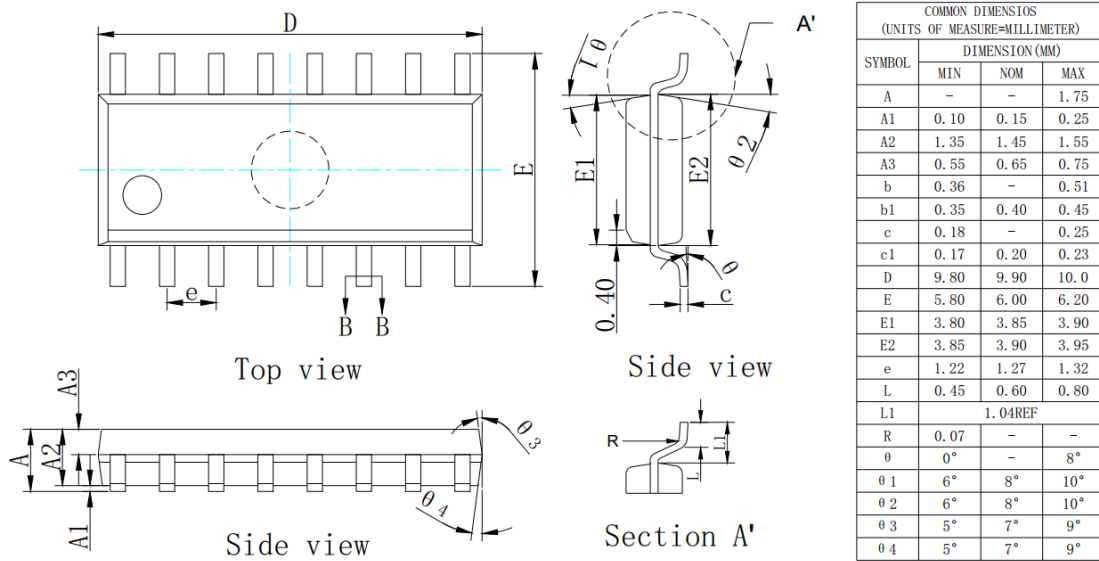
SIDE VIEW  
侧视图

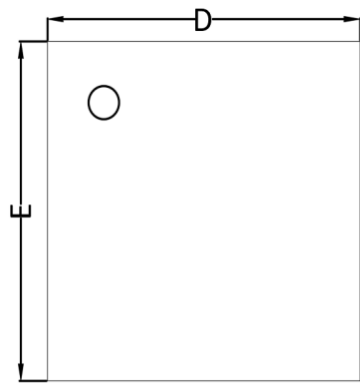
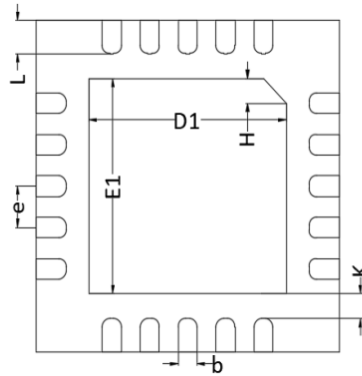
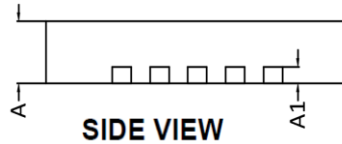


SIDE VIEW  
侧视图



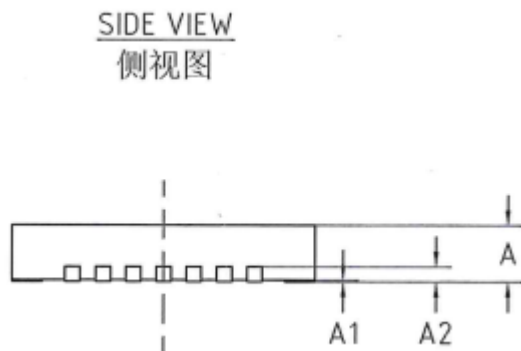
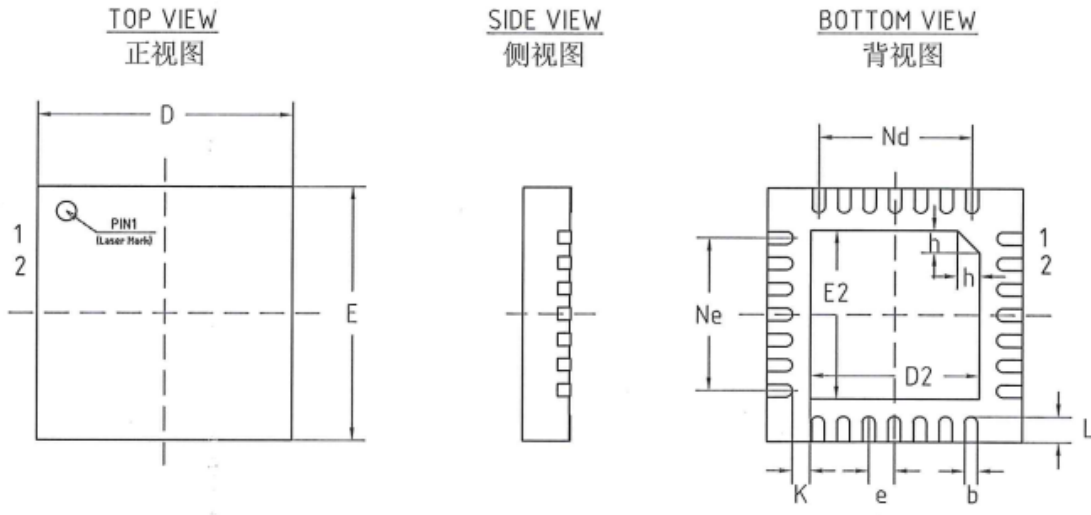
机械尺寸/mm Dimensions			
字符 SYMBOL	最小值 MIN	典型值 NOMINAL	最大值 MAX
A	-	-	2.65
A1	0.10	-	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	-	0.47
c	0.25	-	0.29
D	12.70	12.80	12.90
E	7.40	7.50	7.60
E1	10.10	10.30	10.50
e	1.27 BSC		
L1	1.40REF		
h	0.25	-	0.75
L	0.70	-	1.00
θ	0°	-	8°

**29.4 SOP16L 封装示意图**


**29.5 QFN4\*4-20L 封装示意图**

**TOP VIEW**

**BOTTOM VIEW**

**SIDE VIEW**

 COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0.203 REF		
b	0.20	0.25	0.30
D	3.90	4.00	4.10
E	3.90	4.00	4.10
D1	2.50	2.60	2.70
E1	2.50	2.60	2.70
e	0.40	0.50	0.60
H	0.30 REF		
K	0.20	0.30	0.40
L	0.35	0.40	0.45

**29.6 QFN4\*4-28L 封装示意图**


机械尺寸/mm			
字符 SYMBOL	最小值 MIN	典型值 NOMINAL	最大值 MAX
A	0.70	0.75	0.80
A1	-	0.02	0.05
A2	0.203 REF		
b	0.15	0.20	0.25
D	3.90	4.00	4.10
D2	2.55	2.65	2.75
E	3.90	4.00	4.10
E2	2.55	2.65	2.75
e	0.40 BSC		
K	0.20	0.25	0.30
L	0.35	0.40	0.45
h	0.30	0.35	0.40
Ne	2.40 BSC		
Nd	2.40 BSC		

### 30 修订记录

版本号	修订说明	日期
V0.9	初版编制	2022.11.18
V1.1	正式版 1.1 发布	2023.03.07
V1.12	增加 QFN20 封装、修改错误	2023.05.05
V1.13	增加 SOP28 封装	2023.05.25
V1.15~1.152	增加 SOP16 封装、修改 SOP20A 封装脚位、修改错误	2023.07.08
V1.16	修改 SOP20B 封装脚位，老 SOP20B 版本作废	2023.07.15
V1.17	增加 QFN28 封装	2023.08.28
V1.20	芯片进行改版，主时钟 32MHz 修改为 16MHz，系统时钟分频 8 分频档位改为 1 分频，VIH/VIL 默认值 iovth_s=1 修改为代表 0.5/0.2	2023.08.29
V1.21	修改 QFN20 封装尺寸图、增加 SOP16C 封装、修改文字错误	2023.10.07

拟制	审核	批准
DD	U	



## 31 声明

本手册所述信息仅为您提供说明，该手册不保证没有进一步修改或更新，深圳锐盟半导体有限公司（以下简称本公司）保留随时对本手册及所述的产品和服务进行更改、修改或改进的权利，恕不另行通知。

本公司对所述信息不作任何形式的声明或担保，对因这些信息及使用这些信息而引起的后果不承担任何责任。

除明确规定的定制产品外，包括本手册中描述的本公司的任何产品仅为普通商业、工业、个人和/或家庭应用而设计、开发和制造，禁止用于军事、国防、核能以及可能导致人身伤害、死亡，或是环境破坏等领域。用户应采取任何和所有行动，确保按照适用的法律法规使用和销售产品。

本公司对该手册拥有最终解释权。