

# JZ8FT8801

## 8 位 FLASH 微控制器

### 用户数据手册

版本号 V0.3

## 修改记录说明

版本号	修改说明	备注
V0.0	初版	
V0.1	修订脚位、参数与相关描述	
V0.2	修订脚位图、温频曲线、参数与相关描述	
V0.3	三合一增加注意事项、DAC 数据	

### 注意事项：

#### 1. 三合一相关注意事项：

- (1) 当 JZ8FT8801 在 IIC/SPI 为从机时，从机的指令时钟 (Fcpu) 与系统时钟 (Fosc) 必须要大于或等于主机的指令时钟 (Fcpu) 和系统时钟 (Fosc)，与 IIC/SPI 时钟无关；
- (2) 使用 I2C 从机中断时，当从机收到主机发送的读指令时，需第一时间把首个数据发送出去；如等 TREQ 标志起来以后再发送首个数据，容易造成首个数据丢失，具体参考 DEMO；
- (3) 使用 I2C 从机中断时，在配置 I2C 后，需先将 AA 置一；
- (4) 使用 SPI 时，配置完 SPI 后，CS 线需要优先拉高，避免毛刺干扰通讯；

#### 2. 无论 TC0 或 TC3 开启 RTC 功能都需要使能 RTCEN，但在 RTCEN 使能之后，TC3 只能作为 RTC 使用；



## 目 录

<b>1 芯片简介</b> .....	<b>10</b>
1.1 功能特性 .....	10
1.2 型号说明 .....	12
1.3 引脚分配 .....	13
1.4 引脚描述 .....	16
1.5 系统框图 .....	21
<b>2 中央处理器</b> .....	<b>22</b>
2.1 程序存储区结构 .....	22
2.2 数据存储区 .....	23
2.2.1 数据存储区结构 .....	23
2.2.2 特殊功能寄存器概览 .....	23
2.3 特殊系统寄存器 .....	24
2.3.1 间接寻址 .....	24
2.3.1.1 R180/RSR (RAM 选择低位寄存器) .....	24
2.3.1.2 R184/RSRH (RAM 选择高位寄存器) .....	25
2.3.1.3 R1FD/IAR (间接寻址寄存器) .....	25
2.3.1.4 R1FE/IAR1 (间接寻址寄存器) .....	25
2.3.1.5 R1FF/IAR2 (间接寻址寄存器) .....	25
2.3.2 查表 .....	26
2.3.2.1 跳转表 .....	26
2.3.2.2 ROM 区查表 .....	27
2.3.3 状态寄存器 .....	29
2.3.3.1 R183/STATUS (状态标志寄存器) .....	29
2.3.4 电源域 .....	30
2.3.4.1 R186/TBRDH (电源域控制寄存器) .....	30
<b>3 I/O 端口</b> .....	<b>31</b>
3.1 GPIO 内部结构图 .....	31
3.2 P5 端口 .....	32
3.2.1 R18A/PORT5 (P5 数据寄存器) .....	32
3.2.2 R18D/P5CON (P5 控制寄存器) .....	32
3.2.3 R190/P5PH (P5 上拉控制寄存器) .....	32
3.2.4 R193/P5PD (P5 下拉控制寄存器) .....	33
3.2.5 R19C/P5IWE (P5 输入变化中断、唤醒使能寄存器) .....	33
3.2.6 R1A0/P5AE (P5 模拟口使能寄存器) .....	33
3.3 P6 端口 .....	34
3.3.1 R18B/PORT6 (P6 数据寄存器) .....	34
3.3.2 R18E/P6CON (P6 控制寄存器) .....	34
3.3.3 R191/P6PH (P6 上拉控制寄存器) .....	34
3.3.4 R194/P6PD (P6 下拉控制寄存器) .....	35
3.3.5 R19D/P6IWE (P6 输入变化中断、唤醒使能寄存器) .....	35
3.3.6 R1A1/P6AE (P6 模拟口使能寄存器) .....	35
3.4 P7 端口 .....	36



3.4.1 R18C/PORT7 (P7 数据寄存器)	36
3.4.2 R18F/P7CON (P7 控制寄存器)	36
3.4.3 R192/P7PH (P7 上拉控制寄存器)	36
3.4.4 R195/P7PD (P7 下拉控制寄存器)	37
3.4.5 R19E/P7IWE (P7 输入变化中断、唤醒使能寄存器)	37
3.4.6 R1A2/P7AE (P7 模拟口使能寄存器)	37
3.5 P8 端口	38
3.5.1 R196/PORT8 (P8 数据寄存器)	38
3.5.2 R197/P8CON (P8 控制寄存器)	38
3.5.3 R198/P8PH (P8 上拉控制寄存器)	38
3.5.4 R199/P8PD (P8 下拉控制寄存器)	39
3.5.5 R19A/P8AE (P8 模拟口使能寄存器)	39
3.5.6 R19F/P8IWE (P8 输入变化中断、唤醒使能寄存器)	39
3.5.7 端口输入变化唤醒	40
3.5.8 端口施密特参数	40
<b>4 中断</b>	<b>41</b>
4.1 中断现场保护	42
4.2 中断相关寄存器	42
4.2.1 R1AE/EXINTCON (外部中断控制寄存器)	42
4.2.2 R1D6/INTE0 (中断使能控制寄存器 0)	43
4.2.3 R1D7/INTE1 (中断使能控制寄存器 1)	44
4.2.4 R1D8/INTE2 (中断使能控制寄存器 2)	45
4.2.5 R1DA/INTF0 (中断标志寄存器 0)	46
4.2.6 R1DB/INTF1 (中断标志寄存器 1)	47
4.2.7 R1DC/INTF2 (中断标志寄存器 2)	48
<b>5 复位</b>	<b>49</b>
5.1.1 复位功能概述	49
5.1.2 POR 上电复位	49
5.1.3 WDT 看门狗复位	50
5.1.4 LVR 低电压复位	50
5.1.5 工作频率与 LVR 低压检测关系	51
5.1.6 复位相关寄存器	52
5.1.6.1 R1AF/WDTCON (WDT、唤醒控制寄存器)	52
<b>6 系统时钟与工作模式</b>	<b>53</b>
6.1 系统时钟	53
6.1.1 内部 IHRC 振荡器	53
6.1.2 内部 ILRC 振荡器	53
6.1.3 外部晶体振荡器	53
6.1.4 模块应用说明	54
6.2 工作模式	54
6.2.1 高速模式	55
6.2.2 低速模式	56
6.2.3 空闲模式	56
6.2.4 睡眠模式	57



6.3 相关寄存器 .....	57
6.3.1 R185/WKCON(唤醒控制寄存器) .....	57
6.3.2 R188/CPUCON(CPU 模式控制寄存器) .....	58
6.3.3 R189/IHRCCAL(IHRC 频率微调寄存器) .....	59
6.3.4 R19B/LXTCON(LXT 控制寄存器) .....	59
6.3.5 R1AF/WDTCON(WDT、唤醒控制寄存器) .....	60
<b>7 定时计数器 .....</b>	<b>61</b>
7.1 TC0 定时计数器 .....	61
7.1.1 RTC 模式说明 .....	61
7.1.2 TC0 定时设置说明 .....	62
7.1.3 定时计算说明 .....	62
7.1.4 TC0 唤醒说明 .....	63
7.1.5 TC0 端口电平捕获说明 .....	63
7.1.6 TC0 相关寄存器 .....	64
7.1.6.1 R1AD/TPRE(TC0 预分频器/TC0 捕获低 8 位寄存器) .....	64
7.1.6.2 R1AE/EXINTCON(外部中断控制寄存器) .....	64
7.1.6.3 R1C0/TCOCON(TC0 控制寄存器) .....	65
7.1.6.4 R1C1/TCOC(TC0 计数寄存器) .....	66
7.1.6.5 R1C2/TC3CON(RTC 使能寄存器) .....	66
7.2 TC1 定时计数器 .....	67
7.2.1 TC1 定时设置说明 .....	67
7.2.2 TC1 定时计算说明 .....	67
7.2.3 TC1 空闲模式唤醒说明 .....	68
7.2.4 TC1 相关寄存器 .....	68
7.2.4.1 R1B0/TC1CON(TC1 控制寄存器) .....	68
7.2.4.2 R1B1/TC1PRDL(TC1 低 8 位计数寄存器) .....	69
7.2.4.3 R1B5/TC1PRDTH(TC1 高 4 位计数寄存器) .....	69
7.3 TC2 定时计数器 .....	70
7.3.1 TC2 定时设置说明 .....	70
7.3.2 TC2 定时计算说明 .....	70
7.3.3 TC2 空闲模式唤醒说明 .....	71
7.3.4 TC2 相关寄存器 .....	71
7.3.4.1 R1B8/TC2CON(TC2 控制寄存器) .....	71
7.3.4.2 R1B9/TC2PRDL(TC2 低 8 位计数寄存器) .....	72
7.3.4.3 R1BD/TC2PRDTH(TC2 高 4 位计数寄存器) .....	72
7.4 TC3 定时计数器 .....	73
7.4.1 TC3 定时设置说明 .....	73
7.4.2 TC3 定时计算说明 .....	74
7.4.3 TC3 空闲模式唤醒说明 .....	74
7.4.4 TC3 端口电平捕获 .....	75
7.4.4.1 TC3 捕获模式设置说明 .....	75
7.4.4.2 TC3 捕获模式说明 .....	76
7.4.5 TC3 相关寄存器 .....	79
7.4.5.1 R1C2/TC3CON(TC3 控制寄存器) .....	79



7.4.5.2 R1C3/TC3CL(TC3 低 8 位计数寄存器)	80
7.4.5.3 R1C4/TC3CH(TC3 高 8 位计数寄存器)	80
7.4.5.4 R1C5/TC3RL(TC3 重载低 8 位寄存器)	80
7.4.5.5 R1C6/TC3RH(TC3 重载高 8 位寄存器)	80
7.4.5.6 R1C7/CAPCON(捕获控制寄存器)	80
7.5 TC4 定时计数器	82
7.5.1 TC4 定时设置说明	82
7.5.2 TC4 定时计算说明	83
7.5.3 TC4 空闲模式唤醒说明	83
7.5.4 TC4 相关寄存器	83
7.5.4.1 R1C8/TC4CON(TC4 控制寄存器)	83
7.5.4.2 R1C9/TC4CL(TC4 低 8 位计数寄存器)	84
7.5.4.3 R1CA/TC4CH(TC4 高 8 位计数寄存器)	84
7.6 TC5 定时计数器	85
7.6.1 TC5 定时设置说明	85
7.6.2 TC5 定时计算说明	86
7.6.3 TC5 空闲模式唤醒说明	86
7.6.4 TC5 相关寄存器	86
7.6.4.1 R1CB/TC5CON(TC5 控制寄存器)	86
7.6.4.2 R1CC/TC5CL (TC5 低 8 位计数寄存器)	87
7.6.4.3 R1CD/TC5CH (TC5 高 8 位计数寄存器)	87
<b>8 PWM 脉宽调制</b>	<b>88</b>
8.1 PWM 内部结构与时序	88
8.2 PWM 周期与占空比	89
8.3 PWM 空闲模式唤醒说明	90
8.4 PWM5 比较器门控说明	90
8.5 IPWM 波形以及 Buzzer 设置说明	90
8.6 PWM4DT 中断以及触发 ADC 说明	91
8.7 PWM 刹车中断功能	91
8.8 LED 单线级联	91
8.9 PWM 相关寄存器	92
8.9.1 R186/TBRDH(级联控制寄存器)	92
8.9.2 R1B0/TC1CON(TC1 控制寄存器)	93
8.9.3 R1B1/TC1PRDL(TC1/PWM1234 周期低 8 位寄存器)	93
8.9.4 R1B2/PWM1DTL(PWM1 占空比低 8 位寄存器)	94
8.9.5 R1B3/PWM2DTL(PWM2 占空比低 8 位寄存器)	94
8.9.6 R1B4/PWM3DTL(PWM3 占空比低 8 位寄存器)	94
8.9.7 R1B5/TC1PRDTH(TC1/PWM1234 周期高 4 位及 PWM3 占空比高 4 位寄存器)	94
8.9.8 R1B6/PWM2DTH(PWM2 占空比高 4 位寄存器)	94
8.9.9 R1B7/PWMCON1(PWM 控制寄存器 1)	95
8.9.10 R1B8/TC2CON(TC2 控制寄存器)	96
8.9.11 R1B9/TC2PRDL(TC2/PWM567 周期低 8 位寄存器)	97
8.9.12 R1BA/PWM5DTL(PWM5 占空比低 8 位寄存器)	97
8.9.13 R1BB/PWM6DTL(PWM6 占空比低 8 位寄存器)	97



8.9.14 R1BC/PWM7DTL (PWM7 占空比低 8 位寄存器)	97
8.9.15 R1BD/TC2PRDTH (TC2/PWM567 周期高 4 位及 PWM7 占空比高 4 位寄存器)	97
8.9.16 R1BE/PWM65DTH (PWM65 占空比高 4 位寄存器)	98
8.9.17 R1BF/PWMCON2 (PWM 控制寄存器 2)	98
8.9.18 R1CE/PWM4DTL (PWM4 占空比低 8 位寄存器)	99
8.9.19 R1CF/PWM4CON (PWM4 控制及占空比高位寄存器)	99
8.9.20 R1D0/PWMCON3 (PWM 控制寄存器 3)	100
8.9.21 R1D1/PWMCON4 (PWM 控制寄存器 4)	100
8.9.22 R1D2/PWMCON5 (PWM 控制寄存器 5)	101
8.9.23 R1D3/PWMDEAD (PWM 死区控制寄存器)	102
<b>9 DAC 数模转换</b>	<b>103</b>
9.1 DACVREF 电压说明	103
9.2 DAC 相关寄存器	103
9.2.1 R1A5/DACON (DAC 控制寄存器)	103
9.2.2 R1AC/VREFCAL (VREF 微调寄存器)	103
<b>10 ADC 模数转换</b>	<b>104</b>
10.1 ADC 检测电源电压说明	105
10.2 P51 端口检测 1/4 外部电压说明	105
10.3 ADC 模数转换设置说明	105
10.4 ADC 相关寄存器	106
10.4.1 R1A3/ADATH (ADC 数据高 8 位寄存器)	106
10.4.2 R1A4/ADATL (ADC 数据低 8 位寄存器)	106
10.4.3 R1A5/DACON (DAC 控制寄存器)	106
10.4.4 R1A6/ADCON0 (ADC 控制寄存器 0)	106
10.4.5 R1A7/ADCON1 (ADC 控制寄存器 1)	108
<b>11 CMP 比较器</b>	<b>109</b>
11.1 CMP0 比较器	109
11.1.1 分压电阻输出电压 $V_{internal R}$	110
11.1.2 比较器配置	111
11.1.3 DACVREF 使用方法	111
11.1.4 RFC 电阻频率转换	111
11.1.5 CMP0 相关寄存器	112
11.1.5.1 R185/WKCON (CMP0 门控寄存器)	112
11.1.5.2 R1F9/CMPOCON0 (CMP0 控制寄存器 0)	112
11.1.5.3 R1FA/CMPOCON1 (CMP0 控制寄存器 1)	114
11.1.5.4 R1FC/CMP1CON1 (CMP0 门控控制寄存器)	114
11.2 CMP1 比较器	115
11.2.1 CMP1 相关寄存器	115
11.2.1.1 R1FB/CMP1CON0 (CMP1 控制寄存器 0)	115
11.2.1.2 R1FC/CMP1CON1 (CMP1 控制寄存器 1)	116
11.3 CMP2 比较器	116
<b>12 OPA 运算放大器</b>	<b>117</b>
12.1 OPA 运算放大器概述	117
12.2 OPA 运算放大器工作模式	117

12.3 OPA 相关寄存器 .....	120
12.3.1 R1A8/OPACON0(OPA 控制寄存器 0) .....	120
12.3.2 R1A9/OPACON1(OPA 控制寄存器 1) .....	121
12.3.3 R1AA/OPACON2(OPA 控制寄存器 2) .....	121
12.3.4 R1AB/OPACON3(OPA 控制寄存器 3) .....	122
<b>13 LED 点阵驱动 .....</b>	<b>123</b>
13.1 LED 功能描述 .....	123
13.2 LED 点阵驱动设置说明 .....	129
13.3 LED 点阵相关寄存器 .....	129
13.3.1 R1D4/LEDCON(LED 控制寄存器) .....	129
13.3.2 R1D5/LEDDT(LED 占空比寄存器) .....	130
<b>14 LCD 驱动 .....</b>	<b>131</b>
14.1 LCD 功能描述 .....	131
14.2 LCD 相关寄存器 .....	132
14.2.1 R1F0/LCDCON0(LCD 控制寄存器 0) .....	132
14.2.2 R1F1/LCDCON1(LCD 控制寄存器 1) .....	134
14.2.3 R1F2/LCDCON2(LCD 控制寄存器 2) .....	134
14.2.4 R1F3/LCDCON3(LCD 控制寄存器 3) .....	135
14.2.5 R1F4/LCDCON4(LCD 控制寄存器 4) .....	135
14.2.6 R1F5/SEGCON0(SEG 控制寄存器 0) .....	136
14.2.7 R1F6/SEGCON1(SEG 控制寄存器 1) .....	136
14.2.8 R1F7/SEGCON2(SEG 控制寄存器 2) .....	136
14.2.9 R1F8/LCDDDB(LCD 数据寄存器) .....	137
<b>15 UART 串行接收器/发送器 .....</b>	<b>138</b>
15.1 UART 发送 .....	138
15.2 UART 接收 .....	139
15.3 UART 相关寄存器 .....	140
15.3.1 R1DD/SCON0(UART0 串口控制寄存器) .....	140
15.3.2 R1DE/SBUF(UART0 收发数据寄存器) .....	140
15.3.3 R1DF/SCON1(UART0 串口控制寄存器 1) .....	141
<b>16 UART/IIC/SPI 三选一串行接口 .....</b>	<b>142</b>
16.1 UART 串行接收器/发送器 .....	142
16.1.1 UART 发送 .....	142
16.1.2 UART 接收 .....	143
16.2 IIC 串行接收器/发送器 .....	144
16.2.1 IIC 操作流程 .....	144
16.2.2 IIC 发送与接收 .....	145
16.3 SPI 串行接收器/发送器 .....	146
16.3.1 SPI 工作模式 .....	146
16.3.2 SPI 发送与接收 .....	147
16.4 三合一相关寄存器 .....	148
16.4.1 R1E3/SSICON0(SS1 控制寄存器 0) .....	148
16.4.2 R1E4/SSICON1(SS1 控制寄存器 1) .....	148
16.4.3 R1E5/I2CADDR(I2C 地址寄存器) .....	150



16.4.4	R1E6/SSIBUF (SSI 收发数据寄存器)	150
16.4.5	R1E7/SSISR0 (串口状态寄存器 0)	150
16.4.6	R1E8/SSISR1 (串口状态寄存器 1)	151
<b>17</b>	<b>EEPROM 以及 ISP 在线烧录操作</b>	<b>152</b>
17.1	EEPROM 读写操作	152
17.2	ISP 烧录	152
17.3	EEPROM 与 ISP 相关寄存器	153
17.3.1	R187/TBRDL (EEPROM 地址寄存器)	153
17.3.2	R1E0/E2PCON (EEPROM 控制寄存器)	153
17.3.3	R1E1/E2PDATL (EEPROM 数据低 8 位寄存器)	154
17.3.4	R1E2/E2PDATH (EEPROM 数据高 8 位寄存器)	154
<b>18</b>	<b>CDC 触摸按键检测</b>	<b>155</b>
18.1	触摸检测概述	155
18.2	CDC 结构与时序	155
18.3	CDC 触摸检测设置说明	157
18.3.1	R1E9/TKCON (TK 控制寄存器)	158
18.3.2	R1EA/TKCHS0 (触摸通道使能控制寄存器 0)	159
18.3.3	R1EB/TKCHS1 (触摸通道使能控制寄存器 1)	159
18.3.4	R1EC/TKVS (TK 电压选择寄存器)	159
18.3.5	R1ED/TKCNTH (TK 计数器高位寄存器)	160
18.3.6	R1EE/TKCNTL (TK 计数器低位寄存器)	160
<b>19</b>	<b>OPTION 配置表</b>	<b>161</b>
<b>20</b>	<b>指令集</b>	<b>163</b>
<b>21</b>	<b>电气特性</b>	<b>165</b>
21.1	极限参数	165
21.2	直流电气特性	165
21.3	AD 转换特性	167
21.4	比较器特性	167
21.5	特性曲线图	168
21.5.1	内部低速振荡器-压频特性曲线	168
21.5.2	内部低速振荡器-温频特性曲线	168
21.5.3	内部 16Mhz RC 振荡器-压频特性曲线	169
21.5.4	内部 16Mhz RC 振荡器-温频特性曲线	169
21.6	IHRC 频率微调参数说明	170
21.7	DAC 电压参数	172
<b>22</b>	<b>封装尺寸</b>	<b>174</b>
22.1	32PIN 封装尺寸	174
22.2	28PIN 封装尺寸	176
22.3	24PIN 封装尺寸	177
22.4	20PIN 封装尺寸	178
22.5	16PIN 封装尺寸	179

## 1 芯片简介

### 1.1 功能特性

#### CPU 配置

- 8 位 RISC 内核，支持 53 条精简指令集
- 8K×16-Bit MTP ROM，大于 1 万次擦写
- 256×16-Bit EEPROM，掉电保存数据，大于 1 万次重复擦写
- 384×8-Bit SRAM，支持短暂掉电到 0.4V RAM 数据不擦除
- 12 级堆栈空间
- 8 级可编程电压复位（LVR）
  - 1.8V, 2.0V, 2.2V, 2.6V, 3.0V, 3.4V, 4.0V, 4.6V
- 工作电流小于 2.2 mA（8MHz/5V）
- 工作电流小于 10  $\mu$ A（40KHz/5V）
- 睡眠电流小于 1  $\mu$ A（睡眠模式）

#### 工作电压

- $V_{LVR}$  3.4V~5.5V |  $F_{cpu}=0\sim 8MHz$
- $V_{LVR}$  2.6V~5.5V |  $F_{cpu}=0\sim 4MHz$
- $V_{LVR}$  2.0V~5.5V |  $F_{cpu}=0\sim 2MHz$

#### I/O 配置

- 4 组 30 个双向 I/O 端口：P5, P6, P7, P8，所有端口支持推挽、上拉、下拉、输入高阻
- 所有端口均支持睡眠唤醒和空闲唤醒
- 推挽输出时所有 LED 口支持 3 级推电流可选：8mA、16mA、28mA  
支持 2 级灌电流可选：30mA、60mA
- 推挽输出时 P60~P63 口支持 4 级驱动电流可选：25mA、50mA、80mA、100mA
- 外部中断：EXINT0:P50/P54；EXINT1:P60/P65

#### 工作频率范围

- 内部 IHRC 振荡电路：32MHz/16MHz/8MHz/4MHz，修调精度  $\pm 3\%$
- 内部 ILRC 振荡电路：40KHz
- 外部晶体振荡电路：LXT 低速晶振：32.768KHz；HXT 高速晶振：400K~16MHz
- 时钟周期分频选择：2T, 4T, 8T, 16T, 32T

#### 外围模块

- 12Bit ADC 模数转换器
  - 30 路外部通道：AIN0~AIN29；  
7 路内部通道：0.25VDD、0.25P51、DAC、OPA、TEMP、VREF、GND
  - 参考电压：VDD、内部参考电压 VREF (VBG/2V/3V/4V)、外部输入 EXVREF (P61)
  - ADC 时钟：Fosc 的 1/4/16/64 分频



- 支持采样通道内置电容滤波
- DAC 数模转换
  - 1 路 DAC 电压可输出可比较可 AD 采样
  - 输出电压范围 VBG (1.2V) ~4.9V
- CDC 触摸
  - 最多支持 12 路触摸通道
  - 支持 4 级 LDO
  - 支持跳频
  - 外挂电容范围：推荐 1nF~10nF
- 定时器
  - 8 Bit 定时计数器 TC0
  - 12 Bit 定时计数器 TC1
  - 12 Bit 定时计数器 TC2
  - 16Bit 定时计数器 TC3
  - 16Bit 定时计数器 TC4
  - 16Bit 定时计数器 TC5
- PWM
  - 4 路 1 组 12Bit-PWM1/2/3/4 (PWM1/2/3 支持互补死区，支持取反)
  - 3 路 1 组 12Bit-PWM5/6/7 (两对互补，支持取反)
  - PWM 可触发 AD 采样
  - PWM 可复用为 2 路 RGB LED 级联控制器（驱动幻彩灯）
  - PWM 可控制刹车，电平可调
- 通信模块
  - 1 路 SSI 三合一模块 (支持 UART/IIC/SPI 三种模式)
  - 1 路单独的 UART 串口通信
- 比较器
  - 3 路比较器 CMP0、CMP1、CMP2
  - 比较器可测试 VDD、TEMP、DAC、OPA、虚拟中心点等
  - CMP2 复用 OPA
- OPA
  - 1 路运算放大器 OPA
  - 可输出可比较可 ADC 检测
  - 支持 4 种模式
- LCD 驱动：
  - 4com\*24seg 或 5com\*23seg
  - 1/2Bias 和 1/3Bias 均支持
  - 不用的 COM/SEG 口可以用作其它功能
- LED 点阵
  - 支持最大驱动 9\*10
  - 支持 LED 引脚自由组合，不用的引脚可以用做其它功能
  - 与 LCD 模块复用

## 中断源

TC0~TC5 溢出中断、EXINT0/1, P5/P6/P7/P8 端口输入变化中断、ADC 转换完成中断、PWM4 占空比匹配中断、CMP0 状态变化中断、OPA/CMP2 状态变化中断、UART 串口接收/发送中断、OSC 晶振异常中断、TK 触摸完成中断、端口捕获中断、UART 接收/SPI 中断、UART 发送/IIC 中断, 共 24 个中断源

## 集成开发环境

- 支持 TYPE-C 口升级, 支持带电烧录
- 片上调试 (OCD), ISP
- 4 个硬件断点
- 软件复位, 暂停, 单步, 运行等

## 封装类型

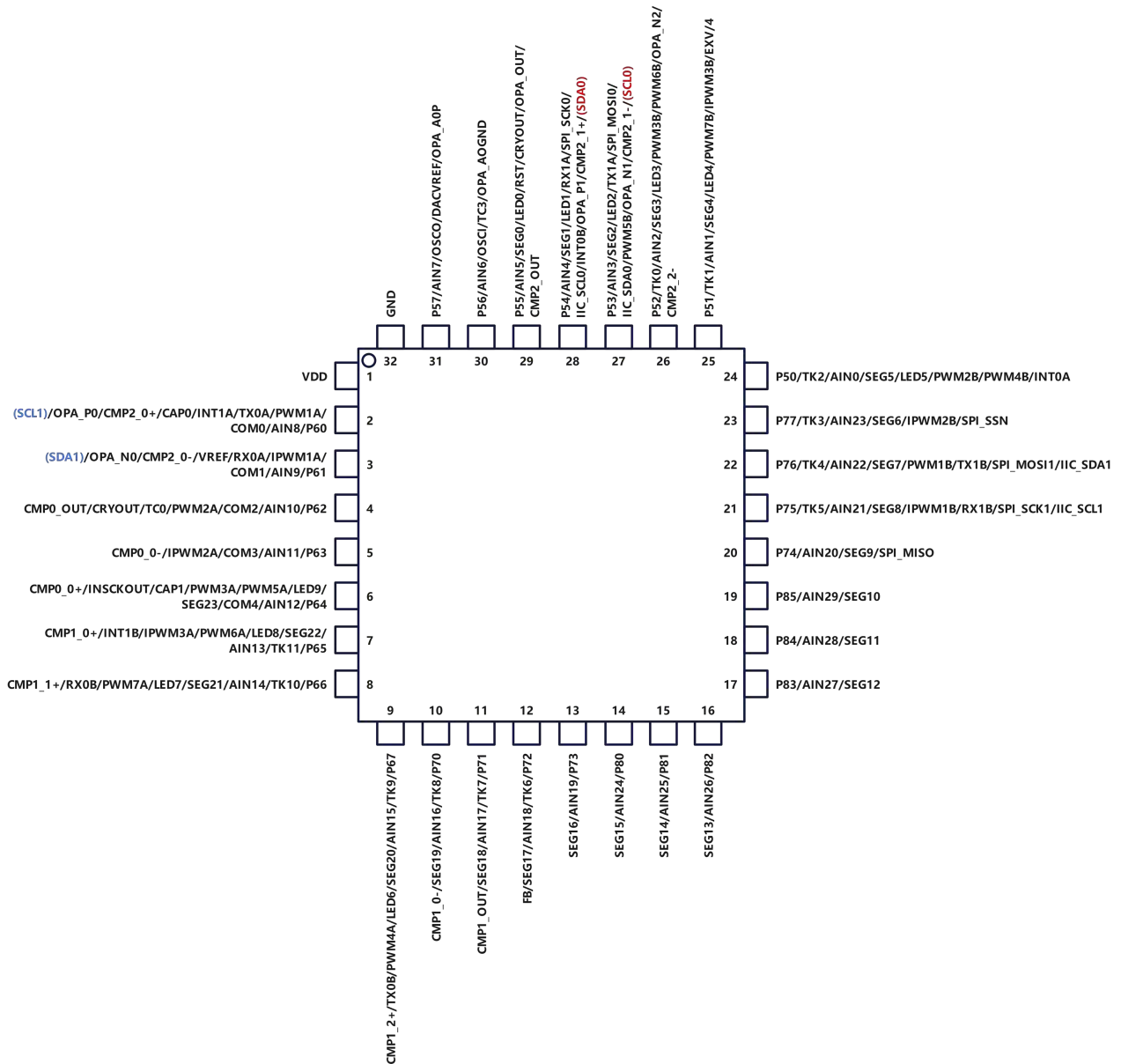
- JZ8FT8801-LQFP32/QFN32
- JZ8FT8801-SSOP28
- JZ8FT8801-SSOP24
- JZ8FT8801-TSSOP20
- JZ8FT8801-SOP16

## 1.2 型号说明

PRODUCT	ROM	RAM	EEPROM	I/O	ADC	LCD	DAC	点阵	PACKAGE
JZ8FT8801-L32	8K*16	384*8	256*16	30	30	4C*23S/5C*22S	Y	9*10	LQFP32
JZ8FT8801-Q32	8K*16	384*8	256*16	30	30	4C*23S/5C*22S	Y	9*10	QFN32
JZ8FT8801-P28	8K*16	384*8	256*16	26	26	4C*20S/5C*19S	Y	9*10	SSOP28
JZ8FT8801-P24	8K*16	384*8	256*16	22	22	4C*16S/5C*15S	Y	9*10	SSOP24
JZ8FT8801-T20	8K*16	384*8	256*16	18	18	4C*12S/5C*11S	Y	9*10	TSSOP20
JZ8FT8801-S16	8K*16	384*8	256*16	14	14	4C*8S/5C*7S	Y	7*8	SOP16

## 1.3 引脚分配

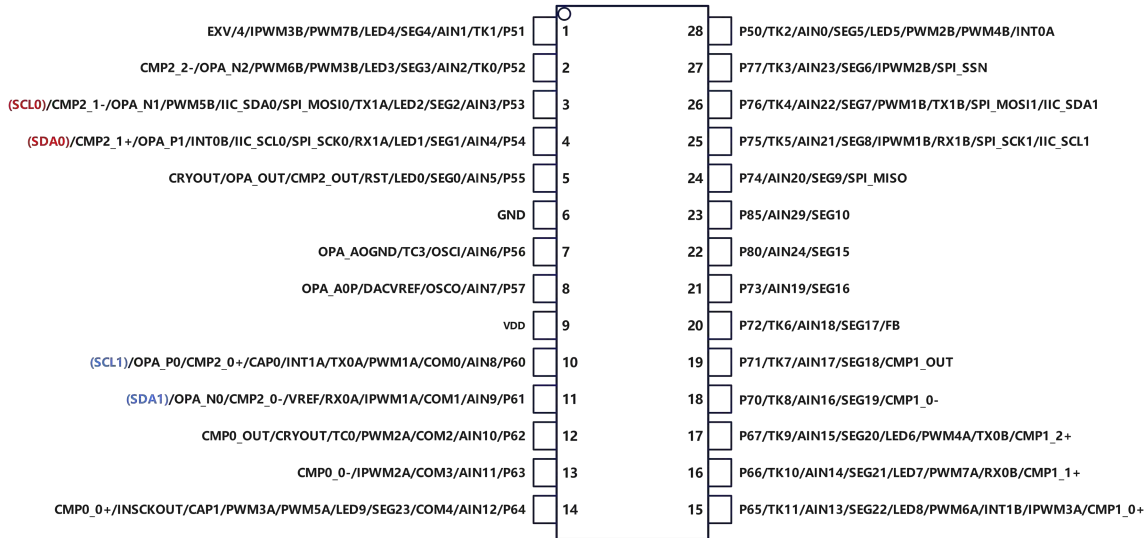
型号: JZ8FT8801-L32/JZ8FT8801-Q32



32PIN 封装引脚图

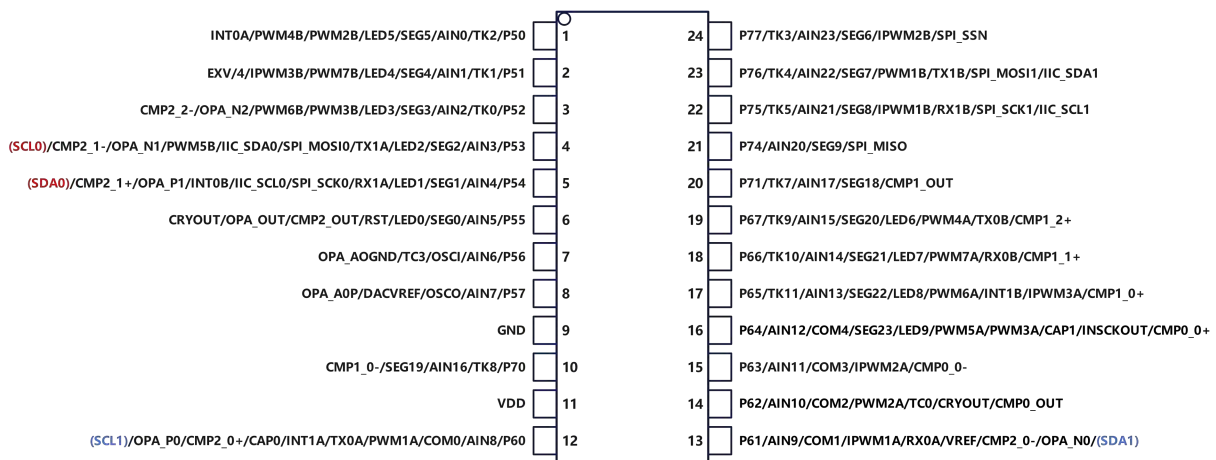
JZ8FT8801 仿真下载和烧录口: P53 (SCL0)、P54 (SDA0) 或者 P60 (SCL1)、P61 (SDA1)

型号: JZ8FT8801-P28



28PIN 封装引脚图

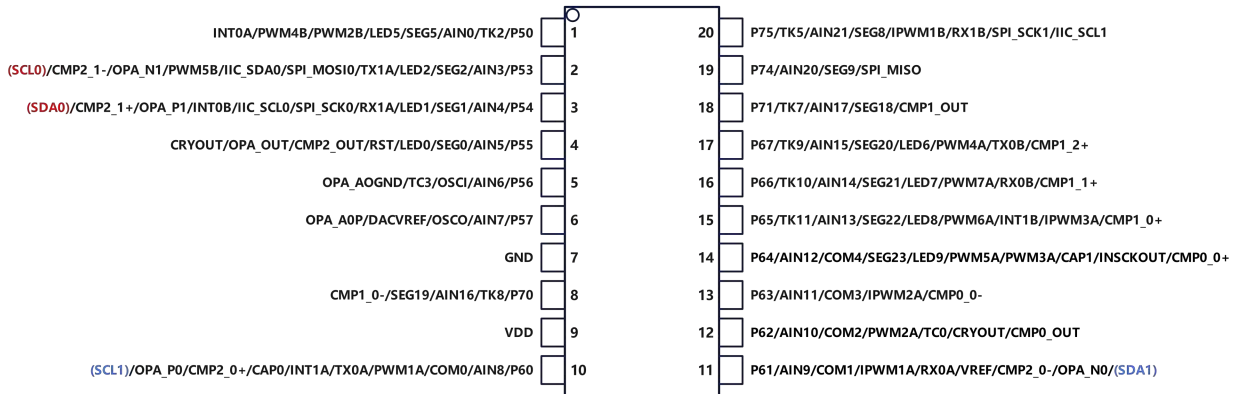
型号: JZ8FT8801-P24



24PIN 封装引脚图

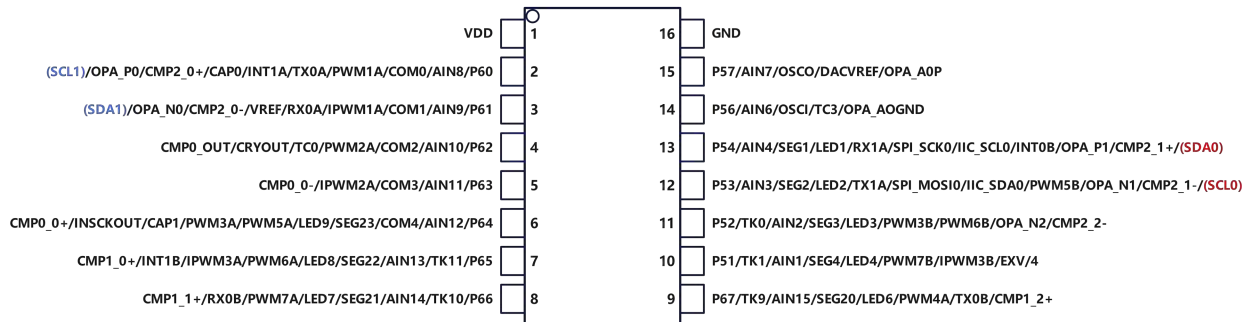
JZ8FT8801 仿真下载和烧录口: P53 (SCL0)、P54 (SDA0) 或者 P60 (SCL1)、P61 (SDA1)

型号: JZ8FT8801-T20



20PIN 封装引脚图

型号: JZ8FT8801-S16



16PIN 封装引脚图

JZ8FT8801 仿真下载和烧录口: P53 (SCL0)、P54 (SDA0) 或者 P60 (SCL1)、P61 (SDA1)



## 1.4 引脚描述

序号	管脚名	I/O	功能描述
P50	P50	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	TK2	I	TK 触摸通道
	AIN0	AN	模拟输入通道
	INT0A	I	外部中断 INT0 输入口
	PWM2B	O	PWM2B 输出
	PWM4B	O	PWM4B 输出
	SEG5	O	LCD 驱动口 5
	LED5	O	LED 驱动口 5
P51	P51	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN1	AN	模拟输入通道
	TK1	I	TK 触摸通道
	IPWM3B	O	IPWM3B 输出
	PWM7B	O	PWM7B 输出
	EXV/4	AN	外部 1/4 ADC 采样
	SEG4	O	LCD 驱动口 4
	LED4	O	LED 驱动口 4
P52	P52	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN2	AN	模拟输入通道
	TK0	I	TK 触摸通道
	PWM3B	O	PWM3B 输出
	PWM6B	O	PWM6B 输出
	OPA_N2/CMP2_2-	AN	OPA(比较器 2) 负极输入口
	SEG3	O	LCD 驱动口 3
	LED3	O	LED 驱动口 3
P53	P53	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN3	AN	模拟输入通道
	PWM5B	O	PWM5B 输出
	SEG2	O	LCD 驱动口 2
	LED2	O	LED 驱动口 2
	OPA_N1/CMP2_1-	AN	OPA(比较器 2) 负极输入口
	TX1A	O	三合一 UART 发送口 0
	SPI_MOSIO	I/O	SPI 主机输出, 从机输入数据口 0
	IIC_SDA0	O	IIC 数据口 0
	SCL0	I	烧录时序口
P54	P54	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN4	AN	模拟输入通道
	INT0B	I	外部中断 INT0 输入口



	SEG1	0	LCD 驱动口 1
	LED1	0	LED 驱动口 1
	OPA_P1/CMP2_1+	AN	OPA(比较器 2) 正极输入口
	RX1A	I	三合一 UART 接收口 0
	SPI_SCK0	0	SPI 主机模式时钟口 0
	IIC_SCL0	0	IIC 时钟口 0
	SDA0	I	烧录时序口
P55	P55	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN5	AN	模拟输入通道
	SEG0	0	LCD 驱动口 0
	LED0	0	LED 驱动口 0
	CRYOUT	0	晶振时钟输出口
	OPA_OUT/CMP2_OUT	0	OPA(比较器 2) 结果输出口
	RST	I	外部复位输入端口
P56	P56	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN6	AN	模拟输入通道
	TC3	I	TC3 外部时钟输入口
	OSCI	I	晶振时钟口
	OPA_AOGND	AN	差分运放端口
P57	P57	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN7	AN	模拟输入通道
	OSCO	I	晶振时钟口
	OPA_AOP	AN	差分运放端口
	DACVREF	0	DAC 参考电压
P60	P60	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强二级可
	AIN8	AN	模拟输入通道
	PWM1A	0	PWM1A 输出
	INT1A	I	外部中断 1 输入口
	CAPO	I	TC3 捕获口 0
	COM0	0	LCDCOM 驱动口 0
	OPA_PO/CMP2_0+	AN	OPA(比较器 2) 正极输入口
	TX0A	0	UART 发送口 0
	SCL1	I	烧录时序口
P61	P61	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强二级可
	AIN9	AN	模拟输入通道
	IPWM1A	0	IPWM1A 输出
	COM1	0	LCDCOM 驱动口 1
	VREF	AN	ADC 外部基准电压输入口
	RX0A	I	UART 通信接收口 0
	OPA_NO/CMP2_0-	AN	OPA(比较器 2) 负极输入口



	SDA1	I	烧录时序口
P62	P62	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强二级可
	AIN10	AN	模拟输入通道
	PWM2A	0	PWM2A 输出
	COM2	0	LCDCOM 驱动口 2
	CRYOUT	0	晶振时钟输出口
	CMPO_OUT	0	比较器 0 结果输出口
	TC0	I	TC0 外部时钟输入口
P63	P63	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强二级可
	AIN11	AN	模拟输入通道
	IPWM2A	0	IPWM2A 输出
	COM3	0	LCDCOM 驱动口 3
	CMPO_0-	AN	比较器 0 负极输入口
P64	P64	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN12	AN	模拟输入通道
	COM4	0	LCDCOM 驱动口 4
	SEG23	0	LCD 驱动口 23
	CAP1	I	TC3 捕获口 1
	INSCKOUT	0	指令时钟输出口
	CMPO_0+	AN	比较器 0 正极输入口
	PWM3A	0	PWM3A 输出
	PWM5A	0	PWM5A 输出
	LED9	0	LED 驱动口 9
P65	P65	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN13	AN	模拟输入通道
	PWM6A	0	PWM6A 输出
	IPWM3A	0	IPWM3A 输出
	TK11	I	TK 触摸口
	SEG22	0	LCD 驱动口 22
	INT1B	I	外部中断 1 输入口
	CMP1_0+	AN	比较器正极输入口
	LED8	0	LED0 驱动口 8
P66	P66	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN14	AN	模拟输入通道
	PWM7A	0	PWM7A 输出
	SEG21	0	LCD 驱动口 21
	TK10	I	TK 触摸口
	RX0B	I	UART 通信接收口
	CMP1_1+	AN	比较器 1 正极输入口
	LED7	0	LED0 驱动口 7

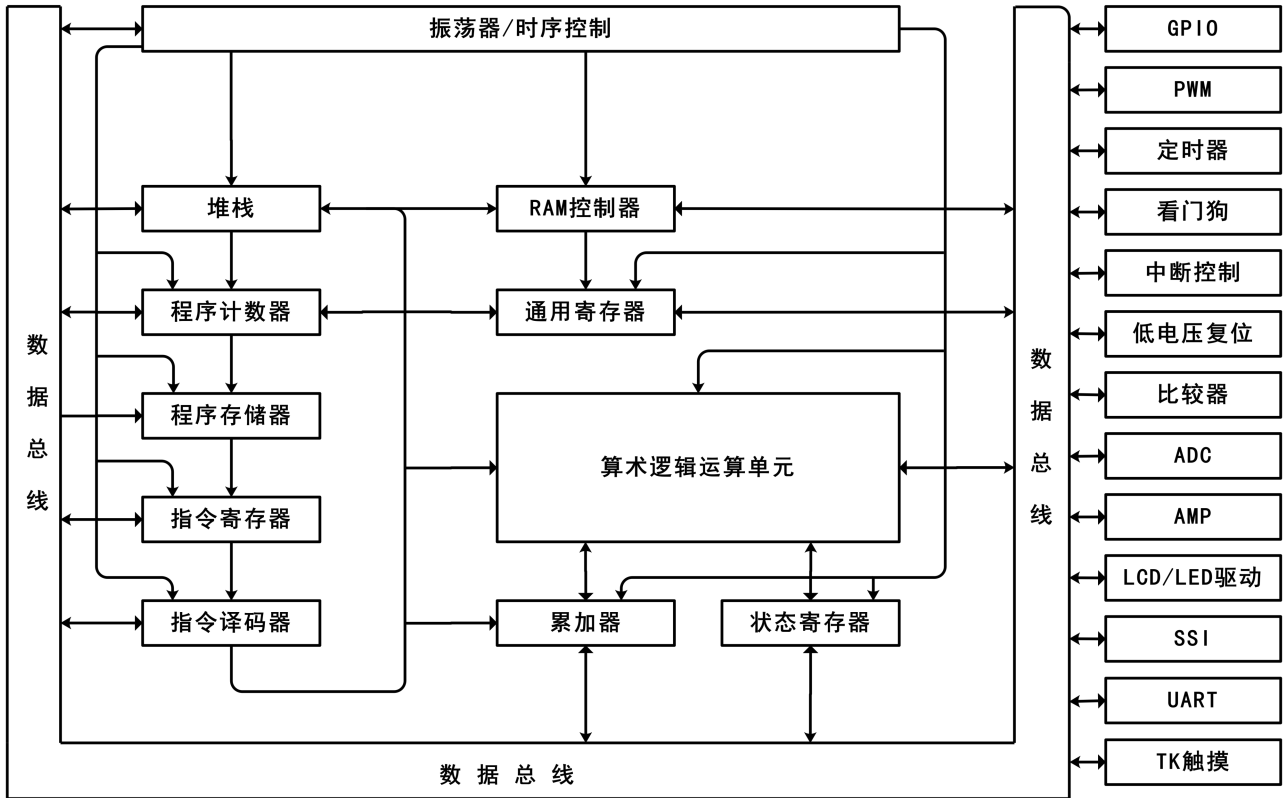


P67	P67	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN15	AN	模拟输入通道
	PWM4A	0	PWM4A 输出
	SEG20	0	LCD 驱动口 20
	TK9	I	TK 触摸口
	TX0B	0	UART 发送口
	CMP1_2+	AN	比较器 1 正极输入口
P70	LED6	0	LEDO 驱动口 6
	P70	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN16	AN	模拟输入通道
	SEG19	0	LCD 驱动口 19
P71	TK8	I	TK 触摸口
	CMP1_0-	AN	比较器 1 负极输入口
	P71	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN17	AN	模拟输入通道
	SEG18	0	LCD 驱动口 18
P72	TK7	I	TK 触摸口
	CMP1_OUT	0	比较器 1 结果输出口
	P72	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN18	AN	模拟输入通道
	SEG17	0	LCD 驱动口 17
P73	TK6	I	TK 触摸口
	FB	I	外部刹车输入口
	P73	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
P74	AIN19	AN	模拟输入通道
	SEG16	0	LCD 驱动口 16
	P74	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN20	AN	模拟输入通道
P75	SPI_MISO	I/O	SPI 主机输入, 从机发送数据口
	SEG9	0	LCD 驱动口 19
	P75	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN21	AN	模拟输入通道
	SEG8	0	LCD 驱动口 8
	TK5	I	TK 触摸口
	IPWM1B	0	IPWM1B 输出
	RX1B	I	三合一 UART 接受口 1
SPI_SCK1	0	SPI 主机模式时钟口 1	
P76	IIC_SCL1	0	IIC 时钟口 1
	P76	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN22	AN	模拟输入通道
P76	SEG7	0	LCD 驱动口 7



	TK4	I	TK 触摸口
	PWM1B	0	PWM1B 输出
	TX1B	0	三合一 UART 发送口 1
	SPI_MOSI1	I/O	SPI 主机输出, 从机输入数据口 1
	IIC_SDA1	0	IIC 数据口 1
P77	P77	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN23	AN	模拟输入通道
	SEG6	0	LCD 驱动口 6
	TK3	I	TK 触摸口
	IPWM2B	0	IPWM2B 输出
	SPI_SSN	I	SPI 从机片选控制口
P80	P80	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN24	AN	模拟输入通道
	SEG15	0	LCD 驱动口 15
P81	P81	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN25	AN	模拟输入通道
	SEG14	0	LCD 驱动口 14
P82	P82	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN26	AN	模拟输入通道
	SEG13	0	LCD 驱动口 13
P83	P83	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN27	AN	模拟输入通道
	SEG12	0	LCD 驱动口 12
P84	P84	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN28	AN	模拟输入通道
	SEG11	0	LCD 驱动口 11
P85	P85	I/O	GPIO, 可编程上下拉, 端口唤醒, 驱动增强
	AIN29	AN	模拟输入通道
	SEG10	0	LCD 驱动口 10
	VDD	--	电源
	GND	--	地

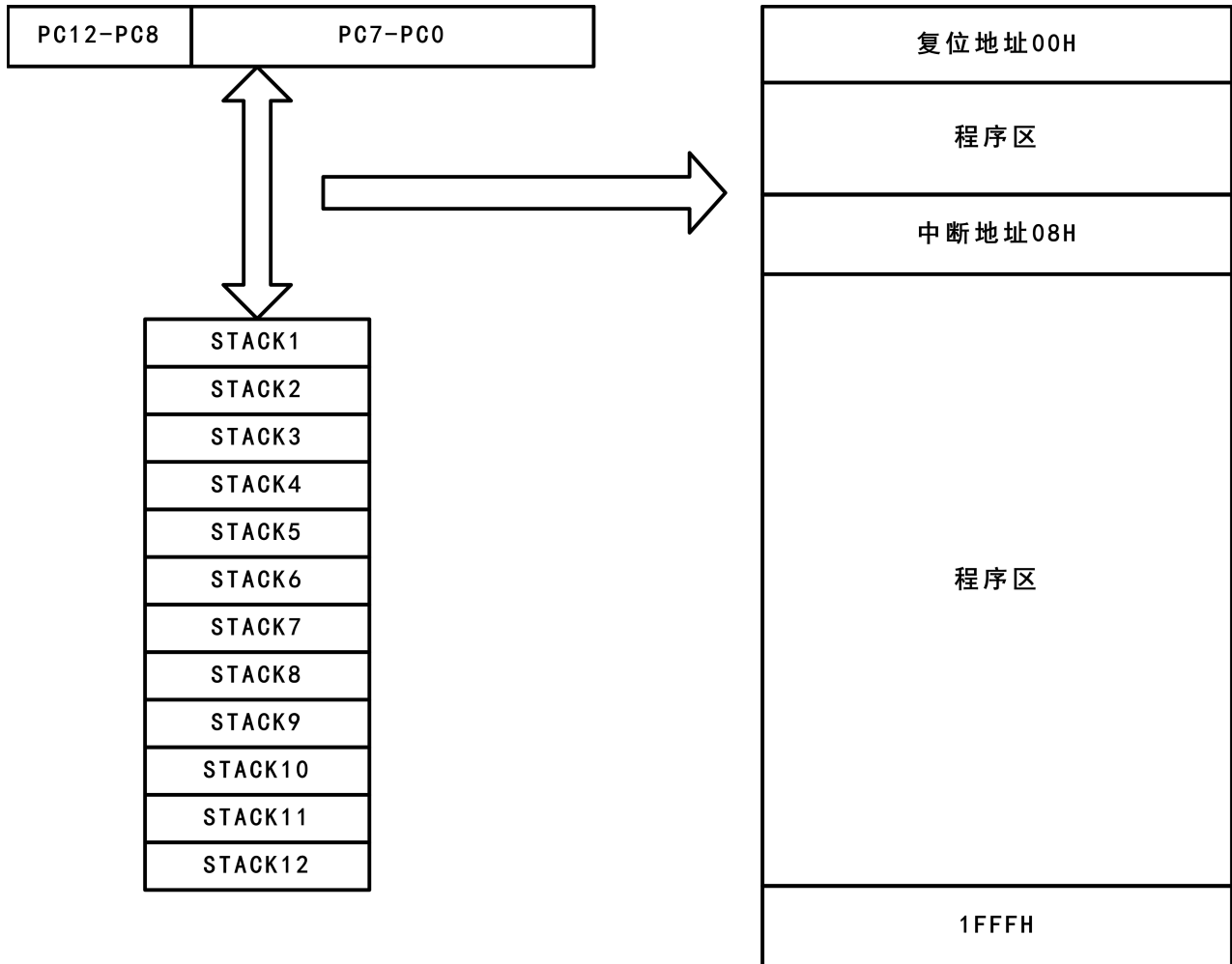
## 1.5 系统框图



系统电路框图

## 2 中央处理器

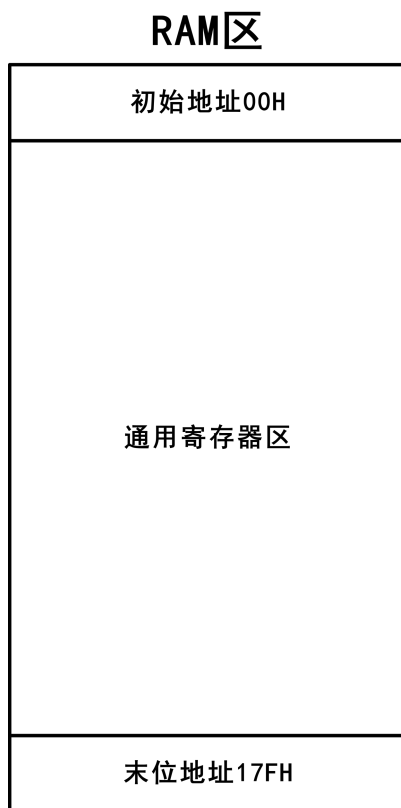
### 2.1 程序存储区结构



程序存储区结构图

## 2.2 数据存储区

### 2.2.1 数据存储区结构



数据存储区

### 2.2.2 特殊功能寄存器概览

地址	0X18_	0X19_	0X1A_	0X1B_	0X1C_	0X1D_	0X1E_	0X1F_
0	RSR	P5PH	P5AE	TC1CON	TC0CON	PWMCON3	E2PCON	LCDCON0
1	PCH	P6PH	P6AE	TC1PRDL	TC0C	PWMCON4	E2PDATL	LCDCON1
2	PCL	P7PH	P7AE	PWM1DTL	TC3CON	PMWCON5	E2PDATH	LCDCON2
3	STATUS	P5PD	ADATH	PWM2DTL	TC3CL	PWMDEAD	SSICON0	LCDCON3
4	RSRH	P6PD	ADATL	PWM3DTL	TC3CH	LEDCON	SSICON1	LCDCON4
5	WKCON	P7PD	DACON	TC1PRDTH	TC3RL	LEDDT	I2CADDR	SEGCON0
6	TBRDH	P8	ADCON0	PWM21DTH	TC3RH	INTE0	SSIBUF	SEGCON1
7	TBRDL	P8CON	ADCON1	PWMCON1	CAPCON	INTE1	SSISR0	SEGCON2
8	CPUCON	P8PH	OPACON0	TC2CON	TC4CON	INTE2	SSISR1	LCDDB
9	IHRCCAL	P8PD	OPACON1	TC2PRDL	TC4CL		TKCON	CMPOCON0
A (10)	PORT5	P8AE	OPACON2	PWM5DTL	TC4CH	INTF0	TKCHS0	CMPOCON1
B (11)	PORT6	LXTCON	OPACON3	PWM6DTL	TC5CON	INTF1	TKCHS1	CMP1CON0
C (12)	PORT7	P5IWE	VREFCAL	PWM7DTL	TC5CL	INTF2	TKVS	CMP1CON1
D (13)	P5CON	P6IWE	TPRE	TC2PRDTH	TC5CH	SCON0	TKCNTH	IAR
E (14)	P6CON	P7IWE	EXINTCON	PWM65DTH	PWM4DTL	SBUF	TKCNTL	IAR1
F (15)	P7CON	P8IWE	WDTCON	PWMCON2	PWM4CON	SCON1		IAR2

## 2.3 特殊系统寄存器

### 2.3.1 间接寻址

数据存储器能被直接或间接寻址，通过 IAR 寄存器可以实现间接寻址操作。

1、RSRH (RSR<8>)、RSR<7:0>寄存器配合 R1FD/IAR 寄存器实现间接寻址操作（寻址范围：0x000 - 0x17F）。用户可以将某个 SRAM 对应的地址放进 RSR 寄存器，然后通过访问间接寻址寄存器 R1FD，此时地址将指向地址为 RSR<8:0>的 SRAM。

2、RSRH (RSR<15:8>)寄存器配合 R1FE/IAR1 寄存器实现间接寻址操作（寻址范围：0x100 - 0x17F）。用户可以将寻址范围内的 SRAM 地址低八位放进 RSRH 寄存器，然后通过访问间接寻址寄存器 R1FE，此时指向 SRAM 的地址为：Bit8 位默认为 1 + RSRH 中对应八位地址。

3、RSR<7:0>寄存器配合 R1FF/IAR2 寄存器实现间接寻址操作（寻址范围：0x00 - 0xFF）。用户可以将寻址范围内的 SRAM 地址低八位放进 RSR 寄存器，然后通过访问间接寻址寄存器 R1FF，此时指向 SRAM 的地址为：Bit8 位默认为 0 + RSR 中对应八位地址。

例：间接寻址清 RAM (0x00~0xff)

```

MOV     A, @0X00
MOV     RSR, A      //间接寻址指针指向 00H
CLR     IAR         //清零 00H 地址所指向的数据
INC     RSR        //地址+1
MOV     A, RSR
XOR     A, @0XFF
JBTS   STATUS, 2
JMP     $-5        //回退 5 条指令，一直清零至 RSR 地址指向 FFH
CLR     0XFF      //清零 FFH 地址所指向的数据
    
```

#### 2.3.1.1 R180/RSR (RAM 选择低位寄存器)

0X180	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSR	RSR<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

## 2.3.1.2 R184/RSRH (RAM 选择高位寄存器)

0X184	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSRH	RSR<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

RSRH 和 RSRL 寄存器在间接寻址方式中用于选择 SRAM 地址（寻址范围：**0x000 - 0x17F**）。

## 2.3.1.3 R1FD/IAR (间接寻址寄存器)

0X1FD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAR	IAR<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

## 2.3.1.4 R1FE/IAR1 (间接寻址寄存器)

0X1FE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAR1	IAR1<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

## 2.3.1.5 R1FF/IAR2 (间接寻址寄存器)

0X1FF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAR2	IAR2<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

间接寻址寄存器并不是一个实际存在的寄存器，它的主要功能是作为间接寻址的指针。任何以 R1FF 作为指针的指令，实际对应的地址是 R180/R184（RAM 选择寄存器）RSR<15:0>所指向的 SRAM 中数据。

## 2.3.2 查表

### 2.3.2.1 跳转表

程序寄存器 PC 控制程序的指令执行顺序，可以寻址整个 ROM 范围，PC 执行指令会自自加 1，永远指向下一条指令的地址。如果执行跳转、条件跳转、PCL 赋值、子程序调用、中断触发及返回等操作时，PC 自动加载指令相关地址并非下一条指令地址。

跳转表可以实现多地址跳转功能，由 PCL 与 ACC 值相加得到新的 PCL 值进行多地址跳转。例如 ACC=1，则表示当前地址为 PCL+1。需要注意的是跳转不能超范围使用，以免功能异常。

例 1：跳转表返回数值 0X4F

```

MOV     A, @0X03
ADD     PCL, A
RETL   @0X3F    //0
RETL   @0X06    //1
RETL   @0X5B    //2
RETL   @0X4F    //3
RETL   @0X66    //4
    
```

例 2：跳转程序 LOOP2

```

MOV     A, @0X02
ADD     PCL, A
JMP     LOOP0    //0
JMP     LOOP1    //1
JMP     LOOP2    //2
JMP     LOOP3    //3
    
```

#### 2.3.2.1.1. R181/PCH(程序计数高位寄存器)

0X181	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCH	-	-	-	PC<12:8>				
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

#### 2.3.2.1.2. R182/PCL(程序计数低位寄存器)

0X182	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	PC<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

程序计数器（PC）是用于记录每个指令周期中 CPU 所要处理的指令的指针。在 CPU 运行周期中，PC 将指令指针推进程序存储器，然后指针自增 1 以进入下一个周期。JZ8FT8801 拥

有一个 13 位宽度的程序计数器 (PC)，其低字节来自可读可写的 PCL 寄存器，高字节来自可读的 PCH 寄存器。

堆栈是用于记录程序返回的指令指针。当调用子程序时，PC 将指令指针压栈。待执行返回指令时，堆栈将指令指针送回 PC，继续进行原来的进程。JZ8FT8801 拥有 12 级堆栈，该堆栈既不占程序存储空间也不占数据存储空间，并且堆栈指针不能读写。

程序计数器 (PC) 及堆栈详细说明如下：

- (1) 寄存器 PC 和内置 12 级堆栈都是 13 位宽，用于 8K×16Bit MTP ROM 的寻址。
- (2) 一般情况下，PC 自增一；复位时，PC 的所有位都被清零。
- (3) 指令“JMP”允许直接载入 13 位地址，然后将 PC+1 推入堆栈。因此，JMP 指令允许 PC 跳转到程序的任一位置。
- (4) 指令“CALL”允许加载 PC 的 13 位地址，然后将 PC+1 推入堆栈。因此，子程序入口地址可位于程序的任一位置。
- (5) 执行“RET”指令时将栈顶数据送到 PC。
- (6) 执行“ADD PCL, A”指令可将一个相对地址与当前 PC 相加，PC 的第九位及以上各位逐次递增。
- (7) 执行“MOV PCL, A”指令可从“A”寄存器加载一个地址到 PC 的低 8 位，PC 的第九位及以上各位保持不变。
- (8) 任何（除“ADD PCL, A”指令外）向 PCL 写入值的指令（例如：“MOV PCL, A”，“BTC PCL, 1”）都会使 PC 的第九位、第十位保持不变。
- (9) 堆栈的工作犹如循环缓冲器，也就是说，压栈 12 次之后，第 13 次压栈时进栈的数据将覆盖第 1 次进栈的数据，而第 14 次压栈时进栈的数据将覆盖第 2 次进栈的数据，依此类推。

### 2.3.2.2 ROM 区查表

TBRDH 和 TBRDL 寄存器可配合 TBRD 指令查询 ROM 区地址的数据。查询数据高 8 位保存到自定义寄存器，低 8 位保存到 ACC。

例 1：查询 ROM 地址 0X00F3 的 16 位数据为 0X339E

```

MOV      A, @0x00
MOV      TBRDH, A
MOV      A, @0xF3
MOV      TBRDL, A
TBRD     reg_tempH           //高 8 位保存到 reg_tempH, 低 8 位保存到 ACC
MOV      reg_tempL, A       //ACC 值保存到 reg_tempL
    
```

Table:

```

ORG      0X00F0             //固定 ROM 程序起始地址为 0X00F0
DW      0X00F3             //0
DW      0X1103             //1
DW      0X22BC             //2
DW      0X339E             //3
    
```

例 2：未知地址，根据标号查询数据 0X1103

```

MOV      A, @Table/256     //地址高 8 位
MOV      reg_Adr_TempH, A  //地址高 8 位保存在 TempH
MOV      A, @Table+1       //地址低 8 位
MOV      reg_Adr_TempL, A  //地址低 8 位保存在 TempL
    
```



```

MOV    a, reg_Adr_TempH    //地址高位
MOV    TBRDH, a
MOV    a, reg_Adr_TempL    //地址低位
MOV    TBRDL, a
TBRD   reg_Dat_TempH       //保存高 8 位数据到 TempH
MOV    reg_Dat_TempL, a    //保存低 8 位数据到 TempL
Table:
DW     0X00F3              //0
DW     0X1103              //1
DW     0X22BC              //2
DW     0X339E              //3
DW     0X444E              //4

```

### 2.3.2.2.1. R186/TBRDH(查表指针高位寄存器)

0X186	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDH	PWM5_LEDEN	PWM1_LEDEN	MTP_POWER	RBIT<12:8>				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<4:0>: RBIT<12:8> - TBRD 指针地址高 4 位。

### 2.3.2.2.2. R187/TBRDL(查表指针低位寄存器)

0X187	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDL	RBIT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: RBIT<7:0> - TBRD 指针地址低 8 位。

RBIT <7:0> 在查表方式中用于选择查表 ROM 地址低 8 位。

## 2.3.3 状态寄存器

系统当前运行状态由状态寄存器（STATUS）标志位指示。STATUS 寄存器可以是任何寄存器的目标寄存器，根据指令的器件逻辑会影响 Z、DC、C 标志位。

### 2.3.3.1 R183/STATUS (状态标志寄存器)

0X183	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	RST	GIE	VDD_POWER	T	P	Z	DC	C
读/写	R	R	R/W	R	R	R/W	R/W	R/W
复位值	0	0	1	1	1	X	X	X

Bit<7>: RST - 复位或唤醒标志位

- 0: 其它复位类型
- 1: 由引脚状态改变唤醒

Bit<6>: GIE - 中断使能标志位

- 0: 由 DIT 指令或硬件中断屏蔽
- 1: 由 EIT/RTI 指令使能中断

Bit<5>: VDD\_POWER - VDD 电源域选择

- 0: 高抗干扰
- 1: 高性能

Bit<4>: T - 时间溢出位

- 0: WDT 溢出
- 1: 执行“SLEEP”和“CWDT”指令或低压复位

Bit<3>: P - 掉电标志位

- 0: 执行“SLEEP”指令
- 1: 上电复位或执行“CWDT”指令

影响 T/P 的事件如下表所示:

类型	RST	T	P
上电复位	0	1	1
工作模式下按 RESET	0	保持	保持
RESET 唤醒	0	1	0
工作模式下 WDT 溢出	0	0	保持
WDT 溢出唤醒	0	0	0
端口输入变化唤醒	1	1	0
执行 CWDT 指令	保持	1	1
执行 SLEEP 指令	保持	1	0

Bit<2>: Z - 零标志位算术或逻辑操作结果为零时置为”1”

- 0: 当算术或者逻辑运算结果不为 0
- 1: 当算术或者逻辑运算结果为 0

Bit<1>: DC - 辅助进位标志

- 0: 执行加法运算时，低四位没有进位产生；/执行减法运算时，低四位产生借位
- 1: 执行加法运算时，低四位有进位产生；/执行减法运算时，低四位没产生借位

Bit<0>: C - 进位标志

- 0: 执行加法运算时，高四位没有进位产生；/执行减法运算时，高四位产生借位
- 1: 执行加法运算时，高四位有进位产生；/执行减法运算时，高四位没产生借位

## 2.3.4 电源域

MTP 和 VDD 提供两种电源域选择提升抗干扰，用户可自行配置是否使能增强抗干扰能力。

### 2.3.4.1 R186/TBRDH(电源域控制寄存器)

0X186	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDH	PWM5_LEDEN	PWM1_LEDEN	MTP_POWER	RBIT<12:8>				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<5>: MTP\_POWER - MTP 电源域选择

- 0: 高抗干扰
- 1: 高性能

## 3 I/O 端口

JZ8FT8801 有 4 组双向 I/O 端口，共 30 个输入，30 个输出，大部分 I/O 可以复用为其它功能。

30 个可编程上拉 I/O 引脚：P50~P57，P60~P67，P70~P77，P80~P85；

30 个可编程下拉 I/O 引脚：P50~P57，P60~P67，P70~P77，P80~P85；

30 个可 OPTION 选择驱动增强 I/O 引脚：P50~P57，P60~P67，P70~P77，P80~P85；

10 个可 OPTION 选择拉电流驱动二级增强 I/O 引脚：P50~P55，P64~P67；

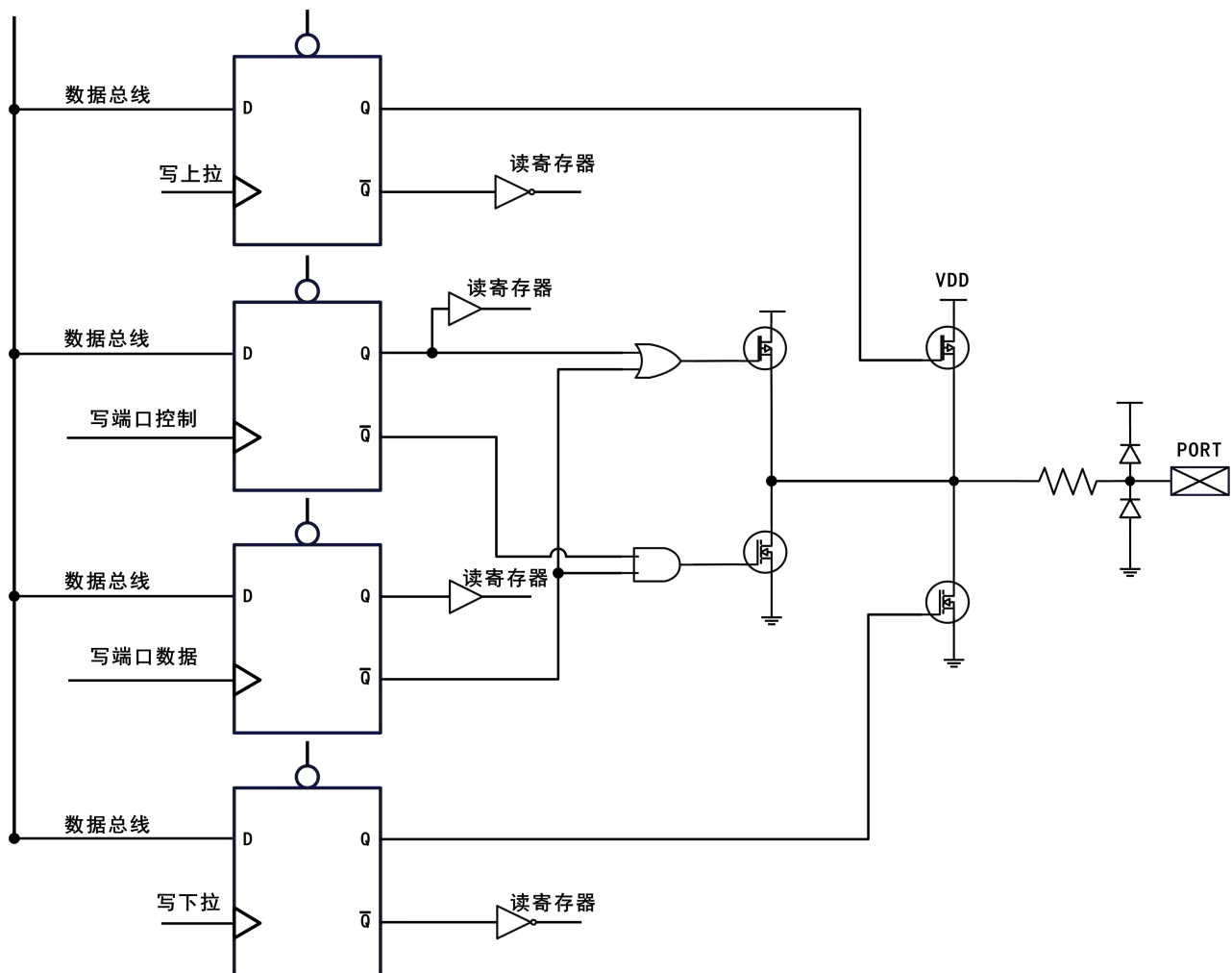
4 个可软件选择电流驱动四级增强 I/O 引脚：P60~P63；

30 个可编程唤醒 I/O 引脚：P50~P57，P60~P67，P70~P77，P80~P85；

30 个可编程模拟口：P50~P57，P60~P67，P70~P77，P80~P85；

### 3.1 GPIO 内部结构图

以下内部结构图仅供参考理解，并不代表实际电路。



I/O 控制寄存器/数据寄存器/上拉/下拉结构电路

## 3.2 P5 端口

P5 口是 8Bit 宽的双向端口。它所对应的方向控制寄存器是 P5CON。将 P5CON 的一个位置 1 (=1) 可以将相应的引脚配置为输入。清零 P5CON 的一个位 (=0) 可将相应的 P5 引脚配置为输出。

P5 口对应的数据寄存器是 PORT5。当端口为输入口时，读 PORT5 寄存器读的是引脚的状态，当端口是输出口时，读 PORT5 寄存器读的是 P5 的数据。

与 P5 口相关寄存器有 PORT5、P5CON、P5PH、P5PD、P5IWE 等。

### 3.2.1 R18A/PORT5 (P5 数据寄存器)

0X18A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT5	PORT5<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P5 端口数据控制

### 3.2.2 R18D/P5CON (P5 控制寄存器)

0X18D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5CON	P5CON<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P5 端口输入输出控制位：

- 0：输出
- 1：输入

### 3.2.3 R190/P5PH (P5 上拉控制寄存器)

0X190	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5PH	P5PH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P5 端口上拉控制位：

- 0：使能
- 1：禁止

## 3.2.4 R193/P5PD (P5 下拉控制寄存器)

0X193	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5PD	P5PD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P5 端口下拉控制位：

- 0: 使能
- 1: 禁止

## 3.2.5 R19C/P5IWE (P5 输入变化中断、唤醒使能寄存器)

0X19C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5IWE	P5IWE<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P5 端口输入变化唤醒控制位：

- 0: 禁止
- 1: 使能

## 3.2.6 R1A0/P5AE (P5 模拟口使能寄存器)

0X1A0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5AE	P5AE<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P5 端口模拟口选择控制位：

- 0: 端口设置为 GPIO
- 1: 端口设置为模拟输入口

## 3.3 P6 端口

P6 口是 8Bit 宽的双向端口。它所对应的方向控制寄存器是 P6CON。将 P6CON 的一个位置 1 (=1) 可以将相应的引脚配置为输入。清零 P6CON 的一个位 (=0) 可将相应的 P6 引脚配置为输出。

P6 口对应的数据寄存器是 PORT6。当端口为输入口时，读 PORT6 寄存器读的是引脚的状态，当端口是输出口时，读 PORT6 寄存器读的是 P6 口的数据。

与 P6 口相关寄存器有 PORT6、P6CON、P6PH、P6PD、P6IWE 等。

### 3.3.1 R18B/PORT6 (P6 数据寄存器)

0X18B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT6	PORT6<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P6 端口数据控制

### 3.3.2 R18E/P6CON (P6 控制寄存器)

0X18E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6CON	P6CON<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P6 端口输入输出控制位：

0：输出

1：输入

### 3.3.3 R191/P6PH (P6 上拉控制寄存器)

0X191	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6PH	P6PH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P6 端口上拉控制位：

0：使能

1：禁止

### 3.3.4 R194/P6PD (P6 下拉控制寄存器)

0X194	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6PD	P6PD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P6 端口下拉控制位：

- 0: 使能
- 1: 禁止

### 3.3.5 R19D/P6IWE (P6 输入变化中断、唤醒使能寄存器)

0X19D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6IWE	P6IWE<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P6 端口输入变化唤醒控制位：

- 0: 禁止
- 1: 使能

### 3.3.6 R1A1/P6AE (P6 模拟口使能寄存器)

0X1A1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6AE	P6AE <7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P6 端口模拟口选择控制位：

- 0: 端口设置为 GPIO
- 1: 端口设置为模拟输入口

## 3.4 P7 端口

P7 口是 8Bit 宽的双向端口。它所对应的方向控制寄存器是 P7CON。将 P7CON 的一个位置 1 (=1) 可以将相应的引脚配置为输入。清零 P7CON 的一个位 (=0) 可将相应的 P7 引脚配置为输出。

P7 口对应的数据寄存器是 PORT7。当端口为输入口时，读 PORT7 寄存器读的是引脚的状态，当端口是输出口时，读 PORT7 寄存器读的是 P7 口的数据。

与 P7 口相关寄存器有 PORT7、P7CON、P7PH、P7PD、P7IWE 等。

### 3.4.1 R18C/PORT7 (P7 数据寄存器)

0X18C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT7	PORT7<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P7 端口数据控制

### 3.4.2 R18F/P7CON (P7 控制寄存器)

0X18F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P7CON	P7CON<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P7 端口输入输出控制位：

- 0: 输出
- 1: 输入

### 3.4.3 R192/P7PH (P7 上拉控制寄存器)

0X192	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P7PH	P7PH<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P7 端口上拉控制位：

- 0: 使能
- 1: 禁止

### 3.4.4 R195/P7PD (P7 下拉控制寄存器)

0X195	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P7PD	P7PD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P7 端口下拉控制位:

- 0: 使能
- 1: 禁止

### 3.4.5 R19E/P7IWE (P7 输入变化中断、唤醒使能寄存器)

0X19E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P7IWE	P7IWE <7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P7 端口输入变化唤醒控制位:

- 0: 禁止
- 1: 使能

### 3.4.6 R1A2/P7AE (P7 模拟口使能寄存器)

0X1A2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P7AE	P7AE <7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P7 端口模拟口选择控制位:

- 0: 端口设置为 GPIO
- 1: 端口设置为模拟输入口

## 3.5 P8 端口

P8 口是 6Bit 宽的双向端口。它所对应的方向控制寄存器是 P8CON。将 P8CON 的一个位置 1 (=1) 可以将相应的引脚配置为输入。清零 P8CON 的一个位 (=0) 可将相应的 P8 引脚配置为输出。

P8 口对应的数据寄存器是 PORT8。当端口为输入口时，读 PORT8 寄存器读的是引脚的状态，当端口是输出口时，读 PORT8 寄存器读的是 P8 口的数据。

与 P8 口相关寄存器有 PORT8、P8CON、P8PH、P8PD 等。

### 3.5.1 R196/PORT8 (P8 数据寄存器)

0X196	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT8	-	-	PORT8<5:0>					
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P8 端口数据控制

### 3.5.2 R197/P8CON (P8 控制寄存器)

0X197	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P8CON	-	-	P8CON<5:0>					
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P8 端口输入输出控制位：

- 0: 输出
- 1: 输入

### 3.5.3 R198/P8PH (P8 上拉控制寄存器)

0X198	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P8PH	-	-	P8PH<5:0>					
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P8 端口上拉控制位：

- 0: 使能
- 1: 禁止

### 3.5.4 R199/P8PD (P8 下拉控制寄存器)

0X199	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P8PD	-	-	P8PD<5:0>					
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

P8 端口下拉控制位：

- 0: 使能
- 1: 禁止

### 3.5.5 R19A/P8AE (P8 模拟口使能寄存器)

0X19A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P8AE	-	-	P8AE<5:0>					
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

P8 端口模拟口选择控制位：

- 0: 端口设置为 GPIO
- 1: 端口设置为模拟输入口

### 3.5.6 R19F/P8IWE (P8 输入变化中断、唤醒使能寄存器)

0X19F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P8IWE	IPWM_DRV5<1:0>		P8IWE<5:0>					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7: 6>: IPWM\_DRV5<1:0> - IPWM 死区互补端口驱动能力选择 (P60/P61/P62/P63)

IPWM_DRV5<1>	IPWM_DRV5<0>	驱动能力
0	0	拉灌 25mA
0	1	拉灌 50mA
1	0	拉灌 80mA
1	1	拉灌 100mA

Bit<5: 0>: P8 口输入变化唤醒控制位

- 0: 禁止
- 1: 使能

## 3.5.7 端口输入变化唤醒

JZ8FT8801 包含 30 个可编程端口输入变化唤醒 I/O: P50~P57, P60~P67, P70~P77, P80~P85。芯片执行“SLEEP”指令可以进入到睡眠模式或者空闲模式。此时, CPU 不执行指令。端口输入变化唤醒可以通过程序选择继续原有的进程(SLEEP 前执行 DI)或执行相应的中断跳转(SLEEP 前执行 EI), 需打开相应的中断使能控制, 端口输入变化唤醒后跳转到中断服务程序。

### 端口状态改变查询方式唤醒设置:

- 1、端口输入变化唤醒口设为输入;
- 2、可以根据需要选择唤醒口的内部上拉或下拉;
- 3、使能端口输入变化唤醒控制 P5IWE/P6IWE/P7IWE/P8IWE;
- 4、执行 DI 指令, 不进入中断地址口;
- 5、执行“SLEEP”指令, IDLE=0 进入睡眠模式或者 IDLE=1 进入空闲模式;
- 6、唤醒后, 执行 SLEEP 的下一条指令;

### 端口状态改变中断方式唤醒设置:

- 1、端口输入变化唤醒口设为输入;
- 2、可以根据需要选择唤醒口的内部上拉或下拉;
- 3、使能端口输入变化唤醒控制 P5IWE/P6IWE/P7IWE/P8IWE;
- 4、使能端口输入变化中断 P5ICIE/P6ICIE/P7ICIE/P8ICIE;
- 5、执行“EI”指令, 等待进入中断地址口;
- 6、执行“SLEEP”指令, IDLE=0 进入睡眠模式或者 IDLE=1 进入空闲模式;
- 7、唤醒后会进入中断地址口, 退出中断后, 执行 SLEEP 下一条指令;

## 3.5.8 端口施密特参数

JZ8FT8801 端口的施密特性, 表格如下(仅作参考):

端口	SMT	
P50~P57	0.25*VDD	0.53*VDD
P60~P67	0.25*VDD	0.53*VDD
P70~P77	0.25*VDD	0.53*VDD
P80~P85	0.25*VDD	0.53*VDD

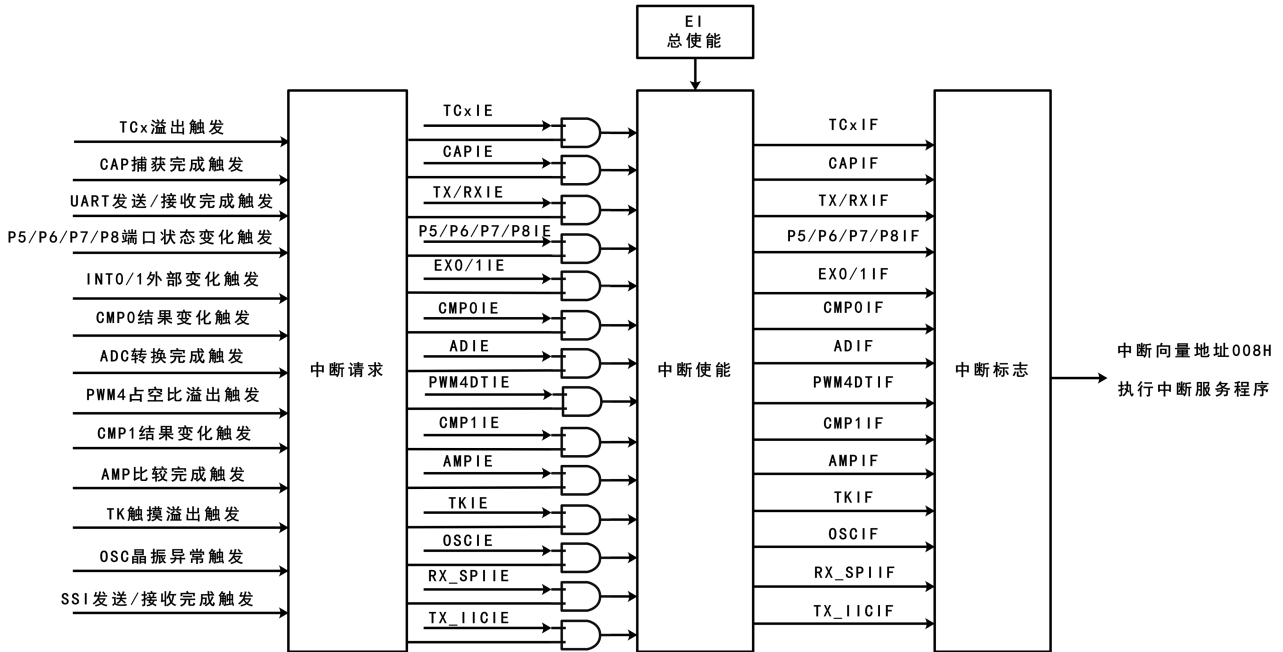
以上参数仅做参考, 请以目标样机实测数据为准。

## 4 中断

JZ8FT8801 具有 24 个中断源，无论是使用其中那一个中断，都必须使能总中断，即“EI”指令。中断向量地址固定为 008H。下面分别是每个中断的特性：

中断源	使能条件	中断标志
TC0 溢出中断	EI + TC0IE=1	TC0IF
TC1 溢出中断	EI + TC1IE=1	TC1IF
TC2 溢出中断	EI + TC2IE=1	TC2IF
TC3 溢出中断	EI + TC3IE=1	TC3IF
TC4 溢出中断	EI + TC4IE=1	TC4IF
TC5 溢出中断	EI + TC5IE=1	TC5IF
CMP0 比较完成中断	EI + CMP0IE=1	CMP0IF
UART 发送\IIC 完成中断	EI + TX_IICIE=1	TX_IICIF
UART 接收\SPI 完成中断	EI + RX_SPIIE=1	RX_SPIIF
UART 发送完成中断	EI + TXIE=1	TXIF
UART 接收完成中断	EI + RXIE=1	RXIF
P5 端口输入变化中断	EI + P5ICIE=1	P5ICIF
P6 端口输入变化中断	EI + P6ICIE=1	P6ICIF
P7 端口输入变化中断	EI + P7ICIE=1	P7ICIF
P8 端口输入变化中断	EI + P8ICIE=1	P8ICIF
INT0 外部中断	EI + EX0IE=1	EX0IF
INT1 外部中断	EI + EX1IE=1	EX1IF
CMP1 比较完成中断	EI + CMP1IE=1	CMP1IF
CMP2 比较完成中断	EI + CMP2IE=1	CMP2IF
ADC 转换完成中断	EI + ADIE=1	ADIF
PWM4 占空比溢出中断	EI + PWM4DTIE=1	PWM4DTIF
TK 触摸中断	EI + TKIE=1	TKIF
CAP 端口捕获中断	EI + CAPIE=1	CAPIF
OSC 晶振异常中断	EI + OSCIE=1	OSCIF

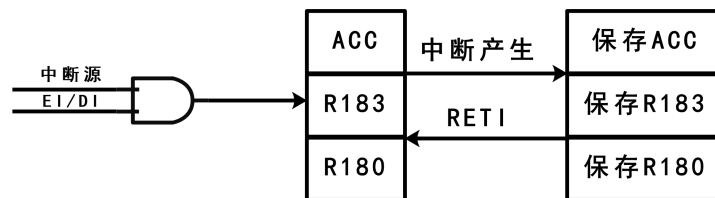
R1DA/R1DB/R1DC 为中断标志寄存器，它们记录了当某个中断产生中断请求后的中断标志位。R1D6/R1D7/R1D8 为中断使能控制寄存器，中断的允许与禁止在这三个寄存器中设置。总中断的允许是通过下“EI”指令，相反，总中断的禁止是通过下“DI”指令。当一个中断产生时，它的下一条指令的执行将从中断向量地址 008H 处执行。在离开中断服务程序之前相应的中断标志位必须清零，这样才能避免中断的误动作。



中断原理示意图

## 4.1 中断现场保护

在响应中断过程中，硬件自带中断保护功能，将 ACC、R183、R180 的内容保存起来，直到离开中断服务程序时，将被保存的值再重新载入 ACC、R183、R180，如此是为了避免在执行中断子程序时，有指令将 ACC、R183、R180 的值改变，导致返回主程序时发生错误。如下图所示：



中断现场保护示意图

## 4.2 中断相关寄存器

### 4.2.1 R1AE/EXINTCON(外部中断控制寄存器)

0X1AE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXINTCON	EXINT1MS<1:0>		EXINTOMS<1:0>		EXINT1S	EXINT0S	TCOCAPS	INTGATE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0



Bit<7:6>: EXINT1MS<1:0> - EXINT1 模式选择位

EXINT1MS<1>	EXINT1MS<0>	EXINT1 模式选择
0	0	功能禁止
0	1	上升沿触发
1	0	下降沿触发
1	1	电平变化触发

Bit<5:4>: EXINTOMS<1:0> - EXINTO 模式选择位

EXINTOMS<1>	EXINTOMS<0>	EXINTO 模式选择
0	0	功能禁止
0	1	上升沿触发
1	0	下降沿触发
1	1	电平变化触发

Bit<3>: EXINT1S - EXINT1 端口选择位

- 0: EXINT1 端口选择 P60
- 1: EXINT1 端口选择 P65

Bit<2>: EXINTOS - EXINTO 端口选择位

- 0: EXINTO 端口选择 P50
- 1: EXINTO 端口选择 P54

## 4.2.2 R1D6/INTE0 (中断使能控制寄存器 0)

0X1D6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE0	PWM4DTIE	ADIE	CMPOIE	EX1IE	EXOIE	P7ICIE	P6ICIE	P5ICIE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: PWM4DTIE - PWM4 占空比匹配中断使能位

- 0: 禁止
- 1: 使能

Bit<6>: ADIE - AD 转换完成中断使能位

- 0: 禁止
- 1: 使能

Bit<5>: CMPOIE - CMP0 比较器结果变化中断使能位

- 0: 禁止
- 1: 使能

Bit<4>: EX1IE - EXINT1 外部中断使能位

- 0: 禁止
- 1: 使能

Bit<3>: EX0IE - EXINT0 外部中断使能位

0: 禁止

1: 使能

Bit<2>: P7ICIE - P7 端口变化中断使能位

0: 禁止

1: 使能

Bit<1>: P6ICIE - P6 端口变化中断使能位

0: 禁止

1: 使能

Bit<0>: P5ICIE - P5 端口变化中断使能位

0: 禁止

1: 使能

### 4.2.3 R1D7/INTE1 (中断使能控制寄存器 1)

0X1D7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE1	P8ICIE	RX_SPIIE	TX_IICIE	CMP1IE	TC3IE	TC2IE	TC1IE	TC0IE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: P8ICIE - P8 端口变化中断使能位

0: 禁止

1: 使能

Bit<6>: RX\_SPIIE - UART 接收/SPI 发送/SPI 接收中断使能位

0: 禁止

1: 使能

Bit<5>: TX\_IICIE - UART 发送/IIC 发送/IIC 接收中断使能位

0: 禁止

1: 使能

Bit<4>: CMP1IE - CMP1 比较器结果变化中断使能位

0: 禁止

1: 使能

Bit<3>: TC3IE - TC3 溢出中断使能位

0: 禁止

1: 使能

Bit<2>: TC2IE - TC2 周期匹配中断使能位

0: 禁止

1: 使能

Bit<1>: TC1IE - TC1 周期匹配中断使能位

0: 禁止

1: 使能

Bit<0>: TC0IE - TC0 溢出中断使能位

0: 禁止

1: 使能

#### 4.2.4 R1D8/INTE2 (中断使能控制寄存器 2)

0X1D8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	TKIE	OSCIE	CAPIE	RX0IE	TX0IE	CMP2IE	TC5IE	TC4IE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TKIE - TK 触摸中断使能位

0: 禁止

1: 使能

Bit<6>: OSCIE - OSC 停振检测中断使能位

0: 禁止

1: 使能

Bit<5>: CAPIE - 捕获中断使能位

0: 禁止

1: 使能

Bit<4>: RX0IE - UART0 接收中断使能位

0: 禁止

1: 使能

Bit<3>: TX0IE - UART0 发送中断使能位

0: 禁止

1: 使能

Bit<2>: CMP2IE - CMP2 比较器结果变化中断使能位

0: 禁止

1: 使能

Bit<1>: TC5IE - TC5 溢出中断使能位

0: 禁止

1: 使能

Bit<0>: TC4IE - TC4 溢出中断使能位

- 0: 禁止
- 1: 使能

## 4.2.5 R1DA/INTF0 (中断标志寄存器 0)

OX1DA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	PWM4DTIF	ADIF	CMPOIF	EX1IF	EX0IF	P7ICIF	P6ICIF	P5ICIF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: PWM4DTIF - PWM4 占空比中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<6>: ADIF - AD 转换完成中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<5>: CMPIF - CMP0 比较器结果变化中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<4>: EX1IF - EXINT1 外部中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<3>: EX0IF - EXINT0 外部中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<2>: P7ICIF - P7 端口变化中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<1>: P6ICIF - P6 端口变化中断使能位

- 0: 未触发中断
- 1: 触发中断

Bit<0>: P5ICIF - P5 端口变化中断标志位

- 0: 未触发中断
- 1: 触发中断



## 4.2.6 R1DB/INTF1 (中断标志寄存器 1)

0X1DB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	P8ICIF	RX_SPIIF	TX_IICIF	CMP1IF	TC3IF	TC2IF	TC1IF	TC0IF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: P8ICIF - P8 端口变化中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<6>: RX\_SPIIF - UART 接收/SPI 发送/SPI 接收中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<5>: TX\_IICIF - UART 发送/IIC 发送/IIC 接收中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<4>: CMP1IF - CMP1 比较器结果变化中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<3>: TC3IF - TC3 溢出中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<2>: TC2IF - TC2 周期匹配中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<1>: TC1IF - TC1 周期匹配中断标志位

- 0: 未触发中断
- 1: 触发中断

Bit<0>: TC0IF - TC0 溢出中断标志位

- 0: 未触发中断
- 1: 触发中断

## 4.2.7 R1DC/INTF2 (中断标志寄存器 2)

0X1DC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF2	TKIF	OSCIF	CAPIF	RX0IF	TX0IF	CMP2IF	TC5IF	TC4IF
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TXIF - TX 触摸中断标志位

0: 未触发中断

1: 触发中断

Bit<6>: OSCIF - OSC 停振检测中断标志位

0: 未触发中断

1: 触发中断

Bit<5>: CAPIF - 捕获中断标志位

0: 未触发中断

1: 触发中断

Bit<4>: RX0IF - UART0 接收中断标志位

0: 未触发中断

1: 触发中断

Bit<3>: TX0IF - UART0 发送中断标志位

0: 未触发中断

1: 触发中断

Bit<2>: CMP2IF - CMP2 比较器结果变化中断标志位

0: 未触发中断

1: 触发中断

Bit<1>: TC5IF - TC5 溢出中断标志位

0: 未触发中断

1: 触发中断

Bit<0>: TC4IF - TC4 溢出中断标志位

0: 未触发中断

1: 触发中断

## 5 复位

### 5.1.1 复位功能概述

JZ8FT8801 系统提供 4 种复位方式：

- POR 上电复位
- RESET 脚输入低电平复位
- WDT 看门狗溢出复位
- LVR 低电压复位

以上任意一种复位发生时，所有的系统寄存器初始化到复位值，程序停止运行，同时程序计数器 PC 清零。复位结束后，系统从向量 0000H 处重新开始运行。

任何一种复位情况都需要一定的响应时间，系统复位机制能够保证 MCU 的可靠复位。不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同振荡器的起振时间都是不固定的。RC 振荡器的起振时间最短，晶体振荡器的起振时间则较长。在用户终端使用的过程中，应注意考虑应用场景对上电复位时间的要求。

### 5.1.2 POR 上电复位

上电复位与 LVR 操作密切相关。系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。

- 上电：系统检测到电源电压上升并等待其稳定；
- 系统初始化：所有的系统寄存器被置为初始值；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束，程序开始运行；

上电复位时间由 OPTION 中的【复位时间】选择决定，如下表所示：

PWRT 与 WDT	复位建立时间
PWRT=WDT/2	3.5ms (跟随看门狗复位时间)
PWRT=WDT/2	13ms (跟随看门狗复位时间)
PWRT=WDT/2	51ms (跟随看门狗复位时间)
PWRT=WDT/2	205ms (跟随看门狗复位时间)
PWRT≠WDT	350us (独立固定复位时间)

## 5.1.3 WDT 看门狗复位

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。

- 看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；
- 系统初始化：所有的系统寄存器被置为初始化默认值；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束，程序开始运行；

### 看门狗唤醒的说明：

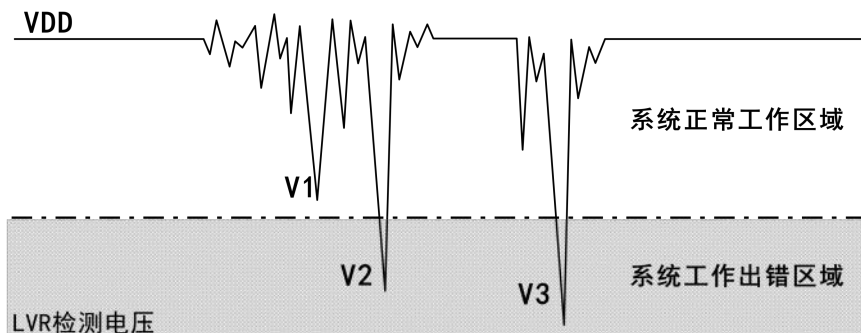
看门狗复位在空闲模式和睡眠模式下都可以复位，系统复位后从地址 0X00 开始执行程序，用户可以判断 R183 (STATUS) 寄存器 Bit4 的时间溢出位，如果为 WDT 溢出则可以判断为是看门狗复位，执行对应的唤醒子程序。**注意，看门狗唤醒实质为看门狗复位的特殊应用。**

### 看门狗定时器应用注意事项：

- ◆ 对看门狗清零之前，检查 I/O 口的状态和 RAM 可增强程序的可靠性；
- ◆ 不能在中断中对看门狗清零，否则无法侦测到主程序跑飞的状况；
- ◆ 程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

## 5.1.4 LVR 低电压复位

掉电复位针对外部因素引起的系统电压跌落情形（例如，干扰或外部负载的变化），掉电可能会引起系统工作状态不正常或程序执行错误。



电压跌落可能会进入系统死区。系统死区意味着电源不能满足系统的最小工作电压要求。上图是一个典型的掉电复位示意图。图中，VDD 受到严重的干扰，电压值降的非常低。虚线

以上区域系统正常工作，在虚线以下的区域内，系统进入未知的工作状态，这个区域称作死区。当 VDD 跌至 V1 时，系统仍处于正常状态；当 VDD 跌至 V2 和 V3 时，系统进入死区，则容易导致出错。以下情况系统可能进入死区：

## DC 运用中：

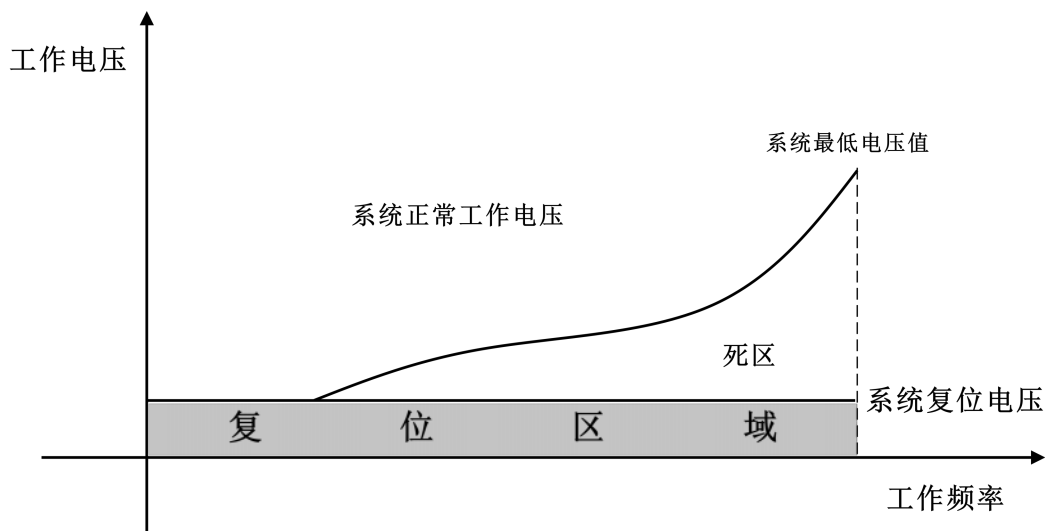
DC 运用中一般都采用电池供电，当电池电压过低或单片机驱动负载时，系统电压可能跌落并进入死区。这时，电源不会进一步下降到 LVR 检测电压，因此系统维持在死区。

## AC 运用中：

系统采用 AC 供电时，DC 电压值受 AC 电源中的噪声影响。当外部负载过高，如驱动马达时，负载动作产生的干扰也影响到 DC 电源。VDD 若由于受到干扰而跌落至最低工作电压以下时，则系统将有可能进入不稳定工作状态。在 AC 运用中，系统上、下电时间都较长。其中，上电时序保护使得系统正常上电，但下电过程却和 DC 运用中情形类似，AC 电源关断后，VDD 电压在缓慢下降的过程中易进入死区。

## 5.1.5 工作频率与 LVR 低压检测关系

为了改善系统掉电复位的性能，首先必须明确系统具有的最低工作电压值。系统最低工作电压与系统执行速度有关，不同的执行速度下最低工作电压值也不同。



如上图所示，系统正常工作电压区域一般高于系统复位电压，同时复位电压由低电压检测（LVR）电平决定。当系统执行速度提高时，系统最低工作电压也相应提高，但由于系统复位电压是固定的，因此在系统最低工作电压与系统复位电压之间就会出现一个电压区域，系统不能正常工作，也不会复位，这个区域即为死区。

为避免出现死区电压，在选择工作频率的时候，要选择相应的 LVR 复位电压点。如下表：

IRC 频率	Clocks 分频	LVR 复位电压点
IRC-16MHz	2 Clocks	LVR=3.4V
IRC-8MHz	2 Clocks	LVR=2.6V
IRC-4MHz	2 Clocks	LVR=2.0V

注：1、工作频率 = 指令周期频率 = IRC 频率/Clocks 分频；

2、此工作频率和 LVR 复位电压点的对应值，只是推荐值，用户在使用过程中，根据用于的具体应用场合可以适当的调整复位电压点。

## 5.1.6 复位相关寄存器

### 5.1.6.1 R1AF/WDTCON (WDT、唤醒控制寄存器)

0X1AF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON	WDTE	WDTPSR<1:0>		LVREN	UARTOWE	CMPOWE	EXINT1WE	EXINTOWE
读/写	R/W	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

Bit<7>: WDTE - WDT 看门狗使能控制

0: 禁止

1: 使能

Bit<6:5>: WDTPSR<1:0> - WDT 预分频选择位：

WDTPSR<1>	WDTPSR<0>	WDT 分频系数
0	0	1:4
0	1	1:8
1	0	1:16
1	1	1:32

Bit<4>: LVREN - LVR 使能控制

0: 使能

1: 禁止

## 6 系统时钟与工作模式

### 6.1 系统时钟

JZ8FT8801 内部集成了 3 种振荡器，高速 RC 振荡器 IHRC、低速 RC 振荡器 ILRC 和外部晶振，可以通过 R188/CPUCON 寄存器实现系统时钟切换高低速振荡器。

#### 6.1.1 内部 IHRC 振荡器

JZ8FT8801 内置 IHRC 高速振荡器，提供 IHRC 频率可配置功能，频率默认值为 8MHz，IHRC 频率可寄存器微调（IHRCCAL 寄存器）。

IHRC 振荡器包含 4M/8M/16M/32MHz 四种频率值。通过设置 OPTION 的配置位，可选择 IHRC 工作频率，下面是它们的对应关系：

RCM	IRC 频率
4M	IHRC 频率选为 4MHz
8M	IHRC 频率选为 8MHz
16M	IHRC 频率选为 16MHz
32M	IHRC 频率选为 32MHz

JZ8FT8801 提供了指令时钟多种分频选择，可以在 OPTION 中选择，适用于更多的场合。如下表：

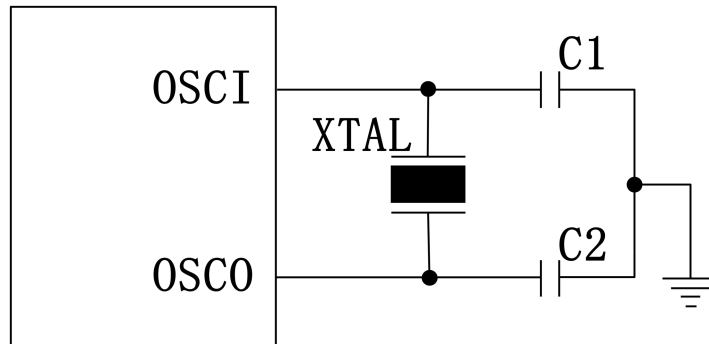
Clocks	Clocks 分频
2clock	分频为 2clock
4clock	分频为 4clock
8clock	分频为 8clock
16clock	分频为 16clock
32clock	分频为 32clock

#### 6.1.2 内部 ILRC 振荡器

JZ8FT8801 内置 ILRC 低速振荡器，提供稳定的 40KHz 低速时钟。

#### 6.1.3 外部晶体振荡器

JZ8FT8801 内置 HXT、LXT 晶振起振电路，在大多数应用中，引脚 OSC0 和 OSC1 上可接晶体或陶瓷谐振器来产生振荡。电路图如下，表中为 C1、C2 的推荐值。由于各个谐振器特性不同，用户应参照其规格选择 C1、C2 的合适值。



晶体振荡器应用电路

晶体振荡器或陶瓷振荡器的电容选择参考：

振荡器模式	频率模式	频率	C1/C2
低速晶体振荡器	LXT	32.768KHz	5~30pF
高速晶体振荡器	HXT	400KHz~16MHz	5~30pF

当选择 LXT 或者选择 RTC 功能外接低速晶振时，配置 R19B/LXTCON 寄存器可以配置晶振检错、晶振起振增强、晶振频率输出功能。

(1) 使能 LXT\_ERREN 开启晶振检错功能，判断 LXT\_ERR 标志位来检测晶振是否有正常起振，如果 LXT\_ERR=1 则晶振正常起振，如果 LXT\_ERR=0 则晶振未正常起振。如果使能 OSC 晶振异常中断，当 LXT\_ERR=0 时可触发中断。

(2) 使能 LXT\_ST 或 LXT\_TRIM 开启晶振起振增强，两个控制位分别使能有效。LXT\_TRIM 使能则分为四级增强修调，当增强修调模式档位越高则起振能力越强芯片功耗越大，档位越低相应的晶振起振能力越弱功耗越低。建议两个控制位必须保持一路打开，以免影响起振。

(3) 使能 CRYOUTEN 可选择从 P55 或 P62 口输出晶振频率。

## 6.1.4 模块应用说明

- 内部振荡器是最常用的振荡模式，该模式可以省去外接的电路；
- 在使用外部时钟输入时，时钟信号要从 OSCI 输入，OSCO 可以悬空；
- 外界条件不同，各振荡模式的时钟频率可能会有轻微差别，使用时应根据需要合理选择；

## 6.2 工作模式

JZ8FT8801 可以在 4 种工作模式下以不同的时钟频率工作，这些模式可以控制振荡器的工作、程序的执行以及模拟电路的功能损耗。

- 高速模式：系统时钟选择内部高速时钟，低速时钟正常工作；

- 低速模式：系统时钟选择内部低速时钟，高速时钟暂停工作；
- 空闲模式：系统时钟正常工作，系统其他部分进入睡眠（TC0、TC1、TC2、TC3、TC4、TC5 选择系统时钟可继续工作并可唤醒系统），还可通过 P5IWE, P6IWE, P7IWE, P8IWE, INTOWE, INT1WE, ADCWE, CMP0IWE, CMP1WE, CMP2WE, UARTWE, TKWE, SSIWE 唤醒；
- 睡眠模式：所有功能暂停工作，系统进入睡眠，可通过TC0（RTC/P62输入），TC3（RTC/P56输入），P5IWE, P6IWE, P7IWE, P8IWE, INTOWE, INT1WE, CMPOWE, CMP1WE, CMP2WE, UARTWE, SSIWE唤醒；

功能模块	高速模式	低速模式	空闲模式	睡眠模式
IHRC	运行	停止	运行	停止
ILRC	运行	运行	运行	停止
CPU 指令	执行	执行	停止	停止
TC0	可工作	可工作	可工作	可工作（P62 输入时钟/RTC）
TC1	可工作	可工作	可工作	停止
TC2	可工作	可工作	可工作	停止
TC3	可工作	可工作	可工作	可工作（P56 输入时钟/RTC）
TC4	可工作	可工作	可工作	停止
TC5	可工作	可工作	可工作	停止
中断	全部有效	全部有效	全部有效 (TC0, TC1, TC2, TC3, TC4, TC5 选系统时钟)	TC0IE, TC3IE, PxICIE, INTxIE, CMPxIE,
唤醒功能	-	-	(TC0, TC1, TC2, TC3, TC4, TC5 选系统时钟可唤醒) PxIWE, INTxWE, ADCWE, CMPOWE, CMP1WE, CMP2WE, UARTWE, TKWE, SSIWE	TCOWE, TC3WE, PxIWE, INTxWE, CMPOWE, CMP1WE, CMP2WE, SSIWE
看门狗	WDT 选项	WDT 选项	WDT 选项控制	WDT 选项控制

## 6.2.1 高速模式

高速模式是系统高速时钟工作模式，系统时钟源由高速 RC 振荡器提供。程序被执行。上电复位或任意一种复位触发后，系统进入高速模式执行程序。高速模式下，高速振荡器正常工作，功耗最大。

- ◆ 程序被执行，所有的功能都可控制；
- ◆ 系统速率为高速；
- ◆ 高速振荡器和内部低速振荡器都正常工作；

- ◆ 高速模式可以切换到低速模式；
- ◆ 从高速模式进入到睡眠模式，唤醒后返回到高速模式；
- ◆ 从高速模式进入到空闲模式，唤醒后返回到高速模式；

## 6.2.2 低速模式

低速模式为系统低速时钟工作模式。系统时钟源由低速 RC 振荡器提供。低速模式由 CPU 模式控制寄存器的 CLKMD 位控制。当 CLKMD=0 时，系统为高速模式；当 CLKMD=1 时，系统进入低速模式。进入低速模式后，系统不能自动禁止高速 RC 振荡器，必须通过 SPTHX=1 来禁止以减少功耗。

- ◆ 程序被执行，所有的功能都可控制；
- ◆ 系统速率为低速；
- ◆ 内部低速振荡器正常工作，高速振荡器停止工作。
- ◆ 低速模式可以切换到高速模式；
- ◆ 从低速模式进入到睡眠模式，唤醒后返回到低速模式；
- ◆ 从低速模式进入到空闲模式，唤醒后返回到低速模式；

## 6.2.3 空闲模式

空闲模式是另外的一种理想状态。在睡眠模式下，所有的功能和硬件设备都被禁止，但在空闲模式下，系统时钟保持工作，空闲模式下的功耗大于睡眠模式下的功耗。空闲模式下，不执行程序，但具有唤醒功能的 TC0, TC1, TC2, TC3, TC4, TC5, P5IWE, P6IWE, P7IWE, P8IWE, INT0WE, INT1WE, ADCWE, CMP0IWE, CMP1WE, CMP2WE, UARTWE, SSIWE 仍正常工作，定时器 TCx 的时钟源为仍在工作的系统时钟。由系统时钟及状态控制寄存器的 IDLE 位决定是否进入空闲模式，当 IDLE=1，执行 SLEEP 后进入空闲模式。

- ◆ 程序停止执行，所有的功能被禁止；
- ◆ 具有唤醒功能的模块正常工作；
- ◆ 系统时钟正常工作；
- ◆ 由高速模式进入到空闲模式，被唤醒后返回到高速模式；
- ◆ 由低速模式进入到空闲模式，被唤醒后返回到低速模式；
- ◆ 空闲模式下的唤醒方式为 P<sub>x</sub>IWE, INT<sub>x</sub>WE, ADCWE, CMP<sub>x</sub>WE, TKWE, UARTWE, SSIWE；
- ◆ 空闲模式下 TC0、TC1、TC2、TC3、TC4 和 TC5 功能仍然有效；

## 6.2.4 睡眠模式

睡眠模式是系统的理想状态，不执行程序，振荡器也停止工作。整个芯片的功耗低于 1uA。睡眠模式可以由 TC0 (RTC/P62 输入时钟)，TC3 (RTC/P56 输入时钟)，P5IWE, P6IWE, P7IWE, P8IWE, INT0WE, INT1WE, CMP1WE, CMP2WE, UARTWE, SSIWE 唤醒。从高速模式或者低速模式进入睡眠模式，被唤醒后将返回到对应模式。由系统时钟及状态控制寄存器的 IDLE 位控制是否进入睡眠模式，当 IDLE=0，执行 SLEEP 后进入睡眠模式。

- ◆ 程序停止执行，所有的功能被禁止；
- ◆ 所有的振荡器，包括外部高速振荡器、内部高速振荡器和内部低速振荡器都停止工作；
- ◆ 功耗低于 1uA；
- ◆ 由高速模式进入到睡眠模式，被唤醒后返回到高速模式；
- ◆ 由低速模式进入到睡眠模式，被唤醒后返回到低速模式；
- ◆ 睡眠模式下的唤醒方式为 TC0 (P62 输入时钟/RTC)，TC3 (P56 输入时钟/RTC)，P5IWE, P6IWE, P7IWE, P8IWE, INT0WE, INT1WE, CMP0WE, CMP1WE, CMP2WE, UARTWE, SSIWE。

## 6.3 相关寄存器

### 6.3.1 R185/WKCON(唤醒控制寄存器)

0X185	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WKCON	ADCGATE1	TC3GATE	TKWE	CMP2WE	CMP1WE	UART1WE	TC5WE	TC4WE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<5>: TKWE - TKWE 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能，可唤醒空闲模式

Bit<4>: CMP2WE - CMP2 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能，可唤醒空闲模式

Bit<3>: CMP1WE - CMP1 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能，可唤醒空闲模式

Bit<2>: UART1WE - SSI 三合一串口唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能，可唤醒空闲模式

Bit<1>: TC5WE - TC5 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<0>: TC4WE - TC4 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

### 6.3.2 R188/CPUCON (CPU 模式控制寄存器)

0X188	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CPUCON	ADCWE	TC3WE	TC2WE	TC1WE	TCOWE	STPHX	CLKMD	IDLE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: ADCWE - ADC 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<6>: TC3WE - TC3 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<5>: TC2WE - TC2 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<4>: TC1WE - TC1 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<3>: TCOWE - TC0 唤醒使能位

- 0: 唤醒禁止
- 1: 唤醒使能, 可唤醒空闲模式

Bit<2>: STPHX - 高速时钟控制位

- 0: 高速时钟正常工作
- 1: 停止高速时钟

Bit<1>: CLKMD - 系统时钟控制位

- 0: 系统时钟使用高速振荡器 (IHRC) 时钟
- 1: 系统时钟使用低速振荡器 (ILRC) 时钟

系统从高速模式进入低速模式时 先设置 CLKMD=1, 后设置 STPHX=1;

系统从低速模式进入高速模式时 先设置 STPHX=0, 后设置 CLKMD=0。

Bit<0>: IDLE - 空闲模式使能位

0: 系统执行 SLEEP 指令时进入睡眠模式

1: 系统执行 SLEEP 指令时进入空闲模式, 系统时钟正常工作

TC0~TC5 和 PWM 在空闲模式下如果选择系统时钟可继续工作, 并可唤醒系统。

### 6.3.3 R189/IHRCCAL (IHRC 频率微调寄存器)

0X189	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IHRCCAL	-	IHRCCAL<6:0>						
读/写	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	X	X	X	X	X	X	X

IHRC 高速振荡器频率微调

### 6.3.4 R19B/LXTCON (LXT 控制寄存器)

0X19B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LXTCON	-	LXT_ERREN	LXT_ST	LXT_ERR	LXT_TRIM<1:0>		CRYOUTEN<1:0>	
读/写	R	R/W	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	1	X	1	1	0	0

Bit<6>: LXT\_ERREN - 低速晶振功能检测使能

0: 禁止

1: 使能

Bit<5>: LXT\_ST - 低速晶振启动增强

0: 禁止

1: 使能

Bit<4>: LXT\_ERR - 低速晶振功能异常标志位

0: 非正常

1: 正常

Bit<3:2>: LXT\_TRIM<1:0> - 低速晶振增强修调

LXT_TRIM1	LXT_TRIM0	低速晶振增强修调档位
0	0	不增强
0	1	1 档
1	0	2 档
1	1	3 档

Bit<1>: CRYOUTEN<1> - P62 端口晶振频率输出

0: 禁止

1: 使能

Bit<0>: CRYOUTEN<0> - P55 端口晶振频率输出

- 0: 禁止
- 1: 使能

### 6.3.5 R1AF/WDTCON (WDT、唤醒控制寄存器)

0X1AF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON	WDTE	WDTPSR<1:0>		LVREN	UARTOWE	CMPOWE	EXINT1WE	EXINTOWE
读/写	R/W	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

Bit<3>: UARTOWE - UART0 唤醒使能位

- 0: 禁止
- 1: 使能

Bit<2>: CMPOWE - CMPO 唤醒使能位

- 0: 禁止
- 1: 使能

Bit<1>: EXINT1WE - EXINT1 唤醒使能位

- 0: 禁止
- 1: 使能

Bit<0>: EXINTOWE - EXINT0 唤醒使能位

- 0: 禁止
- 1: 使能

## 7 定时计数器

### 7.1 TC0 定时计数器

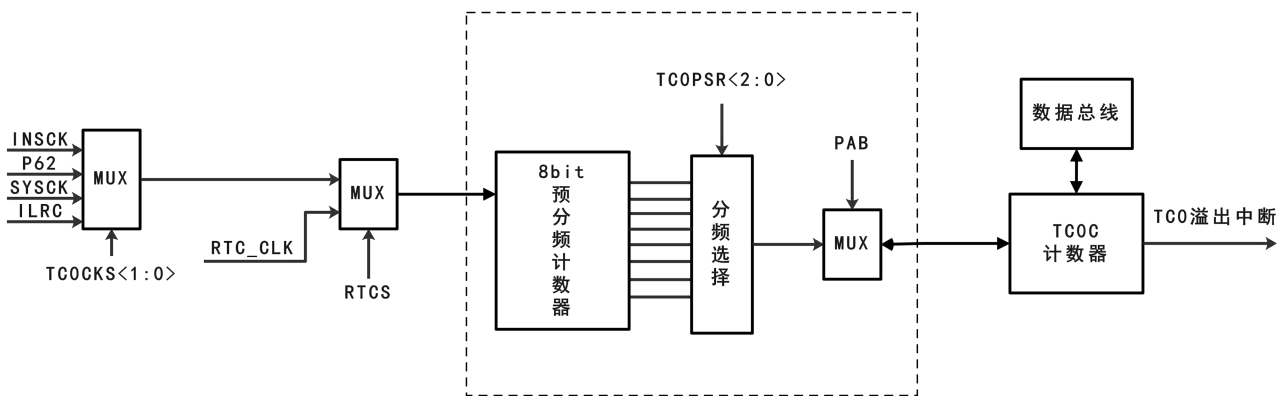
JZ8FT8801提供一个8位定时计数器作为TC0预分频器，TC0CON寄存器的PAB位决定是否预分频，TCOPSR<2:0>三位决定预分频比。

TC0 是一个 8Bit 上行计数器，只要有时钟就工作。时钟源既可以是内部指令时钟、系统时钟或者低速时钟，也可以选择外部时钟（由 P62 引脚输入，触发沿可选）或 RTC 时钟（外接 32.768KHz 晶振），时钟到来计数器实现加 1。

TC0 计数溢出可以形成中断信号。

在 IDLE 模式下，TC0 溢出中断可以唤醒电路，唤醒后可以选择进中断或者继续执行原程序。

在睡眠模式下，TC0 设置为 P62 外部输入时钟或 RTC 时钟可以中断溢出并唤醒电路。



TC0 结构框图

#### 7.1.1 RTC 模式说明

R1C0/TC0CON 寄存器选择 RTC 时钟、设置 TC0EDG=1，使能 R1C2/TC3CON 中 RTCEN=1 且 option 中【RTC】选择使能，打开 RTC 模式。此模式下 TC0 中断时钟源为外部 32.768KHz 低速晶振（默认 4 分频），主时钟为所选 IRC 时钟。使能 RTC 后建议外加 5~30pF 范围晶振电容，根据需求设置合适容值。

软件使能 R19B/LXTCON 寄存器中第 5 位 LXT\_ST=1 可增强低速晶振起振，当晶振起振后关闭使能，以免增加功耗。同时使能第 6 位 LXT\_ERREN 开启晶振检测使能，检测第 4 位 LXT\_ERR 只读标志位，当 LXT\_ERR=1 时晶振正常起振，LXT\_ERR=0 时晶振异常。

RTC 定时精度和外接晶振以及电容有直接关系，建议使用时选用高精度器件。电源波动也会影响定时，需要在电源地加电容滤波优化。

## 7.1.2 TC0 定时设置说明

- 1、给 TC0C 寄存器赋初始值；
- 2、设置 TC0CON 寄存器的值（选择时钟源及预分频比）；
- 3、作为计数器使用，需要在 TC0CON 寄存器选择 TC0 信号源为外部输入信号；当选择信号源为 RTC 时钟时，需要在 TC0CON 寄存器选择 RTC, TC0EDG=1、RTCEN=1 并打开 OPTION 中的 RTC 使能；
- 4、当配置都设置好之后，最后再设置 TC0CON 寄存器中的 TC0EN 对定时计数器使能；
- 5、若需要执行中断功能，须设置 INTE1 寄存器中的 TC0IE (Bit0) 为 1，并执行 EI 指令；
- 6、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC0 中断标志位；
- 7、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC0 中断标志位；

## 7.1.3 定时计算说明

TC0 定时功能通过写值到 TC0C 寄存器，给定时器赋初始值，定时器从初始值位置开始累加，直至定时器溢出产生中断。

**TC0 定时时间计算公式（选择内部系统时钟）：**

$$\text{TC0 定时时间} = (1/F_{\text{osc}}) \times (\text{TC0 分频}) \times (256 - \text{TC0 初始值})$$

示例：

$F_{\text{osc}}=8 \text{ MHz}$ ，TC0 分频选择=8 分频，TC0 初始值=156；

$$\text{TC0 定时时间} = (1/8) \times (8) \times (256 - 156) = 100 \text{ us}$$

**TC0 定时时间计算公式（选择外部输入时钟）：**

$$\text{TC0 定时时间} = (1/\text{外部输入时钟}) \times (\text{TC0 分频}) \times (256 - \text{TC0 初始值})$$

示例：

外部输入时钟=1 MHz，TC0 分频选择=4 分频，TC0 初始值=156；

$$\text{TC0 定时时间} = (1/1) \times (4) \times (256 - 156) = 400 \text{ us}$$

**TC0 定时时间计算公式（选择 RTC 时钟）：**

$$\text{TC0 定时时间} = (1/32.768\text{K}) \times (\text{TC0 分频}) \times (256 - \text{TC0 初始值}) \times 4$$

示例:

外部晶振=32.768KHz, TC0 分频选择=32 分频, TC0 初始值=0;

$$TC0 \text{ 定时时间} = (1/32768) \times (32) \times (256) \times (4) = 1s$$

## 7.1.4 TC0 唤醒说明

### 1. 空闲模式

在空闲模式下, 指令时钟停止执行, 具有唤醒功能的模块和系统时钟正常工作, 其它功能被禁止。

TC0 可以唤醒空闲模式, 设置 R188/CPUCON 寄存器 Bit3 位 TC0WE 为 1, IDLE = 1 加上 SLEEP 指令系统进入空闲模式, TC0 定时器正常工作 (选择系统时钟)。当 TC0 定时器溢出后, 系统被唤醒。若使能 TC0IE 及 EI, 则唤醒后进入中断, 若执行 DI, 则唤醒后执行下一条指令。

### 2. 睡眠模式

在睡眠模式下, 振荡器停止运行, 所有功能都被禁止, 仅外围输入时钟可以继续工作。

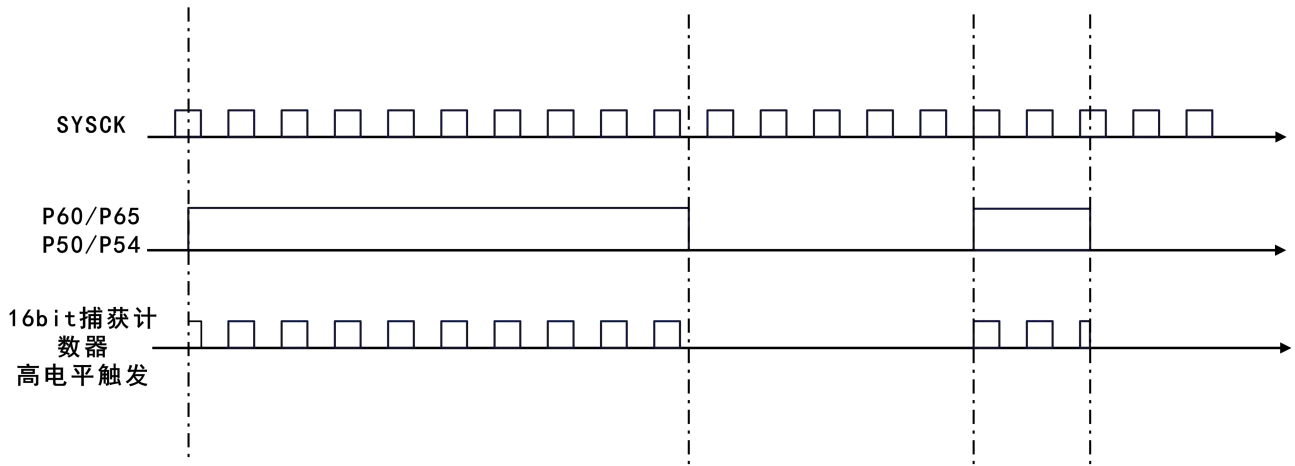
IDLE = 0 加上 SLEEP 指令系统进入睡眠模式。当时钟源选择 P62 外部时钟或 RTC 晶振时钟时, TC0 按照外接的 32.768KHz 晶振运行, 主时钟按照 IRC 运行。设置 R188/CPUCON 寄存器 Bit3 位 TC0WE 为 1, 当定时器溢出后 TC0 可以唤醒睡眠模式。若在睡眠前使能 TC0IE 及 EI, 则唤醒后进入中断, 若执行 DI, 则唤醒后执行下一条指令。

## 7.1.5 TC0 端口电平捕获说明

JZ8FT8801 提供外部端口电平捕获功能。

捕获的信号源可通过 EXINTCON (1AE) 寄存器选择 P60、P65、P50、P54 任意一个外部端口进行捕获。通过 EXINTCON 寄存器 Bit0 位 INTGATE 使能捕获模式, 通过复用 TC0 8Bit 预分频器 TPRE (电平捕获计数器低 8 位) 和 8Bit 计数器 TCOC (电平捕获计数器高 8 位) 组合为 16Bit 电平捕获计数器, 同时配合外部中断实现电平变化触发中断, 在中断程序中判断上一个电平的捕获时间。如果信号周期较长, 需注意定时器计数器可能的溢出问题, 通过适当设置定时器的工作模式和预分频值来匹配信号特性。用户可以通过外部中断口下降沿触发中断并在中断程序中读取 0x1C1/TCOC、0x1AD/TPRE 寄存器的计数值, 以此计算高电平时间。当外部中断口为高电平时 TC0 预分频器和计数器开始计数, 当外部中断口为低电平则 TC0 停止计数。捕获低电平时间则配置上升沿触发, 在低电平时计数, 高电平时停止计数。

捕获可通过 R1FC/CMP1CON1 寄存器使能 CMPGATE 选择 CMP0OUT 作为信号源进行捕获。使用原理及方法见 [RFC 电阻频率转换](#)。



16bit捕获计数器由低八位的TBRE和高八位的TC0组成

## 7.1.6 TC0 相关寄存器

### 7.1.6.1 R1AD/TPRE(TC0 预分频器/TC0 捕获低 8 位寄存器)

0X1AD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPRE	TPRE<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

### 7.1.6.2 R1AE/EXINTCON(外部中断控制寄存器)

0X1AE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXINTCON	EXINT1MS<1:0>		EXINT0MS<1:0>		EXINT1S	EXINT0S	TCOCAPS	INTGATE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:6>: EXINT1MS<1:0> - EXIN T1 模式选择位

EXINT1MS<1>	EXINT1MS<0>	EXINT1 模式选择
0	0	功能禁止
0	1	上升沿触发
1	0	下降沿触发
1	1	电平变化触发

Bit<5:4>: EXINTOMS<1:0> - EXINT0 模式选择位

EXINTOMS<1>	EXINTOMS<0>	EXINT0 模式选择
0	0	功能禁止
0	1	上升沿触发
1	0	下降沿触发
1	1	电平变化触发

Bit<3>: EXINT1S - EXINT1 端口选择位

0: EXINT1 端口选择 P60

1: EXINT1 端口选择 P65

Bit<2>: EXINTOS - EXINT0 端口选择位

0: EXINT0 端口选择 P50

1: EXINT0 端口选择 P54

Bit<1>: TCOCAPS - TCO 捕获输入源选择位 (CMPGATE=1 时, TCOCAPS 选择无效)

0: EXINT0

1: EXINT1

Bit<0>: INTGATE - TCO 外部端口捕获门控使能位

0: 禁止

1: 使能

### 7.1.6.3 R1C0/TCOCON (TCO 控制寄存器)

0X1C0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCOCON	TCOEN	TCOCKS<1:0>		TCOEDG	TCOPTEN	TCOPSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TCOEN - TCO 定时计数使能位

0: 禁止 TCO

1: 使能 TCO

Bit<6:5>: TCOCKS<1:0> - TCO 信号源选择位

00: 指令时钟

01: 外部输入信号 (P62)

10: 系统时钟

11: ILRC 时钟 (低速振荡器) /RTC 时钟 (TCOEDG=1 且使能 **RTCEN**)

Bit<4>: TCOEDG - TCO 信号边沿选择位

0: TCO 引脚信号发生由高到低变化加 1

1: TCO 引脚信号发生由低到高变化加 1

Bit<3>: TCOPTEN - 预分频器使能位

- 0: 禁止 TCO 预分频
- 1: 使能 TCO 预分频

Bit<2:0>: TCOPSR<2:0> - TCO 预分频选择控制位:

TCOPSR<2>	TCOPSR<1>	TCOPSR<0>	TCO 分频系数
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

#### 7.1.6.4 R1C1/TC0C (TC0 计数寄存器)

0X1C1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0C	TC0C<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

#### 7.1.6.5 R1C2/TC3CON (RTC 使能寄存器)

0X1C2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3CON	TC3EN	RTCEN	TC3CKS<1:0>		TC3PTEN	TC3PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<6>: RTCEN - RTC 模式选择位

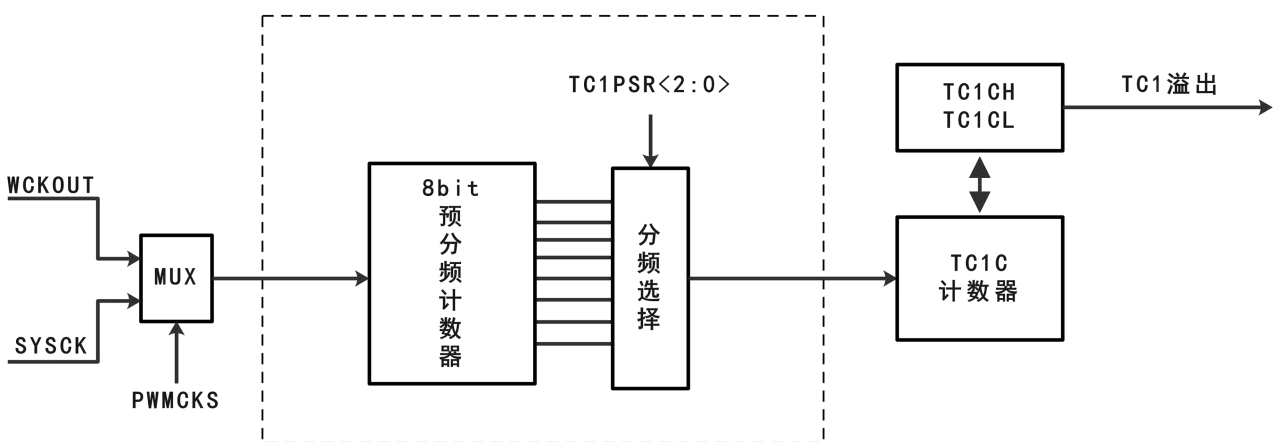
- 0: 禁止
- 1: 使能 (注: RTC 使能之后, TC3 只能用作 RTC 功能)

## 7.2 TC1 定时计数器

TC1定时计数器提供一个8位预分频器，TC1CON寄存器的TC1PSR<2:0>三位决定预分频比。在TC1模式下每次TC1EN使能，预分频寄存器就被清零。

TC1 是一个 12Bit 上行计数器。TC1 定时器需使能 TC1EN 才能工作，时钟源是内部系统时钟，每个时钟周期到来，计数器实现加 1。TC1 计数值与 TC1PRD 设定值一致产生溢出，若使能 TC1IE 及 EI，系统跳转到对应的中断向量地址，执行中断服务程序。

在 IDLE 模式下，TC1 中断可以唤醒电路（时钟为系统时钟），唤醒后可以选择进中断或者继续执行原程序，方法参考端口输入变化唤醒。



TC1 结构框图

### 7.2.1 TC1 定时设置说明

- 1、给 TC1PRD<11:0>寄存器赋初始值；
- 2、设置 TC1CON 寄存器，按需配置预分频比；
- 3、使能 R1D7/INTE1 寄存器的 TC1IE 打开中断使能，并执行 EI 指令；
- 4、使能 TC1EN，打开 TC1 定时器计数；
- 5、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC1 中断标志位；

### 7.2.2 TC1 定时计算说明

TC1 定时功能通过写值到 TC1PRD<11:0>，定时器从初始值 0x01 开始累加，直至定时器计数值与 TC1PRD<11:0>值匹配，则溢出执行中断服务程序。

TC1 定时时间计算公式：

$$\text{TC1 定时时间} = (1/\text{Fosc}) \times (\text{TC1 分频}) \times (\text{TC1PRD})$$

示例:

Fosc=16MHz, TC1 分频选择=16 分频, TC1PRD 值=512;

$$\text{TC1 定时时间} = (1/16) \times (16) \times (512) = 512\mu\text{s}$$

## 7.2.3 TC1 空闲模式唤醒说明

TC1 可以唤醒空闲模式, 使能 R188/CPUCON 寄存器 Bit4 位 TC1WE。IDLE = 1 加上 SLEEP 指令系统进入空闲模式, TC1 定时器正常工作。当 TC1 定时器溢出后, 系统被唤醒。若使能 TC1IE 及 EI, 则唤醒后进入中断, 若执行 DI, 则唤醒后执行下一条指令。

## 7.2.4 TC1 相关寄存器

### 7.2.4.1 R1B0/TC1CON(TC1 控制寄存器)

0X1B0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1CON	TC1EN	TC21EN	-	TC1CKS	TC1PTEN	TC1PSR<2:0>		
读/写	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC1EN - TC1/PWM1234 计数器使能控制

(若设置 TC1PRD 寄存器, TC1 计数到预设值复位, TC1 从 1 开始计数)

0: 禁止

1: 使能

Bit<6>: TC21EN - TC1/PWM1234、TC2/PWM567 计数器同时使能控制位

0: 禁止

1: 使能

Bit<5>: 未使用

Bit<4>: TC1CKS - TC1 时钟选择

0: 系统时钟

1: 低速时钟

Bit<3:0>: TC1PTEN、TC1PSR2~TC1PSR0 分频系数选择位:

TC1PTEN	TC1PSR<2>	TC1PSR<1>	TC1PSR<0>	TC1 分频系数
0	0	0	0	1:1
1	0	0	0	1:2
1	0	0	1	1:4
1	0	1	0	1:8



1	0	1	1	1:16
1	1	0	0	1:32
1	1	0	1	1:64
1	1	1	0	1:128
1	1	1	1	1:256

### 7.2.4.2 R1B1/TC1PRDL (TC1 低 8 位计数寄存器)

0X1B1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRDL	TC1PRD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

TC1 (PWM1、PWM2、PWM3、PWM4) 周期低 8 位

### 7.2.4.3 R1B5/TC1PRDTH (TC1 高 4 位计数寄存器)

0X1B5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRDTH	TC1PRD<11:8>				PWM3DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

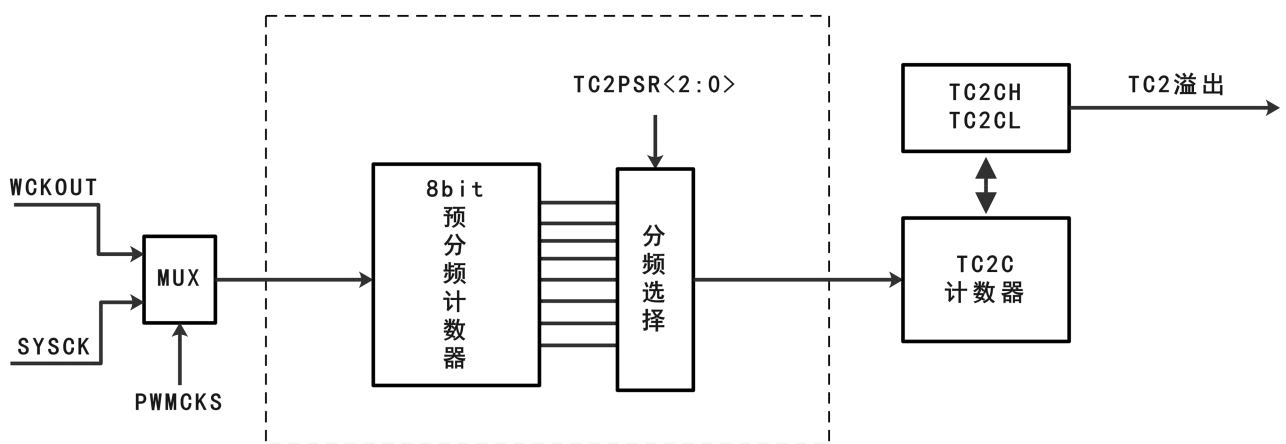
TC1 (PWM1、PWM2、PWM3、PWM4) 周期高 4 位

## 7.3 TC2 定时计数器

TC2定时计数器提供一个8位预分频器，TC2CON寄存器的TC2PSR<2:0> 三位决定预分频比。在TC2模式下每次TC2EN使能，预分频寄存器就被清零。

TC2 是一个 12Bit 上行计数器。TC2 定时器需使能 TC2EN 才能工作，时钟源是内部系统时钟，每个时钟周期到来，计数器实现加 1。TC2 计数值与 TC2PRD 设定值一致产生溢出，若使能 TC2IE 及 EI，系统跳转到对应的中断向量地址，执行中断服务程序。

在 IDLE 模式下，TC2 中断可以唤醒电路（时钟为系统时钟），唤醒后可以选择进中断或者继续执行原程序，方法参考端口输入变化唤醒。



TC2 结构框图

### 7.3.1 TC2 定时设置说明

- 1、给 TC2PRD<11:0>寄存器赋初始值；
- 2、设置 TC2CON 寄存器，按需配置预分频比；
- 3、使能 R1D7/INTE1 寄存器的 TC2IE 打开中断使能，并执行 EI 指令；
- 4、使能 TC2EN，打开 TC2 定时器计数；
- 5、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC2 中断标志位；

### 7.3.2 TC2 定时计算说明

TC2 定时功能通过写值到 TC2PRD<11:0>，定时器从初始值 0x01 开始累加，直至定时器计数值与 TC2PRD<11:0>值匹配，则溢出执行中断服务程序。

TC2 定时时间计算公式：

$$\text{TC2 定时时间} = (1/F_{\text{osc}}) \times (\text{TC2 分频}) \times (\text{TC2PRD})$$

示例:

Fosc=16 MHz, TC2 分频选择=16 分频, TC2PRD 值=512;

$$TC2 \text{ 定时时间} = (1/16) \times (16) \times (512) = 512\mu s$$

### 7.3.3 TC2 空闲模式唤醒说明

TC2 可以唤醒空闲模式, 使能 R188/CPUCON 寄存器 Bit5 位 TC2WE。IDLE = 1 加上 SLEEP 指令系统进入空闲模式, TC2 定时器正常工作。当 TC2 定时器溢出后, 系统被唤醒。若使能 TC2IE 及 EI, 则唤醒后进入中断, 若执行 DI, 则唤醒后执行下一条指令。

### 7.3.4 TC2 相关寄存器

#### 7.3.4.1 R1B8/TC2CON (TC2 控制寄存器)

0X1B8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CON	TC2EN	PWM5GATE<1:0>		TC2CKS	TC2PTEN	TC2PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC2EN - TC2/PWM567 计数器使能控制

- 0: 禁止
- 1: 使能

Bit<6:5>: PWM5GATE<1:0> - PWM5 门控输出使能控制位

PWM5GATE<1>	PWM5GATE<0>	PWM5 门控输出
0	0	无限制
0	1	PWM5 输出由比较器 CMP0 结果控制, 当 CMP0OUT=0 输出 PWM, 否则输出低电平
1	0	PWM5 输出由外部中断 EXINT0 控制, 当 EXINT0=1 输出 PWM, 否则输出低电平
1	1	禁止使用

Bit<4>: TC2CKS - TC2 时钟选择

- 0: 系统时钟
- 1: 低速时钟

Bit<3:0>: TC2PTEN、TC2PSR2~TC2PSR0 分频系数选择位:

TC2PTEN	TC2PSR<2>	TC2PSR<1>	TC2PSR<0>	TC2 分频系数
0	0	0	0	1:1
1	0	0	0	1:2



1	0	0	1	1:4
1	0	1	0	1:8
1	0	1	1	1:16
1	1	0	0	1:32
1	1	0	1	1:64
1	1	1	0	1:128
1	1	1	1	1:256

### 7.3.4.2 R1B9/TC2PRDL (TC2 低 8 位计数寄存器)

0X1B9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDL	TC2PRD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

### 7.3.4.3 R1BD/TC2PRDTH (TC2 高 4 位计数寄存器)

0X1BD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDTH	TC2PRD<11:8>				PWM7DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.4 TC3 定时计数器

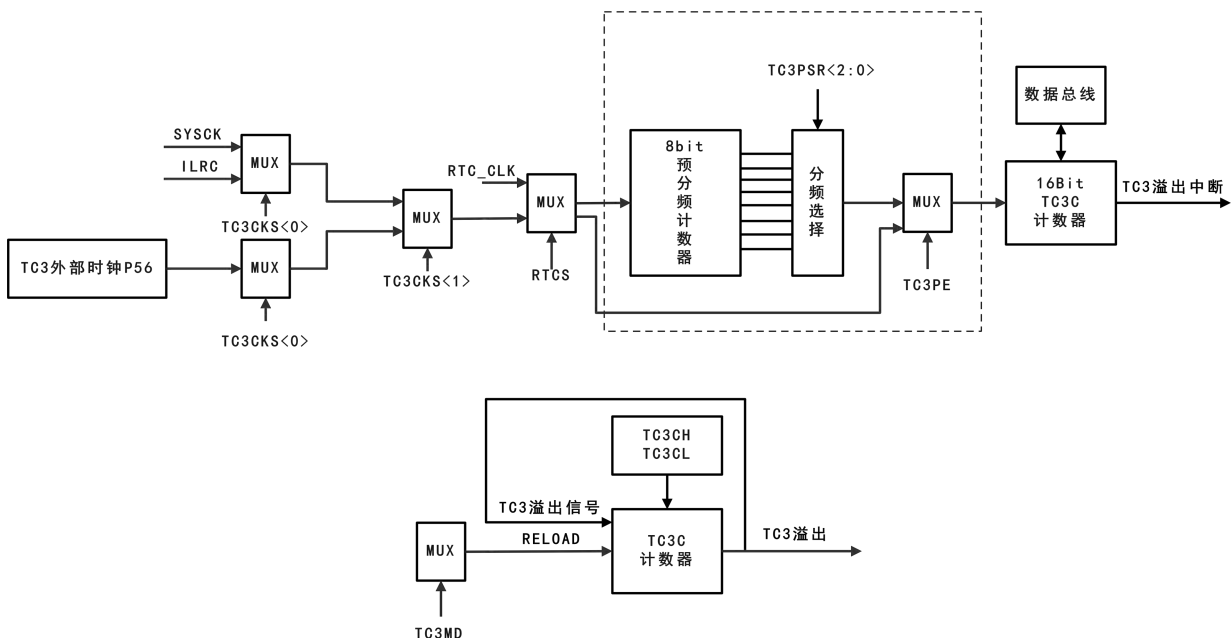
JZ8FT8801提供一个8位定时计数器作为TC3的预分频器，由TC3CON寄存器的TC3PE位决定预分频器的使能，TC3PSR<2:0>三位决定预分频比。

TC3 是一个 16Bit 上行计数器，只要有时钟就工作。时钟源既可以是系统时钟、ILRC 时钟，也可以选择外部时钟（由 P56 引脚输入，触发沿可选）或 RTC 晶振作为时钟源（默认四分频），时钟到来计数器实现加 1。需要注意的是如果用到 RTC 晶振功能不论是 TC0 还是 TC3 都需要使能 RTCEN，但是使能之后 TC3 就只能用作 RTC 功能，不受内部时钟控制。

TC3 计数溢出可以形成中断信号。

在 IDLE 模式下，TC3 中断可以唤醒电路，唤醒后可以选择进中断或者继续执行原程序。

在睡眠模式下，TC3 设置为 P56 外部输入时钟或 RTC 时钟可以中断溢出并唤醒电路。



TC3 结构框图

### 7.4.1 TC3 定时设置说明

- 1、给 TC3C 寄存器 TC3CL、TC3CH 赋初始值；
- 2、设置 TC3CON 寄存器的值（选择时钟源及预分频比）；
- 3、作为计数器使用，需要在 TC3CON 寄存器选择 TC3 信号源为外部输入信号；当选择信号源为 RTC 晶振时，需要在 TC3CON 寄存器选择 RTC 模式并使能 OPTION 中的 RTC\_EN；
- 4、若使用重载功能时，需要在 CAPCON 寄存器将 TC3MOD 设置为 0 自动重载模式，并在 TC3RL、TC3RH 设置重载值；

- 5、当配置都设置好之后，最后再设置 TC3CON 寄存器中的 TC3EN 对定时计数器使能；
- 6、若需要执行中断功能，须设置 INTE1 寄存器中的 TC3IE (Bit3) 为 1，并执行 EI 指令；
- 7、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC3 中断标志位；

## 7.4.2 TC3 定时计算说明

TC3 定时功能通过写值到 TC3C 寄存器，给定时器赋初始值，定时器从初始值位置开始累加，直至定时器溢出产生中断。如果不设重载，计数从 0 开始累加；如果设置重载，计数从重载值开始累加。

**TC3 定时时间计算公式（选择内部系统时钟）：**

$$\text{TC3 定时时间} = (1/\text{Fosc}) \times (\text{TC3 分频}) \times (65536 - \text{TC3 初始值})$$

示例：

Fosc=8MHz，TC3 分频选择=8 分频，TC3R=65436；

$$\text{TC3 定时时间} = (1/8) \times (8) \times (65536 - 65436) = 100 \text{ us}$$

**TC3 定时时间计算公式（选择外部输入时钟）：**

$$\text{TC3 定时时间} = (1/\text{外部输入时钟}) \times (\text{TC3 分频}) \times (65536 - \text{TC3 初始值})$$

示例：

外部输入时钟=16 MHz，TC3 分频选择=16 分频，TC3R=65436；

$$\text{TC3 定时时间} = (1/16) \times (16) \times (65536 - 65436) = 100 \text{ us}$$

## 7.4.3 TC3 空闲模式唤醒说明

在空闲模式下，系统指令停止执行，具有唤醒功能的模块和系统时钟正常工作，其它功能被禁止。

TC3 可以唤醒空闲模式，设置 R188/CPUCON 寄存器 Bit6 位 TC3WE 为 1，IDLE = 1 加上 SLEEP 指令系统进入空闲模式，TC3 定时器正常工作（选择系统时钟）。当 TC3 定时器溢出后，系统被唤醒。若使能 TC3IE 及 EI，则唤醒后进入中断，若执行 DI，则唤醒后执行下一条指令。重载模式下的唤醒和中断与 TC3 定时计数器一致。

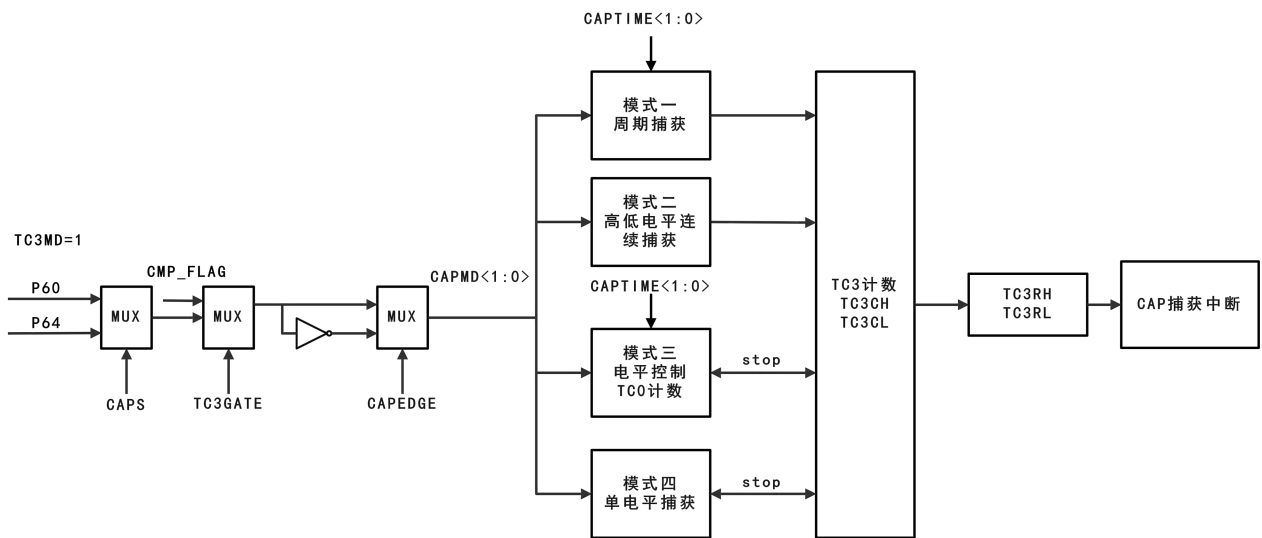
当时钟源选择 P56 外部时钟或 RTC 晶振时钟时，使能 TC3WE=1，直接执行 SLEEP 睡眠指令，也可以唤醒系统。

## 7.4.4 TC3 端口电平捕获

TC3 定时计数器具有捕获功能，捕获的信号源可以是 P60、P64 外部端口或者比较器的比较结果 CMP0UT，选择 TC3MD 模式选择位为捕获模式，复用 TC3 重载寄存器 TC3RL、TC3RH，用来保存捕获到的计数值。每组捕获源可以选择上升沿或者下降沿来设定捕获开始和计数。捕获共有四组模式可以选择，由 CAPCON 捕获控制寄存器中的 CAPMMD<1:0>进行选择，其中涉及到周期个数可选模式，计数值由 CAPIME<1:0>进行选择。

捕获可以产生中断信号，使能 INTE1 寄存器的 CAPIE (bit4)。

如果信号周期较长，需注意定时器计数器可能的溢出问题，通过适当设置定时器的工作模式和预分频值来匹配信号特性。



TC3 捕获模式结构框图

### 7.4.4.1 TC3 捕获模式设置说明

- 1、给 TC3C、TC3R 寄存器赋初始值；
- 2、设置 TC3CON 寄存器的值（选择时钟源及预分频比）；
- 3、作为计数器使用，需要在 TC3CON 寄存器选择 TC3 信号源为外部输入信号；当选择信号源为 RTC 晶振时，需要在 TC3CON 寄存器选择 RTC 模式并使能 OPTION 中的 RTC\_EN；
- 4、使用捕获模式时，需要在 CAPCON 寄存器将 TC3MOD 设置为 1 捕获模式，并设置 CAPEDG 捕获边沿选择位，设置 CAPS 捕获输入源或者 TC3GATE (R185)=1，选择捕获模式 CAPMD，根据捕获模式选择捕获计数选择位 CAPTIME；

- 5、当配置都设置好之后，设置 CAPEN 捕获使能，最后再设置 TC3CON 寄存器中的 TC3EN 对定时计数器使能；
- 6、若需要执行中断功能，须设置 INTE1 寄存器中的 CAPIE (Bit4) 为 1，并执行 EI 指令；
- 7、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 CAPIF 中断标志位；

## 7.4.4.2 TC3 捕获模式说明

Bit<3:2>: CAPMD<1:0> - 捕获模式选择位

CAPMD<1>	CAPMD<0>	捕获模式选择
0	0	模式一
0	1	模式二
1	0	模式三
1	1	模式四

捕获模式	捕获模式说明
模式一	周期捕获，周期个数可选 (1, 4, 8, 16)
模式二	高低电平连续捕获
模式三	电平控制 TC3 计数，电平个数可选 (1, 4, 8, 16)
模式四	单电平捕获

Bit<1:0>: CAPTIME<1:0> - 捕获计数选择位

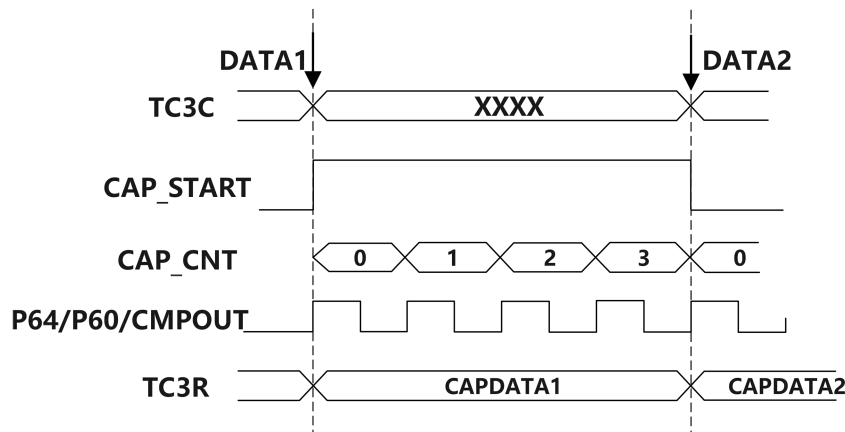
CAPTIME<1>	CAPTIME<0>	捕获计数值选择
0	0	1
0	1	4
1	0	8
1	1	16

### 模式一：周期捕获

周期捕获可以捕捉输入信号一个或者多个周期的时间。

如图所示的周期捕获模式，CAPCON (R1C7) = 0xD1，选择上升沿触发捕获开始，选择捕捉 4 个周期。当捕获源的第一个上升沿到来时，捕获第一个计数值 DATA1，并将 TC3 计数值记录到 TC3RL、TC3RH。当第五个上升沿到来时，捕获第二个计数值 DATA2，并将 TC3 计数值记录到 TC3RL、TC3RH，之后捕获停止，CAPEN 置位为 0。通过计算 DATA1、DATA2 的差值可以实现对捕获源信号的精确测量。TC3 计数器在此期间不停止计数。

如设置了捕获中断使能，在两次捕获发生时均可进入中断程序。



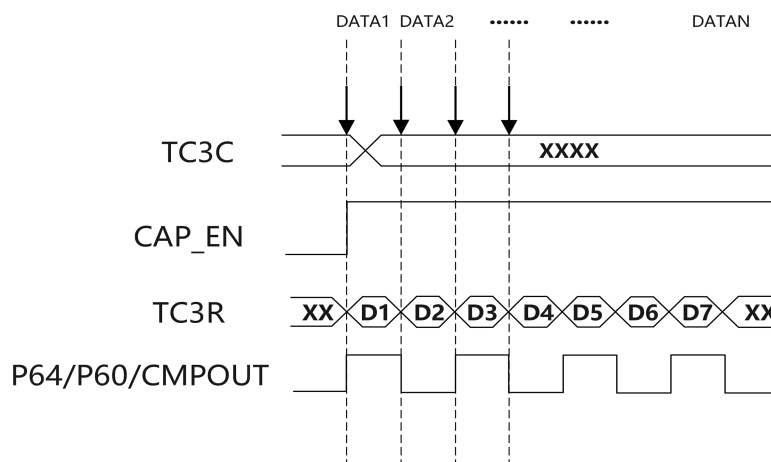
模式一：周期捕获

### 模式二：高低电平连续捕获

高低电平连续捕获可以监测并记录捕获源信号高低电平变化的时刻及持续时间。

如图所示的高低电平连续捕获模式，此时对 CAPEDGE 设置为 1 或 0 均可，TC3 正常计数，只要捕获信号源产生了高低电平变化，就会捕获当前计数值 DATA，并将该值记录到 TC3RL、TC3RH，TC3 计数器不停止计数。

如设置了捕获中断使能，在每次捕获发生时均可进入中断程序。



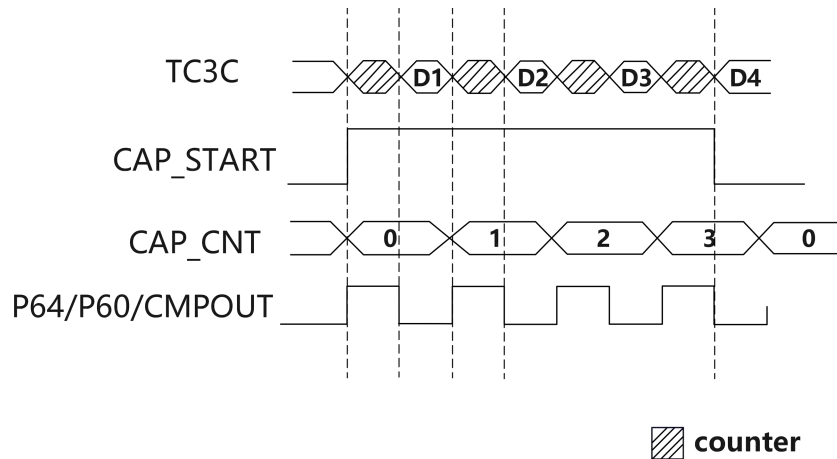
模式二：高低电平连续捕获

### 模式三：电平控制 TC3 计数

电平控制计数器计数可以在高电平或者低电平条件下控制 TC3 的计数。CAPEDG 可以选择高电平或者低电平使能计数。如下图所示，CAPCON=0xD9，在 4 个周期的高电平内计数，当高电平变化为低电平时，TC3 计数停止，计数器的计数值保持为 DATA<sub>n</sub>，在下一个高电平内，TC3 计数器在 DATA<sub>n</sub> 的基础上继续计数。使能计数的电平个数可以设置 CAPTIME<1:0>捕获计数选择位来确定，在相应的电平个数内 TC3 计数器停止计数，捕获使能会自动关闭。在电平计数

期间如遇到计数器溢出，使能中断唤醒可以正常进入中断或唤醒系统。若使能捕获中断 CAPIE，在最后一个电平计数结束时，可以进入捕获中断程序。

若更改 CAPEDGE 为 1，即使能低电平期间计数，高电平期间停止计数。

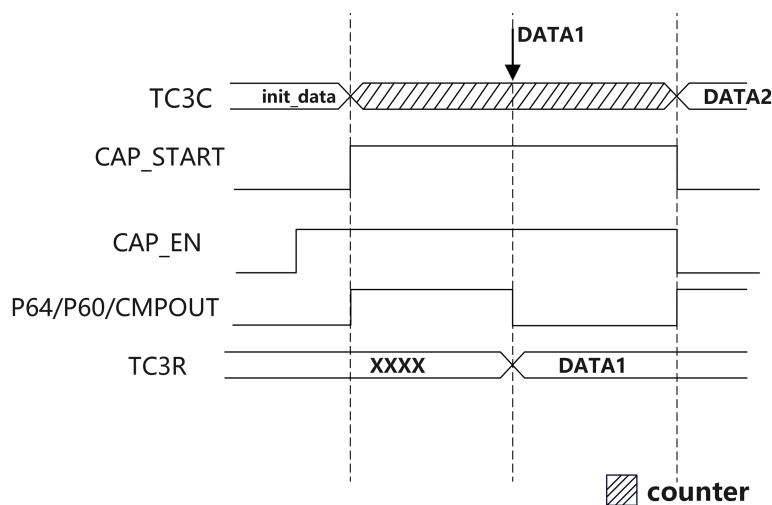


模式三：电平控制 TC3 计数

#### 模式四：单电平捕获

单电平捕获可以监测外部捕获源信号。下图中 CAPCON=0xDC，当信号达到并保持在预设的电平（比如高电平）时，TC3 计数器开始计时，直到该信号转换为另一个电平（变为低电平）时，此时计数值 DATA1 会保存在 TC3RL、TC3RH 中（TC3 计数器不停止），以此来确定该电平状态的持续时间。如果在计数期间遇到计数器溢出，使能中断唤醒可以正常进入中断或唤醒系统。若使能捕获中断 CAPIE，捕获产生时，可以进入捕获中断程序。

该模式下只会在捕获源信号的一个周期内进行，周期结束后 TC3 计数器停止计数，捕获使能也会自动关闭。若更改 CAPEDGE 为 1，低电平转变为高电平时产生捕获。



模式四：单电平捕获

## 7.4.5 TC3 相关寄存器

### 7.4.5.1 R1C2/TC3CON (TC3 控制寄存器)

0X1C2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3CON	TC3EN	RTCEN	TC3CKS<1:0>		TC3PTEN	TC3PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC3EN - TC3 定时计数使能位

0: 禁止 TC3

1: 使能 TC3

Bit<6>: RTCEN - RTC 模式选择位

0: 禁止

1: 使能

Bit<5:4>: TC3CKS<1:0> - TC3 信号源选择位

00: 系统时钟

01: ILRC 时钟 (低速振荡器)

10: 外部输入信号 (P56), 信号上升沿计数

11: 外部输入信号 (P56), 信号下降沿计数

Bit<3>: TC3PTEN - 预分频器使能位

0: 禁止 TC3 预分频

1: 使能 TC3 预分频

Bit<2:0>: TC3PSR<2:0> - TC3 预分频选择控制位:

TC3PSR<2>	TC3PSR<1>	TC3PSR<0>	TC3 分频系数
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

## 7.4.5.2 R1C3/TC3CL (TC3 低 8 位计数寄存器)

0X1C3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3CL	TC3C<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.4.5.3 R1C4/TC3CH (TC3 高 8 位计数寄存器)

0X1C4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3CH	TC3C<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.4.5.4 R1C5/TC3RL (TC3 重载低 8 位寄存器)

0X1C5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3RL	TC3R<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.4.5.5 R1C6/TC3RH (TC3 重载高 8 位寄存器)

0X1C6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3RH	TC3R<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.4.5.6 R1C7/CAPCON (捕获控制寄存器)

0X1C7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON	CAPEN	TC3MD	CAPEDGE	TC3CAPS	CAPMD<1:0>		CAPTIME<1:0>	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: CAPEN - 捕获使能位

0: 禁止 TC3 捕获功能

1: 使能 TC3 捕获功能

Bit<6>: TC3MD - TC3 模式选择位

- 0: 自动重载模式
- 1: 捕获模式

Bit<5>: CAPEDGE - 捕获边沿选择位

- 0: 上升沿触发捕获开始, 下降沿触发捕获计数
- 1: 下降沿触发捕获开始, 上升沿触发捕获计数

Bit<4>: TC3CAPS - TC3 捕获输入源选择位 (TC3GATE=1 时, TC3CAPS 选择无效)

- 0: P60
- 1: P64

Bit<3:2>: CAPMD<1:0> - 捕获模式选择位

CAPMD<1>	CAPMD<0>	捕获模式选择
0	0	模式一
0	1	模式二
1	0	模式三
1	1	模式四

捕获模式	捕获模式说明
模式一	周期捕获, 周期个数可选 (1, 4, 8, 16)
模式二	高低电平连续捕获
模式三	电平控制 TC3 计数, 电平个数可选 (1, 4, 8, 16)
模式四	单电平捕获

Bit<1:0>: CAPTIME<1:0> - 捕获计数选择位

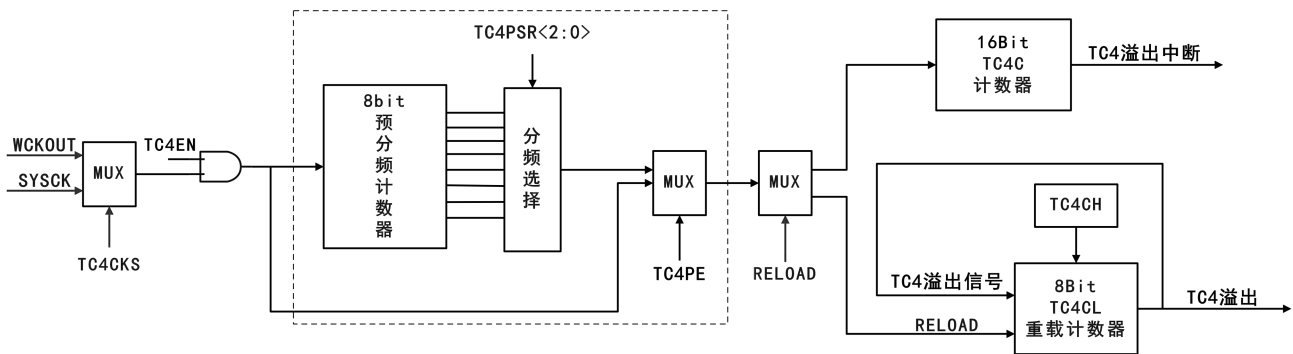
CAPTIME<1>	CAPTIME<0>	捕获计数值选择
0	0	1
0	1	4
1	0	8
1	1	16

## 7.5 TC4 定时计数器

TC4定时计数器提供一个8位预分频器，TC4CON寄存器的TC4PSR<2:0>三位决定预分频比。

TC4 是一个 8/16 Bit 上行计数器，由 TC4HEN 高 8 位计数器选择位进行控制。当选择 8bit 计数器功能时，计数器高位 TC4CH 不工作，仅计数器低位 TC4CL 在计数。当使能 RELOAD，即选择 TC4 为 8bit 可重载计数器，TC4CH 寄存器为重载赋值寄存器，此时 TC4HEN 置 0。TC4 定时器需使能 TC4EN 才能工作，时钟源是内部系统时钟或者低速时钟，每个时钟周期到来，计数器实现加 1。TC4 计数值计满产生溢出，若使能 TC4IE 及 EI，系统跳转到对应的中断向量地址，执行中断服务程序。

在 IDLE 模式下，TC4 中断可以唤醒电路（时钟选择系统时钟），唤醒后可以选择进中断或者继续执行原程序，方法参考端口输入变化唤醒。



TC4 结构框图

### 7.5.1 TC4 定时设置说明

- 1、给 TC4C 寄存器 TC4CL、TC4CH 赋初始值；
- 2、设置 TC4CON 寄存器的值（选择时钟源及预分频比），设置 TC4HEN=0 选择 TC4 作为 8bit 计数器功能，设置 TC4HEN=1 选择 TC4 作为 16 bit 计数器功能；
- 3、若选择 TC4 为 8bit 可重载计数器，设置 TC4CON 寄存器 RELOAD=1，设置 TC4CH 寄存器为重载赋值寄存器，此时 TC4HEN 置 0。
- 4、当配置都设置好之后，最后再设置 TC4CON 寄存器中的 TC4EN 对定时计数器使能；
- 5、若需要执行中断功能，须设置 INTE1 寄存器中的 TC4IE (Bit3) 为 1，并执行 EI 指令；
- 6、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC4 中断标志位；

## 7.5.2 TC4 定时计算说明

TC4 定时功能通过写值到 TC4CL、TC4CH 寄存器，定时器从初始值开始累加，直至定时器溢出产生中断。

TC4 定时时间计算公式（8bit 计数器模式）：

$$TC4 \text{ 定时时间} = (1/F_{osc}) \times (TC4 \text{ 分频}) \times (256 - TC4 \text{ 初始值})$$

示例：

$F_{osc}=8\text{MHz}$ ，TC4 分频选择=8 分频，TC4 初始值=100；

$$TC4 \text{ 定时时间} = (1/8) \times (8) \times (256 - 100) = 156 \text{ us}$$

TC4 定时时间计算公式（16bit 计数器模式）：

$$TC4 \text{ 定时时间} = (1/F_{osc}) \times (TC4 \text{ 分频}) \times (65536 - TC4 \text{ 初始值})$$

示例：

$F_{osc}=8\text{MHz}$ ，TC4 分频选择=8 分频，TC4 初始值=64936；

$$TC4 \text{ 定时时间} = (1/8) \times (8) \times (65536 - 64936) = 600\text{us}$$

## 7.5.3 TC4 空闲模式唤醒说明

TC4 可以唤醒空闲模式，使能 R185/WKCON 寄存器 Bit0 位 TC4WE，选择 TC4 时钟源为系统时钟。IDLE = 1 加上 SLEEP 指令系统进入空闲模式，TC4 定时器正常工作。当 TC4 定时器溢出后，系统被唤醒。若使能 TC4IE 及 EI，则唤醒后进入中断，若执行 DI，则唤醒后执行下一条指令。

当时钟源选择 ILRC 时，使能 TC4WE=1，直接执行 SLEEP 睡眠指令，也可以唤醒系统

## 7.5.4 TC4 相关寄存器

### 7.5.4.1 R1C8/TC4CON(TC4 控制寄存器)

0X1C8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC4CON	TC4EN	TC4CKS	TC4RELOAD	TC4HEN	TC4PTEN	TC4PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC4EN - TC4 定时计数使能位

0: 禁止 TC4

1: 使能 TC4

Bit<6>: TC4CKS - TC4 时钟源选择位

- 0: 系统时钟
- 1: 低速时钟

Bit<5>: TC4RELOAD - TC4 重载模式选择位

- 0: TC4 为 8/16 bits 计数器, 无重载功能
- 1: TC4 为 8 bits 可重载计数器, TC4CH 寄存器为重载赋值寄存器, 保持 TC4HEN=0

Bit<4>: TC4HEN - TC4 高 8 位计数器选择位

- 0: TC4 为 8 bits 计数器
- 1: TC4 为 16 bits 计数器

Bit<3>: TC4PTEN - 预分频器使能位

- 0: 禁止 TC4 预分频
- 1: 使能 TC4 预分频

Bit<2:0>: TC4PSR<2:0> - TC4 预分频选择控制位:

TC4PSR<2>	TC4PSR<1>	TC4PSR<0>	TC4 分频系数
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

#### 7.5.4.2 R1C9/TC4CL (TC4 低 8 位计数寄存器)

0X1C9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC4CL	TC4C<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

#### 7.5.4.3 R1CA/TC4CH (TC4 高 8 位计数寄存器)

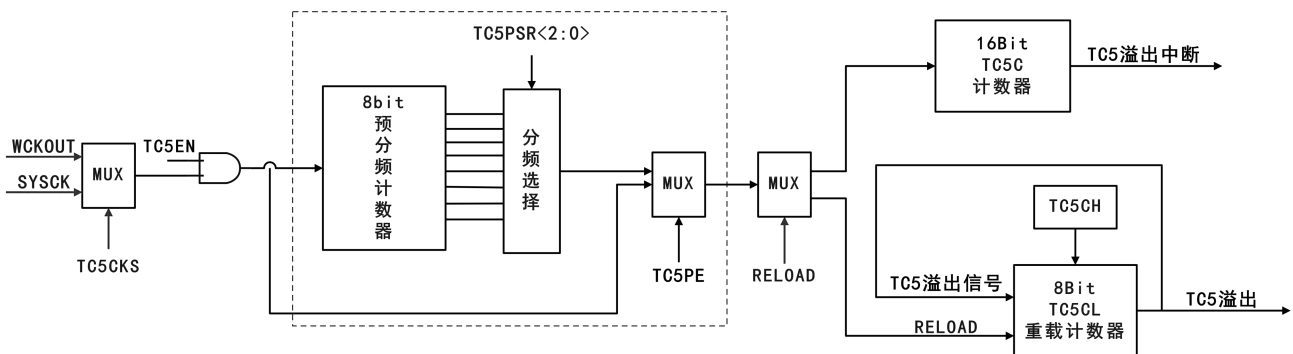
0X1CA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC4CH	TC4C<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 7.6 TC5 定时计数器

TC5定时计数器提供一个8位预分频器，TC4CON寄存器的TC4PSR<2:0>三位决定预分频比。

TC4 是一个 8/16 Bit 上行计数器，由 TC4HEN 高 8 位计数器选择位进行控制。当选择 8bit 计数器功能时，计数器高位 TC5CH 不工作，仅计数器低位 TC5CL 在计数。当使能 RELOAD，即选择 TC5 为 8bit 可重载计数器，TC5CH 寄存器为重载赋值寄存器，此时 TC5HEN 置 0。TC5 定时器需使能 TC5EN 才能工作，时钟源是内部系统时钟或者低速时钟，每个时钟周期到来，计数器实现加 1。TC5 计数值计满产生溢出，若使能 TC5IE 及 EI，系统跳转到对应的中断向量地址，执行中断服务程序。

在 IDLE 模式下，TC5 中断可以唤醒电路（时钟选择系统时钟），唤醒后可以选择进中断或者继续执行原程序，方法参考端口输入变化唤醒。



TC5 结构框图

### 7.6.1 TC5 定时设置说明

- 1、给 TC5C 寄存器 TC5CL、TC5CH 赋初始值；
- 2、设置 TC5CON 寄存器的值（选择时钟源及预分频比），设置 TC5HEN=0 选择 TC5 作为 8bit 计数器功能，设置 TC5HEN=1 选择 TC5 作为 16 bit 计数器功能；
- 3、若选择 TC5 为 8bit 可重载计数器，设置 TC5CON 寄存器 RELOAD=1，设置 TC5CH 寄存器为重载赋值寄存器，此时 TC5HEN 置 0。
- 4、当配置都设置好之后，最后再设置 TC5CON 寄存器中的 TC5EN 对定时计数器使能；
- 5、若需要执行中断功能，须设置 INTE1 寄存器中的 TC5IE (Bit3) 为 1，并执行 EI 指令；
- 6、系统执行中断服务程序将自动保存 ACC、STATUS 及 RSR 数据，执行 RETI 指令后，数据自动恢复，退出中断前要清除 TC5 中断标志位；

## 7.6.2 TC5 定时计算说明

TC5 定时功能通过写值到 TC5CL、TC5CH 寄存器，定时器从初始值开始累加，直至定时器溢出产生中断。

TC5 定时时间计算公式（8bit 计数器模式）：

$$\text{TC5 定时时间} = (1/F_{\text{osc}}) \times (\text{TC5 分频}) \times (256 - \text{TC5 初始值})$$

示例：

$F_{\text{osc}}=8\text{MHz}$ ，TC5 分频选择=8 分频，TC5 初始值=100；

$$\text{TC5 定时时间} = (1/8) \times (8) \times (256 - 100) = 156 \text{ us}$$

TC5 定时时间计算公式（16bit 计数器模式）：

$$\text{TC5 定时时间} = (1/F_{\text{osc}}) \times (\text{TC5 分频}) \times (65536 - \text{TC5 初始值})$$

示例：

$F_{\text{osc}}=8\text{MHz}$ ，TC5 分频选择=8 分频，TC5 初始值=64936；

$$\text{TC5 定时时间} = (1/8) \times (8) \times (65536 - 64936) = 600\text{us}$$

## 7.6.3 TC5 空闲模式唤醒说明

TC5 可以唤醒空闲模式，使能 R185/WKCON 寄存器 Bit0 位 TC5WE，选择 TC5 时钟源为系统时钟。IDLE = 1 加上 SLEEP 指令系统进入空闲模式，TC5 定时器正常工作。当 TC5 定时器溢出后，系统被唤醒。若使能 TC5IE 及 EI，则唤醒后进入中断，若执行 DI，则唤醒后执行下一条指令。

当时钟源选择 ILRC 时，使能 TC5WE=1，直接执行 SLEEP 睡眠指令，也可以唤醒系统。

## 7.6.4 TC5 相关寄存器

### 7.6.4.1 R1CB/TC5CON(TC5 控制寄存器)

0X1CB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC5CON	TC5EN	TC5CKS	TC5RELOAD	TC5HEN	TC5PTEN	TC5PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC5EN - TC5 定时计数使能位

0: 禁止 TC5

1: 使能 TC5

Bit<6>: TC5CKS - TC5 时钟源选择位

- 0: 系统时钟
- 1: 低速时钟

Bit<5>: TC5RELOAD - TC5 重载模式选择位

- 0: TC5 为 8/16 bits 计数器, 无重载功能
- 1: TC5 为 8 bits 可重载计数器, TC5CH 寄存器为重载赋值寄存器, 保持 TC5HEN=0

Bit<4>: TC5HEN - TC5 高 8 位计数器选择位

- 0: TC5 为 8 bits 计数器
- 1: TC5 为 16 bits 计数器

Bit<3>: TC5PTEN - 预分频器使能位

- 0: 禁止 TC5 预分频
- 1: 使能 TC5 预分频

Bit<2:0>: TC5PSR<2:0> - TC5 预分频选择控制位:

TC5PSR<2>	TC5PSR<1>	TC5PSR<0>	TC5 分频系数
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

#### 7.6.4.2 R1CC/TC5CL (TC5 低 8 位计数寄存器)

0X1CC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC5CL	TC5C<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

#### 7.6.4.3 R1CD/TC5CH (TC5 高 8 位计数寄存器)

0X1CD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC5CH	TC5C<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8 PWM 脉宽调制

JZ8FT8801 提供 2 组 12 bit TC1、TC2 计数器，两组计数器可分为共七路 PWM，PWM1-4 复用 TC1 计数器，PWM5-7 复用 TC2 计数器来产生脉宽调制信号，PWM 输出波形由周期及占空比决定，传输速率为周期倒数，每路 PWM 均支持取反。芯片提供死区互补输出，PWM1-3 可以通过直接互补取反映射到端口输出 IPWM1-3 与原来的 PWM1-3 作为死区互补信号；还可以通过 PWM6 和 PWM7 异或后作为 IPWM5，与 PWM5 作为死区互补信号。

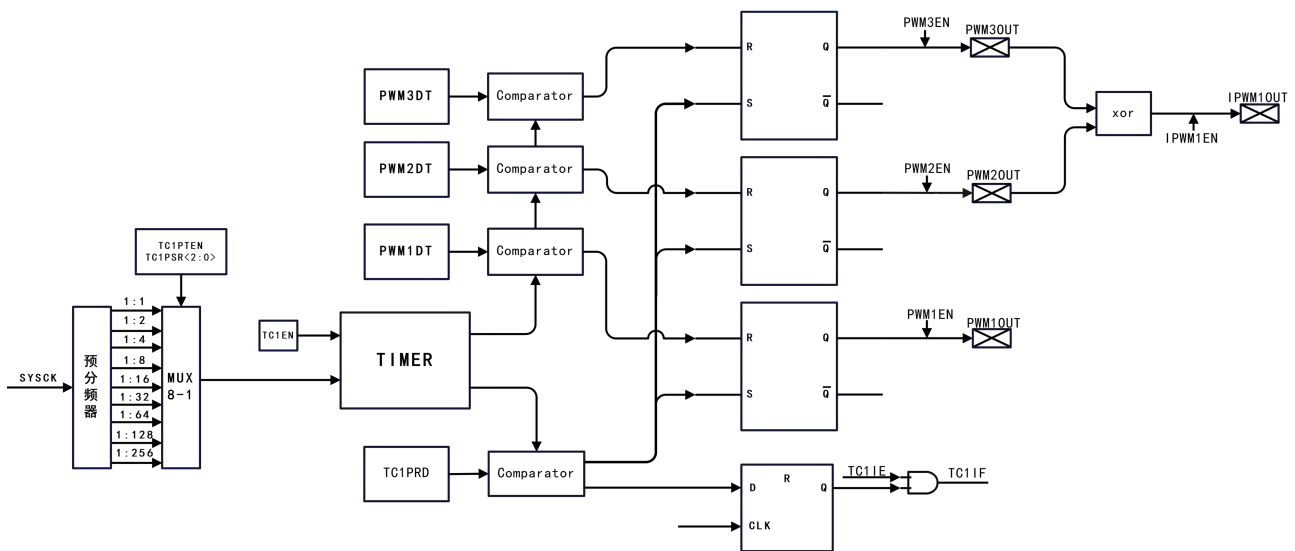
PWM 带有周期溢出中断，使能后可跳转到中断。

PWM1-3 支持刹车功能，输入源可电平控制刹车方式，可选择刹车输出模式为立即刹车或周期结束刹车，刹车结束后，手动清除标志位可重新开始输出 PWM。

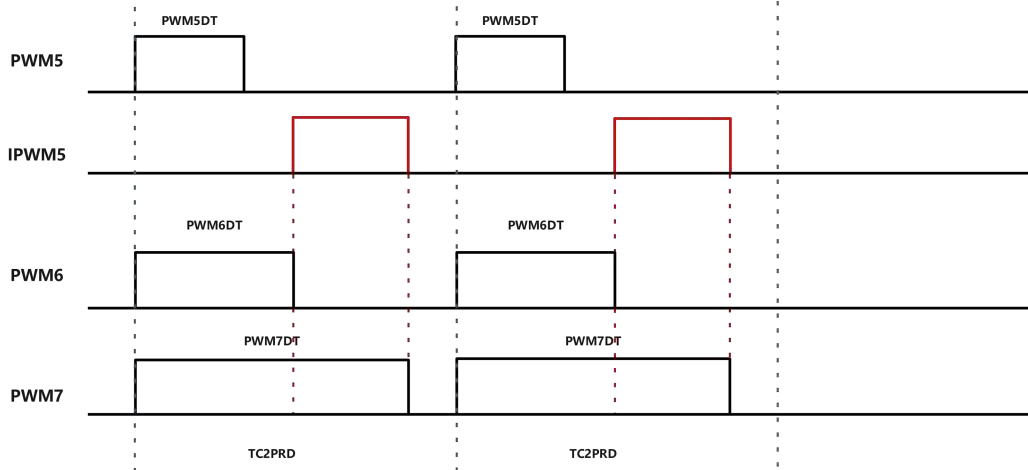
在 IDLE(空闲模式)下，TC1、TC2 在 CPUCON 寄存器中使能 TC1WE、TC2WE 可唤醒系统。

### 8.1 PWM 内部结构与时序

以下内部结构图仅供参考理解，并不代表实际电路。



PWM 工作结构电路



3 路共周期 PWM 与 IPWM5 时序说明

## 8.2 PWM 周期与占空比

PWM 提供一个带 8bit 可编程预分频的 12bit 时钟计数器，作为 PWM 模块的波特率时钟发生器。可通过使能 PWM 控制寄存器中的 TCxEN，使能计数器功能。通过 TCxPSR<2:0>控制位，可进行计数器的预分频设置。INVxH 和 INVxL 控制输出取反，PWMxE 和 IPWMxE 分别使能 PWM 功能和 PWM 互补功能。

PWM 周期通过写值到 TCxPRDL（低八位）和 TCxPRDTH（高四位）周期寄存器，当计数器的值与 PRD 值相等，在下一个递增周期发生如下事件：

- 计数器清零；
- 对应 PWM 输出引脚置高电平；
- 产生 PWM 周期溢出中断（若使能）；
- PWM 占空比由 PWMxDTL/TCxPRDTH 寄存器锁存到 DT/TIMER 比较寄存器；

**PWM 周期计算公式：**

$$\text{PWM 周期} = (\text{PRD}) \times (\text{PWM\_CK}) \times (\text{PWM 分频})$$

示例：

PWMxRD=100, Fosc=16MHz, PWMCKS=1, PWM 分频选择=16 分频；

$$\text{PWM 周期} = (100) \times (1/16) \times (16) = 100 \text{ us}$$

PWM 占空比通过写值到 PWM 占空比寄存器，当计数器的值溢出清零时，PWMxDT 的值被锁存到 DT/TIMER 比较寄存器。当 DT/TIMER 比较寄存器的值与 TIMER 计数器的值相等时，PWM 输出引脚置为低电平。PWMxDT 的值可以在任何时候被写入，但 DT/TIMER 比较寄存器的值只有在周期溢出时写入：

PWM 占空比计算公式:

$$\text{PWM 占空比} = (\text{DT}) \times (\text{PWM\_CK}) \times (\text{PWM 分频})$$

示例:

PWMxDT=50, Fosc=16MHz, PWMCKS=1, PWM 分频选择=16 分频;

$$\text{PWM 占空比} = (50) \times (1/16) \times (16) = 50 \text{ us}$$

## 8.3 PWM 空闲模式唤醒说明

在空闲模式下, CPU 指令停止执行, 具有唤醒功能的模块和系统时钟正常工作, 其它功能被禁止。

PWM 可以唤醒空闲模式, 设置 TC1/TC2WE 唤醒使能。设置对应的周期或占空比中断使能, 将 IDLE (R188) 置 1, 配合 SLEEP 指令系统进入空闲模式, PWM 定时器正常工作。

当满足对应中断条件, 系统被唤醒, 若使能 EI, 则唤醒后进入中断, 若使能 DI 则唤醒后执行下一条指令。

## 8.4 PWM5 比较器门控说明

PWM5 的门控功能由 TC2CON 寄存器的 PWM5GATE<1:0>进行控制。当 PWM5GATE<0>=1 时, PWM5 输出由比较器结果控制, CMP0OUT=0 时输出 PWM 波形。当 PWM4GATE<1>=1 时, PWM5 输出由外部中断 EXINT0 控制 (P50 或 P54), EXINT=1 时输出 PWM 波形。

## 8.5 IPWM 波形以及 Buzzer 设置说明

IPWM1、IPWM2、IPWM3 是直接 PWM1/2/3 对应取反输出到对应口上, 原 PWM 可继续输出, 使能 IPWM5 波形不是直接取反, 而是使能 IPWM5EN, 将 PWM6 与 PWM7 异或后进行输出。若此时使能 PWM6EN, 可以通过 PWM6 输出选择控制位选择从 P51/P62 输出; 使能 PWM6EN, 可以通过 PWM6 输出选择控制位选择从 P51/P62 输出。同时使能支持同时从两个端口输出 IPWM5 波形。

PWM 模块中还设计了蜂鸣器功能, 用户可通过 BZXEN (R1B7、R1BF) 控制位使能 PWM 输出蜂鸣器, 蜂鸣器可通过 PWM1S (R1B7) 与 PWM5S (R1BF) 进行端口选择, 分别从 P60/P76 口与 P52/P62 口输出, 需要注意的是, 使能蜂鸣器需将 PWM1EN 与 PWM5EN 清零并将对应输出口设置为输出。

## 8.6 PWM4DT 中断以及触发 ADC 说明

PWM4DT 提供占空比触发中断的功能，在使能 PWM4DTIE (R1D6) 后并执行 EI 指令可根据配置的 PWM7 占空比进入中断程序。

PWM 可选择通过 PWM4 占空比出发 ADC 转换使能，通过配置 ADCGATE (R1A7) 置 1，那么 ADC 转换将在 PWM4 占空比下降沿被触发。

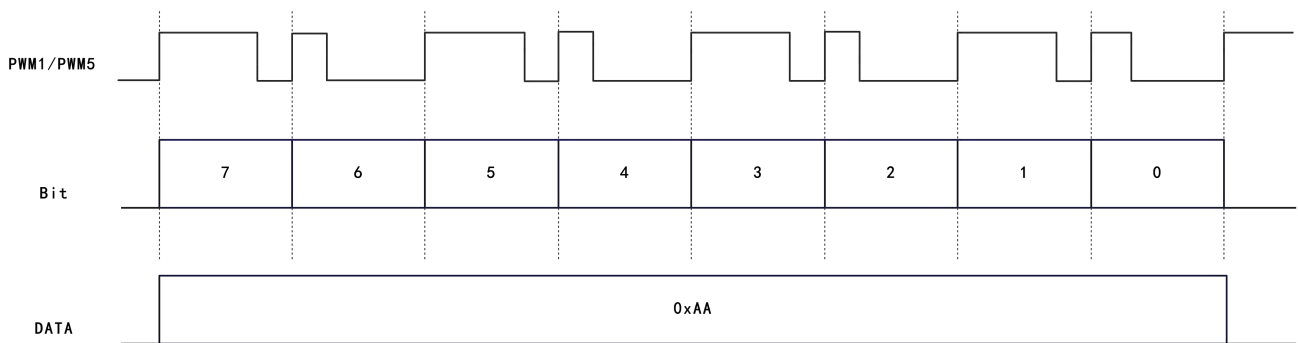
## 8.7 PWM 刹车中断功能

JZ8FT8801 支持 PWM 刹车中断功能，通过 PWMCON4 (R1D4) 寄存器进行配置，刹车源可选择 P72 口或者 CMP0OUT 来触发，在选择 IO 口进行触发时，还可通过配置 PWMST\_IOS 位选择高/低电平触发刹车，刹车模式可分为刹车后等待当前周期结束与立即刹车两种，通过 PWMST\_MODES 进行选择，刹车共支持 PWM1~3 三路，可通过各自的刹车使能位进行选择。

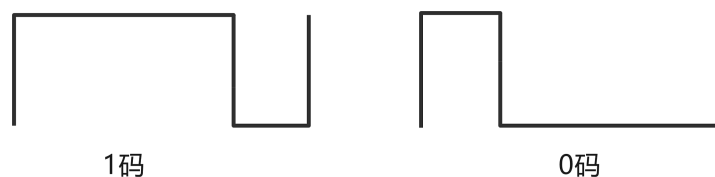
在选择 IO 口作为刹车触发源时，用户还可通过对 PWM1~3 三路输出口对应的数据寄存器进行配置，根据需求选择停留的高低电平。如 PWM1 输出选择了 P60 口进行输出，将 P6 数据寄存器 (R18B) 的 Bit0 位配置为 1，那么在刹车触发后 PWM1 将会保持高电平从 P60 口输出。

## 8.8 LED 单线级联

JZ8FT8801 支持 2 路单线级联 LED 驱动，通过 PWM1、PWM5 端口输出级联控制时序。级联 LED 驱动时序说明图如下：



单线级联LED时序说明



PWM1 和 PWM5 为两路独立的单线级联 LED 驱动控制，如有需要可以同时使能。

## 1. PWM1 作为 LED 驱动控制：

- 0 码型高电平时间：PWM1 占空比寄存器（PWM1DTL）设定；
- 1 码型高电平时间：PWM2 占空比寄存器（PWM2DTL）设定；
- 码元周期：TC1PRD 周期寄存器（TC1PRDL）设定；
- 级联数据：PWM3 占空比寄存器（PWM3DTL）设定；

在使能 PWM1\_LEDEN（R186）+TC1EN 后开始发送 8 位级联数据，数据由 PWM1 端口输出。8 位级联数据发送结束后 TC1EN 自动复位清零，重新置位 TC1EN 即开始下一个 8 位级联数据的发送。

## 2. PWM5 作为 LED 驱动控制：

- 0 码型高电平时间：PWM5 占空比寄存器（PWM5DTL）设定；
- 1 码型高电平时间：PWM6 占空比寄存器（PWM6DTL）设定；
- 码元周期：TC2PRD 周期寄存器（TC2PRDL）设定；
- 级联数据：PWM7 占空比寄存器（PWM7DTL）设定；

在使能 PWM5\_LEDEN（R186）+TC2EN 后开始发送 8 位级联数据，数据由 PWM5 端口输出。8 位级联数据发送结束后 TC2EN 自动复位清零，重新置位 TC2EN 即开始下一个 8 位级联数据的发送。

## 8.9 PWM 相关寄存器

### 8.9.1 R186/TBRDH(级联控制寄存器)

0X186	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDH	PWM5_LEDEN	PWM1_LEDEN	MTP_POWER	RBIT<12:8>				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: PWM5\_LEDEN - PWM5\_LED 级联控制位

0: 禁止

1: 使能

Bit<6>: PWM1\_LEDEN - PWM1 LED 级联控制位

0: 禁止

1: 使能

## 8.9.2 R1B0/TC1CON (TC1 控制寄存器)

0X1B0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1CON	TC1EN	TC21EN	-	TC1CKS	TC1PTEN	TC1PSR<2:0>		
读/写	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC1EN - TC1/PWM1234 计数器使能控制

(若设置 TC1PRD 寄存器, TC1 计数到预设值复位, TC1 从 1 开始计数)

0: 禁止

1: 使能

Bit<6>: TC21EN - TC1/PWM1234、TC2/PWM567 计数器同时使能控制位

0: 禁止

1: 使能

Bit<5>: 未使用

Bit<4>: TC1CKS - TC1 时钟选择

0: 系统时钟

1: 低速时钟

Bit<3:0>: TC1PTEN、TC1PSR2~TC1PSR0 分频系数选择位:

TC1PTEN	TC1PSR<2>	TC1PSR<1>	TC1PSR<0>	TC1 分频系数
0	0	0	0	1:1
1	0	0	0	1:2
1	0	0	1	1:4
1	0	1	0	1:8
1	0	1	1	1:16
1	1	0	0	1:32
1	1	0	1	1:64
1	1	1	0	1:128
1	1	1	1	1:256

## 8.9.3 R1B1/TC1PRDL (TC1/PWM1234 周期低 8 位寄存器)

0X1B1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRDL	TC1PRD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

TC1 (PWM1、PWM2、PWM3、PWM4) 周期低 8 位

## 8.9.4 R1B2/PWM1DTL (PWM1 占空比低 8 位寄存器)

0X1B2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DTL	PWM1DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.5 R1B3/PWM2DTL (PWM2 占空比低 8 位寄存器)

0X1B3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DTL	PWM2DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.6 R1B4/PWM3DTL (PWM3 占空比低 8 位寄存器)

0X1B4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DTL	PWM3DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.7 R1B5/TC1PRDTH (TC1/PWM1234 周期高 4 位及 PWM3 占空比高 4 位寄存器)

0X1B5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRDTH	TC1PRD<11:8>				PWM3DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

TC1 (PWM1、PWM2、PWM3、PWM4) 周期高 4 位及 PWM3 占空比高 4 位

## 8.9.8 R1B6/PWM21DTH (PWM21 占空比高 4 位寄存器)

0X1B6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM21DTH	PWM2DT<11:8>				PWM1DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

PWM2 占空比高 4 位, PWM1 占空比高 4 位



## 8.9.9 R1B7/PWMCON1 (PWM 控制寄存器 1)

0X1B7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON1	BZ1EN	PWM3S	PWM2S	PWM1S	IPWM1S	PWM3EN	PWM2EN	PWM1EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: BZ1EN - BZ1 波形输出使能控制位

0: 禁止

1: 输出 BZ1 波形 (需设置 BZ1EN=1, PWM1EN=0)

(设置 P60/P76 为输出才能输出 BZ1 波形)

Bit<6>: PWM3S - PWM3 输出选择控制位

0: PWM3 从 P64 输出

1: PWM3 从 P52 输出

Bit<5>: PWM2S - PWM2 输出选择控制位

0: PWM2 从 P62 输出

1: PWM2 从 P50 输出

Bit<4>: PWM1S - PWM 输出选择控制位

0: PWM1 从 P60 输出

1: PWM1 从 P76 输出

Bit<3>: IPWM1S - IPWM1 输出选择控制位

0: IPWM1 从 P61 输出

1: IPWM1 从 P75 输出

Bit<2>: PWM3EN - PWM3 使能控制位

0: 禁止

1: 输出 PWM3 波形使能

(需要设置对应端口为输出才能输出 PWM3 波形)

Bit<1>: PWM2EN - PWM2 使能控制位

0: 禁止

1: 输出 PWM2 波形使能

(需要设置对应端口为输出才能输出 PWM2 波形)

Bit<0>: PWM1EN - PWM1 使能控制位

0: 禁止

1: 输出 PWM1 波形使能

(需要设置对应端口为输出才能输出 PWM1 波形, 且 BZ1EN=0)

## 8.9.10 R1B8/TC2CON (TC2 控制寄存器)

0X1B8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CON	TC2EN	PWM5GATE<1:0>		TC2CKS	TC2PTEN	TC2PSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TC2EN - TC2/PWM567 计数器使能控制

- 0: 禁止
- 1: 使能

Bit<6:5>: PWM5GATE<1:0> - PWM5 门控输出使能控制位

PWM5GATE<1>	PWM5GATE<0>	PWM5 门控输出
0	0	无限制
0	1	PWM5 输出由比较器 CMP0 结果控制，当 CMP0OUT=0 输出 PWM，否则输出低电平
1	0	PWM5 输出由外部中断 EXINT0 控制，当 EXINT0=1 输出 PWM，否则输出低电平
1	1	禁止使用

Bit<4>: TC2CKS - TC2 时钟选择

- 0: 系统时钟
- 1: 低速时钟

Bit<3:0>: TC2PTEN、TC2PSR2~TC2PSR0 分频系数选择位:

TC2PTEN	TC2PSR<2>	TC2PSR<1>	TC2PSR<0>	TC2 分频系数
0	0	0	0	1:1
1	0	0	0	1:2
1	0	0	1	1:4
1	0	1	0	1:8
1	0	1	1	1:16
1	1	0	0	1:32
1	1	0	1	1:64
1	1	1	0	1:128
1	1	1	1	1:256

## 8.9.11 R1B9/TC2PRDL (TC2/PWM567 周期低 8 位寄存器)

0X1B9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDL	TC2PRD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

TC2 (PWM5、PWM6、PWM7) 周期低 8 位

## 8.9.12 R1BA/PWM5DTL (PWM5 占空比低 8 位寄存器)

0X1BA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DTL	PWM5DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.13 R1BB/PWM6DTL (PWM6 占空比低 8 位寄存器)

0X1BB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM6DTL	PWM6DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.14 R1BC/PWM7DTL (PWM7 占空比低 8 位寄存器)

0X1BC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7DTL	PWM7DT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 8.9.15 R1BD/TC2PRDTH (TC2/PWM567 周期高 4 位及 PWM7 占空比高 4 位寄存器)

0X1BD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDTH	TC2PRD <11:8>				PWM7DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

TC2 (PWM5、PWM6、PWM7) 周期高 4 位及 PWM7 占空比高 4 位



## 8.9.16 R1BE/PWM65DTH (PWM65 占空比高 4 位寄存器)

0X1BE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM65DTH	PWM6DT<11:8>				PWM5DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

PWM6 占空比高 4 位，PWM5 占空比高 4 位

## 8.9.17 R1BF/PWMCON2 (PWM 控制寄存器 2)

0X1BF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON2	BZ2EN	PWM7S	PWM6S	PWM5S	IPWM5EN	PWM7EN	PWM6EN	PWM5EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: BZ2EN - BZ2 波形输出使能控制位

0: 禁止

1: 输出 BZ2 波形 (需设置 BZ2EN=1, PWM5EN=0)

(设置 P53/P64 为输出才能输出 BZ2 波形)

Bit<6>: PWM7S - PWM7 输出选择控制位

0: PWM7 从 P66 输出

1: PWM7 从 P51 输出

Bit<5>: PWM6S - PWM6 输出选择控制位

0: PWM6 从 P65 输出

1: PWM6 从 P52 输出

Bit<4>: PWM5S - PWM5 输出选择控制位

0: PWM5 从 P64 输出

1: PWM5 从 P53 输出

Bit<3>: IPWM5EN - IPWM5 使能控制位

0: 禁止

1: 输出 PWM6 与 PWM7 异或

(根据 PWM6EN (P65/P52) 或 PWM7EN (P66/P51) 选择输出端口)

Bit<2>: PWM7EN - PWM7 使能控制位

0: 禁止

1: 输出 PWM7 波形使能

(需要设置对应端口为输出才能输出 PWM7 波形)

Bit<1>: PWM6EN - PWM6 使能控制位

- 0: 禁止
- 1: 输出 PWM6 波形使能  
(需要设置对应端口为输出才能输出 PWM6 波形)

Bit<0>: PWM5EN - PWM5 使能控制位

- 0: 禁止
- 1: 输出 PWM5 波形使能  
(需要设置对应端口为输出才能输出 PWM5 波形, 且 BZ2EN=0)

### 8.9.18 R1CE/PWM4DTL (PWM4 占空比低 8 位寄存器)

0X1CE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DTL	PWM4DTL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

### 8.9.19 R1CF/PWM4CON (PWM4 控制及占空比高位寄存器)

0X1CF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4CON	IPWM2INV	IPWM1INV	PWM4S	PWM4EN	PWM4DTH<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: IPWM2INV - IPWM2 输出取反使能位

- 0: 禁止
- 1: 使能

Bit<6>: IPWM1INV - IPWM1 输出取反使能位

- 0: 禁止
- 1: 使能

Bit<5>: PWM4S - PWM4 输出选择控制位

- 0: PWM4 从 P67 输出
- 1: PWM4 从 P50 输出

Bit<4>: PWM4EN - PWM4 使能控制位

- 0: 禁止
- 1: 输出 PWM4 波形使能  
(需要设置对应端口为输出才能输出 PWM4 波形)

## 8.9.20 R1D0/PWMCON3 (PWM 控制寄存器 3)

0X1D0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON3	PWM4_PULSE_EN	PWM4INV	PWM7INV	PWM6INV	PWM5INV	PWM3INV	PWM2INV	PWM1INV
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: PWM4\_PULSE\_EN - PWM4 占空比单次触发使能位

0: 禁止

1: 使能

Bit<6>: PWM4INV - PWM4 输出取反使能位

0: 禁止

1: 使能

Bit<5>: PWM7INV - PWM7 输出取反使能位

0: 禁止

1: 使能

Bit<4>: PWM6INV - PWM6 输出取反使能位

0: 禁止

1: 使能

Bit<3>: PWM5INV - PWM5 输出取反使能位

0: 禁止

1: 使能

Bit<2>: PWM3INV - PWM3 输出取反使能位

0: 禁止

1: 使能

Bit<1>: PWM2INV - PWM2 输出取反使能位

0: 禁止

1: 使能

Bit<0>: PWM1INV - PWM1 输出取反使能位

0: 禁止

1: 使能

## 8.9.21 R1D1/PWMCON4 (PWM 控制寄存器 4)

0X1D1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON4	IPWM3INV	PWMBF	PWMBMS	PWMBIOS	PWMBSS	PWMB3EN	PWMB2EN	PWMB1EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit<7>: IPWM3INV - IPWM3 输出取反使能位  
 0: 禁止  
 1: 使能
- Bit<6>: PWMBF - PWM 刹车标志位  
 0: 无刹车  
 1: 刹车状态 (清零后 PWM 继续运行)
- Bit<5>: PWMBMS - PWM 刹车模式选择  
 0: 刹车后等待当前周期结束  
 1: 立即刹车
- Bit<4>: PWMBIOS - IO 作为刹车源时刹车电平选择  
 0: 低电平刹车  
 1: 高电平刹车
- Bit<3>: PWMBSS - PWM 刹车源选择位  
 0: IO 端口 P72  
 1: 比较器 0 输出结果 CMP0OUT
- Bit<2>: PWMB3EN - PWM3 刹车使能位  
 0: 禁止  
 1: 使能
- Bit<1>: PWMB2EN - PWM2 刹车使能位  
 0: 禁止  
 1: 使能
- Bit<0>: PWMB1EN - PWM1 刹车使能位  
 0: 禁止  
 1: 使能

## 8.9.22 R1D2/PWMCON5 (PWM 控制寄存器 5)

0X1D2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON5	IPWM3S	IPWM2S	DEAD3EN	DEAD2EN	DEAD1EN	IPWM3EN	IPWM2EN	IPWM1EN
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit<7>: IPWM3S-IPWM3 输出选择控制位  
 0: IPWM3 从 P65 输出  
 1: IPWM3 从 P51 输出
- Bit<6>: IPWM2S-IPWM2 输出选择控制位  
 0: IPWM2 从 P63 输出  
 1: IPWM2 从 P77 输出

Bit<5>: DEAD3EN - PWM3 死区使能位

0: 禁止

1: 使能

Bit<4>: DEAD2EN - PWM2 死区使能位

0: 禁止

1: 使能

Bit<3>: DEAD1EN - PWM1 死区使能位

0: 禁止

1: 使能

Bit<2>: IPWM3EN - IPWM3 (死区) 输出使能位

0: 禁止

1: 使能

Bit<1>: IPWM2EN - IPWM2 (死区) 输出使能位

0: 禁止

1: 使能

Bit<0>: IPWM1EN - IPWM1 (死区) 输出使能位

0: 禁止

1: 使能

(如果只打开 IPWMxEN 就是 IPWM 输出;如果需要死区互补输出,则 DEADxEN/IPWMxEN 都打开)

### 8.9.23 R1D3/PWMDEAD (PWM 死区控制寄存器)

0X1D3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDEAD	PWMDEAD<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7: 0>: PWMDEAD<7:0> - PWM1、PWM2、PWM3 的死区时间设置

## 9 DAC 数模转换

### 9.1 DACVREF 电压说明

芯片内置 DAC 模块，通过置高 DACON(R1A5)的 DACEN 使能 DAC 模块，DACVREF 通过 VREFCAL(R1AC)寄存器调整输出电压，电压范围从 VBG (1.2V) 到 4.9V (理论值，实际有差异)。DAC 电压值可以从 P57 口输出，也可以通过 ADC 或比较器检测 DAC 电压值。

### 9.2 DAC 相关寄存器

#### 9.2.1 R1A5/DACON(DAC 控制寄存器)

0X1A5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DACON	DACEN	DACOUT	VCMP5	-	ADAT<11:8>			
读/写	R/W	R/W	R/W	R	R	R	R	R
复位值	0	0	0	0	X	X	X	X

Bit<7>: DACEN - DAC 模块使能控制位

- 0: 禁止
- 1: 使能

Bit<6>: DACOUT - DAC 电压输出控制位

- 0: 输出禁止 (P57)
- 1: 输出使能 (P57)

#### 9.2.2 R1AC/VREFCAL(VREF 微调寄存器)

0X1AC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VREFCAL	VREFCAL<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

DACVREF 微调，数据参考节 [21.7. DAC 电压参数](#)

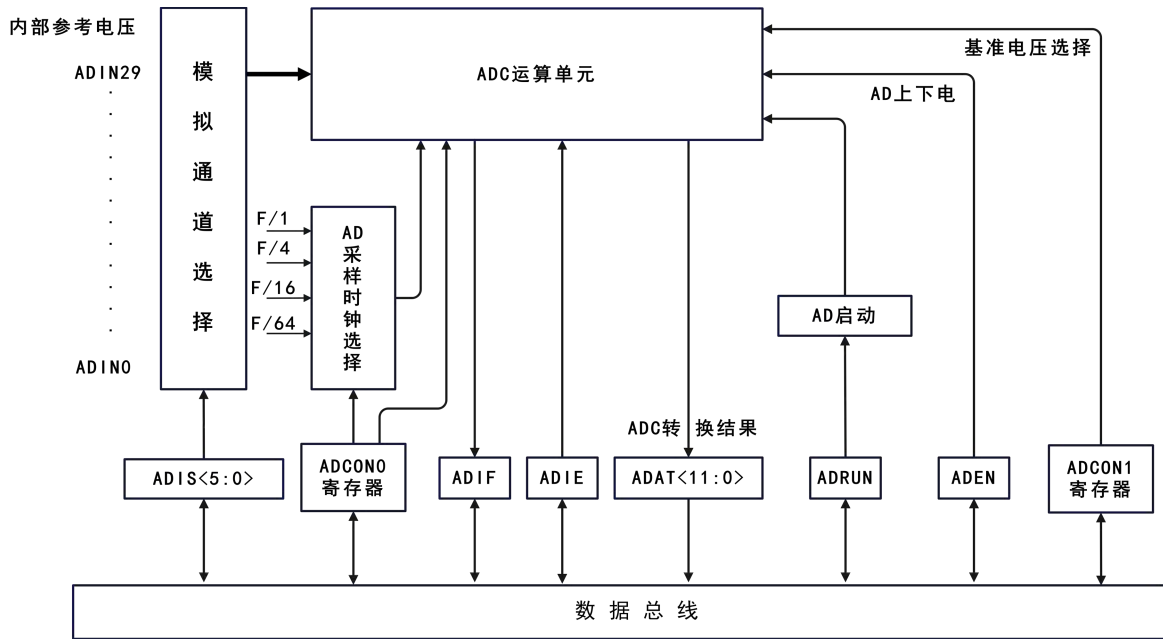
## 10 ADC 模数转换

JZ8FT8801 ADC 模块提供一个 12 位精度的 SAR AD 转换器，包含 30 路模拟通道。

ADC 转换器采用逐次逼近式模数转换，结果存储到 [R1A3/ADATH\(ADC 数据高 8 位寄存器\)](#)，[R1A4/ADATL\(ADC 数据低 8 位寄存器\)](#)，[R1A5/DACON\(DAC 控制寄存器\)](#) 结果寄存器中，通过 ADCON0、ADCON1 控制寄存器设置通道选择、内外部参考电压源、采样时钟、ADC 启动及 ADC。

AD 转换完成可以进入中断，也可以用查询 R1A7/ADCON1 的 ADRUN 位来判断。

如果启动了 AD 唤醒功能，AD 转换的完成可以从睡眠模式或者空闲模式下唤醒。



AD 转换功能示意图

### AD 转换时间：

AD 转换 1 个 bit 的时间定义为 TAD，一个完整的 12bit-ADC 转换需要 16 个 TAD 周期。必须严格按照 TAD 时间规范才能保证正常采样并得到正确结果。不同基准电压和不同 VDD 时所需要设置的时间也不一样，可参照以下表格设置合理分频。

基准电压/V	工作电压/V	最快分频设置 Fosc=16MHz	转换时间/us
VDD	4.0~5.5	Fosc/16	16
	2.7~4.0	Fosc/64	64
4	4.0~5.5	Fosc/16	16
	2.7~4.0	Fosc/64	64
3	4.0~5.5	Fosc/16	16
	2.7~4.0	Fosc/64	64
2	4.0~5.5	Fosc/16	16
	2.7~4.0	Fosc/64	64

AD 采样电压值计算：

$$\text{ADC 采样电压} = \frac{\text{采样值}}{4096} * \text{参考电压}$$

例：参考电压为内部基准 3V，采样值为 0x800=2048

$$\text{ADC 采样电压} = \frac{2048}{4096} * 3 = 1.5\text{V}$$

## 10.1 ADC 检测电源电压说明

通过配置 ADCCON1 (R1A6) 寄存器，通道选择位 ADIS<5:0>=011110 选择 1/4VDD 作为采样通道，EXTVS 选择分压源 VDD，VREF<2:0>选择参考电压，通过 ADC 转换得到转换结果。可以推出 VDD 的电压值：

$$\text{转换结果} = \left( \frac{\text{VDD}}{4} / \text{VREF} \right) * 4096$$

则  $\text{VDD} = (\text{转换结果}/4096) * 4 * \text{VREF}$

## 10.2 P51 端口检测 1/4 外部电压说明

JZ8FT8801 内置一路采集外部电压的 1/4 分压电阻通道，可以从 P51 端口输入一个高压，通过分压电阻后的值应小于 ADC 基准电压，此时可以正常采样，通过计算后能出此高压电压值：

$$\text{P51 输入电压} = \frac{\text{采样值}}{4096} * \text{参考电压} * 4$$

例：VDD=5V，参考电压为内部基准 4V，采样值为 0x800=2048

$$\text{P51 输入电压} = \frac{2048}{4096} * 4 * 4 = 8\text{V}$$

## 10.3 ADC 模数转换设置说明

- 1、ADC 输入端口使能及模拟通道选择，设置 R1A0/P5ADE、R1A1/P6ADE、R1A2/P7AE、R19A/P8AE、R1A6/ADCON1 寄存器；
- 2、ADC 参考电压、时钟分频，设置 R1A5/ADCON0、R1A6/ADCON1 寄存器；
- 3、如果需要用到中断功能，设置 ADIE=1，执行“EI”指令；
- 4、置“ADEN=1”开始 AD 供电电压；
- 5、根据需要选择 IDLE/SLEEP 模式，设置 AD 唤醒功能，ADCWE=1；
- 6、置“ADRUN=1”开始 AD 转换；
- 7、等待中断或 ADRUN 被清 0，如果 AD 中断发生，则离开中断程序时需将 ADIF 清 0；
- 8、保存转换的结果。如果需要做多次 AD 转换，跳到步骤 6；

## 10.4 ADC 相关寄存器

### 10.4.1 R1A3/ADATH(ADC 数据高 8 位寄存器)

0X1A3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADATH	ADAT<11:4>							
读/写	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

Bit<7:0>: ADAT<11:4> - AD 转换结果高 8 位

### 10.4.2 R1A4/ADATL(ADC 数据低 8 位寄存器)

0X1A4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADATL	ADAT<7:0>							
读/写	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

Bit<7:0>: ADAT<7:0> - AD 转换结果低 8 位

### 10.4.3 R1A5/DACON(DAC 控制寄存器)

0X1A5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DACON	DACEN	DACOUT	VCMP5	-	ADAT<11:8>			
读/写	R/W	R/W	R/W	R	R	R	R	R
复位值	0	0	0	0	X	X	X	X

Bit<5>: VCMP5-ADC 比较电压源选择

0: 选择 VREF

1: 选择 VDD

Bit<4>: 未使用

Bit<3:0>: ADAT<11:8>-AD 转换结果高 4 位

### 10.4.4 R1A6/ADCON0(ADC 控制寄存器 0)

0X1A6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADIS<5:0>						EXTVS	ADCAP
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:2>: ADIS<5:0>-AD 输入口选择

ADIS<5>	ADIS<4>	ADIS<3>	ADIS<2>	ADIS<1>	ADIS<0>	通道选择
0	0	0	0	0	0	AIN0/P50
0	0	0	0	0	1	AIN1/P51
0	0	0	0	1	0	AIN2/P52
0	0	0	0	1	1	AIN3/P53
0	0	0	1	0	0	AIN4/P54
0	0	0	1	0	1	AIN5/P55
0	0	0	1	1	0	AIN6/P56
0	0	0	1	1	1	AIN7/P57
0	0	1	0	0	0	AIN8/P60
0	0	1	0	0	1	AIN9/P61
0	0	1	0	1	0	AIN10/P62
0	0	1	0	1	1	AIN11/P63
0	0	1	1	0	0	AIN12/P64
0	0	1	1	0	1	AIN13/P65
0	0	1	1	1	0	AIN14/P66
0	0	1	1	1	1	AIN15/P67
0	1	0	0	0	0	AIN16/P70
0	1	0	0	0	1	AIN17/P71
0	1	0	0	1	0	AIN18/P72
0	1	0	0	1	1	AIN19/P73
0	1	0	1	0	0	AIN20/P74
0	1	0	1	0	1	AIN21/P75
0	1	0	1	1	0	AIN22/P76
0	1	0	1	1	1	AIN23/P77
0	1	1	0	0	0	AIN24/P80
0	1	1	0	0	1	AIN25/P81
0	1	1	0	1	0	AIN26/P82
0	1	1	0	1	1	AIN27/P83
0	1	1	1	0	0	AIN28/P84
0	1	1	1	0	1	AIN29/P85
0	1	1	1	1	0	0.25* (VDD 或 P51)
0	1	1	1	1	1	DACVREF
1	0	0	0	0	0	OPA
1	0	0	0	0	1	TEMP
1	0	0	0	1	0	VREF (VDD 作基准)
1	0	0	0	1	1	GND

Bit<1>: EXTVS - 采样通道过 1/4 电阻分压源选择

0: 选择 VDD 作为 1/4 电阻分压源

1: 选择 P51 作为 1/4 电阻分压源

Bit<0>: ADCAP - ADC 采样通道电容选择控制位

- 0: 禁止
- 1: 使能

## 10.4.5 R1A7/ADCON1 (ADC 控制寄存器 1)

0X1A7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADRUN	ADEN	ADPSR<1:0>		ADCGATE	VREF<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit<7>: ADRUN - ADC 开始供电采样

- 0: ADC 无采样或采样结束
- 1: ADC 开始采样

Bit<6>: ADEN - ADC 模块使能

- 0: ADC 禁止 (睡眠模式下关闭 ADC, 否则有功耗)
- 1: ADC 使能

Bit<5:4>: ADPSR<1:0> - ADC 时钟分频选择 (Fosc 为系统时钟频率)

ADPSR<1>	ADPSR<0>	ADC 时钟
0	0	Fosc/16
0	1	Fosc/4
1	0	Fosc/64
1	1	Fosc/1

Bit<3>: ADCGATE - ADC 门控选择位

- 0: 门控禁止, 由软件 ADCEN 使能
- 1: 门控使能, 由 PWM4 占空比下降沿触发 ADC 采样使能

Bit<2:0>: VREF<2:0>-参考电压选择

VREF<2>	VREF<1>	VREF<0>	参考电压
0	0	0	VBG 1.2V
0	0	1	基准 2V
0	1	0	基准 3V
0	1	1	基准 4V
1	0	0	EXVREF
1	X	1	VDD

## 11 CMP 比较器

JZ8FT8801 内置 CMP0、CMP1、CMP2 共 3 个硬件比较器。

### 11.1 CMP0 比较器

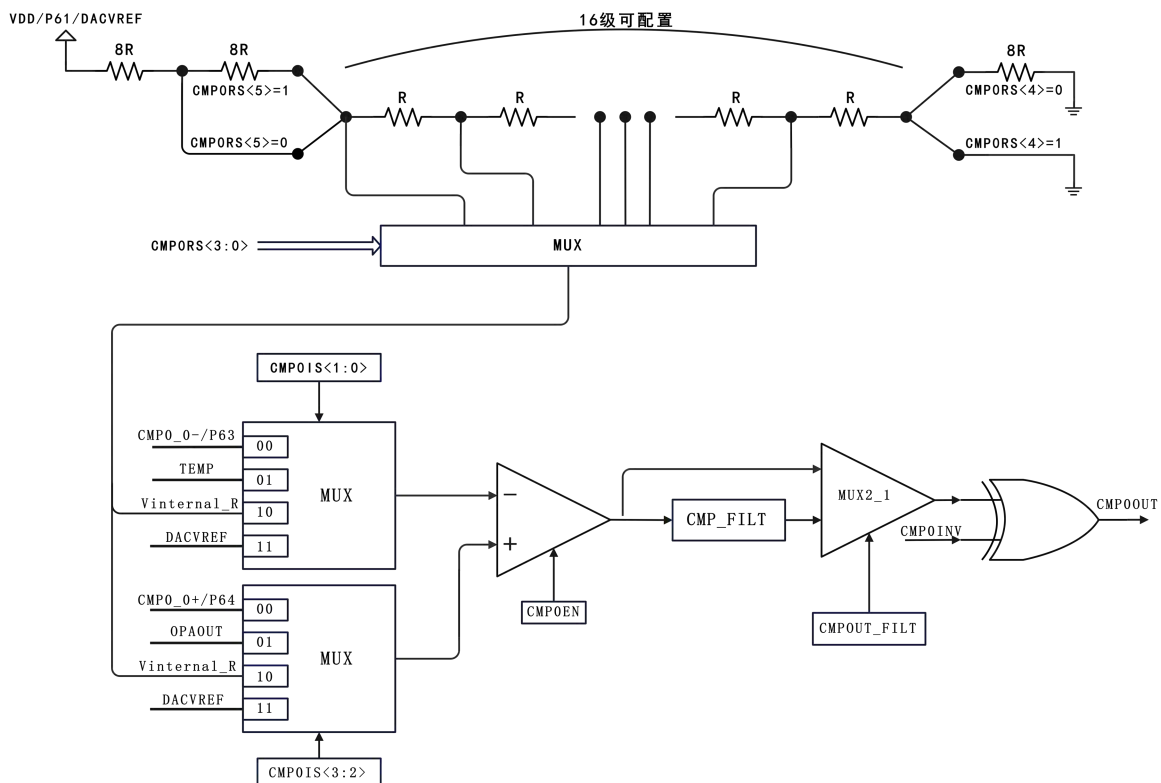
CMP0 可以从比较器输入引脚、电阻分压  $V_{\text{internal R}}$  与 DACVREF 三者任选其二进行比较，选择其中一个作为正输入，选择其中另一个作为负输入。比较器的负输入可以是 P63、TEMP、分压电阻（由 CMPRS<5:0>选择）、DACVREF。比较器的正输入可以是 P64、OPAOUT、分压电阻（由 CMPRS<5:0>选择）、DACVREF。分压电阻可通过 CMPCON 控制寄存器配置选择输入电压源为 P61 口输入、VDD 或 DACVREF。（同一端口不可同时作为正极和负极输入，分压电阻输入源也不能同时作为正极和负极）

比较器的输出结果可通过 CMPOCON1 (1FA) 寄存器 CMPOEO 位选择 P62 口输出，支持输出取反。芯片支持输出结果数字滤波，可在 OPTION 中选择。

比较器支持正负极交换进行比较，可通过 CMPOCON1 寄存器 CMP0INV 位进行使能正负极交换。

比较器支持输出结果变化触发中断，使能 CMPIE 可产生中断信号。

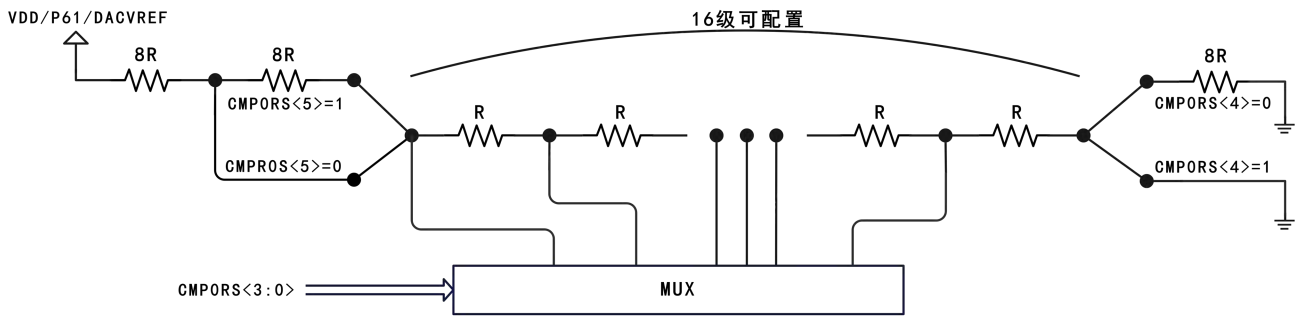
比较器支持输出结果变化唤醒，使能 CMPWE 可唤醒空闲模式和睡眠模式。在空闲模式和睡眠模式下，如果使能 CMPEN，会使能低速振荡器，睡眠功耗会变高。



比较器原理图说明

## 11.1.1 分压电阻输出电压 $V_{\text{internal R}}$

分压电阻输出电压  $V_{\text{internal R}}$  由一连串电阻所组成，可以产生不同层次的参考电压。  
 $\text{CMPORS}\langle 5:0 \rangle$  共同决定  $V_{\text{internal R}}$  的大小， $\text{CMPORS}\langle 5 \rangle$  和  $\text{CMPRS}\langle 4 \rangle$  选择  $V_{\text{internal R}}$  的最高和最低值， $\text{CMPORS}\langle 3:0 \rangle$  用于选择所要的电压水平，是由  $V_{\text{internal R}}$  的最高和最低值平均分成 16 等分，由  $\text{CMPORS}\langle 3:0 \rangle$  选择出来。



比较器修调示意图

例 1:  $\text{CMPORS}\langle 5 \rangle = 0$ ,  $\text{CMPORS}\langle 4 \rangle = 0$

取值范围:  $V_{\text{internal R}} = (3/4) * VDD \sim (1/4) * VDD + (1/32) * VDD$

计算公式:  $V_{\text{internal R}} = (1/4) * VDD + \frac{(n+1)}{32} * VDD$

例 2:  $\text{CMPORS}\langle 5 \rangle = 0$ ,  $\text{CMPORS}\langle 4 \rangle = 1$

取值范围:  $V_{\text{internal R}} = (2/3) * VDD \sim (1/24) * VDD$

计算公式:  $V_{\text{internal R}} = \frac{(n+1)}{24} * VDD$

例 3:  $\text{CMPORS}\langle 5 \rangle = 1$ ,  $\text{CMPORS}\langle 4 \rangle = 0$

取值范围:  $V_{\text{internal R}} = (3/5) * VDD \sim (1/5) * VDD + (1/40) * VDD$

计算公式:  $V_{\text{internal R}} = (1/5) * VDD + \frac{(n+1)}{40} * VDD$

例 4:  $\text{CMPORS}\langle 5 \rangle = 1$ ,  $\text{CMPORS}\langle 4 \rangle = 1$

取值范围:  $V_{\text{internal R}} = (1/2) * VDD \sim (1/32) * VDD$

计算公式:  $V_{\text{internal R}} = \frac{(n+1)}{32} * VDD$

$n = \text{CMPORS}\langle 3:0 \rangle$

## 11.1.2 比较器配置

使用比较器时需使能 CMPOEN（此时会打开用于对输出结果 CMPOUT 滤波的 ILRC，功耗会增加），设置 CMPOIS<5:0>选择正负极输入源，将正负极的端口设置为输入，同一端口不可同时作为正极和负极输入，分压电阻输入源也不能同时作为正极和负极。

若选择了电阻分压作为输入源，根据需要设置寄存器 CMPORS<5:0>，选择所需的分压电阻输出电压  $V_{\text{internal R}}$  进行比对。

例 1:

选择 P63 作为负端输入和  $V_{\text{internal R}}$  的电压为  $(18/32)*VDD$  作为正输入， $V_{\text{internal R}}$  选择 CMPORS<5>=0, CMPORS<4>=0 的配置方式，CMPORS<3:0>=4b' 1001 (n=9) 以得到

$$V_{\text{internal R}} = (1/4)*VDD + [(9+1)/32]*VDD = (18/32)*VDD$$

当配置完成后，使能 CMPOCON0(R1F9)控制寄存器的 Bit7 位 CMPOEN，比较器开始对比，并可通过配置 CMPOCON1 寄存器中 CMPOE0 位选择将比较结果从 P62 输出来（P62 口需设为输出），也可通过配置 CMPOCON1 寄存器的 Bit7 位 CMP0INV 将输出结果取反。

## 11.1.3 DACVREF 使用方法

内置 DACVREF 参考电压可以提供 VBG 到 4.9V 的电压，它可以测量外部电源电压水平。具体实现可以通过选择 DACVREF 参考电压做比较器的负输入去和正输入  $V_{\text{internal R}}$  比较。将  $V_{\text{internal R}}$  的电源设置为 VDD，利用调整  $V_{\text{internal R}}$  电压水平和 DACVREF 电压比较，就可以知道 VDD 的电压。如果 n (CMPRS[3: 0]十进制) 是让  $V_{\text{internal R}}$  最接近 1.2V (DAC 最小值)，以 3.12.1 中的四种情况为例，VDD 的电压就可以通过下列公式计算：

$$\text{For using Case 1: } VDD = [32/(n+9)] * 1.2V;$$

$$\text{For using Case 2: } VDD = [24/(n+1)] * 1.2V;$$

$$\text{For using Case 3: } VDD = [40/(n+9)] * 1.2V;$$

$$\text{For using Case 4: } VDD = [32/(n+1)] * 1.2V;$$

## 11.1.4 RFC 电阻频率转换

JZ8FT8801 内置 RFC 电阻频率转换功能，通过使用 CMP0 的比较结果控制 TC0 捕获，实现对外部电压变化的时间计数。

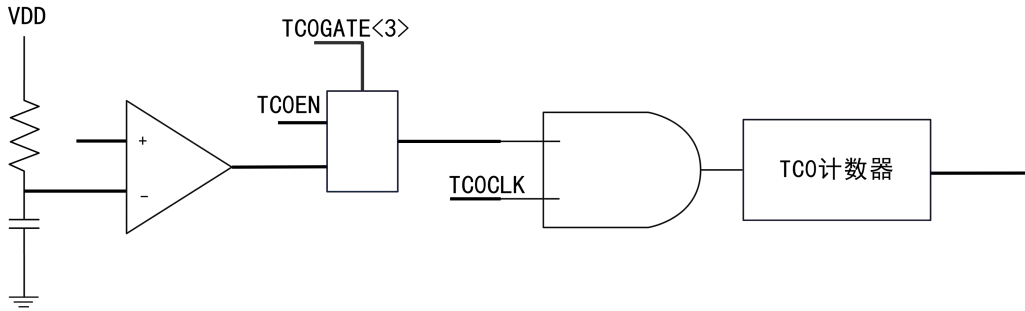
### RFC 原理:

RFC 端口电压的变化影响比较器的输出结果。当使能 R1FC 寄存器的 CMPOGATE 位后，TC0 计数由比较器的输出结果 CMPOOUT 控制。RFC 端口电压低于比较器正极电压时，CMPOOUT=1 时 TC0 开始计数；RFC 端口电压高于比较器正极电压时，CMPOOUT=0 时 TC0 停止计数，TC0 计数值记录了 RFC 端口电压充电到预设电压值（比较器正极电压）的时间。

### 应用步骤:

- 1、设置 PxAE 寄存器，选择比较器的模拟端口；

- 2、设置 CMP0CON0、CMP0CON1 寄存器，设定比较器的正极输入源为固定电压，设定比较器的负极输入源为外部端口（以图示为例），设置 CMP1CON1 打开比较器门控；
- 3、设置比较器负极输入源的端口输出低电平，对端口电容进行放电；
- 4、清零 TCO 定时器初始值；
- 5、设置比较器负极输入源的端口为模拟输入口，开始采样 RFC 端口电压；
- 6、读取 TCO 计数值，计算 RFC 端口充电时间；



RFC 结构说明图

## 11.1.5 CMP0 相关寄存器

### 11.1.5.1 R185/WKCON (CMP0 门控寄存器)

0X185	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WKCON	ADCGATE1	TC3GATE	TKWE	CMP2WE	CMP1WE	UART1WE	TC5WE	TC4WE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: ADCGATE1 - CMP0 结果变化触发 ADC 门控使能位

0: 禁止

1: 使能

Bit<6>: TC3GATE - CMP0OUT 门控 TC3 使能位

0: 禁止

1: 使能

### 11.1.5.2 R1F9/CMP0CON0 (CMP0 控制寄存器 0)

0X1F9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMP0CON0	CMPOEN	CMPOOUT	CMPORS<5:0>					
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

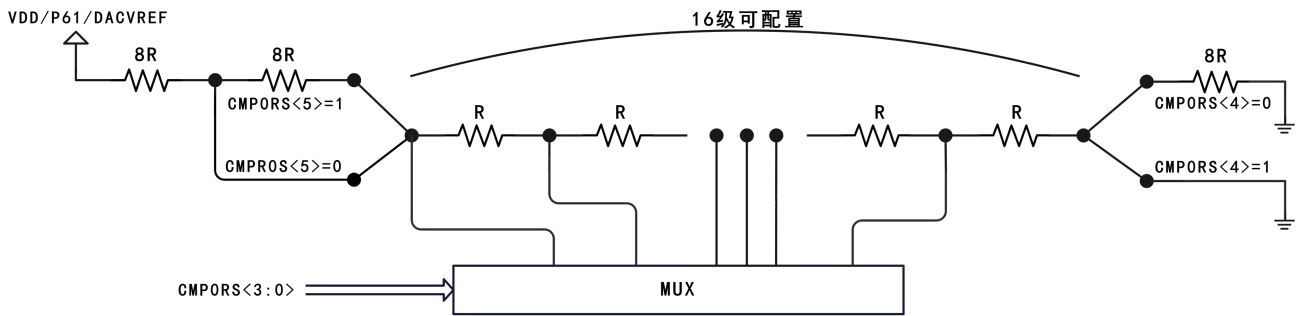
Bit<7>: CMPOEN - CMP0 使能控制位

- 0: 禁止
- 1: 使能

Bit<6>: CMP0OUT - CMP0 结果输出位

- 0: 比较器 0 正极电压小于负极电压
- 1: 比较器 0 正极电压大于负极电压

Bit<5:0>: CMPORS<5:0>-比较器修调位



当 CMPOCON1 寄存器=0x0B, 比较器正极选择电阻分压, 比较器负极选择 DACVREF, VREFCAL=0x00(1.21V), 不同电阻分压档位 CMPRS<5:0>对应不同的检测电压, 具体如下表:

CMPORS	检测电压/V	CMPORS	检测电压/V	CMPORS	检测电压/V	CMPORS	检测电压/V
00H	4.30	10H	29.04	20H	5.38	30H	38.72
01H	3.87	11H	14.52	21H	4.84	31H	19.36
02H	3.52	12H	9.68	22H	4.40	32H	12.91
03H	3.23	13H	7.26	23H	4.03	33H	9.68
04H	2.98	14H	5.81	24H	3.72	34H	7.74
05H	2.77	15H	4.84	25H	3.46	35H	6.45
06H	2.58	16H	4.15	26H	3.23	36H	5.53
07H	2.42	17H	3.63	27H	3.03	37H	4.84
08H	2.28	18H	3.23	28H	2.85	38H	4.30
09H	2.15	19H	2.90	29H	2.69	39H	3.87
0AH	2.04	1AH	2.64	2AH	2.55	3AH	3.52
0BH	1.94	1BH	2.42	2BH	2.42	3BH	3.23
0CH	1.84	1CH	2.23	2CH	2.30	3CH	2.98
0DH	1.76	1DH	2.07	2DH	2.20	3DH	2.77
0EH	1.68	1EH	1.94	2EH	2.10	3EH	2.58
0FH	1.61	1FH	1.82	2FH	2.02	3FH	2.42

注: 灰色部分检测电压禁止使用

## 11.1.5.3 R1FA/CMP0CON1 (CMP0 控制寄存器 1)

0X1FA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMP0CON1	CMP0INV	CMPOOE	CMP0IS<5:0>					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: CMP0INV - CMP0 正负极交换

0: 禁止

1: 使能

Bit<6>: CMPOOE - 比较器 CMP0 结果输出使能

0: P62 作为 GPIO

1: 比较器结果从 P62 输出

Bit<5:4>: CMP0IS<5:4> - 分压电阻输入电压源选择

CMP0IS<5>	CMP0IS<4>	分压电阻输入源
0	0	VDD
0	1	P61
1	0	DACVREF

Bit<3:2>: CMP0IS<3:2> - 比较器正极输入源选择

CMP0IS<3>	CMP0IS<2>	正极输入源
0	0	CMP0_0+/P64
0	1	OPAOUT
1	0	电阻分压
1	1	DACVREF

Bit<1:0>: CMP0IS<1:0> - 比较器负极输入源选择

CMP0IS<1>	CMP0IS<0>	负极输入源
0	0	CMP0_0-/P63
0	1	TEMP (温度传感器)
1	0	电阻分压
1	1	DACVREF

## 11.1.5.4 R1FC/CMP1CON1 (CMP0 门控控制寄存器)

0X1FC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMP1CON1	CMPOFILT	CMPOGATE	CMP1PS	CMPOPS	CMP1FILT	CMP1INV	CMP1OE	CMP1ZXD
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: CMP0FILT - 比较器 CMP0 结果输出内部滤波选择

- 0: CMP0 比较结果滤波关闭
- 1: CMP0 比较结果滤波使能

Bit<6>: CMPGATE - CMP0OUT 作为 TCO 的门控

- 0: 禁止
- 1: 使能

Bit<4>: CMPOPS - CMP0 比较器边缘触发控制位

- 0: CMPOUT0 上升沿触发
- 1: CMPOUT0 下降沿触发

## 11.2 CMP1 比较器

CMP1 可以比较器输入引脚、虚拟中心点进行比较。比较器负输入可以是 P70、虚拟中心点、虚拟中心点输出 P70。比较器正输入可以是 P65、P66、P67。

### 11.2.1 CMP1 相关寄存器

#### 11.2.1.1 R1FB/CMP1CON0 (CMP1 控制寄存器 0)

0X1FB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
<b>CMP1CON0</b>	CMP1EN	CMP1OUT	CMP1VPS<1:0>		CMP1VNS<1:0>		CMP1_HYS<1:0>	
读/写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: CMP1EN - CMP1 使能控制位

- 0: 禁止
- 1: 使能

Bit<6>: CMP1OUT - CMP1 结果输出位

- 0: 比较器 1 正极电压小于负极电压
- 1: 比较器 1 正极电压大于负极电压

Bit<5:4>: CMP1VPS<1:0> - 比较器 CMP1 正极输入源选择

CMP1VPS<1>	CMP1VPS<0>	正极输入源
0	0	CMP1_0+/P65
0	1	CMP1_1+/P66
1	X	CMP1_2+/P67

Bit<3:2>: CMP1VNS<1:0> - 比较器 CMP1 负极输入源选择

CMP1VNS<1>	CMP1VNS<0>	负极输入源
0	1	CMP1_0-/P70



1	0	虚拟中心点
1	1	虚拟中心点输出 P70

Bit<1:0>: CMP1\_HYS<1:0> - 比较器 CMP1 迟滞档位选择 (档位待确认)

CMP1_HYS<1>	CMP1_HYS<0>	迟滞档位选择
0	0	无迟滞
0	1	迟滞选择 20mv
1	0	迟滞选择 40mv
1	1	迟滞选择 60mv

## 11.2.1.2 R1FC/CMP1CON1 (CMP1 控制寄存器 1)

0X1FC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMP1CON1	CMPOFILT	CMPOGATE	CMP1PS	CMPOPS	CMP1FILT	CMP1INV	CMP1OE	CMP1ZXD
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<5>: CMP1PS - CMP1 比较器边缘触发控制位

- 0: CMP1OUT 上升沿触发
- 1: CMP1OUT 下降沿触发

Bit<3>: CMP1FILT - 比较器 CMP1 结果输出内部滤波选择

- 0: CMP1 比较结果滤波关闭
- 1: CMP1 比较结果滤波使能

Bit<2>: CMP1INV - CMP1 正负极交换

- 0: 禁止
- 1: 使能

Bit<1>: CMP1OE - 比较器 CMP1 结果输出使能

- 0: P71 作为 GPIO
- 1: 比较器结果从 P71 输出

Bit<0>: CMP1ZXD - 比较器 CMP1 虚拟中心点

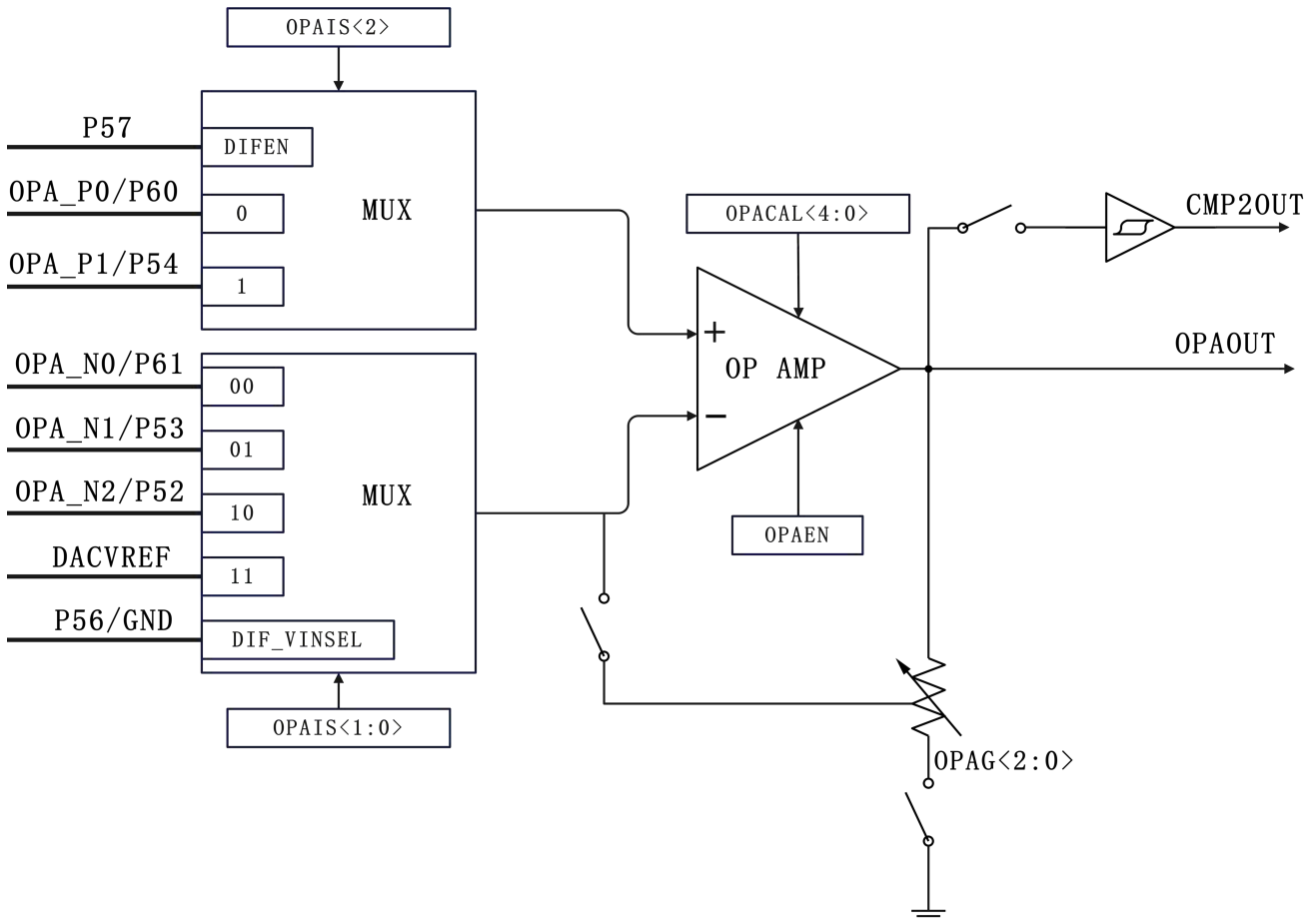
## 11.3 CMP2 比较器

CMP2 是复用的 OPA 寄存器，使用方法以及寄存器见 [OPA 运算放大器-模式 4](#)。

## 12 OPA 运算放大器

### 12.1 OPA 运算放大器概述

运算放大器 (Operational Amplifier) 是一种差分放大器，具有高输入电阻、低输出电阻、高开放增益（开环增益），主要作用与对输入信号进行放大和运算。需要注意的是，输入电压不能超过  $(VDD-1)V$ ，且放大的输出电压也必须小于  $VDD$ 。



OPA 系统框图

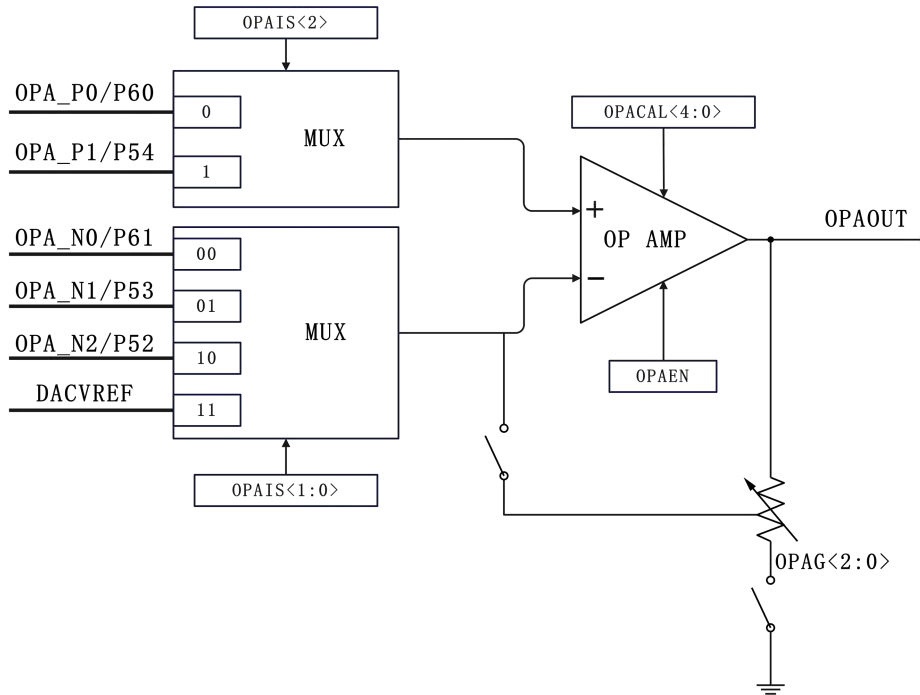
### 12.2 OPA 运算放大器工作模式

JZ8FT8801 的运算放大器主要提供了以下四种工作模式：

模式一：普通运放，内置电阻放大模式

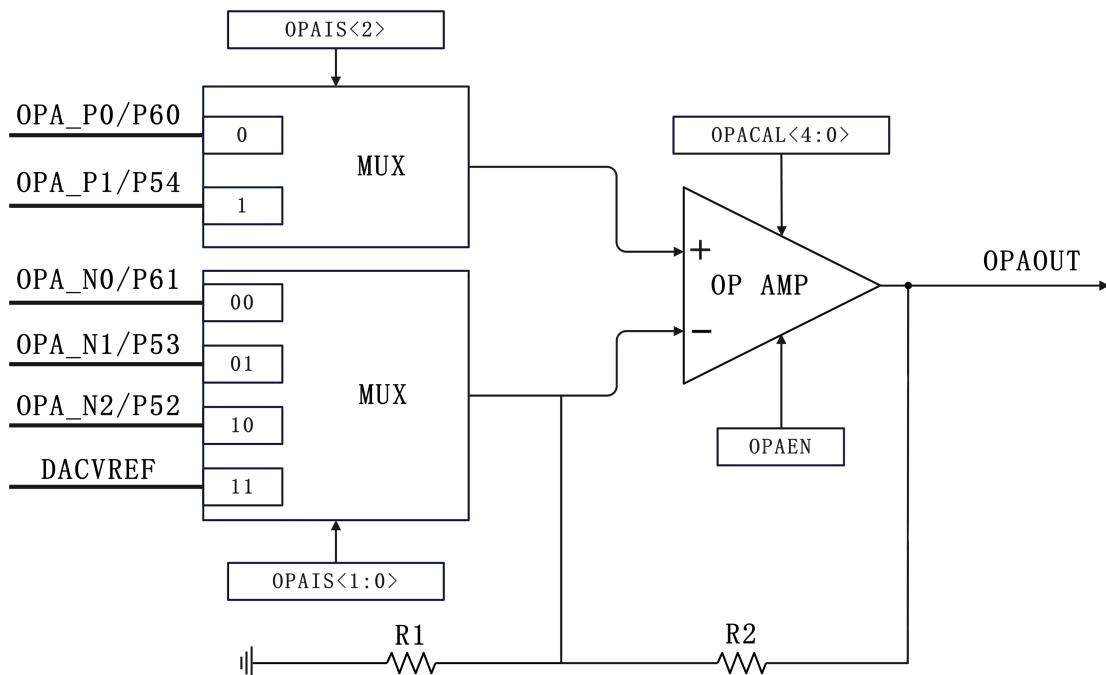
首先将 OPACON0 (R1A8) 寄存器中的 OPAIS<2:0> 正负极端口进行配置，当 OPAEN 置 1，使能运算放大器后，正负端口将连接至 IO 口；对 OPACON1 (R1A9) 寄存器中的 OPAMS 置高选择内置电阻模式，根据需求配置 OPAG<2:0> 选择放大倍数，可通过 OPACON2 (R1AA) 寄存器中的

VOFSIGN 位以及 OPACAL<4:0>对运放失调电压进行修调。OPA 负端接到内部电阻，正端输入外部电压，输出电压为正端电压的放大倍数。



## 模式二：普通运放，外置电阻放大模式

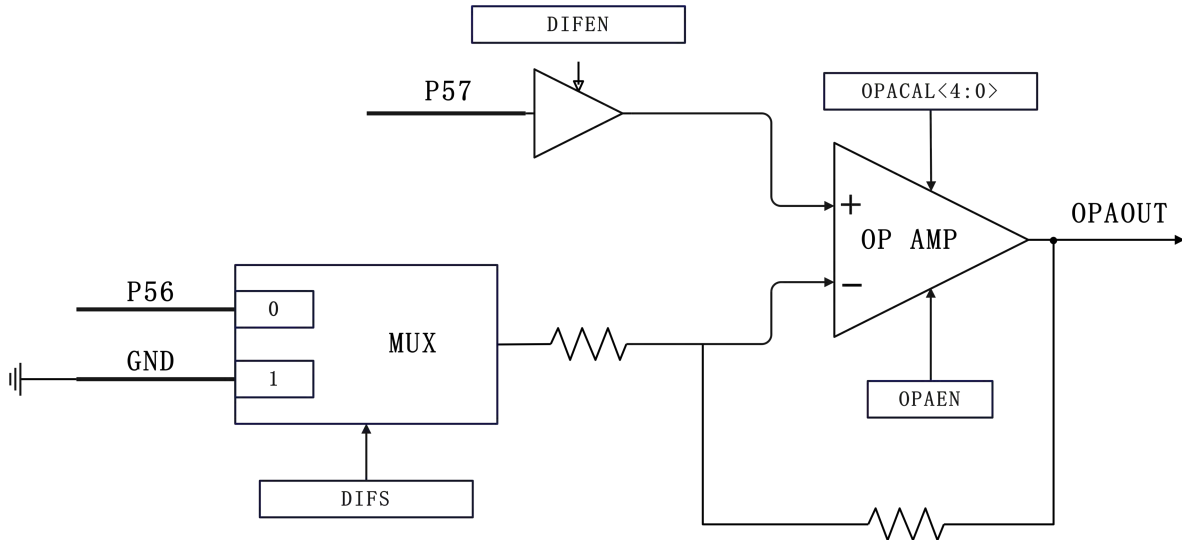
首先将 OPACON0 (R1A8) 寄存器中的 OPASIS<2:0>正负极端口进行配置，当 OPAEN 置 1，使能运算放大器后，正负端口将连接至 IO 口；对 OPACON1 (R1A9) 寄存器中的 OPAMS 置低选择外置电阻模式。OPA 负端接两个外部电阻 (R1、R2) 之间，正端输入外部电压，输出电压为正端电压的  $((R1+R2)/R1)$  倍。



## 模式三：差分运放模式

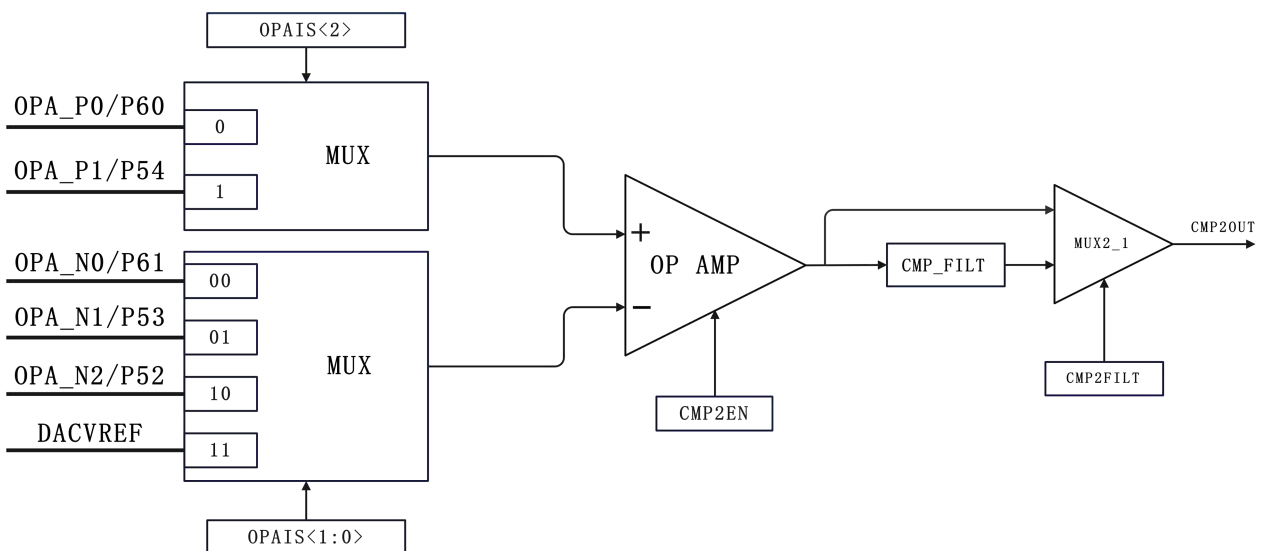
首先对 OPACON3 (R1A8) 寄存器中的 DIFS 进行配置选择 P56 口还是 GND 为差分运放负极 (VN)，对 DIFGAIN<1:0> 进行配置选择差分运放的放大倍数，差分运放正极 (VP) 默认为 P57 口，使能 DIFEN 进行差分运放功能。

$$\text{OPAOUT} = \text{DIFGAIN} * (\text{VP} - \text{VN}) + \text{VP}$$



## 模式四：比较器模式

首先将 OPACON0 (R1A8) 寄存器中的 OPAIS<2:0> 正负极端口进行配置，随后对 CMP2EN 进行使能，并关闭 OPA 使能。此时 VP 作为比较器的输入正极，VN 作为比较器的输入负极。可根据需求将 CMP2IE (R1D8) 使能比较器中断，还可以通过 CMP2FILT (R1A9) 控制位选择比较结果是否通过滤波功能，以及配置 CMP2WE (R185) 使能唤醒功能。



## 12.3 OPA 相关寄存器

### 12.3.1 R1A8/OPACON0 (OPA 控制寄存器 0)

0X1A8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON0	OPAEN	OPAOE	CMP2EN	CMP2OUT	CMP2OE	OPAIS<2:0>		
读/写	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: OPAEN - OPA 使能位

0: 关闭 OPA

1: 使能 OPA

Bit<6>: OPAOE - OPA 输出使能位

0: P55 作为 GPIO

1: OPA 输出至 P55

Bit<5>: CMP2EN - 比较器 CMP2 使能位

0: 关闭 CMP2

1: 使能 CMP2

Bit<4>: CMP2OUT - CMP2 结果输出位

0: 比较器 2 正极电压小于负极电压

1: 比较器 2 正极电压大于负极电压

Bit<3>: CMP2OE - 比较器 CMP2 结果输出使能

0: P55 作为 GPIO

1: 比较器结果从 P55 输出

Bit<2>: OPAIS<2> - CMP2/OPA 正极端口选择位

0: 选择 P60

1: 选择 P54

Bit<1: 0>: OPAIS<1: 0> - OPA\_VIN 选择位

OPAIS<1>	OPAIS<0>	负极端口选择
0	0	P61
0	1	P53
1	0	P52
1	1	DACVREF



## 12.3.2 R1A9/OPACON1 (OPA 控制寄存器 1)

0X1A9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON1	-	BIASEN	BIASS	CMP2FILT	OPAMS	OPAG<2:0>		
读/写	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<6>: BIASEN - 偏置电流使能位

0: 使能

1: 禁止

Bit<5>: BIASS - LCD/CRYOSC/CMP 偏置电流档位切换

0: 默认

1: 增强

Bit<4>: CMP2FILT - 比较器 CMP2 结果输出内部滤波选择

0: CMP2 比较结果滤波关闭

1: CMP2 比较结果滤波使能

Bit<3>: OPAMS - OPA 运放模式选择位

0: 外置电阻模式

1: 内置电阻模式

Bit<2:0>: OPAG<2:0> - OPA 运放内置电阻模式放大倍数选择

OPAG<2>	OPAG<1>	OPAG<0>	OPA 放大倍数
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

## 12.3.3 R1AA/OPACON2 (OPA 控制寄存器 2)

0X1AA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON2	TEMPEN	TEMPS	VOFSIGN	OPACAL<4:0>				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TEMPEN - 温度检测模块使能

0: 禁止

1: 使能 (温度越高电压值越小)

Bit<6>: TEMPS - 温度检测档位切换

0: 默认

1: 增强 (电压增大一倍)

Bit<5>: VOFSIGN - 运放失调电压校准极性

0: 往负极修调

1: 往正极修调

Bit<4:0>: OPACAL<4:0> - 运放失调电压修调

OPACAL<4:0>运放失调电压修调 4mv 一档, 一共 32 档。从 0~124mv

### 12.3.4 R1AB/OPACON3 (OPA 控制寄存器 3)

0X1AB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON3	-	OPATEST<1:0>		DIFEN	OPARS	DIFS	DIFGAIN<1:0>	
读/写	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<6:5>: OPATEST<1:0> - OPA 失调测试模式选择

OPATEST<1>	OPATEST<0>	模式
0	0	正常模式
0	1	端口短接地
1	0	端口短接正极
1	1	端口短接负极

Bit<4>: DIFEN - 差分运放控制位

0: 禁止

1: 使能

Bit<3>: OPARS - OPA 输出电阻选择位

0: 无电阻

1: 10K 电阻

Bit<2>: DIFS - 差分运放负极选择位

0: 选择 P56 为差分运放负极

1: 选择 GND 为差分运放负极

Bit<1:0>: DIFGAIN<1:0> - 差分运放的放大倍数选择

DIFGAIN<1>	DIFGAIN<0>	差分放大倍数
0	0	4
0	1	8
1	0	12
1	1	16

## 13 LED 点阵驱动

JZ8FT8801 芯片内置 LED 驱动模块，该驱动电路支持最大 9\*10 模式 90 灯 LED 驱动，可配置 LEDCON 寄存器 LEDDUTY<2:0>选择点阵模式 4\*5、5\*6、6\*7、7\*8、8\*9、9\*10 不同大小的点阵对应的灯地址不变，LED 数据通过 R1F8 寄存器配合 R180 寄存器间接寻址，分别对地址 LEDRAM0~8 配置，LEDRAM 位宽为 10Bit，间接寻址方式为五位写入，也就是 LEDRAM0 低五位对应 R180 地址为 0 时的数据，LEDRAM0 高五位对应 R180 地址为 1 时的数据，以此类推。LED 时钟源可以选择系统时钟或者指令时钟，通过 LEDPSR<2:0>对时钟进行 4/8/16/32/64/128 分频。灯的亮度可以通过 LED 占空比寄存器 LEDDT<6:0>进行调节。

### 13.1 LED 功能描述

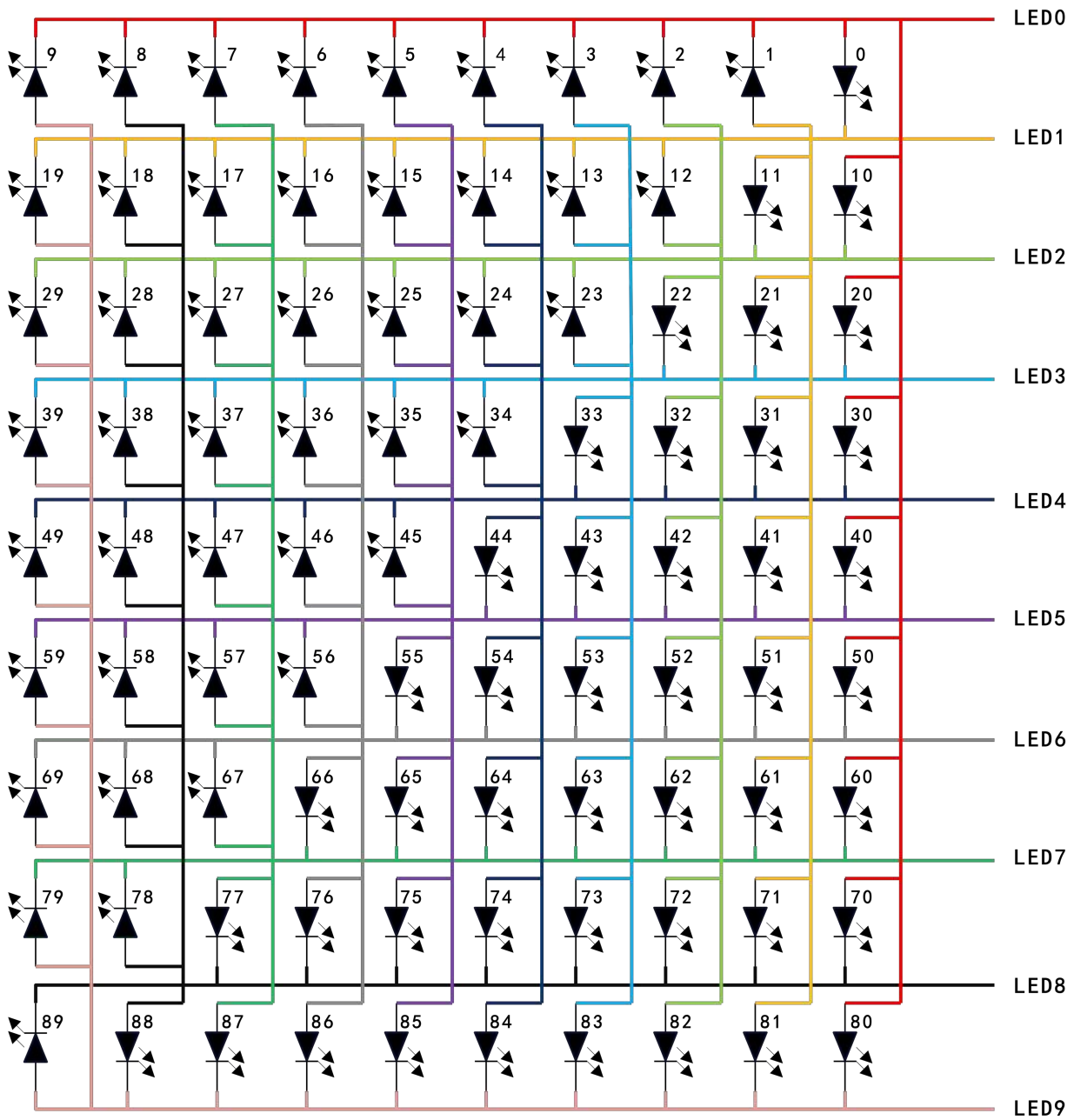
LED 点阵不同模式主要有三种扫描模式，一般通用 7\*7 及以上点阵主要以三灯扫描模式为主，即一次最大可同时点亮三个灯（共阴极），其他较小点阵模式则是双灯与单灯扫描模式为主，即一次最多可点亮两个灯或一个灯。LEDRAM0~8 应表格中显示对应地址的亮灯情况，1 表示亮灯，0 表示不亮灯。

设置寄存器 LEDEN=1，将可启用 LED 点矩阵模式。设置 LEDDT<6:0>为 0x7F 时亮度最高。可以通过设置 LEDHOLD 提供扫描暂停功能。

LEDRAM	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDRAM0	9	8	7	6	5	4	3	2	1	0
LEDRAM1	19	18	17	16	15	14	13	12	11	10
LEDRAM2	29	28	27	26	25	24	23	22	21	20
LEDRAM3	39	38	37	35	35	34	33	32	31	30
LEDRAM4	49	48	47	46	45	44	43	42	41	40
LEDRAM5	59	58	57	56	55	54	53	52	51	50
LEDRAM6	69	68	67	66	65	64	63	62	61	60
LEDRAM7	79	78	77	76	75	74	73	72	71	70
LEDRAM8	89	88	87	86	85	84	83	82	81	80

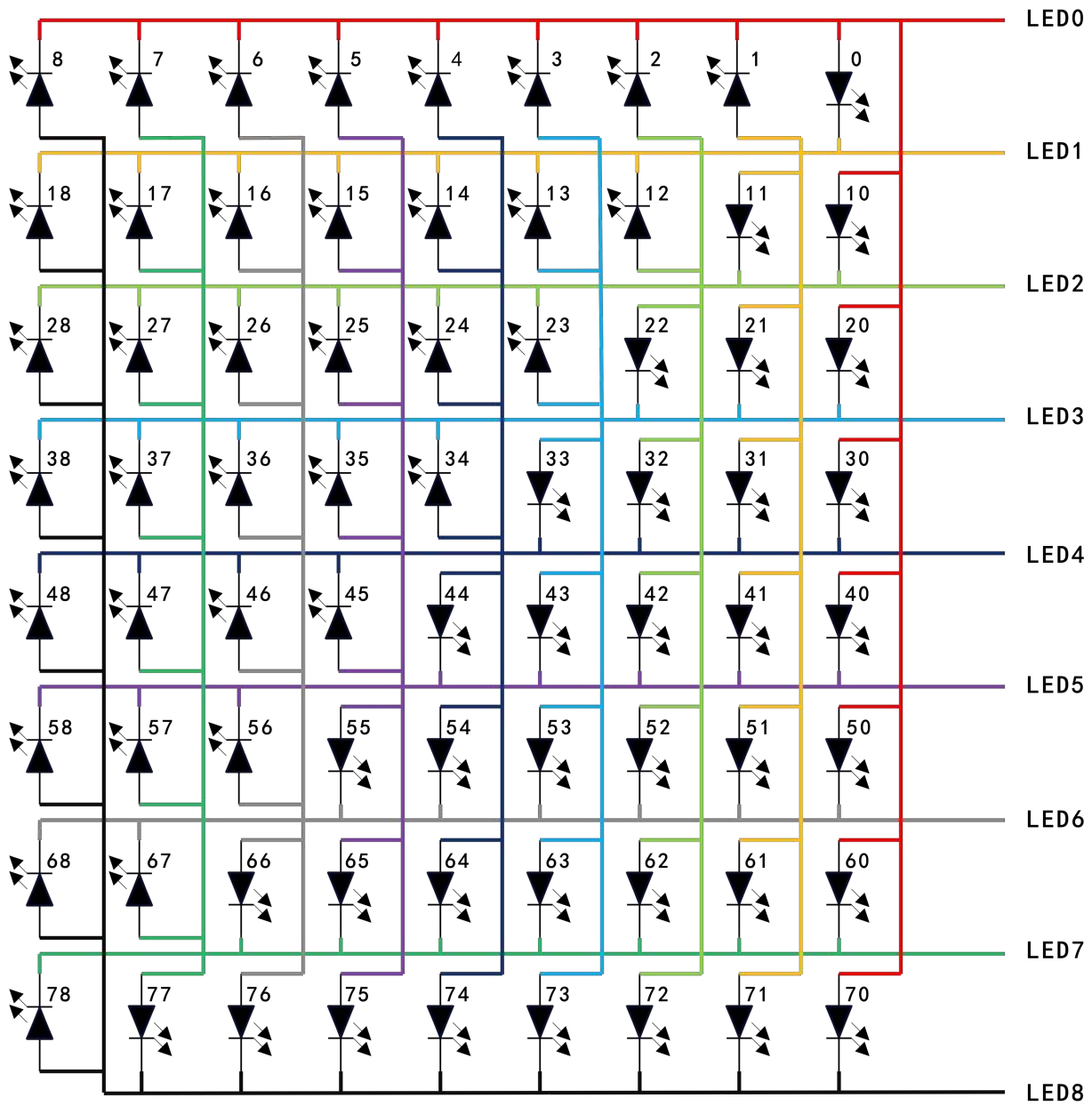


9\*10 矩阵:



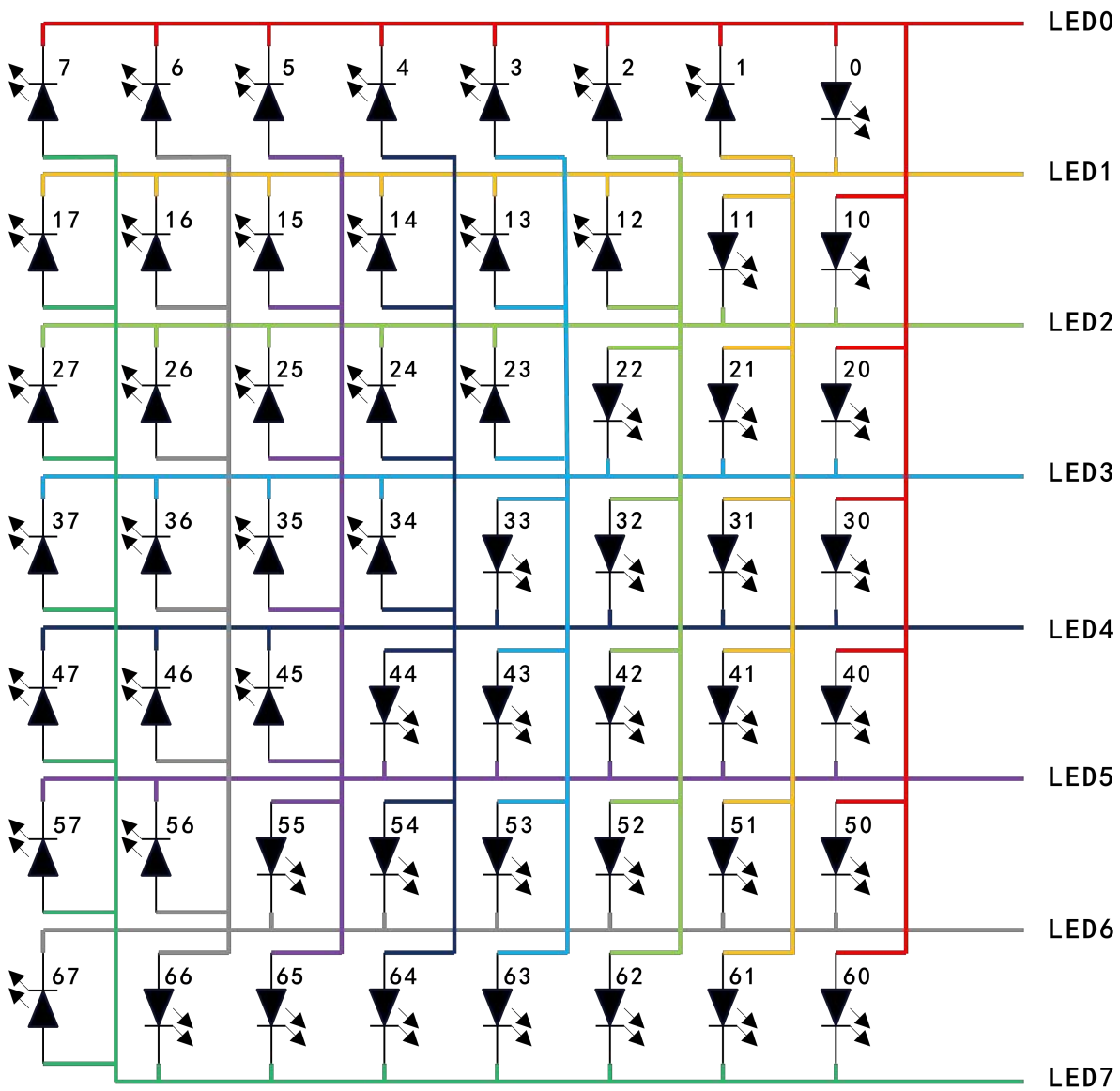


8\*9 矩阵:



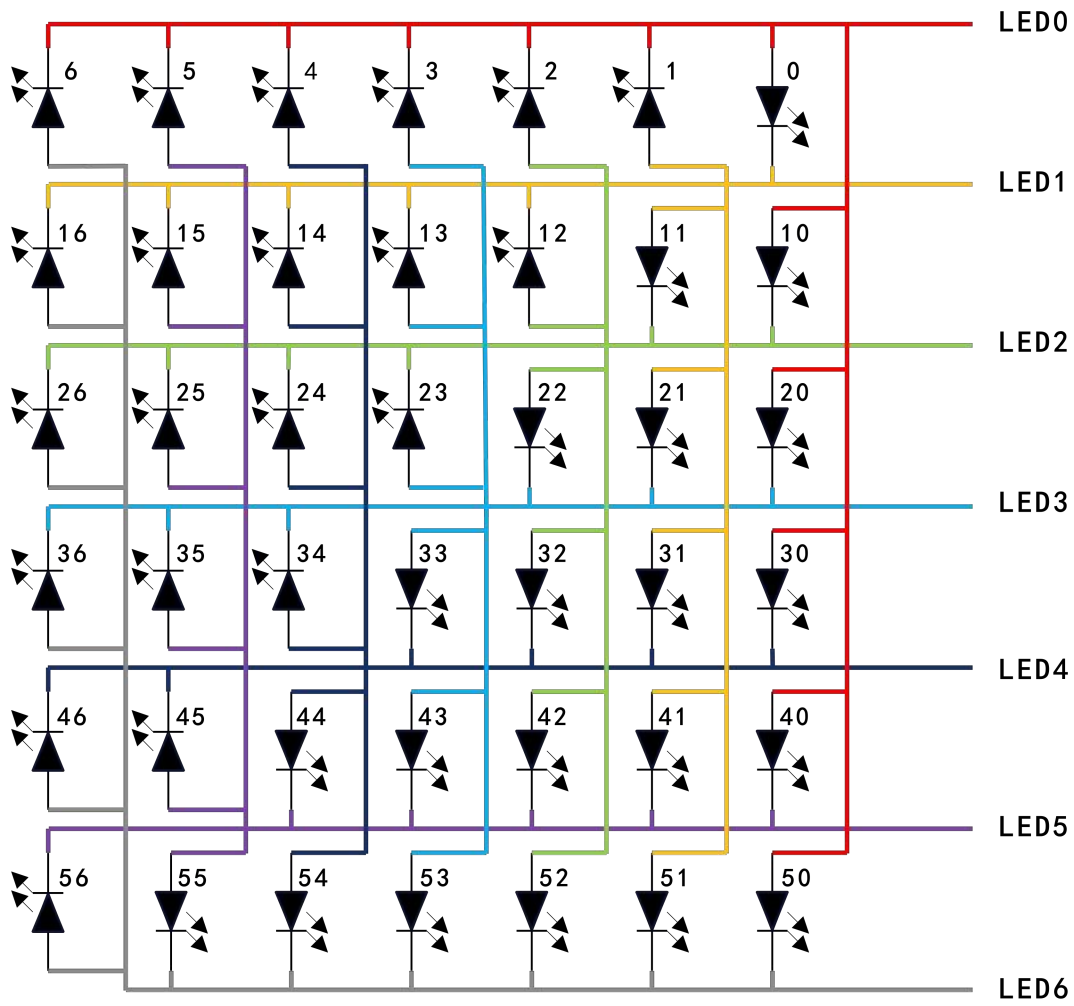


7\*8 矩阵:



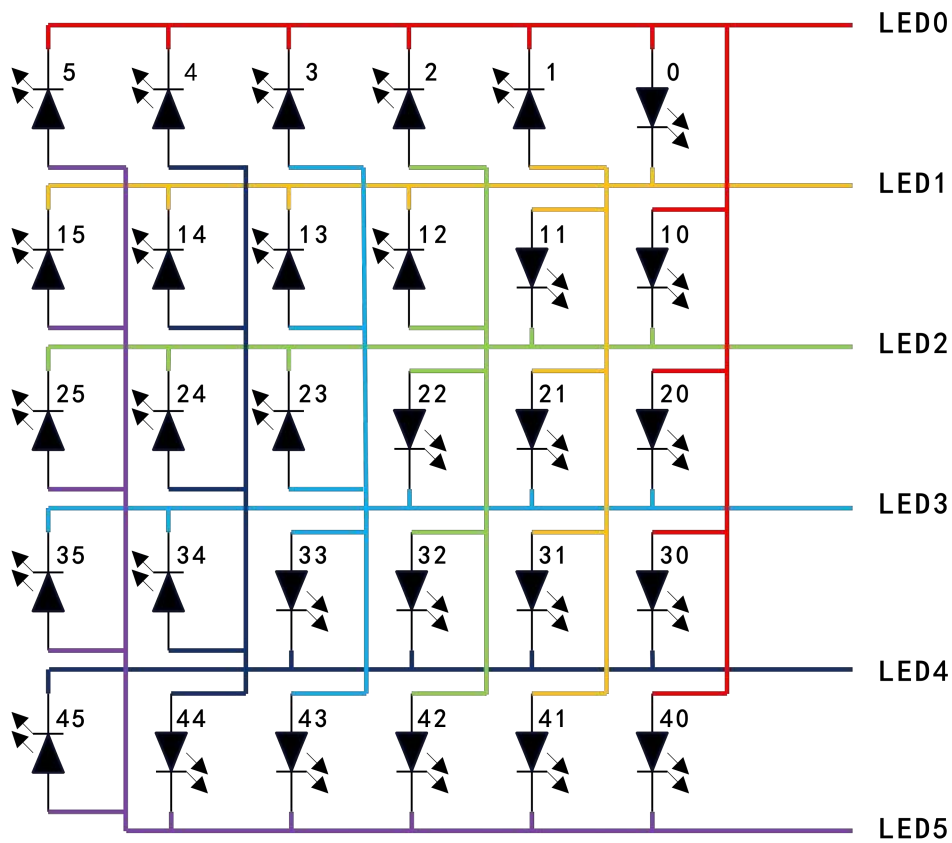


6\*7 矩阵:

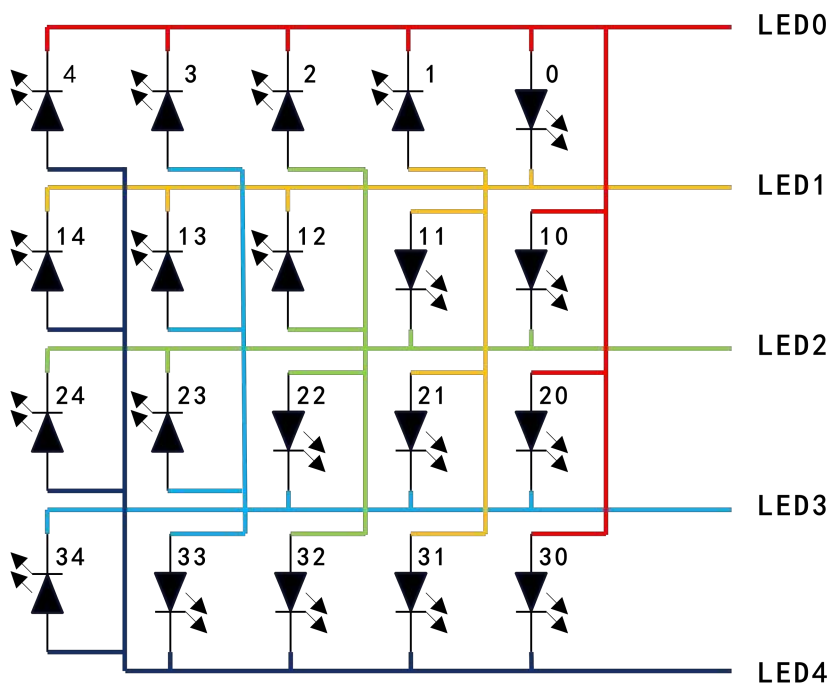




5\*6 矩阵:



4\*5 矩阵:



## 13.2 LED 点阵驱动设置说明

1. 设置寄存器 LEDRAM0~9 写入值，对应点亮相应的 LED 灯
2. 设置 LED 占空比寄存器 LEDDT<6:0>配置所 LED 灯需要的亮度
3. 配置 LED 控制寄存器使能 LED，配置点阵模式，选择 LED 时钟时钟源和分频比
4. 如需要暂停 LED 点阵扫描，设置 LEDHOLD=1，LED 端口输出暂停且保持；当 LEDHOLD=0，LED 端口输出恢复并继续扫描。

## 13.3 LED 点阵相关寄存器

### 13.3.1 R1D4/LEDCON(LED 控制寄存器)

0X1D4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON	LEDEN	LEDDUTY<2:0>			LEDCKS	LEDPSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: LEDEN - LED 使能位

- 0: 禁止
- 1: 使能

Bit<6:4>: LEDDUTY<2:0> - LED 点阵模式选择选择位

LEDDUTY<2>	LEDDUTY<1>	LEDDUTY<0>	LED 点阵模式
0	0	0	4*5
0	0	1	5*6
0	1	X	6*7
1	0	X	7*8
1	1	0	8*9
1	1	1	9*10

Bit<3>: LEDCKS - LED 时钟源选择位

- 0: 系统时钟
- 1: 指令时钟

Bit<2:0>: LEDPSR<2:0> - LED 时钟分频选择位

LEDPSR<2>	LEDPSR<1>	LEDPSR<0>	分频比
0	0	0	4 clock
0	0	1	8 clock
0	1	0	16 clock
0	1	1	32 clock
1	0	0	64 clock
1	0	1	128 clock



## 13.3.2 R1D5/LEDDT(LED 占空比寄存器)

0X1D5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDT	LEDHOLD	LEDDT<6:0>						
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: LEDHOLD - LED 暂停保持位

0: LED 正常运行

1: LED 暂停保持

Bit<6:0>: LEDDT<6:0> - LED 亮度选择位

## 14 LCD 驱动

### 14.1 LCD 功能描述

JZ8FT8801 LCD 模块具有 5 个 common 引脚和 23 个 segment 引脚。每个引脚有一个对应的寄存器使能位，通过 R1F1 低五位以及 R1F5、R1F6、R1F7 寄存器进行控制，能独立控制该引脚的使能与否，未使能的引脚可作为 GPIO 或是其他功能引脚正常使用，由于 COM4 与 SEG23 复用了 P64 端口，所以在使用时需注意不能同时使能。

LCD 模块中，时序控制器根据系统时钟来设定显示存取和占空比的时序，以满足显示需求。R1F0、R1F1 是 LCD 驱动器的命令寄存器，它包括对 LCD 使能/禁止、LCD 驱动类型选择、占空比（1/2、1/3、1/4、1/5）、偏压类型、时钟源和 LCD 帧频率的控制。

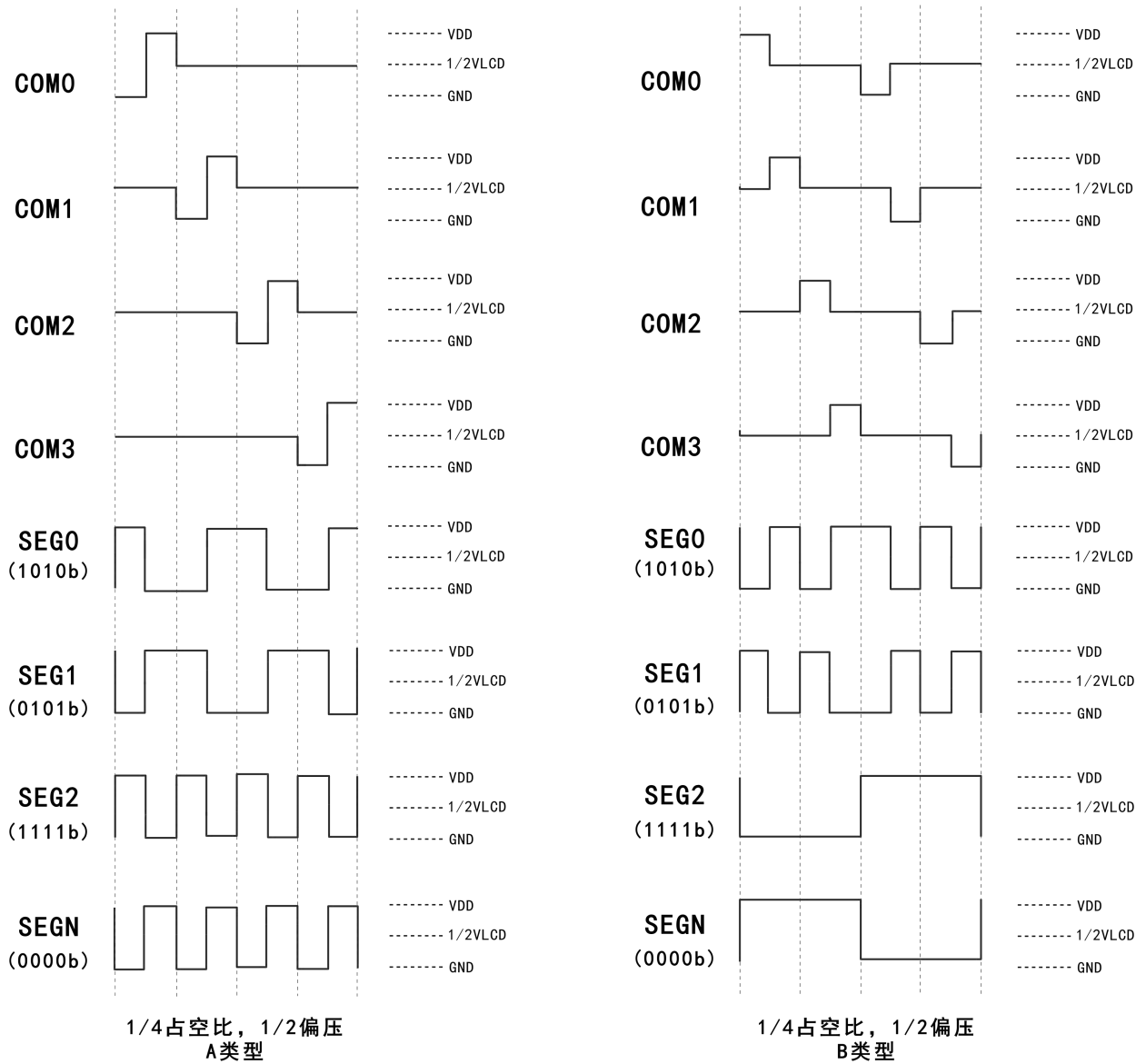
LCD 和 LED 驱动模块，它们共用数据寄存器 LCDDB(R1F8)。LCDDB<7:0>是 LCD RAM 数据读写位，配合 0x180 寄存器，0x180 为 LCDRAM 地址寄存器。LCDRAM0-23，每个 LCDRAM 只取低 5 位为有效位。

不同规格参数的屏可能需要匹配不同帧频扫描以达到更好的显示效果，JZ8FT8801 根据 LCD 时钟源以及分频选项不同，每种占空比提供有 16 种帧频可供选择。根据不同的选项配置可以用以下公式计算帧频：

$$\text{帧频} = \text{LCDCKS} / \text{基础分频} / \text{Duty} / 2$$

LCDADDR	LCDDATA					
0x00	Bit4	Bit3	Bit2	Bit1	Bit0	SEGO
0x01	Bit4	Bit3	Bit2	Bit1	Bit0	SEG1
0x02	Bit4	Bit3	Bit2	Bit1	Bit0	SEG2
0x03	Bit4	Bit3	Bit2	Bit1	Bit0	SEG3
.....	...	...	...	...	...	.....
0x22	Bit4	Bit3	Bit2	Bit1	Bit0	SEG22
0x23	Bit4	Bit3	Bit2	Bit1	Bit0	SEG23

下面分别为 1/2 偏压，1/4 占空比 A/B 类的 LCD 波形



## 14.2 LCD 相关寄存器

### 14.2.1 R1F0/LCDCON0 (LCD 控制寄存器 0)

0X1F0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LDCON	LCDEN	TYPE	DS<1:0>		LCDPSR<3:0>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: LCDEN - LCD 使能位

0: 禁止

1: 使能

Bit<6>: TYPE - LCD 类型选择位

0: A 类型

1: B 类型

Bit<5:4>: DS<1:0> - LCD 占空比选择位

DS<1>	DS<0>	LCD 占空比
0	0	1/2
0	1	1/3
1	0	1/4
1	1	1/5

Bit<3:0>: LCDPSR<3:0> - LCD 时钟分频计数值

LCDPSR<3>	LCDPSR<2>	LCDPSR<1>	LCDPSR<0>	分频系数
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

LCDCKS	LCDPSR	帧频			
		1/5Duty	1/4Duty	1/3Duty	1/2Duty
32768	1	546	512	546	512
32768	2	273	256	273	256
32768	3	182	171	182	171
32768	4	137	128	137	128
32768	5	109	102	109	102
32768	6	91	85	91	85
32768	7	78	73	78	73
32768	8	68	64	68	64



32768	9	61	57	61	57
32768	10	55	51	55	51
32768	11	50	47	50	47
32768	12	46	43	46	43
32768	13	42	39	42	39
32768	14	39	37	39	37
32768	15	36	34	36	34
32768	16	34	32	34	32

注：（1）1/2 占空比的基础分频是 16clock；1/3 占空比的基础分频是 10clock；1/4 占空比的基础分频是 8clock；1/5 占空比的基础分频是 6clock；

## 14.2.2 R1F1/LCDCON1 (LCD 控制寄存器 1)

0X1F1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON1	LCDBS	VLCCKS<1:0>			COMEN<4:0>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: LCDBS - LCD BIAS 选择位

0: 1/2BIAS

1: 1/3BIAS

Bit<6: 5>: VLCCKS<1:0> - LCD 时钟源选择位

0x: 系统时钟的 512 分频

10: 外部 32.768KHz 晶振

11: 低速时钟

Bit<4:0>: COMEN<4:0> - COM 使能位

0: 禁止

1: 使能

## 14.2.3 R1F2/LCDCON2 (LCD 控制寄存器 2)

0X1F2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON2	SPEED<2:0>			VLCDS<4:0>				
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:5>: SPEED<2:0> - LCD 小电阻模式开启时间

$T_{spd} = (SPEED - LCDDEAD) * LCDCK$  (注:  $SPEED - LCDDEAD > 0$ )

$T_{spd} = 0$  (注:  $SPEED - LCDDEAD = 0$ )

Bit<4:0>: VLCD<4:0> - VLCD 电压选择

VLCD[4:0]	VLCD	VLCD[4:0]	VLCD
00000	0.40VDD	01000	0.67VDD
00001	0.43VDD	01001	0.70VDD
00010	0.46VDD	01010	0.74VDD
00011	0.49VDD	01011	0.78VDD
00100	0.53VDD	01100	0.83VDD
00101	0.56VDD	01101	0.87VDD
00110	0.60VDD	01110	0.91VDD
00111	0.63VDD	01111	0.95VDD
		1xxxx	VDD

## 14.2.4 R1F3/LCDCON3 (LCD 控制寄存器 3)

0X1F3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON3	LCDDEAD<1:0>		RLCDS<5:0>					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:6>: LCDDEAD<1:0> - COM/SEG 口切换死区时间

$$T_{dead} = (LCDDEAD - 0.5) * LCDCKS \quad (\text{注: } LCDDEAD > 0)$$

$$T_{dead} = 0 \quad (\text{注: } LCDDEAD = 0)$$

注: LCDCKS 时钟为 LCDCON1 寄存器 LCDCKS<1:0>选择的 LCD 时钟源时钟;

Bit<5:0>: RLCDS<5:0> - 偏压的分压电阻选择信号, 000000 电阻最大, 111111 电阻最小。

## 14.2.5 R1F4/LCDCON4 (LCD 控制寄存器 4)

0X1F4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON4	LCDDRVS	RLCDST	RLCDT<5:0>					
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: LCDDRVS - RLCD 切换模式选择

0: 手动切换模式, 每次手动改变 RLCDS<5:0>或开启 BUFF

1: 自动切换模式, 自动切换小电阻增强模式

注: (1)RLCDS<5:0>设定为大电阻阻值, RLCDT<5:0>设定为小电阻阻值, 小电阻对应较大的驱动电流, 大电阻对应较小的驱动电流;

(2) 建议设置 RLCDS < RLCDT, 设置时必须选自动切换模式, 手动切换模式时固定只有大电阻;

(3) 平时不切换的时候建议选大电阻, 功耗更小。

Bit<6>: RLC DST - 小电阻模式驱动能力增强使能位

- 0: 禁止
- 1: 使能

Bit<5:0>: RLC DT<5:0> - 自动切换模式下的小电阻值

## 14.2.6 R1F5/SEGCON0 (SEG 控制寄存器 0)

0X1F5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEGCON0	SEGEN<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: SEGEN<7:0> - SEG 使能位

- 0: 禁止
- 1: 使能

## 14.2.7 R1F6/SEGCON1 (SEG 控制寄存器 1)

0X1F6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEGCON1	SEGEN<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: SEGEN<15:8> - SEG 使能位

- 0: 禁止
- 1: 使能

## 14.2.8 R1F7/SEGCON2 (SEG 控制寄存器 2)

0X1F7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEGCON2	SEGEN<23:16>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: SEGEN<23:16> - SEG 使能位

- 0: 禁止
- 1: 使能



## 14.2.9 R1F8/LCDDDB(LCD 数据寄存器)

0X1F8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDDDB	-	-	-	LCDDDB<4:0>				
读/写	-	-	-	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0

Bit<4:0>: LCDDDB<4:0> - LCD RAM 数据读写位，配合 0x180 寄存器，0x180 为 LCDRAM 地址寄存器

## 15 UART 串行接收器/发送器

通用异步收发器 (UART) 串口是一个串行 I/O 通信外设。串行接口应用于低速数据传输，以及与低速外围设备通信。JZ8FT8801 支持一个字节数据长度，传输格式有起始位、8 位数据、奇偶校验位和停止位。可编程的波特率支持不同速度的外围设备。

串口通信模式主要有以下三种模式，由 SCON0 寄存器高两位进行选择。

模式 1，8 位异步全双工通信模式，由一个起始位，8 个数据位组成，收发波特率可以选择 TC4 或者 TC5 的溢出信号作为时钟源。

模式 2，9 位异步固定波特率通信模式，由一个起始位，8+1 个数据位组成，收发波特率可选择系统时钟的 8 分频或者 16 分频，通过 SCON1 寄存器的 SMOD 控制位进行选择。

模式 3，9 位异步可变波特率通信模式，由一个起始位，8+1 个数据位组成，收发波特率可以选择 TC4 或者 TC5 的溢出信号作为时钟源。

UART 在通讯之前必须先初始化串口控制寄存器，选择工作方式和波特率，如果是模式一和模式三还需要先初始化定时器 TC4/TC5。

在所有模式中，任何对 UART 收发数据寄存器 SBUF 的写操作都会启动发送。

UART 模式列表：

SM<0:1>	模式	类型	波特率	帧长度	起始位	停止位	第 9 位
01	1	异步	TC0/TC3 溢出	10 位	1 位	1 位	-
10	2	异步	SYSCK 32 分频/64 分频	11 位	1 位	1 位	0/1
11	3	异步	TC0/TC3 溢出	11 位	1 位	1 位	0/1

**波特率计算：**

UART 的时钟来自于 TC4 和 TC5 的 8 分频或 16 分频，根据这个时间匹配目标波特率。

例：目标波特率 9600bps，UART 时钟来自于 TC4 的 8 分频

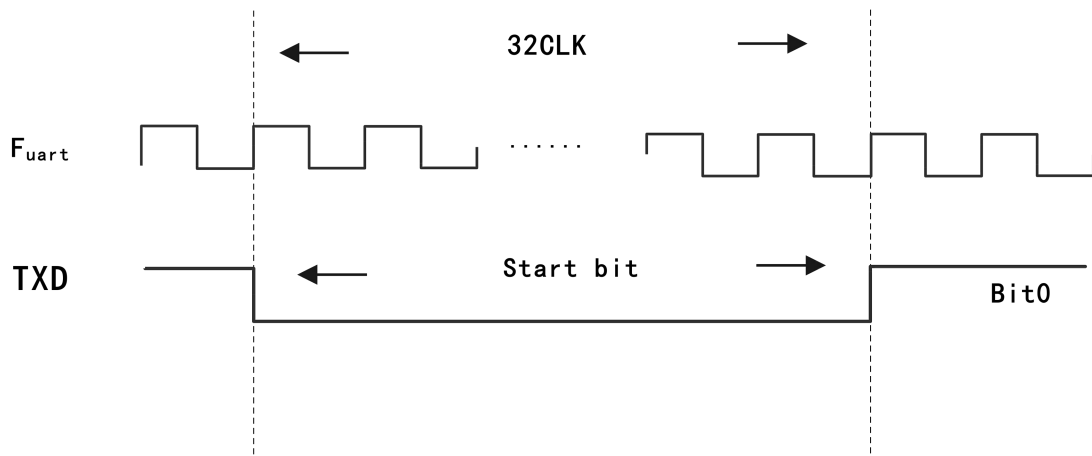
$$TC4 = 1/9600/8 = 13\mu s$$

### 15.1 UART 发送

UART 串行发送数据主要操作如下：

1. 将 UART 串口控制寄存器 SCON0、SCON1 进行配置，选择对应模式、端口以及波特率。
2. 将数据写入 UART 收发数据寄存器 SBUF 中。
3. UART 将开始发送。
4. 数据将串行从 TX 引脚发送，发送数据从 LSB 到 MSB 输出。
5. 停止位：输出一个停止位（“1”）。

下图为 UART 发送计数器时序（32CLK 仅供参考，波特率可通过配置位选择）

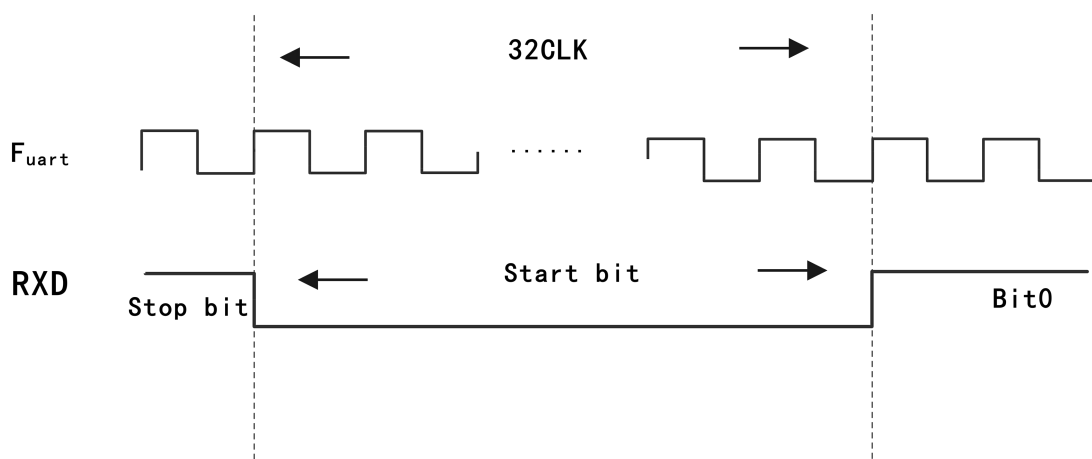


## 15.2 UART 接收

UART 串行接收数据主要操作如下：

1. 将 SCON0 寄存器的 REN 位置高使能 UART 接收功能。UART 监视 RXD 引脚，并在检测到起始位时进行内部同步。
2. 起始位，输出一个“0”位。
3. 接收的数据按从 LSB 到 MSB 的顺序移入 UBUF 寄存器。
4. 奇偶校验位，可由 SCON0 寄存器进行选择，输出一个奇偶校验位。
5. 停止位：输出一个停止位（“1”）。
6. 接收奇偶校验位和停止位，收到最后一个字符后，SCON0 寄存器的 RI 位将设置为“1”。  
若使能 UART 中断并使能 EI，这将发生 UART 中断。
7. 从 SBUF 寄存器读取接收的数据。将 RI 位由软件清零（若使能中断）。

下图为 UART 接收计数器时序（32CLK 仅供参考，波特率可通过配置位选择）



## 15.3 UART 相关寄存器

### 15.3.1 R1DD/SCON0 (UART0 串口控制寄存器)

0X1DD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON0	SM0	SM1	SM2	REN0	TB8	RB8	TI0	RI0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:6>: SM0:SM1 模式选择

SM0	SM1	工作模式	波特率
0	1	8 位 UART	定时器溢出频率的 1/8
1	0	9 位 UART	系统时钟的 1/8 或 1/16 由 SMOD 决定
1	1	9 位 UART	定时器溢出频率的 1/8

Bit<5>: SM2 - 多机通讯控制位

0: 接收时不论第 9 位 (RB8) 为 0 或 1, 将数据送入 SBUF, 并将 RI0 置 1 产生中断请求

1: 接收时当第 9 位 (RB8) 为 1 时, 将数据送入 SBUF, 并将 RI0 置 1 产生中断请求

Bit<4>: REN - UART 接收使能控制

0: 禁止接收

1: 使能接收

Bit<3>: TB8 - UART 发送数据的第 9 位

Bit<2>: RB8 - UART 接收数据的第 9 位

Bit<1>: TI0 - 发送中断标志位

发送停止位硬件置 1, 软件清 0

Bit<0>: RI0 - 接收中断标志位

收到停止位硬件置 1, 软件清 0

### 15.3.2 R1DE/SBUF (UART0 收发数据寄存器)

0X1DE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF	SBUF<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

串行发送时, MCU 对 SBUF 写入数据

串行接收时, MCU 从 SBUF 读取数据



## 15.3.3 R1DF/SCON1 (UART0 串口控制寄存器 1)

0X1DF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	-	-	UARTS	TXDS	RXDS	SMOD	TCLK	RCLK
读/写	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:6>: 未使用位

Bit<5>: UARTS - UART 端口选择

0: 选择 UART0A 组端口, 对应 TX0A、RX0A

1: 选择 UART0B 组端口, 对应 TX0B、RX0B

Bit<4>: TXDS - UART 发送口控制

0: 禁止 UART 发送口

1: 使能 UART 发送口 (UARTS =0 选 P60 , UARTS=1 选 P67)

Bit<3>: RXDS - UART 接收口控制

0: 禁止 UART 接收口

1: 使能 UART 接收口 (UARTS =0 选 P61 , UARTS=1 选 P66)

Bit<2>: SMOD - UART 波特率控制位 (仅对模式 2 有效)

0: UART 波特率使用系统时钟的 1/8

1: UART 波特率使用系统时钟的 1/16

Bit<1>: TCLK - UART 发送波特率控制位 (仅对模式 1、3 有效)

0: UART 发送波特率使用 TC4 溢出频率

1: UART 发送波特率使用 TC5 溢出频率

Bit<0>: RCLK - UART 接收波特率控制位 (仅对模式 1、3 有效)

0: UART 接收波特率使用 TC4 溢出频率

1: UART 接收波特率使用 TC5 溢出频率

## 16 UART/IIC/SPI 三选一串行接口

JZ8FT8801 内部集成了三选一串行接口线路，可方便 MCU 与不同接口的器件或者设备的连接。用户可通过配置寄存器 SSICON0 (R1E3) 寄存器 MODE1~0 两位选择 UART、IIC 和 SPI 任意一种通信模式。

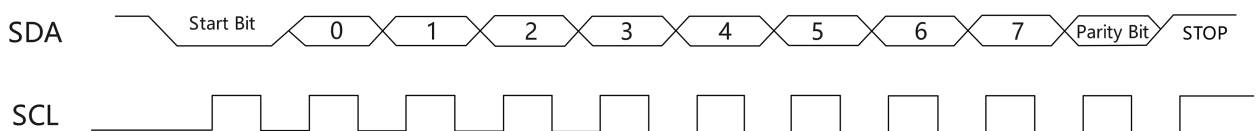
### 16.1 UART 串行接收器/发送器

通用同步/异步收发器 (UART) 串口是一个串行 I/O 通信外设。串行接口应用于低速数据传输，以及与低速外围设备通信。JZ8FT8801 支持一个字节数据长度，传输格式有起始位、8 位数据、奇偶校验位和停止位。可编程的波特率支持不同速度的外围设备。

串口通信模式主要有以下两种模式，由 SSICON1 (R1E4) 寄存器 Bit5 位进行选择。

模式1，8位通信模式，由一个起始位，8个数据位组成，收发波特率可以选择系统时钟的分频，TC4或者TC5的溢出信号作为时钟源。

模式2，9位通信模式，由一个起始位，8+1个数据位组成，第九位数据通过 SSICON1 (R1E4) 寄存器 Bit1 位进行发送，收发通信波特率可选择系统时钟的分频，TC4或者TC5的溢出信号作为时钟源。

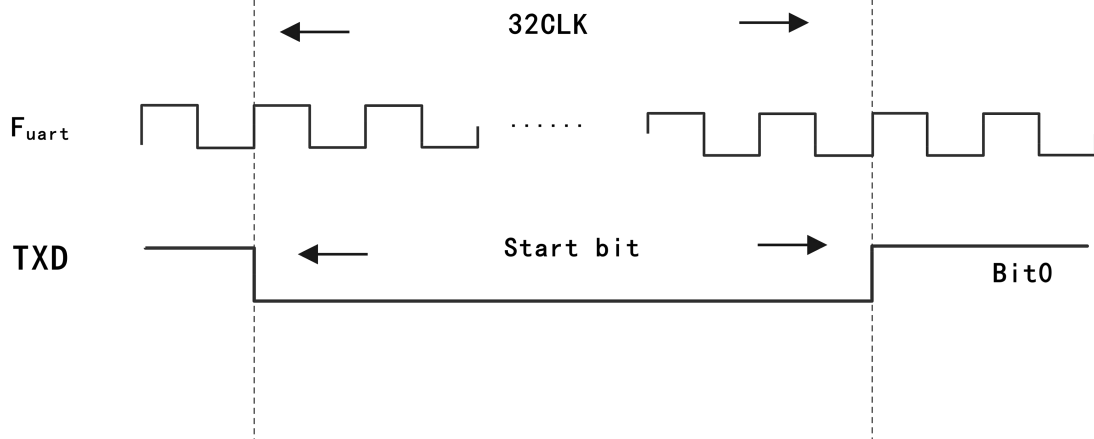


#### 16.1.1 UART 发送

UART 串行发送数据主要操作如下：

1. 将 UART 串口控制寄存器 SSICON0 (R1E3)、SSICON1 (R1E4) 进行配置，选择对应模式、端口以及波特率，发送需使能 TEN。
2. 将数据写入 UART 收发数据寄存器 SSIBUF (R1E6) 中，UART 将自动开始发送。
3. 数据将串行从 TX 引脚发送，发送数据从 LSB 到 MSB 输出。
4. 停止位：输出一个停止位 (“1”)。

下图为 UART 发送计数器时序 (32CLK 仅供参考，波特率可通过配置位选择)

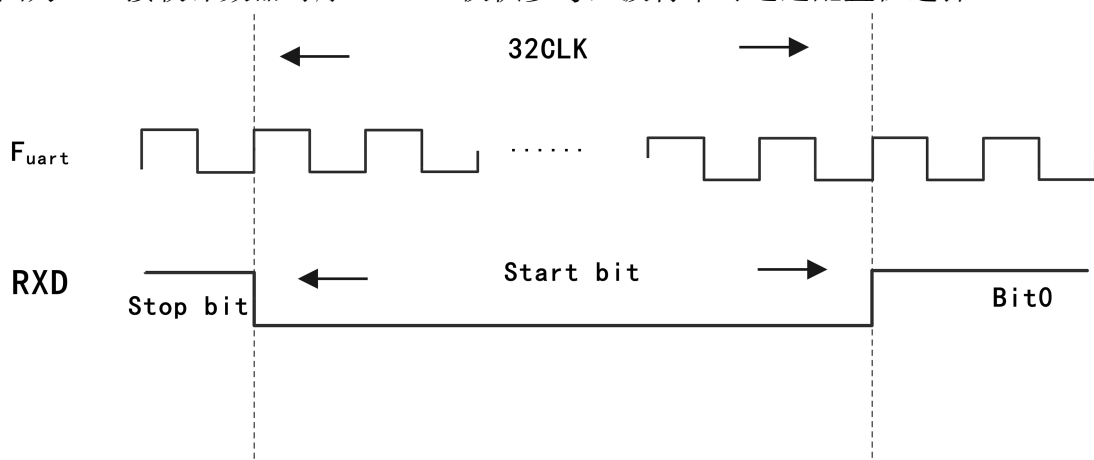


## 16.1.2 UART 接收

UART串行接收数据主要操作如下：

1. 将SSICON1 (R1E4) 寄存器的REN位置高使能UART接收功能。UART监视RXD引脚，并在检测到起始位时进行内部同步。
2. 起始位，输出一个“0”位。
3. 接收的数据按从LSB 到MSB 的顺序移入BUF寄存器，接收8为数据后将写入SSIBUF。
4. 奇偶校验位，使能9位UART模式时可以读取SSISR1 寄存器RBIT8。
5. 停止位：输出一个停止位（“1”）。
6. 接收奇偶校验位和停止位，收到最后一个字符后，SCON0寄存器的RI位将设置为“1”。  
若使能UART中断并使能EI，这将发生UART 中断。
7. 从SSIBUF寄存器读取接收的数据。将RI位由软件清零。

下图为UART接收计数器时序（32CLK仅供参考，波特率可通过配置位选择）

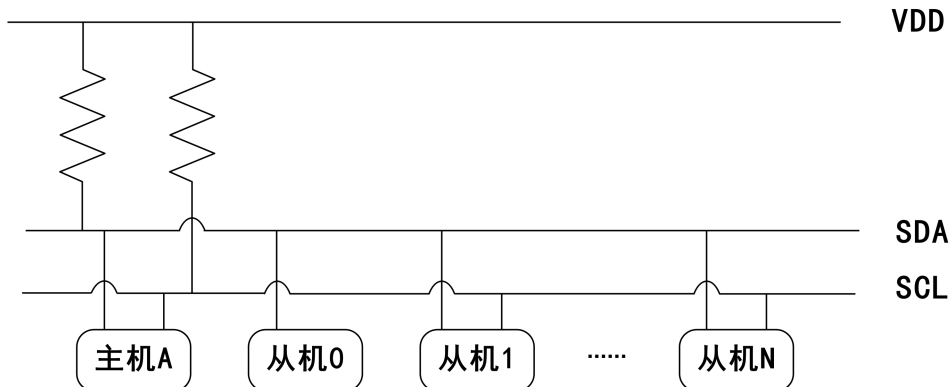


## 16.2 IIC 串行接收器/发送器

IIC是一种同步串行半双工通信总线方式。该总线允许同时连接多个设备，通过每个芯片的特定地址进行选择。

IIC总线在某一时刻，总线只允许有一个设备处于发送状态，所发出的数据被总线上所有芯片进行接收，IIC通信协议包含有设备地址，只有发送方携带的地址与某个接收方地址相同时，接收方才能执行相关的指令。

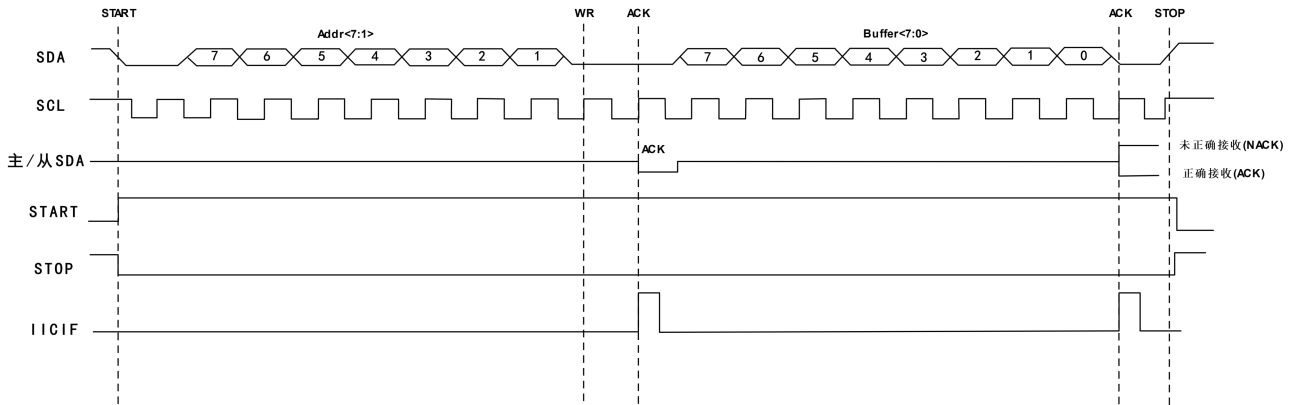
IIC总线规定，当设备处于空闲状态时，时钟和数据两根总线都处于高电平状态。



### 16.2.1 IIC 操作流程

IIC串行发送数据主要操作如下

1. 开始信号，SCL保持高电平，SDA由高电平变为低电平（下降沿）；
2. 通过I2CADDR（1E5）寄存器发送 I2C 设备地址，每个 I2C 器件都有一个设备地址，通过发送具体的设备地址来决定访问哪个 I2C 器件。
3. I2C 器件地址后面跟着一个读写位（WR），为 0 表示写操作，为 1 表示读操作；
4. 地址匹配的从机在第九个时钟周期发送 ACK 应答信号，ACK（低电平）接受成功，继续传输，NACK（高电平），表示数据接收失败或终止传输。
5. 数据传输（写操作），主设备向从设备写数据，在地址匹配且接收到应答信号后，主设备开始发送字节，每字节后从机应答ACK。
6. 数据传输（读操作），主设备读取从设备发送过来的数据，在地址匹配接收到从机ACK信号后，等待从机发送字节，主设备在每字节后应答ACK。当应答信号为NACK时，停止发送。
7. 重复起始条件，当主设备需要切换读写方向或访问另一从机时，不释放总线继续传输，回到操作2；
8. 停止条件，SCL保持高电平，SDA由低电平转变为高电平（上升沿）。



## 16.2.2 IIC 发送与接收

IIC发送数据主要操作如下：

1. 设置SSICON0 (R1E3) 寄存器，使能SSIEN，选择IIC模式，按需设置时钟分频。
2. 发送起始位需要写入地址及WR控制，若不是起始位则将数据写入SSIBUF。
3. 设置SSICON1 (R1E4) 寄存器，按需配置映射端口，设置主机模式，如需发送地址则使能START，如需结束通信则使能STOP。
4. 如需发送完成进入中断，则使能TX\_IICIE。
5. 设置SSICON1 (R1E4) 寄存器，使能RUN\_EN，开始发送数据。

## 16.3 SPI 串行接收器/发送器

SPI 是一种同步、全双工、主从式串行通信协议，该总线由MISO，MOSI，SCK和CS四条总线组成，可通过时钟极性和相位配置实现不同模式的通讯。

以下为SPI四条总线的主要作用：

SCK (Serial Clock)：时钟信号，由主设备提供，决定数据传输的节拍。

MOSI (Master Out, Slave In)：主设备输出数据，从设备输入数据引脚。

MISO (Master In, Slave Out)：主设备输入数据，从设备输出数据引脚。

CS/NSS (Chip Select)：片选引脚，低有效，主设备控制，用于选择当前通信的从设备。

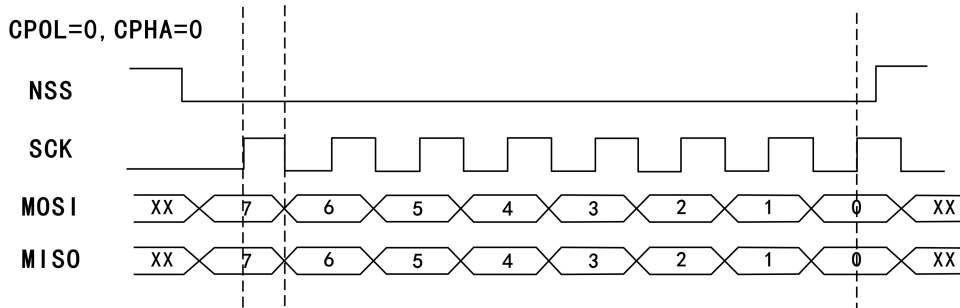
### 16.3.1 SPI 工作模式

SPI可以通过SSICON1 (R1E4) 寄存器Bit<6:5>对时钟极性和相位进行配置。

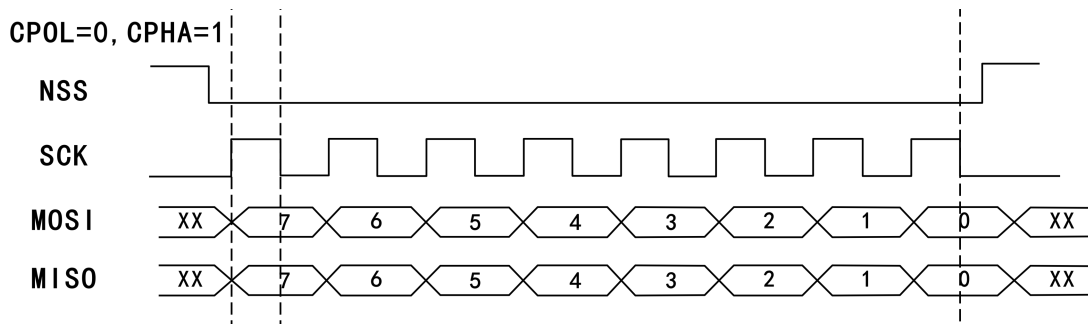
- CPOL=0, SPI总线空闲时，时钟线SCK为低电平。
- CPOL=1, SPI总线空闲时，时钟线SCK为高电平。
- CPHA=0, 在SCK第一个跳变沿，主机对MISO引脚电平采样；主机的数据发送则在第二个跳边沿。
- CPHA=1, 在SCK第二个跳变沿，主机对MISO引脚电平采样。主机的数据发送则在第一个跳边沿。

实现以下四个模式：

Mode0: CPOL=0、CPHA=0: 此时空闲态时，SCK处于低电平，数据采样是在第1个边沿，就是SCK由低电平到高电平的跳变，数据采样是在上升沿，数据发送是在下降沿。

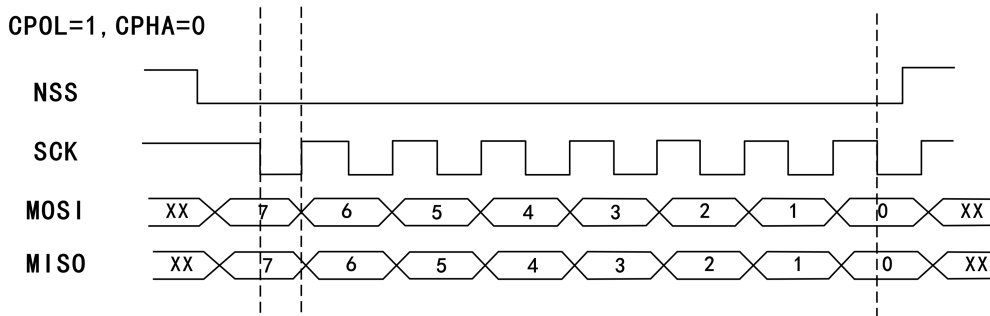


Mode1: CPOL=0, CPHA=1: 此时空闲态时，SCK 处于低电平，数据采样是在第二个边沿，即 SCK 由低电平到高电平的跳变，所以数据采样是在下降沿，数据发送是在上升沿。

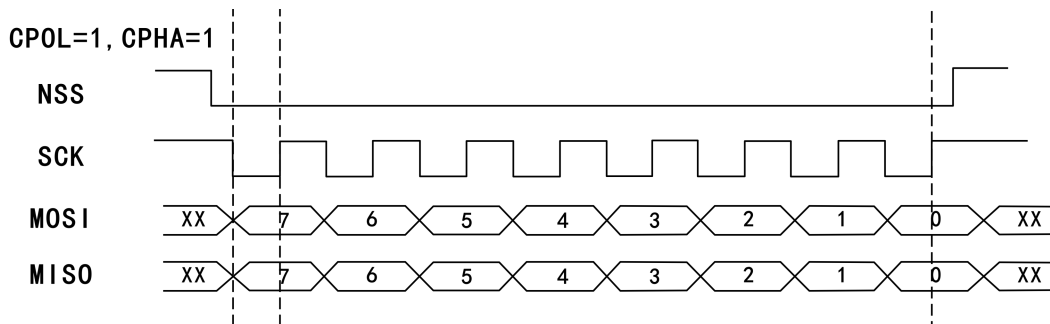




Mode2: CPOL=1, CPHA=0: 此时空闲态时, SCK 处于高电平, 数据采样是在第 1 个边沿, 即 SCK 由高电平到低电平的跳变, 所以数据采样是在下降沿, 数据发送是在上升沿。



Mode3: CPOL=1, CPHA=1: 此时空闲态时, SCK 处于高电平, 数据采样是在第二个边沿, 即 SCK 由低电平到高电平的跳变, 所以数据采样是在上升沿, 数据发送是在下降沿。



## 16.3.2 SPI 发送与接收

SPI主机模式发送数据主要操作如下:

1. 设置SSICON0(R1E3)寄存器, 使能SSIEN, 选择SPI模式, 按需设置时钟分频。
2. 设置SSICON1(R1E4)寄存器, 按需配置映射端口, 设置主机模式, 设置时钟极性和相位。
3. 如需发送完成进入中断, 则使能RX\_SPIIE。
4. 将需要发送的数据存入SSIBUF
5. 设置SSICON1(R1E4)寄存器, 使能RUN\_EN, 开始发送数据。

SPI主机模式接收数据流程:

1. 主设备发送数据的同时, 从设备通过 MISO 线返回响应数据。
2. 如使能RX\_SPIIE, 则接收完成进入中断。
3. 读取 SSIBUF 寄存器可获取接收到的数据。

## 16.4 三合一相关寄存器

### 16.4.1 R1E3/SSICON0 (SSI 控制寄存器 0)

0X1E3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSICON0	SSIEN	MODES<1:0>		SSICKS	SSIPTEN	SSIPSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: SSIEN - SSI 三合一通信模块使能

- 0: 禁止
- 1: 使能

Bit<6:5>: MODES<1:0>-模式选择位

- 00: 未选择
- 01: UART 模式 (UART 默认 8 分频)
- 10: IIC 模式 (IIC 默认 2 分频)
- 11: SPI 模式 (SPI 默认 2 分频)

Bit<4:3>: SSICKS: SSIPTEN - SSI 时钟源选择

- 00: 选择系统时钟 (IIC/SPI 禁止选 SYSCK, 最快支持系统时钟 2 分频)
- 01: 选择分频时钟
- 10: TC4 溢出 (IIC 默认 4 分频)
- 11: TC5 溢出 (IIC 默认 4 分频)

Bit<2:0>: SSIPSR<2:0>-SSI 分频选择控制位:

SSIPSR<2>	SSIPSR<1>	SSIPSR<0>	SSI 分频
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

### 16.4.2 R1E4/SSICON1 (SSI 控制寄存器 1)

0X1E4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSICON1	RUN_EN	START_EN	STOP_EN	MSTR	TEN1	REN1	AA	SSIOS
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: RUN\_EN - 发送开始

- 0: 禁止
- 1: 使能 (UART/SPI 写 BUFF 自动触发)

Bit<6>: START\_EN - I2C 发送 START/SPI 模式下 CPOL

- 0: 禁止
- 1: 使能

CPOL=0, SPI 总线空闲时, 时钟线 SCLK 为低电平。

CPOL=1, SPI 总线空闲时, 时钟线 SCLK 为高电平。

Bit<5>: STOP\_EN - I2C 发送 STOP 使能/9 位 UART 模式使能 (发送 AA) /SPI 模式下 CPHA

- 0: 禁止
- 1: 使能

CPHA=0, 在 SCLK 第一个跳变沿, 主机对 MISO 引脚电平采样; 主机的数据发送则在第二个跳边沿。

CPHA=1, 在 SCLK 第二个跳变沿, 主机对 MISO 引脚电平采样。主机的数据发送则在第一个跳边沿。

Bit<4>: MSTR - 主从机模式选择

- 0: 从机模式
- 1: 主机模式

Bit<3>: TEN - 发送使能控制位 (UART 及 SPI 模式如发送数据需使能)

- 0: 禁止
- 1: 使能

Bit<2>: REN - 接收使能控制位 (UART 及 SPI 模式如接收数据需使能)

- 0: 禁止
- 1: 使能

Bit<1>: AA - I2C 应答允许位/UART bit9 位发送数据/SPI 模式下发送数据取反

- 0: 禁止
- 1: 使能

Bit<0>: SSIOS - SSI 端口映射选择位

0:

通信口	UART	IIC	SPI
P53	TX1A	SDA0	MOSIO
P54	RX1A	SCL0	SCK0
P77	-	-	CS
P74	-	-	MISO

1:

通信口	UART	IIC	SPI
P76	TX1B	SDA1	MOSI1
P75	RX1B	SCL1	SCK1
P77	-	-	CS
P74	-	-	MISO

### 16.4.3 R1E5/I2CADDR (I2C 地址寄存器)

0X1E5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADDR	I2CADDR<6:0>							WR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

### 16.4.4 R1E6/SSIBUF (SSI 收发数据寄存器)

0X1E6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSIBUF	SSIBUF<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

串行发送时，MCU 对 SBUF 写入数据

串行接收时，MCU 从 SBUF 读取数据

写数据和读数据共用地址，读取为外部输入数据，写入数据不可读

### 16.4.5 R1E7/SSISR0 (串口状态寄存器 0)

0X1E7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSISR0	-	BUS_BUSY	MSTR_IDLE	ARB_LOST	DATA_ACK	ADD_ACK	MSTR_ERROR	MSTR_BUSY
读/写	R	R	R	R/W	R/W	R/W	R/W	R
复位值	0	X	1	0	0	0	0	0

Bit<6>: BUS\_BUSY - 主控模式/从控模式下 IIC 总线忙标志位

0: 写 0 清 0

1: IIC 总线忙

Bit<5>: MSTR\_IDLE - IIC 主控模式空闲标志位

0: 工作状态

1: 空闲状态

Bit<4>: ARB\_LOST - I2C 主控模式仲裁标志位

0: 写 0 清除

1: 丢失总线控制权

Bit<3>: DATA\_ACK - I2C 主控模式寻址应答标志位

0: 写 0 清除

1: 最后一次发送数据没有应答

Bit<2>: ADD\_ACK - I2C 主控模式寻址应答标志位

0: 写 0 清除

1: 最后一次寻址没有应答

Bit<1>: MSTR\_ERROR - I2C 主控模式错误标志位

0: 写 0 清除

1: 寻址从机没有应答/发送数据没有应答/I2C 总线仲裁冲突

Bit<0>: MSTR\_BUSY - I2C 主控模块忙标志位

0: 未传输

1: I2C 模块正在传输数据（包含收发）

## 16.4.6 R1E8/SSISR1(串口状态寄存器 1)

0X1E8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSISR1	-	-	TI1	RI1	RBIT8	SENDFIN	TREQ	RREQ
读/写	R	R	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<5>: TI1 - UART 发送完成标志位

0: 写 0 清 0

1: 发送完成

Bit<4>: RI1 - UART 接收完成标志位

0: 写 0 清 0

1: 接收完成

Bit<3>: RBIT8 - UART 接收数据的第 9 位

Bit<2>: SENDFIN - I2C 从机模式下发送完毕标志位

0: 写 0 清 0

1: 主控器件不再需要数据，已经完成本次数据传送

Bit<1>: TREQ - I2C 从机模式准备发送标志

0: 写 SBUF 清 0

1: 作为发送器件已经被寻址或主控器件已经做好接收数据的准备

Bit<0>: RREQ - I2C 从机模式接收完成标志

0: 写 0 清 0

1: 接收完成

## 17 EEPROM 以及 ISP 在线烧录操作

### 17.1 EEPROM 读写操作

JZ8FT8801 内置 256 Byte 16bits EEPROM 数据存储器，出厂默认数据为 0xFFFF，支持用户程序在带电工作中实时地读出或写入数据，写入数据时 MCU 将停止指令运行，待写入结束后继续执行指令。

#### 写 EEPROM:

在写入 EEPROM 单元前，用户必须先将地址写入 TBRDL 寄存器，用于选择 EEPROM 地址，再通过对 EEPROM 数据寄存器 E2PDATL/E2PDATH 进行写值，最后打开 E2PLOCK 解锁 EEPROM 写操作，将 E2PWR 位置 1，一个写过程启动后，将 E2PWR 清零将不会影响写周期，写周期完成时，E2PWR 位将由硬件清零，在下次进行写入前由用户主动打开。

**注意：EEPROM 写期间，指令时钟停止，MCU 不执行任何指令。高速振荡器和低速振荡器正常运行，系统时钟正常；**

#### 读 EEPROM:

要读取 EEPROM 存储器单元，用户必须先将地址写入 TBRDL 寄存器，将 E2PCON 寄存器的 E2PRD 位置 1。使能读控制位后，执行 TBRD 指令，EEPROM 中的数据被存入对应的 R（数据高 8 位）和 ACC（数据低 8 位）中。

### 17.2 ISP 烧录

JZ8FT8801 支持常规烧写、在板带电烧写 2 种不同的烧写方式。在板带电烧写需要用户提前配置好烧写端口以及烧写功能。

#### 1、常规烧写:

OPTION 选项【烧写端口设置】选择【普通烧录/仅在复位时作烧写口（程序可以使用）】，OPTION 选项【烧写功能选择】选择【常规烧写（烧写器供电烧写）】。以上配置不影响烧录端口使用。

#### 2、在板带电烧写:

在板带电烧写需要 MCU 提前选择“常规烧写”烧好程序，不能空片烧录，在板带电烧写模式下烧录器需要接 3 根线 GND、SDA、SCL，VDD 禁止接。Option 选项【ISP 在板带电烧写口】选择在板带电烧写脚 P53/P54 或 P60/P61。支持以下三种烧写方式。

**方式一：仅在复位时作烧写口，程序可以使用，软件解锁**

OPTION 选项【烧写端口设置】选择【普通烧录/仅在复位时作烧写口（程序可以使用）】，OPTION 选项【烧写功能选择】选择【在板带电烧写】。以上配置烧录脚可以做 GPIO 正常使用，程序需要配置特定寄存器并且使能外部中断在烧录脚上接收（端口设为输入）烧录器发送的时钟数据，正确解码后软件解锁 ISP 功能，详细见 ISP demo。

软件解锁 ISP 主要分为以下两步：

- 1) 程序写 0110 到 E2PCON(0x1E0) 寄存器高 4 位 E2PLOCK<3:0>中，完成第一步解锁；
- 2) 程序写 0xC5 到 E2PDATH(0x1E2) 寄存器再写 0x3A 到 E2PDATL(0x1E1) 寄存器，完成第二步解锁；

完成两步解锁后，MCU 硬件将 SDA、SCL 设为输入口，等待外部时序控制 MCU 复位。

**方式二：设为烧写口，程序可以控制，不用软件解锁**

OPTION 选项【烧写端口设置】选择【设为烧写口，程序可以控制】

OPTION 选项【烧写功能选择】选择【在板带电烧写】。

以上配置烧录脚可以做 GPIO 正常使用，程序可以控制端口设为输入或输出功能，但注意在烧录的时候端口必须设为输入，禁止设为输出以免影响烧录。在**外部有上拉或者下拉电阻**的时候建议选择此烧写方式。

**方式三：仅作烧写口，程序不可控**

OPTION 选项【烧写端口设置】选择【仅作烧写口（强制上拉），程序无法控制，芯片可随时切入烧写模式】或【仅作烧写口（强制下拉），程序无法控制，芯片可随时切入烧写模式】

OPTION 选项【烧写功能选择】选择【在板带电烧写】。

以上配置**烧录脚需要悬空**，若需要 Type-C 口升级，则接 CC1/CC2 端口。

**注：无论使用哪种在板带电烧写方式，烧写时烧写口都必须为输入状态。**

## 17.3 EEPROM 与 ISP 相关寄存器

### 17.3.1 R187/TBRDL (EEPROM 地址寄存器)

0X187	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDL	RBIT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: RBIT<7:0> - TBRD 指针地址低 8 位。

RBIT <7:0> 在 EEPROM 读写中用于选择 EEPROM 地址。

### 17.3.2 R1E0/E2PCON (EEPROM 控制寄存器)

0X1E0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E2PCON	E2PLOCK<3:0>				-	E2PTIME	E2PRS	E2PWREN
读/写	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:4>: E2PLOCK<3:0> - EEPROM 功能解锁位

1001: 打入数据 **1001** 解锁 EEPROM 写操作

0110: 打入数据 **0110** 解锁 ISP 功能

Bit<3>: 未使用

Bit<2>: E2PTIME - EEPROM 写时间选择位

0: EEPROM 写时间选择快档位

1: EEPROM 写时间选择慢档位

Bit<1>: E2PRS - EEPROM 区域选择位

0: MTP ROM

1: EEPROM

Bit<0>: E2PWREN - EEPROM 写使能位

0: 禁止

1: 使能

### 17.3.3 R1E1/E2PDATL (EEPROM 数据低 8 位寄存器)

0X1E1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E2PDATL	E2PDAT<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

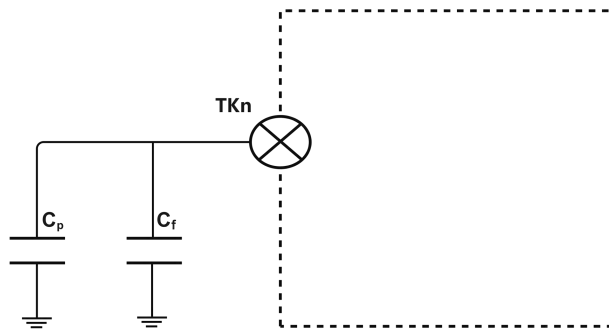
### 17.3.4 R1E2/E2PDATH (EEPROM 数据高 8 位寄存器)

0X1E2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
E2PDATH	E2PDAT<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

## 18 CDC 触摸按键检测

### 18.1 触摸检测概述

JZ8FT8801 的触摸按键检测基于自电容感应原理，当手指触摸按键表面，人体的导电性质和大质量构成了接地的平行板电容，如触摸按键说明示意图所示， $C_p$  为按键电容（包含 PCB、导线、触摸板等寄生电容）， $C_f$  为手指触摸后的等效电容。芯片通过 CDC 转换电容值为数字信号，判断数值的变化以确认按键行为。



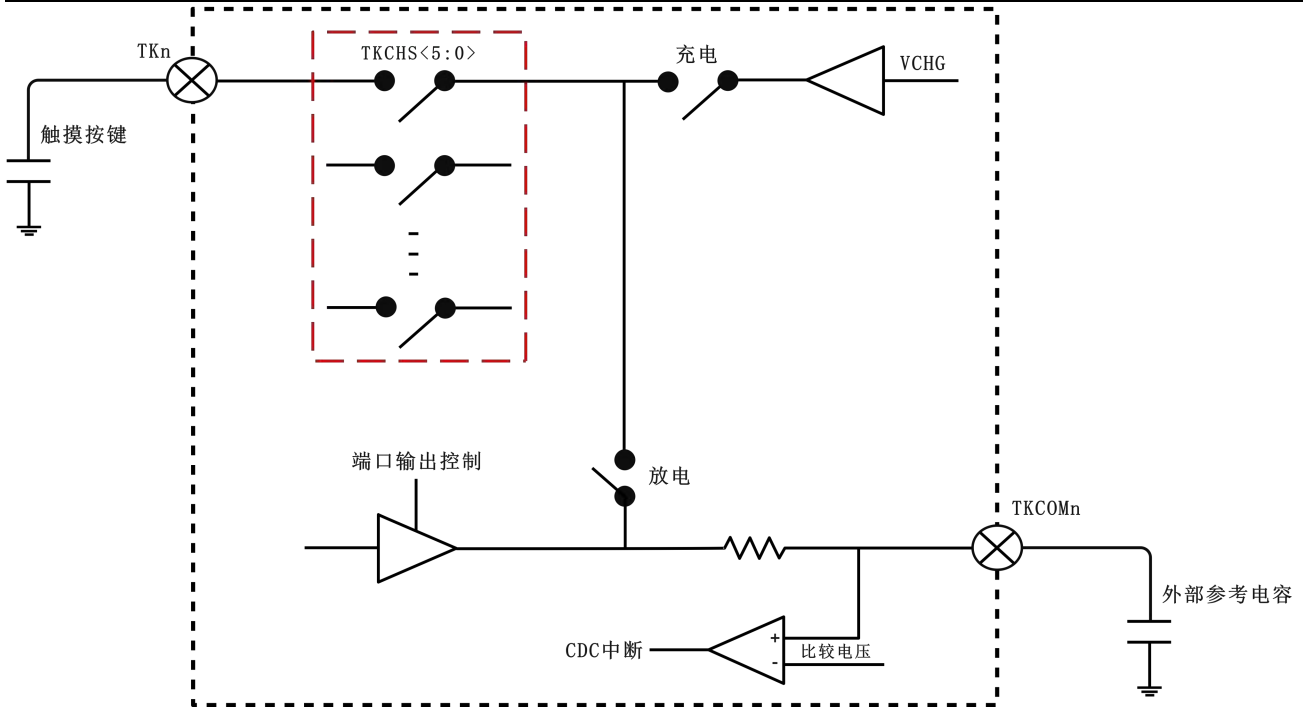
触摸按键说明

芯片内部集成有 Bandgap，四级 LDO（4V/3V/2V/1.5V），CDC 专用振荡器，提供两种触摸检测模式。

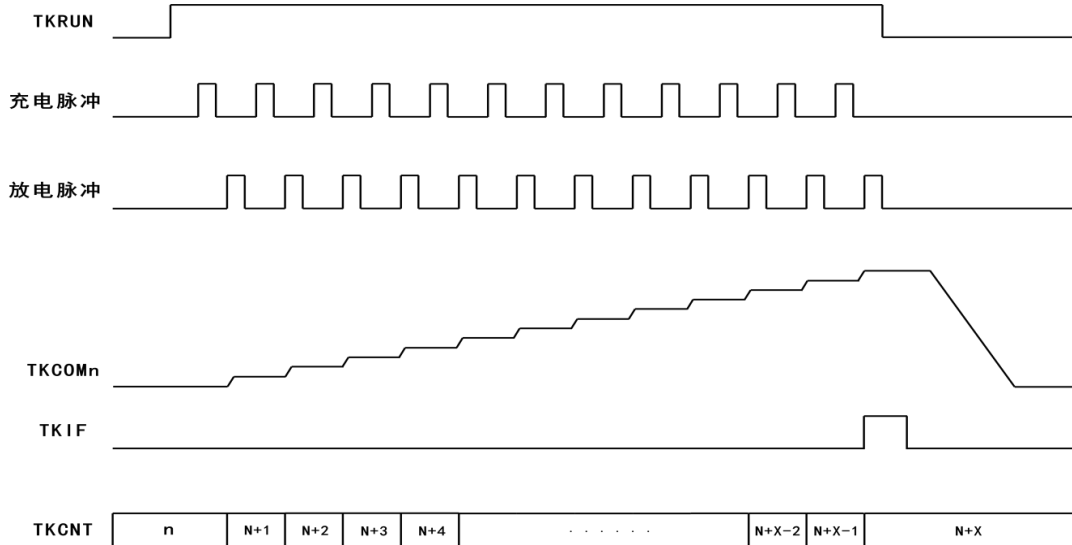
### 18.2 CDC 结构与时序

JZ8FT8801 内部集成 CDC（Capacitance-to-Digital Converter）电容数字转换器用于触摸按键的检测。芯片提供两种触摸检测模式分别为基于电荷转移模式和基于振荡模式。

电荷转移模式是一种精确度好、可配置性灵活的检测模式。电荷转移模式下 TKCOM 端口需外接电容，TK 计数器时钟源为系统时钟，可通过 TKCKS<2:0>选择 CDC 转换时钟分频。CDC 充电电压 VCHG 可通过 TKVS<2:0>选择，分别可选 VDD、4V（LDO）、3V（LDO）、2V（LDO）、1V（LDO）。CDC 比较电压可通过 TKVLS<4:3>选择，分别可选 1.2V（Bandgap）、0.5\*VCHG、0.75\*VCHG、0.9\*VCHG。电荷转移模式通过先给触摸按键充电，然后放电给外部参考电容，TK 计数器同步计数，通过比较外部参考电容电压与比较电压，产生 CDC 中断，此时硬件自动停止 TK 计数器计数，程序控制 IO 口将 TKCOM 端口电压放电至地。通过读取 TK 计数器数值判断触摸按键电容值的变化，以此判断按键行为是否发生。具体功能结构及时序如下图所示：

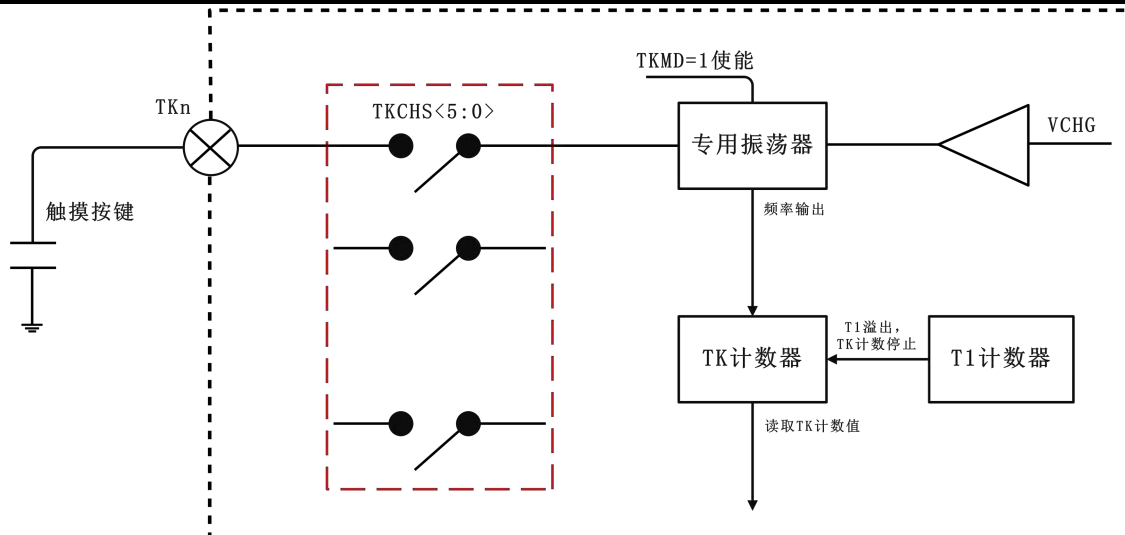


电荷转移模式结构说明



电荷转移模式时序说明

振荡模式为一种简单易操作的检测模式，但是可配置性及精确度不如电荷转移模式。振荡模式下 TKCOM 端口无需外接电容，TK 计数器时钟源为专用振荡器时钟，专用振荡器与触摸按键检测端口相连，当发生触摸按键行为，按键电容发生变化，专用振荡器频率同步发生改变，TC0 定时器溢出时 TK 计数器停止计数，此时判断 TK 计数器的计数值以判断按键行为是否发生。该模式可设置为自动检测方式，TC0EN 使能时启动采样，TC0 定时器溢出后停止。具体功能结构如下图所示：



振荡模式结构说明

## 18.3 CDC 触摸检测设置说明

### 电荷转移模式：

- 1、选择模式为电荷转移模式；
- 2、配置上拉充电电压 VCHG；
- 3、配置比较电压；
- 4、使能 CDC 模块 TKEN=1；
- 5、配置 CDC 相关通道及转换时钟；
- 6、设置 TKCOM 管脚输出 0，对外接电容放电；
- 7、设置 TKCOM 管脚为输入模式；
- 8、清除触摸按键计数寄存器（TKCNTH、TKCNTL）；
- 9、启动 CDC 通道转换（TKRUN=1）；
- 10、等待转换完成（TKRUN=0）或 CDC 中断触发；
- 11、读取触摸按键计数寄存器（TKCNTH、TKCNTL），判断是否有按键行为发生；
- 12、重复 4-11 对不同通道进行扫描；

### 振荡模式：

- 1、选择模式为振荡模式；
- 2、配置上拉充电电压 VCHG（TKVS<2:0>），作为振荡器参考电压；
- 3、配置 TK 自动采样（TKASET）；
- 4、使能 CDC 模块 TKEN=1；
- 5、启动 CDC 通道转换（TKRUN=1）；
- 6、配置 CDC 相关通道；

- 7、配置 TC0 定时器相关时钟、定时时间、中断使能；
- 8、使能 TC0 定时器（TCCEN=1）；
- 9、等待 TC0 定时器溢出，TC0 中断触发；
- 10、读取触摸按键计数寄存器（TKCNTH、TKCNTL），判断是否有按键行为发生；
- 11、重复 6-10 对不同通道进行扫描；

### 18.3.1 R1E9/TKCON(TK 控制寄存器)

0X1E9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCON	TKRUN	TKEN	TKTPEN	TKASET	TKCKS	TKPSR<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7>: TKRUN - CDC 通道转换启动控制位

- 0: 转换完成，自动清零
- 1: 启动转换

Bit<6>: TKEN - TK 模块使能控制位

- 0: 禁止
- 1: 使能

Bit<5>: TKTPEN - TK 跳频功能使能

- 0: 禁止
- 1: 使能

Bit<4>: TKASET - TK 自动采样控制位

- 0: 禁止触摸采样自动启停
- 1: 使能触摸采样自动启停，TCOEN=1 时启动触摸采样，TCO 溢出时停止采样

Bit<3>: TKCKS - TK 时钟源选择

- 0: 系统时钟
- 1: 低速时钟

Bit<2:0>: TKPSR<2:0> - CDC 转换时钟分频选择位

KPSR<2>	KPSR<1>	KPSR<0>	时钟分频选择
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

## 18.3.2 R1EA/TKCHS0 (触摸通道使能控制寄存器 0)

0X1EA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCHS0	TKCHS<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TKCHS<7:0> - TK 输入口使能控制位

0: 禁止 TKCHS<7:0>通道高位开始依次对应

1: 使能 TKCHS<7:0>通道高位开始依次对应

P71、P72、P75、P76、P77、P50、P51、P52 为模拟输入口

## 18.3.3 R1EB/TKCHS1 (触摸通道使能控制寄存器 1)

0X1EB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCHS1	-	-	-	-	TKCHS<11:8>			
读/写	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:4>: 未使用

Bit<3:0>: TKCHS<11:8> - TK 输入口使能控制位

0: 禁止 TKCHS<11:8>通道

1: 使能 TKCHS<11:8>通道高位开始依次对应 P65、P66、P67、P70 为模拟输入口

## 18.3.4 R1EC/TKVS (TK 电压选择寄存器)

0X1EC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKVS	TKCAPS<2:0>			TKVLS<1:0>		TKVS<2:0>		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:5>: TKCAPS<2:0> - 触摸通道并联电容选择

TKCAPS <2>	TKCAPS <1>	TKCAPS <0>	并联电容
0	0	0	2.8pf
0	0	1	2.4pf
0	1	0	2.0pf
0	1	1	1.6pf
1	0	0	1.2pf
1	0	1	0.8pf
1	1	0	0.4pf
1	1	1	0pf



Bit<4:3>: TKVLS<1:0> - 比较电压选择位

KVLS<1>	KVLS<0>	比较电压
0	0	0.5*VCHG
0	1	0.625*VCHG
1	0	0.75*VCHG
1	1	0.875*VCHG

Bit<2:0>: TKVS<2:0> - 上拉充电电压选择位

KVS<2>	KVS<1>	KVS<0>	上拉充电电压 VCHG
0	0	0	1.2V
0	0	1	2V
0	1	0	3V
0	1	1	4V
1	0	0	VDD

### 18.3.5 R1ED/TKCNTH(TK 计数器高位寄存器)

0X1ED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCNTH	TKC<15:8>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TKC<15:8> - TK 计数器高 8 位

### 18.3.6 R1EE/TKCNTL(TK 计数器低位寄存器)

0X1EE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCNTL	TKC<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit<7:0>: TKC<7:0> - TK 计数器低 8 位

## 19 OPTION 配置表

CODE OPTION	选项	功能描述
看门狗	使能	看门狗 WDT 使能
	禁止	看门狗 WDT 禁止
Clocks 分频	2 Clocks	指令时钟分频选择 2 Clocks
	4 Clocks	指令时钟分频选择 4 Clocks
	8 Clocks	指令时钟分频选择 8 Clocks
	16 Clocks	指令时钟分频选择 16 Clocks
	32 Clocks	指令时钟分频选择 32 Clocks
振荡模式	IRC 模式	主时钟选择内部 IHRC 振荡器
	LXT 模式	主时钟选择外部低速晶体振荡器
	HXT 模式	主时钟选择外部高速晶体振荡器
IRC 频率	4M	IRC 频率选择 4M
	8M	IRC 频率选择 8M
	16M	IRC 频率选择 16M
	29.4M	IRC 频率选择 29.4M
	32M	IRC 频率选择 32M
低压复位选项	使能: LVR always	默认 LVR 功能一直使能
	使能: 软件控制	LVR 使能位软件可改
低压复位	LVR=1.8V	低压复位点选择 1.8V
	LVR=2.0V	低压复位点选择 2.0V
	LVR=2.2V	低压复位点选择 2.2V
	LVR=2.5V	低压复位点选择 2.5V
	LVR=3.0V	低压复位点选择 3.0V
	LVR=3.4V	低压复位点选择 3.4V
	LVR=4.0V	低压复位点选择 4.0V
复位时间	PWRT=3.5ms, WDT=6.4ms	复位建立时间=3.5ms, WDT 溢出时间 (4 分频) = 6.4ms
	PWRT=13ms, WDT=25ms	复位建立时间=13ms, WDT 溢出时间 (4 分频) = 25ms
	PWRT=51ms, WDT=102ms	复位建立时间=51ms, WDT 溢出时间 (4 分频) = 102ms
	PWRT=205ms, WDT=409ms	复位建立时间=205ms, WDT 溢出时间 (4 分频) = 409ms
	PWRT=350us, WDT=6.4ms	复位建立时间=350us, WDT 溢出时间 (4 分频) = 6.4ms
	PWRT=350us, WDT=25ms	复位建立时间=350us, WDT 溢出时间 (4 分频) = 25ms
	PWRT=350us, WDT=102ms	复位建立时间=350us, WDT 溢出时间 (4 分频) = 102ms
	PWRT=350us, WDT=409ms	复位建立时间=350us, WDT 溢出时间 (4 分频) = 409ms
RTC	使能	RTC 功能 使能
	禁止	RTC 功能 禁止
P55 端口	as RESET	P55 端口作 外部复位口
	as GPIO	P55 端口作 通用输入输出口



P55 复位口 外部上拉	使能	P55 复位口外部上拉功能 使能
	禁止	P55 复位口外部上拉功能 禁止
CMP0 迟滞	无迟滞	比较器 0 结果无迟滞
	20mV	比较器 0 结果迟滞电压为 20mV
	40mV	比较器 0 结果迟滞电压为 40mV
	60mV	比较器 0 结果迟滞电压为 60mV
普通端口 拉电流增强	使能	普通端口拉电流增强功能 使能
	禁止	普通端口拉电流增强功能 禁止
LED 端口 灌电流增强	使能	LED 端口灌电流增强功能 使能
	禁止	LED 端口灌电流增强功能 禁止
LED 端口 拉电流一级增强	使能	LED 端口拉电流一级增强功能 使能
	禁止	LED 端口拉电流一级增强功能 禁止
LED 端口 拉电流二级增强	使能	LED 端口拉电流二级增强功能 使能
	禁止	LED 端口拉电流二级增强功能 禁止
封装脚位	16pin	芯片封装脚位选择 16 脚
	20pin	芯片封装脚位选择 20 脚
烧录模式选择	FLASH	烧录模式选择 FLASH
	EEPROM	烧录模式选择 EEPROM
	FLASH + EEPROM	烧录模式选择 FLASH + EEPROM
烧录端口设置	普通烧录/仅在复位时作烧写口 (程序可以用)	常规烧录或 ISP 软件解锁烧写口, 程序可以控制使用端口, 但在烧录时端口必须为输入状态
	设为烧写口, 程序可以控制	ISP 在板烧录, 不用软件解锁烧写口, 程序可以控制使用端口, 但在烧录时端口必须为输入
	仅作烧写口(强制下拉), 程序无法控制, 芯片可随时切入烧写模式	ISP 在板烧写口只作为烧录口并强制下拉, 程序不可以控制
	仅作烧写口(强制上拉), 程序无法控制, 芯片可随时切入烧写模式	ISP 在板烧写口只作为烧录口并强制上拉, 程序不可以控制
烧写功能选择	常规烧写(烧写器供电烧写)	常规烧写(烧写器供电烧写)
	在板带电烧写	在板带电烧写
ISP 在板带电烧写 口	P53/P54	P53/P54 作为在板带电烧写口
	P60/P61	P60/P61 作为在板带电烧写口
OS 测试	使能	OS 测试使能
	禁止	OS 测试禁止
上电时间选择	快上电	电源与地之间的电容 $\leq 50\mu\text{F}$
	慢上电	$50\mu\text{F} \leq$ 电源与地之间的电容 $\leq 470\mu\text{F}$

## 20 指令集

指令	指令动作	标志位影响
ADD A, R	$A+R \rightarrow A$	Z, C, DC
ADD R, A	$A+R \rightarrow R$	Z, C, DC
AND A, R	$A \& R \rightarrow A$	Z
AND R, A	$A \& R \rightarrow R$	Z
CLRA	$0 \rightarrow A$	Z
CLR R	$0 \rightarrow R$	Z
INVA R	$\neg R \rightarrow A$	Z
INV R	$\neg R \rightarrow R$	Z
DA	A 寄存器调整为 BCD 值	C
DECA R	$R-1 \rightarrow A$	Z
DEC R	$R-1 \rightarrow R$	Z
DJA R	$R-1 \rightarrow A$ , skip if zero	-
DJ R	$R-1 \rightarrow R$ , skip if zero	-
INCA R	$R+1 \rightarrow A$	Z
INC R	$R+1 \rightarrow R$	Z
IJA R	$R+1 \rightarrow A$ , skip if zero	-
IJ R	$R+1 \rightarrow R$ , skip if zero	-
MOV R, A	$A \rightarrow R$	-
MOV A, R	$R \rightarrow A$	Z
MOV R, R	$R \rightarrow R$	Z
OR A, R	$A \vee R \rightarrow A$	Z
OR R, A	$A \vee R \rightarrow R$	Z
SUB A, R	$R-A \rightarrow A$	Z, C, DC
SUB R, A	$R-A \rightarrow R$	Z, C, DC
XOR A, R	$A \oplus R \rightarrow A$	Z
XOR R, A	$A \oplus R \rightarrow R$	Z
BTC R, b	$0 \rightarrow R(b)$	-
BTS R, b	$1 \rightarrow R(b)$	-
JBTC R, b	if $R(b)=0$ , skip	-
JBTS R, b	if $R(b)=1$ , skip	-



LCR R	$R(n) \rightarrow R(n+1), R(7) \rightarrow C, C \rightarrow R(0)$	C
LCA R	$R(n) \rightarrow A(n+1), R(7) \rightarrow C, C \rightarrow A(0)$	C
RCR R	$R(n) \rightarrow R(n-1), R(0) \rightarrow C, C \rightarrow R(7)$	C
RCA R	$R(n) \rightarrow A(n-1), R(0) \rightarrow C, C \rightarrow A(7)$	C
SWAP R	$R(0-3) \leftrightarrow R(4-7)$	-
SWAPA R	$R(0-3) \rightarrow A(4-7), R(4-7) \rightarrow A(0-3)$	-
ADD A, k	$A+k \rightarrow A$	Z, C, DC
AND A, k	$A \& k \rightarrow A$	Z
MOV A, k	$k \rightarrow A$	-
OR A, k	$A \vee k \rightarrow A$	Z
SUB A, k	$k-A \rightarrow A$	Z, C, DC
XOR A, k	$A \oplus k \rightarrow A$	Z
CALL k	$PC+1 \rightarrow [SP], (Page, k) \rightarrow PC$	-
DI	禁止中断	-
EI	使能中断	-
JMP k	$K (Page, k) \rightarrow PC$	-
NOP	空指令	-
RET	$[堆栈顶端] \rightarrow PC$	-
RETI	$[堆栈顶端] \rightarrow PC, 使能中断$	-
RETL k	$k \rightarrow A, [堆栈顶端] \rightarrow PC$	-
SLEEP	$0 \rightarrow WDT, 振荡器停止振荡 (睡眠模式)$	T, P
CWDT	$0 \rightarrow WDT$	T, P
TBRD R	机器码 bit7~0 给 ACC 机器码 bit15~8 给 R (通用寄存器)	-

## 21 电气特性

### 21.1 极限参数

工作温度.....	-40°C~85°C
存储温度.....	-65°C~150°C
输入电压.....	V <sub>SS</sub> -0.3V~V <sub>DD</sub> +0.5V
输出电压.....	V <sub>SS</sub> -0.3V~V <sub>DD</sub> +0.5V
工作电压.....	1.8V~5.5V

### 21.2 直流电气特性

(V<sub>DD</sub>=5V, 工作温度=25°C, 除非另有情况说明)

工作特性

符号	参数	最小	典型	最大	单位	条件
F <sub>osc</sub> (SYSCK)	系统时钟	-	32	-	MHz	频率值通过 option 【IRC 频率】选择 (修调误差±3%)
		-	16	-	MHz	
		-	8	-	MHz	
		-	4	-	MHz	
F <sub>cpu</sub>	指令时钟	2T	-	8	MHz	V <sub>DD</sub> =3.4V~5.5V
		4T/8T	-	4	MHz	V <sub>DD</sub> =2.6V~5.5V
		16T/32T	-	2	MHz	V <sub>DD</sub> =2.0V~5.5V
T <sub>PWRT</sub>	上电复位保持时间	-	350	-	μs	PWRT 独立时间
		复位时间=WDT 溢出时间			ms	PWRT=WDT/2
T <sub>RST</sub>	外部复位脉冲宽度	2	-	-	μs	25°C
T <sub>UIS</sub>	三合一模块工作电压	2.6	-	-	V	25°C

### 低电压复位 (LVR)

符号	参数说明	最小	典型	最大	单位
I <sub>LVR</sub>	复位电流 (LVR=1.8V)	-	0.6	-	μA
V <sub>LVR</sub>	LVR 阈值	-10%	1.8	+10%	V
		-10%	2.0	+10%	
		-10%	2.2	+10%	
		-10%	2.6	+10%	
		-15%	3.0	+15%	
		-15%	3.4	+15%	
		-15%	4.0	+15%	
		-15%	4.6	+15%	

**注：为避免工艺波动，实际产品请留好余量**

## I/O 端口电路

符号	参数	最小	典型	最大	单位	条件	
VIL	低电平阈值	0	-	0.25*VDD	V	25°C	
VIH	高电平阈值	0.53*VDD	-	VDD	V	25°C	
I <sub>OH1</sub>	源电流 (LED 端口)	L0	7	8	9	mA	25°C, VDD=5V, VOH=4.4V
		L1	14	16	18		
		L2	26	28	30		
I <sub>OH2</sub>	源电流 (普通端口)	L0	14	16	18	mA	
		L1	26	28	30		
I <sub>OH3</sub>	源电流 (大电流口)	L0	25	27	29		
		L1	49	50	52		
		L2	78	80	82		
		L3	104	106	108		
I <sub>OL1</sub>	灌电流 (普通端口/LED 口)	L0	29	33	35	mA	25°C, VDD=5V, VOL=0.6V
		L1 (LED 端口)	59	62	65		
I <sub>OL2</sub>	灌电流 (大电流口)	L0	27	29	31		
		L1	53	55	57		
		L2	85	87	89		
		L3	118	120	123		
R <sub>PH</sub>	上拉电阻 (P5/P6/P7)	-	50	-	KΩ	25°C, VDD=5V	
R <sub>PL</sub>	下拉电阻 (P5/P6/P7)	-	50	-	KΩ	25°C, VDD=5V	

普通端口: P56~P57、P70~P77、P80~P85

LED 端口: P50~P55、P64~P67

大电流口: P60~P63

## 工作电流 (IDD)

参数	SYSCK	典型值@ VDD			单位
		2.0V	3.0V	5.0V	
正常模式 (4T)	32 MHz	0.96	1.4	2.19	mA
	16 MHz	0.5	0.74	1.12	
	8 MHz	0.26	0.39	0.59	
	4 MHz	0.16	0.23	0.34	
	40 KHz	5.5	6.4	8.6	uA
Sleep 模式 (WDT off, LVR on)	-	0.3	0.4	0.9	uA
Sleep 模式 (WDT on, LVR on)	-	3.9	5.4	8.5	
Sleep 模式 (WDT off, LVR on, ADC on)	-	20	34	95	
Sleep 模式 (WDT off, LVR on, CMPO on)	-	5	5.5	8	

## MTP 和 EEPROM

参数		最小值	典型值	最大值	单位	条件
VDD-READ	MTP/EEPROM	1.8	-	5.5	V	25°C, F <sub>CPU</sub> ≤ 8MHz
VDD-WRIT	MTP	2.1	-	5.5	V	25°C
	EEPROM	2.1	-	5.5	V	25°C
Endurance	MTP/EEPROM	-	10000	-	次	25°C
Retention	MTP/EEPROM	-	10	-	年	25°C
T <sub>WRIT</sub> 快写时间	MTP/EEPROM	-	4.5	-	ms	25°C
T <sub>WRIT</sub> 慢写时间	MTP/EEPROM	-	7.5	-	ms	25°C
TERASE 擦除时间	MTP/EEPROM	-	1.5	-	ms	25°C
IP/E 擦写电流	MTP/EEPROM	-	0.8	-	mA	25°C, VDD=5.5V

## 21.3 AD 转换特性

(V<sub>DD</sub>=5V, V<sub>SS</sub>=0V, 工作温度=25°C)

符号	参数说明	条件	最小	典型	最大	单位
V <sub>DD</sub>	ADC 工作电压	-	2.5	-	5.5	V
I <sub>VDD</sub>	ADC 工作电流	VDD=V <sub>REF</sub> =5V	-	95	-	uA
V <sub>AIN</sub>	模拟输入电压	-	V <sub>SS</sub>	-	V <sub>REF</sub>	V
EXVREF	外部参考电压	-	2.0	-	V <sub>DD</sub>	V
V <sub>REF</sub>	参考电压输出	VBG, 2V, 3V, 4V	-	±1	±2	%
T <sub>AD</sub>	转换时钟周期	VDD=2.5~5.5V	1	-	-	us
T <sub>CN</sub>	AD 转换时间	VDD=2.5~5.5V	-	16	-	T <sub>AD</sub>
RN	分辨率	-	-	12	-	Bit
VDDmin	最小供电电压	-	-	VREF+0.5	-	V
E <sub>IL</sub>	积分误差	V <sub>DD</sub> =V <sub>REF</sub> =5V	-	±8.0	-	LSB
E <sub>DL</sub>	微分误差		-	±4.0	-	LSB
E <sub>OFF</sub>	偏移误差		-	±5.0	-	LSB
E <sub>GN</sub>	增益误差		-	±2.0	-	LSB

**注：误差由评估得出，不在生产中测试**

## 21.4 比较器特性

(V<sub>DD</sub>=5V, V<sub>SS</sub>=0V, 工作温度=25°C)

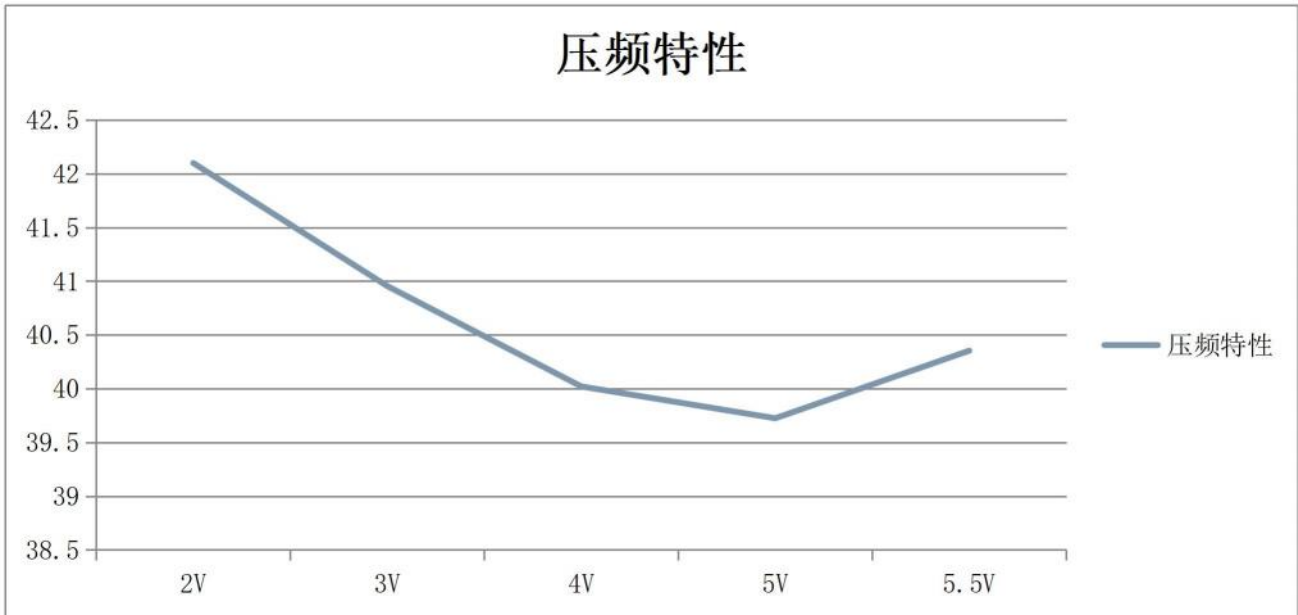
符号	参数	最小	典型	最大	单位	条件
V <sub>DD</sub>	工作电压范围	1.8	-	5.5	V	16M/4T
V <sub>OS</sub>	比较器迟滞电压	-	20	-	mV	16M/4T
		-	40	-	mV	
		-	60	-	mV	
Tr	比较器响应时间	-	392	-	ns	16M/4T
		-	504	-	ns	不开数字滤波
		-	48	-	us	16M/4T

## 21.5 特性曲线图

本章所列的曲线图仅作设计参考,其中给出的部分数据可能超出了芯片指定的工作范围,为保证芯片的正常工作,请严格参照电气特性说明。

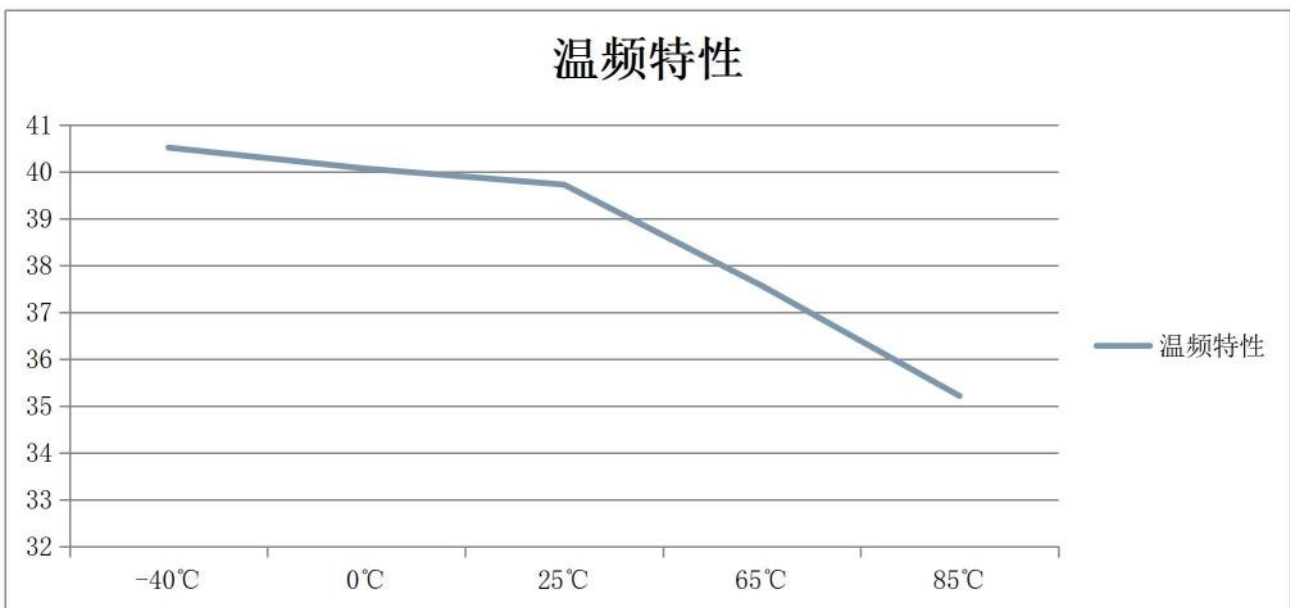
### 21.5.1 内部低速振荡器-压频特性曲线

工作温度在 25°C 条件下: (单位 KHz)



### 21.5.2 内部低速振荡器-温频特性曲线

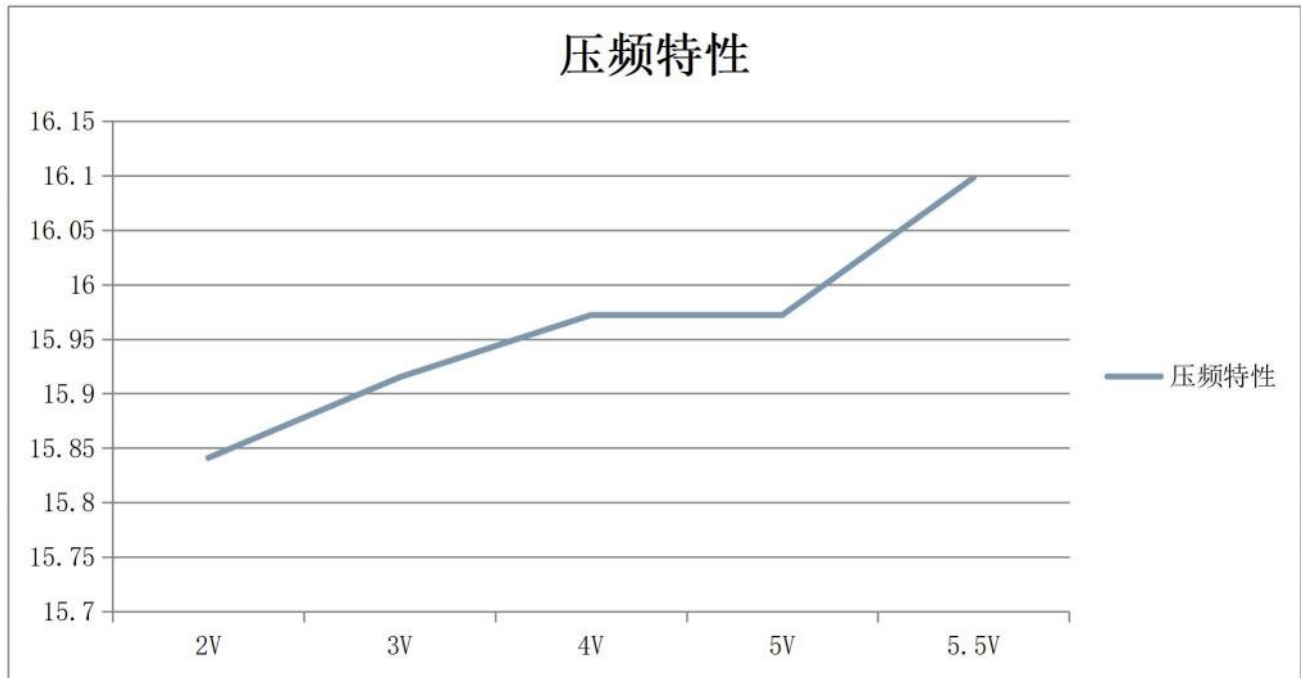
工作电压在 5V 条件下: (单位 KHz)





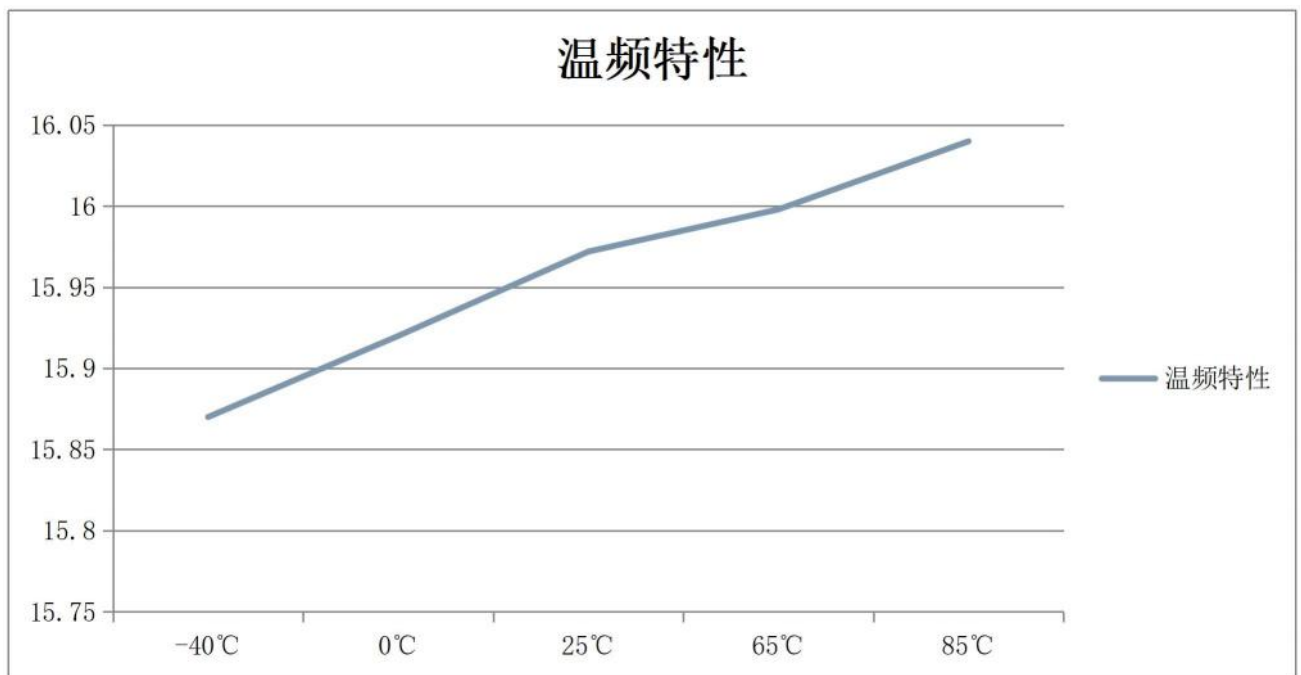
### 21.5.3 内部 16Mhz RC 振荡器-压频特性曲线

工作温度在 25°C 条件下：(单位 MHz)



### 21.5.4 内部 16Mhz RC 振荡器-温频特性曲线

工作电压在 5V 条件下：(单位 MHz)



## 21.6 IHRC 频率微调参数说明

参数说明仅供作变化趋势参考，实际有偏差。

IHRCCAL 修 调值 (HEX)	频率 (MHz)	修调偏差	IHRCCAL 修 调值 (HEX)	频率 (MHz)	修调偏差
00	7.8685	-51.27%	40	10.5419	-34.71%
01	7.8989	-51.08%	41	10.6021	-34.34%
02	7.9306	-50.88%	42	10.6584	-33.99%
03	7.9615	-50.69%	43	10.7153	-33.63%
04	7.9946	-50.48%	44	10.7758	-33.26%
05	8.0244	-50.30%	45	10.8314	-32.91%
06	8.0567	-50.10%	46	10.8925	-32.54%
07	8.0900	-49.89%	47	10.9519	-32.17%
08	8.1230	-49.69%	48	11.0149	-31.78%
09	8.1554	-49.49%	49	11.0798	-31.38%
0A	8.1911	-49.27%	4A	11.1415	-30.99%
0B	8.2221	-49.08%	4B	11.2065	-30.59%
0C	8.2612	-48.83%	4C	11.2811	-30.13%
0D	8.2928	-48.64%	4D	11.3397	-29.77%
0E	8.3260	-48.43%	4E	11.4024	-29.38%
0F	8.3594	-48.23%	4F	11.4741	-28.93%
10	8.3979	-47.99%	50	11.5460	-28.49%
11	8.4313	-47.78%	51	11.6136	-28.07%
12	8.4694	-47.54%	52	11.6864	-27.62%
13	8.5057	-47.32%	53	11.7510	-27.22%
14	8.5400	-47.11%	54	11.8237	-26.77%
15	8.5790	-46.87%	55	11.8957	-26.32%
16	8.6136	-46.65%	56	11.9714	-25.85%
17	8.6503	-46.42%	57	12.0446	-25.40%
18	8.6891	-46.18%	58	12.1222	-24.92%
19	8.7255	-45.96%	59	12.1995	-24.44%
1A	8.7629	-45.73%	5A	12.2718	-23.99%
1B	8.8015	-45.49%	5B	12.3573	-23.46%
1C	8.8443	-45.22%	5C	12.4258	-23.04%
1D	8.8817	-44.99%	5D	12.5138	-22.50%
1E	8.9246	-44.73%	5E	12.5884	-22.03%
1F	8.9603	-44.50%	5F	12.6776	-21.48%
20	9.0002	-44.26%	60	12.7655	-20.94%
21	9.0346	-44.04%	61	12.8476	-20.43%
22	9.0782	-43.77%	62	12.9325	-19.90%
23	9.1206	-43.51%	63	13.0183	-19.37%
24	9.1629	-43.25%	64	13.1138	-18.78%
25	9.2062	-42.98%	65	13.2061	-18.21%

26	9.2536	-42.69%	66	13.2976	-17.64%
27	9.2929	-42.44%	67	13.3926	-17.05%
28	9.3373	-42.17%	68	13.4878	-16.46%
29	9.3862	-41.87%	69	13.5799	-15.89%
2A	9.4265	-41.62%	6A	13.6794	-15.28%
2B	9.4737	-41.32%	6B	13.7778	-14.67%
2C	9.5194	-41.04%	6C	13.8836	-14.01%
2D	9.5634	-40.77%	6D	13.9823	-13.40%
2E	9.6137	-40.46%	6E	14.0871	-12.75%
2F	9.6657	-40.13%	6F	14.1931	-12.09%
30	9.7111	-39.85%	70	14.3019	-11.42%
31	9.7610	-39.54%	71	14.4097	-10.75%
32	9.8052	-39.27%	72	14.5217	-10.06%
33	9.8555	-38.96%	73	14.6339	-9.36%
34	9.9056	-38.65%	74	14.7487	-8.65%
35	9.9545	-38.35%	75	14.8661	-7.93%
36	10.0069	-38.02%	76	14.9850	-7.19%
37	10.0607	-37.69%	77	15.1078	-6.43%
38	10.1097	-37.38%	78	15.2292	-5.68%
39	10.1618	-37.06%	79	15.3525	-4.91%
3A	10.2126	-36.75%	7A	15.4803	-4.12%
3B	10.2720	-36.38%	7B	15.6073	-3.34%
3C	10.3227	-36.07%	7C	15.7431	-2.49%
3D	10.3786	-35.72%	7D	15.8757	-1.67%
3E	10.4337	-35.38%	7E	16.0069	-0.86%
3F	10.4836	-35.07%	7F	16.1458	0.00%

## 21.7 DAC 电压参数

(VDD=5V, VSS=0V, 工作温度=25°C)

VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V
0x00	1.208	0x10	1.4335	0x20	1.6615	0x30	1.886
0x01	1.2205	0x11	1.4465	0x21	1.674	0x31	1.899
0x02	1.235	0x12	1.461	0x22	1.6885	0x32	1.913
0x03	1.247	0x13	1.473	0x23	1.7005	0x33	1.9255
0x04	1.263	0x14	1.4885	0x24	1.7165	0x34	1.942
0x05	1.276	0x15	1.501	0x25	1.729	0x35	1.954
0x06	1.2895	0x16	1.515	0x26	1.743	0x36	1.968
0x07	1.3025	0x17	1.5275	0x27	1.7555	0x37	1.98
0x08	1.321	0x18	1.546	0x28	1.774	0x38	1.998
0x09	1.333	0x19	1.558	0x29	1.7865	0x39	2.011
0x0A	1.3475	0x1A	1.573	0x2A	1.801	0x3A	2.025
0x0B	1.36	0x1B	1.585	0x2B	1.813	0x3B	2.0375
0x0C	1.375	0x1C	1.6	0x2C	1.8285	0x3C	2.053
0x0D	1.388	0x1D	1.613	0x2D	1.841	0x3D	2.065
0x0E	1.402	0x1E	1.627	0x2E	1.855	0x3E	2.079
0x0F	1.414	0x1F	1.639	0x2F	1.8675	0x3F	2.0915

VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V
0x40	2.099	0x50	2.325	0x60	2.5515	0x70	2.777
0x41	2.112	0x51	2.337	0x61	2.564	0x71	2.7895
0x42	2.127	0x52	2.352	0x62	2.579	0x72	2.8045
0x43	2.139	0x53	2.364	0x63	2.591	0x73	2.816
0x44	2.1545	0x54	2.38	0x64	2.607	0x74	2.832
0x45	2.167	0x55	2.393	0x65	2.619	0x75	2.8445
0x46	2.1815	0x56	2.4065	0x66	2.6335	0x76	2.8585
0x47	2.194	0x57	2.4185	0x67	2.6455	0x77	2.8705
0x48	2.2115	0x58	2.437	0x68	2.664	0x78	2.889
0x49	2.2245	0x59	2.449	0x69	2.6765	0x79	2.901
0x4A	2.239	0x5A	2.4635	0x6A	2.6915	0x7A	2.916
0x4B	2.2505	0x5B	2.476	0x6B	2.7035	0x7B	2.928
0x4C	2.2665	0x5C	2.491	0x6C	2.719	0x7C	2.9435
0x4D	2.279	0x5D	2.5035	0x6D	2.7315	0x7D	2.956
0x4E	2.293	0x5E	2.518	0x6E	2.7455	0x7E	2.97
0x4F	2.305	0x5F	2.53	0x6F	2.758	0x7F	2.9825

VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V
0x80	2.994	0x90	3.22	0xA0	3.4475	0xB0	3.6715
0x81	3.007	0x91	3.2325	0xA1	3.46	0xB1	3.6845
0x82	3.021	0x92	3.247	0xA2	3.4745	0xB2	3.699
0x83	3.033	0x93	3.259	0xA3	3.4865	0xB3	3.712
0x84	3.049	0x94	3.275	0xA4	3.5025	0xB4	3.728
0x85	3.062	0x95	3.288	0xA5	3.5155	0xB5	3.74
0x86	3.076	0x96	3.302	0xA6	3.5295	0xB6	3.754
0x87	3.0885	0x97	3.314	0xA7	3.541	0xB7	3.766
0x88	3.1065	0x98	3.332	0xA8	3.56	0xB8	3.784
0x89	3.119	0x99	3.3445	0xA9	3.5725	0xB9	3.797
0x8A	3.1335	0x9A	3.359	0xAA	3.587	0xBA	3.811
0x8B	3.146	0x9B	3.371	0xAB	3.5995	0xBB	3.823
0x8C	3.161	0x9C	3.3865	0xAC	3.6145	0xBC	3.8385
0x8D	3.1745	0x9D	3.399	0xAD	3.627	0xBD	3.851
0x8E	3.188	0x9E	3.4135	0xAE	3.6415	0xBE	3.865
0x8F	3.2005	0x9F	3.425	0xAF	3.6525	0xBF	3.8775

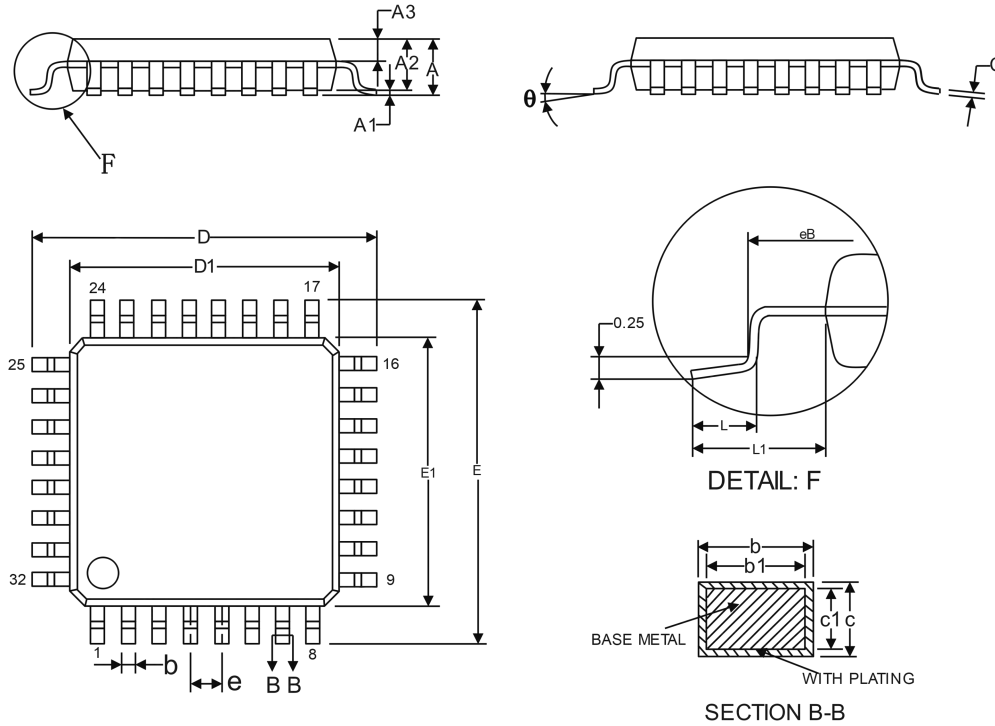
VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V	VREFCAL	电压/V
0xC0	3.8845	0xD0	4.11	0xE0	4.3375	0xF0	4.5615
0xC1	3.8975	0xD1	4.1225	0xE1	4.3495	0xF1	4.574
0xC2	3.911	0xD2	4.137	0xE2	4.3645	0xF2	4.589
0xC3	3.9235	0xD3	4.15	0xE3	4.377	0xF3	4.6015
0xC4	3.939	0xD4	4.1655	0xE4	4.3925	0xF4	4.6175
0xC5	3.9525	0xD5	4.1775	0xE5	4.405	0xF5	4.63
0xC6	3.9665	0xD6	4.192	0xE6	4.419	0xF6	4.644
0xC7	3.9785	0xD7	4.2045	0xE7	4.4315	0xF7	4.656
0xC8	3.9975	0xD8	4.2225	0xE8	4.449	0xF8	4.674
0xC9	4.01	0xD9	4.235	0xE9	4.462	0xF9	4.6865
0xCA	4.0245	0xDA	4.2495	0xEA	4.4765	0xFA	4.701
0xCB	4.0365	0xDB	4.2615	0xEB	4.4885	0xFB	4.713
0xCC	4.0525	0xDC	4.2765	0xEC	4.5035	0xFC	4.728
0xCD	4.065	0xDD	4.2895	0xED	4.5165	0xFD	4.741
0xCE	4.079	0xDE	4.303	0xEE	4.5305	0xFE	4.755
0xCF	4.091	0xDF	4.316	0xEF	4.5425	0xFF	4.767

**注：DAC 输出电压 3V~5V 误差范围±2%，以上数据为 5V 电压下均值电压值，仅作参考。**

## 22 封装尺寸

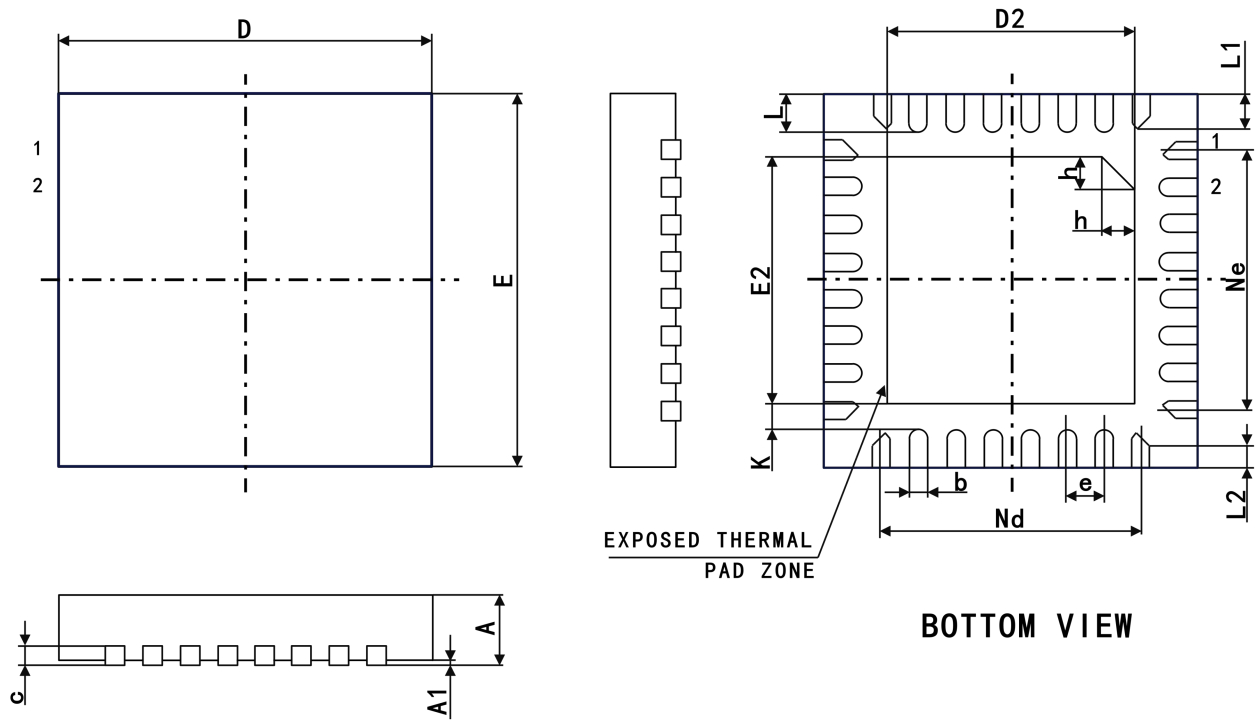
### 22.1 32PIN 封装尺寸

(单位: mm)



Symbol	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.33	—	0.41
b1	0.32	0.35	0.38
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.80BSC		
L	0.45	—	0.75
L1	1.00REF		
$\theta$	0°	—	7°

LQFP32 封装尺寸



BOTTOM VIEW

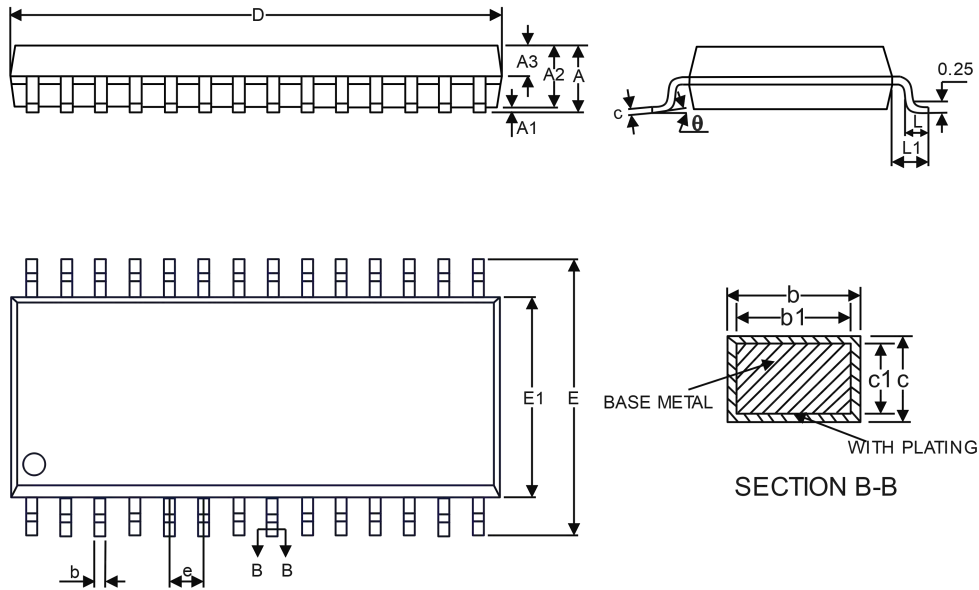
Symbol	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.60	2.65	2.70
e	0.40BSC		
Nd	2.80BSC		
E	3.90	4.00	4.10
E2	2.60	2.65	2.70
Ne	2.80BSC		
K	0.20	—	—
L	0.35	0.40	0.45
L1	0.30	0.35	0.40
L2	0.15	0.20	0.25
h	0.30	0.35	0.40

QFN32 封装尺寸



## 22.2 28PIN 封装尺寸

(单位: mm)



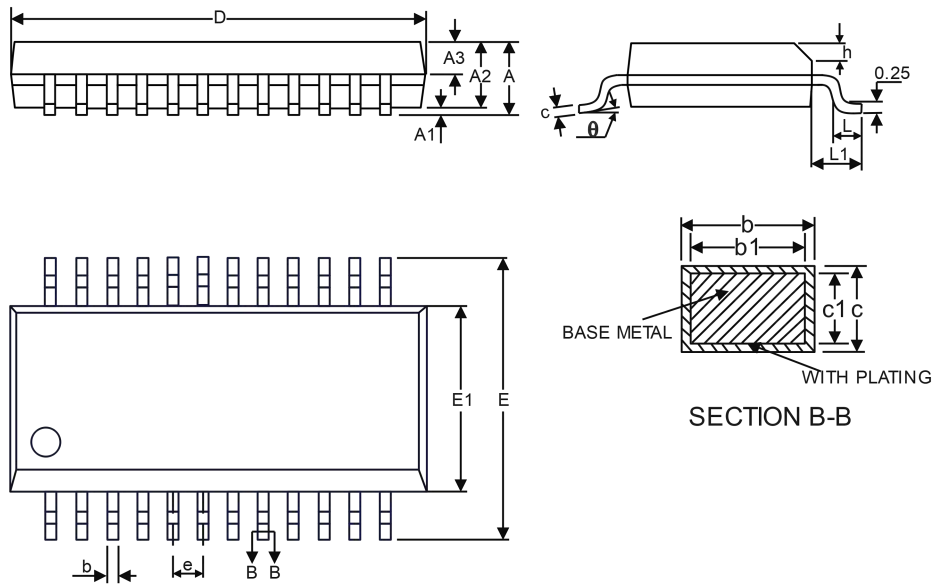
Symbol	MILLIMETER		
	MIN	NOM	MAX
A	—	—	2.00
A1	0.05	—	0.25
A2	1.65	1.75	1.85
A3	0.75	0.80	0.85
b	0.28	—	0.36
b1	0.27	0.30	0.33
c	0.15	—	0.19
c1	0.14	0.15	0.16
D	10.10	10.20	10.30
E	7.60	7.80	8.00
E1	5.20	5.30	5.40
e	0.65BSC		
L	0.75	—	1.05
L1	1.25REF		
θ	0°	—	8°

SSOP28 封装尺寸



### 22.3 24PIN 封装尺寸

(单位: mm)



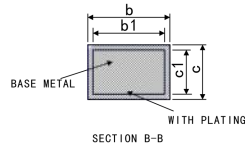
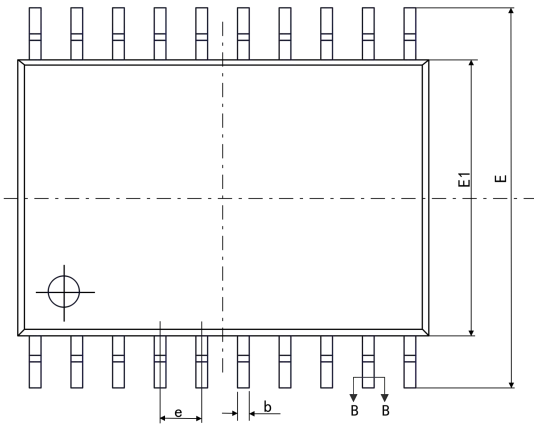
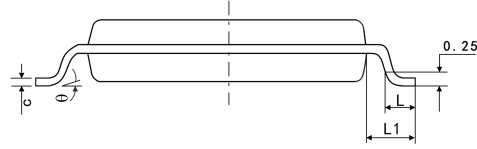
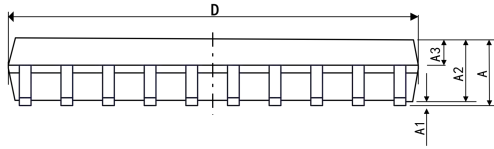
Symbol	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
$\theta$	0°	—	8°

SSOP24 封装尺寸



## 22.4 20PIN 封装尺寸

(单位: mm)



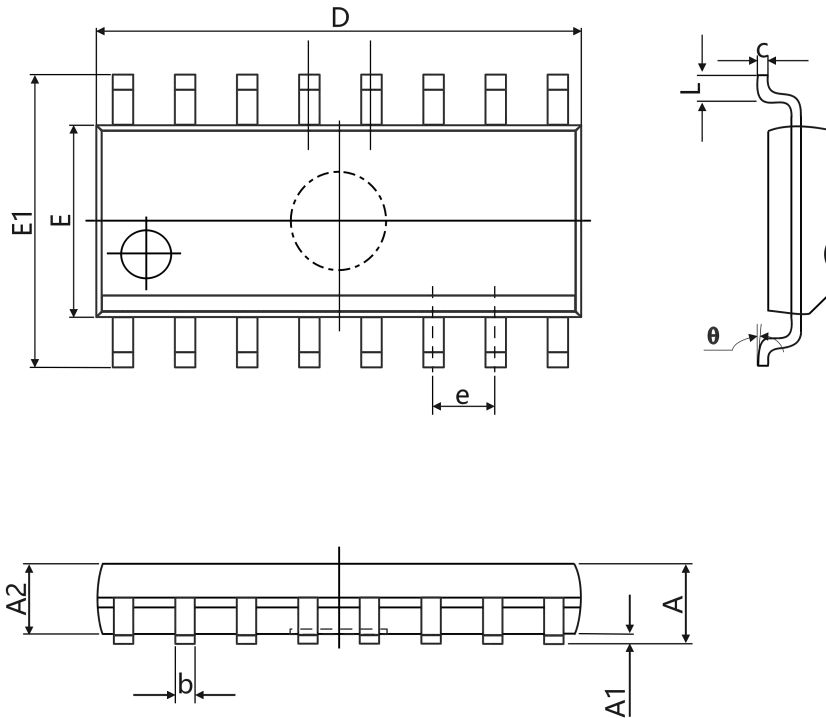
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	0.08	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.28
b1	0.19	0.22	0.25
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

TSSOP20 封装尺寸



## 22.5 16PIN 封装尺寸

(单位: mm)



Symbol	Dimensions In Millimeter		Dimensions In Inches	
	MIN	MAX	MIN	MAX
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.007	0.010
D	9.800	10.200	0.386	0.402
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

SOP16 封装尺寸