JZ8F8620

8 位 FLASH 微控制器
用户数据手册

版本号 V1.0

免责声明

无锡晶哲科技有限公司（简称：无锡晶哲科技）保留关于该规格书中产品的可靠性、功能和设计方面的改进作进一步说明的权利。由于使用本用户手册中的信息或内容而导致的直接，间接，特别附带结果的损害，无锡晶哲科技没有义务负责。本用户手册中提到的其应用仅仅是用来做说明，本公司不保证这些应用没有更深入的测试就能适用。本规格书中提到的软件（如果有），都是依据授权或保密合约所合法提供的，并且只能在这些合约的许可条件下使用或者复制。无锡晶哲科技的产品不是专门设计来应用于生命维持的用具，装置或者系统。无锡晶哲科技的产品不支持而且禁止在这些方面的应用。本用户手册内容如有变动恕不另作通知，具体更新信息，请参考公司官方网站 www.wxjzkj.com

修改记录说明

版本号	修改说明	备注
V0.0	初稿	
V0.1	修订错误描述，更新电气特性参数	
V1.0	发布初版	

注意事项：

1、EEPROM 的 0XFF/0XFE/0XFD/0XFC 四个地址用于内部存储，0XFF/0XFE 存储参考电压基准值 2V。0XFD/0XFC 存储 0.255VDD 通道值，具体用法详询 FAE。

2、睡眠前需保证 ADC 采样完成之后必须将 AD 基准设置为 VDD 即 ADCON1（R1A7）=0X07 且通道不能同时为 VDD，不然会有功耗；

目 录

1 芯片简介 8
1.1 功能特性 8
1.2 型号说明 9
1.3 引脚分配 9
1.4 引脚描述 11
1.5 系统框图 14
2 中央处理器 15
2.1 程序存储区结构 15
2.2 数据存储区 16
2.2.1 数据存储区结构 16
2.2.2 特殊功能寄存器概览 16
2.3 特殊系统寄存器 17
2.3.1 间接寻址 17
2.3.1.1 R180/RSR(RAM 选择低位寄存器) 17
2.3.1.2 R184/RSRH(RAM 选择高位寄存器) 17
2.3.1.3 R1FD/IAR0(间接寻址寄存器 0) 18
2.3.1.4 R1FE/IAR1(间接寻址寄存器) 18
2.3.1.5 R1FF/IAR2(间接寻址寄存器) 18
2.3.2 查表 18
2.3.2.1 跳转表 18
2.3.2.2 ROM 区查表 20
2.3.3 状态寄存器 21
2.3.3.1 R183/STATUS(状态标志寄存器) 21
2.3.4 电源域 22
2.3.4.1 R183/STATUS(状态标志寄存器) 22
2.3.4.2 R186/TBRDH(查表指针高位寄存器) 22
3 I/O 端口 23
3.1 GPIO 内部结构图 23
3.2 P5 端口 24
3.2.1 R18A/PORT5(P5 数据寄存器) 24
3.2.2 R18D/P5CON(P5 控制寄存器) 24
3.2.3 R190/P5PH(P5 上拉控制寄存器) 24
3.2.4 R193/P5PD(P5 下拉控制寄存器) 24
3.2.5 R19A/P5WD(P5 弱驱使能寄存器) 25
3.2.6 R19C/P5IWE(P5 输入变化中断、唤醒使能寄存器) 25
3.2.7 R1A0/P5AE(P5 模拟口使能寄存器) 25
3.3 P6 端口 26
3.3.1 R18B/PORT6(P6 数据寄存器) 26
3.3.2 R18E/P6CON(P6 控制寄存器) 26
3.3.3 R191/P6PH(P6 上拉控制寄存器) 26
3.3.4 R194/P6PD(P6 下拉控制寄存器) 27
3.3.5 R19B/P6OD(P6 开漏使能寄存器) 27

3.3.6 R19D/P6IWE(P6 输入变化中断、唤醒使能寄存器) 27
3.3.7 R1A1/P6AE(P6 模拟口使能寄存器) 27
3.4 P7 端口 28
3.4.1 R18C/PORT7(P7 数据寄存器) 28
3.4.2 R18F/P7CON(P7 控制寄存器) 28
3.4.3 R192/P7PH(P7 上拉控制寄存器) 28
3.4.4 R195/P7PD(P7 下拉控制寄存器) 29
3.4.5 R19E/P7IWE(P7 输入变化中断、唤醒使能寄存器) 29
3.4.6 R1A2/P7AE(P7 模拟口使能寄存器) 29
3.5 端口输入变化唤醒 29
3.6 端口施密特参数 30
4 中断 31
4.1 中断现场保护 32
4.2 中断相关寄存器 32
4.2.1 R1AE/EXINTCON(外部中断控制寄存器) 32
4.2.2 R1D6/INTE0(中断使能控制寄存器 0) 33
4.2.3 R1D7/INTE1(中断使能控制寄存器 1) 34
4.2.4 R1DA/INTF0(中断标志寄存器 0) 34
4.2.5 R1DB/INTF1(中断标志寄存器 1) 35
5 复位 37
5.1 复位功能概述 37
5.2 POR 上电复位 37
5.3 WDT 看门狗复位 38
5.4 LVR 低电压复位 38
5.5 工作频率与 LVR 低压检测关系 39
5.6 复位相关寄存器 40
5.6.1 R1AF/WDTCON(WDT、唤醒控制寄存器) 40
6 系统时钟与工作模式 41
6.1 系统时钟 41
6.1.1 内部 IHRC 振荡器 41
6.1.2 内部 ILRC 振荡器 41
6.1.3 模块应用说明 41
6.2 工作模式 42
6.2.1 高速模式 42
6.2.2 低速模式 43
6.2.3 空闲模式 43
6.2.4 睡眠模式 44
6.3 相关寄存器 44
6.3.1 R185/WKCON(唤醒控制寄存器) 44
6.3.2 R188/CPUCON(CPU 模式控制寄存器) 44
6.3.3 R189/IHRCCAL(IHRC 频率微调寄存器) 45
6.3.4 R1AF/WDTCON(WDT、唤醒控制寄存器) 46
7 定时计数器 47
7.1 TC0 定时计数器 47

7.1.1 TC0 定时设置说明 47
7.1.2 TC0 定时计算说明 48
7.1.3 TC0 空闲模式唤醒说明 48
7.1.4 TC0 端口电平捕获说明 48
7.1.5 TC0 相关寄存器 49
7.1.5.1 R1AD/TPRE(TC0 预分频器/TC0 捕获低 8 位寄存器) 49
7.1.5.2 R1AE/EXINTCON(外部中断捕获控制寄存器) 49
7.1.5.3 R1C0/TC0CON(TC0 控制寄存器) 50
7.1.5.4 R1C1/TC0C(TC08 位计数寄存器) 51
7.2 TC1 定时计数器 52
7.2.1 TC1 定时设置说明 52
7.2.2 TC1 定时计算说明 52
7.2.3 TC1 空闲模式唤醒说明 53
7.2.4 TC1 相关寄存器 53
7.2.4.1 R1B0/TC1CON(TC1 控制寄存器) 53
7.2.4.2 R1B1/TC1PRDL(TC1/PWM123 周期低 8 位寄存器) 54
7.2.4.3 R1B5/TC1PRDTH(TC1 定时计数器高 4 位寄存器) 54
7.3 TC2 定时计数器 55
7.3.1 TC2 定时设置说明 55
7.3.2 TC2 定时计算说明 55
7.3.3 TC2 空闲模式唤醒说明 56
7.3.4 TC2 相关寄存器 56
7.3.4.1 R1B8/TC2CON(TC2 控制寄存器) 56
7.3.4.2 R1B9/TC2PRDL(TC2 定时计数器低 8 位寄存器) 57
7.3.4.3 R1BD/TC2PRDTH(TC2 定时计数器高 4 位寄存器) 57
7.4 TC3 定时计数器 58
7.4.1 TC3 定时设置说明 58
7.4.2 TC3 定时计算说明 59
7.4.3 TC3 空闲模式唤醒说明 59
7.4.4 TC3 相关寄存器 59
7.4.4.1 R1C8/TC3CON(TC3 控制寄存器) 59
7.4.4.2 R1C9/TC3CL(TC3 低 8 位计数寄存器) 60
7.4.4.3 R1CA/TC3CH(TC3 高 8 位计数寄存器) 60
8 PWM 脉宽调制 61
8.1 PWM 内部结构与时序 61
8.2 PWM 周期与占空比 62
8.3 PWM 空闲模式唤醒说明 63
8.4 PWM 比较器门控说明 63
8.5 IPWM 波形以及 Buzzer 设置说明 63
8.6 PWM7DT 中断以及触发 ADC 说明 64
8.7 PWM 刹车功能 64
8.8 PWMINV 取反功能 64
8.9 LED 单线级联 64
8.10 PWM 相关寄存器 65

8.10.1 R186/TBRDH(查表指针高位寄存器) 65
8.10.2 R1B0/TC1CON(TC1 控制寄存器) 66
8.10.3 R1B1/TC1PRDL(TC1/PWM123 周期低 8 位寄存器) 67
8.10.4 R1B2/PWM1DTL(PWM1 占空比低 8 位寄存器) 67
8.10.5 R1B3/PWM2DTL(PWM2 占空比低 8 位寄存器) 67
8.10.6 R1B4/PWM3DTL(PWM3 占空比低 8 位寄存器) 67
8.10.7 R1B5/TC1PRDTH(TC1/PWM123 周期高 4 位及 PWM3 占空比高 4 位寄存器) 67
8.10.8 R1B6/PWM21DTH(PWM21 占空比高 4 位寄存器) 68
8.10.9 R1B7/PWMCON1(PWM 控制寄存器 1) 68
8.10.10 R1B8/TC2CON(TC2 控制寄存器) 69
8.10.11 R1B9/TC2PRDL(TC2/PWM4567 周期低 8 位寄存器) 70
8.10.12 R1BA/PWM4DTL(PWM4 占空比低 8 位寄存器) 70
8.10.13 R1BB/PWM5DTL(PWM5 占空比低 8 位寄存器) 70
8.10.14 R1BC/PWM6DTL(PWM6 占空比低 8 位寄存器) 70
8.10.15 R1BD/TC2PRDTH(TC2/PWM4567 周期高 4 位及 PWM6 占空比高 4 位寄存器) 70
8.10.16 R1BE/PWM54DTH(PWM54 占空比高 4 位寄存器) 71
8.10.17 R1BF/PWMCON2(PWM 控制寄存器 2) 71
8.10.18 R1C2/PWM7DTL(PWM7 占空比低 8 位寄存器) 72
8.10.19 R1C3/PWM7CON(PWM7 控制及占空比高位寄存器) 72
8.10.20 R1C4/PWMINVCON(PWM 取反控制寄存器) 73
9 ADC 模数转换 74
9.1 ADC 检测电源电压说明 75
9.2 P51 端口检测 0.255 外部电压说明 75
9.3 ADC 模数转换设置说明 76
9.4 ADC 相关寄存器 76
9.4.1 R185/WKCON(唤醒控制寄存器) 76
9.4.2 R1A3/ADATH(ADC 数据高 8 位寄存器) 76
9.4.3 R1A4/ADATL(ADC 数据低 8 位寄存器) 76
9.4.4 R1A5/ADATH1(ADC 数据高 4 位寄存器) 77
9.4.5 R1A6/ADCON0(ADC 控制寄存器 0) 77
9.4.6 R1A7/ADCON1(ADC 控制寄存器 1) 78
10 CMP 比较器 80
10.1 分压电阻输出电压 Vinternal R 81
10.2 比较器配置 82
10.3 RFC 电阻频率转换 82
10.4 CMP 相关寄存器 83
10.4.1 R1C3/PWM7CON(RFC 门控控制寄存器) 83
10.4.2 R1F8/CMPCON(CMP 总控制寄存器) 83
10.4.3 R1F9/CMPCON0(CMP 控制寄存器 0) 84
10.4.4 R1FA/CMPCON1(CMP 控制寄存器 1) 85
11 LED 点阵驱动 87
11.1 LED 功能描述 87
11.2 LED 点阵设置说明 89
11.3 LED 点阵相关寄存器 89

11.3.1 R1F0/LEDCON0(LED 控制寄存器 0) 89
11.3.2 R1F1/LEDCON1(LED 控制寄存器 1) 90
11.3.3 R1F2/LEDCOM(LED COM 口控制寄存器) 91
11.3.4 R1F3/LEDDT0(LED 占空比寄存器 0) 91
11.3.5 R1F4/LEDDT1(LED 占空比寄存器 1) 91
11.3.6 R1F5/LEDDTS(LED DUTY 选择寄存器) 92
11.3.7 R1F6/LEDDB(LED 数据寄存器) 92
12 UART/IIC/SPI 三选一串口 93
12.1 UART 串口接收器/发送器 93
12.1.1 UART 发送 93
12.1.2 UART 接收 94
12.2 IIC 串口接收器/发送器 95
12.2.1 IIC 操作流程 95
12.2.2 IIC 发送与接收 96
12.3 SPI 串口接收器/发送器 97
12.3.1 SPI 工作模式 97
12.3.2 SPI 发送与接收 98
12.4 SSI 相关寄存器 99
12.4.1 R1E3/SSICON0(SSI 控制寄存器 0) 99
12.4.2 R1E4/SSICON1(SSI 控制寄存器 1) 100
12.4.3 R1E5/I2CADDR(I2C 地址寄存器) 101
12.4.4 R1E6/SSIBUF(SSI 收发数据寄存器) 101
12.4.5 R1E7/SSISR0(串口状态寄存器 0) 101
12.4.6 R1E8/SSISR1(串口状态寄存器 1) 102
13 EEPROM 以及 ISP 在线烧录操作 104
13.1 EEPROM 读写操作 104
13.2 ISP 在线烧录时序 104
13.3 相关寄存器 105
13.3.1 R187/TBRDL(查表指针低位寄存器) 105
13.3.2 R1E0/E2PCON(EEPROM 控制寄存器) 105
13.3.3 R1E1/E2PDATL(EEPROM 数据低 8 位寄存器) 106
13.3.4 R1E2/E2PDATH(EEPROM 数据高 8 位寄存器) 106
14 OPTION 配置表 107
15 指令集 109
16 电气特性 111
16.1 极限参数 111
16.2 直流电气特性 111
16.3 AD 转换特性 113
16.4 比较器特性 113
16.5 特性曲线图 114
16.5.1 内部低速振荡器 - 压频特性曲线 114
16.5.2 内部低速振荡器 - 温频特性曲线 114
16.5.3 内部 16Mhz RC 振荡器 - 压频特性曲线 115
16.5.4 内部 16Mhz RC 振荡器 - 温频特性曲线 115

16.6 IHRC 频率微调参数说明 116
17 封装尺寸 118
17.1 24PIN 封装尺寸 118
17.2 20PIN 封装尺寸 119
17.3 16PIN 封装尺寸 120
17.4 8PIN 封装尺寸 121

1 芯片简介

1.1 功能特性

CPU 配置

- 8K×16－Bit MTP ROM
- 256×16－Bit EEPROM
- 336×8－Bit SRAM
- 8 级堆栈空间
- 8 级可编程电压复位（LVR）
  1.7V，1.9V，2.1V，2.4V
  2.8V，3.0V，3.5V，3.9V
- 工作电流小于 2 mA（8MHz/5V）
- 工作电流小于 10 μA（40KHz/5V）
- 睡眠电流小于 3 μA（睡眠模式）

工作电压

- 工作电压范围：

V_{LVR} 3.0V～5.5V|Fcpu=0～8MHz
V_{LVR} 2.4V～5.5V|Fcpu=0～4MHz
V_{LVR} 2.1V～5.5V|Fcpu=0～2MHz

I/O 配置

- 3 组 22 个双向 IO 端口：P5/P6/P7，所有端口支持推挽、上拉、下拉、输入高阻
- P6 口支持开漏输出
- P50～52 口支持弱驱动
- P50～52，P60～67，P70 支持高灌电流
- 所有端口均支持睡眠唤醒和空闲唤醒
- 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±3%
- 内部 ILRC 振荡电路：40KHz
- 时钟周期分频：2T，4T，8T，16T，32T

外围模块

- 12Bit ADC 模数转换器
  - 18 路外部通道：AIN0～AIN17；
  - 4 路内部通道：0.255VDD，0.255P51，VREF，GND
  - 参考电压：VDD、VREF(VBG/2V/3V/4V)、外部输入 EXVREF(P61)
- 定时器
  - 8 Bit 定时计数器 TC0
  - 12 Bit 定时计数器 TC1/TC2
  - 8bit/16Bit 定时计数器 TC3
- PWM
  - 3 路 1 组 12Bit－PWM123(一对互补)
  - 4 路 1 组 12Bit－PWM4567(一对互补)
  - 任意 PWM 可取反
  - PWM7 占空比可触发 AD 采样
  - 2 路 RGB LED 级联控制器(PWM1/PWM4)
- 1 路 SSI(支持 UART/IIC/SPI 三种模式)
- 1 路比较器 CMP
  - 可测试 VDD 电压
  - 与 PWM4～PWM7 联动输出电平(刹车)
  - CMP 结果变化可触发 AD 采样
- LED 点阵
  - 驱动最大支持 9*8，128 级亮度可调
  - LED 口可任意组合
  - 支持手动和自动两种模式

中断源

- 4 个定时器中断
- P6/P7/P8 输入变化中断
- 其他外设中断

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

- JZ8F8620－SSOP24
- JZ8F8620－TSSOP20
- JZ8F8620－SOP16
- JZ8F8620－SOP8

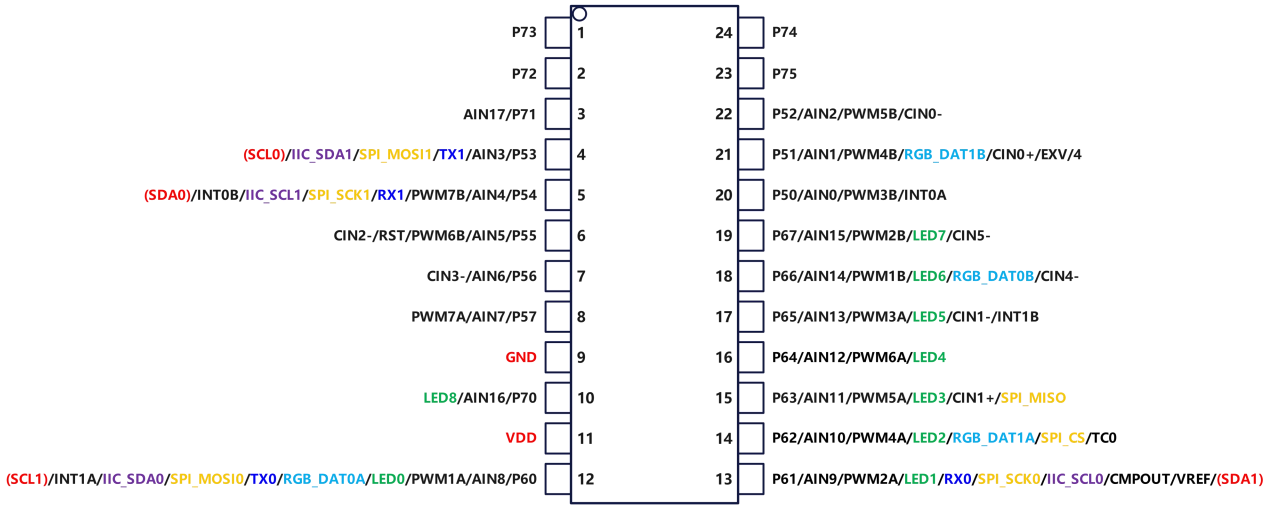
1.2 型号说明

PRODUCT	ROM	RAM	EEPROM	I/O	PWM	ADC	点阵	IIC	UART	SPI	PACKAGE
JZ8F8620P24	8K*16	336*8	256*16	22	7*12	18	8*9	1	1	1	SSOP24
JZ8F8620T20	8K*16	336*8	256*16	18	7*12	18	8*9	1	1	1	TSSOP20
JZ8F8620S16	8K*16	336*8	256*16	14	7*12	14	7*8	1	1	1	SOP16
JZ8F8620S08	8K*16	336*8	256*16	6	6*12	6	-	1	1	-	SOP8

1.3 引脚分配

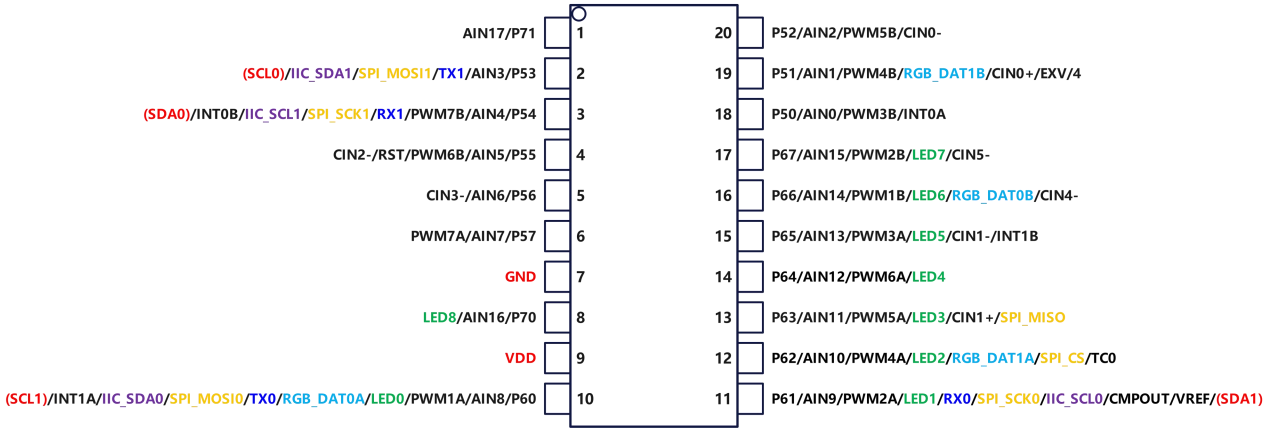
JZ8F8620 烧录下载与仿真口：P53(SCL0)、P54(SDA0)或 P60(SCL1)、P61(SDA1)

型号：JZ8F8620P24



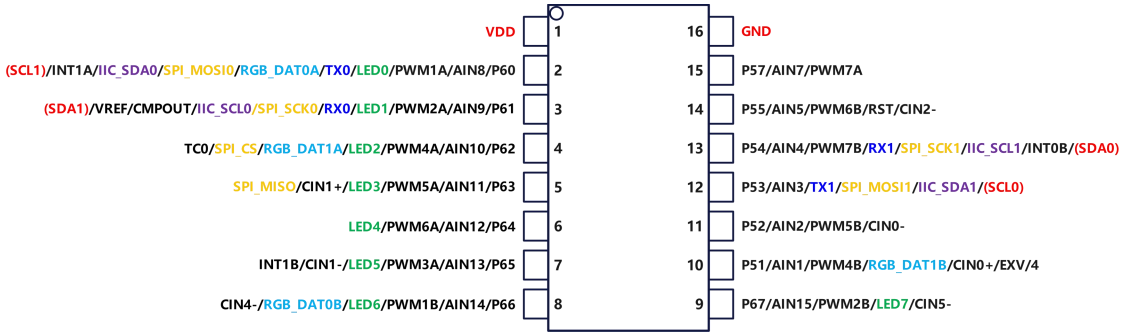
SSOP24 封装引脚图

型号：JZ8F8620T20



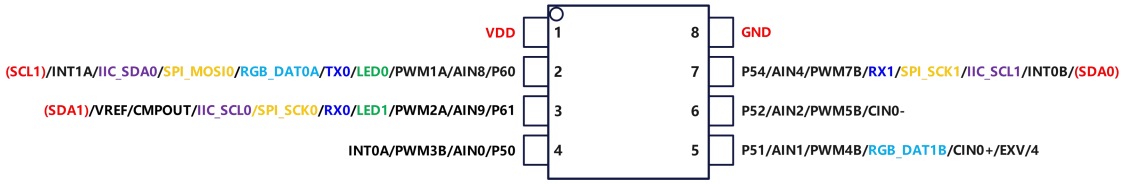
TSSOP20 封装引脚图

型号：JZ8F8620S16



SOP16 封装引脚图

型号：JZ8F8620S08



SOP8 封装引脚图

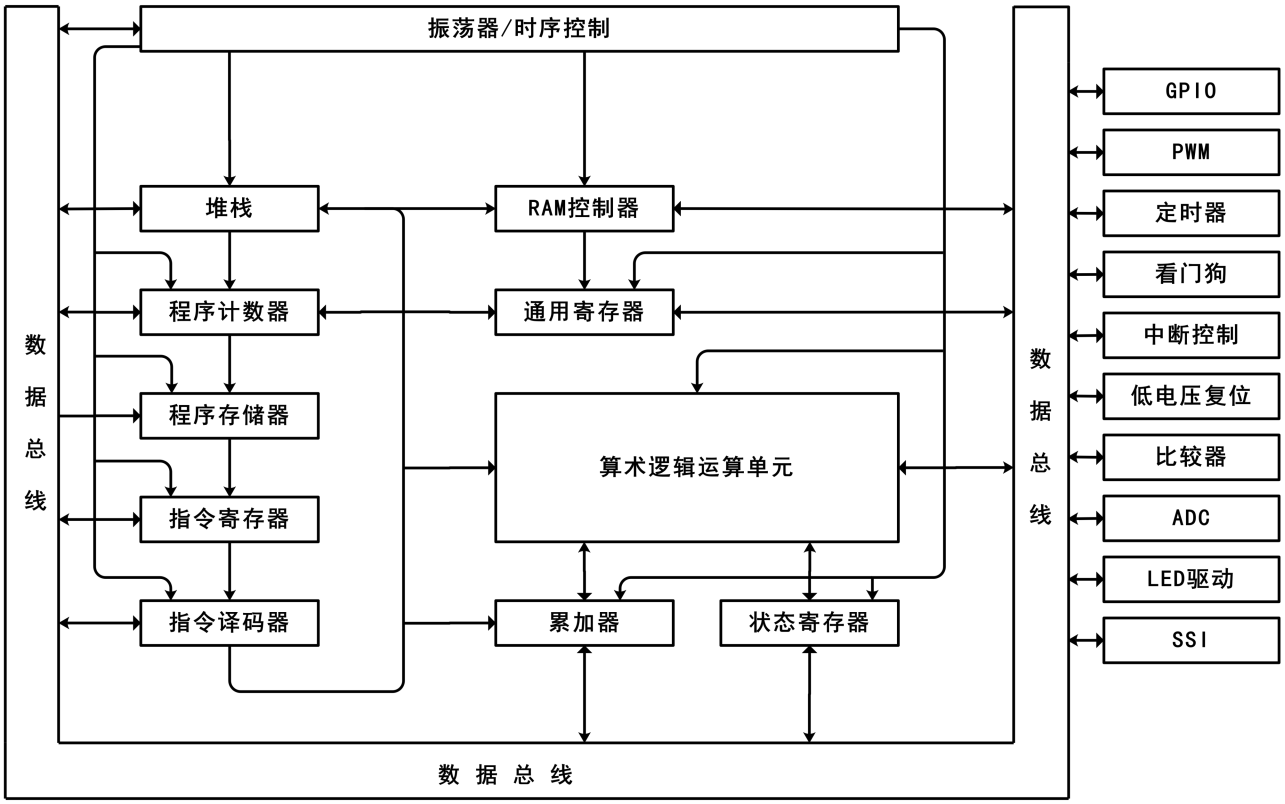
1.4 引脚描述

序号	管脚名	I/O	功能描述
P50	P50	I/O	GPIO，可编程上下拉、弱拉灌电流、二级高灌电流、四级拉电流，端口唤醒
	AIN0	AN	模拟输入通道
	INT0A	I	外部中断 INT0 输入口
	PWM3B	O	PWM3B 输出
P51	P51	I/O	GPIO，可编程上下拉、弱拉灌电流、二级高灌电流、四级拉电流，端口唤醒
	AIN1	AN	模拟输入通道
	CIN0+	I	比较器正极输入口
	PWM4B	O	PWM4B 输出
	RGB_DAT1B	O	幻彩驱动输出脚
	EXV/4	AN	外部 1/4 ADC 采样
P52	P52	I/O	GPIO，可编程上下拉、弱拉灌电流、二级高灌电流、四级拉电流，端口唤醒
	AIN2	AN	模拟输入通道
	CIN0-	I	比较器负极输入口
	PWM5B	O	PWM5B 输出
P53	P53	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
	AIN3	AN	模拟输入通道
	TX1	O	UART 通信发送口 1
	SPI_MOSI1	I/O	SPI 主机输出，从机输入数据口 1
	IIC_SDA1	O	IIC 数据口 1
	(SCL0)	I	烧录时序口
P54	P54	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
	AIN4	AN	模拟输入通道
	INT0B	I	外部中断 INT0 输入口
	PWM7B	O	PWM7B 输出
	RX1	I	UART 通信接收口 1
	SPI_SCK1	O	SPI 主机模式时钟口 1
	IIC_SCL1	O	IIC 时钟口 1
	(SDA0)	I	烧录时序口
P55	P55	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
	AIN5	AN	模拟输入通道
	CIN2-	I	比较器负极输入口
	PWM6B	O	PWM6B 输出
	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，拉电流增强

	AIN6	AN	模拟输入通道
	CIN3-	I	比较器负极输入口
P57	P57	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
	AIN7	AN	模拟输入通道
	PWM7A	O	PWM7A 输出
P60	P60	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN8	AN	模拟输入通道
	PWM1A	O	PWM1A 输出
	RGB_DAT0A	O	幻彩驱动输出脚
	INT1A	I	外部中断 1 输入口
	SPI_MOSI0	O	SPI 主机模式数据口 0
	IIC_SDA0	O	IIC 数据口 0
	LED0	O	LED 驱动口 0
	TX0	O	UART 发送口 0
	(SCL1)	I	烧录时序口
P61	P61	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN9	AN	模拟输入通道
	PWM2A	O	PWM2A 输出
	LED1	O	LED 驱动口 1
	CMPOUT	O	比较器结果输出口
	VREF	AN	ADC 外部基准电压输入口
	RX0	I	UART 通信接收口 0
	SPI_SCK0	O	SPI 主机模式时钟口 0
	IIC_SCL0	O	IIC 时钟口 0
	(SDA1)	I	烧录时序口
P62	P62	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN10	AN	模拟输入通道
	PWM4A	O	PWM4A 输出
	RGB_DAT1A	O	幻彩驱动输出脚
	LED2	O	LED 驱动口 2
	SPI_CS	I	SPI 从机片选控制口
	TC0	I	TC0 外部时钟输入口
P63	P63	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN11	AN	模拟输入通道
	PWM5A	O	PWM5A 输出
	LED3	O	LED 驱动口 3
	CIN1+	AN	比较器正极输入口

	SPI_MISO	I/O	SPI 主机输入，从机发送数据口
P64	P64	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN12	AN	模拟输入通道
	PWM6A	O	PWM6A 输出
	LED4	O	LED 驱动口 4
P65	P65	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN13	AN	模拟输入通道
	PWM3A	O	PWM3A 输出
	INT1B	I	外部中断 1 输入口
	CIN1-	AN	比较器负极输入口
	LED5	O	LED 驱动口 5
P66	P66	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN14	AN	模拟输入通道
	PWM1B	O	PWM1B 输出
	RGB_DAT0B	O	幻彩驱动输出脚
	CIN4-	AN	比较器负极输入口
	LED6	O	LED 驱动口 6
P67	P67	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN15	AN	模拟输入通道
	PWM2B	O	PWM2B 输出
	CIN5-	AN	比较器负极输入口
	LED7	O	LED 驱动口 7
P70	P70	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN16	AN	模拟输入通道
	LED8	O	LED 驱动口 8
P71	P71	I/O	GPIO，可编程上下拉、二级高灌电流、四级拉电流，端口唤醒
	AIN17	AN	模拟输入通道
P72	P72	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
P73	P73	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
P74	P74	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
P75	P75	I/O	GPIO，可编程上下拉，端口唤醒，拉电流增强
	VDD	--	电源
	GND	--	地

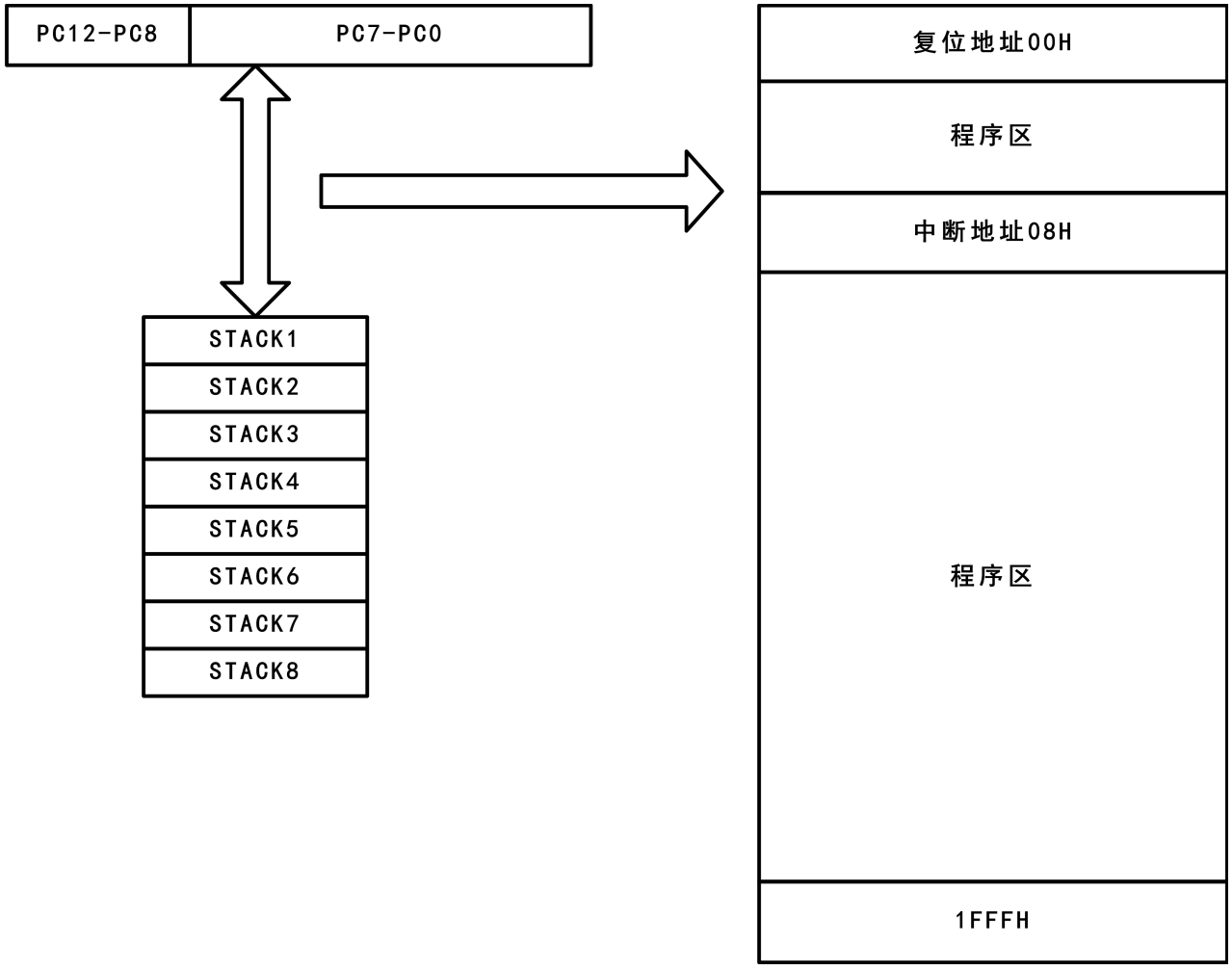
1.5 系统框图



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		E2PCON	LEDCON0
1	PCH	P6PH	P6AE	TC1PRDL	TC0C		E2PDATL	LEDCON1
2	PCL	P7PH	P7AE	PWM1DTL	PWM7DTL		E2PDATH	LEDCOM
3	STATUS	P5PD	ADATH	PWM2DTL	PWM7CON		SSICON0	LEDDT0
4	RSRH	P6PD	ADATL	PWM3DTL	PWMINVCON		SSICON1	LEDDT1
5	WKCON	P7PD	ADATH1	TC1PRDTH			I2CADDR	LEDDTS
6	TBRDH		ADCON0	PWM21DTH		INTE0	SSIBUF	LEDDB
7	TBRDL		ADCON1	PWMCON1		INTE1	SSISR0	
8	CPUCON			TC2CON	TC3CON		SSISR1	CMPCON
9	IHRCCAL			TC2PRDL	TC3CL			CMPCON0
A(10)	PORT5	P5WD		PWM5DTL	TC3CH	INTF0		CMPCON1
B(11)	PORT6	P6OD		PWM6DTL		INTF1		
C(12)	PORT7	P5IWE		PWM7DTL				
D(13)	P5CON	P6IWE	TPRE	TC2PRDTH				IAR0
E(14)	P6CON	P7IWE	EXINTCON	PWM65DTH				IAR1
F(15)	P7CON		WDTCON	PWMCON2				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<8>、RSR<7:0>寄存器配合 R1FD 寄存器实现间接寻址操作（寻址范围：0x000－0x14F）。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<15:8>寄存器配合 R1FE 寄存器实现间接寻址操作（寻址范围：0x100－0x14F）。用户可以将在寻址范围内的 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 寄存器实现间接寻址操作（寻址范围：0x00－0xFF）。用户可以将在寻址范围内的 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	RSRH<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 RSR 寄存器在间接寻址方式中用于选择 SRAM 地址（寻址范围：0x000－0x14F）。

2.3.1.3 R1FD/IAR0(间接寻址寄存器 0)

0X1FD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAR0	IAR0<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（RAM 选择寄存器）RSR<7:0>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
RET L   @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。JZ8F8620 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。JZ8F8620 拥有 8 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 8 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 8 次之后，第 9 次压栈时进栈的数据将覆盖第 1 次进栈的数据，而第 10 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_tempH,A         //地址高 8 位保存在 TempH
    MOV     A,@Table            //地址低 8 位
    ADD     A,@1                //地址偏移
    JBTC    C                   //判断进位，是否溢出
    INC     reg_tempH           //进位
    MOV     reg_templ,A         //地址低 8 位保存在 TempL
    MOV     a,reg_tempH         //地址高位
    MOV     TBRDH,a
    MOV     a,reg_templ         //地址低位
    MOV     TBRDL,a
    TBRD    reg_tempH           //保存高 8 位数据到 TempH
    MOV     reg_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	PWM4_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 5 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。

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<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 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<5>：VDD_POWER－VDD 电源域抗干扰增强选择

0：禁止

1：使能

2.3.4.2 R186/TBRDH(查表指针高位寄存器)

0X186	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDH	PWM4_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 电源域抗干扰增强选择

0：禁止

1：使能

3 I/O 端口

JZ8F8620 有 3 组双向 I/O 端口，共 22 个输入，22 个输出，大部分 I/O 可以复用为其它功能。

22 个可编程上拉 I/O 引脚：P50～P57，P60～P67，P70～P75；

22 个可编程下拉 I/O 引脚：P50～P57，P60～P67，P70～P75；

13 个可 OPTION 控制，软件使能驱动增强 I/O 引脚：P50～P52，P60～P67，P70～P71；

9 个可 OPTION 选择拉电流驱动增强 I/O 引脚：P53～P57，P72～P75；

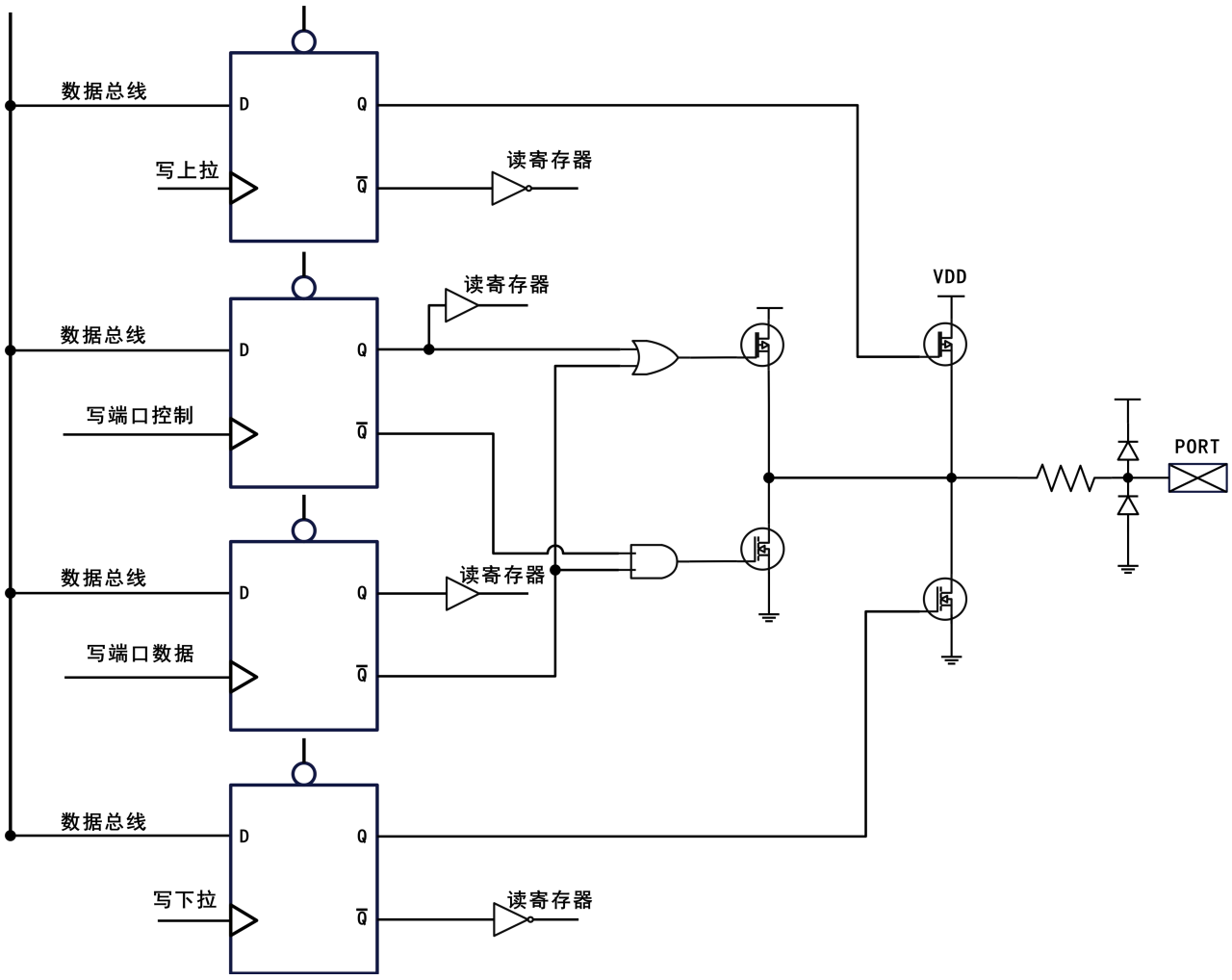
3 个可软件控制弱驱动 I/O 引脚：P50～P52；

22 个可编程唤醒 I/O 引脚：P50～P57，P60～P67，P70～P75；

18 个可编程模拟口：P50～P57，P60～P67，P70～P71；

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。



IO 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、P5WD、P5IW、P5AE 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 R19A/P5WD(P5 弱驱使能寄存器)

0X19A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5WD	-	-	-	-	-	P5WD<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

P5 弱驱控制位：

1：使能

0：禁止

注意：P5WD 的 Bit<7:3>禁止写 1。

3.2.6 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

3.2.7 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、P6OD、P6IWE、P6AE 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 R19B/P6OD(P6 开漏使能寄存器)

0X19B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P6OD	P6OD<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 开漏控制位：

0：禁止

1：使能

3.3.6 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

3.3.7 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 6Bit 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、P7AE 等。

3.4.1 R18C/PORT7(P7 数据寄存器)

0X18C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT7	-	-	PORT7<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

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<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

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<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

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<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

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<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

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<1:0>	
读/写	R	R	R	R	R	R	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 端口输入变化唤醒

JZ8F8620 包含 22 个可编程端口输入变化唤醒 I/O：P50～P57，P60～P67，P70～P75。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；

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；

4、使能端口输入变化中断 P5ICIE/P6ICIE/P7ICIE；

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；

3.6 端口施密特参数

JZ8F8620 端口的施密特特性，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～P75	0.25*VDD	0.53*VDD

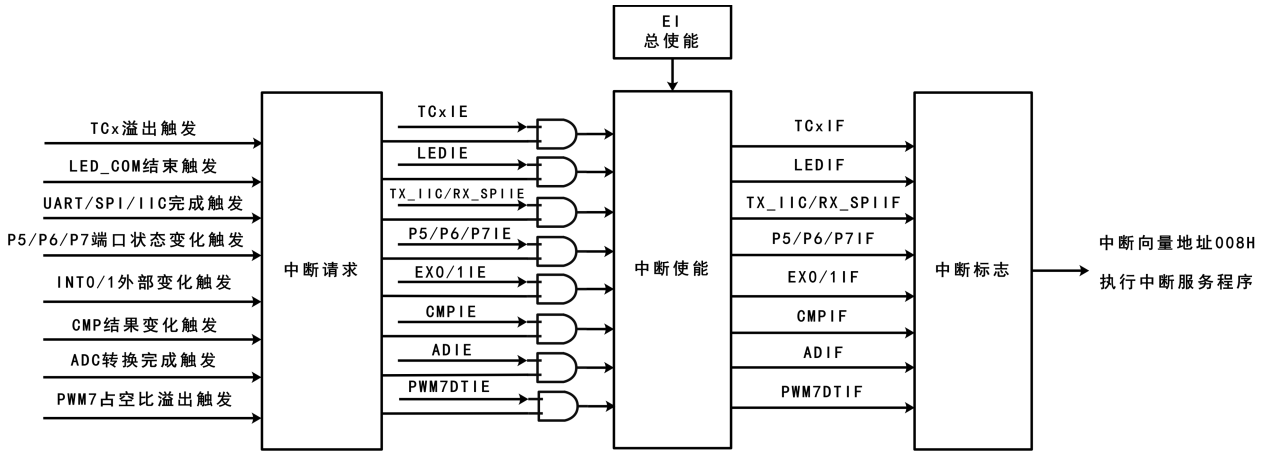
以上参数仅做参考，请以目标样机实测数据为准。

4 中断

JZ8F8620 具有 15 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
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
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
INT0 外部中断	EI + EX0IE=1	EX0IF
INT1 外部中断	EI + EX1IE=1	EX1IF
CMP 比较完成中断	EI + CMPIE=1	CMPIF
ADC 转换完成中断	EI + ADIE=1	ADIF
PWM7 占空比溢出中断	EI + PWM7DTIE=1	PWM7DTIF
LED COM 结束中断	EI + LEDIE=1	LEDIF

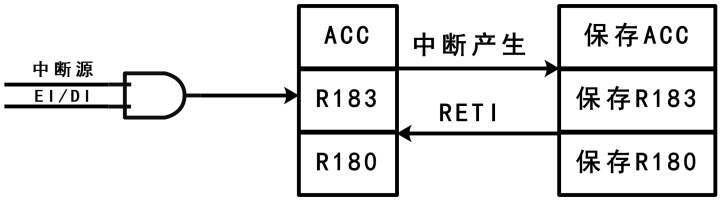
R1DA/R1DB 为中断标志寄存器，它们记录了当某个中断产生中断请求后的中断标志位。R1D6/R1D7 为中断使能控制寄存器，中断的允许与禁止在这两个寄存器中设置。总中断的允许是通过下“EI”指令，相反，总中断的禁止是通过下“DI”指令。当一个中断产生时，它的下一条指令的执行将从中断向量地址 008H 处执行。在离开中断服务程序之前相应的中断标志位必须清零，这样才能避免中断的误动作。



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>		EXINT0MS<1:0>		EXINT1S	EXINT0S	CAPS	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>：EXINT0MS<1:0>－EXINT0 模式选择位

EXINT0MS<1>	EXINT0MS<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>：EXINT0S－EXINT0 端口选择位

0：EXINT0 端口选择 P50

1：EXINT0 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	PWM7DTIE	ADIE	CMPIE	EX1IE	EX0IE	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>：PWM7DTIE－PWM7 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>：CMPIE－CMP 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	-	RX_SPIIE	TX_IICIE	LEDIE	TC3IE	TC2IE	TC1IE	TC0IE
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<7>：未使用

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>：LEDIE－LED COM 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 R1DA/INTF0(中断标志寄存器 0)

0X1DA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	PWM7DTIF	ADIF	CMPIF	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>：PWM7DTIF－PWM7 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－CMP 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

4.2.5 R1DB/INTF1(中断标志寄存器 1)

0X1DB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	-	RX_SPIIF	TX_IICIF	LEDIF	TC3IF	TC2IF	TC1IF	TC0IF
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<7>：未使用

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>：LEDIF－LED COM 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

5 复位

5.1 复位功能概述

JZ8F8620 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。

5.2 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)

5.3 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：

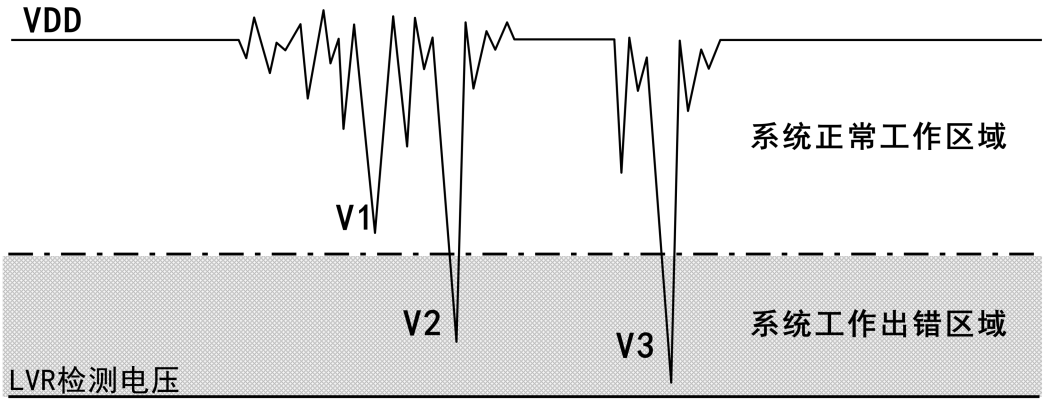
看门狗复位在空闲模式和睡眠模式下都可以复位，系统复位后从地址 0X00 开始执行程序，用户可以判断 R183（STATUS）寄存器 Bit4 的时间溢出位，如果为 WDT 溢出则可以判断为是看门狗复位，执行对应的唤醒子程序。注意，看门狗唤醒实质为看门狗复位的特殊应用。

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。

5.4 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：

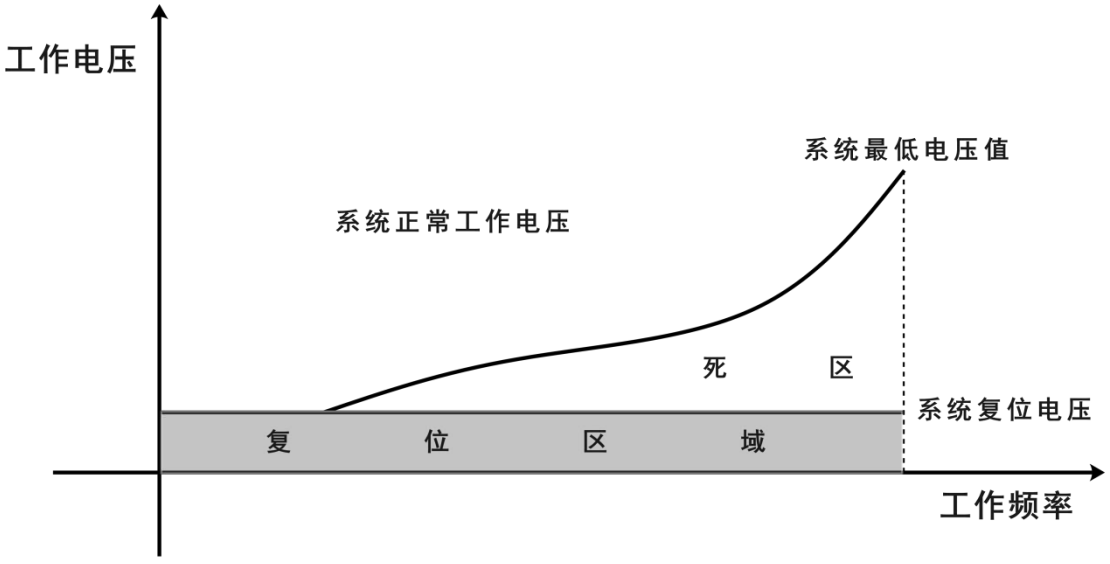
DC 运用中一般都采用电池供电，当电池电压过低或单片机驱动负载时，系统电压可能跌落并进入死区。这时，电源不会进一步下降到 LVR 检测电压，因此系统维持在死区。

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。

5.5 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：

Fosc 频率	Clocks 分频	LVR 复位电压点
Fosc - 16MHz	2 Clocks	LVR=3.0V
Fosc - 8MHz	2 Clocks	LVR=2.4V
Fosc - 4MHz	2 Clocks	LVR=2.1V

注：1、工作频率=指令周期频率=Fosc 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。

5.6 复位相关寄存器

5.6.1 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	-	CMPWE	EXINT1WE	EXINT0WE
读/写	R/W	R/W	R/W	R/W	R	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（不建议使用）

6 系统时钟与工作模式

6.1 系统时钟

JZ8F8620 内部集成了 2 种振荡器，高速 RC 振荡器 IHRC、低速 RC 振荡器 ILRC，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

JZ8F8620 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

JZ8F8620 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

JZ8F8620 内置 ILRC 低速振荡器，提供稳定的 40KHz 低速时钟。

6.1.3 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；
- 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

JZ8F8620 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 选择系统时钟可继续工作并可唤醒系统），还可通过 P5IWE，P6IWE，P7IWE，INT0WE，INT1WE，ADCWE，CMPWE，UARTWE 唤醒；
- 睡眠模式：所有功能暂停工作，系统进入睡眠，可通过TC0（P62输入时钟），P5IWE，P6IWE，P7IWE，INT0WE，INT1WE，CMPWE，UARTWE，UARTWE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）
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	停止
中断	全部有效	全部有效	全部有效 (TC0 选系统时钟)	TC0IE, PxICIE, INTxIE, CMPIE
唤醒功能	-	-	(TC0 选系统时钟可唤醒) PxIWE, EXxWE, ADCWE, CMPWE, UARTWE	TC0WE, PxIWE, INTxWE, CMPWE, UARTWE
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，P5IWE，P6IWE，P7IWE，INT0WE，INT1WE，ADCWE，CMPWE，UARTWE 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 PxIWE，INTxWE，ADCWE，CMPWE，UARTWE；
- 空闲模式下 TC0、TC1、TC2、TC3 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 3uA。睡眠模式可以由 TC0（P62 输入时钟），P5IWE，P6IWE，P7IWE，INT0WE，INT1WE，CMPWE，UARTWE 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 3uA；
- 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），P5IWE，P6IWE，P7IWE，INT0WE，INT1WE，CMPWE，UARTWE。

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	-	UARTWE
读/写	R	R	R	R	R	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0

Bit<0>：UARTWE－SSI 三合一串口唤醒使能位

0：唤醒禁止

1：唤醒使能

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	TC0WE	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

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>：TC0WE－TC0 唤醒使能位

0：唤醒禁止

1：唤醒使能

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、TC3 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 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	-	CMPWE	EXINT1WE	EXINT0WE
读/写	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	1	0	0	0	0	0	0	0

Bit<2>：CMPWE－CMP 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>：EXINT0WE－EXINT0 唤醒使能位

0：禁止

1：使能

7 定时计数器

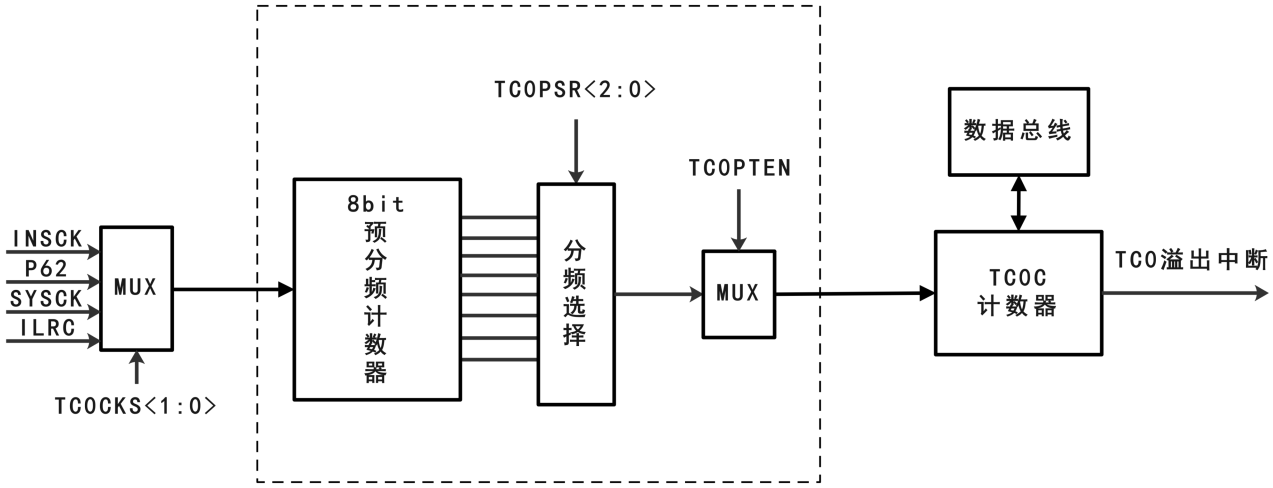
7.1 TC0 定时计数器

JZ8F8620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，TC0PSR<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），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 外部输入时钟可以中断溢出并唤醒电路。



TC0 结构框图

7.1.1 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；

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；

7.1.2 TC0 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）：

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

示例：

Fosc=8 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；

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

7.1.3 TC0 空闲模式唤醒说明

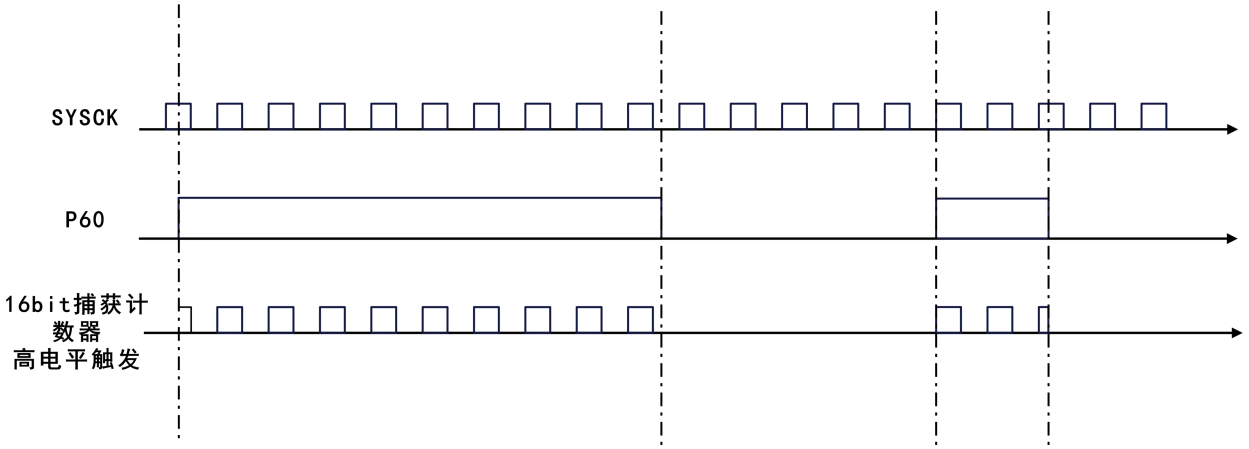
在空闲模式下，系统指令停止执行，具有唤醒功能的模块和系统时钟正常工作，其它功能被禁止。

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。

7.1.4 TC0 端口电平捕获说明

JZ8F8620 提供外部端口电平捕获功能，捕获的信号源可通过 EXINTCON（R1AE）寄存器选择 P60、P65、P50、P54 任意一个外部端口或者比较器的比较结果 CMPOUT 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 TC0C（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。

外部中断口电平捕获通过复用 TC0 预分频器和计数器实现，设置 EXINTCON 寄存器 INTGATE=1 使能 TC0 外部端口捕获，对 TC0 计数器进行配置，同时使能外部中断功能。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/TC0C、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。



16bit捕获计数器由低八位的TBRE和高八位的TC0组成

7.1.5 TC0 相关寄存器

7.1.5.1 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

7.1.5.2 R1AE/EXINTCON(外部中断捕获控制寄存器)

0X1AE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXINTCON	EXINT1MS<1:0>		EXINT0MS<1:0>		EXINT1S	EXINT0S	CAPS	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>：EXINT0MS<1:0>－EXINT0 模式选择位

EXINT0MS<1>	EXINT0MS<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>：EXINT0S－EXINT0 端口选择位

0：EXINT0 端口选择 P50

1：EXINT0 端口选择 P54

Bit<1>：CAPS－TC0 捕获输入源选择位（CMPGATE=1 时，CAPS 选择无效）

0：EXINT0

1：EXINT1

Bit<0>：INTGATE－TC0 外部端口捕获门控使能位

0：禁止

1：使能

7.1.5.3 R1C0/TC0CON(TC0 控制寄存器)

0X1C0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CON	TC0EN	TC0CKS<1:0>		TC0EDG	TC0PTEN	TC0PSR<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>：TC0EN－TC0 定时计数使能位

0：禁止 TC0

1：使能 TC0

Bit<6:5>：TC0CKS<1:0>－TC0 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）

Bit<4>：TC0EDG－TC0 信号边沿选择位

0：TC0 引脚信号发生由高到低变化加 1

1：TC0 引脚信号发生由低到高变化加 1

Bit<3>：TC0PTEN－预分频器使能位

0：禁止 TC0 预分频

1：使能 TC0 预分频

Bit<2:0>：TC0PSR<2:0>－TC0 预分频选择控制位：

TC0PSR<2>	TC0PSR<1>	TC0PSR<0>	TC0 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

7.1.5.4 R1C1/TC0C(TC08 位计数寄存器)

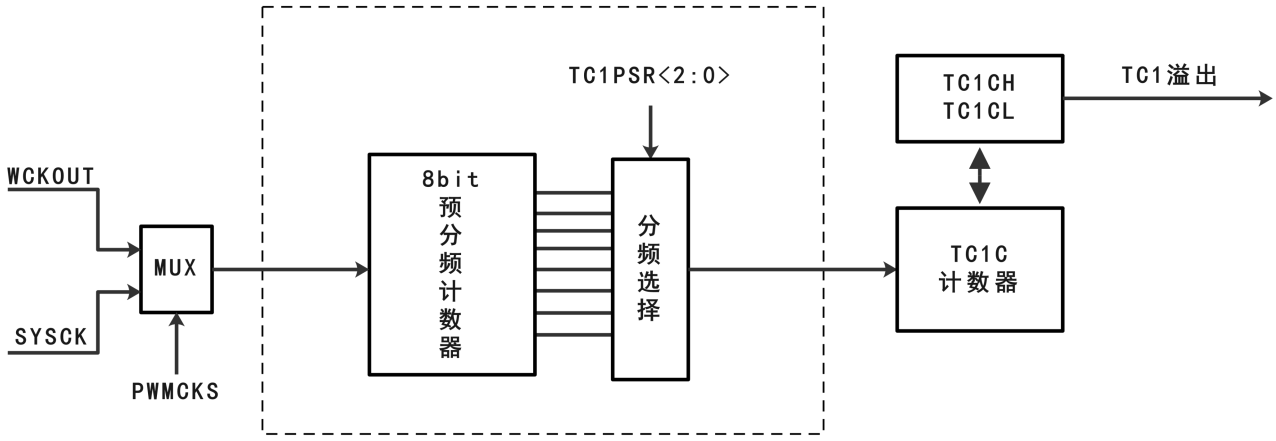
0X1C1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0C	TC0C<7:0>							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

7.2 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=16 MHz，TC1 分频选择=16 分频，TC1PRD 值=512；

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

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	PWM1GATE	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/PWM123 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/PWM123、TC2/PWM4567 计数器同时使能控制位

0：禁止

1：使能

Bit<4>：TC1CKS－TC1 时钟选择

0：系统时钟

（系统时钟为 32M，周期、占空比变化时禁止使用 1 分频，使用 2 分频最低工作电压 2.7V）

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/PWM123 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

7.2.4.3 R1B5/TC1PRDTH(TC1 定时计数器高 4 位寄存器)

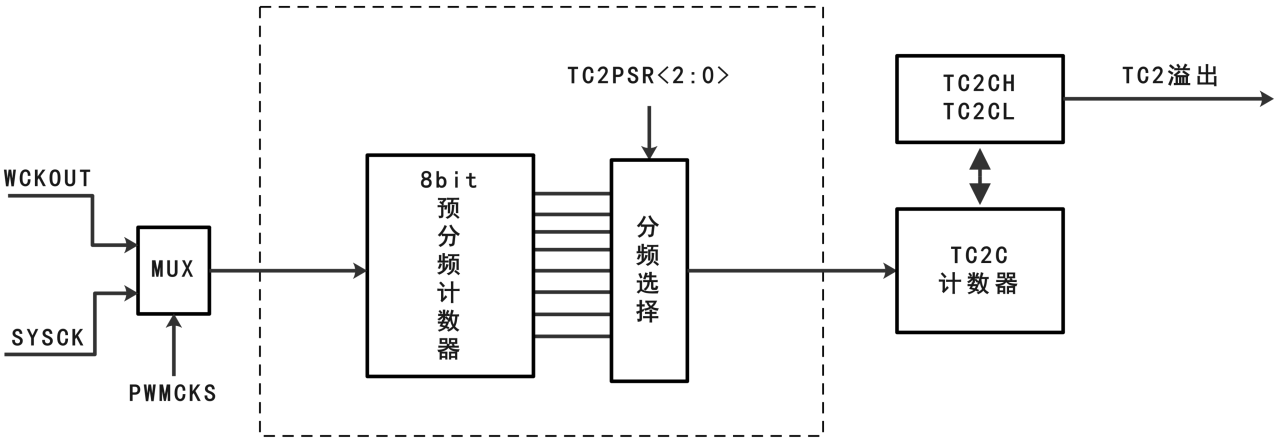
0X1B5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1PRDTH	TC1PRD<11:8>				PWM3DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

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：

$$TC2\ 定时时间 = (1/Fosc) \times (TC2\ 分频) \times (TC2PRD)$$

示例：

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

$$TC2\ 定时时间 = (1/16) \times (16) \times （512） = 512us$$

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	PWM4GATE	-	TC2CKS	TC2PTEN	TC2PSR<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>：TC2EN－TC2/PWM567 计数器使能控制

0：禁止

1：使能

Bit<4>：TC2CKS－TC2 时钟选择

0：系统时钟

（系统时钟为 32M，周期、占空比变化时禁止使用 1 分频，使用 2 分频最低工作电压 2.9V）

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 定时计数器低 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

7.3.4.3 R1BD/TC2PRDTH(TC2 定时计数器高 4 位寄存器)

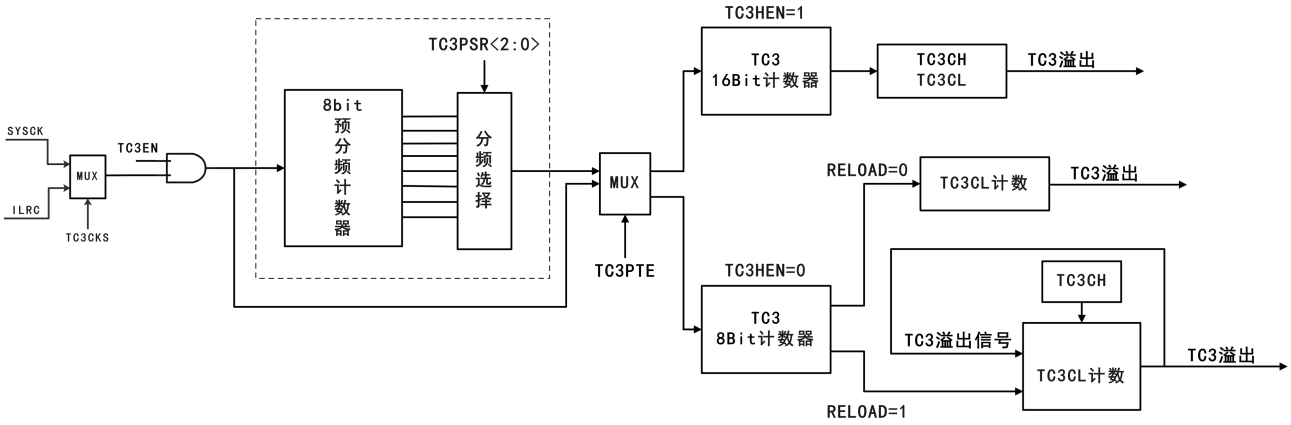
0X1BD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDTH	TC2PRD <11:8>				PWM6DT<11:8>			
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

7.4 TC3 定时计数器

JZ8F8620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 8/16 Bit 上行计数器，由 TC3HEN 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 TC3CH 不工作，仅计数器低位 TC3CL 在计数。当使能 RELOAD，即选择 TC3 为 8bit 可重载计数器，TC3CH 寄存器为重载赋值寄存器，此时 TC3HEN 置 0。TC3 定时器需使能 TC3EN 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。TC3 计数值计满产生溢出，若使能 TC3IE 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，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，方法参考端口输入变化唤醒。在睡眠模式下，TC3 设置为 ILRC 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），设置 TC3HEN=0 选择 TC3 作为 8bit 计数器功能，设置 TC3HEN=1 选择 TC3 作为 16 bit 计数器功能；

3、若选择 TC3 为 8bit 可重载计数器，设置 TC3CON 寄存器 RELOAD=1，设置 TC3CH 寄存器为重载赋值寄存器，此时 TC3HEN 置 0。

4、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；

5、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；

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 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。

TC3 定时时间计算公式（选择内部系统时钟）：

$$\text{TC3 定时时间} = (1/\text{Fosc}) \times (\text{TC3 分频}) \times (256 - \text{TC3 初始值})$$

示例：

Fosc=8MHz，TC3 分频选择=8 分频，TC3 初始值=100；

$$\text{TC3 定时时间} = (1/8) \times (8) \times (256 - 100) = 156\ \text{us}$$

TC3 定时时间计算公式（选择外部输入时钟）：

$$\text{TC3 定时时间} = (1/\text{外部输入时钟}) \times (\text{TC3 分频}) \times (256 - \text{TC3 初始值})$$

示例：

外部输入时钟=1 MHz，TC3 分频选择=4 分频，TC3 初始值=156；

$$\text{TC3 定时时间} = (1/1) \times (4) \times (256 - 156) = 400\ \text{us}$$

7.4.3 TC3 空闲模式唤醒说明

TC3 可以唤醒空闲模式，使能 R188/CPUCON 寄存器 Bit6 位 TC3WE，选择 TC3 时钟源为系统时钟。IDLE = 1 加上 SLEEP 指令系统进入空闲模式，TC3 定时器正常工作。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。

当时钟源选择 ILRC 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 相关寄存器

7.4.4.1 R1C8/TC3CON(TC3 控制寄存器)

0X1C8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC3CON	TC3EN	TC3CKS	TC3RELOAD	TC3HEN	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>：TC3CKS－TC3 时钟源选择位

0：系统时钟

1：低速时钟

Bit<5>：TC3RELOAD－TC3 重载模式选择位

0：TC3 为 8/16 bits 计数器，无重载功能

1：TC3 为 8 bits 可重载计数器，TC3CH 寄存器为重载赋值寄存器，保持 TC3HEN=0

Bit<4>：TC3HEN－TC3 高 8 位计数器选择位

0：TC3 为 8 bits 计数器

1：TC3 为 16 bits 计数器

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

7.4.4.2 R1C9/TC3CL(TC3 低 8 位计数寄存器)

0X1C9	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

7.4.4.3 R1CA/TC3CH(TC3 高 8 位计数寄存器)

0X1CA	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

8 PWM 脉宽调制

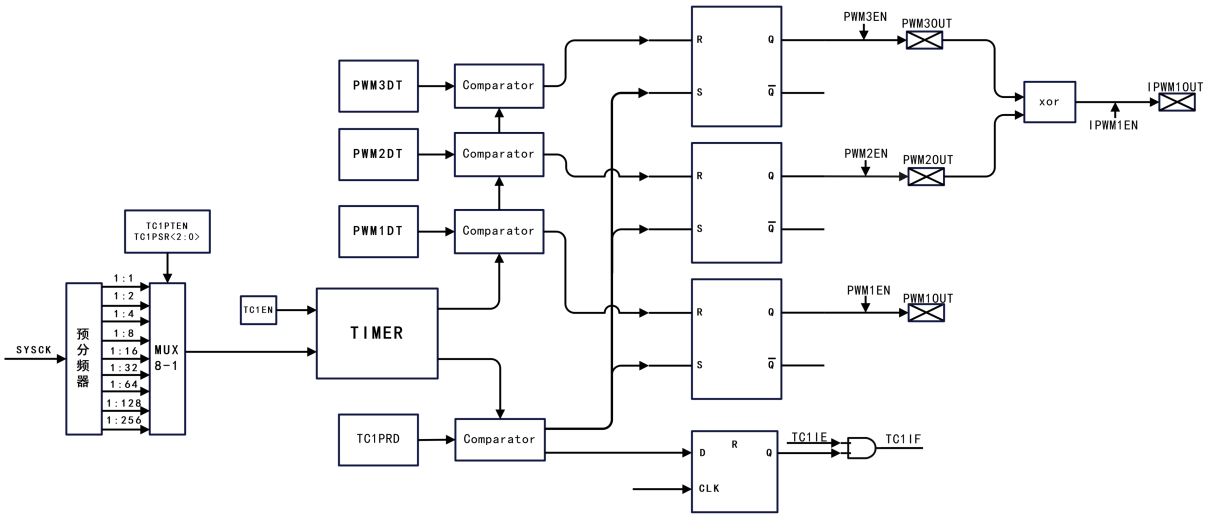
JZ8F8620 提供 2 组 12 bit TC1、TC2 计数器，两组计数器可分为共七路 PWM，PWM1 – 3 复用 TC1 计数器，PWM4 – 7 复用 TC2 计数器来产生脉宽调制信号，PWM 输出波形由周期及占空比决定，传输速率为周期倒数。PWM1 – 7 支持按位取反功能，可通过 0x1C4 寄存器对需要取反的一路进行使能。芯片提供死区互补输出，通过 PWM2 和 PWM3 异或后作为 IPWM1，与 PWM1 作为死区互补信号。(IPWM4 同 IPWM1)

PWM 带有周期溢出中断，使能后可跳转到中断。

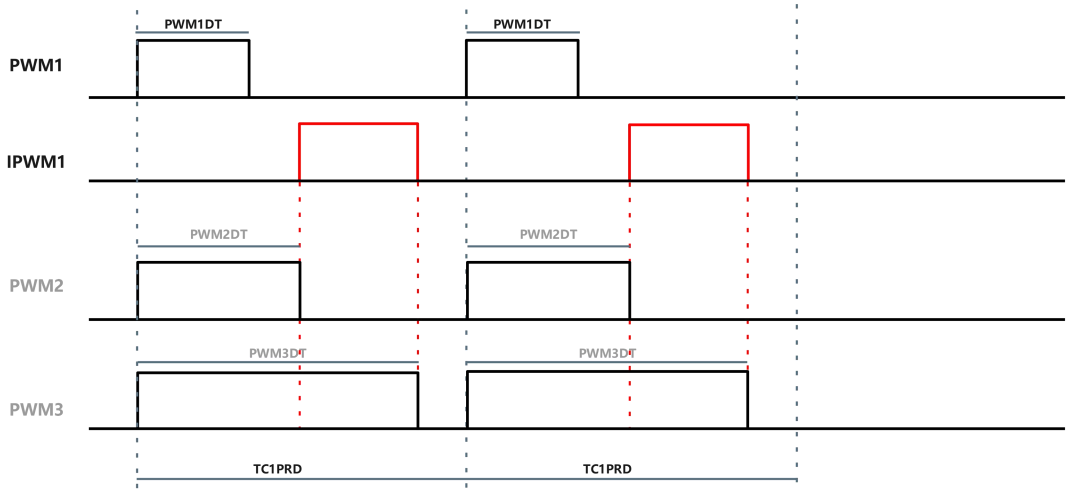
在 IDLE(空闲模式)下，TC1、TC2 在 TC1、TC2 控制寄存器中选择 TC1CKS=1、TC2CKS=1 并且在 CPU 模式控制寄存器中使能 TC1WE、TC2WE 可唤醒系统。

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 IPWM 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 控制输出口取反，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：

TCxPRDL=100，Fosc=16MHz，PWM 分频选择=2 分频；

$$\text{PWM 周期} = (100) \times (1/16) \times (2) = 12.5\ \text{us}$$

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：

TCxPRDL=50，Fosc=16MHz，PWM 分频选择=2 分频；

$$\text{PWM 占空比} = (50) \times (1/16) \times (2) = 6.25\ \text{us}$$

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，选择时钟源为系统时钟(SYSCK)。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 PWM 比较器门控说明

PWM1 的门控功能由 TC1CON 寄存器的 PWM1GATE 进行控制。当 PWM1GATE=1 时，PWM1 输出由比较器结果控制，CMP_FLAG=0 时输出 PWM 波形。

PWM4 的门控功能由 TC2CON 寄存器的 PWM4GATE 进行控制。当 PWM4GATE=1 时，PWM4 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

IPWM 波形不是直接取反，而是使能 IPWM1EN，将 PWM2 和 PWM3 进行异或后进行输出。若此时使能 IPWM1EN，可以通过 PWM2 输出选择控制位选择从 P61/P67 输出；使能 PWM3EN，可以通过 PWM3 输出选择控制位选择从 P50/P65 输出。同时使能支持同时从两个端口输出 IPWM1 波形。

同理使能 IPWM4EN，是将 PWM5 与 PWM6 异或后进行输出。若此时使能 PWM5EN，可以通过 PWM5 输出选择控制位选择从 P52/P63 输出；使能 PWM6EN，可以通过 PWM6 输出选择控制位选择从 P55/P64 输出。同时使能支持同时从两个端口输出 IPWM4 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 PWM4S（R1BF）进行端口选择，分别从 P60/P66 口与 P51/P62 口输出，需要注意的是，使能蜂鸣器需将 PWM1EN 与 PWM4EN 清零并将对应输出口设置为输出。

8.6 PWM7DT 中断以及触发 ADC 说明

PWM7DT 提供占空比触发中断的功能，在使能 PWM7DTIE（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 PWM7 占空比触发 ADC 转换使能，通过配置 ADCGATE0（R1A7）置 1，那么 ADC 转换将在 PWM7 占空比下降沿被触发。

8.7 PWM 刹车功能

JZ8F8620 支持 PWM 刹车功能，通过 CMPCON（R1F8）寄存器进行配置，刹车源为 CMPIF 来触发，CMPIF 高电平期间刹车，刹车模式为立即刹车，刹车共支持 PWM4－7 四路，可通过 CON_SEL0<1:0>CMP 联动通道进行选择。

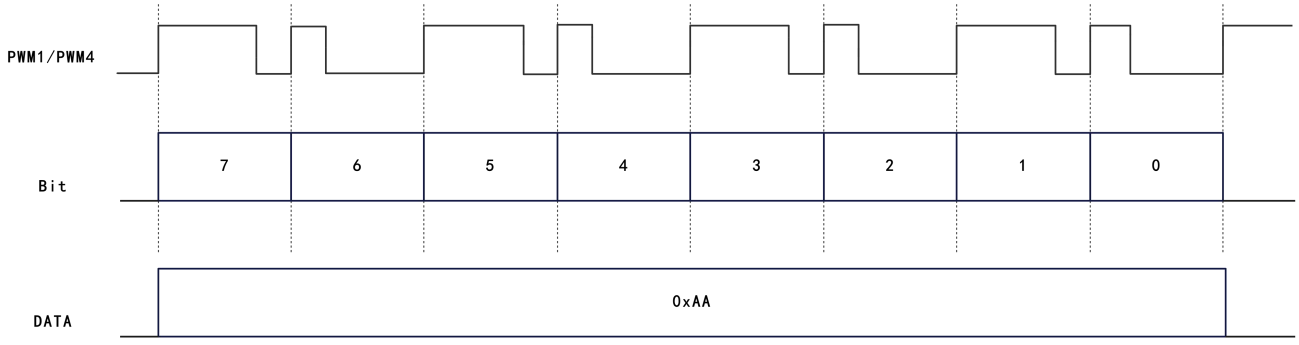
用户还可通过根据需求配置 CON_OUT 来选择联动输出电平，为 PWM4－7 四路输出选择刹车后停留的高低电平。如联动通道 PWM7，并且使能联动输出，PWM7 选择了 P57 口进行输出，联动输出电平配置为 1 高电平，那么 CMPIF 在刹车触发后 PWM7 将会保持高电平从 P57 口输出。

8.8 PWMINV 取反功能

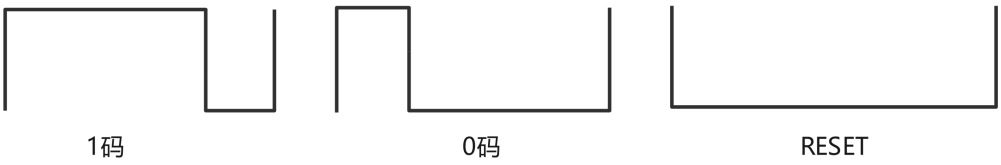
JZ8F8620 支持 7 路 PWM 取反输出功能。通过设置 PWM 取反控制寄存器（R1C4）进行配置，配置好 PWMx 的周期，占空比。使能 TC1/TC2，并使能相关 PWM 取反控制 bit 位，PWMx 将在对应端口上进行取反输出。

8.9 LED 单线级联

JZ8F8620 支持 2 路单线级联 LED 驱动，通过 PWM1、PWM4 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 PWM4 两路独立的单线级联 LED 驱动控制。

1.PWM1 作为 LED 驱动控制：

- 0 码型的高电平时间：PWM1 占空比寄存器（PWM1DTL）设定；
- 1 码型的高电平时间：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.PWM4 作为 LED 驱动控制：

- 0 码型的高电平时间：PWM4 占空比寄存器（PWM4DTL）设定；
- 1 码型的高电平时间：PWM5 占空比寄存器（PWM5DTL）设定；
- 码元周期：TC2PRD 周期寄存器（TC2PRDL）设定；
- 级联数据：PWM6 占空比寄存器（PWM6DTL）设定；

在使能 PWM4_LEDEN（R186）+TC2EN 后开始发送 8 位级联数据，数据由 PWM4 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。

8.10 PWM 相关寄存器

8.10.1 R186/TBRDH(查表指针高位寄存器)

0X186	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBRDH	PWM4_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>：PWM4_LEDEN－PWM4_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

8.10.2 R1B0/TC1CON(TC1 控制寄存器)

0X1B0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1CON	TC1EN	TC21EN	PWM1GATE	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/PWM123 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/PWM123、TC2/PWM4567 计数器同时使能控制位

0：禁止

1：使能

Bit<5>：PWM1GATE－PWM1OUT 比较器门控

0：禁止

1：使能

（PWM1 输出由 CMPOUT 控制，当 CMPOUT=0 时输出 PWM 波形，CMPOUT=1 时保持 PWM1 为低电平）

Bit<4>：TC1CKS－TC1 时钟选择

0：系统时钟

（系统时钟为 32M，周期、占空比变化时禁止使用 1 分频，使用 2 分频最低工作电压 2.7V）

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

8.10.3 R1B1/TC1PRDL(TC1/PWM123 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

8.10.4 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

8.10.5 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

8.10.6 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

8.10.7 R1B5/TC1PRDTH(TC1/PWM123 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）周期高 4 位及 PWM3 占空比高 4 位

8.10.8 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

8.10.9 R1B7/PWMCON1(PWM 控制寄存器 1)

0X1B7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON1	BZ1EN	PWM3S	PWM2S	PWM1S	IPWM1EN	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/P66 为输出才能输出 BZ1 波形）

Bit<6>：PWM3S－PWM3 输出选择控制位

0：PWM3 从 P65 输出

1：PWM3 从 P50 输出

Bit<5>：PWM2S－PWM2 输出选择控制位

0：PWM2 从 P61 输出

1：PWM2 从 P67 输出

Bit<4>：PWM1S－PWM 输出选择控制位

0：PWM1 从 P60 输出

1：PWM1 从 P66 输出

Bit<3>：IPWM1EN－IPWM1 使能控制位

0：禁止

1：输出 PWM2 与 PWM3 异或

（根据 PWM2S（P61/P67)或 PWM3S（P65/P50)选择输出端口）

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）

8.10.10 R1B8/TC2CON(TC2 控制寄存器)

0X1B8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2CON	TC2EN	PWM4GATE	-	TC2CKS	TC2PTEN	TC2PSR<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>：TC2EN－TC2/PWM4567 计数器使能控制

0：禁止

1：使能

Bit<6>：PWM4GATE－PWM4 门控输出使能控制位 1

0：无限制

1：PWM4 输出由外部中断 EXINT0 控制，当 EXINT0=1 输出 PWM，否则输出低电平

Bit<5>：未定义

Bit<4>：TC2CKS－TC2 时钟选择

0：系统时钟

（系统时钟为 32M，周期、占空比变化时禁止使用 1 分频，使用 2 分频最低工作电压 2.9V）

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

8.10.11 R1B9/TC2PRDL(TC2/PWM4567 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

8.10.12 R1BA/PWM4DTL(PWM4 占空比低 8 位寄存器)

0X1BA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DTL	PWM4DT<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

8.10.13 R1BB/PWM5DTL(PWM5 占空比低 8 位寄存器)

0X1BB	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

8.10.14 R1BC/PWM6DTL(PWM6 占空比低 8 位寄存器)

0X1BC	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

8.10.15 R1BD/TC2PRDTH(TC2/PWM4567 周期高 4 位及 PWM6 占空比高 4 位寄存器)

0X1BD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2PRDTH	TC2PRD <11:8>				PWM6DT<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（PWM4、PWM5、PWM6、PWM7）周期高 4 位及 PWM6 占空比高 4 位

8.10.16 R1BE/PWM54DTH(PWM54 占空比高 4 位寄存器)

0X1BE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM54DTH	PWM5DT<11:8>				PWM4DT<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

PWM5 占空比高 4 位，PWM4 占空比高 4 位

8.10.17 R1BF/PWMCON2(PWM 控制寄存器 2)

0X1BF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON2	BZ2EN	PWM6S	PWM5S	PWM4S	IPWM4EN	PWM6EN	PWM5EN	PWM4EN
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 P62/P51 为输出才能输出 BZ2 波形）

Bit<6>：PWM6S－PWM6 输出选择控制位

0：PWM6 从 P64 输出

1：PWM6 从 P55 输出

Bit<5>：PWM5S－PWM5 输出选择控制位

0：PWM5 从 P63 输出

1：PWM5 从 P52 输出

Bit<4>：PWM4S－PWM4 输出选择控制位

0：PWM4 从 P62 输出

1：PWM4 从 P51 输出

Bit<3>：IPWM4EN－IPWM4 使能控制位

0：禁止

1：输出 PWM5 与 PWM6 异或

（根据 PWM5EN（P63/P52)或 PWM6EN（P55/P64)选择输出端口）

Bit<2>：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<1>：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）

Bit<0>：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，且 BZ2EN=0）

8.10.18 R1C2/PWM7DTL(PWM7 占空比低 8 位寄存器)

0X1C2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7DTL	PWM7DTL<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

8.10.19 R1C3/PWM7CON(PWM7 控制及占空比高位寄存器)

0X1C3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7CON	CMPGATE	PWM_PULSE_EN	PWM7S	PWM7EN	PWM7DTH<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<6>：PWM_PULSE_EN—PWM7 占空比单次触发使能位

0：禁止

1：使能

Bit<5>：PWM7S－PWM7 输出选择控制位

0：PWM7 从 P57 输出

1：PWM7 从 P54 输出

Bit<4>：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<3:0>：PWM7DTH<11:8>－PWM7 占空比高 4 位

8.10.20 R1C4/PWMINVCON(PWM 取反控制寄存器)

0X1C4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMINVCON	-	PWM7INV	PWM6INV	PWM5INV	PWM4INV	PWM3INV	PWM2INV	PWM1INV
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>：PWM7INV－PWM7 输出取反使能位

0：禁止

1：使能

Bit<5>：PWM6INV－PWM6 输出取反使能位

0：禁止

1：使能

Bit<4>：PWM5INV－PWM5 输出取反使能位

0：禁止

1：使能

Bit<3>：PWM4INV－PWM4 输出取反使能位

0：禁止

1：使能

Bit<2>：PWM3INV－PWM4 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

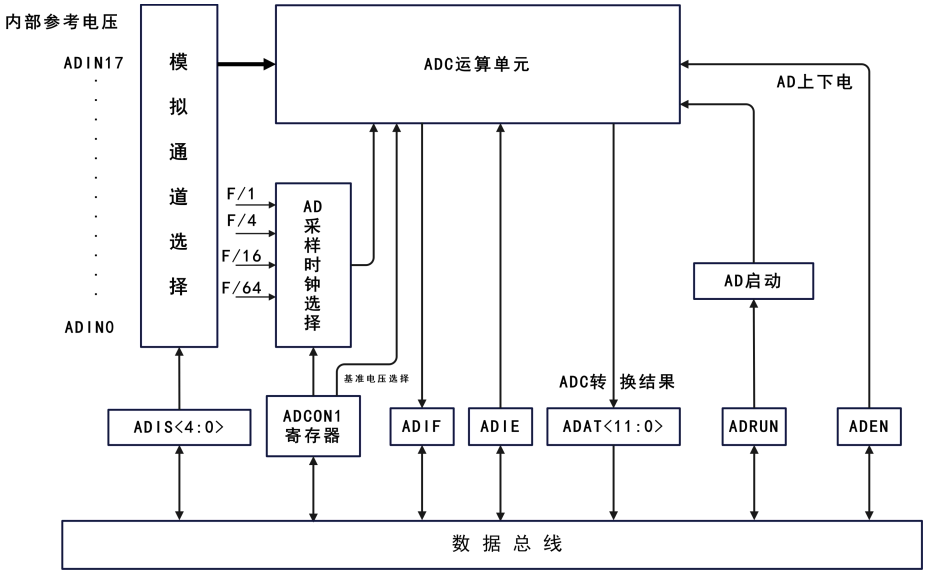
9 ADC 模数转换

JZ8F8620 ADC 模块提供一个 12 位精度的 SAR AD 转换器，包含 18 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/ADCON0(ADC 控制寄存器 0)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 功耗。

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

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

9.1 ADC 检测电源电压说明

ADC 测 VDD 电源电压有以下两种方式：

方式一：0.255VDD

通过配置 ADCCON0(R1A6)寄存器，通道选择位 ADIS<4:0>=10010 选择 0.255VDD 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：

$$\text{转换结果}=（\text{VDD}*0.255/\text{VREF}）*4096$$

则

$$\text{VDD} = (\text{转换结果}/4096）/ 0.255 * \text{VREF}$$

方式二：VREF 反推 VDD

通过配置 ADCCON0(R1A6)寄存器，通道选择位 ADIS<4:0>=10011 选择 VREF 作为采样通道，此时 ADC 基准电压系统自动设置为 VDD。VREF<2:0>选择参考电压作为采样电压，通过 ADC 转换得到转换结果，根据得到的结果反推 VDD 电压：

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

则

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

9.2 P51 端口检测 0.255 外部电压说明

JZ8F8620 内置一路采集外部电压的 0.255 分压电阻通道，可以从 P51 端口输入一个电压，电压值不能高于 VDD，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，通过计算后能出此电压值：

$$\text{P51 输入电压} = \frac{\text{采样值}}{4096} * \text{参考电压} / 0.255$$

例：VDD=5V，参考电压为内部基准 2V，采样值为 0x800=2048

$$\text{P51 输入电压} = \frac{2048}{4096} * 2 / 0.255 = 3.92\text{V}$$

9.3 ADC 模数转换设置说明

1、ADC 输入口使能及模拟通道选择，设置 R1A0/P5ADE、R1A1/P6ADE、R1A2/P7AE 寄存器；

2、ADC 参考电压、时钟分频，设置 R1A6/ADCON0、R1A7/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；

9.4 ADC 相关寄存器

9.4.1 R185/WKCON(唤醒控制寄存器)

0X185	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WKCON	-	-	-	-	-	ADCGATE1	-	UARTWE
读/写	R	R	R	R	R	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0

Bit<2>：ADCGATE1 - CMP 结果变化触发 ADC 门控使能位

0：禁止

1：使能

9.4.2 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

9.4.3 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

9.4.4 R1A5/ADATH1(ADC 数据高 4 位寄存器)

OX1A5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADATH1	VREFOUT	-	-	-	ADAT<11:8>			
读/写	R/W	R	R	R	R	R	R	R
复位值	0	0	0	0	X	X	X	X

Bit<7>：VREFOUT－参考电压输出使能

0：禁止

1：使能（P65 端口设为模拟输入）

Bit<3:0>：ADAT<11:8>－AD 转换结果高 4 位

9.4.5 R1A6/ADCON0(ADC 控制寄存器 0)

OX1A6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADIS<4:0>					VCMPS	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<7:3>：ADIS<4:0>－AD 输入口选择

ADIS<4>	ADIS<3>	ADIS<2>	ADIS<1>	ADIS<0>	通道选择
0	0	0	0	0	AIN0/P50
0	0	0	0	1	AIN1/P51
0	0	0	1	0	AIN2/P52
0	0	0	1	1	AIN3/P53
0	0	1	0	0	AIN4/P54
0	0	1	0	1	AIN5/P55
0	0	1	1	0	AIN6/P56
0	0	1	1	1	AIN7/P57
0	1	0	0	0	AIN8/P60
0	1	0	0	1	AIN9/P61
0	1	0	1	0	AIN10/P62
0	1	0	1	1	AIN11/P63
0	1	1	0	0	AIN12/P64
0	1	1	0	1	AIN13/P65
0	1	1	1	0	AIN14/P66
0	1	1	1	1	AIN15/P67
1	0	0	0	0	AIN16/P70
1	0	0	0	1	AIN17/P71
1	0	0	1	0	0.255*（VDD 或 P51）

1	0	0	1	1	VREF(VDD 作基准)
1	0	1	0	0	GND

Bit<2>：VCMPS－ADC 比较电压源选择

0：选择 VREF（VDD 为参考电压）

1：选择 VDD（内部参考电压）

Bit<1>：EXTVS－采样通道过 0.255 电阻分压源选择

0：选择 VDD 作为 0.255 电阻分压源

1：选择 P51 作为 0.255 电阻分压源（分压前电压不能超过 VDD）

Bit<0>：ADCAP－ADC 采样通道电容选择控制位

0：禁止

1：使能

9.4.6 R1A7/ADCON1(ADC 控制寄存器 1)

0X1A7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADRUN	ADEN	ADPSR<1:0>		ADCGATE0	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

0：ADC 无采样或采样结束

1：ADC 开始采样

Bit<6>：ADEN

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（推荐）
1	1	Fosc/1

Bit<3>：ADCGATE0－ADC 门控选择位

0：门控禁止，由软件 ADCEN 使能

1：门控使能，由 PWM7 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（不推荐）

0	0	1	基准 2V
0	1	0	基准 3V
0	1	1	基准 4V
1	0	0	EXVREF（P61）
1	X	1	VDD

10 CMP 比较器

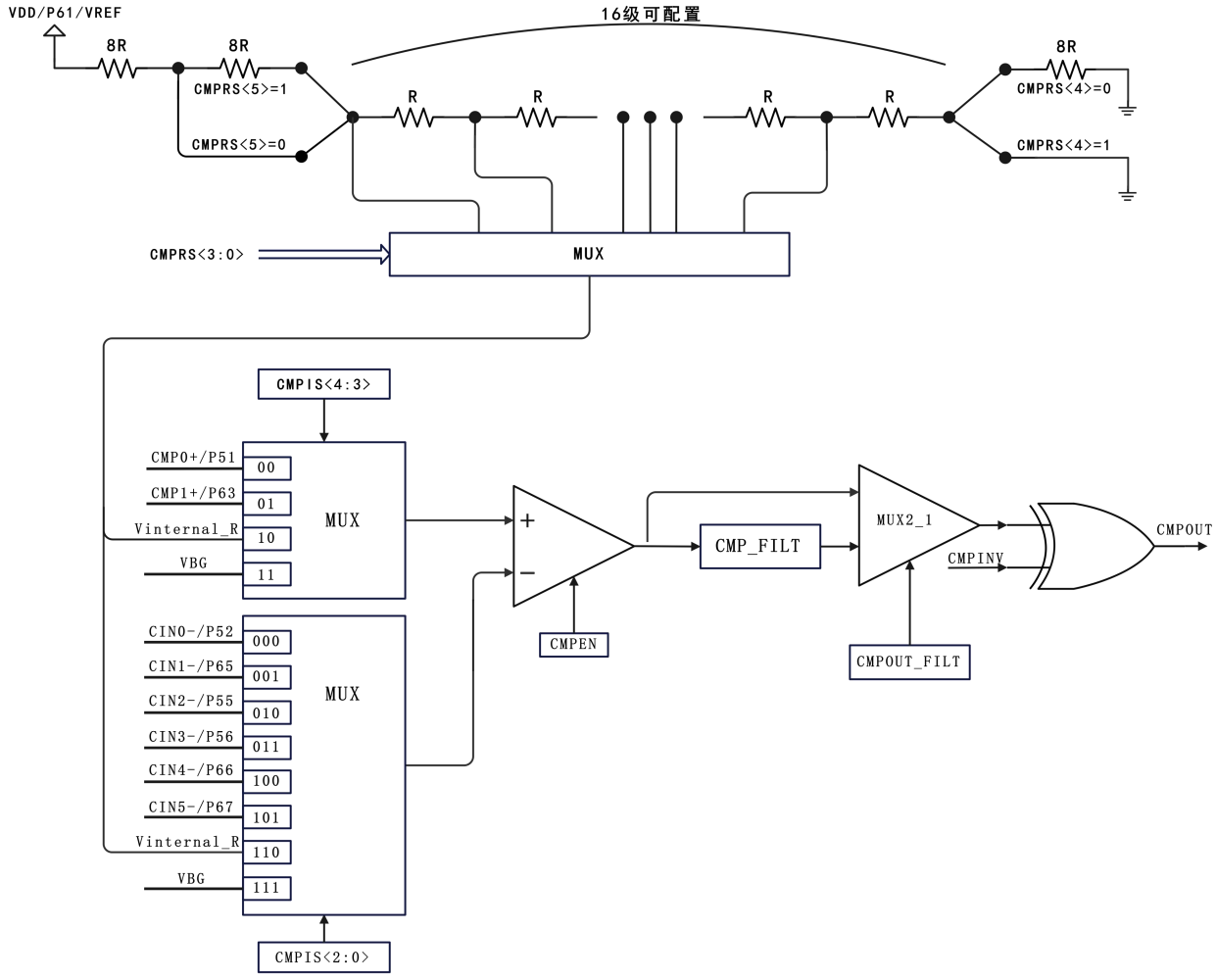
JZ8F8620 内置一个硬件比较器，可以从输入引脚、电阻分压 $V_{internal\ R}$ 进行比较，选择其中一个作为正输入，选择其中另一个作为负输入。比较器的负输入可以是端口、分压电阻（由 CMPRS<5:0>选择）、VBG。比较器的正输入可以是端口、分压电阻（由 CMPRS<5:0>选择）、VBG。(同一端口不可同时作为正极和负极输入,分压电阻输入源也不能同时作为正极和负极。)

分压电阻通过 CMPCON 寄存器配置选择输入电压源为 P61 口输入、VDD 或内置基准 VREF。

比较器的输出结果可通过 CMPCON(1F8)寄存器 CMPOE 位选择 P61 口输出。芯片支持输出结果数字滤波，迟滞可在 OPTION 中选择。

比较器支持正负极交换进行比较，可通过 CMPCON1 寄存器 CMPINV 位使能正负极交换。

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，可以在 CMPCON 选择比较器结果的上升沿或者下降沿触发。

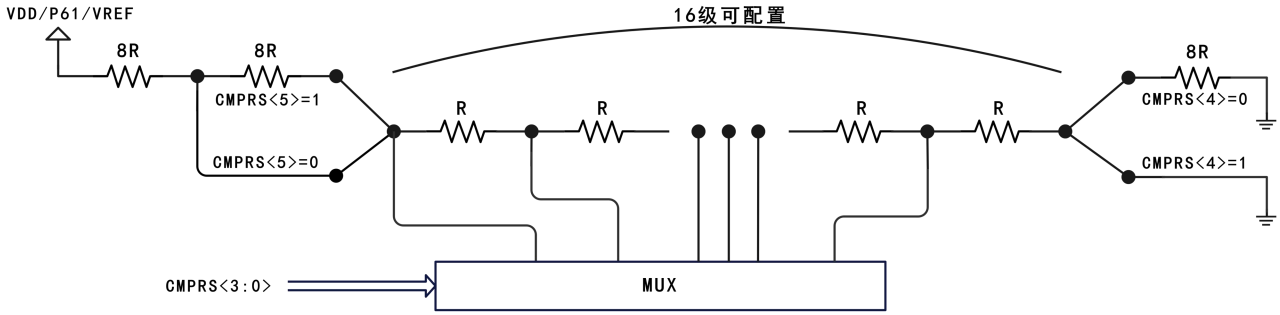


比较器原理图说明

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。

10.1 分压电阻输出电压 Vinternal R

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



比较器修调示意图

例 1：CMPRS<5>=0，CMPRS<4>=0

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

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

例 2：CMPRS<5>=0，CMPRS<4>=1

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

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

例 3：CMPRS<5>=1，CMPRS<4>=0

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

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

例 4：CMPRS<5>=1，CMPRS<4>=1

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

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

n=CMPRS<3:0>

10.2 比较器配置

使用比较器时需使能 CMPEN（此时会打开用于对输出结果 CMPOUT 滤波的 LRC，功耗会增加），设置 CMPIS<4:0>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 CMPRS<5:0>，选择所需的分压电阻输出电压 $V_{internal\ R}$ 进行比对。

例 1：

选择 P52 作为负端输入和 $V_{internal\ R}$ 的电压为(18/32)*VDD 作为正输入，$V_{internal\ R}$ 选择 CMPRS<5>=0，CMPRS<4>=0 的配置方式，CMPRS<3:0>=4b’1001(n=9) 以得到

Vinternal R = (1/4) * VDD + [(9+1)/32] * VDD = (18/32) * VDD

当配置完成后，使能 CMPCON0(R1F9)控制寄存器的 Bit7 位 CMPEN，比较器开始对比，并可通过配置 CMPCON1 寄存器中 CMPOE 位选择将比较结果从 P61 输出来（P61 口需设为输出），也可通过配置 CMPCON1 寄存器的 Bit7 位 CMPINV 将输出结果取反。

10.3 RFC 电阻频率转换

JZ8F8620 内置 RFC 电阻频率转换功能，通过使用比较器的比较结果控制 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 CMPGATE 位后，TC0 计数由比较器的输出结果 CMPOUT 控制。RFC 端口电压低于比较器正极电压时，CMPOUT=1 时 TC0 开始计数；RFC 端口电压高于比较器正极电压时，CMPOUT=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；

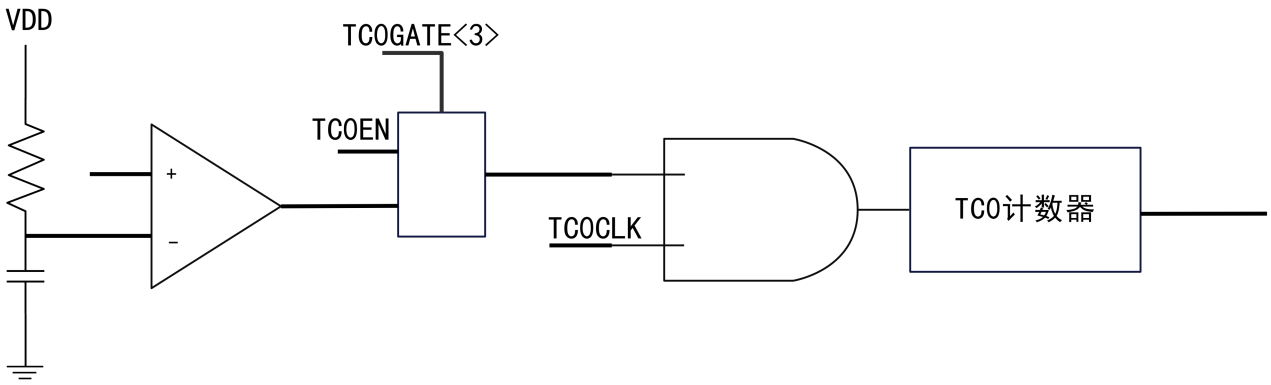
2、设置 CMPCON0、CMPCON1 寄存器，设定比较器的正极输入源为固定电压，设定比较器的负极输入源为外部端口（以图示为例）；

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 TC0 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

10.4 CMP 相关寄存器

10.4.1 R1C3/PWM7CON(RFC 门控控制寄存器)

0X1C3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7CON	CMPGATE	PWM_PULSE_EN	PWM7S	PWM7EN	PWM7DTH<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>：CMPGATE－TCO 捕获门控使能位

0：禁止

1：使能门控，CMPOUT 作为捕获源

10.4.2 R1F8/CMPCON(CMP 总控制寄存器)

0X1F8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON	-	CMPPS	CMPFILT	CMPOE	CON_OUT	CON_SEL<1:0>		CON_EN
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>：CMPPS－CMP 比较器边沿触发控制位

0：CMPOUT 上升沿触发

1：CMPOUT 下降沿触发

Bit<5>：CMPFILT－比较器 CMP 结果输出内部滤波选择

0：CMP 比较结果滤波关闭

1：CMP 比较结果滤波使能

Bit<4>：CMPOE－比较器 CMP 结果输出使能

0:P61 作为 GPIO

1:比较器结果从 P61 输出

Bit<3>：CON_OUT－CMP 联动输出电平

0：CMP 联动输出口低电平

1：CMP 联动输出口高电平

Bit<2:1>：CON_SEL<1:0>－CMP 联动通道选择

CON_SEL<1>	CON_SEL<0>	CMP 联动通道选择
0	0	PWM4
0	1	PWM5
1	0	PWM6
1	1	PWM7

Bit<0>：CON_EN－CMP 联动输出使能位

0：禁止

1：使能

10.4.3 R1F9/CMPCON0(CMP 控制寄存器 0)

0X1F9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON0	CMPEN	CMPOUT	CMPRS<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>：CMPEN－CMP 使能控制位

0：禁止

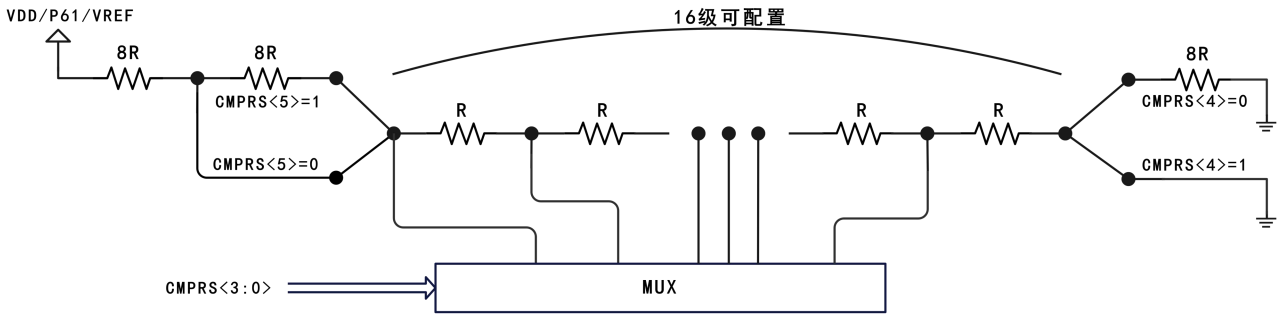
1：使能

Bit<6>：CMPOUT－CMP 结果输出位

0：比较器正极电压小于负极电压

1：比较器正极电压大于负极电压

Bit<5:0>：CMPRS<5:0>－比较器修调位



当 CMPCON1 寄存器=0x17，分压电阻输入选择 VDD，比较器正极选择电阻分压，比较器负极选择 VBG，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（灰色部分检测电压禁止使用）：

CMPRS	检测电压	CMPRS	检测电压	CMPRS	检测电压	CMPRS	检测电压
00H	4.27V	10H	28.80 V	20H	5.33V	30H	38.40 V
01H	3.84V	11H	14.40 V	21H	4.80V	31H	19.20 V
02H	3.49V	12H	9.60 V	22H	4.36V	32H	12.80 V
03H	3.20V	13H	7.20 V	23H	4.00V	33H	9.60 V
04H	2.95V	14H	5.76 V	24H	3.69V	34H	7.68 V
05H	2.74V	15H	4.80 V	25H	3.43V	35H	6.40 V
06H	2.56V	16H	4.11 V	26H	3.20V	36H	5.49 V
07H	2.40V	17H	3.60 V	27H	3.00V	37H	4.80 V
08H	2.26V	18H	3.20 V	28H	2.82V	38H	4.27 V
09H	2.13V	19H	2.88 V	29H	2.67V	39H	3.84 V
0AH	2.02V	1AH	2.62 V	2AH	2.53V	3AH	3.49 V
0BH	1.92V	1BH	2.40 V	2BH	2.40V	3BH	3.20 V
0CH	1.83V	1CH	2.22 V	2CH	2.29V	3CH	2.95 V
0DH	1.75V	1DH	2.06 V	2DH	2.18V	3DH	2.74 V
0EH	1.67V	1EH	1.92 V	2EH	2.09V	3EH	2.56 V
0FH	1.60V	1FH	1.80 V	2FH	2.00V	3FH	2.40 V

10.4.4 R1FA/CMPCON1(CMP 控制寄存器 1)

0X1FA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON1	CMPINV	CMPIS<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>：CMPINV－CMP 正负极交换

0：禁止

1：使能

Bit<6:5>：CMPIS<6:5>－分压电阻输入电压源选择

CMPIS<6>	CMPIS<5>	分压电阻输入源
0	0	VDD
0	1	P61
1	0	VREF

Bit<4:3>：CMPIS<4:3>－比较器正极输入源选择

CMPIS<4>	CMPIS<3>	正极输入源
0	0	CIN0+/P51

0	1	CIN1+/P63
1	0	电阻分压
1	1	VBG

Bit<2:0>：CMPIS<2:0>－比较器负极输入源选择

CMPIS<2>	CMPIS<1>	CMPIS<0>	负极输入源
0	0	0	CIN0－/P52
0	0	1	CIN1－/P65
0	1	0	CIN2－/P55
0	1	1	CIN3－/P56
1	0	0	CIN4－/P66
1	0	1	CIN5－/P67
1	1	0	电阻分压
1	1	1	VBG

11 LED 点阵驱动

JZ8F8620 芯片内置 LED 驱动模块，该驱动电路支持最大 8*9 模式 72 灯 LED 驱动，可配置 LEDCOM<8:0>任意组合选择不同的点阵模式，对应 IO 设定为 LED 端口。手动扫描模式下，LEDCOM<8:0>选择当前扫描的 COM 口，由 LEDDTS<8:0>选择 IO 设定为 LED 端口。

不同大小的点阵对应的灯地址不变，LED 数据通过 R1F6 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 8Bit。LED 时钟源可以选择系统时钟或者指令时钟，通过 LEDPSR<5:0>对时钟进行分频。

灯的亮度在自动模式下，根据 LEDDTS<7:0>选择对应 LED COMx 口占空比选择配置寄存器为 LEDDT0<7:0>或 LEDDT1<7:0>；手动模式下，则灯的亮度固定为 LEDDT0<7:0>。

如果使能 LEDIE，在每次 COM 扫描结束后都会进入中断。一般在手动模式下，进中断可重新配置 LEDDTS<8:0>选择扫描的点阵模式，配置 LEDCOM<8:0>选择扫描的 COM 口，配置 LEDDT0<7:0>选择扫描的亮度。

11.1 LED 功能描述

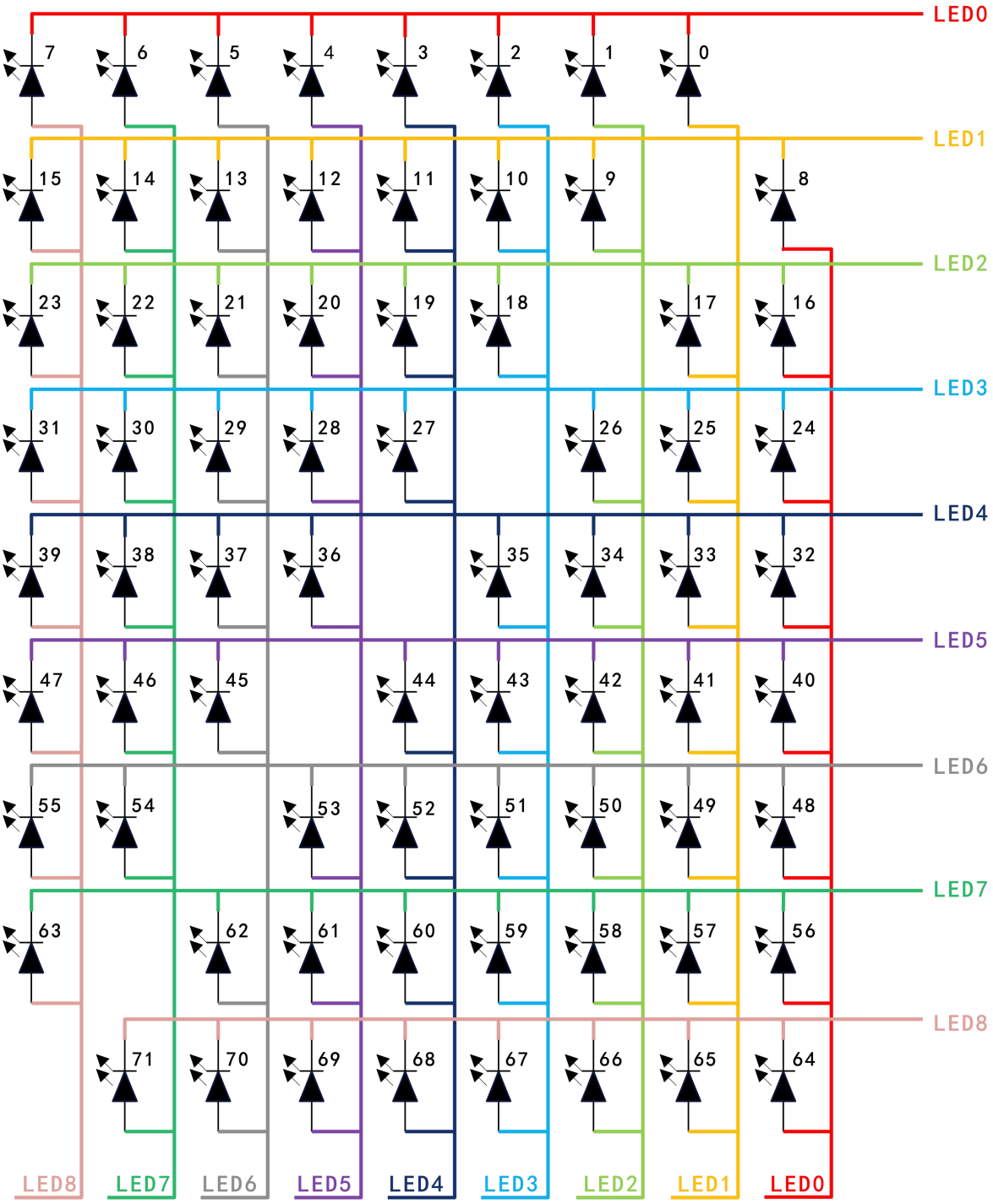
LED 点阵扫描模式，一般一次扫描整行（LEDRAMx，1 个 COM），从第一行（COM0）开始逐渐往下扫描，用户可自行组合 COM。使用自动模式会自动向下扫描，使用手动模式需要在一次扫描结束后使能下一次扫描的 COM。LEDRAM0~8 应表格中显示对应地址的亮灯情况，1 表示亮灯，0 表示不亮灯。

设置寄存器 LEDEN=1，将可启用 LED 点矩阵模式。设置 LEDDTx<7:0>为 0xFF 时亮度最高。下表红色标注为 RSR 当前地址下，LEDDB 为该值时点亮的对应的灯。

LEDRAM	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RSR	LEDDB
LEDRAM0	7	6	5	4	3	2	1	0	0	0b0000 0001
LEDRAM1	15	14	13	12	11	10	9	8	1	0b0000 0010
LEDRAM2	23	22	21	20	19	18	17	16	2	0b0000 0100
LEDRAM3	31	30	29	28	27	26	25	24	3	0b0000 1000
LEDRAM4	39	38	37	36	35	34	33	32	4	0b0001 0000
LEDRAM5	47	46	45	44	43	42	41	40	5	0b0010 0000
LEDRAM6	55	54	53	52	51	50	49	48	6	0b0100 0000
LEDRAM7	63	62	61	60	59	58	57	56	7	0b1000 0000
LEDRAM8	71	70	69	68	67	66	65	64	8	0b0000 0001

下图举例最大支持 8*9 矩阵扫描，其它方式类推。

8*9 矩阵：


LED0
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
7 6 5 4 3 2 1 0
15 14 13 12 11 10 9 8
23 22 21 20 19 18 17 16
31 30 29 28 27 26 25 24
39 38 37 36 35 34 33 32
47 46 45 44 43 42 41 40
55 54 53 52 51 50 49 48
63 62 61 60 59 58 57 56
71 70 69 68 67 66 65 64
LED8 LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0

11.2 LED 点阵设置说明

1. 设置寄存器 LEDDB 写入值，对应点亮相应的 LED 灯。
2. 配置 LEDCON0 使能 LED，设置 LED 自动模式或者手动模式，并选择 LED 时钟源和分频比以及按需使能驱动电流增强。
3. 配置 LEDCOM<8:0>。自动模式下选择使能对应 COM 口（可自由组合）；手动模式下选择当前扫描的 COM 口。
4. 配置 LEDDT0/1 寄存器设置亮度。如果选择了自动模式，配置 LEDDTS<8:0>对应 COM 选择 LEDDT0/1 为 LED 亮度；手动模式固定 LEDDT0 为 LED 亮度，配置 LEDDTS<8:0>设置 LED 口。
5. 根据需要选择使能 LEDIE，一般在手动模式下，在中断程序中重新配置点阵模式，下一次扫描的 COM 口以及 LED 亮度。

11.3 LED 点阵相关寄存器

11.3.1 R1F0/LEDCON0(LED 控制寄存器 0)

0X1F0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON0	LEDEN	LEDMS	LEDCKS	LEDPSR<1:0>		LEDDRV<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

注：手动扫描模式下，每次 COM 结束后 LEDEN 将复位，重新配置好参数后需再次使能 LEDEN。

Bit<6>：LEDMS－LED 模式选择位

0：自动扫描（双 DUTY，由 LEDDTS<8:0>选择。LEDCOM<8:0>选择需要扫描的 COM 口同时选择对应的 IO 为 LED 口，COM 口可任意组合。）

1：手动扫描（默认 DUTY 为 LEDDT0，LEDDTS<8:0>选择 IO 为 LED 口，LEDCOM<8:0>选择当前需要扫描的 COM 口。）

Bit<5>：LEDCKS－LED 时钟源选择位

0：系统时钟

1：指令时钟

Bit<4:3>：LEDPSR<1:0>－LED 时钟分频选择位

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

Bit<2:1>：LEDDRV<2:1>－LED 端口拉电流驱动增强选择位

LEDDRV<2>	LEDDRV<1>	拉电流能力选择
0	1	8mA
0	**0**	**15mA**
1	1	20mA
1	0	25mA

注：默认驱动能力为 15mA

Bit<0>：LEDDRV<0>－LED 端口灌电流驱动增强选择位

0：LED 端口灌电流能力为 30mA

1：LED 端口灌电流能力为 75mA

11.3.2 R1F1/LEDCON1(LED 控制寄存器 1)

0X1F1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON1	-	-	LEDCOM<8>	LEDDTS<8>	LEDPSR<5:2>			
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

Bit<5>：LEDCOM<8>－LEDCOM8 使能位

0：禁止

1：使能

Bit<4>：LEDDTS<8>－LEDCOM8 占空比选择配置位

0：LEDDT0

1：LEDDT1

Bit<3:0>：LEDPSR<5:2>－LED 时钟分频计数值

LEDPSR<5>	LEDPSR<4>	LEDPSR<3>	LEDPSR<2>	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

11.3.3 R1F2/LEDCOM(LED COM 口控制寄存器)

0X1F2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOM	LEDCOM<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>：LEDCOM<7:0>－LEDCOM<7:0>使能位

0：禁止

1：使能

注：自动扫描模式下，LEDCOM<8:0>任意选择，组合不同的点阵，对应 IO 设定为 LED 端口。
手动扫描模式下，LEDCOM<8:0>选择当前扫描的 COM 口，由 LEDDTS<8:0>选择 IO 设定为 LED 端口。

11.3.4 R1F3/LEDDT0(LED 占空比寄存器 0)

0X1F3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDT0	LEDDT0<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>：LEDDT0<7:0>－LED 亮度选择 0

11.3.5 R1F4/LEDDT1(LED 占空比寄存器 1)

0X1F4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDT1	LEDDT1<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>：LEDDT1<7:0>－LED 亮度选择 1

11.3.6 R1F5/LEDDTS(LED DUTY 选择寄存器)

0X1F5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDTS	LEDDTS<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>：LEDDTS<7:0>－LED COM 口占空比选择配置寄存器

0：LED COMx 的占空比使用 LEDDT0 配置寄存器

1：LED COMx 的占空比使用 LEDDT1 配置寄存器

注：手动扫描模式下，LEDDTS<8:0>用来选择 LED 口，对应 P70、P6<7:0>，使能后对应 IO 将作为 LED 专用。

11.3.7 R1F6/LEDDB(LED 数据寄存器)

0X1F6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDB	LEDDB<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>：LCDDB<7:0>－LCD RAM 数据读写位，配合 0x180 LCDRAM 的地址寄存器

LEDRAM	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDRAM0	7	6	5	4	3	2	1	0
LEDRAM1	15	14	13	12	11	10	9	8
LEDRAM2	23	22	21	20	19	18	17	16
LEDRAM3	31	30	29	28	27	26	25	24
LEDRAM4	39	38	37	36	35	34	33	32
LEDRAM5	47	46	45	44	43	42	41	40
LEDRAM6	55	54	53	52	51	50	49	48
LEDRAM7	63	62	61	60	59	58	57	56
LEDRAM8	71	70	69	68	67	66	65	64

12 UART/IIC/SPI 三选一串口

JZ8F8620 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。

12.1 UART 串口接收器/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。JZ8F8620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，TC3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，TC3的溢出信号作为时钟源。

波特率计算：

UART 的时钟来自于 TC3 的 8 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 TC3 的 8 分频

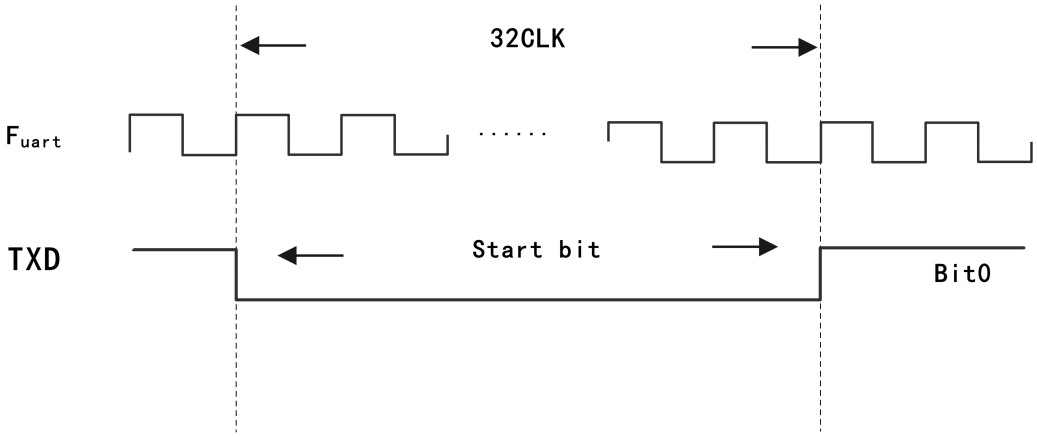
$$TC3 = 1/9600/8 = 13us$$

12.1.1 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）

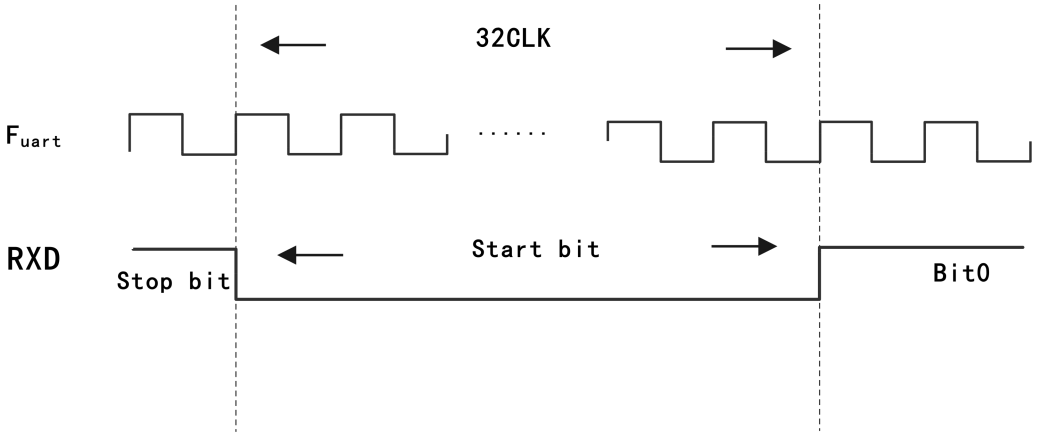


12.1.2 UART 接收

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）

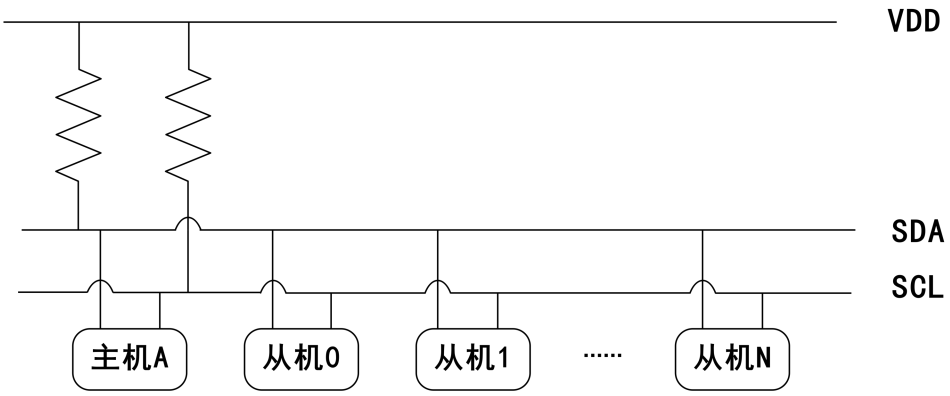


12.2 IIC 串口接收器/发送器

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。

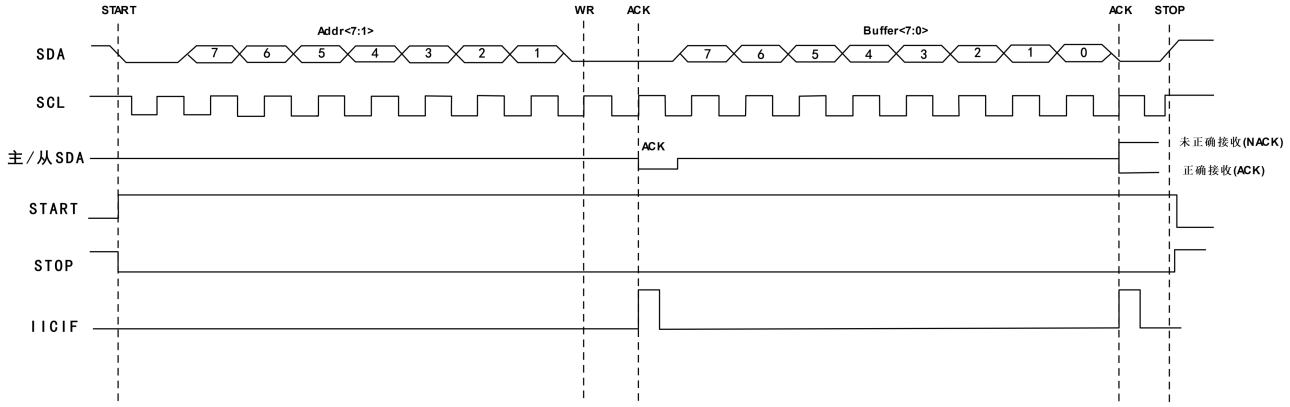
IIC总线规定，当设备处于空闲状态时，时钟和数据两根总线都处于高电平状态。



12.2.1 IIC 操作流程

IIC串口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）。



12.2.2 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。

12.3 SPI 串口接收器/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（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。

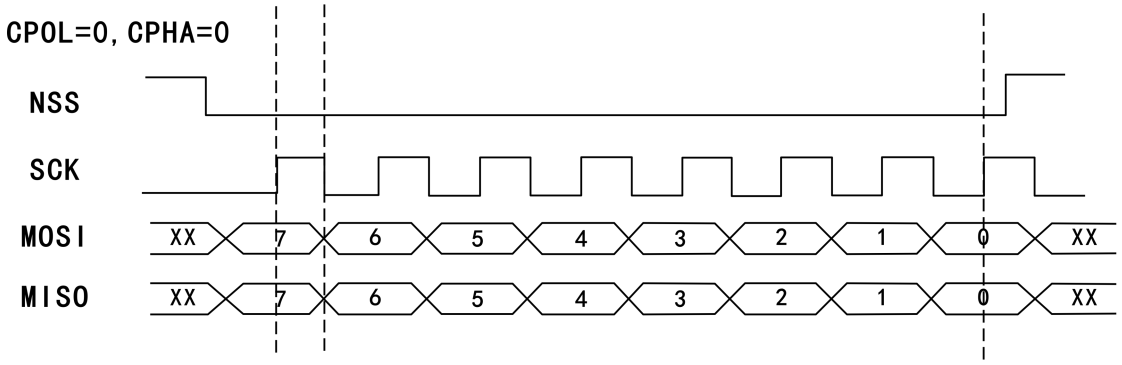
12.3.1 SPI 工作模式

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。

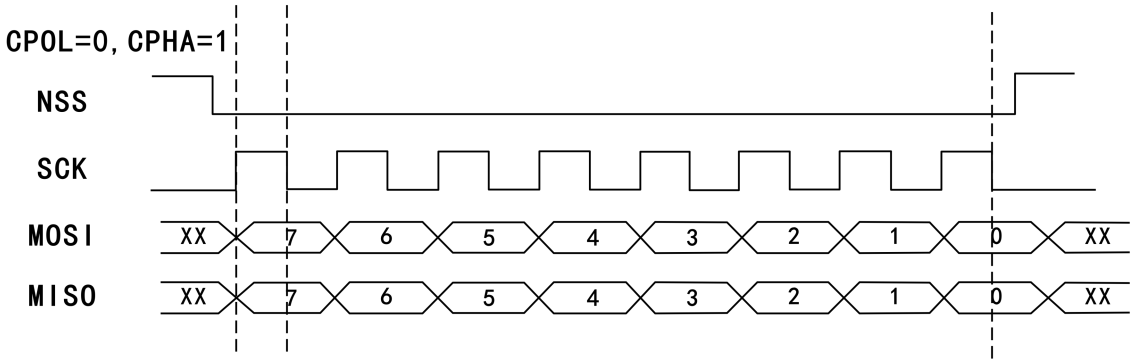
- CPOL=0，SPI总线空闲时，时钟线SCK为低电平。
- CPOL=1，SPI总线空闲时，时钟线SCK为高电平。
- CPHA=0，在SCK第一个跳变沿，主机对MISO引脚电平采样；主机的数据发送则在第二个跳边沿。
- CPHA=1，在SCK第二个跳变沿，主机对MISO引脚电平采样。主机的数据发送则在第一个跳边沿。

实现以下四个模式：

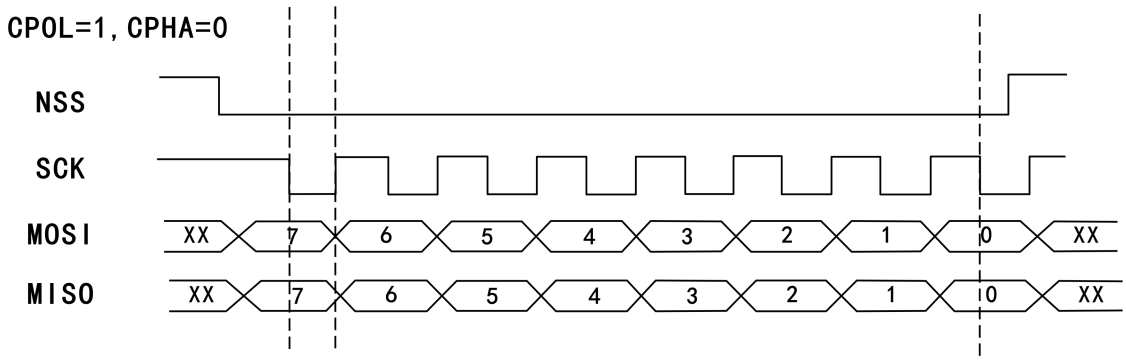
Mode0：CPHA=0、CPOL=0：空闲时 SCK 为低电平。数据在第一个边沿（上升沿）采样，在第二个边沿（下降沿）发送。



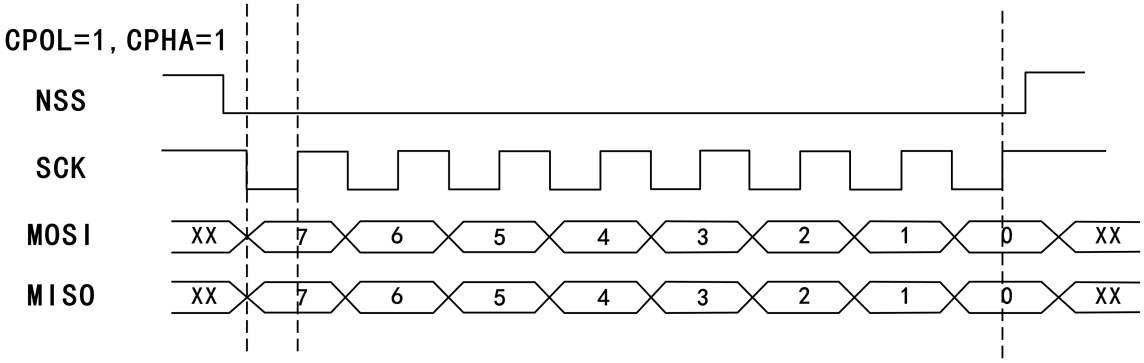
Mode1：CPOL=0、CPHA=1：空闲时 SCK 为低电平。数据在第一个边沿（上升沿）发送，在第二个边沿（下降沿）采样。



Mode2：CPOL=1、CPHA=0：空闲时 SCK 为高电平。数据在第一个边沿（下降沿）采样，在第二个边沿（上升沿）发送。



Mode3：CPOL=1、CPHA=1：空闲时 SCK 为高电平。数据在第一个边沿（下降沿）发送，在第二个边沿（上升沿）采样。



12.3.2 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。
3. 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。
4. 如需发送完成进入中断，则使能RX_SPIIE。
5. 将需要发送的数据存入SSIBUF
6. 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。

12.4 SSI 相关寄存器

12.4.1 R1E3/SSICON0(SSI 控制寄存器 0)

0X1E3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	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：禁止使用

11：TC3 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

12.4.2 R1E4/SSICON1(SSI 控制寄存器 1)

0X1E4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	Bit0
SSICON1	RUN_EN	START_EN	STOP_EN	MSTR	TEN	REN	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 SCK 为低电平。

CPOL=1，SPI 总线空闲时，时钟线 SCK 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 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。

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
P60	TX0	SDA0	MOSI0
P61	RX0	SCL0	SCK0
P62			CS
P63			MISO

1：

通信口	UART	IIC	SPI
P53	TX1	SDA1	MOSI1
P54	RX1	SCL1	SCK1
P62			CS
P63			MISO

12.4.3 R1E5/I2CADDR(I2C 地址寄存器)

0X1E5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	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

12.4.4 R1E6/SSIBUF(SSI 收发数据寄存器)

0X1E6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	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 SSIBUF 写入数据

串行接收时，MCU 从 SSIBUF 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

12.4.5 R1E7/SSISR0(串口状态寄存器 0)

0X1E7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	Bit0
SSISR0	-	BUS_BUSY	MSTR_IDLE	ARB_LOST	DATA_ACK	ADD_ACK	MSTR_ERROR	MSTR_BUSY
读/写	R	R/W	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 清除

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）

12.4.6 R1E8/SSISR1(串口状态寄存器 1)

0X1E8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	BiTC1	Bit0
SSISR1	-	-	TI	RI	RBIT8	SENDFIN	TREQ	RREQ
读/写	R	R	R/W	R/W	R	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0

Bit<5>：TI－UART 发送完成标志位

0：写 0 清 0

1：发送完成

Bit<4>：RI－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

13 EEPROM 以及 ISP 在线烧录操作

13.1 EEPROM 读写操作

JZ8F8620 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 清零将不会影响词写周期，写周期完成时，E2PWR 位将由硬件清零，在下一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。

注：读写 EERPROM 完成后需要将区域还原为 MTPROM，以免程序异常。

13.2 ISP 在线烧录时序

JZ8F8620 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（推荐方式二）：

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 2 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(不推荐)

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：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（强制无上下拉），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。

13.3 相关寄存器

13.3.1 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>－在 EEPROM 读写中用于选择 EEPROM 地址。

13.3.2 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 ROM 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（操作 EEPROM 需置 1）

0：MTP ROM（谨慎操作）

1：EEPROM（读写 EEPROM 完成之后要还原为 MTPROM）

Bit<0>：E2PWREN－EEPROM 写使能位

0：禁止

1：使能

13.3.3 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

13.3.4 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

14 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
IRC 频率	4M	IRC 频率选择 4M
	8M	IRC 频率选择 8M
	16M	IRC 频率选择 16M
	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=1.7V	低压复位点选择 1.7V
	LVR=1.9V	低压复位点选择 1.9V
	LVR=2.1V	低压复位点选择 2.1V
	LVR=2.4V	低压复位点选择 2.4V
	LVR=2.8V	低压复位点选择 2.8V
	LVR=3.0V	低压复位点选择 3.0V
	LVR=3.5V	低压复位点选择 3.5V
	LVR=3.9V	低压复位点选择 3.9V
复位时间	PWRT=3.2ms,WDT=6.4ms	复位建立时间=3.2ms,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
P55 端口	RESET	P55 端口作外部复位口
	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
CMP 迟滞	无迟滞	比较器结果无迟滞
	20mV	比较器结果迟滞电压为 20mV
	40mV	比较器结果迟滞电压为 40mV
	60mV	比较器结果迟滞电压为 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
P60-P67，P70 驱动增强控制	允许控制	P60-P67，P70 驱动增强功能 可以软件控制
	禁止	P60-P67，P70 驱动增强功能 禁止

P50-P52，P71 驱动增强控制	允许控制	P50-P52，P71 驱动增强功能 可以软件控制
	禁止	P50-P52，P71 驱动增强功能 禁止
封装脚位	24pin	芯片封装脚位选择 24 脚
	20pin	芯片封装脚位选择 20 脚
	16pin	芯片封装脚位选择 16 脚
	8pin	芯片封装脚位选择 8 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
	仅作烧写口(强制无上下拉),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悬空，程序不可以控制
	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

15 指令集

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

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

16 电气特性

16.1 极限参数

工作温度.....－40℃～85℃
存储温度.....－65℃～150℃
输入电压..... Vss－0.3V～Vdd+0.5V
输出电压..... Vss－0.3V～Vdd+0.5V
工作电压..... 1.8V－5.5V

16.2 直流电气特性

(V_{DD}=5V，工作温度=25℃，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
Fosc(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	
Fcpu	指令时钟	2T	－	－	8	MHz	VDD=3.0V～5.5V
		4T/8T	－	－	4	MHz	VDD=2.4V～5.5V
		16T/32T	－	－	2	MHz	VDD=2.1V～5.5V
T_{PWRT}	上电复位保持时间		复位时间=WDT 溢出时间/2			ms	PWRT=WDT/2
T_{RST}	外部复位脉冲宽度		2	－	－	μs	25℃
T_{UIS}	三合一模块工作电压		2.2	－	－	V	25℃

低电压复位（LVR）

符号	参数说明	最小	典型	最大	单位
I_{LVR}	复位电流（LVR=1.7V）	－	38	－	μA
V_{LVR}	LVR 阈值	-10%	1.7	+10%	V
		-10%	1.9	+10%	
		-10%	2.1	+10%	
		-10%	2.4	+10%	
		-15%	2.8	+15%	
		-15%	3.0	+15%	
		-15%	3.5	+15%	
		-15%	3.9	+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℃
VIH	高电平阈值		0.53*VDD	-	VDD	V	25℃
I_{OH}	源电流	L0（弱驱动口）	2	3	4	mA	25℃，VDD=5V，VOH=4.4V
		L1（除普通口）	7	8	9		
		L2（所有口，默认）	13	15	17		
		L3（除普通口）	18	20	22	mA	
		L4（所有口）	23	25	27		
I_{OL}	灌电流	L0（弱驱动口）	5	6	7	mA	25℃，VDD=5V，VOL=0.6V
		L1（所有口，默认）	28	30	34		
		L2（除普通口）	73	75	77		
R_{PH}	上拉电阻		40	45	50	KΩ	25℃，VDD=5V
R_{PL}	下拉电阻		40	45	50	KΩ	25℃，VDD=5V

弱驱动端口：P50～52

普通口：P53～57、P72～75

高灌电流口：P50～52、P60～67、P70～71

工作电流（IDD）

参数	SYSCK	最小	典型	最大	单位
正常模式（4T）	32 MHz	-	1.57	-	mA
	16 MHz	-	0.83	-	
	8 MHz	-	0.46	-	
	4 MHz	-	0.28	-	
	40 KHz	-	7	-	uA
Sleep 模式（WDT off，LVR on）	-	-	1.5	3	uA
Sleep 模式（WDT on，LVR on）	-	-	5.6	-	
Sleep 模式（WDT off，LVR on，ADC on）	-	-	105	-	
Sleep 模式（WDT off，LVR on，CMP on）	-	-	4.4	-	

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℃，FCPU≤8MHz
VDD - WRIT	MTP	2	-	5.5	V	25℃
	EEPROM	2	-	5.5	V	25℃
Endurance	MTP/EEPROM	-	10000	-	次	25℃
Retention	MTP/EEPROM	-	10	-	年	25℃
T_{WRIT} 快写时间	MTP/EEPROM	-	4.8	-	ms	25℃
T_{WRIT} 慢写时间	MTP/EEPROM	-	7.1	-	ms	25℃
TERASE 擦除时间	MTP/EEPROM	-	1.5	-	ms	25℃
IP/E 擦写电流	MTP/EEPROM	-	0.8	-	mA	25℃，VDD=5.5V

16.3 AD 转换特性

(V_{DD} =5V，V_{SS}=0V，工作温度=25℃)

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

注：（1）误差由评估得出，不在生产中测试

（2）不建议 VBG 作为 AD 基准

16.4 比较器特性

(V_{DD} =5V，V_{SS}=0V，工作温度=25℃)

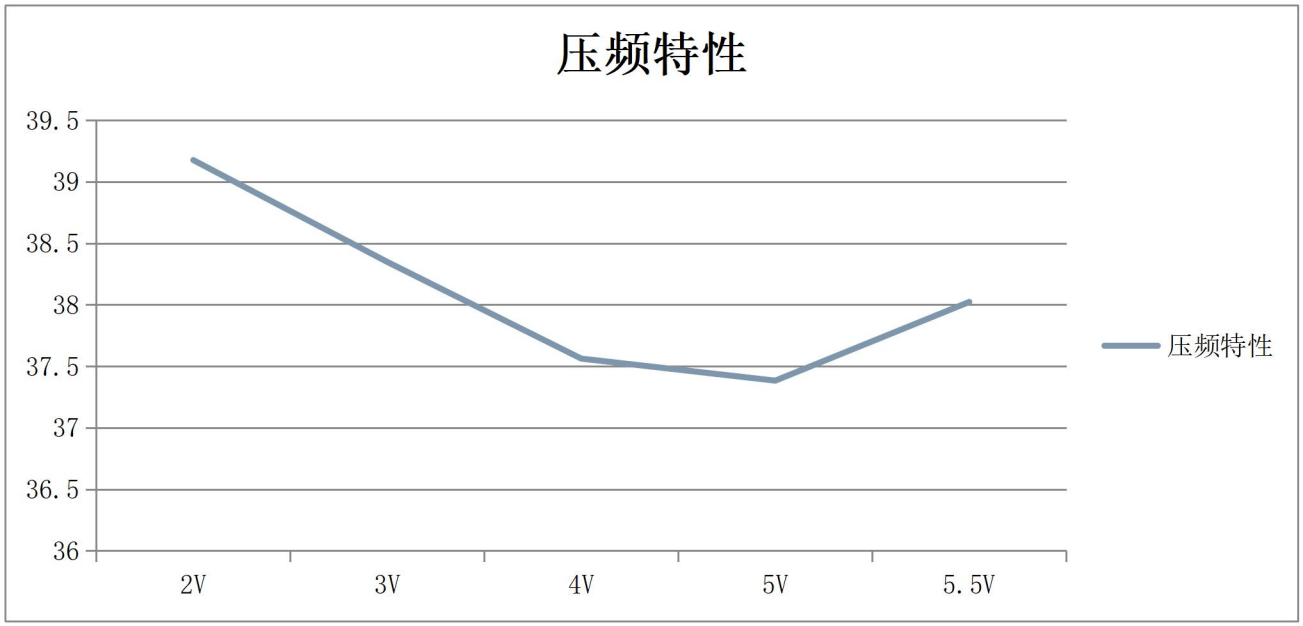
符号	参数	最小	典型	最大	单位	条件
V_{DD}	工作电压范围	1.8	-	5.5	V	16M/4T
Vos	比较器迟滞电压	-	20	-	mV	16M/4T
		-	40	-	mV	
		-	60	-	mV	
Tr	比较器响应时间	-	800	-	ns	16M/4T
		-	1.3	-	us	不开数字滤波
		-	50	-	us	16M/4T

16.5 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。

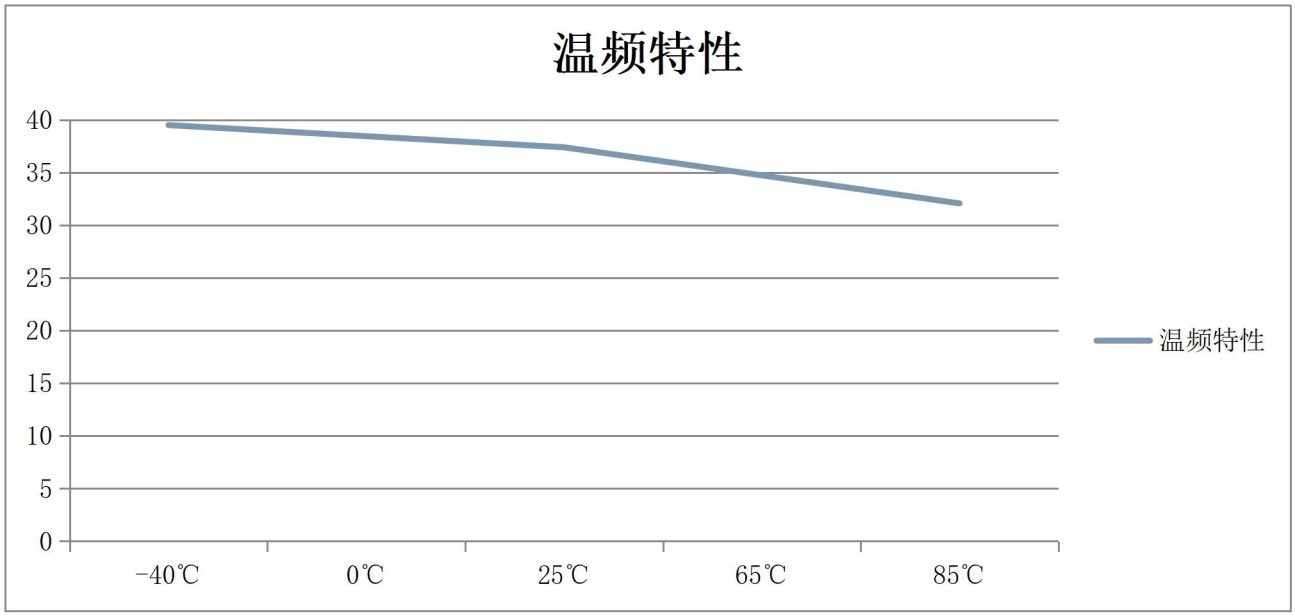
16.5.1 内部低速振荡器 – 压频特性曲线

工作温度在 25℃条件下：（单位 KHz）



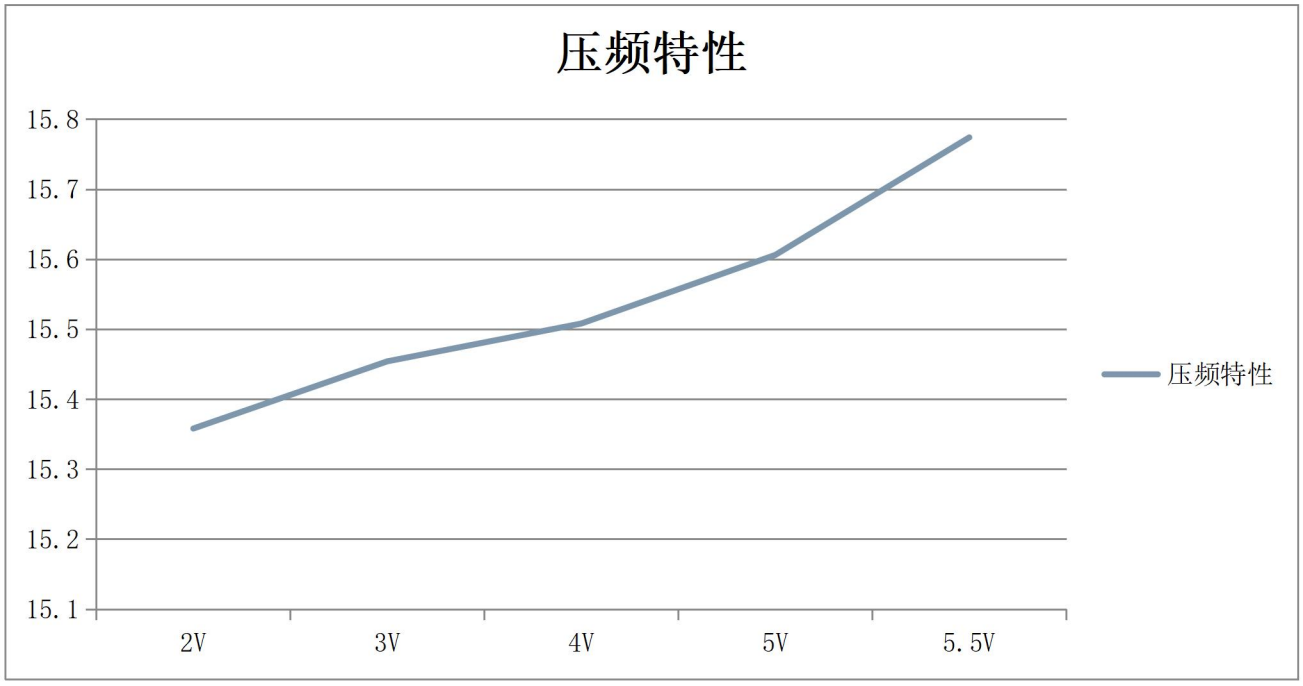
16.5.2 内部低速振荡器 – 温频特性曲线

工作电压在 5V 条件下：（单位 KHz）



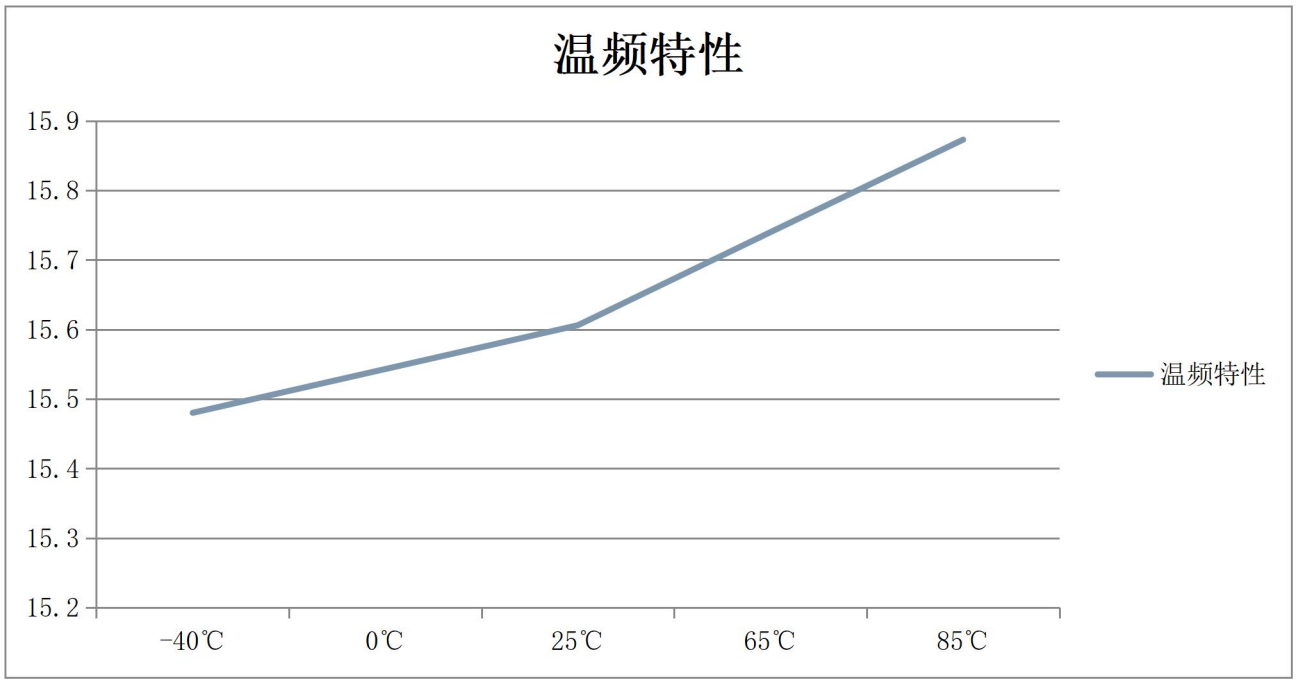
16.5.3 内部 16Mhz RC 振荡器 - 压频特性曲线

工作温度在 25℃条件下：(单位 MHz)



16.5.4 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)



16.6 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。

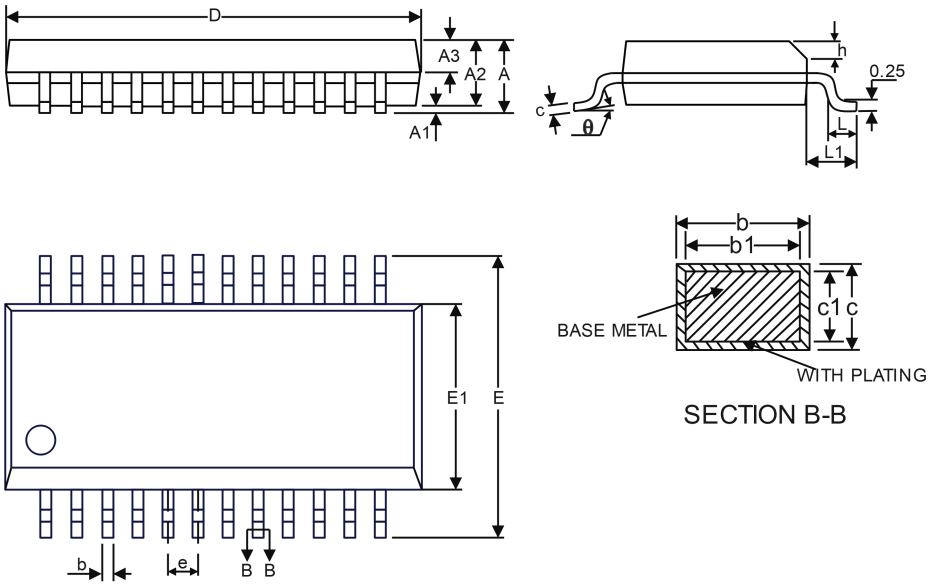
IHRCCAL 修调值（十六进制）	频率（MHz）	修调偏差	IHRCCAL 修调值（十六进制）	频率（MHz）	修调偏差
00	7.8685	-51.27%	40	10.5419	-34.71%
01	7.8989	-51.08%	41	10.6021	-34.34%
02	7.9306	-50.88%	42	10.6584	-33.99%
03	7.9615	-50.69%	43	10.7153	-33.63%
04	7.9946	-50.48%	44	10.7758	-33.26%
05	8.0244	-50.30%	45	10.8314	-32.91%
06	8.0567	-50.10%	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%

17 封装尺寸

17.1 24PIN 封装尺寸

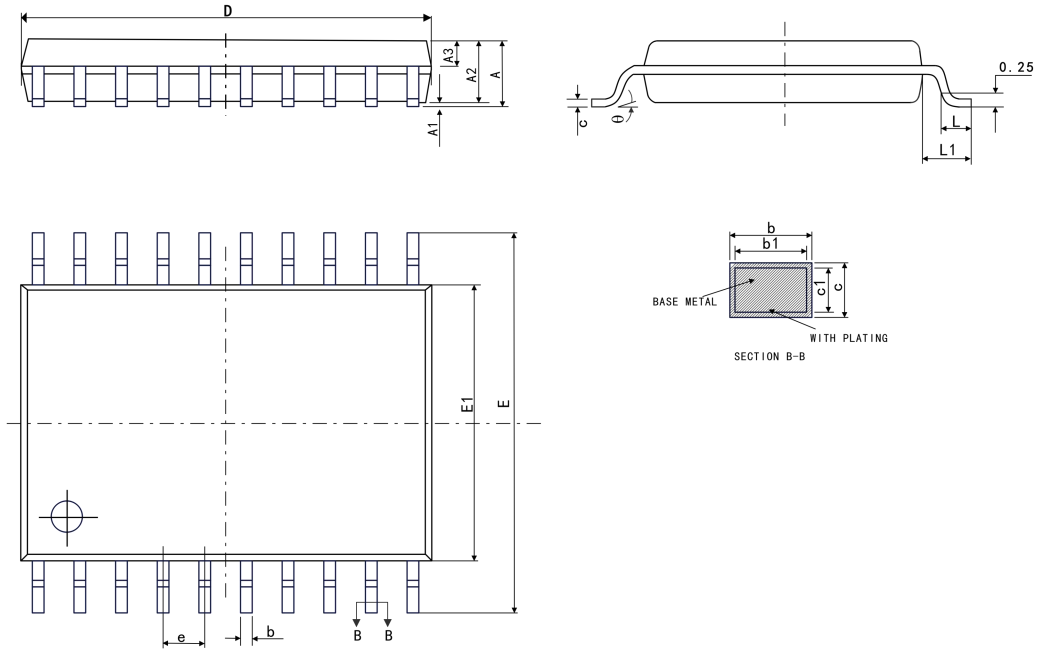
（单位：mm）



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		
θ	0°	——	8°

SSOP24 封装尺寸

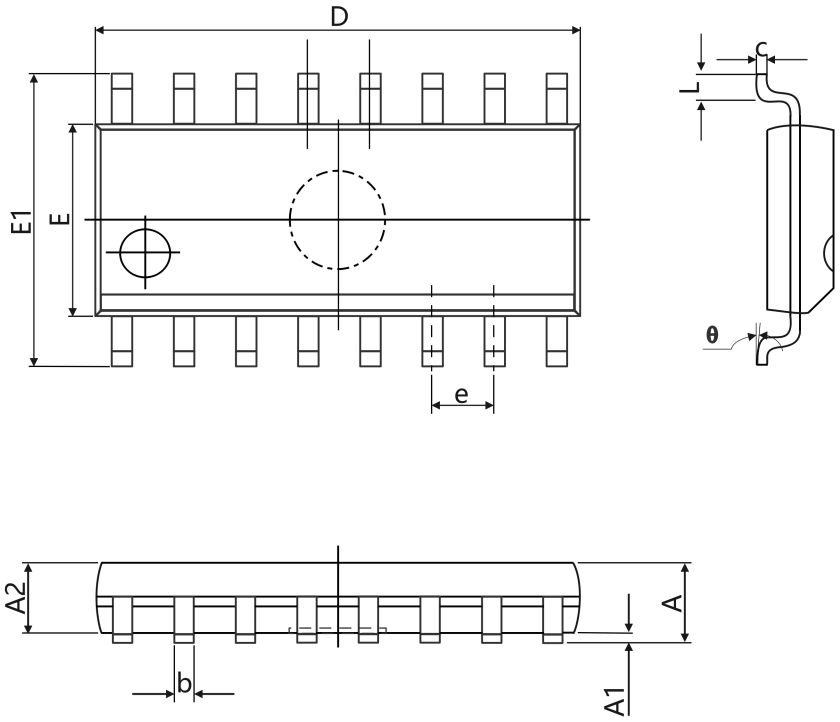
17.2 20PIN 封装尺寸



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

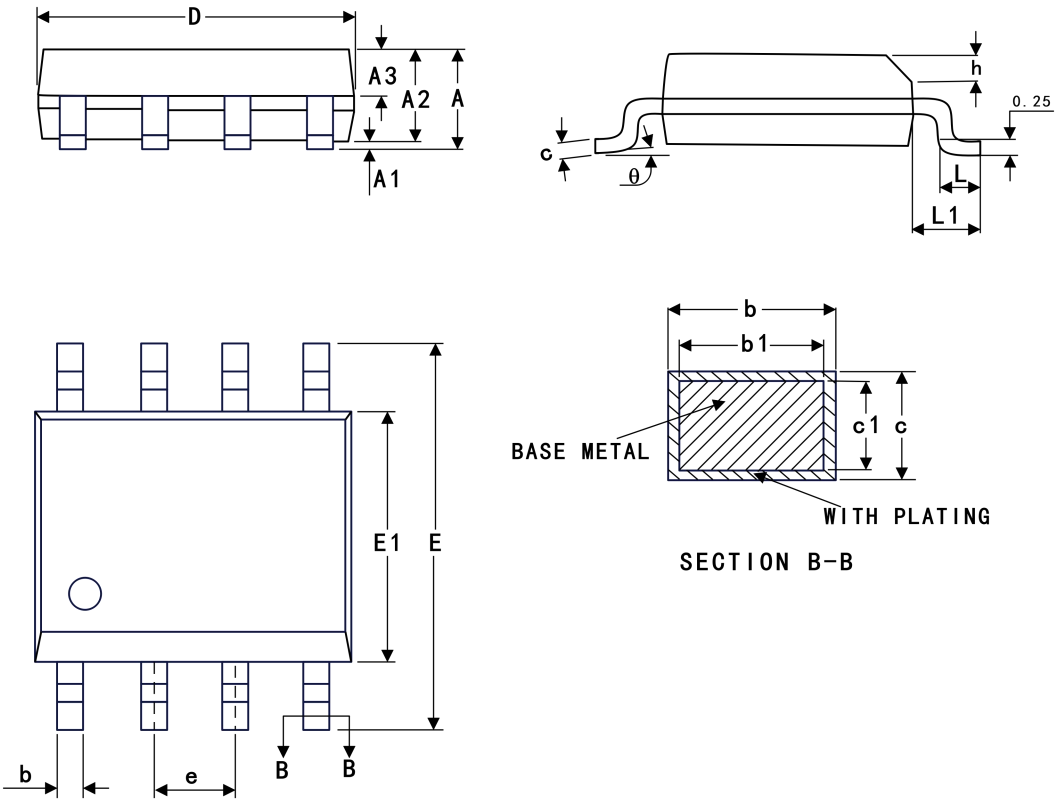
17.3 16PIN 封装尺寸



Symbol	Dimensions In Millimeter		Dimensions In Lnches	
	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

17.4 8PIN 封装尺寸



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	_	_	1.75
A1	0.10	_	0.23
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	_	0.47
b1	0.38	0.41	0.44
c	0.20	_	0.24
c1	0.19	0.20	0.21
D	4.80	4.90	5.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	_	0.50
L	0.50	_	0.80
L1	1.05REF		
θ	0	-	8°

SOP8 封装尺寸