



JSC69F007/8/9

带 24 位高精度 ADC 微控制器

用户数据手册 V1.9

目录

JSC69F007/8/9	6
1 产品概述.....	6
2 应用领域.....	0
3 功能框图.....	0
4 封装类型.....	1
4.1 JSC69F009(LQFP100)	1
4.2 JSC69F008(LQFP80)	2
4.3 JSC69F007(LQFP64)	3
4.4 引脚说明.....	4
5 CPU	9
5.1 存储器结构.....	9
5.2 直接寻址 IRAM 和 SFR	9
5.3 间接寻址 IRAM.....	10
5.4 外部 RAM(XRAM).....	10
5.5 程序存储器(IROM)	10
5.6 程序存储器安全.....	11
5.7 数据指针.....	11
5.8 堆栈.....	11
5.9 特殊功能寄存器 (SFR).....	12
5.10 系统控制寄存器.....	13
6 复位和上电控制.....	17
6.1 复位配置和上电控制.....	17
6.2 上电流程.....	18
6.3 LVD 复位.....	18
6.4 看门狗复位.....	18
6.5 外部引脚复位.....	18
6.6 软件复位.....	18
6.7 复位和上电控制寄存器.....	19
7 系统时钟和电源管理.....	20
7.1 系统时钟.....	20
7.2 Noise Filter	20
7.3 电源管理.....	21
7.4 系统时钟和电源管理寄存器.....	22
8 中断.....	25
8.1 中断操作.....	25
8.2 中断优先级.....	25
8.3 中断寄存器.....	27
9 MDU	33
9.1 乘法 (16 位 x16 位)	33
9.2 除法 (32 位/16 位和 16 位/16 位)	33
9.3 移位和规范化.....	34
9.4 错误标志.....	34

9.5 溢出标志 (MDOV)	34
9.6 MDU 寄存器	35
10GPIO	36
10.1 输入输出控制	36
10.2 输入数据和输出数据	39
10.3 内置上拉寄存器	40
10.4 与 LCD 共用的引脚	41
11 外部中断	42
11.1 外部中断寄存器	42
12 TCx16 位定时器/计数器	43
12.1 定时器计数器框图和时钟源选择	43
12.2 TC0/TC1/TC2 定时器操作	44
12.3 TC0/TC1/TC2 计数功能和寄存器	45
12.4 TC0/TC1/TC2 自动重载功能和寄存器	46
12.5 TC0/TC1/TC2 的 PWM 功能	47
12.6 TC0/TC1/TC2 事件计数器功能	48
12.7 TC0、TC1 和 TC2 寄存器	49
13 T0 定时器	56
13.1 T0 操作	56
13.2 T0 模式寄存器	58
13.3 T0C 计数寄存器	59
13.4 其他相关寄存器	59
14 Buzzer 功能	61
15 LCD 驱动	62
15.1 LCD RAM 位置	65
15.2 LCD 控制寄存器	66
16 ADC	69
16.1 ADC 操作设置	70
16.2 ADC 输入通道和 PGIA	70
16.3 模拟输入缓存器	71
16.4 ADC 参考电压	71
16.5 ADC 增益和 ADC 输出码	72
16.6 ADC 中断控制	72
16.7 ADC 转换速率	72
16.8 ADC 控制寄存器	74
16.9 温度传感器(TS)	79
16.10 ADC 应用注意事项	80
17 升压稳压器 CPR	81
17.1 Charge Pump 寄存器	82
17.2 Charge Pump Regulator 设置	83
18 比较器	84
18.1 比较器控制寄存器	85
19 OPA	86
19.1 OP-Amp 控制寄存器	87

20UART.....	88
20.1 概述.....	88
20.2 UART 模式.....	88
20.2.1 UART 模式 0: 同步 8 位收发器.....	88
20.2.2 UART 模式 1: 异步 8 位收发器.....	89
20.2.3 UART 模式 2/3: 异步 9 位收发器.....	90
20.3 UART 波特率.....	90
20.4 UART 寄存器.....	92
21SPI.....	95
21.1 SPI 概述.....	95
21.2 SPI 端口配置.....	95
21.3 SPI 硬件描述.....	96
21.4 SPI 时钟分频.....	97
21.5 SPI 主控模式.....	97
21.5.1 写冲突.....	97
21.6 SPI 从模式.....	98
21.6.1 写冲突.....	98
21.7 SPI 时钟控制逻辑.....	99
21.7.1 SPI 时钟相位与极性控制.....	99
21.7.2 SPI 传输格式.....	99
21.7.3 CPHA=0 传输格式.....	99
21.7.4 CPHA=1 传输格式.....	100
21.8 SPI 数据传输.....	101
21.8.1 SPI 传输启动.....	101
21.8.2 SPI 传输结束.....	101
21.9 SPI 时序图.....	101
21.9.1 主控模式传输.....	101
21.9.2 从动模式传输.....	102
21.10 SPI 中断.....	102
21.11 SPI 寄存器.....	103
22I2C.....	106
22.1 I2C 概述.....	106
22.2 I2C 端口配置.....	106
22.3 I2C 总线传输方式.....	107
22.3.1 开始和停止条件.....	107
22.3.2 地址数据格式.....	108
22.3.3 应答 ACK.....	108
22.3.4 I2C 时钟时序.....	109
22.4 I2C 主控模式.....	110
22.4.1 I2C 主控模式时钟频率.....	110
22.4.2 I ² C 主控模式控制与状态.....	110
22.5 I2C 从动模式.....	112
22.6 I2C 中断.....	114
22.7 仲裁和同步模式.....	114

22.8 I2C 状态码和应用程序实例.....	115
22.8.1 状态码.....	115
22.8.2 I2C 程序示例.....	116
22.9 I2C 寄存器.....	119
23 在线编程.....	122
23.1 扇区擦除.....	122
23.2 编程.....	122
23.3 在线编程寄存器.....	123
24 系统配置 OPTION.....	125
24.1 OPTION 信息.....	125
24.2 程序加密.....	126
24.3 晶振电容选择.....	127
24.4 CPU 时钟源.....	128
24.5 Noise Filter.....	129
24.6 看门狗.....	130
24.6.1 看门狗模式选择.....	130
24.6.2 看门狗溢出周期.....	131
24.7 Bandgap chop 相位选择.....	132
24.8 Bandgap Chop 时钟.....	133
25 电气参数.....	134
25.1 极限参数.....	134
25.2 系统操作特性.....	135
25.3 GPIO 特性.....	136
25.4 ADC 特性.....	136
25.5 Charge Pump Regulator 特性.....	139
25.6 OP-Amp 电气特性.....	140
25.7 比较器电气特性.....	140
25.8 LCD 电气特性.....	140
25.9 Flash 存储器特性.....	141
25.10 特殊条件或环境中使用的建议.....	141
26 系统应用.....	142
27 指令集.....	145
27.1 符号说明.....	145
27.2 运算指令.....	146
27.3 逻辑指令.....	147
27.4 数据传输指令.....	148
27.5 布尔运算指令.....	149
27.6 程序跳转指令.....	149
28 JSC69F007/8/9 Start-Kit.....	150
29 模拟设置与应用.....	150
30 订购信息.....	151
30.1 命名规则.....	151
30.2 LQFP100 PIN.....	152
30.3 LQFP80 PIN.....	153

30.4 LQFP64 PIN	154
31 版本说明	155

JSC69F007/8/9

1 产品概述

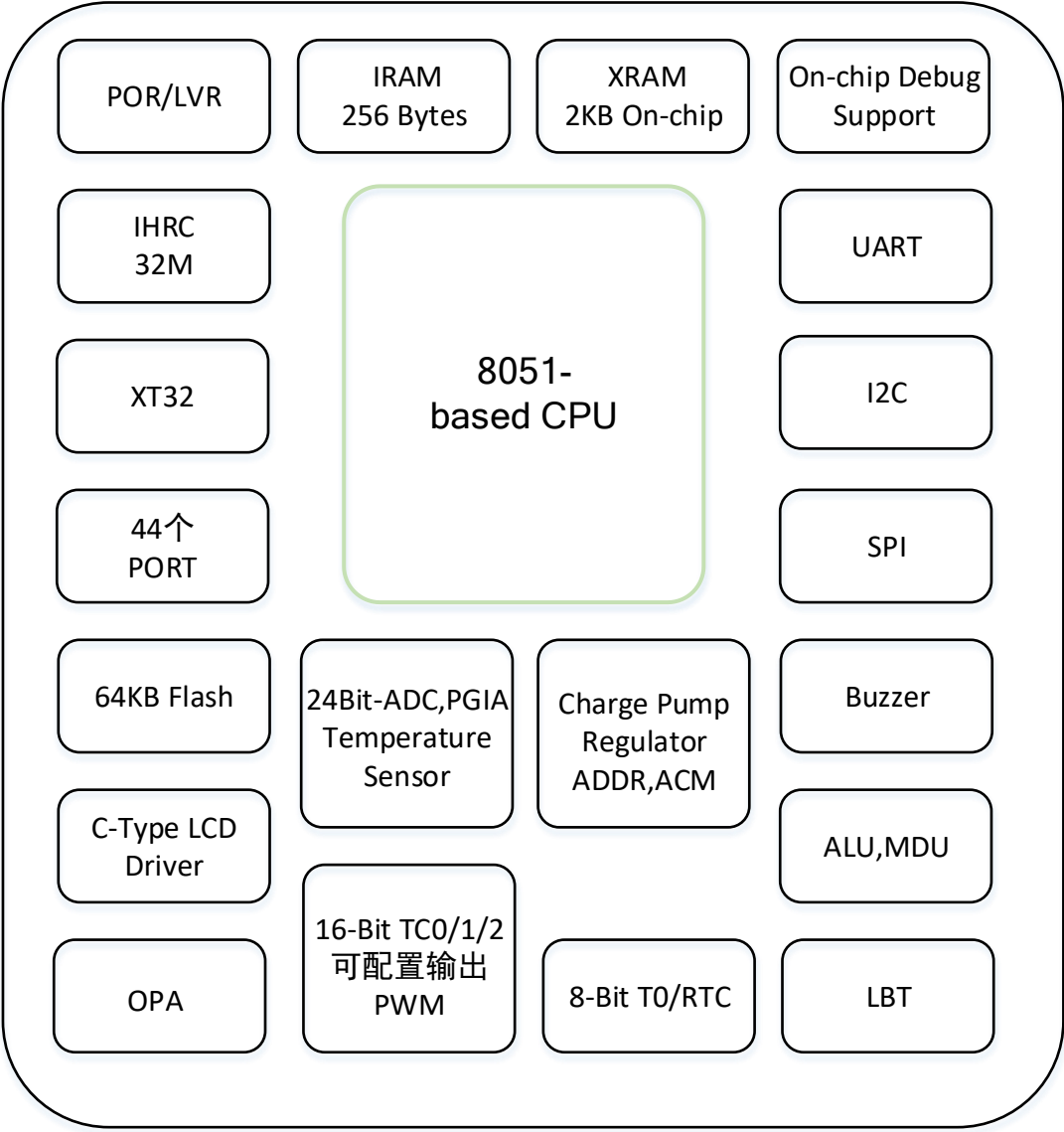
- ◆ **增强型 8051, 减少指令周期时间(最高达到 80C51 的 12 倍)**
 - 最高达 32MHz 可调的 CPU 频率
 - 内部 32MHz 时钟发生器(IHRC)
 - 实时时钟, 带 32.768KHz 晶振
 - Fcpu: 16MIPs~0.25MIPs
- ◆ **存储器**
 - 64KB 非易失性 Flash 存储器(IROM)支持总线编程功能
 - 256 字节内部 RAM(IRAM)
 - 2KB 外部(XRAM)
- ◆ **I/O 引脚**
 - 20 个纯 I/O 引脚, 24 个与 LCD 功能共用的 I/O 引脚
- ◆ **中断: 10 个中断源, 可控制其优先权, 且对应唯一的中断向量**
 - 8 个内部中断: T0、TC0、TC1、TC2、ADC、UART、I2C 和 SPI
 - 2 个外部中断: INT0、INT1
- ◆ **硬件乘法/除法单元和 2 组 DPTR**
- ◆ **定时器系统**
 - 1 组 8 位基本定时器 T0, 带 RTC
 - 3 组 16 位定时器/计数器(TC0~TC2), 带自动重载、PWM、蜂鸣器输出
 - 1 个蜂鸣器(引脚 P15): 1KHz/2KHz/4KHz/8KHz
- ◆ **通信模块 SPI/I2C/UART 接口**
- ◆ **比较器: 低电压检测 2.2V~3.6V(每阶 0.2V)**
- ◆ **电源: 2.0V~5.5V(AVDD/DVDD)**
- ◆ **升压稳压器**
 - 输入电压: 2.0V~5.5V
 - AVDDR 稳压输出为模拟电路供电 (ADC/PGIA/OP...)
 - AVDDR 电压选项: 2.7V/3.0V/3.3V/3.6V
 - ACM 1V 为 ADC 通用电压
- ◆ **PGIA**
 - 可编程控制增益放大器, 高输入阻抗
 - 增益: 1x/4x/8x/16x/32x/64x/128x
- ◆ **24 位 Delta Sigma ADC**
 - 单一或差分通道配置
 - 9 个外部输入通道: AI1~AI9
 - 4 个内部通道: 1/8*VDD 检测、温度传感器、ACM 和 GND
 - 内部或者外部参考电压
 - 转换输出速率: 2Hz~7.8KHz
- ◆ **运算放大器 OPA: 1 组运算放大器**
- ◆ **LCD 驱动: C-Type LCD 驱动, 高达 252 点**
 - 4COM 或 6COM, 1/3 偏压
 - 4COM*44SEG 或者 6COM*42SEG, LCD 引脚和 IO 引脚共用
 - 可调节的 VLCD 输出 2.8V~4.5V
- ◆ **片上调试**
 - 曼彻斯特编码的单线调试接口
 - 2 个硬件断点
 - 可配软件断点
 - Flash 数据安全保护
 - 可选外部供电和仿真器供电

2 应用领域

- 血压计
 - 血糖仪
 - 其他测量应用
- 温度计

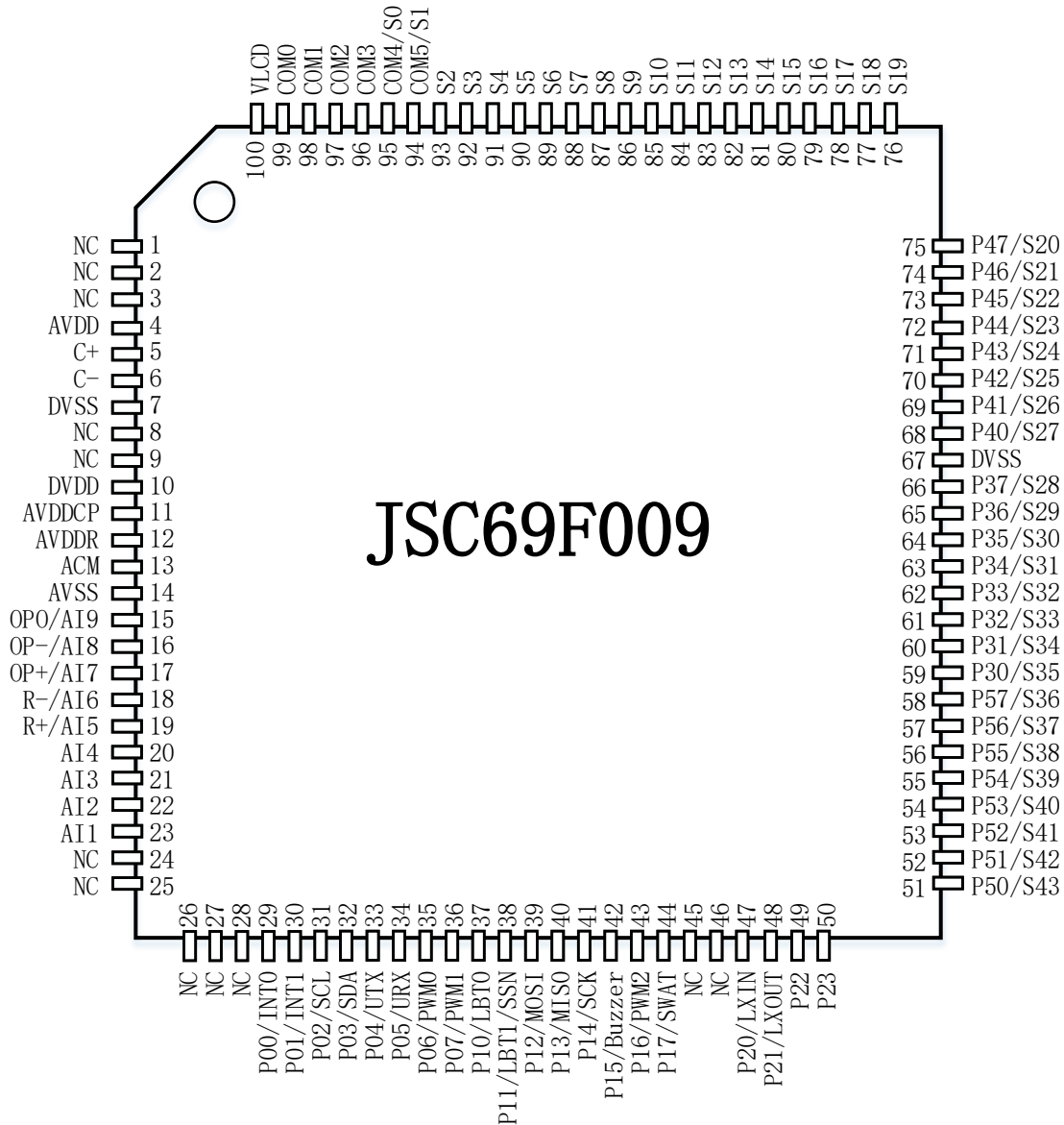
体重秤

3 功能框图

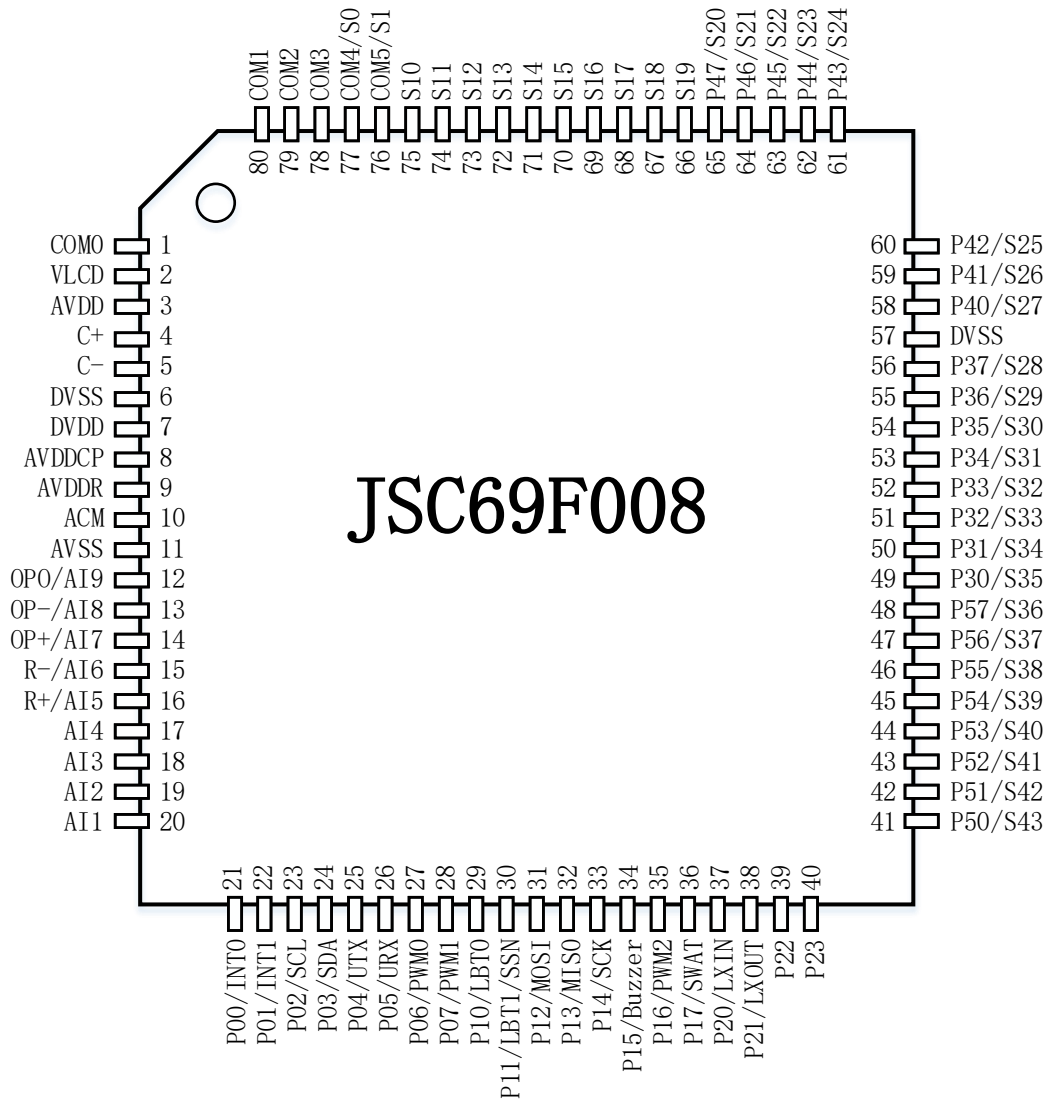


4 封装类型

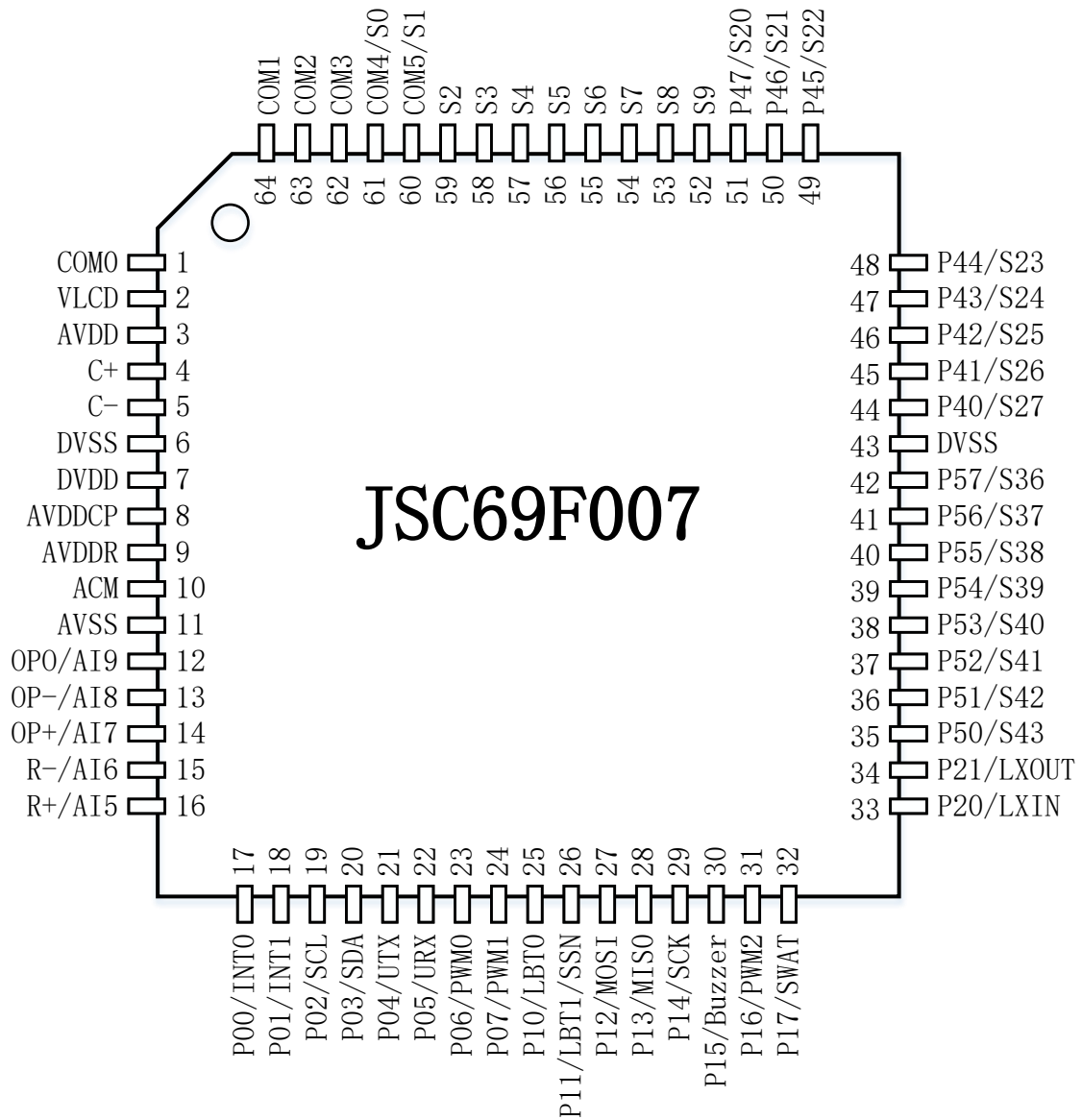
4.1 JSC69F009(LQFP100)



4.2 JSC69F008(LQFP80)



4.3 JSC69F007(LQFP64)



4.4 引脚说明

电源引脚

引脚名称	类型	功能说明
DVDD	Power	数字电源(2.2V~5.5V)
DVSS	Power	数字地(0V)
AVDD	Power	模拟电源(2.2V~5.5V)
AVSS	Power	模拟地(0V)

P0

引脚名称	类型	功能说明
P0.0	Digital I/O	GPIO: 双向输入输出引脚
INT0	Digital I/O	INT0: 中断输入引脚
TC0CLK	Digital Input	TC0: 外部信号输入源
P0.1	Digital I/O	GPIO: 双向输入输出引脚
INT1	Digital I/O	INT1: 中断输入引脚
TC1CLK	Digital Input	TC1: 外部信号输入源
P0.2	Digital I/O	GPIO: 双向输入输出引脚
SCL	Digital I/O	I2C: 时钟输出(主机)时钟输入(从机)
TC2CLK	Digital Input	TC2: 外部信号输入源
P0.3	Digital I/O	GPIO: 双向输入输出引脚
SDA	Digital I/O	I2C: 数据引脚
P0.4	Digital I/O	GPIO: 双向输入输出引脚
UTX	Digital Output	UART: 数据输出引脚
P0.5	Digital I/O	GPIO: 双向输入输出引脚
URX	Digital Output	UART: 数据输入引脚
P0.6	Digital I/O	GPIO: 双向输入输出引脚
PWM0	Digital Output	PWM0: 输出引脚
P0.7	Digital I/O	GPIO: 双向输入输出引脚
PWM1	Digital Output	PWM1: 输出引脚

P1

引脚名称	类型	功能说明
P1.0 LBT0	Digital I/O Analog Input	GPIO: 双向输入输出引脚 比较器外部输入引脚
P1.1 LBT1 SSN	Digital I/O Analog Input Digital Input	GPIO: 双向输入输出引脚 比较器外部输入引脚 SPI: 从模式选择引脚
P1.2 MOSI	Digital I/O Digital I/O	GPIO: 双向输入输出引脚 SPI: 发送引脚 (主机) 接收引脚 (从机)
P1.3 MISO	Digital I/O Digital I/O	GPIO: 双向输入输出引脚 SPI: 接收引脚 (主机) 发送引脚
P1.4 SCK	Digital I/O Digital I/O	GPIO: 双向输入输出引脚 SPI: 时钟输出 (主机) 时钟输入 (从机)
P1.5 Buzzer	Digital I/O Digital Output	GPIO: 双向输入输出引脚 Buzzer 输出引脚
P1.6 PWM2	Digital I/O Digital Output	GPIO: 双向输入输出引脚 PWM2 输出引脚
P1.7 SWAT	Digital I/O Digital I/O	GPIO: 双向输入输出引脚 单线仿真接口 (*注: P1.7 口不能接任何器件@仿真模式)

P2

引脚名称	类型	功能说明
P2.0 LXIN	Digital I/O Analog Input	GPIO: 双向输入输出引脚 系统时钟: 外部时钟输入引脚
P2.1 LXOUT	Digital I/O Analog Output	GPIO: 双向输入输出引脚 系统时钟: 驱动外部晶振
P2.2	Digital I/O	GPIO: 双向输入输出引脚
P2.3	Digital I/O	GPIO: 双向输入输出引脚

P3

引脚名称	类型	功能说明
P3.0 SEG35	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.1 SEG34	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.2 SEG33	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.3 SEG32	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.4 SEG31	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.5 SEG30	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.6 SEG29	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P3.7 SEG28	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚

P4

引脚名称	类型	功能说明
P4.0 SEG27	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.1 SEG26	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.2 SEG25	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.3 SEG24	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.4 SEG23	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.5 SEG22	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.6 SEG21	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P4.7 SEG20	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚

P5

引脚名称	类型	功能说明
P5.0 SEG43	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.1 SEG42	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.2 SEG41	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.3 SEG40	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.4 SEG39	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.5 SEG38	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.6 SEG37	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚
P5.7 SEG36	Digital I/O LCD Output	GPIO: 双向输入输出引脚 LCD: LCD 驱动 Segment 引脚

LCD

引脚名称	类型	功能说明
VLCD	LCD Output	LCD: 输出电压
COM0	LCD Output	LCD: LCD 驱动 Common 引脚
COM1	LCD Output	LCD: LCD 驱动 Common 引脚
COM2	LCD Output	LCD: LCD 驱动 Common 引脚
COM3	LCD Output	LCD: LCD 驱动 Common 引脚
COM4	LCD Output	LCD: LCD 驱动 Common 引脚
SEG0	LCD Output	LCD: LCD 驱动 Segment 引脚
COM5 SEG1	LCD Output LCD Output	LCD: LCD 驱动 Common 引脚 LCD: LCD 驱动 Segment 引脚
SEG2~SEG19	LCD Output	LCD: LCD 驱动 Segment 引脚

模拟脚

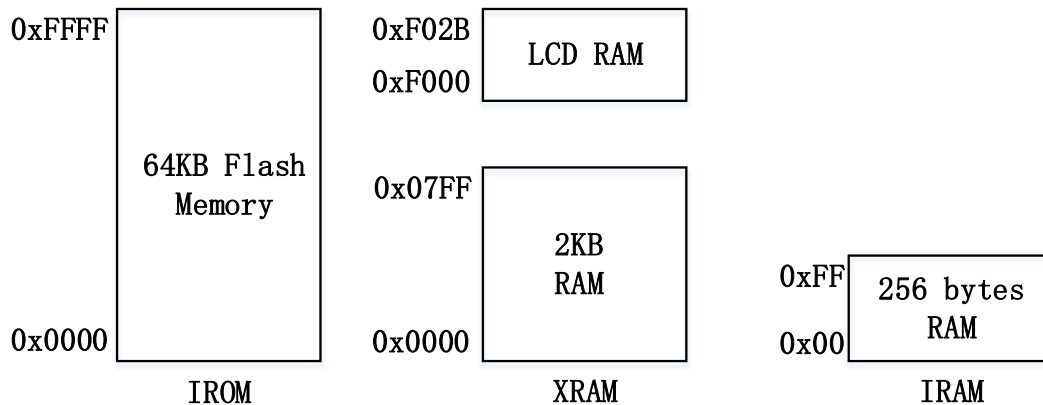
引脚名称	类型	功能说明
AVDD	Power	模拟电压输入端
AVSS	Power	模拟地
C+、C-	-	Charge pump 电容连接引脚 使能 Charge pump 时连接 1uF 电容
AVDDCP	Pump Output	Charge Pump 输出引脚 使能 Charge pump 时连接 2.2uF 电容到 DVSS
AVDDR	Analog Output	Regulator 输出模拟端，连接 1uF 电容到 AVSS。 可选项: 2.7V/3.0V/3.3V/3.6V
ACM	Analog Output	ACM 电压输出 1V 作为 ADC 通用电压（仅在灌电流时）连接 0.1uF 电容到 AVDDR
AI1	Analog Input	ADC: 输入通道 1
AI2	Analog Input	ADC: 输入通道 2
AI3	Analog Input	ADC: 输入通道 3
AI4	Analog Input	ADC: 输入通道 4
AI5	Analog Input	ADC: 输入通道 5
R+	Analog Input	ADC: 外部参考电压正极输入引脚
AI6	Analog Input	ADC: 输入通道 6
R-	Analog Input	ADC: 外部参考电压负极输入引脚
AI7	Analog Input	ADC: 输入通道 7
OP+	Analog Input	OPA: 正极输入引脚
AI8	Analog Input	ADC: 输入通道 8
OP-	Analog Input	OPA: 负极输入引脚
AI9	Analog Input	ADC: 输入通道 9
OPO	Analog Output	OPA: 输出引脚

5 CPU

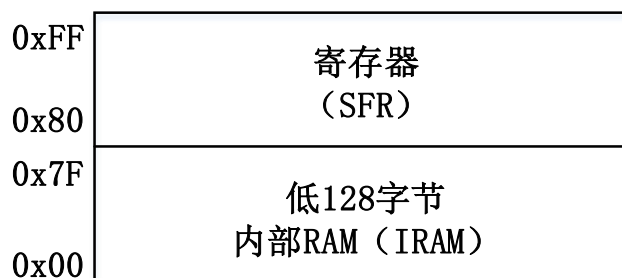
JSC69F007/8/9 系列是一颗增强型的 8051 微控制器，且完全兼容 MCS-51 指令集，因此可以使用目前流行的编译环境仿真（例如 Keil C51）。总的来说，在相同的频率下，JSC69F007/8/9 的 CPU 要比原始的 8051 快 9.4 到 12.1 倍。

5.1 存储器结构

JSC69F007/8/9 内建了三个存储器：内部 RAM(IRAM)，外部 RAM(XRAM)，以及程序存储器 (IROM)。内部 RAM 由 256 个字节组成，具有较高的存取性能（支持直接寻址和间接寻址）。相比之下，外部 RAM 有 2KB 大小，但需要更长的存取周期。程序存储器是一个 64KB 的 Flash 存储器，最快的存取速度可达 16MHz。



5.2 直接寻址 IRAM 和 SFR



直接寻址指令（如 MOV A, direct）可以访问低 128 字节的内部 RAM（地址范围：00-7FH）和所有系统寄存器（SFR，地址范围：0x80-0xFF）。

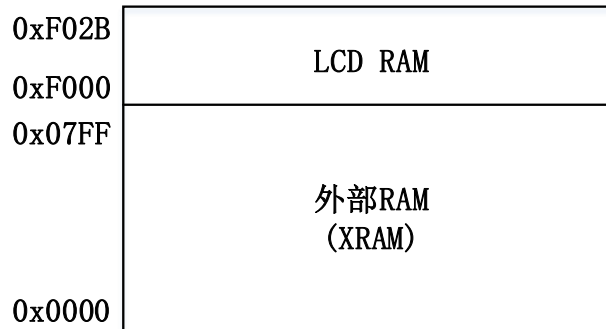
除此之外，内部 RAM 的最低 32 字节（0x00-0x1F）可以看作是 4 组 R0-R7 工作寄存器，这 4 组工作寄存器可通过汇编指令（如 MOV A, R0）进行寻址。内部 RAM 的 0x20-0x2F 和每个 SFR 的结束段 0x0/0x8 都是位可寻址。

5.3 间接寻址 IRAM



虽然直接寻址指令可访问的内部 RAM 的周期比间接寻址的要少，第二种寻址类型可以访问内部 RAM 的所有区域，且是唯一可以访问内部 RAM 的高 128 字节（0x80-0xFF）的寻址方式。

5.4 外部 RAM(XRAM)



外部 RAM 增加了变量的容量，但跟内部 RAM 相比，它的访问能力是最低的，这是因为频繁地使用变量和本地变量是为了存入内部 RAM，而外部 RAM 的绝大多数用法都是特定的。外部 RAM 可以作为优先级别较低的，或者在 ROM 中预装查表以加速访问周期的变量存储区域。LCD RAM 也位于外部 RAM 中。

5.5 程序存储器(IROM)



程序存储器是 Flash 存储器，可以存储程序代码，ROM 查表数据以及其它的临时修改数据。可通过调试工具如 JSC-Link Adapter 进行升级，也可以通过在线程序处理来自行更新。

5.6 程序存储器安全

JSC69F007/8/9 内置 ROM 加密机制, 防止 Flash ROM 资料被破解。当使能加密功能时, 就无法读出 ROM 里面的内容, 所有的 ROM 地址都只能读到 0x00 的数据。

5.7 数据指针

在执行 MOVX 和 MOVC 指令时, 数据指针可帮助指定 XRAM 和 IROM 地址。该单片机有 2 组数据指针 (DPH/DPL 和 DPH1/DPL1), 可以通过 DPS 寄存器进行选择。DPC 寄存器控制 2 个功能: 选择下一个数据指针和自动加减数据指针功能。

选择下一个数据指针的功能是在执行 MOVX @DPTR 指令后, 可将指针指向所想要的地址。换句话说, DPS 可在 2 个数据指针之间自动切换。启用该方法: 首先写入 0 到 DPSEL, 填 1 到 NDPS; 然后写入 1 到 DPSEL, 填 0 到 NDPS 寄存器。

自动加减数据指针的功能是在执行 MOVX @DPTR 指令后, 可使得指针指向的地址自动加 1 或减 1。因此, 它能够连续的访问外部存储器, 而不需要重复的去指定数据指针指向的地址。这些功能由 DPC 寄存器控制, 每个 DPTR 都有单独的 DPC 寄存器位, 在数据传输中提供高灵活性。DPC 寄存器地址 0x93 指向窗口, 在那里使用 DPS 寄存器选择实际的 DPC, 与 DPTR 相同。

5.8 堆栈



可从内部 RAM (IRAM) 中分出任意一部分作为堆栈使用, 但要求手动分配以保证堆栈区域不会与其它 RAM 的变量重叠。堆栈溢出或者下溢会导致其它 RAM 的变量重写错误, 故在分配堆栈的区域时必须考虑到这些问题以避免这种情况的发生。

默认情况下, 堆栈指针 (SP 寄存器) 指向 0x07, 就是指堆栈的区域从 IRAM 地址中的 0x08 开始。换句话说, 如果计划想将堆栈区域设置为从 IRAM 的 0xC0 开始, 就要在系统复位后将 SP 寄存器设置为 0xBF。

一条汇编 PUSH 指令占用堆栈中的一个字节, LCALL, ACALL 指令以及中断分别占用堆栈中的两个字节 POP 指令释放一个字节, RET/RETI 指令释放两个字节。

5.9 特殊功能寄存器 (SFR)

BIN HEX	000	001	010	011	100	101	110	111
F8	P5	P0M	P1M	P2M	P3M	P4M	P5M	PFLAG
F0	B	P0UR	P1UR	P2UR	P3UR	P4UR	P5UR	SRST
E8	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
E0	ACC	SPSTA	SPCON	SPDAT	P1OC	CLKSEL	CLKCMD	-
D8	S0CON2	-	I2CDAT	I2CADR	I2CCON	I2CSTA	-	-
D0	PSW	CHS	VREG	AMPM	ADCM1	ADCM2	BZRM	LBTM
C8	-	TC1M	TC1RL	TC1RH	TC1CL	TC1CH	TC1DL	TC1DH
C0	IRCON	TC0M	TC0RL	TC0RH	TC0CL	TC0CH	TC0DL	TC0DH
B8	IEN1	IP1	S0RELH	T0M	T0C	MIN	SEC	-
B0	P3	LCDM1	LCDM2	-	-	ADCDL	ADCDM	ADCDH
A8	IEN0	IP0	S0RELL	-	-	-	-	-
A0	P2	TC2M	TC2RL	TC2RH	TC2CL	TC2CH	TC2DL	TC2DH
98	S0CON	S0BUF	IEN2	OPM	PEKEY	P5CON	P3CON	P4CON
90	P1	P1W	DPS	DPC	PECMD	PEROMA	PEROMP	PEROMD
88	FLCTIM	-	-	-	-	-	CKCON	PEDGE
80	P0	SP	DPL	DPH	DPL1	DPH1	WDTR	PCON

5.10 系统控制寄存器

SP-堆栈指针寄存器

地址: 81H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SP	SP[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	1	1	1

位号	符号	描述
7:0	SP[1:0]	堆栈指针

DPL-数据指针 0 低字节寄存器

地址: 82H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL	DPL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPL[7:0]	DPTR0 的低字节

DPH-数据指针 0 高字节寄存器

地址: 83H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH	DPH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPH[7:0]	DPTR0 的高字节

DPL1-数据指针 1 低字节寄存器

地址: 84H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL1	DPL1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPL1[7:0]	DPTR1 的低字节

DPH1-数据指针 1 高字节寄存器
地址: 85H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH1	DPH1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPH1[7:0]	DPTR1 的高字节

DPS-数据指针选择寄存器
地址: 92H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	--							DPSEL
R/W	--							R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	DPSEL	DPTR 选择控制位。 0: 选择 DPH/DPL (DPTR0) ; 1: 选择 DPH1/DPL1 (DPTR1) 。

DPC-数据指针控制寄存器
地址: 93H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPC	--	--	--	--	NDPS	ATMS	ATMD	ATME
R/W	--	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	NDPS	下一个 DPTR 选择位。 执行 MOVX @DPTR 指令后, 自动加载该位到 DPSEL。
2:1	ATMS/ATMD	自动加减数据指针 (使能 ATME 位时有效)。 00: 执行 MOVX @DPTR 指令后+1; 01: 执行 MOVX @DPTR 指令后-1; 10: 执行 MOVX @DPTR 指令后+2; 11: 执行 MOVX @DPTR 指令后-2。
0	ATME	自动加减数据指针功能。 0: 关闭; 1: 使能。

ACC 寄存器
地址: E0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	ACC[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ACC[7:0]	8 位数据寄存器用于转移或操控 ALU 和数据存储器之间的数据, 若操作结果溢出 (OV) 或者由借位 (C 或 AC), 以及相等情况 (P) 发生时, 该标志位会在 PSW 寄存器中进行设置。

B-乘法/除法指令数据缓存器
地址: F0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
B	B[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	B[7:0]	B 寄存器在使用乘法和除法指令时使用, 而且还能作为 scratch-pad 寄存器来保留临时数据。

PSW-系统标志寄存器
地址: D0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	CY	进位标志。 0: 加法运算后没有进位、减法运算有借位发生或移位后移出逻辑“0”或比较运算的结果<0; 1: 加法运算后有进位、减法运算没有借位发生或移位后移出逻辑“1”或比较运算的结果≥0。
6	AC	辅助进位标志。 0: BCD 操作时没有从 ACC 的第三位开始执行; 1: BCD 操作时从 ACC 的第三位开始执行。
5	F0	通用标志位, 可任意设定。
4:3	RS[1:0]	寄存器 bank 选择控制位, 用于选择工作寄存器 bank。 00: 00H-07H (Bank0); 01: 08H-0FH (Bank1); 10: 10H-17H (Bank2); 11: 18H-1FH (Bank3)。
2	OV	溢出标志。 0: 算术操作时, ACC 没有溢出; 1: 算术操作时, ACC 溢出。
1	F1	通用标志位, 可任意设定。
0	P	奇偶标志位。 0: A 中 1 的个数为偶数; 1: A 中 1 的个数为奇数。

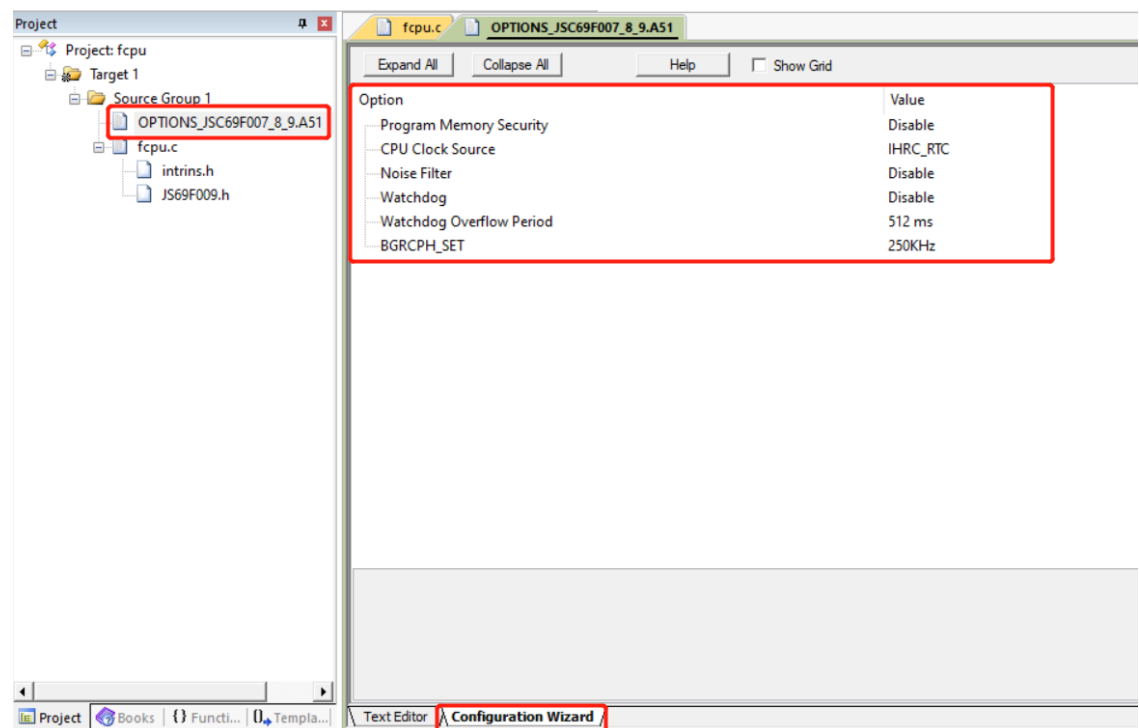
6 复位和上电控制

复位和上电控制有下列方式：低电压检测（LVD），看门狗，可编程的软件复位引脚和软件复位。前面 3 种方式可触发额外的上电流程，随后单片机初始化所有的寄存器，并且程序从复位向量（ROM 地址 0x0000）处重新开始执行。

6.1 复位配置和上电控制

关于复位源的合适参数和 CPU 时钟源选择，调试工具具体选项如下所示：

- 程序加密
- CPU 时钟源
- Noise Filter
- 看门狗



6.2 上电流程

LVD、看门狗和外部复位引脚可触发上电流程，在复位信号结束和执行程序之间的时候发生。

总的来说，上电流程包括 2 个阶段：电源稳定期和时钟稳定期。

在典型条件下，电源稳定期花费 5ms，然后单片机自动选择 CPU 时钟源。驱动选择的时钟源后，系统计数时钟周期的 4096 次以确保时钟稳定。

6.3 LVD 复位

低电压检测监控 VDD 引脚的电压，其电压点为 1.6V。VDD 的电压低于 1.6V 时，MCU 复位。

6.4 看门狗复位

看门狗是周期性的复位信号发生器，用于监控程序的执行流程。其内部定时器可在程序流程的检测点被清零，因此，只有在发生软件问题后才会产生实际的复位信号。通过写入 0x5A 到 WDTR 就可以在程序中设置检测点。

WDT 时钟预分器	Clock/1	Clock/2	Clock/4	Clock/8
看门狗间隔时间	64ms	128ms	256ms	512ms

看门狗的工作模式如下所示：

Always mode: 其内部定时器在 CPU 的所有工作模式（NORMAL，IDLE，SLEEP）下都在计数。

Enable mode: 其内部定时器只在 CPU 的 Normal 模式下计数，在 IDLE 和 SLEEP 模式下不会触发看门狗复位。

Disable mode: 其内部定时器在 CPU 的所有工作模式下都不计数，在这种模式下不会触发看门狗复位。当看门狗一直处于工作状态时，系统会消耗额外的功耗。

6.5 外部引脚复位

JSC69F007/8/9 系列没有外部复位引脚。

6.6 软件复位

连续设置 SRSTREQ 位后会产生软件复位，因此，该程序使能固件的性能以复位单片机（如更新固件后复位）。下面的这段 C 程序代码就因为反复地设置 SRST 寄存器的最小位而导致软件复位。

```
SRST = 0x01;  
SRST = 0x01;
```

6.7 复位和上电控制寄存器

PFLAG-复位标志寄存器

地址: FFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PFLAG	POR	WDT	--	--	--	--	--	--
R/W	R	R	--	--	--	--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	POR	若单片机已经由 LVD 触发复位, 则该位自动置 1
6	WDT	若单片机已经由看门狗触发复位, 则该位自动置 1

SRST-软件复位控制寄存器

地址: F7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SRST	--	--	--	--	--	--	--	SRSTREQ
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	SRSTREQ	连续设置该位 2 次触发软件复位

WDTR-看门狗定时器清零寄存器

地址: 86H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTR	WDTR[7:0]							
R/W	W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	WDTR[7:0]	由 WDTR 寄存器清看门狗定时器, 写入 0x5A 到 WDTR 寄存器复位看门狗定时器。

7 系统时钟和电源管理

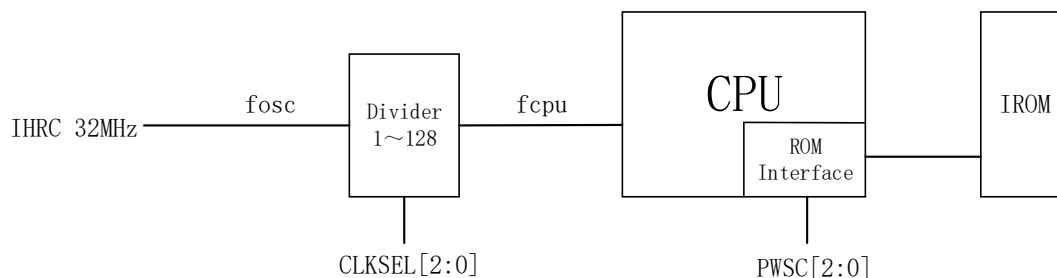
单片机内置 3 个不同的操作模式：**NORMAL** 模式，**IDLE** 模式和 **STOP** 模式以省电。

NORMAL模式是指CPU和外设功能都正常工作，系统时钟是通过所选择的时钟源、程序设置的分频数、以及程序设置的ROM等待时间所确定的。**IDLE**模式是指CPU时钟暂停的状态，但保留了外设功能（如定时器、SPI、UART和 I2C）。与之相反的是，**STOP**模式则禁止所有功能和时钟发生器，直至唤醒信号将系统唤醒进入**NORMAL**模式。在**STOP**模式下，LCD和T0-RTC可以正常工作，此功能可以为低待机电流提供应用。

7.1 系统时钟

该单片机内置时钟发生器（**IHRC 32MHz**）。在复位或上电过程中，系统自动加载 **IHRC**。**IHRC** 可以看作是 **Fosc**，**Fosc** 一旦选定就不能再改变。

当系统时钟稳定后，**Fosc** 可以分频为 $Fosc/1 \sim Fosc/128$ ，由 **CLKSEL** 寄存器控制。CPU 使用分频之后的时钟作为它的时钟频率(称作 **Fcpu**)。写入 0x69 到 **CLKCMD** 寄存器时设置 **CLKSEL**。



ROM 接口位于 **CPU** 和 **IROM**（程序存储器）之间，可设置 **ROM** 读取周期以支持低速程序存储器。例如：**CPU** 计划运行在 **32MHz**，而 **IROM** 只能运行在 **16MHz** 以下，则在 **CKCON** 寄存器中必须设置 **ROM** 读取周期为增加 1 个以上的 **Fcpu**。

$$\text{IROM fetching周期} = \frac{F_{\text{cpu}}}{\text{PWSC}[2:0] + 1} \leq 16\text{MHz}, \text{PWSC}[2:0] = 0 \sim 7$$

7.2 Noise Filter

Noise Filter 功能由 **Noise Filter** 选项控制，是一个低通滤波器，并支持晶振模式。这个选项的目的在于滤除外部高速震荡源上的高频噪声干扰。高干扰环境下，强烈建议使能 **Noise Filter** 以减少噪音干扰。

7.3 电源管理

复位信号和上电结束后，CPU 以 F_{cpu} 的速率开始执行程序。系统以 **Normal** 模式开始工作。

PCON 寄存器的最低 2 位（bit0-IDLE 和 bit1-STOP）控制单片机的电源管理部分。

若 IDLE 位设置为 1，只有 CPU 时钟源被关闭。因此，在这种状态下，外设功能（如定时器、PWM、SPI、UART 和 I2C）和时钟发生器（IHRC 32MHz）仍然正常工作。P0/P1 输入的任何改变和中断事件都会导致单片机返回到 **Normal** 模式，IDLE 位自动清零。

若 STOP 位设置为 1，CPU、外设和时钟发生器都处于停止状态，在这个模式下，寄存器中存储的数据和 RAM 都保持不变。P0/P1 输入的任何改变都可将单片机唤醒并使系统重新开始执行，STOP 位自动清零。

*注：用户用 C 编译器开发时，强烈建议使用 IDLE 和 STOP 宏来控制单片机的系统模式，来代替直接设置 IDLE 和 STOP 位。

表 7-1 退出低功耗方式

模式	唤醒方式	唤醒条件	备注
IDLE 模式	任何中断的发生	使能全局中断 EAL(IEN0.7)，使能其他中断源	-
	P0/P1 输入的任何改变	设置 P0/P1 端口为输入模式，使其端口状态由高变低或由低变高	-
STOP 模式	P0/P1 输入的任何改变	设置 P0/P1 端口为输入模式，使其端口状态由高变低或由低变高	-
	T0 溢出	设置 T0ENB=1 并且 T0TB=1。 当 T0 溢出时，系统被唤醒	

7.4 系统时钟和电源管理寄存器

CKCON-扩展周期寄存器

地址: 8EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	--	PWSC2	PWSC1	PWSC0	ESYN	EWSC2	EWSC1	EWSC0
R/W	--	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	1	0	0	0	1

位号	符号	描述
6:4	PWSC[2:0]	增加读取程序存储器的周期。 000: 无; (若 $F_{cpu} \leq 16\text{MHz}$, 设置为 000) 001: 1 个周期; (若 $F_{cpu} > 16\text{MHz}$, 设置为 001) 010: 2 个周期; (保留) 011: 3 个周期; (保留) 100: 4 个周期; (保留) 101: 5 个周期; (保留) 110: 6 个周期; (保留) 111: 7 个周期。 (保留)
3	ESYN	增加额外的写数据到 XRAM 的周期。 (用户设置为 0)
2:0	EWSC[2:0]	增加读取 XRAM 的周期。 000: 无; (用户设置为 0) 001: 1 个周期; 010: 2 个周期; 011: 3 个周期; (保留) 100: 4 个周期; (保留) 101: 5 个周期; (保留) 110: 6 个周期; (保留) 111: 7 个周期。 (保留)

CLKSEL-时钟切换选择寄存器
地址: E5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKSEL	--	--	--	--	--	CLKSEL2	CLKSEL1	CLKSEL0
R/W	--	--	--	--	--	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

位号	符号	描述
2:0	CLKSEL[2:0]	CLKSEL 中的设置将在写 CLKCMD 之后生效。 000: Fcpu = Fosc/128; 001: Fcpu = Fosc/64; 010: Fcpu = Fosc/32; 011: Fcpu = Fosc/16; 100: Fcpu = Fosc/8; 101: Fcpu = Fosc/4; 110: Fcpu = Fosc/2; 111: Fcpu = Fosc/1。 *设置 CLKSEL[2:0]后, CLKCMD 寄存器必须写入 0x69 以应用 CLKSEL 的设置。

CLKCMD-时钟切换控制寄存器
地址: E6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCMD	CMD[7:0]							
R/W	W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	CMD[7:0]	写入 0x69 来应用 CLKSEL 的设置

PCON-电源控制寄存器
地址: 87H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	--	--	--	P2SEL	GF0	STOP	IDLE
R/W	R/W	--	--	--	R/W	R/W	W	W
复位值	0	0	0	0	1	0	0	0

位号	符号	描述
7	SMOD	请参考 UART 章节。
3	P2SEL	高命令地址字节配置位。执行 MOVX @Ri 操作时选择地址的高字节 (XRAM[15:8])。 0: XRAM[15:8]=P2REG, P2REG 为 P2 输出寄存器的内容; 1: XRAM[15:8]=00。
2	GF0	通用标志。
1	STOP	1: 单片机切换到 STOP 模式。
0	IDLE	1: 单片机切换到 IDLE 模式。

P1W-P1 唤醒控制寄存器
地址: 91H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	P1nW	0: 禁止 P1.n 的唤醒功能; 1: 使能 P1.n 的唤醒功能。

8 中断

JSC69F007/8/9 包含 10 个中断源（2 个外部中断和 8 个内部中断），分为 4 个优先级别。每个中断源包含 1 个或多个中断请求标志。有中断发生时，相对应的中断请求位置为逻辑 1。若同时使能中断使能位和全局中断使能（EAL=1）时，有中断请求时则执行该中断服务程序（ISR）。多数请求振荡器标志位必须由软件清零，而有些中断请求标志位由硬件自动清零。最后，执行 RETI 指令后，整个 ISR 结束。中断源的简称，中断向量，优先级别和控制位如下表所示：

中断源	使能中断标志位	请求标志位(IRQ)	IRQ 清除	优先级/向量
系统复位	-	-	-	0/0x0000
INT0	EX0	IE0	自动清 0	1/0x0003
INT1	EX1	IE1	自动清 0	2/0x000B
TC0	ETC0	TCF0	自动清 0	3/0x0013
T0	ET0	TF0	自动清 0	4/0x0093
TC1	ETC1	TFC1	自动清 0	5/0x001B
TC2	ETC2	TFC2	自动清 0	6/0x009B
UART	ES0	TI0/RI0	软件清 0	7/0x0023
SPI	ESPI	SPIF/MODF	软件清 0	8/0x00A3
ADC	EADC	ADCF	软件清 0	9/0x002B
I2C	EI2C	SI	软件清 0	10/0x00AB

8.1 中断操作

中断操作由中断请求标志位和中断使能位控制。中断请求标志位显示中断源的状态，与中断功能的状态（使能或禁止）无关。同时使能中断使能位和全局中断使能位（EAL=1）且中断请求标志位有效时，程序计数器指向中断向量（0x03-0xEB），系统执行相对应的中断服务程序 ISR。

8.2 中断优先级

每个中断源都有特定的优先级别。若同时发生 2 个中断，系统会先执行优先级别高的 ISR，然后再执行优先级别低的 ISR。下一次的 ISR 必须要等待前面的 ISR 执行完成之后才可以执行，不理睬中断的优先级别。

对应特定的优先权需求，需要用到 4 级的优先级别（级别 0-级别 3）。所有的中断源按优先级别分为 6 大类（Group0-Group5），每组设置为同样的特定优先级别，由寄存器 IP0/IP1 设置，级别 3 最高，级别 0 最低。同组的中断源共用同样的优先级别，对于同样的优先级别，优先权的规则按照默认的优先权。

优先级别	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

首先执行优先级别较高的 ISR，优先级别较高的 ISR 可以打断执行中的优先级别较低的 ISR，直到优先级别较高的 ISR 执行完成之后再执行优先级别较低的 ISR。

Group		
Group 0	INT0	-
Group 1	INT1	-
Group 2	TC0	T0
Group 3	TC1	TC2
Group 4	UART	SPI
Group 5	ADC	I2C

8.3 中断寄存器

IP0-中断优先权寄存器 0

地址: A9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP0	--	--	IP05	IP04	IP03	IP02	IP01	IP00
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

位号	符号	描述
5:0	IP0[5:0]	中断优先权。和 IP1 寄存器的相对应的位结合在一起可以指定相对应 Group 的中断优先级别。 设置: IP0x = 1, 为 xGroup 设置高优先级位

IP1-中断优先权寄存器 1

地址: B9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP1	--	--	IP15	IP14	IP13	IP12	IP11	IP10
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

位号	符号	描述
5:0	IP1[5:0]	中断优先权。和 IP0 寄存器的相对应的位结合在一起可以指定相对应 Group 的中断优先级别。 设置: IP1x = 1, 为 xGroup 设置高优先级位

***例:** 优先级 GP0>GP1>GP2>GP3=GP4=GP5

IP0 = 0X05;

IP1 = 0X03;

***例:** 优先级 GP5>GP4>GP3>GP2=GP1=GP0

IP0 = 0X28;

IP1 = 0X30;

IEN0-中断使能寄存器 0
地址: A8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EAL	--	--	ES0	--	EX1	ET0	EX0
R/W	R/W	--	--	R/W	--	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	EAL	所有中断使能控制位。 0: 禁止; 1: 使能。
4	ES0	UART 中断控制位。 0: 禁止; 1: 使能。
2	EX1	外部中断 P0.1 (INT1) 中断控制位。 0: 禁止; 1: 使能。
1	ET0	T0 定时器中断控制位。 0: 禁止; 1: 使能。
0	EX0	外部中断 P0.0 (INT0) 中断控制位。 0: 禁止; 1: 使能。

IEN1-中断使能寄存器 1
地址: B8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	--	--	--	--	--	--	ESPI	EI2C
R/W	--	--	--	--	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
1	ESPI	SPI 中断控制位。 0: 禁止; 1: 使能。
0	EI2C	I2C 中断控制位。 0: 禁止; 1: 使能。

IEN2-中断使能寄存器 2
地址: 9AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN2	--	--	--	EADC	ETC2	ETC1	ETC0	--
R/W	--	--	--	R/W	R/W	R/W	R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	EADC	ADC 中断控制位。 0: 禁止; 1: 使能。
3	ETC2	TC2 溢出中断控制位。 0: 禁止; 1: 使能。
2	ETC1	TC1 溢出中断控制位。 0: 禁止; 1: 使能。
1	ETC0	TC0 溢出中断控制位。 0: 禁止; 1: 使能。

IRCON-中断请求寄存器
地址: C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCON	ADCF	TCF2	TCF1	TCF0	--	TF0	IE1	IE0
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

位号	符号	描述
7	ADCF	ADC 中断请求标志位。 0: 无 ADC 中断请求; 1: 请求 ADC 中断。
6	TCF2	TC2 中断请求标志位。 0: 无 TC2 中断请求; 1: 请求 TC2 中断。
5	TCF1	TC1 中断请求标志位。 0: 无 TC1 中断请求; 1: 请求 TC1 中断。
4	TCF0	TC0 中断请求标志位。 0: 无 TC0 中断请求; 1: 请求 TC0 中断。
2	TF0	T0 定时器中断请求标志位。 0: 无 T0 中断请求; 1: 请求 T0 中断。
1	IE1	外部中断 P0.1 (INT1) 请求标志位。 0: 无 INT1 中断请求; 1: 请求 INT1 中断。
0	IE0	外部中断 P0.0 (INT0) 请求标志位。 0: 无 INT0 中断请求; 1: 请求 INT0 中断。

S0CON-UART 控制寄存器
地址: 98H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
1	TI0	UART 发送中断请求标志位, 显示 UART 串行传输的完成状态。在模式 0 位 8 结束时, 或者其它模式的停止位开始时由硬件设置为 1; 必须由软件清零。 0: 无 UART 发送中断请求; 1: UART 发送请求中断。
0	RI0	UART 接收中断请求标志位, 显示 UART 串行接收的完成状态。在模式 0 位 8 结束时, 或者其它模式的停止位中间时由硬件设置为 1; 必须由软件清零。 0: 无 UART 接收中断请求; 1: UART 接收请求中断。

SPSTA-SPI 状态寄存器
地址: E1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSTA	SPIF	WCOL	SSERR	MODF	--	--	--	--
R/W	R	R/W	R/W	R	--	--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	SPIF	SPI 完成通讯标志位。 通讯结束时自动设置为 1; 读取 SPSTA、SPDAT 寄存器时自动清零
4	MODF	模式错误标志位。

I2CCON-I2C 接口操作控制寄存器
地址: DCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	SI	串行中断标志位。 当进入了 I2C 的 26 个状态当中的 25 个状态时，SI 标志位会被硬件置 1，只有当 I2C 状态寄存器为 0xF8 时，SI 标志位才没有被置 1，表示没有可用的相关状态信息。SI 标志位必须由软件清零，必须通过写 0 到 SI 标志才可以清 SI 标志位，写入 1 到该位并不能更改 SI 的值。读取 SPSTA、SPDAT 寄存器时自动清零

9 MDU

乘除法单元是一个内置算法协处理器，可以使单片机去执行额外扩展的算法操作。该单元提供 32 位无符号的除法，16 位无符号的乘法，移位和规范化操作，可以通过写入 MD0-MD5 寄存器的不同顺序来识别这些操作。

9.1 乘法（16 位 x16 位）

乘法的基础原理包括 3 部分：被乘数、乘数和得数。进行乘法操作要求按以下顺序写入乘数和被乘数：MD0（被乘数的低字节），MD4（乘数的低字节），MD1（被乘数的高字节），然后 MD5（乘数的高字节）。写入 MD5 寄存器结束时，自动开始乘法操作，乘法操作共需要 11 个 CPU 周期，其有效的得数可通过特定的顺序读取：MD0（LSB）、MD1、MD2，然后 MD3（MSB）寄存器。

9.2 除法（32 位/16 位和 16 位/16 位）

MDU 支持 2 种除法操作：32 位/16 位和 16 位/16 位。第一种需要 17 个 CPU 周期来进行计算；第二种只需要 9 个周期。

32 位/16 位除法以下面特定顺序写入除数和被除数：MD0、MD1、MD2、MD3、MD4 和 MD5。32 位被除数存入 MD3（最高有效位）到 MD0 寄存器，16 位除数存入 MD5 到 MD4 寄存器（MSB 位于 MD5 寄存器）。

16 位/16 位除法只需要 4 个寄存器，按 MD0、MD1、MD4、MD5 的顺序写入除数和被除数。16 位被除数存入 MD1 到 MD0 寄存器（最高有效位位于 MD1），16 位除数存入 MD5 到 MD4 寄存器（最高有效位位于 MD5）。MDU 从写入数据到 MD5 寄存器之后开始计算，需要 9 个或者 17 个 CPU 周期，取决于被除数的长度。

计算得出的商存入 MD3 到 MD0 寄存器（32 位/16 位除法），或者 MD1 到 MD0 寄存器（16 位/16 位除法），最低有效位位于 MD0 寄存器。不管是否执行了除法操作，余数都放在 MD5（MSB）和 MD4 寄存器中。但是必须最后读取 MD5 以指示是否完成全部除法操作。

9.3 移位和规范化

移位和规范化操作移动 32 位寄存器（MD3 到 MD0，MSB 位于 MD3）一段特定或者不确定的次数。

移位操作中，通过指定的位数左移或者右移 32 位无符号的整数，移动的方向和位数可以通过 ARCON 寄存器来指定。移位操作需要 3-18 个 CPU 周期，取决于移位的次数。

规范化操作中，32 位无符号的整数会重复地向左移位，直到最高有效位（MD3 寄存器的第 7 位）为 1。规范化操作需要 4-19 个 CPU 周期，取决于实际移位的次数。

移位操作和规范化操作都以下的写入顺序开始：MD0、MD1、MD2、MD3，最后写 ARCON 寄存器。最终结果存入 MD0-MD3 寄存器中，以下面顺序来读取：MD0、MD1、MD2、MD3。

9.4 错误标志

错误标志 MDEF 表示出现未正确运行的操作（算法操作被重新开始或者被新操作打断），检测机制会在写数据到 MDx 寄存器时自动使能，在完成计算后对 MDx 寄存器进行读取指令时关闭。

出现下列情形时，错误标志 MDEF 置 1：

1. 在 MDU 进行计算时，对 MDx 寄存器或 ARCON 寄存器中任意一个进行写操作（算法操作重新开始或被中断）；
2. 在错误标志机制使能的情况下，在 MDU 进行计算时，对 MDx 寄存器进行读取操作。此时错误标志位会被置 1，但是计算不会被打断。

错误标志位只有在读“ARCON”寄存器之后才会复位，且该标志位为只读。

9.5 溢出标志（MDOV）

出现下列情形时，溢出标志 MDOV 置 1：

- 1、除数为 0；
- 2、乘法的结果大于 0xFFFF；
- 3、MD3 的最高有效位置 1（MD3.7=1）时开始规范化操作；

MDU 操作与上述情形不符合时，溢出标志清零。注意：溢出标志只受硬件控制，不能执行写操作。

9.6 MDU 寄存器

MDx-MDU 控制寄存器 (MD0-MD5)

地址: E9H-EEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDx	MDx7	MDx6	MDx5	MDx4	MDx3	MDx2	MDx1	MDx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	MDx[7:0]	乘法/除法寄存器

ARCON-MDU 算法控制寄存器

地址: EFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ARCON	MDEF	MDOV	SLR	SC[4:0]				
R/W	R/W	R/W	R/W	R/W				
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	MDEF	MDU 错误标志 MDEF 显示错误的操作 (算法操作重新开始或者被新操作打断)
6	MDOV	MDU 溢出标志 MDU 操作时发生溢出
5	SLR	移位方向 0: 左移 1: 右移
4:0	SC[4:0]	移位计数器 写入 0x00: 执行规范化操作; 完成后可读取实际的移位时间。 写入其它值: 指定移位操作的次数。

10 GPIO

JSC69F007/8/9 共有 44 个 GPIO 引脚, 和 8051 只有开漏输出不同, JSC69F007/8/9 还内置推挽输出结构, 以增强其驱动能力。

10.1 输入输出控制

由 P0M-P5M 寄存器控制输入输出模式。

PxM-Px 输入/输出控制寄存器 (P0M-P1M)

地址: F9H/FAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxM	Px7M	Px6M	Px5M	Px4M	Px3M	Px2M	Px1M	Px0M
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	Px7M	Px.7 的模式选择控制位 0: 输入模式; 1: 输出模式。
6	Px6M	Px.6 的模式选择控制位 0: 输入模式; 1: 输出模式。
5	Px5M	Px.5 的模式选择控制位 0: 输入模式; 1: 输出模式。
4	Px4M	Px.4 的模式选择控制位 0: 输入模式; 1: 输出模式。
3	Px3M	Px.3 的模式选择控制位 0: 输入模式; 1: 输出模式。
2	Px2M	Px.2 的模式选择控制位 0: 输入模式; 1: 输出模式。
1	Px1M	Px.1 的模式选择控制位 0: 输入模式; 1: 输出模式。
0	Px0M	Px.0 的模式选择控制位 0: 输入模式; 1: 输出模式。

P2M-P2 输入/输出控制寄存器
地址: FBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2M	--	--	--	--	P23M	P22M	P21M	P20M
R/W	--	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	P23M	P2.3 的模式选择控制位 0: 输入模式; 1: 输出模式。
2	P22M	P2.2 的模式选择控制位 0: 输入模式; 1: 输出模式。
1	P21M	P2.1 的模式选择控制位 0: 输入模式; 1: 输出模式。
0	P20M	P2.0 的模式选择控制位 0: 输入模式; 1: 输出模式。

PxM-Px 输入/输出控制寄存器 (P3M-P5M)
地址: FCH-FEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxM	Px7M	Px6M	Px5M	Px4M	Px3M	Px2M	Px1M	Px0M
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	Px7M	Px.7 的模式选择控制位 0: 输入模式; 1: 输出模式。
6	Px6M	Px.6 的模式选择控制位 0: 输入模式; 1: 输出模式。
5	Px5M	Px.5 的模式选择控制位 0: 输入模式; 1: 输出模式。
4	Px4M	Px.4 的模式选择控制位 0: 输入模式; 1: 输出模式。
3	Px3M	Px.3 的模式选择控制位 0: 输入模式; 1: 输出模式。
2	Px2M	Px.2 的模式选择控制位 0: 输入模式; 1: 输出模式。
1	Px1M	Px.1 的模式选择控制位 0: 输入模式; 1: 输出模式。
0	Px0M	Px.0 的模式选择控制位 0: 输入模式; 1: 输出模式。

P1OC-开漏功能控制寄存器
地址: E4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1OC	--	--	--	P05OC	P04OC	P14OC	P13OC	P12OC
R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	P05OC	P0.5 开漏输出模式控制位。 0: 禁止; 1: 使能, 输出高电平时变为输入模式。
3	P04OC	P0.4 开漏输出模式控制位。 0: 禁止; 1: 使能, 输出高电平时变为输入模式。
2	P14OC	P1.4 开漏输出模式控制位。 0: 禁止; 1: 使能, 输出高电平时变为输入模式。
1	P13OC	P1.3 开漏输出模式控制位。 0: 禁止; 1: 使能, 输出高电平时变为输入模式。
0	P12OC	P1.2 开漏输出模式控制位。 0: 禁止; 1: 使能, 输出高电平时变为输入模式。

10.2 输入数据和输出数据

当从 P0~P5 寄存器进行读操作时, 当前引脚的逻辑电平将取决于它的外部状态。在某些情况下, 当 IO 口在和其他功能复用时, 例如 UART 或 I2C, 读操作依然可行。

写给 P0~P5 寄存器的值将被马上锁存, 然而, 只有在 P0M~P5M 被设置为输出模式之后才会被输出。如果这个引脚已经是输出模式了, 任何写到 P0~P5 寄存器的值将会马上输出到这个引脚上。

Px-Px 数据缓存器 (P0-P1)
地址: 80H/90H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	Px[7:0]	读出 1/0: Px.[7:0]为逻辑高电平/低电平 写入 1/0: 输出高电平/低电平 (Px[7:0]M=1 时使能)

P2-P2 数据缓存器
地址: A0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2	--	--	--	--	P23	P22	P21	P20
R/W	--	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	1	1

位号	符号	描述
3:0	P2[3:0]	读出 1/0: P2.[3:0]为逻辑高电平/低电平 写入 1/0: 输出高电平/低电平 (P2[3:0]M=1 时使能)

Px-Px 数据缓存器(P3-P5)
地址: B0H/E8H/F8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	Px[7:0]	读出 1/0: Px.[7:0]为逻辑高电平/低电平 写入 1/0: 输出高电平/低电平 (Px[7:0]M=1 时使能)

10.3 内置上拉寄存器

P0UR-P5UR 寄存器管理每个引脚的内部 100KΩ (典型值) 的上拉电阻。

PxUR-Px 上拉电阻控制寄存器(P0UR-P1UR)
地址: F1H/F2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUR	Px7UR	Px6UR	Px5UR	Px4UR	Px3UR	Px2UR	Px1UR	Px0UR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	Px[7:0]UR	Px.7 的内置上拉电阻控制位。 0: 禁止*; 1: 使能。

*如果引脚为输出模式或者模拟功能, 建议禁止上拉电阻

P2UR-P2 上拉电阻控制寄存器
地址: F3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2UR	--	--	--	--	P23UR	P22UR	P21UR	P20UR
R/W	--	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3:0	P2[3:0]UR	P2.[3:0]的内置上拉电阻控制位。 0: 禁止*; 1: 使能。

*如果引脚为输出模式或者模拟功能, 建议禁止上拉电阻

PxUR-Px 上拉电阻控制寄存器(P3UR-P5UR)
地址: F4H/F5H/F6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUR	Px7UR	Px6UR	Px5UR	Px4UR	Px3UR	Px2UR	Px1UR	Px0UR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	Px[7:0]UR	Px.[7:0]的内置上拉电阻控制位。 0: 禁止*; 1: 使能。

*如果引脚为输出模式或者模拟功能, 建议禁止上拉电阻

10.4 与 LCD 共用的引脚

JSC69F007/8/9 内置 LCD 功能, LCD 驱动输出引脚与 GPIO 共用, 通过 PxCON 寄存器进行设置。

PxCON-Px 配置控制寄存器(P3CON-P5CON)
地址: 9EH/9FH/9DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxCON	PxCON7	PxCON6	PxCON5	PxCON4	PxCON3	PxCON2	PxCON1	PxCON0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	PxCON[7:0]	Px[7:0]功能控制位 0: LCD 功能 1: GPIO 功能

11 外部中断

外部中断 INT0, INT1 内置边沿触发功能, 由 PEDGE 寄存器控制。使能外部中断 (EX0/EX1) 和全局中断 (EAL) 后, 发生边沿触发事件时, 外部中断请求标志位 (IE0/IE1) 置 1, 程序计数器跳转到中断向量 (ORG 0x0003/0x000B) 并执行中断服务程序。在执行 ISR 之前由硬件清中断请求标志。

11.1 外部中断寄存器

PEDGE-外部中断边沿控制寄存器

地址: 8FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEDGE	--	--	--	--	EX1G1	EX1G0	EX0G1	EX0G0
R/W	--	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	0	1	0

位号	符号	描述
3:2	EX1G[1:0]	外部中断 INT1 触发沿控制位。 00: 保留; 01: 上升沿; 10: 下降沿 (默认); 11: 上升/下降沿。
1:0	EX0G[1:0]	外部中断 INT0 触发沿控制位。 00: 保留; 01: 上升沿; 10: 下降沿 (默认); 11: 上升/下降沿。

12 TCx16 位定时器/计数器

16 位二进制定时器 TC0/TC1/TC2 具有基本定时器、事件计数器和 PWM 功能。基本定时器功能支持标志显示 (TCF_x 位) 和中断操作 (中断向量), 间隔时间由 TCxM、TCxC 和 TCxR 寄存器控制。事件计数器更改 TCx 时钟源从系统时钟 (F_{cpu}/F_{osc}/X_{tal}) 为外部时钟类似信号 (如 32768Hz 晶振)。TCx 为计数器记录外部时钟数以进行测量。TCx 内置占空比/周期可编程控制的 PWM, PWM 的周期和占空比由 TCx 定时器时钟速率、TCxR 和 TCxD 寄存器控制, 故 PWM 具有良好的性能来处理红外载波信号、马达控制和亮度调节等。

TCx 定时器的主要功能如下:

- 16 位可编程计数定时器:

根据选择的时钟频率产生周期性的信号

- 中断功能:

TCx 定时器功能支持中断, TCx 定时器溢出时, TCxIRQ 有效, 系统执行中断服务程序。

- 事件计数器:

事件计数器对外部事件计数。时钟频率产生周期性的信号。

- 占空比/周期可编程控制的 PWM:

PWM 的占空比/周期由 TCxR 和 TCxD 寄存器控制。

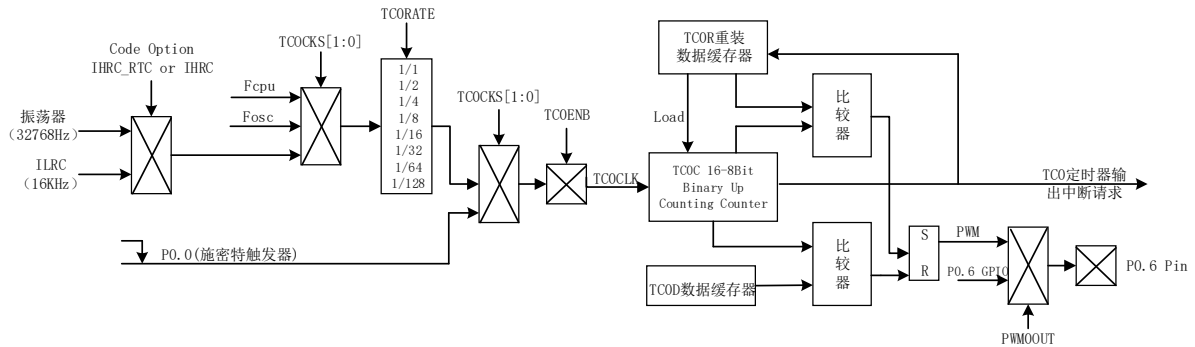
- IDLE 模式功能:

Normal 和 IDLE 模式下, 所用 TCx 功能 (定时器、PWM、事件计数、自动重装) 正常工作, STOP 模式下则停止工作。

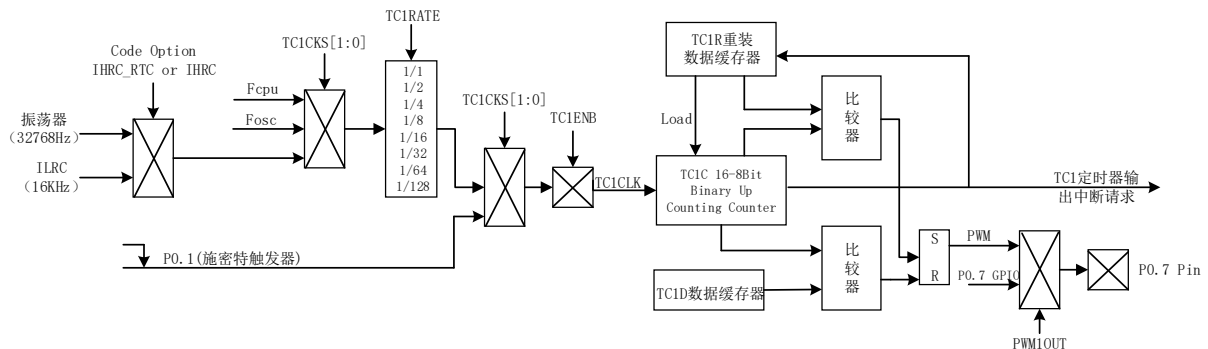
12.1 定时器计数器框图和时钟源选择

下图是 TC0、TC1 和 TC2 的时钟选择电路, 每个都有 3 个内部时钟源 (F_{cpu}、F_{osc} 和 32K X_{tal}) 和 1 个外部信号输入源可供选择。下面分别是 TC0、TC1 和 TC2 内部结构框图。

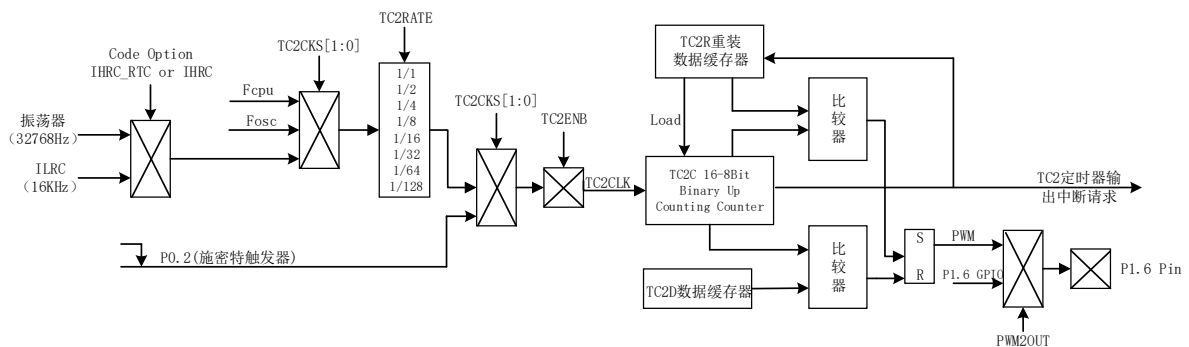
TC0 结构框图:



TC1 结构框图:



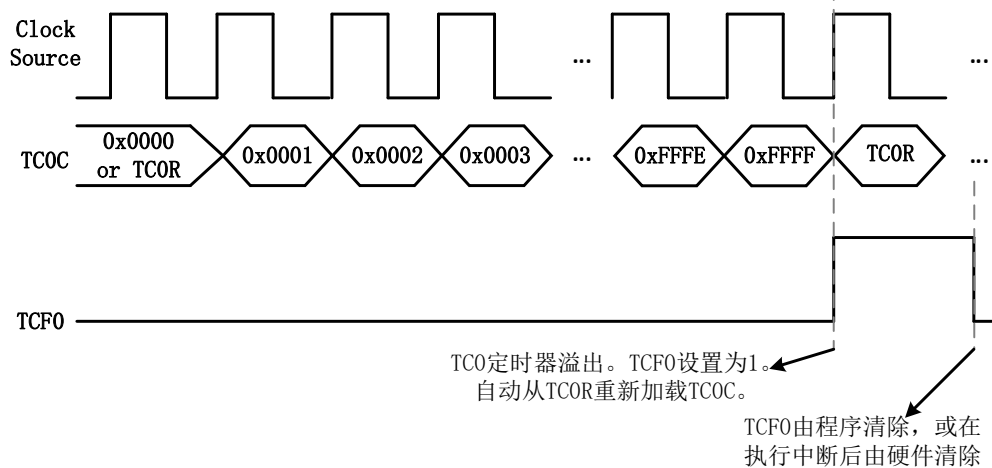
TC2 结构框图:



12.2 TC0/TC1/TC2 定时器操作

以下说明以 TC0 为例, TC0 定时器由 TC0ENB 控制, 当 TC0ENB=1 时, TC0 开始计数。一个计数周期即为一个时钟源周期。TC0ENB=1 时, TC0C 为 TC0 计数器并开始计数, TC0C 溢出 (从 0xFFFF 到 0x0000) 时, 中断标志位 TCF0 置 1 以显示溢出状态。执行中断后, 中断标志 TCF0 或由程序清零, 或由硬件自动清零。TCx 内置自动重装功能始终处于使能状态, TC0 溢出后, TC0R 的值自动存入 TC0C。TC0 具有双重缓存器, 若 TC0R 的值由程序进行更改, 则在下一次溢出时自动装载新值, 否则 TC0 间隔时间将出错。如果使能 TC0 中断功能 (ETC0=1), 在 TC0 溢出时系统执行中断服务程序。

$$\text{TC0 间隔时间} = (65536 - \text{TC0R}) * 1 / (\text{F}_{\text{TC0 clock source}} / \text{TC0Rate})$$



12.3 TC0/TC1/TC2 计数功能和寄存器

以下说明以 TC0 为例，16 位计数器 TC0C 溢出时，TCF0 置 1，由硬件或软件清零。TC0C 决定 TC0 的间隔时间，由下面公式来计算出一个合适的值。首先必须写入一个合适的值到 TC0C 和 TC0R 寄存器，然后再使能 TC0 定时器以确保第一个周期无误。TC0 溢出后，自动加载 TC0R 寄存器的值到 TC0C 寄存器中，无需通过程序操作。

TC0C 初始值的计算公式如下：

$$\text{TC0C 初始值} = 65536 - (\text{TC0 间隔时间} \times \text{TC0 时钟速率})$$

TC0CH-TC0 定时器高字节计数器

地址: C5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CH	TC0CH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TC0CH[7:0]	TC0 计数器的高字节

TC0CL-TC0 定时器低字节计数器

地址: C4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CL	TC0CL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TC0CL[7:0]	TC0 计数器的低字节

12.4 TC0/TC1/TC2 自动重装载功能和寄存器

以下说明以 TC0 为例, TC0 内置自动重装功能, TC0R 寄存器存储重装值。TC0C 溢出时, TC0R 的值自动装入 TC0C 中。TC0 定时器工作在计时模式时, 要通过修改 TC0R 寄存器来修改 TC0 的间隔时间, 而不是通过修改 TC0C 寄存器。在 TC0 定时器溢出后, TC0R 会将新的值装载到 TC0C 寄存器中。但在初次设置 TC0M 时, 必须要在开启 TC0 定时器前把 TC0C 以及 TC0R 设置成相同的值。

TC0 为双重缓存器结构。若程序对 TC0R 进行了修改, 那么修改后的 TC0R 值首先被暂存在 TC0R 的第一个缓存器中, TC0 溢出后, TC0R 的新值就会被存入 TC0R 缓存器中, 从而避免 TC0 中断时间出错以及 PWM 误动作。

TC0R 初始值的计算公式如下:

$$\text{TC0R 初始值} = 65536 - (\text{TC0 间隔时间} \times \text{TC0 时钟速率})$$

TC0RH-TC0 计数器重装缓存器高字节

地址: C3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0RH	TC0RH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TC0RH[7:0]	TC0 重装缓存器的高字节

TC0RL-TC0 计数器重装缓存器低字节

地址: C2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0RL	TC0RL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TC0RL[7:0]	TC0 重装缓存器的低字节

例: 计算 TC0C 和 TC0R 的值, TC0 的间隔时间为 10ms。

$$\text{TC0 时钟源 } F_{\text{cpu}} = 32\text{MHz}/32 = 1\text{MHz}, \text{ TC0RATE} = 000(F_{\text{cpu}}/128)$$

$$\text{TC0 间隔时间} = 10\text{ms}. \text{ TC0 时钟速率} = 32\text{MHz}/32/128$$

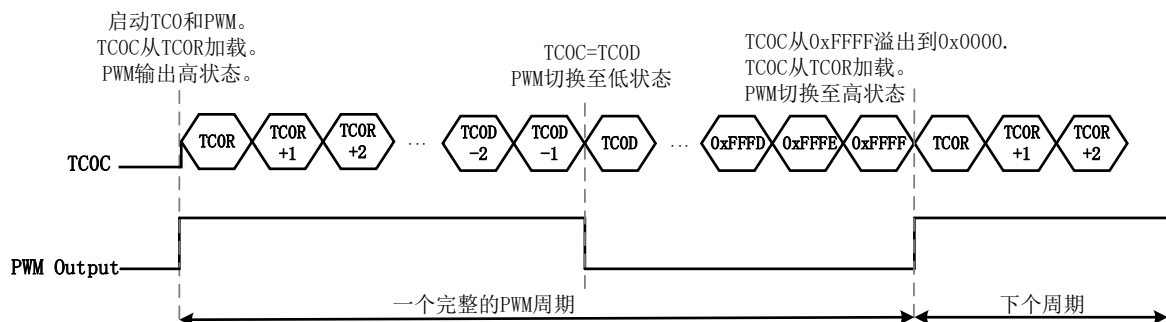
$$\text{TC0C/TC0R 初始值} = 65536 - (\text{TC0 间隔时间} \times \text{输入时钟})$$

$$= 65536 - (10\text{ms} \times 32\text{MHz}/32/128)$$

$$= 65536 - (10 \times 2 \times 32 \times 106/32/128) = \text{FFB2H}$$

12.5 TC0/TC1/TC2 的 PWM 功能

以下说明以 TC0 输出 PWM0 为例，可编程控制占空比/周期的 PWM 可以提供不同的 PWM 信号。使能 TC0 定时器且 PWM0OUT=1（使能 PWM 输出）时，由 PWM 输出引脚 P0.6 输出 PWM 信号。PWM 首先输出高电平，然后输出低电平。TC0R 寄存器控制 PWM 的周期，TC0D 决定 PWM 的占空比（脉冲高电平的长度）。使能 TC0 定时器时，设置 TC0C 的初始值为 TC0R 重装值。当 TC0C=TC0D 时，PWM 高脉冲完成，转换为低电平。当 TC0 溢出（TC0C 计数由 0xFFFF 到 0x0000），一个完整的 PWM 周期完成。PWM 转变为高电平进入下一个周期。PWM 自动将 TC0R 的值装入 TC0C 中，以保持 PWM 的连续性。在 PWM 输出的过程由程序更改 PWM 的周期，则在下一个周期开始输出新的占空比的 PWM 信号。



TC0D 决定 PWM 的占空比。PWM 模式下，TC0R 控制 PWM 的周期，TC0D 控制 PWM 的占空比。其操作基于定时计数器的值。TC0C=TC0D 时，PWM 输出低电平，这样易于配置 TC0D 以选择正确的 PWM 占空比。

TC0D 初始值的计算公式如下：

$$\text{TC0D 初始值} = \text{TC0R} + (\text{PWM 高脉冲宽度周期} / \text{TC0 时钟 rate})$$

例：计算 TC0D 的值，PWM 信号为 100Hz，1/3 占空比。

TC0 时钟源为 $F_{\text{cpu}} = 32\text{MHz} / 32 = 1\text{MHz}$, $\text{C0RATE} = 000$ ($F_{\text{cpu}} / 128$)

TC0R=FFB2H (TC0RH: FFH, TC0RL: B2H)，TC0 间隔时间=10ms，PWM 周期为 100Hz 占空比 1/3 条件下，高脉冲的宽度约为 3.33ms。

TC0D 间隔时间= FFB2H+(PWM 高脉冲宽度周期/TC0 时钟 rate)

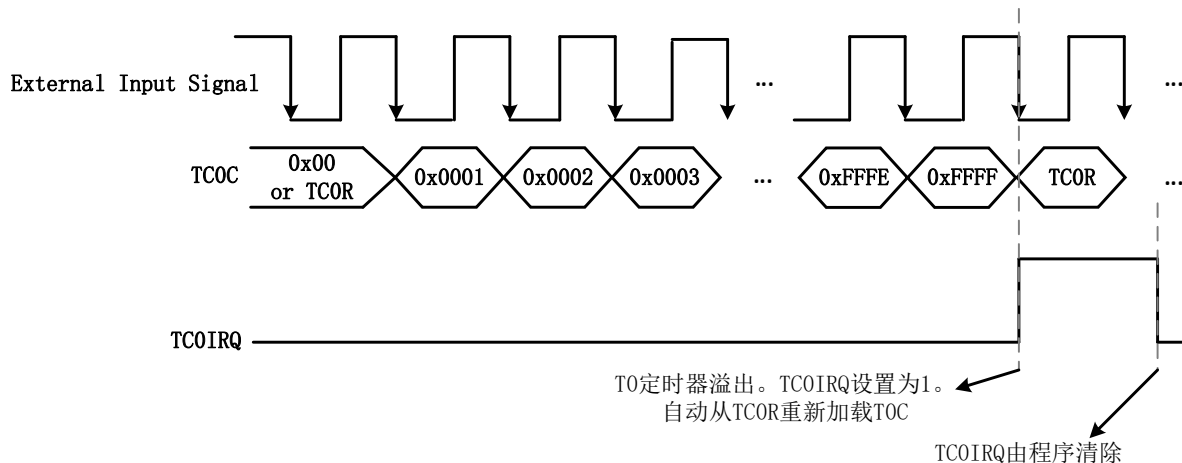
$$= \text{FFB2H} + (3.33\text{ms} * 32\text{MHz} / 32 / 128)$$

$$= \text{FFB2H} + 1\text{AH} = \text{FFCCH}$$

12.6 TC0/TC1/TC2 事件计数器功能

以下说明以 TC0 为例，TC0 事件计数器设置 TC0 时钟源为外部输入引脚 P0.0。TC0CKS1=1 时，TC0 时钟源切换为外部输入引脚 P0.0，TC0 事件计数器的触发沿为下降沿触发。下降沿触发时，TC0C 向上计一个数。TC0C 溢出时（从 0xFFFF 到 0x0000），TC0 触发溢出事件。使能 TC0 事件计数器功能时，禁止外部事件计数器输入引脚 GPIO 模式的唤醒功能以避免计数器触发系统唤醒功能而导致不能保持省电模式。使能 TC0 事件计数器功能时，也要禁止外部事件计数器输入引脚的外部中断功能，则 P00IRQ=0。

事件计数器通常用于测量外部连续信号，如连续脉冲、R/C 振荡信号等。这些信号的相位与单片机的主时钟信号并不同步，针对不同的应用，使用 TC0 事件计数器进行测量时，需要在程序中计算出信号速率。



注：TC0 外部输入信号为 P00
 TC1 外部输入信号为 P01
 TC2 外部输入信号为 P02

12.7 TC0、TC1 和 TC2 寄存器

TC0M-TC0 定时器模式控制寄存器

地址: C1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0M	TC0ENB	TC0RATE[2:0]			TC0CKS[1:0]		PWM0OUT	--
R/W	R/W	R/W			R/W		R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	TC0ENB	TC0 功能控制位。 0: 禁止; 1: 使能。
6:4	TC0RATE[2:0]	TC0 定时器时钟分频位。 000: clock/128; 100: clock/8; 001: clock/64; 101: clock/4; 010: clock/32; 110: clock/2; 011: clock/16; 111: clock/1。
3:2	TC0CKS[1:0]	TC0 定时器时钟源选择位。 00: Fcpu; 01: Fosc; 10: 外部 32K 晶振; 11: P0.0 (事件计数器)
1	PWM0OUT	PWM0 输出控制位。 0: 禁止, P0.6 为 GPIO 模式; 1: 使能, P0.6 输出 PWM 信号。

注: 若选择内部低速时钟作为 TC0 时钟源, 则在不接外部晶振时配置 TC0CKS[1:0]=10 即可;

TC1M-TC1 定时器模式控制寄存器
地址: C9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC1M	TC1ENB	TC1RATE[2:0]			TC1CKS[1:0]		PWM1OUT	--
R/W	R/W	R/W			R/W		R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	TC1ENB	TC1 功能控制位。 0: 禁止; 1: 使能。
6:4	TC1RATE[2:0]	TC1 定时器时钟分频位。 000: clock/128; 100: clock/8; 001: clock/64; 101: clock/4; 010: clock/32; 110: clock/2; 011: clock/16; 111: clock/1。
3:2	TC1CKS[1:0]	TC1 定时器时钟源选择位。 00: Fcpu; 01: Fosc; 10: 外部 32K 晶振; 11: P0.1 (事件计数器)
1	PWM1OUT	PWM1 输出控制位。 0: 禁止, P0.7 为 GPIO 模式; 1: 使能, P0.7 输出 PWM 信号。

注: 若选择内部低速时钟作为 TC1 时钟源, 则在不接外部晶振时配置 TC1CKS[1:0]=10 即可;

TC2M-TC2 定时器模式控制寄存器
地址: A1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC2M	TC2ENB	TC2RATE[2:0]			TC2CKS[1:0]		PWM2OUT	--
R/W	R/W	R/W			R/W		R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	TC2ENB	TC2 功能控制位。 0: 禁止; 1: 使能。
6:4	TC2RATE[2:0]	TC2 定时器时钟分频位。 000: clock/128; 100: clock/8; 001: clock/64; 101: clock/4; 010: clock/32; 110: clock/2; 011: clock/16; 111: clock/1。
3:2	TC2CKS[1:0]	TC2 定时器时钟源选择位。 00: Fcpu; 01: Fosc; 10: 外部 32K 晶振; 11: P0.2 (事件计数器)
1	PWM2OUT	PWM2 输出控制位。 0: 禁止, P1.6 为 GPIO 模式; 1: 使能, P1.6 输出 PWM 信号。

注: 若选择内部低速时钟作为 TC2 时钟源, 则在不接外部晶振时配置 TC2CKS[1:0]=10 即可;

定时器时钟控制列表:

TCxCKS[1:0]	TCxRATE[2:0]	TCx Clock	TCxCKS[1:0]	TxRATE[2:0]	TCx Clock
00	000	Fcpu/128	01	000	Fosc/128
	001	Fcpu/64		001	Fosc/64
	010	Fcpu/32		010	Fosc/32
	011	Fcpu/16		011	Fosc/16
	100	Fcpu/8		100	Fosc/8
	101	Fcpu/4		101	Fosc/4
	110	Fcpu/2		110	Fosc/2
	111	Fcpu/1		111	Fosc/1
10	000	32KHz/128	11	NA	事件计数器功能
	001	32KHz/64			
	010	32KHz/32			
	011	32KHz/16			
	100	32KHz/8			
	101	32KHz/4			
	110	32KHz/2			
	111	32KHz/1			

注: TCx=TC0、TC1、TC2

TCxCH-TCx 计数器高字节(TC0CH、TC1CH、TC2CH)

地址: C5H/CDH/A5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCH	TCxCH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxCH[7:0]	TCx 计数器的高字节

TCxCL-TCx 计数器低字节(TC0CL、TC1CL、TC2CL)

地址: C4H/CCH/A4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCL	TCxCL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxCL[7:0]	TCx 计数器的低字节

TCxRH-TCx 重装缓存器高字节(TC0RH、TC1RH、TC2RH)
地址: C3H/CBH/A3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxRH	TCxRH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxRH[7:0]	TCx 重装缓存器的高字节

TCxRL-TCx 重装缓存器低字节(TC0RL、TC1RL、TC2RL)
地址: C2H/CAH/A2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxRL	TCxRL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxRL[7:0]	TCx 重装缓存器的低字节

TCxDH-PWM 占空比控制缓存器高字节(TC0DH、TC1DH、TC2DH)
地址: C7H/CFH/A7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxDH	TCxDH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxDH[7:0]	TCx 占空比控制的高字节

TCxDL-PWM 占空比控制缓存器低字节(TC0DL、TC1DL、TC2DL)
地址: C6H/CEH/A6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxDL	TCxDL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TCxDL[7:0]	TCx 占空比控制的低字节

IEN2-中断使能寄存器 2
地址: 9AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN2	--	--	--	EADC	ETC2	ETC1	ETC0	--
R/W	--	--	--	R/W	R/W	R/W	R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	EADC	ADC 中断控制位。 0: 禁止; 1: 使能。
3	ETC2	TC2 溢出中断控制位。 0: 禁止; 1: 使能。
2	ETC1	TC1 溢出中断控制位。 0: 禁止; 1: 使能。
1	ETC0	TC0 溢出中断控制位。 0: 禁止; 1: 使能。

IRCON-中断请求寄存器
地址: C0H

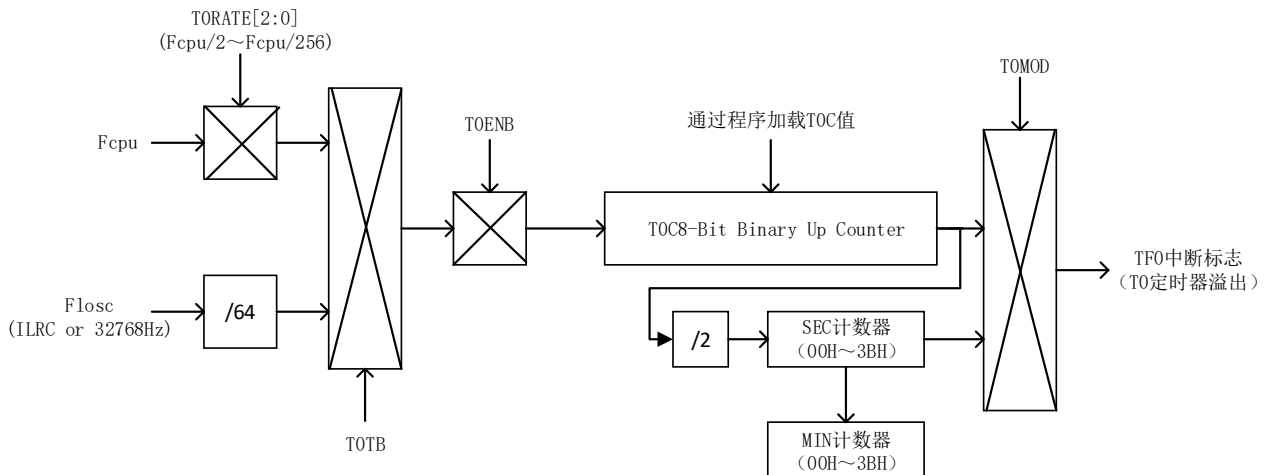
Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCON	ADCF	TCF2	TCF1	TCF0	--	TF0	IE1	IE0
R/W	R/W	R/W	R/W	R/W	--	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	ADCF	ADC 中断请求标志位。 0: 无 ADC 中断请求; 1: 请求 ADC 中断。
6	TCF2	TC2 中断请求标志位。 0: 无 TC2 中断请求; 1: 请求 TC2 中断。
5	TCF1	TC1 中断请求标志位。 0: 无 TC1 中断请求; 1: 请求 TC1 中断。
4	TCF0	TC0 中断请求标志位。 0: 无 TC0 中断请求; 1: 请求 TC0 中断。
2	TF0	T0 中断请求标志位。 0: 无 T0 中断请求; 1: 请求 T0 中断。
1	IE1	外部 P0.1 中断 (INT1) 中断请求标志位。 0: 无 INT1 中断请求; 1: 请求 INT1 中断。
0	IE0	外部 P0.0 中断 (INT0) 中断请求标志位。 0: 无 INT0 中断请求; 1: 请求 INT0 中断。

13 T0 定时器

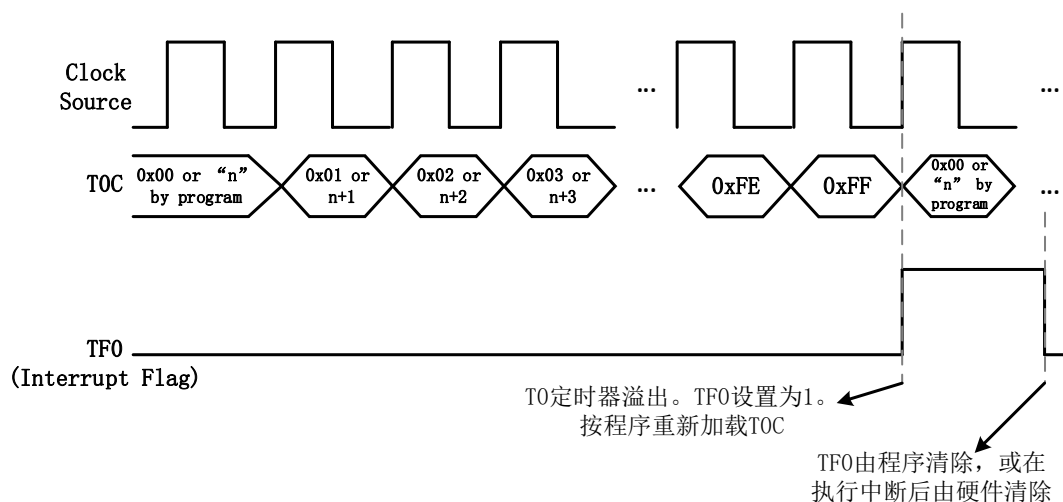
8 位二进制基本定时器 T0，支持中断标志 (TF0) 和中断操作 (中断向量 0x93)。可以通过 T0M 和 T0C 寄存器控制间隔时间，支持 RTC 功能。T0 内置 STOP 模式唤醒功能，STOP 模式下，T0 溢出时，系统被唤醒进入 Normal 模式。

- 8 位可编程计数定时器：根据选择的时钟频率周期性的产生中断请求。
- 中断功能：T0 定时器支持中断功能，当 T0 溢出，TF0 有效，程序计数器跳到中断向量地址执行中断。
- RTC 功能：T0 RTC 功能由 T0TB 位控制，T0MOD=0 时，RTC 的周期为 0.5sec@32KHz，T0MOD=1 时，T0 中断周期为 60sec@32KHz。
- STOP 模式功能：STOP 模式下，T0 正常工作，并可在 T0ENB=1&T0TB=1@IHRC_RTC 时将系统从 STOP 模式被唤醒，T0 溢出后，TF0 有效，系统被唤醒。



13.1 T0 操作

T0 定时器由 T0ENB 控制。当 T0ENB=0 时，T0 停止工作；当 T0ENB=1 时，T0 开始计数。T0C 溢出 (从 0xFF 到 0x00) 时，TF0 置 1 显示溢出状态并由程序清零。T0 没有双重缓存器，故 T0 溢出时由程序加载新值给 T0C，以选定合适的间隔时间。如果使能 T0 中断 (ET0=1)，T0 溢出后系统执行中断服务程序，在中断下必须由程序清 TF0。T0 可以在 Normal 模式、Idle 模式和 Stop 模式下工作。Stop 模式下，T0 继续计数，设置 TF0I 为 1，T0 溢出时系统被唤醒。



T0 时钟源为 F_{cpu} （指令周期），由 $T0RATE[2:0]$ 控制其分频 $F_{cpu}/2 \sim F_{cpu}/256$ 。T0 长度是 8 位（256 步），一个计数周期是每个输入时钟的周期。

T0RATE[2:0]	T0 时钟	高速模式 ($F_{cpu}=1MIP$)	
		最高溢出时间	单步时间= $max/256$
000	$F_{cpu}/256$	65.563ms	256us
001	$F_{cpu}/128$	32.768ms	128us
010	$F_{cpu}/64$	16.384ms	64us
011	$F_{cpu}/32$	8.192ms	32us
100	$F_{cpu}/16$	4.096ms	16us
101	$F_{cpu}/8$	2.048ms	8us
110	$F_{cpu}/4$	1.024ms	4us
111	$F_{cpu}/2$	0.512ms	2us

13.2 T0 模式寄存器

模式寄存器 T0M 设置 T0 的工作模式，包括 T0 前置分频器、时钟源等，这些设置必须在使能 T0 定时器之前完成。

T0M-T0 定时器模式控制寄存器

地址: **BBH**

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0M	T0ENB	T0RATE[2:0]			--	--	T0MOD	T0TB
R/W	R/W	R/W			--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	T0ENB	T0 使能控制位。 0: 禁止; 1: 使能。
6:4	T0RATE[2:0]	T0 时钟源选择位。 000: Fcpu/256. 100: Fcpu/16. 001: Fcpu/128. 101: Fcpu/8. 010: Fcpu/64. 110: Fcpu/4. 011: Fcpu/32. 111: Fcpu/2.
1	T0MOD	T0 中断模式控制位。 0: T0C 寄存器溢出时 T0 中断; 1: SEC 寄存器溢出时 T0 中断。
0	T0TB	T0 RTC 功能控制位。 0: 禁止 RTC 功能, T0 时钟源为 Fcpu; 1: 使能 RTC 功能, T0 时钟源为低速时钟。

注: 若选择内部低速时钟作为 T0 时钟源, 则在不接外部晶振时配置 T0TB =1 即可;

13.3 T0C 计数寄存器

8 位计数器 T0C 溢出时, TF0 置 1 并由程序清零, 用来控制 T0 的中断间隔时间。必须保证写入正确的值到 T0C 寄存器, 然后使能 T0 定时器以保证第一个周期准确无误。T0 溢出后, 由程序加载一个正确的值到 T0C 寄存器。

T0C-T0 计数寄存器

地址: BCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0C	T0C[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	T0C[7:0]	T0 计数寄存器

T0C 初始值的计算公式如下:

$$\text{T0C 初始值} = 256 - (\text{T0 中断间隔时间} \times \text{T0 时钟 Rate})$$

例: 计算 T0C, T0 的间隔时间为 10ms。

$$\text{T0 时钟源 } F_{\text{cpu}} = 32\text{MHz}/32 = 1\text{MHz}, \text{ T0RATE} = 001 (F_{\text{cpu}}/128)$$

$$\text{T0 间隔时间} = 10\text{ms}, \text{ T0 时钟速率} = 32\text{MHz}/32/128$$

$$\text{T0C 初始值} = 256 - (\text{T0 中断间隔时间} \times \text{输入时钟})$$

$$= 256 - (10\text{ms} \times 32\text{MHz}/32/128)$$

$$= 256 - (10 \times 2 \times 32 \times 106/32/128)$$

$$= \text{B2H}$$

13.4 其他相关寄存器

SEC-T0 秒钟寄存器

地址: BEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEC	--	--	SEC[5:0]					
R/W	--	--	R/W					
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
5:0	SEC[5:0]	T0 SEC 计数器 SEC 计数器范围为 0x00~0x3B(0~59) SEC 的值为 0x3B 且再加 1 时, 计数器溢出并复位为 0x00, 且 MIN 计数器加 1。

MIN-T0 分钟寄存器
地址: BDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MIN	--	--	MIN[5:0]					
R/W	--	--	R/W					
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
5:0	MIN[5:0]	T0 MIN 计数器。 MIN 计数器范围为 0x00H~0x3B (0~59)。 MIN 的值为 0x3B 再加 1 时, 计数器溢出并复位为 0x00。

IEN0-中断使能寄存器 0
地址: A8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EAL	--	ET2	ES0	ET1	EX1	ET0	EX0
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

位号	符号	描述
1	ET0	T0 中断使能控制位。 0: 禁止; 1: 使能。

IRCON-中断请求寄存器
地址: C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCON	ADCF	TCF2	TCF1	TCF0	--	TF0	IE1	IE0
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

位号	符号	描述
2	TF0	T0 中断请求标志位。 0: 无 T0 中断请求; 1: 请求 T0 中断。

14 Buzzer 功能

Buzzer 功能由 BZRENB 位控制，可设置为 GPIO 模式或 Buzzer 模式。Buzzer 通过 P1.5 输出方波信号，50% 占空比，其输出频率通过设置 BZRCKS[1:0] 控制。

BZRM-Buzzer 控制寄存器

地址: D6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZRM	--	--	--	--	--	BZRCKS1	BZRCKS0	BZRENB
R/W	--	--	--	--	--	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
2:1	BZRCKS[1:0]	Buzzer 输出频率。 00: 0.98KHz; 01: 1.96KHz; 10: 3.9KHz; 11: 7.8KHz。
0	BZRENB	Buzzer 输出控制位。 0: GPIO 模式; 1: Buzzer 模式, P15 输出 Buzzer 信号。

15 LCD 驱动

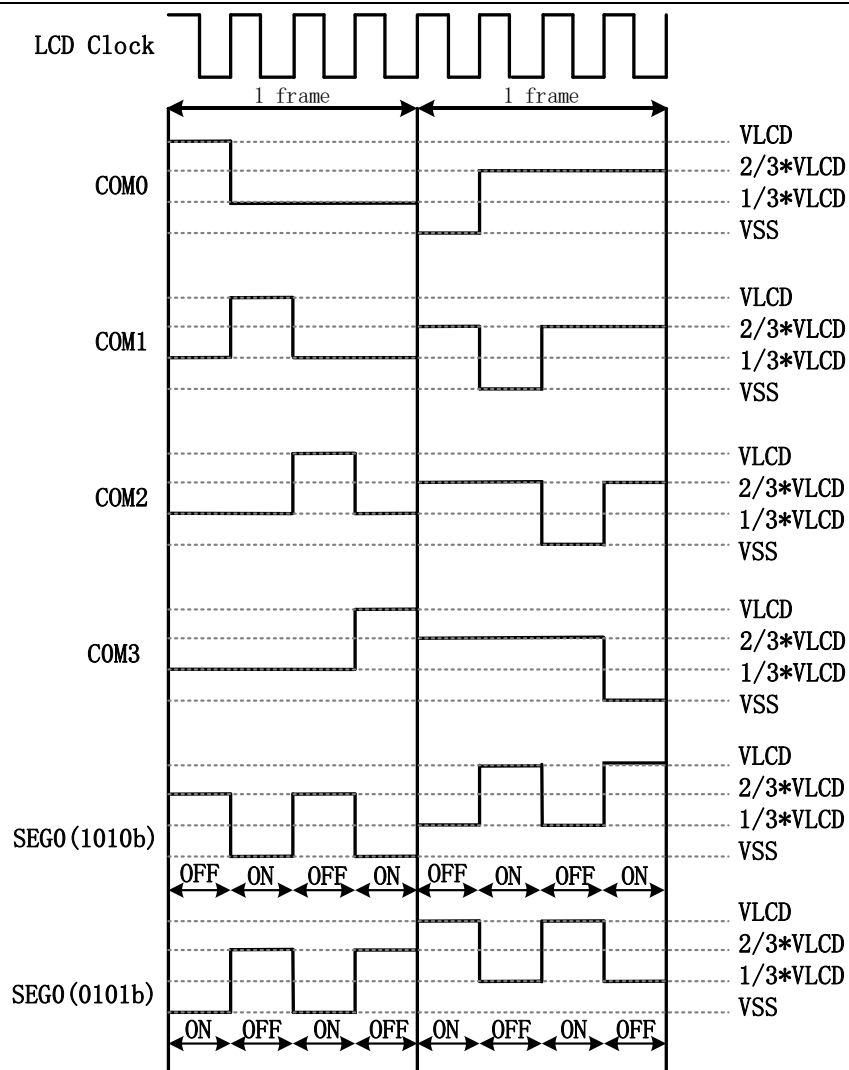
JSC69F007/8/9 LCD 驱动为 C 型结构, 具体为 4x44 或 6x42, LCD 的扫描时序为 1/4 或 1/6 占空比, 1/3 偏压, 具有 176 点或 252 点 LCD 驱动。C 型 LCD 驱动可输出可调节的 VLCD 输出电压和可调节的驱动电源以支持 LCD 面板的多样性。VLCD 引脚与 DVSS 之间必须连接一个 0.1uF 的电容器用于 C 型 LCD 驱动。

LCD 驱动器输出 4 或 6 个 com 波形, 由 LCDMODE 位控制, 输出帧速率由 LCDRATE 位控制, 详见下表所示。

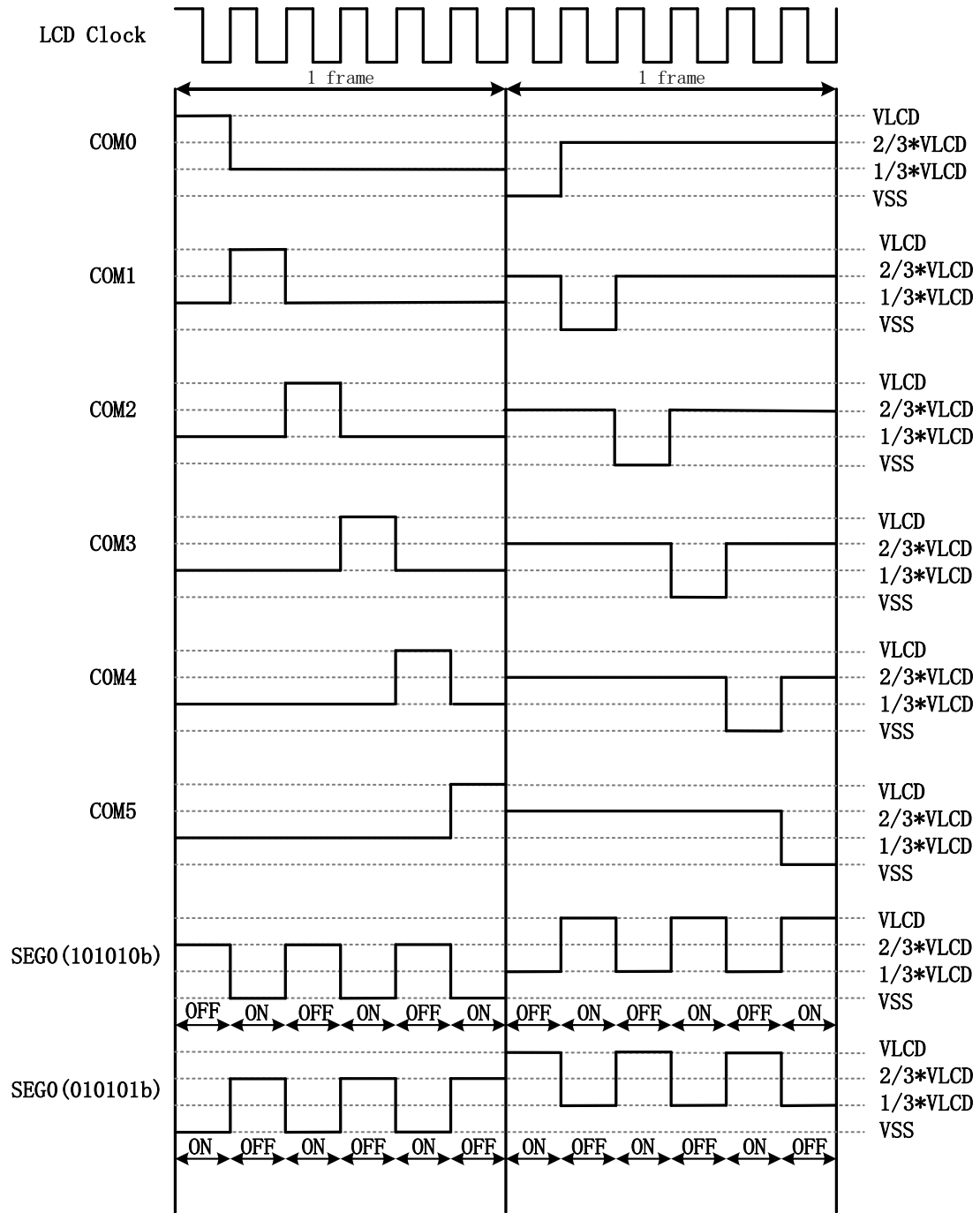
- 启动 LCD: 必须在 BGM(LCDM2.5=0, 即低功耗模式)时使能 LCD, 延时 200ms 后切换至 Normal(BGM=1)模式。
- Stop 模式功能: STOP 模式下 LCDEN=1&STOP=1 时, LCD 功能保持工作状态。
- 低功耗模式功能: 由 BGM 位控制 LCD 低功耗功能。

控制寄存器		LCD 时钟(Hz)	Frame Rate (Hz)	Type
LCDRATE	LCDMODE			
0	0	32KHz/128=256	64	4-COM(1/4 duty)
0	1		43	6-COM(1/6 duty)
1	0	32KHz/64=512	128	4-COM(1/4 duty)
1	1		85	6-COM(1/6 duty)

LCD 驱动波形, 1/4 duty, 1/3 bias:



LCD 驱动波形, 1/4 duty, 1/3 bias:



LCD Drive Waveform, 1/6 duty, 1/3 bias

15.1 LCD RAM 位置

LCD RAM 位于地址 0xF000 到 0xF02B 之间。

LCD RAM 地址与 COM0~COM3 vs SEG0~SEG43 LCD 的关系如下表所示:

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
	COM0	COM1	COM2	COM3	-	-	-	-
SEG0	F000H.0	F000H.1	F000H.2	F000H.3	N/A	N/A	N/A	N/A
SEG1	F001H.0	F001H.1	F001H.2	F001H.3	N/A	N/A	N/A	N/A
SEG2	F002H.0	F002H.1	F002H.2	F002H.3	N/A	N/A	N/A	N/A
SEG3	F003H.0	F003H.1	F003H.2	F003H.3	N/A	N/A	N/A	N/A
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
SEG43	F02BH.0	F02BH.1	F02BH.2	F02BH.3	N/A	N/A	N/A	N/A

LCD RAM 地址与 COM0~COM5 vs SEG2~SEG43 LCD 的关系如下表所示:

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
	COM0	COM1	COM2	COM3	COM4	COM5	-	-
SEG0 (COM4)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SEG1 (COM5)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SEG2	F002H.0	F002H.1	F002H.2	F002H.3	F002H.4	F002H.5	N/A	N/A
SEG3	F003H.0	F003H.1	F003H.2	F003H.3	F003H.4	F003H.5	N/A	N/A
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
SEG43	F02BH.0	F02BH.1	F02BH.2	F02BH.3	F02BH.4	F02BH.5	N/A	N/A

15.2 LCD 控制寄存器

LCDM1-LCD 模式控制寄存器 1

地址: B1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDM1	--	--	--	LCDBNK	LCDMODE	LCDRATE	LCDPEN	LCDEN
R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	LCDBNK	LCD blank 控制位。 0: 正常显示; 1: 关闭所有的 LCD 点。
3	LCDMODE	LCD 驱动 4-COM 或 6-COM 控制位。 0: 4-COM; 1: 6-COM。
2	LCDRATE	LCD 时钟速率控制位。 0: 256Hz; 1: 512Hz。
1	LCDPEN	C-Type LCD pump 使能位。 0: 禁止; 1: 使能。
0	LCDEN	LCD 驱动控制位。 0: 禁止; 1: 使能。

注: 必须先使能 LCD Pump (LCDPEN=1) 并等待 1ms, 然后使能 LCD 驱动 (LCDEN=1)

LCDM2-LCD 模式控制寄存器 2
地址: B2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDM2	-	-	BGM	DUTY1	DUTY0	VCP2	VCP1	VCP0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	1	0	0	0	1	1

位号	符号	描述
5	BGM	LCD 低功耗模式控制位。 0: 使能低功耗模式; 1: Normal 模式。
4:3	DUTY[1:0]	LCD pump 输出驱动能力控制位。 00: 25%; 10: 50%; 01: 37.5%; 11: 100%。 *注: BGM=0 时, DUTY [1:0]可编程控制用于 LCD 低功耗模式。
2:0	VCP[2:0]	VLCD 输出电压控制位。 000: 保留; 011: 3.2V; 001: 2.8V; 100: 3.6V; 010: 3.0V; 101: 4.5V; Others: 保留。 *注: 电池电压必须大于 1.9V。@VLCD 输出电压 2.8~3.6V

P3CON-P3 配置控制寄存器
地址: 9EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3CON	P3CON[7:0]							
R/W	R/W							
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	P3CON[7:0]	P3 功能控制位。 0: 设置为 LCD 功能 (SEG35~SEG28); 1: 设置为 GPIO 功能 (P30~P37)。

P4CON-P4 配置控制寄存器
地址: 9FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4CON	P4CON[7:0]							
R/W	R/W							
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	P4CON[7:0]	P4 功能控制位。 0: 设置为 LCD 功能 (SEG27~SEG20) ; 1: 设置为 GPIO 功能 (P40~P47) 。

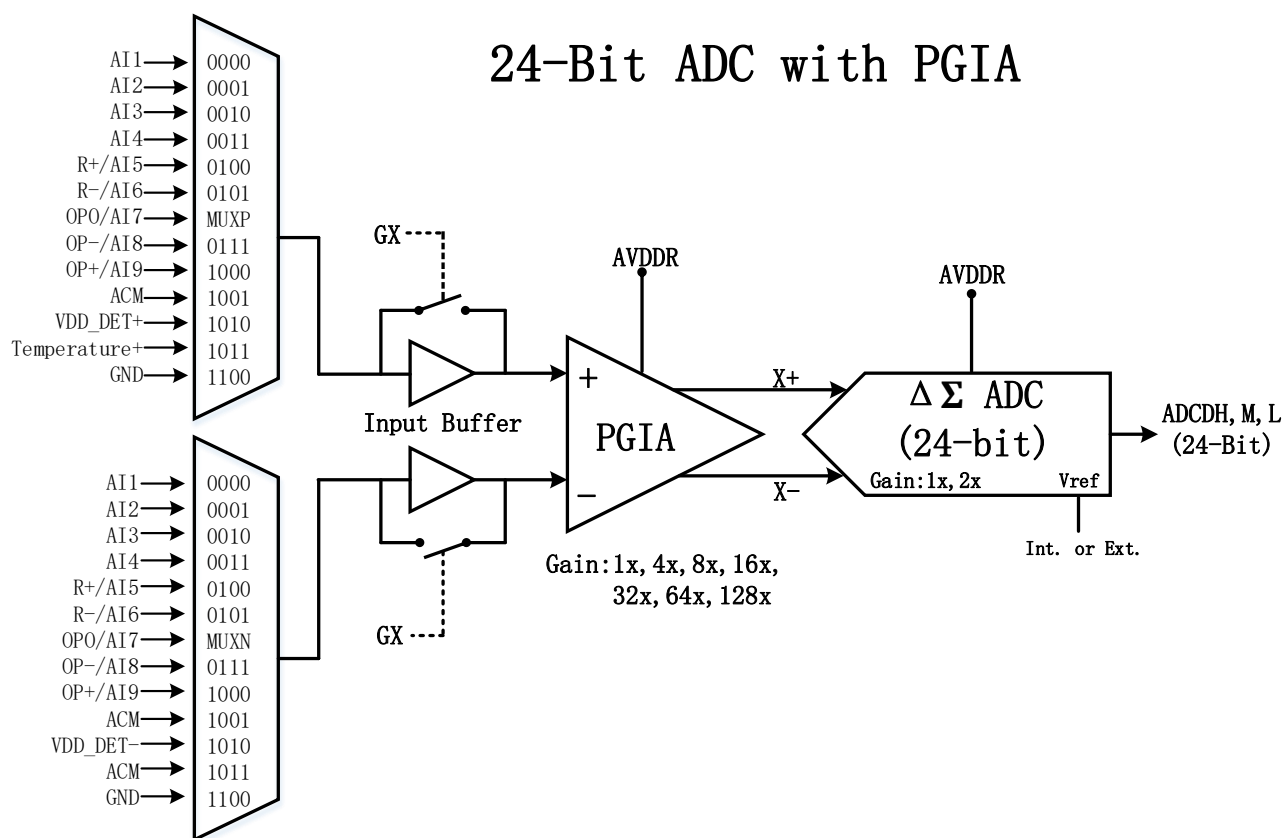
P5CON-P5 配置控制寄存器
地址: 9DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5CON	P5CON[7:0]							
R/W	R/W							
复位值	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	P5CON[7:0]	P5 功能控制位。 0: 设置为 LCD 功能 (SEG43~SEG36) ; 1: 设置为 GPIO 功能 (P50~P57) 。

16 ADC

JSC69F007/8/9 集成高分辨率 $\Delta\Sigma$ ADC，低噪音可编程增益放大器（PGIA），可输出 24 位分辨率，带 22 位精度。ADC 的转换速率范围为 2Hz 到 7.8KHz，ADC 的参考电压由内部或者外部（R+/R-）提供。ADC 可在 NORMAL 模式和 IDLE 模式下正常工作，若使能中断，ADC 结束后可执行中断或者唤醒功能。ADC 内部增益的选择选项为：1x 和 2x。PGIA 通过配置的 2 个多路选通器，MUXP 和 MUXN 输入，提供差分类型的输入和单端类型的输入。优化 ADC 可在传感测量或医学测量中来测量小电平的单端或双端信号。低噪音、高稳定性可编程增益放大器（PGIA）的选择选项为：1x、4x、8x、16x、32x、64x 和 128x，配合 ADC 中适应这些应用。



16.1 ADC 操作设置

在开始 AD 转换之前，必须先完成下面配置的设置。具体设置如下：

- 1、必须使能模拟电源 AVDDR，ACM 和 Bandgap。（通过 AVDDREN、ACMEN 和 BGREN 位设置）
- 2、选择 ADC 输入通道。（通过 MUXP[3:0]和 MUXN[3:0]设置）
- 3、输入缓存器设置：若应用 PGIA 则关闭输入缓存器；若关闭 PGIA 则开启输入缓存器，传感器要求高输入阻抗。（由 GX 位设置）
- 4、如有需要则使能 PGIA，并设置 PGIA 增益。（由 PGIAEN，GS[2:0]位控制）
- 5、开启 Chopper 功能和 PGIA 和 ADC 的时钟。（由 PCHPEN、ACHPEN[1:0]、AMPCKS[1:0] 位设置）
- 6、设置 ADC 参考电压为内部电压或外部 R+/R-。（有 IRVS[2:0]位控制）
- 7、通过设置 ADC 时钟和过采样率，从而设置 ADC 转换速率。（由 ADCKS[1:0]和 OSR[2:0] 位设置）
- 8、如有需要则设置 ADC 中断功能并使能 ADC。（由 EADC 和 ADCEN 位控制）
- 9、使能 ADENB 位后，ADC 准备好进行转换。（24 位 ADCDH、ADCDM、ADCDL）

16.2 ADC 输入通道和 PGIA

ADC 内置 9 个外部通道输入和 4 个内部通道输入，AI1~AI9 用于测量 9 个差分模拟信号源，由通道控制寄存器 CHS 控制。4 个内部通道包括 ACM、GND、VDD_DET（VDD 检测）和温度通道，AI7~AI9 通道与 OPA 输入输出端共用。ADC 外部参考电压通过 AI5 和 AI6 输入。VDD_DET 通道为 $1/8 \times VDD$ 电压输入 ADC，以通过 ADC 测量电池电量。温度传感器嵌入到 IC 内部以测量 IC 周边的温度或粗略测试室温。

JSC69F007/8/9 包括低噪音、高稳定性可编程增益放大器（PGIA），其选择选项为：1x、4x、8x、16x、32x、64x 和 128x，作为差分模式或者单端模式进行输入配置。PGIA 有高输入阻抗特性（约 5G-Ohm），适合连接传感器进行高输入阻抗应用。

模拟输入信号通道选择列表:

MUXP[3:0]	PGIA 正极输入	MUXN[3:0]	PGIA 负极输入
0000	AN1	0000	AN1
0001	AN2	0001	AN2
0010	AN3	0010	AN3
0011	AN4	0011	AN4
0100	AN5/R+	0100	AN5/R+
0101	AN6/R-	0101	AN6/R-
0110	AN7/OP+	0110	AN7/OP+
0111	AN8/OP-	0111	AN8/OP-
1000	AN9/OPO	1000	AN9/OPO
1001	ACM	1001	ACM
1010	VDD_DET+	1010	VDD_DET-
1011	Temperature+	1011	ACM
1100	AVSS	1100	AVSS
Others	NA.	Others	NA.

16.3 模拟输入缓存器

输入缓存器包括 ADC 信号输入缓存器和 ADC 外部参考输入缓存器 R+/R-, 提供一个高阻抗的模拟输入信号, 以减小 ADC 的输入电流, 在进行灵敏测量时以避免负载的影响。PGIA 选择 1x 时, 传感器的输出信号避开 PGIA, 直接连接到 ADC 的输入端。在这种情况下, 必须使能输入缓存器功能 (GX=1)。若 ADC 选择外部 Vref, 输入缓存器 R+/R-也必须使能 (GR=1)。

16.4 ADC 参考电压

ADC 参考电压可选择外部 R+/R-输入或者内部电压源输入, 通过设置 ADCM1 寄存器的 IRVS[2:0]进行选择, 具体见下表所示。

IRVS[2:0]		ADC 参考电压			
		AVDDR 2.7V	AVDDR 3.0V	AVDDR 3.3V	AVDDR 3.6V
0xx	External	R+/R-			
100	0.2*AVDDR	0.54V	0.6V	0.66V	0.72V
101	0.3*AVDDR	0.81V	0.9V	0.99V	1.08V
110	0.4*AVDDR	1.08V	1.2V	1.32V	1.44V
111	0.5*AVDDR	1.35V	reserved	reserved	reserved

16.5 ADC 增益和 ADC 输出码

ADC 内置内部增益选项: x1 和 x2, 用于额外的信号的放大。通过 ADCM1 寄存器的 ADGN 位选择 ADC 的增益。ADC 输出 24 位数据, 由 ADCDH、ADCDM、ADCDL 寄存器组合而成, 带符号位数字格式, 位 ADCB23 为 ADC 数据的符号位。参考下面公式可计算 ADC 的转换数据, ADC 输出码计算如下:

24 位格式, ADC 输出码位于 ADCDH、ADCDM、ADCDL:

$$\frac{\Delta \text{Input} \times \text{PGIA_Gain} \times \text{ADC_Gain}}{\text{Vref}} \times 2^{(24-1)} = +(2^{(24-1)} - 1) \sim -2^{(24-1)}$$

PGIA_Gain: x1~x128

ADC_Gain: x1~x2

Vref: 0.3V~1.5V

16 位格式, ADC 输出码位于 ADCDH 和 ADCDM:

$$\frac{\Delta \text{Input} \times \text{PGIN_Gain} \times \text{ADC_Gain}}{\text{Vref}} \times 2^{(16-1)} = +32767 \sim -32768$$

PGIA_Gain: x1~x128

ADC_Gain: x1~x2

Vref: 0.3V~1.5V

16.6 ADC 中断控制

ADC 完成转换后, 无论是否使能 EADC, ADCF=1。若 EADC 和触发时间 ADCF 都设置为 1, 系统执行中断。若 EADC=0, ADCF=1, 则系统不会执行中断。用户需注意多中断下的操作。

16.7 ADC 转换速率

ADC 提供可变的输出转换速率, 范围为 2Hz~7.8KHz, 由 ADCM2 寄存器的 ADCKS[1:0]和 OSR[2:0]位控制。调节 ADC 时钟 (ADCKS) 和过采样率, 可获得合适的 ADC 输出字节速度。在高分辨率应用下, 过采样率设置建议的最大值 32768。低速的 ADC 转换要比快速的稳定。ADC 应用中, 需要权衡 ADC 的输出速率和有效转换精度 (ENOB)。下表显示 ADC 输出字节速度的设置。

$$\text{ADC 转换速率} = \text{ADC Clock} / \text{OSR}$$

使能 ADC 或者切换输入通道后, ADC 输出在第 3 个数据开始稳定, 第 1 个和第 2 个 ADC 输出数据是不稳定的, 第 3、第 4 和第 5 数据都是稳定的。

ADC 转换速率列表:

ADCKS[1:0]	OSR[2:0]	ADC 时钟	输出速率	ADCKS[1:0]	OSR[2:0]	ADC 时钟	输出速率
00	000	500KHz	7.8kHz	01	000	125KHz	1.95kHz
00	001		3.9kHz	01	001		976Hz
00	010		1.95kHz	01	010		488Hz
00	011		488Hz	01	011		122Hz
00	100		122Hz	01	100		30.5Hz
00	101		61Hz	01	101		15.2Hz
00	110		30.5Hz	01	110		7.6Hz
00	111		15.2Hz	01	111		3.8Hz
01	000	250KHz	3.9kHz	11	000	62.5KHz	976Hz
01	001		1.95kHz	11	001		488Hz
01	010		976Hz	11	010		244Hz
01	011		244Hz	11	011		61Hz
01	100		61Hz	11	100		15.2Hz
01	101		30.5Hz	11	101		7.6Hz
01	110		15.2Hz	11	110		3.8Hz
01	111		7.6Hz	11	111		1.9Hz

16.8 ADC 控制寄存器

CHS-ADC 输入通道选择寄存器

地址: D1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CHS	MUXP[3:0]				MUXN[3:0]			
R/W	R/W				R/W			
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:4	MUXP3:0]	ADC 正极输入通道选择位。 0000: AI1 0101: AI6/R- 1010: VDD_DET+ 0001: AI2 0110: AI7/OP+ 1011: Temperature+ 0010: AI3 0111: AI8/OP- 1100: AVSS 0011: AI4 1000: AI9/OPO others: N/A 0100: AI5/R+ 1001: ACM
3:0	MUXN[3:0]	ADC 负极输入通道选择位。 0000: AI1 0101: AI6/R- 1010: VDD_DET- 0001: AI2 0110: AI7/OP+ 1011: ACM 0010: AI3 0111: AI8/OP- 1100: AVSS 0011: AI4 1000: AI9/OPO others: N/A 0100: AI5/R+ 1001: ACM

AMPM-PGIA 控制寄存器
地址: D3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AMPM	GX	AMPCKS1	AMPCKS0	GS2	GS1	GS0	PCHPEN	PGIAEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	GX	ADC 输入缓存器使能位。 0: 禁止; 1: 使能。(PGIA 增益设置为 x1)
6:5	AMPCKS[1:0]	PGIA chopper 频率控制位。 00: ADCKS/128 10: ADCKS/16 01: ADCKS/32 11: ADCKS/8 (在所有应用中设置为 01)
4:2	GS[2:0]	PGIA 增益选择位。 000: 1x 100: 32x 001: 4x 101: 64x 010: 8x 110: 128x 011: 16x 111: reserved
1	PCHPEN	PGIA chopper 使能位。 0: 禁止; 1: 使能。(使能 PGIA 时必须使能该位)
0	PGIAEN	PGIA 功能使能位。 0: 禁止; 1: 使能。

例: 在应用中设置 PGIA 增益为 1x 时, AI+/AI- 信号绕过 PGIA, 直接输入 ADC。禁止 PGIA (PGIAEN=0) 以省电, 必须使能 ADC 输入缓存器 (GX=1) 用于 ADC 的输入高阻抗特性。

- 1: AMPM |= 0x10; // Enable X+/X- Unit Gain Buffer @GAIN
- 2: AMPM &= 0xE3; // PGIA x 1, GX Buff always set 1.
- 3: AMPM |= 0x01; // Enable PGIA

*注: 当输入缓存器使能 (GX=1 or GR=1), 信号的输入绝对电压范围必须为 0.4V~1.4V。

ADCM1-ADC 控制寄存器 1
地址: D4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCM1	GR	IRVS2	IRVS1	IRVS0	ADGN	ACHPEN1	ACHPEN0	ADCEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	GR	ADC 参考缓存器使能位。 0: 禁止; 1: 使能。(ADC Vref 设置为外部 R+/R-)
6:4	IRVS[2:0]	ADC 参考电压选择位。 0xx: 外部 R+/R-; 100: 内部 0.2 x AVDDR; 101: 内部 0.3 x AVDDR; 110: 内部 0.4 x AVDDR; 111: 内部 0.5 x AVDDR。
3	ADGN	ADC 增益选择位。 0: 1x; 1: 2x。
2:1	ACHPEN[1:0]	ADC chopper 控制位。 11: 使能。(必须在所有应用中设置为 11)
0	ADCEN	ADC 功能使能位。 0: 禁止; 1: 使能。

***例: ADC Vref 设置为外部 R+/R-时, 必须使能 ADC 的 R+/R-输入缓存器 (GR=1) 用于 ADC Vref 输入高阻抗特性。**

- 1: ADCM1 &=0X8F; //EXT Vref
- 2: ADCM1 |= 0x80; // Enable GR Unit Gain Buffer.@EXT Vref
- 3: ADCM1 |= 0x01; // Enable ADC

ADCM2-ADC 控制寄存器 2
地址: D5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCM2	--	--	ADCKS1	ADCKS0	OSR2	OSR1	OSR0	DRDY
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

位号	符号	描述
5:4	ADCKS[1:0]	ADC 时钟选择位。 00: 500KHz; 10: 125KHz; 01: 250KHz; 11: 62.5KHz。
3:1	OSR[2:0]	ADC Over-Sampling-Rate 选择位。 000: 64; 100: 4096; 001: 128; 101: 8192; 010: 256; 110: 16384; 011: 1024; 111: 32768。
0	DRDY	ADC 转换完成位。 0: ADCDH、ADCDM 和 ADCDL 转换数据没有准备好; 1: ADC 输出 (更新) 新的转换数据到 ADCDH、ADCDM 和 ADCDL。

ADCDH-ADC 输出数据高字节
地址: B7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDH	ADCB[23:16]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ADCB[23:16]	ADC 输出高字节

ADCDM-ADC 输出数据中间字节
地址: B6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDM	ADCB[15:8]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ADCB[15:8]	ADC 输出中间字节

ADCDL-ADC 输出数据低字节
地址: B5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDL	ADCB[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ADCB[7:0]	ADC 输出低字节

IEN2-中断使能寄存器 2
地址: 9AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN2	--	--	--	EADC	ETC2	ETC1	ETC0	--
R/W	--	--	--	R/W	R/W	R/W	R/W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	EADC	ADC 中断控制位。 0: 禁止; 1: 使能。

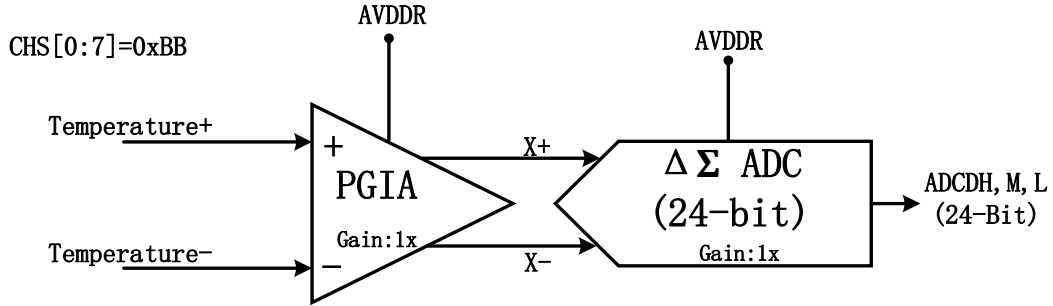
IRCON-中断请求寄存器
地址: C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCON	ADCF	TCF2	TCF1	TCF0	--	TF0	IE1	IE0
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

位号	符号	描述
7	ADCF	ADC 中断请求标志位。 0: 无 ADC 中断请求; 1: 请求 ADC 中断。

16.9 温度传感器(TS)

在应用中，不同的环境温度会使传感器的特性也有所不同，为了得到不同的温度信息，JSC69F007/8/9 内置了一个温度传感器（TS）来测量工作环境温度。通过相对应的 PGIA 通道达到测量环境温度的目的。



在 25°C 的环境下， $V(TS)$ 典型值大约是 $V_{TS\ Offset}$ 。如果温度上升 10°C， $V(TS)$ 就会下降 16mV ($V_{TS}=V_{TS\ Offset}+16mV$)；相反，若温度下降 10°C， $V(TS)$ 则会上升 16mV ($V_{TS}=V_{TS\ Offset}-16mV$)。

温度	(Ts+)-(Ts-)	ADC Vref	ADC 输出(16bit)
15°C	$V_{TS\ Offset}+16mV$	0.81V	ADC Offset+647
25°C	$V_{TS\ Offset}$	0.81V	ADC Offset
35°C	$V_{TS\ Offset}-16mV$	0.81V	ADC Offset-647

*注 1：当选择温度传感器时，PGIA 的增益要选择为 1，否则会出错。

*注 2：这里的温度传感器只是一个参考数据而不是真实的室温，在精确的应用中，请选用外部的温度传感器。

通过 $V(TS)$ ADC 输出，可以得到温度信息和系统补偿。

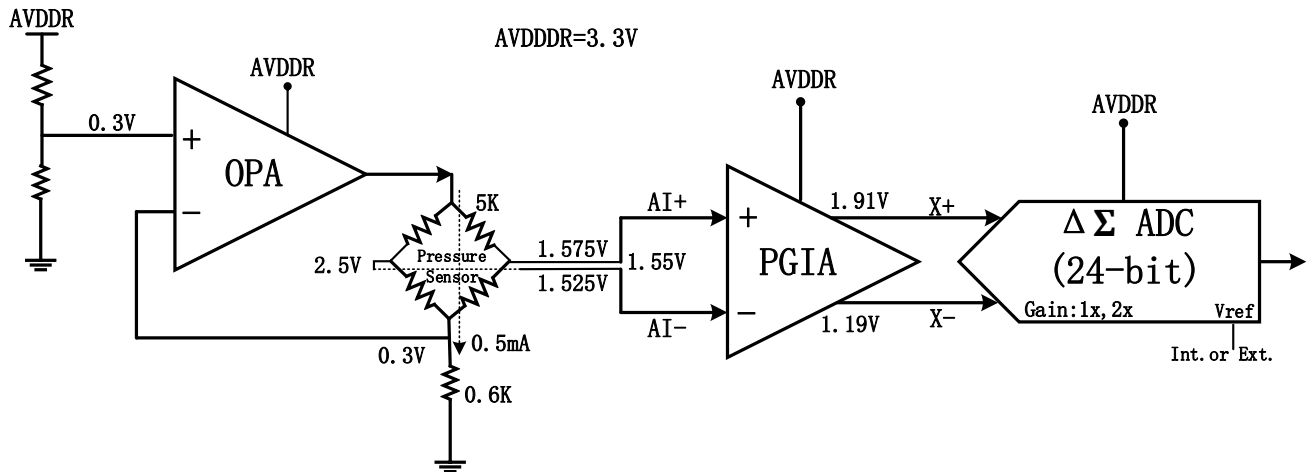
注 1：每颗单片机的 $V(TS)$ 和温度曲线都是有差异的，建议在应用温度传感器时进行室内温度校准。

注 2：温度传感器的典型温度参数是 -1.6mV/C，每颗单片机不是完全相同的。(-64.7CmV/°C)。

SYM	DESCRIPTION	PARAMETER	MIN	TYP	MAX	UNIT
TR	温度传感器范围	AVDDR=3V,VDD=5V	-	-	-	°C
TS	温度传感器灵敏度	AVDDR=3V,VDD=5V	1.4	-1.6	1.8	mV/°C
ETS	温度传感器精度	1 个温度点 25°C 校准	-10		+10	%
		2 个温度点校准	-1		+1	%

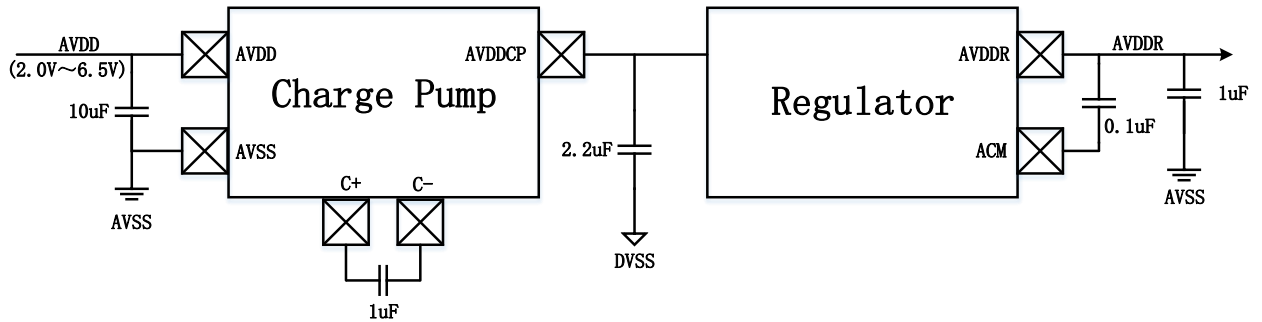
16.10 ADC 应用注意事项

ADC 应用中,ADC 设计要注意 PGIA 输入信号(AI+/AI-)的绝对电压和 ADC 输入信号(X+/X-)的绝对电压。PGIA 的输入电压的范围为 $0V \sim AVDDR-1V$,输出信号的电压的范围为 $0.3V \sim AVDDR-1V$ 。ADC 输入信号(X+/X-)的绝对电压的范围为 $0 \sim AVDDR-1V$ @GXBuff=OFF。应用电路如下图所示。



17 升压稳压器 CPR

JSC69F007/8/9 内置电压升压稳压器 (CPR) 为 ADC、PGIA、OP 和外部传感器提供稳定的电压 AVDDR。升压稳压器设计为高效的升压和降压模式, 在 VDD 的输入范围为 2.0V~5.5V 时应用。可调节 AVDDR 输出电压 2.7V、3.0V、3.3V 和 3.6V, 最大驱动电流为 10mA。另一种 ACM 1V 用于 ADC common 电压, 仅在 sink-type 下工作。



17.1 Charge Pump 寄存器

VREG-电压 Regulator 控制寄存器

地址: D2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VREG	BGREN	AVDDRS1	AVDDRS0	AVDDREN	ACMEN	CPCKS	CPMOD1	CPMOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	BGREN	ADC Band Gap 使能位。 0: 禁止; (系统在 STOP 模式下是必须禁止) 1: 使能。(使能 AVDDR、ADC、PGIA、OPA 和 LBT 功能时必须先使能)
6:5	AVDDRS[1:0]	AVDDR regulator 输出电压控制位。 00: 2.7V; 10: 3.3V; 01: 3.0V; 11: 3.6V。
4	AVDDREN	AVDDR regulator 功能使能位。 0: 禁止; 1: 使能。注: AVDDR 为模拟部分的电压源。
3	ACMEN	ACM regulator 功能使能位。 0: 禁止; 1: 使能。(使能 ADC 时必须使能)
2	CPCKS	Charge pump 时钟控制位。 0: 500KHz; (请设置为 500KHz) 1: 1MHz。
1:0	CPMOD[1:0]	Charge pump 模式控制位。 00: 禁止; 01: Buck-Boost 模式; 10: 保留; 11: 2x Pump 模式。

下表为 Pump 的工作模式:

CPMOD[1:0]	Charge Pump 模式	Pump Behavior
00	Disable Mode	VDD 输入范围为 2.0V~5.5V - 禁止 Pump - AVDDCP=VDD
01	Buck-Boost Mode	VDD 输入范围为 2.0V~5.5V - 使能 Pump - AVDDCP=AVDDR+0.5V - AVDDCP=3.2V (禁止 AVDDR)
10	保留	-
11	2x Pump Mode	VDD 输入范围为 3.6V~5.5V - 禁止 Pump (VDD>3.6V 时必须禁止) - AVDDCP=VDD VDD 输入范围为 2.0V~3.4V - 使能 Pump - AVDDCP=2x VDD (使能或禁止 AVDDR)

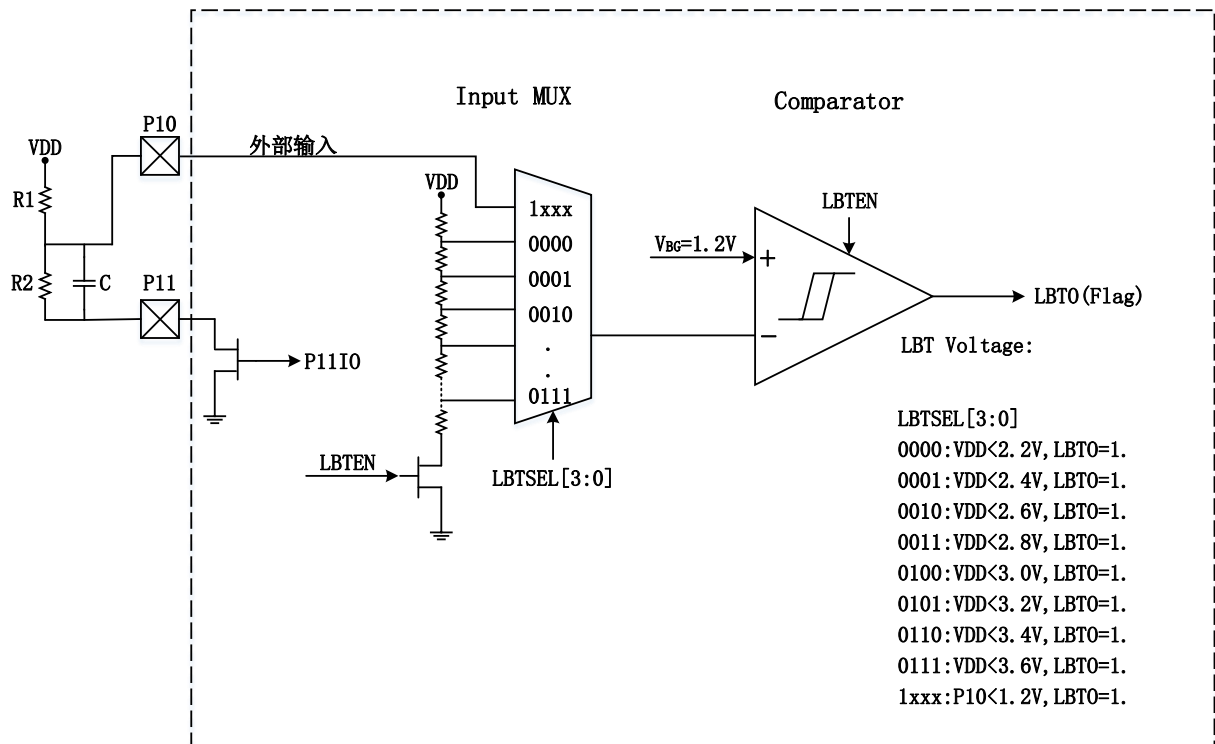
VDD: Pump 输入电压
AVDDCP: Pump 输出电压
Pump 模拟功能的使能流程:
1、使能 BandGap; 2、使能 Pump; 3、延时 1ms 等待 Pump 输出稳定; 4、使能 AVDDR/ACM;
5、延时 0.5ms; 6、使能 ADC、PGIA、OPA。

17.2 Charge Pump Regulator 设置

VDD 输入范围	Charge Pump	AVDDCP	AVDDR	Note
5.5V~4.0V	禁止	无电容 CAVDDCP 和 CPUMP AVDDCP 与 AVDD 短路	接 1uF 电容到 AVSS 2.7V、3V、3.3V、3.6V 有效	无双倍功耗
5.5V~3.3V	禁止		接 1uF 电容到 AVSS 2.7V、3V 有效	
5.5V~3.0V	禁止		接 1uF 电容到 AVSS 2.7V 有效	
5.5V~2.0V	使能	接 2.2uF 电容 CAVDDCP 到 DVSS (接 1uF 电容 CPUMP 到 C+/C-)	接 1uF 电容到 AVSS 2.7V、3V、3.3V、3.6V 有效	AVDDR 消耗双倍功耗

18 比较器

JSC69F007/8/9 内置比较器功能：正极输入电压大于负极输入电压时，比较器输出高电平（LBTO=1）；正极输入电压小于负极输入电压时，比较器输出低电平（LBTO=0）。比较器的正极电压始终为内部 1.2V，负极电压由外部 P10 和 VDD 分压控制。针对不同的应用，比较器提供标志显示（LBTO），类似于低电压检测。内部 VDD 分压后作为比较器的负极输入电压，共有 8 个电压点，由 LBTSEL[2:0]寄存器控制，其范围为 2.1V~3.5V，每次 0.2V 一个电压点。



18.1 比较器控制寄存器

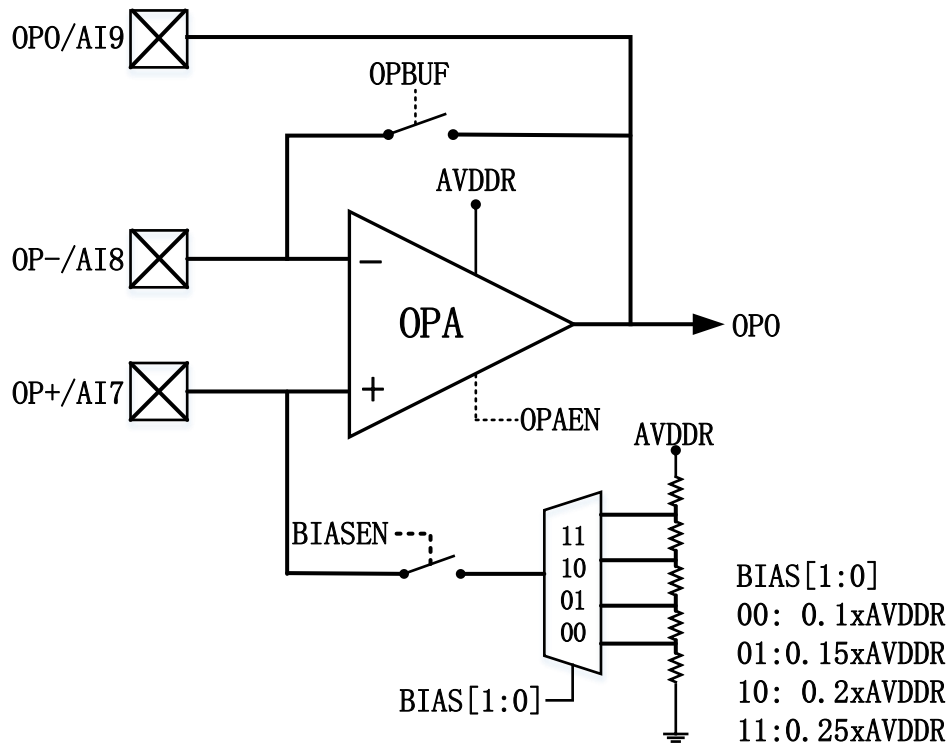
LBTM-比较器和电池低电压检测控制寄存器
地址: D7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LBTM	--	P11IO	LBTSEL3	LBTSEL2	LBTSEL1	LBTSEL0	LBT0	LBTEN
R/W	--	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	0	0	0

位号	符号	描述
6	P11IO	P1.1 配置位。 0: P1.1 为 GPIO 模式; 1: 设置 P1.1 LBT 功能, P1.1 内部连接到地。
5	LBTSEL3	比较器负极输入原选择位。 0: P1.0 为 GPIO 模式; 1: 外部 VDD 电压分频。
4:2	LBTSEL[2:0]	内部 VDD 电压分频控制位。 000: VDD<2.2V, LBTO=1; 001: VDD<2.4V, LBTO=1; 010: VDD<2.6V, LBTO=1; 011: VDD<2.8V, LBTO=1; 100: VDD<3.0V, LBTO=1; 101: VDD<3.2V, LBTO=1; 110: VDD<3.4V, LBTO=1; 111: VDD<3.6V, LBTO=1。
1	LBTO	比较器输出标志位。 0: 比较器负极输入电压大于 1.2V; 1: 比较器负极输入电压小于 1.2V。
0	LBTEN	比较器功能使能位。 0: 禁止; 1: 使能。

19 OPA

JSC69F007/8/9 内置 1 个可操作的放大器 OPA，OP-Amp 的电源由 AVDDR 提供，输入输出信号的电压范围为 $0V \sim AVDDR - 0.1V$ 。OP-Amp 的输入输出引脚与 ADC 输入通道引脚(AI7~AI9)共用。当 OPBUF=0, BIASEN=0 时，OP 进行两个输入的比较。当 OPBUF=1 时，OP_Amp 可设置为缓存器模式。当 BIASEN=1 时，OPA 的正极接内部 AVDDR 分压，可进行端口输入与内部设定的 4 个内置电压可进行比较。OPA 的各种功能可以由 BIAS[1:0]和 BIASEN 控制。



19.1 OP-Amp 控制寄存器

OPM-OP-AMP 控制寄存器

地址: 9BH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPM	--	--	--	BIAS1	BIAS0	BIASEN	OPBUF	OPAEN
R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4:3	BIAS[1:0]	偏置电压输出控制位。 00: $0.1 \times AVDD$; 01: $0.15 \times AVDD$; 10: $0.2 \times AVDD$; 11: $0.25 \times AVDD$ 。
2	BIASEN	Bias 功能使能位。 0: 禁止; 1: 使能, Vbias 输出连接到 OP+。
1	OPBUF	OPA 缓存器模式控制位。 0: 禁止; 1: 使能。OPO 与 OP-由 MOS-SW 短接。
0	OPAEN	OPA 功能使能位。 0: 禁止; 1: 使能。

20 UART

20.1 概述

串行端口为全双工的，它可以同时发送和接收。接收模块有接收缓存，意味着在接收前一个数据被读取之前，串口可以同时接收第二个数据。接收和发送都是对 **SBUF** 进行操作访问。写入 **SBUF** 数据将直接传到发送寄存器。写入 **SBUF** 数据加载到发送寄存器，而读取 **SBUF** 是访问一个具有独立物理地址的接收寄存器。

UART（通用同步异步收发器）提供 **1MHz** 灵活的全双工传输速率。**UART** 共有 **4** 种操作模式：一种同步，三种异步的。同步模式发送 **8** 位数据，没有起始/停止位；其它模式则分别支持 **8** 位和 **9** 位数据的发送，包含起始/停止位。模式 **2** 和 **3** 具有多处理器通信的特殊功能，通过将 **S0CON** 寄存器中的 **SM20** 位置 **1** 来使能该功能。主机处理器首先发送识别目标从机的地址字节。地址字节与数据字节不同，因为地址字节中第 **9** 位为 **1**，数据字节为 **0**。在 **SM20=1** 时，从机不会被数据字节中断。地址字节将中断所有从机。寻址的从机将清除其 **SM20** 位，并准备接收将要来的数据字节。未被寻址的从机使 **SM20** 置 **1** 并忽略传入的数据。

※由 **P0.4** 输出 **UTX** 信号，**P0.5** 接收 **URX** 信号。端口 **P04**、**P05** 不需要进行设置，在选择 **UART** 功能时会自动设置。

20.2 UART 模式

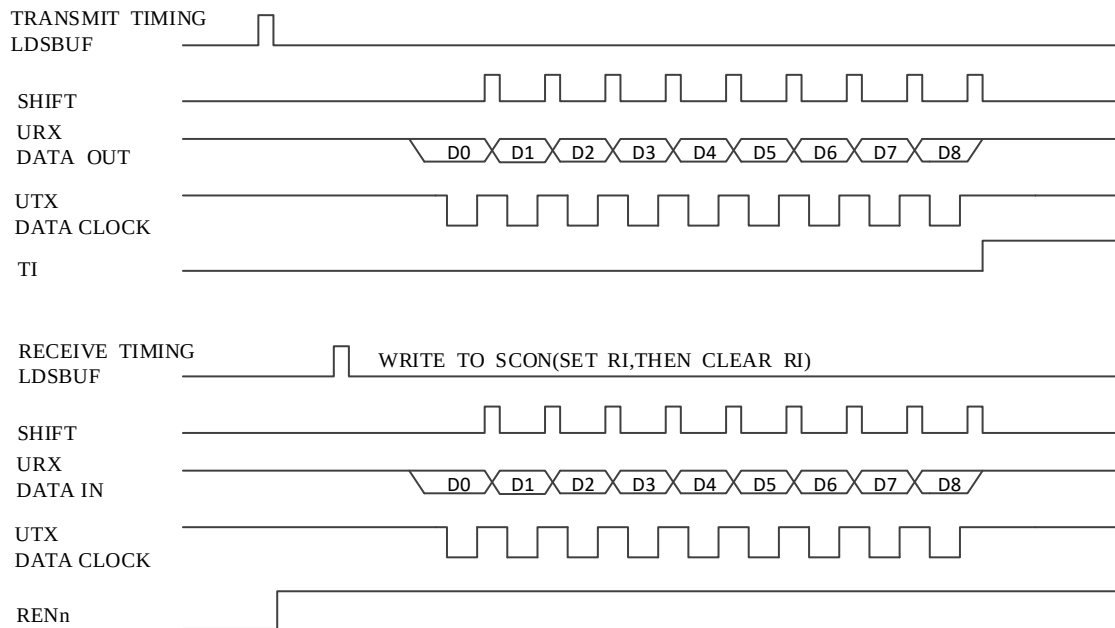
20.2.1 UART 模式 0：同步 8 位收发器

模式 **0** 下，**UART** 发送/接收 **8** 位长度的数据，没有起始/停止位。总线的首位为最低有效位 **LSB**，波特率固定为 $F_{cpu}/12$ 。接收数据时，设置 **REN0** 位为 **1**，先设置 **RI0** 为 **1**，然后将 **RI0** 位清零开始接收，接收 **8** 位数据后接收标志 **RI0** 置 **1**；发送时，先 **REN0** 位使能，然后写入数据到 **S0BUF** 开始发送。此时数据从 **URX** 送出，直到 **8** 位数据传输完成，完成后 **TI0** 标志位置 **1**。

模式 **0** 是与外部设备进行同步通信的方式。在该模式下，串行数据由 **URX** 脚进行收发，而 **UTX** 脚用于产生移位时钟。这种方式下是以半双工的形式进行通信，每帧接收或发送 **8** 位数据。数据的最低位被最先发送或接收。无论发送或接收数据，串行时钟将一直由 **MCU** 产生，因此串口模式 **0** 为主机模式。

下图是模式 0 时序图:

图 20-1 串口模式 0 时序图



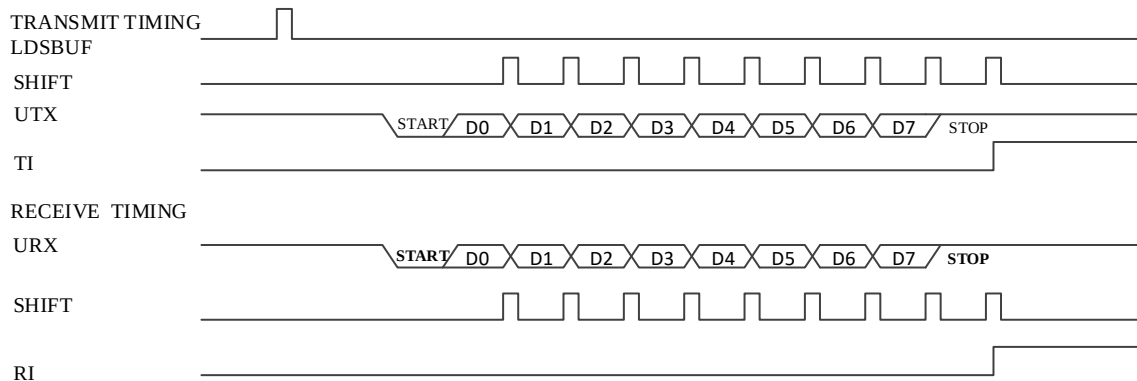
20.2.2 UART 模式 1: 异步 8 位收发器

模式 1 为异步全双工的方式, UART 操作为典型的格式协议: 包括起始位, 8 位数据位和停止位。波特率由 S0RELH/S0RELL 寄存器, 或取决于 BD 寄存器的 TC2 的溢出周期控制。

通过写入数据到 S0BUF 寄存器开始发送, 发送的首位是起始位 (始终为 0), 然后是 S0BUF 的数据 (首先 LSB), 发送完停止位 (始终为 1) 后, 有通知/中断时自动将 TI0 位设置为 1。

若使能接收功能 (REN0=1), UART 从主机同步起始位。把数据位的第一位当成最低有效位 (LSB), 接收完成后, S0BUR 会进行更新。

图 20-2 串口模式 1 时序图



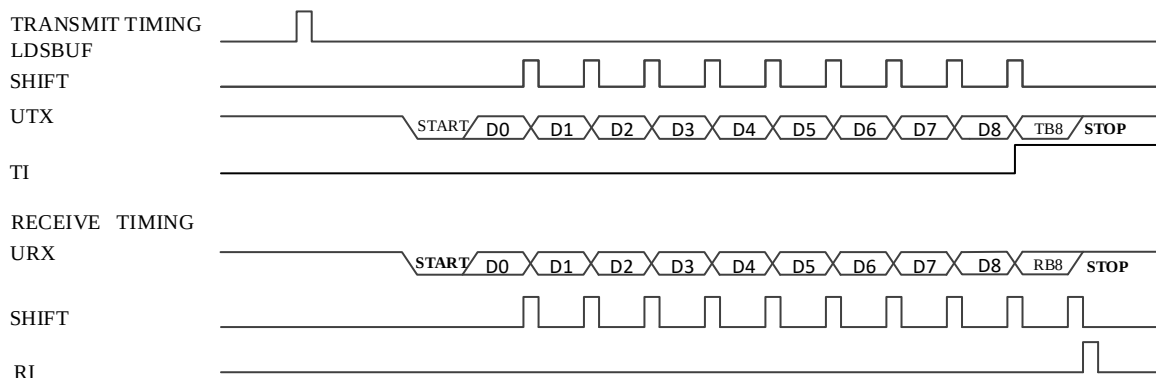
20.2.3 UART 模式 2/3: 异步 9 位收发器

模式 2 和模式 3 都是全双工异步通信, 包含起始位、9 位数据位和停止位。模式 2 的波特率只有 2 个选项: $F_{cpu}/32$ 和 $F_{cpu}/64$; 而模式 3 的波特率可通过 S0RELH/S0RELL 寄存器或 TC2 的溢出周期来控制。

第 9 位数据位 (在 S0BUF 寄存器的最高位传输完毕之后传输) 可通过写入 TB80 寄存器来传送, 接收后在 RB80 中进行读取。也可以一直设置第 9bit 数据为 1, 表示每次发送 2 个 STOP 信号, 也可以将第 9bit 数据作为奇偶校验位。

第 9bit 数据也经常使用来自定义为一对多的通信协议, 当 SM20 寄存器设置为 1 时, 仅仅收到的第 9bit 数据为 1 时, 才会产生接收中断。利用这个功能就可以实现一主多从的通信协议。所有的从机都设置 SM20 为 1, 主机端先发送需要通信的从机地址, 同时设置第 9bit 为 1, 这样所有的从机都会产生接收中断。从机判断接收到的地址是否为发给自己的。如果地址匹配, 则对应的从机就将 SM20 清 0, 地址不匹配的从机, 保持 SM20 为 1。同时主机再发送后续的数据时, 同时将第 9bit 数据设置为 0 再发出, 那么其他从机就不会再产生接收中断。

图 20-3 串口模式 2、3 时序图



20.3 UART 波特率

UART 模式 0 的波特率是固定的, 为 $F_{cpu}/12$; 模式 2 有 2 个波特率选项, 由 PCON 寄存器的 SMOD 位选择控制, 其选项分别为 $F_{cpu}/32$ (SMOD=1) 和 $F_{cpu}/64$ (SMOD=0)。

UART 模式 1 和模式 3 的波特率由 S0RELH/S0RELL 寄存器 (BD=1) 或者 TC2 溢出周期 (BD=0) 产生。通过设置 SMOD 位可以将波特率增加 1 倍。

在模式 1 和模式 3 中若选择 S0RELH/S0RELL (BD=1), 则波特率由下面公式产生:

$$\text{Baud Rate} = 2^{\text{SMOD}} \times \frac{F_{cpu}}{64 \times (1024 - \{S0RELH, S0RELL\})}$$

普通 UART 波特率的设置建议如下表设置 ($F_{cpu}=32\text{MHz}$):

Baud Rate	SMOD	S0RELH	S0RELL	Accuracy
4800bps	1	0x03	0x30	-0.16%
9600bps	1	0x03	0x98	-0.16%
19200bps	1	0x03	0xCC	-0.16%
38400bps	1	0x03	0XE6	-0.16%
56000bps	1	0x03	0xEE	0.79%
57600bps	1	0x03	0xEF	-2.12%
115200bps	1	0x03	0xF7	3.55%
128kbps	1	0x03	0xF8	2.34%
250kbps	1	0x03	0xFC	0%
500kbps	1	0x03	0xFE	0%
1Mbps	1	0x03	0xFF	0%

在模式 1 和模式 3 中若选择 TC2 溢出周期（BD=0），则波特率由下面公式产生。TC2 必须为 16 位自动重装模式，这样才能产生周期性的溢出信号。

$$\text{Baud Rate} = 2^{\text{SMOD}} \times \frac{F_{\text{cpu}}}{32 \times \text{Timer Period}}$$

TC2 溢出周期普通 UART 波特率的设置建议如下表设置（Fcpu=32MHz）：

Baud Rate	SMOD	Timer Period	TC2CH/L	TC2RH/L	Accuracy
4800bps	1	13.021us	0xFE5F	0xFE5F	-0.08%
9600bps	1	6.510us	0xFF30	0xFF30	0.16%
19200bps	1	3.255us	0xFF98	0xFF98	0.16%
38400bps	1	1.628us	0xFFCC	0xFFCC	0.16%
56000bps	1	1.116us	0xFFDC	0xFFDC	-0.80%
57600bps	1	1.085us	0xFFDD	0xFFDD	-0.80%
115200bps	1	0.543us	0xFFEF	0xFFEF	2.08%
128kbps	1	0.488us	0xFFFF0	0xFFFF0	-2.40%
250kbps	1	0.250us	0xFFF8	0xFFF8	0%
500kbps	1	0.125us	0xFFFC	0xFFFC	0%

* 注：系统时钟 Fcpu 最小设置值 Fcpu<Timer Period 值，@PWSC[2: 0]=001

另：如 TC2 定时器溢出期是 6.510us，系统时钟 Fcpu 可以设置 32m/64 或 32m/128

20.4 UART 寄存器

S0CON-UART 控制寄存器

地址: 98H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	0	0	0	0	0	0

位号	符号	描述
7:6	SM[0:1]	UART 模式选择位。 00: 模式 0; 01: 模式 1; 10: 模式 2; 11: 模式 3。
5	SM20	多重处理器通讯控制位 (模式 2、3)。 0: 禁止; 1: 使能。
4	REN0	UART 接收功能控制位。 0: 禁止; 1: 使能。 *注: 使用模式 0 时, 无论接收发送都要使能 REN0。
3	TB80	发送数据的第 9 位 (模式 2、3)。
2	RB80	接收数据的第 9 位。
1	TI0	发送 UART 的中断标志。
0	RI0	接收 UART 的中断标志。

S0CON2-串口波特率选择寄存器

地址: D8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON2	BD	--	--	--	--	--	--	--
R/W	R/W	--	--	--	--	--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	BD	波特率产生器选择位 (模式 1、3)。 0: TC2 溢出周期; 1: 由寄存器 S0REXH/S0RELL 控制。

S0BUF-UART 数据缓存器
地址: 99H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0BUF	S0BUF[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	S0BUF[7:0]	写入数据的操作触发 UART 通讯（首先 LSB），可在 package 结束时读取接收到的数据。

PCON-电源控制寄存器
地址: 87H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	--	--	--	P2SEL	GF0	STOP	IDLE
R/W	R/W	--	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	SMOD	UART 波特率加倍使能 UART 在模式 1、2、3 时，设置此位，波特率会加倍 0: 波特率不加倍 1: 波特率加倍

S0RELH-UART 重装高字节
地址: BAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0RELH	--	--	--	--	--	--	S0REL9	S0REL8
R/W	--	--	--	--	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
1:0	S0REL[9:8]	UART 重装寄存器，与 S0RELL 共同设置产生 UART 波特率。

S0RELL-UART 重装低字节
地址: AAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0RELL	S0REL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	S0REL[7:0]	UART 重装寄存器, 与 S0RELH 共同设置产生 UART 波特率。

IEN0-中断使能寄存器 0
地址: A8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EAL	--	ET2	ES0	ET1	EX1	ET0	EX0
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

位号	符号	描述
7	EAL	使能中断, 请参考中断章节。
4	ES0	使能 UART 中断。

21 SPI

21.1 SPI 概述

此 SPI 是一个完全可配置的 SPI 主机/从机设备，允许用户配置串行时钟信号 SCLK 的极性和相位。SPI 允许 MCU 与串行的外围设备进行通信。串行时钟线（SCLK）与两个独立串行数据线上信息的移位和采样同步。SPI 数据同时发送和接收。

SPI 系统具有足够的灵活性，可以与多家制造商的许多标准产品外设直接连接。系统可以配置为主机设备或从机设备。为了适应大多数可用的同步串行外设，时钟控制逻辑允许选择时钟极性，并选择两种不同的时钟协议。当 SPI 被配置为主机设备时，串行时钟可在七种不同的比特率中选择其一。

SPI 的特性如下：

- ◆ 全双工同步串行数据传输
- ◆ 支持主/从机模式
- ◆ 七个 SPI 主控波特率：系统时钟的 1/2、1/4、1/8、1/16、1/32、1/64、1/128
- ◆ 从机时钟频率高达 $F_{sys}/8$
- ◆ 具有可编程极性和相位的串行时钟
- ◆ 产生中断
- ◆ 写入碰撞标志保护
- ◆ 8 位数据，先传输最高有效位（MSB），后传输最低有效位（LSB）
- ◆ 特殊功能寄存器连接主机 CPU
- ◆ 无双向端口，标准 SPI 引脚外部连接到三态缓冲器
- ◆ 支持四种传输格式
- ◆ 简单接口允许轻松连接到微控制器

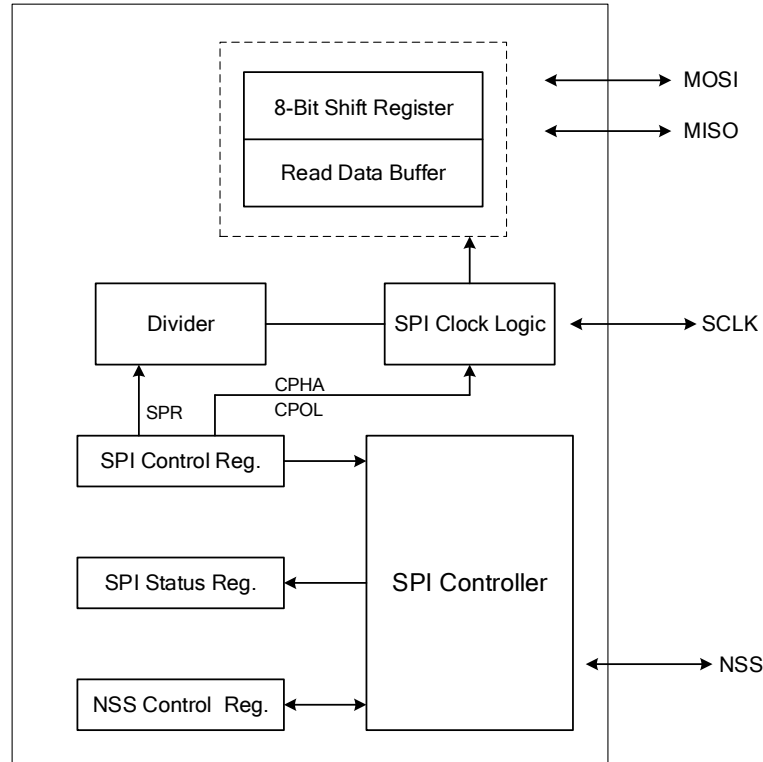
21.2 SPI 端口配置

使用 SPI 功能必须先配置 SPI 相关寄存器，然后再将相应端口配置成 SPI 通道，P14 为 SCK 通道，P13 为 MISO 通道，P12 为 MOSI 通道，P11 为 NSS 通道。详见 SPI 寄存器部分描述。

21.3 SPI 硬件描述

SPI 系统中的中心元件是包含移位寄存器和读取数据缓冲区的模块。系统在发送方向上为单缓冲器，在接收方向上为双缓冲器。这意味着直到前一个数据传输完，新的数据才能写入移位器；然而，接收的数据被传送到并行读取数据缓冲器中，所以移位器可以自由地接收第二串行字符。只要在下一个串行字符准备好传输之前，第一个字符从读取数据缓存器中读出，就不会发生覆盖状态。

图 21-1 SPI 框图



与 SPI 相关联的引脚有：NSS、SCLK、MOSI、MISO。

主机模式下的 **NSS** 输入引脚用于检测模式故障错误。此引脚上的低电平表示多主机系统中的其他设备已成为主机设备，并尝试将 SPI 选择为从机设备。从机模式下的 **NSS** 输入引脚用于使能传输。

在主机模式下，**SCLK** 引脚用作 SPI 时钟信号参考。当主机设备启动传输时，**SCLK** 引脚上会自动生成八个时钟周期。

- 当 SPI 被配置为从机设备时，**MOSI** 引脚是从机设备输入数据线，**MISO** 是从机设备输出数据线。
- 当 SPI 被配置为主机设备时，**MISO** 引脚是主机设备输入数据线，**MOSI** 是主机设备输出数据线。

21.4 SPI 时钟分频

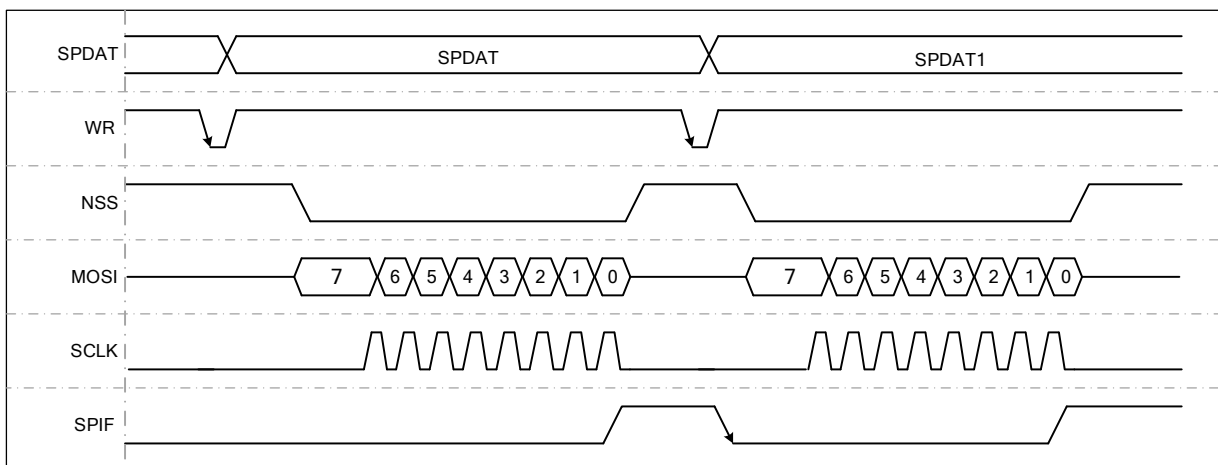
SM2-SM0 控制 SPI 时钟分频

SM2	SM1	SM0	System clock divided by
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128

21.5 SPI 主控模式

SPI 控制模式共有 7 种类型的时钟发生器，从 $F_{sys}/2 \sim F_{sys}/128$ ，产生的时钟由 SCK 引脚输出。当 SPI 配置为主机模式时，SPI 通过 NSS 线选择从机设备，通过写入 SPDAT 启动传输。当新字节写入 SPDAT 时，SPI 开始传输。串行时钟 SCLK 由 SPI 生成，在主机模式下 SPI 使能，SCLK 输出。

图 21-1 自动从机选择线使用



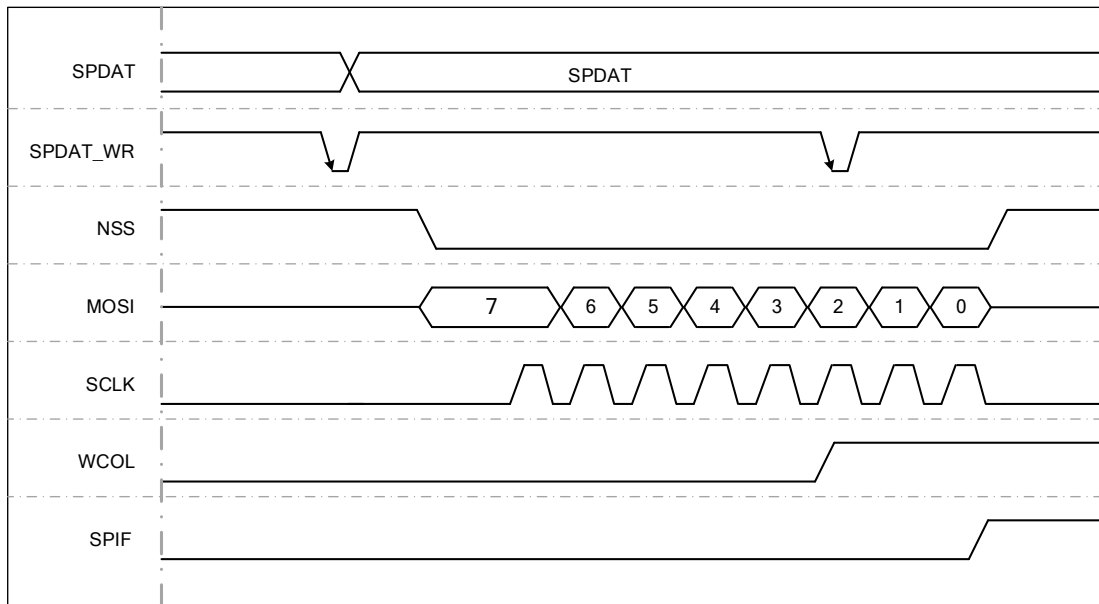
21.5.1 写冲突

如果在传输过程中写入了 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，导致错误的写入数据不会写入移位器。写冲突由 SPSTA 寄存器中的 WCOL 标志指示。

当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。如果要清除 WCOL 位，用户应执行以下步骤：

- 读取 SPSTA 寄存器
- 访问 SPDAT 寄存器

图 21-2 SPI 主模式下的写冲突错误



写冲突的具体条件为：在数据传输过程中，NSS 为低时，第一个数据开始发送时刻到第八个 SCLK 下降沿期间。如果在此期间写 SPDAT，则会出现写冲突，WCOL 置 1。

21.6 SPI 从模式

设定 MSTR 为 0，SPI 将工作在从机模式。当作为从机模式时，SPCLK 管脚变为输入脚，NSS 管脚也变为输入脚。当从机设备的 NSS 管脚不为低时，主机设备不能与从机交换数据

在从机模式下，数据通过 MOSI 管脚从主机向从机传输，通过 MISO 管脚从从机向主机传输。通过主机 SPCLK 的时钟控制，数据进入位移寄存器。在移位寄存器接收到一个字节后，数据将移到读数据缓存，同时 SPIF 置 1。

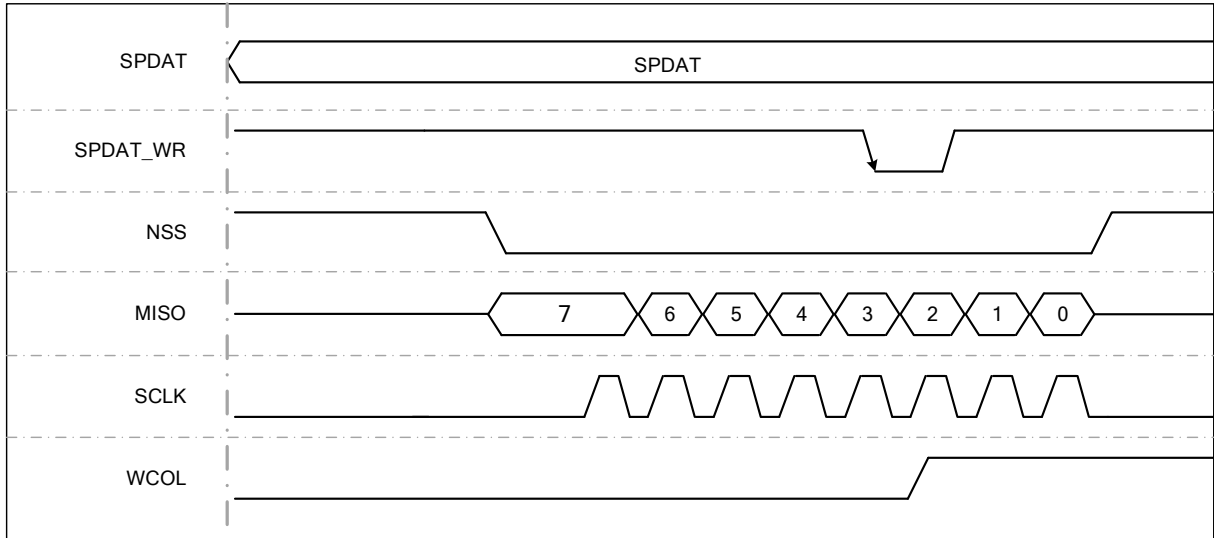
21.6.1 写冲突

如果在传输过程中写入 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，并且导致错误的写入数据不会写入移位器。写冲突由 SPSTA 寄存器中的 WCOL 标志指示。

当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。要清除 WCOL 位，用户应执行以下顺序：

- 读取 SPSTA 寄存器的内容
- 访问 SPDAT 寄存器（读或写）

图 21-3 SPI 从模式下的写冲突错误



21.7 SPI 时钟控制逻辑

21.7.1 SPI 时钟相位与极性控制

软件可以在 SPI 控制寄存器(SPCON) 中选择使用两个控制位的(串行时 SCLK 的相位和极性)四种组合中的任何一种。时钟极性由 CPOL 控制位指定，CPOL 控制位选择高电平或低电平有效时钟。时钟相位(CPHA) 控制位选择两种基本不同的传输格式中的一种。主机 SPI 设备和通信从机设备的时钟相位和极性应相同。

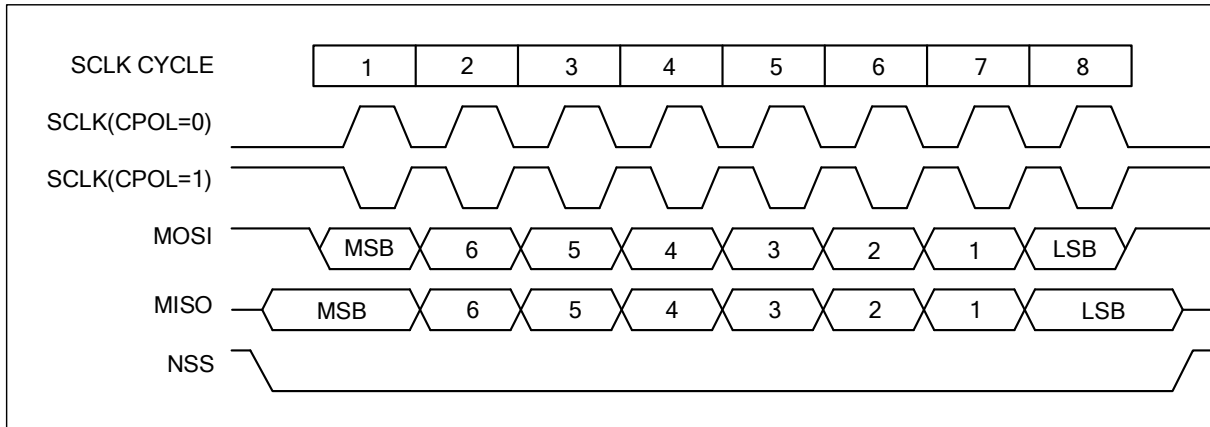
21.7.2 SPI 传输格式

在 SPI 传输期间，数据同时发送 (串行移出)并接收 (串行移入)。串行时钟线与两条串行数据线移位和采样同步。从机选择线允许单独选择从机 SPI 设备：未选择的从机设备不会干扰 SPI 总线活动。在 SPI 主机设备上，从机选择线可以有选择性的用于指示多主机总线竞争。

21.7.3 CPHA=0 传输格式

下图显示了 CPHA 为 0 的 SPI 传输的时序图。SCLK 显示两个波形：一个用于 CPOL 等于 0，另一个用于 CPOL 等于 1。通过 SCLK 该图可以描述为主机设备或从机设备时序图，主进/从出(MISO)和主出/从进(MOSI)引脚直接连接在主机和从机之间。MISO 信号是从机输出，MOSI 信号是主机输出。NSS 线是从机的选择输入；主机的 NSS 引脚未显示，但假定为无效。主机的 NSS 引脚必须为高电平。该时序图在功能上描述了如何进行传输。

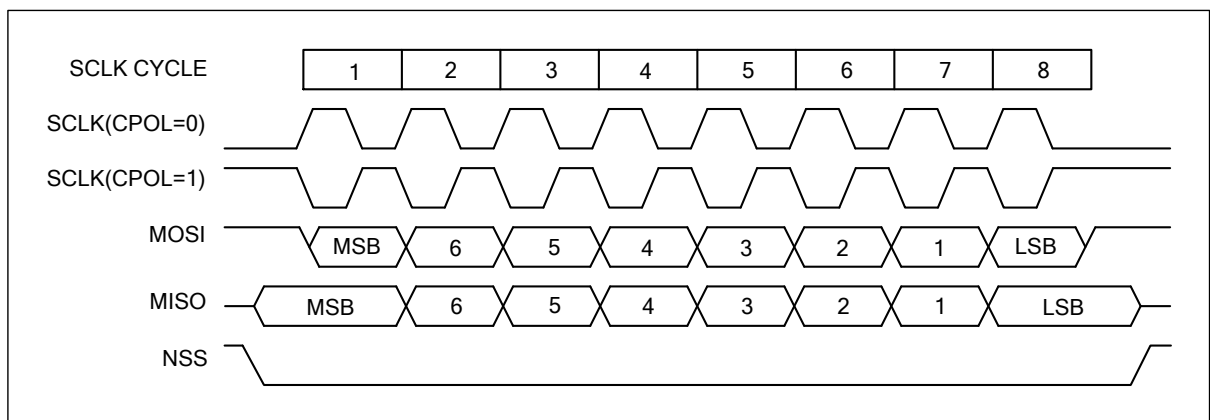
图 21-4 CPHA=0 SPI 传输格式



21.7.4 CPHA=1 传输格式

下图是 CPHA=1 的 SPI 传输的时序图。SCLK 显示两个波形：一个用于 CPOL=0，另一个用于 CPOL=1。由于 SCLK、MISO 和 MOSI 引脚直接连接在主机和从机之间，所以该图可以解释为主机或从机时序图。MISO 信号是从机输出，MOSI 信号是主机输出。NSS 线是从机的选择输入；主机的 NSS 引脚未显示，但假定为无效。主机的 NSS 引脚必须为高电平，或必须重新配置为不影响 SPI 的通用输出。

图 21-5 CPHA=1 SPI 传输格式



21.8 SPI 数据传输

21.8.1 SPI 传输启动

所有 SPI 传输都由主 SPI 设备启动和控制。作为从机设备, SPI 根据选择的 CPHA 格式, 将考虑传输开始于第一个 SCLK 边沿或 NSS 的下降沿。当 CPHA=0 时, NSS 的下降沿表示传输的开始。当 CPHA=1 时 SCLK 上的第一个边沿表示传输的开始。无论哪种 CPHA 模式, 通过使 NSS 线为高电平可以中止传输, 但会使 SPI 从机逻辑和计数器复位。选择的 SCLK 速率对从机操作没有影响, 因为主机的时钟正在控制传输。

当 SPI 配置为主机时, 通过写入 SPDAT 的软件启动传输。

21.8.2 SPI 传输结束

当 SPIF 标志置 1 时, SPI 传输完成。当 SPI 被配置为主机时, SPIF 在第八个 SCLK 周期循环结束时置位。当 CPHA 等于 1 时, SCLK 在第八个 SCLK 周期的最后一半处于不活动状态。

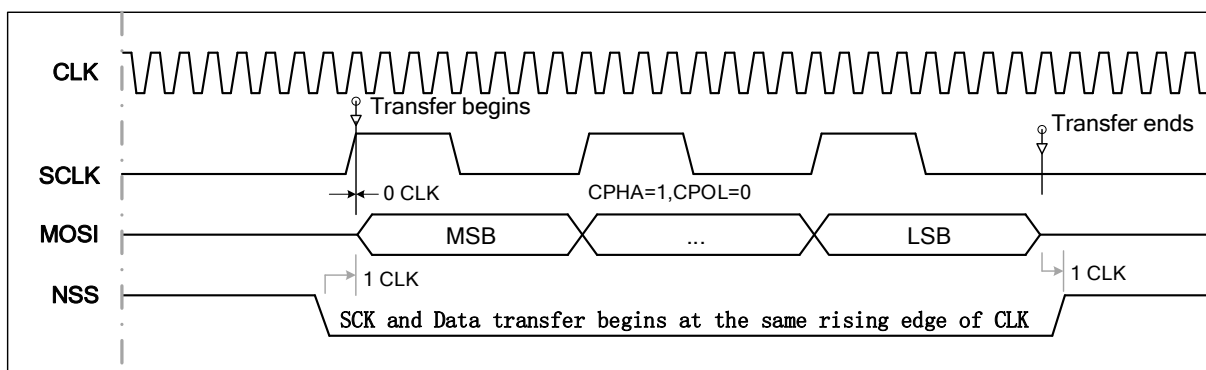
因为 SCLK 线可以与从机的 MCU 时钟异步, 并且从机不能像访问 SCLK 周期那样访问主机尽可能多的信息, 所以当 SPI 作为从机运行时, 结束周期是不同的。例如, 当 CPHA=1 时, 其中最后一个 SCLK 边沿在第八个 SCLK 周期的中间发生, 从机无法知道上一个 SCLK 周期是何时结束。由于这些原因, 从机认为在串行数据的最后一位被采样之后, 传输完成, 这对应于第八个 SCLK 周期的中间。

SPIF 标志设置在传输结束时, 但是 NSS 线仍然为低电平时, 从机不允许将新数据写入 SPDAT。

21.9 SPI 时序图

21.9.1 主控模式传输

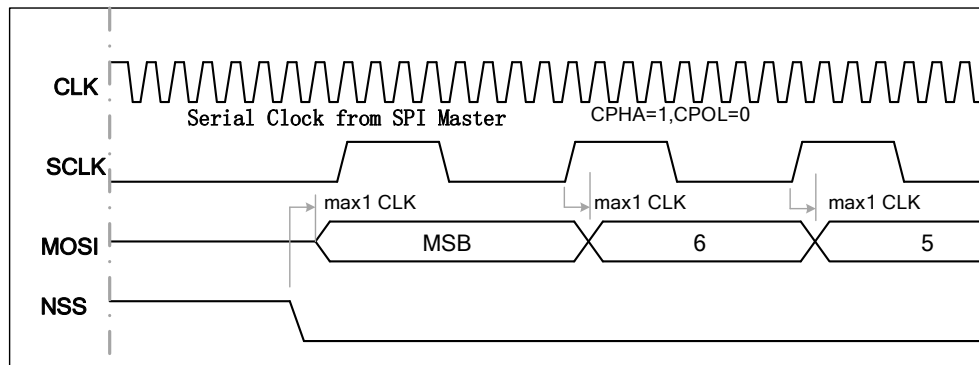
图 21-6 主模式时序图



21.9.2 从动模式传输

在从机模式下的传输开始时，串行输出(MISO)上的数据出现在从机选择(NSS)线上的下降沿之后。

图 21-7 从模式时序图



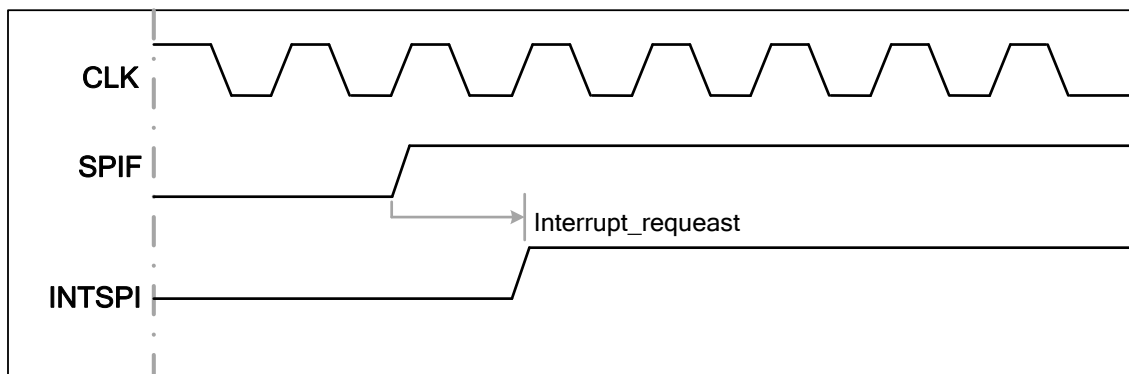
21.10 SPI 中断

SPI 的两个状态标志可以产生中断请求，分别是 SPIF 和 MODF。

使能 SPI 中断必须将其使能位 ESPI 置 1，且将总中断使能位 EAL 置 1。

- 如果 SPI 相关的中断使能打开，SPI 完成自身数据传输后标志位 SPIF=1 时，CPU 将进入中断服务程序。SPIF 位通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。

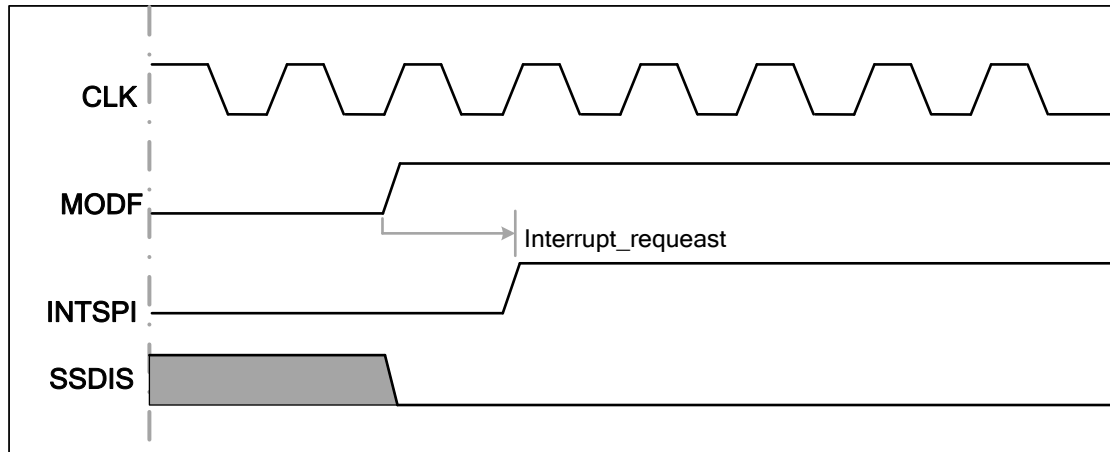
图 21-8 SPIF 中断



Interrupt Request Generated by "spif" Flag

- 如果当 SSN 输入的电平与所选操作模式不一致时，MODF 标志由硬件自动置位（SPI_MS 模块配置为主模式，并且在 SSN 输入处检测到低电平）。该位通过读取 SPSTA 寄存器来由软件清零。

图 21-9 MODF 中断



Interrupt Request Generated by “modf” Flag

21.11 SPI 寄存器

SPSTA-SPI 状态寄存器

地址: E1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSTA	SPIF	WCOL	SSERR	MODF	--	--	--	--
R/W	R	R	R	R	--	--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	SPIF	SPI 通讯完成标志位 数据传输完成后由硬件设置。 通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。
6	WCOL	写操作冲突标志位 在通讯时对 SPDAT 执行写操作自动设置为 1 通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。
5	SSERR	同步从模式错误标志位 若 SSN 错误控制时自动设置为 1 通过清 SPEN 自动清零。
4	MODF	模式错误标志位。也可以通过软件来清除

SPCON-SPI 控制寄存器
地址: E2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCON	SM2	SPEN	--	MSTR	CPOL	CPHA	SM1	SM0
R/W	R/W	R/W	--	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	1	0	1	0	0

位号	符号	描述
7、1、0	SM[2:0]	SPI 波特率发生器（仅在主模式下有效） 000: Fcpu/2 001: Fcpu/4 010: Fcpu/8 011: Fcpu/16 100: Fcpu/32 101: Fcpu/64 110: Fcpu/128 111: reserved
6	SPEN	SPI 通讯使能 0: 禁止; 1: 使能。
4	MSTR	SPI 模式选择位 0: 从模式; 1: 主模式。
3	CPOL	SCK 引脚空闲状态位 0: SCK 低电平空闲; 1: SCK 高电平空闲。
2	CPHA	数据的时钟相位锁存控制位 0: 当 SCKI/SCKO 非空闲时, 锁存数据; 1: 当 SCKI/SCKO 空闲时, 锁存数据。

SPDAT-SPI 数据寄存器
地址: E3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDAT	SPDAT[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	SPDAT[7:0]	SPDAT 是“接收数据”寄存器的读/写缓冲区 当写入 SPDAT 时, 数据被直接放入移位寄存器(没有传输缓冲区)。 读取 SPDAT 时, 返回位于接收缓冲区中的值, 而不是移位寄存器中的值。

P1M-P1 输入/输出控制寄存器
地址: FAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4	P14M	P1.4 的模式选择控制位 0: 输入模式, 从模式时 P1.4(SCK)设置为输入模式。 1: 输出模式, 主模式时 P1.4(SCK)设置为输出模式。
3	P13M	P1.3 的模式选择控制位 0: 输入模式, 主模式时 P1.3(MISO)设置为输入模式。 1: 输出模式, 从模式时 P1.3(MISO)设置为输出模式。
2	P12M	P1.2 的模式选择控制位 0: 输入模式, 从模式时 P1.2(MOSI)设置为输入模式。 1: 输出模式, 主模式时 P1.2(MOSI)设置为输出模式。
1	P11M	P1.1 的模式选择控制位 0: 输入模式, 从模式时 P1.1(NSS)设置为输入模式。 1: 输出模式。
Else		请参考其他章节。

22 I2C

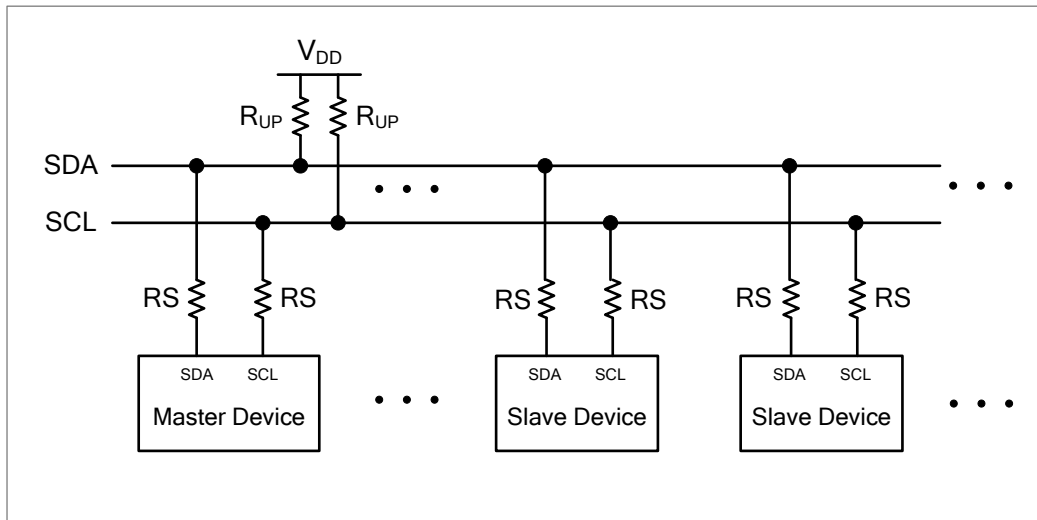
22.1 I2C 概述

该模块提供微处理器和 I2C 总线之间的接口。且支持仲裁和时钟同步，以便允许在多主机系统中运行。I2C 支持正常、快速模式。

其特性如下：

- ◆ 支持 4 种工作方式：主机发送、主机接收、从机发送、从机接收。
- ◆ 支持 2 种传输速度模式：
 - 标准（高达 100Kb/s）；
 - 快速（高达 400Kb/s）；
- ◆ 执行仲裁和时钟同步。
- ◆ 支持多主机系统。
- ◆ 主机方式支持 I2C 总线上的 7 位寻址模式。
- ◆ 从机方式支持 I2C 总线上的 7 位寻址模式。
- ◆ 中断产生。

图 22-1 I2C 连接框图



22.2 I2C 端口配置

若使用 I2C 功能，则首先应将相应端口配置成 SCL，SDA 通道，即 P03 为 SDA，P02 为 SCL。且 P02、P03 均设为输入模式。配置 I2C 通道后，该组端口默认为开漏状态，即输出数据为开漏输出。可通过 PXUP 配置是否使能 SCL，SDA 口内部的上拉电阻，或者在芯片外部增加上拉电阻。

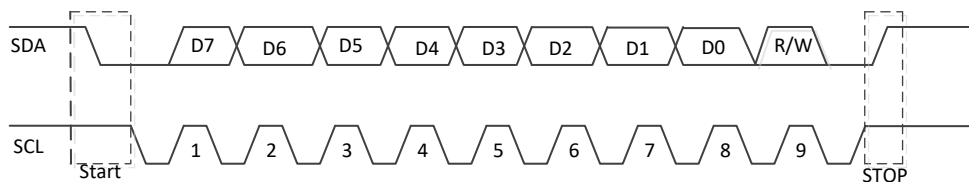
22.3 I2C 总线传输方式

I2C 发送结构包括 **START(S)**起始位，8 位地址字节，一个或多个数据字节，以及 **STOP (P)**停止位。起始位由主机产生，以便对传输进行初始化。

发送的数据为最高有效位（**MSB**）优先。在地址字节中，高 7 位为地址位，低位为数据方向位（**R/W**）**R/W=0** 时，表示该传输为写操作；**R/W=1** 时，表示该传输为读操作。

接收到每个字节后，接收器（主机或从机）必须发送一个 **ACK** 信号。若发送器没有接收到 **ACK** 信号，则会识别为 **NACK** 信号。在写操作中，主机发送数据给从机，然后等待从机返回 **ACK** 信号。在读操作中，从机发送数据给主机，然后等待主机返回 **ACK** 信号。最后，主机产生一个停止信号来结束数据传输

图 22-2 I2C 数据传输框图

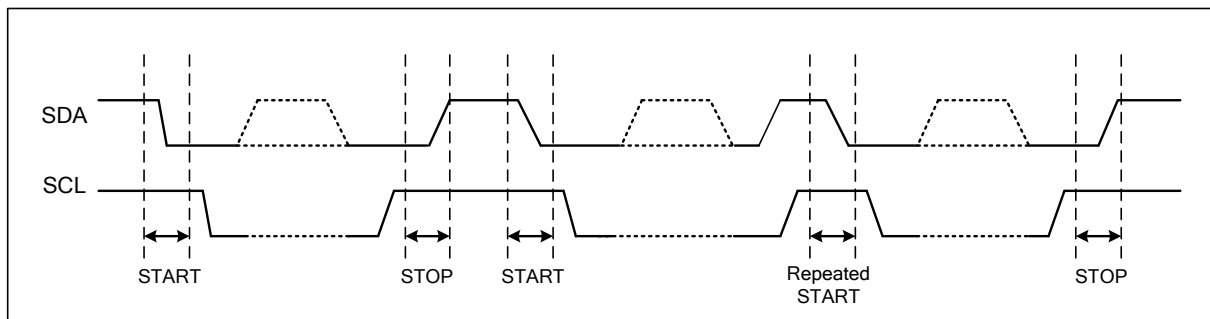


22.3.1 开始和停止条件

I2C 总线时序定义了起始 **START** 和结束 **STOP** 的条件。时钟 **SCL** 位高时，数据线 **SDA** 由高电平变低电平的跳变被认为是 **START** 标志。时钟 **SCL** 位高时，数据线 **SDA** 由低电平变为高电平的跳变被认为是 **STOP** 标志。起始和结束由主机产生。

若主机发出 **STOP** 停止信号，传输停止。主机也可以不发停止位，直接再次发出起始 **START** 信号，则继续之前通信，或者换另一个地址通信。

图 22-3 I2C 停止和开始条件

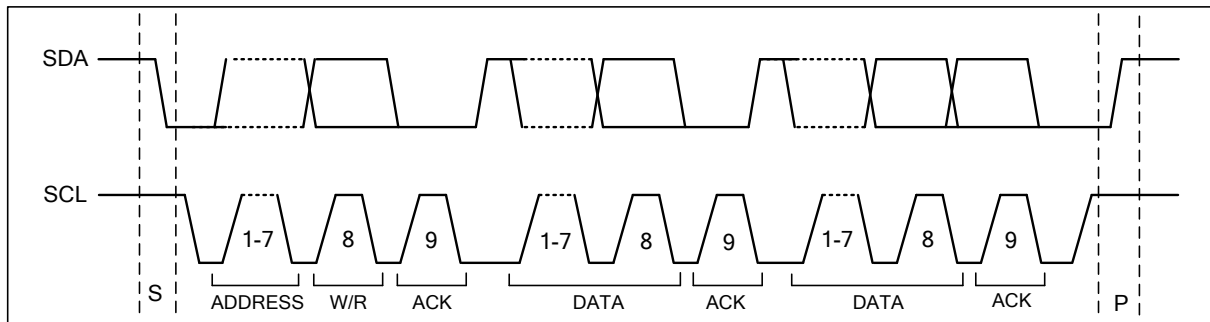


起始条件，重复起始条件和停止条件

22.3.2 地址数据格式

起始位 **START** 之后, 第一个字节必须是 7 位地址 **SLA**+第 8 位读写方向位(**R/W**), 用以定义目标从机地址以及对从机写入或读出数据。若第 8 位是 0, 即 **SLA+W**, 表示下个字节开始主机向从机写数据; 若是 1, 即 **SLA+R**, 就表示下字节开始, 主机向从机读数据。所以, 一个典型数据包包含起始位 **START**, **SLA+W/R**, 一个或多个字节数据, 最后是停止位 **STOP**。当一个从机已被寻址及读写方向通过字节 **SLA+W/R**, 随后的 8 位数据就跟随之前的设定进行传输。

图 22-4 I2C 传输格式



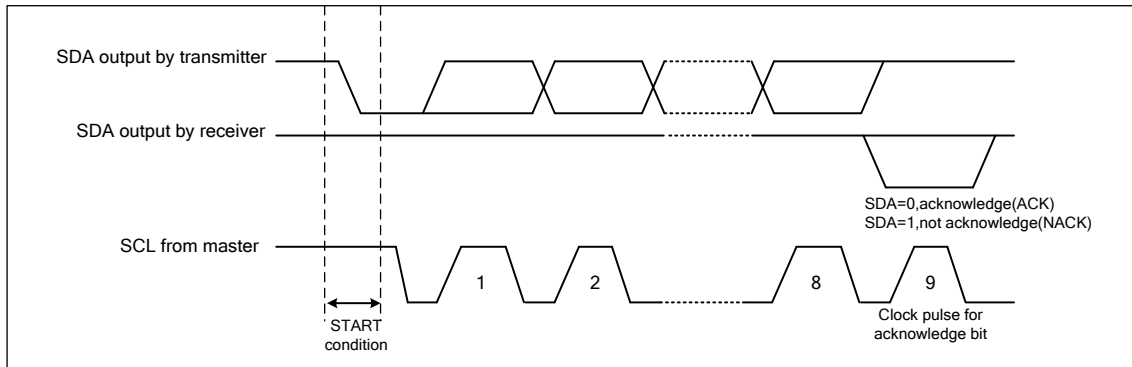
一个传输中的I2C数据格式

22.3.3 应答 ACK

每字节开始传输后, **SCL** 的第 9 个脉冲用于传输应答位 (**ACK**)。主机或者从机通过将 **SDA** 拉低, 来产生应答位 **ACK**。应答位时钟由主机产生。发送端设备在应答位时钟高电平周期内, 需放弃对 **SDA** 的控制。在应答位时钟周期的高电平时, **SDA** 保持低电平用以表示接收端已成功接收到发送端的数据。通常被寻址的从机在整个传输过程中每字节都需要回复应答位。当该从机接收无法应答 (**NACK**) 从机地址时, 从机将 **SDA** 线保持高以便主机产生停止(**STOP**)或发送重复开始(**START**)信号。

若从机没有应答地址后, 将自身切换到未寻址从机模式, 从而无法接收更多数据字节。这时从机将 **SDA** 线拉高, 此时主机应发送停止 **STOP** 信号或重复起始(**repeated START**)信号。若是主机接收, 主机控制着收发字节个数, 主机在最后一个字节收发结束后不发应答位信号, 从机端将切换为未寻址从机模式, 并释放 **SDA** 线, 以便主机直接发停止位 **STOP** 或重复发开始位 **START**。

图 22-5 I2C 应答



22.3.4 I2C 时钟时序

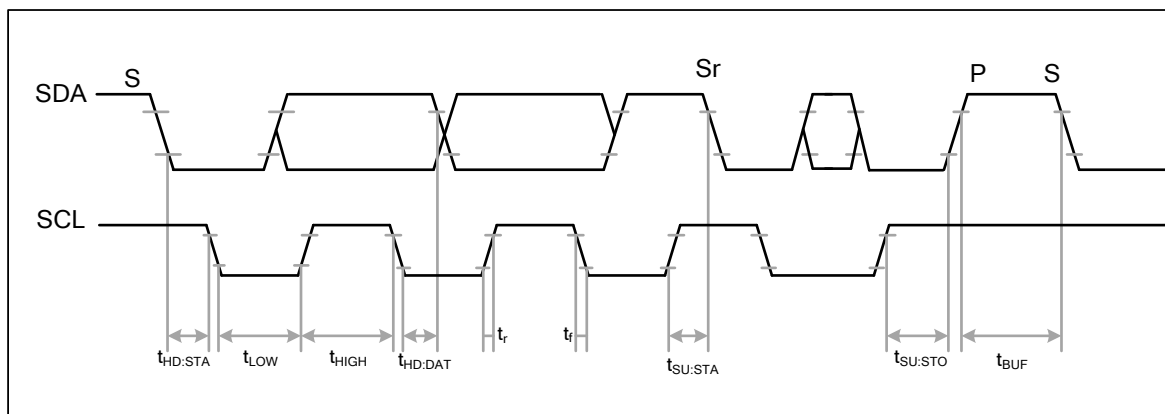
下图为 I²C 传输过程的最小应遵循的时序。

$t_{SU:STA}$ 和 t_{HIGH} 时序的计算假设 $t_r < \text{clk}$ 周期。如果 t_r 大于 clk 周期, 则 I²C 内核检测到时钟延长。

clk 为系统时钟, SCL 为 I²C 的时钟周期

Time	最小值	Time	最小值
$t_{HD:STA}$	15clk	$t_{HD:DAT}$	7clk
t_{LOW}	30clk	$t_{SU:DAT}$	5clk
t_{HIGH}	30clk	$t_{SU:STO}$	23clk
$t_{SU:STA}$	15clk	t_{BUF}	55clk

图 22-6 I2C 时钟时序



I2C Clock Timings

22.4 I2C 主控模式

有四个寄存器用于与主机连接：操作控制、状态、数据接收和数据发送寄存器。

寄存器		地址
写入	读取	
主控模式控制寄存器 I2CCON	主控模式状态寄存器 I2CSTA	DCH、DDH
主控发送数据寄存器 I2CDAT	主控模式接收数据寄存器 I2CDAT	DAH

22.4.1 I2C 主控模式时钟频率

当 I²C 处于主模式时，这个可编程时钟脉冲发生器提供时钟脉冲。当 I²C 处于从模式时，时钟发生器被抑制。时钟生成器的功能由“I2CCON”寄存器的“CR0”、“CR1”和“CR2”位控制。下表列出了不同设置下的时钟速率：

CR2	CR1	CR0	I2C Prescaler	比特率(Kbit/s)			
				4MHz	8MHz	16MHz	32MHz
0	0	0	256	15.6	31.3	62.5	125
0	0	1	224	17.9	35.7	71.4	142.9
0	1	0	192	20.8	41.7	83.3	166.7
0	1	1	160	25	50	100	200
1	0	0	960	4.2	83.3	16.7	33.3
1	0	1	120	33.3	66.7	133.3	266.7
1	1	0	60	66.7	133.3	266.7	533.3
1	1	1	(TC2 定时器溢出速率)/8				

22.4.2 I²C 主控模式控制与状态

I²C 主控模式主要有 4 个控制位：ENSI, STA, STO, AA 位，STA 位将产生 START 或 RESTART 条件。STO 位决定数据传输是否再周期结束时停止，或者继续。通过控制寄存器写入 STO=1, STA=1, ENSI=1，以执行相应的开始和停止。通过软件写入数据寄存器进行对从机的读写操作。通过设置 AA 位，主机将在收到数据时进行返回 ACK 或者 NACK。当操作完成，产生中断。可以从状态寄存器中读取相应的状态，并可在数据寄存器读取数据。

通过以下的组合列表可实现主控模式下各种操作：

STA：发送起始信号

STOP：发送结束信号

AA：使能应答

主控模式操作：（SLA：从机地址。W:写。R: 读）

软件操作	操作解释	主控模式操作控位			软件操作后 I ² CSTA 寄存器值	I ² C 硬件下一步操作
		STA	STO	AA		
发送 START	发送开始或者重复开始	1	0	x	08H: 已发送 START 信号	发送 SLA+R;接收 ACK 发送 SLA+W
					10H: 已发送 Repeated START 信号	发送 SLA+R;接收 ACK;
						发送 SLA+W; I ² C 切换到主机接收模式
发送 STOP	发送停止	0	1	0		
发送 ACK	主机接收数据后返回 ACK	0	0	1	50H: 已接收数据字节; 并已回复 ACK	读取已经接收的数据
发送 NACK	主机接收数据后返回 NACK	0	0	0	58H: 已接收数据字节; 并已返回 NACK	读取已经接收的数据
发送 SLA+W	发送从机地址并进行写操作	x	0	x	18H: 已发送 SLA+W, 并已经接收到 ACK 20H: 已经发送 SLA+W 并已接收 NACK	发送数据; 接收 ACK;
						发送重复 START 开始信号;
						发送 STOP 结束信号, 复位 STO 标志位;
						发送 STOP 结束信号后接着再发送 START 开始信号, 复位 STO 标志位
发送 I ² CDAT 中的数据	发送数据	0	0	0	28H: 已经发送 I ² CDAT 寄存器中的自己, 并已接收 ACK 30H: 已经发送 I ² CDAT 寄存器中的自己, 并已接收 NACK	发送数据; 接收 ACK;
						发送重复 START 开始信号;
						发送 STOP 结束信号, 复位 STO 标志位;
						发送 STOP 结束信号后接着再发送 START 开始信号, 复位 STO 标志位

发送 SLA+R	发送从机地址并读操作	x	0	x	40H: 发送 SLA+R 并已经接收 ACK	接收数据字节; 返回 NACK 接收数据字节; 返回 ACK
					48H: 发送 SLA+R 并已经接收 NACK	发送重复 STATRXINH 发送 STOP 信号; 发送 STOP 信号, 接着发送 START 信号

22.5 I2C 从动模式

有五个寄存器用于与主机连接: 操作控制、状态、从机地址、数据接收和数据发送寄存器。

寄存器		地址
写入	读取	
从动地址寄存器 I2CADR	从动地址寄存器 I2CADR	DBH
主控模式控制寄存器 I2CCON	主控模式状态寄存器 I2CSTA	DCH、DDH
主控发送数据寄存器 I2CDAT	主控模式接收数据寄存器 I2CDAT	DAH

在从动模式下, 必须往地址寄存器中写入本机地址。地址寄存器由七个地址位组成。该地址寄存器可以读取或写入地址。

从动模式下通过下面组合列表可以各种操作:

(SLA: 从机地址。W:写。R: 读)

I2C 操作	操作解释	从动模式操作控位			接收到信号或发送后 I2CSTA 寄存器值	I2C 硬件下一步操作
		STA	STO	AA		
接收到 SLA+R	接收到本机地址并且进行读数据操作	x	0	0	A8H: 接收到 SLA+R 并已经返回 ACK	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
接收到 SLA+R/W 时仲裁丢失或者接收到 SLA+R	接收到从机地址读写操作时丢失或者接收到本机地址并进行读操作	x	1	0	B0H: 接收到 SLA+R/W 仲裁丢失。或者接收到 SLA+R 并返回 ACK	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
发送 I2CDAT 中的数据	本机发送数据	x	0	0	B8H: 发送数据字节	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
发送 I2CDAT 中的数据	数据字节发送	0	0	0	C0H: 数据字节发送并接收到 NACK	切换到未寻址从机模式; 从机地址未识别
		0	0	1		切换到未寻址从机模式; 从

						机地址将被识别
		1	0	0		切换到未寻址从机模式: 总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式; 从机地址将被识别; 总线空闲后将发送开始信号
发送 I ² C DAT 中的数据	数据字节发送	0	0	0	C8H: 最后一个字节数据已经发送, 并接收到 ACK	切换到未寻址从机模式; 从机地址未识别
		0	0	1		切换到未寻址从机模式; 从机地址将被识别
		1	0	0		切换到未寻址从机模式: 总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式; 从机地址将被识别; 总线空闲后将发送开始信号
接收到 SLA+W	接收到本机地址并进行写入操作	x	0	0	60H: 已经接收到 SLA+W, 并返回 ACK	接收数据字节; 返回 NACK;
				1		接收数据字节; 返回 ACK;
接收到 SLA+R/W 时仲裁丢失; 并接收到对应的 SLAR+W	接收到从机地址读写操作时丢失或者接收到本机地址并进行读操作	x	0	0	68H: 接收到 SLA+R/W 仲裁丢失。或者接收到 SLA+R 并返回 ACK	接收数据字节; 返回 NACK;
				1		接收数据字节; 返回 ACK;
接收到一个字节 DATA	从机接寻址成功并已经接收到数据	x	0	0	80H: 寻址成功, 接收到数据, 并已经回复 ACK	接收数据字节; 返回 NACK;
				1		接收数据字节; 返回 ACK
接收到一个字节 DATA	从机寻址成功并接收到数据	0	0	0	88H: 从机寻址成功, 接收到 DATA, 并回复 NACK	切换到未寻址从机模式; 从机地址未识别
		0	0	1		切换到未寻址从机模式; 从机地址将被识别
		1	0	0		切换到未寻址从机模式: 总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式; 从机地址将被识别; 总线空闲后将发送开始信号

接收到 STOP 信号或者重复 START 信号	当从机接收器或者从机发生器仍然被寻址时，接收到了 STOP 结束信号或者重复 START 开始信号	0	0	0	A0H: 接收到 STOP，或者接收到重复的 START	切换到未寻址从机模式;从机地址未识别
		0		1		切换到未寻址从机模式; 从机地址将被识别
		1		0		切换到未寻址从机模式: 总线空闲后将发送开始信号
		1		1		切换到未寻址从机模式; 从机地址将被识别; 总线空闲后将发送开始信号

22.6 I2C 中断

I²C 中断标志位是 SI，如果 EI2C（IEN1.0）=1，且 EAL=1，当 I²C 中断发生时，CPU 会进入中断服务程序去执行中断代码。SI 需要软件清零。

当 SI 置位时，用户可以去读取 I2CSTA 寄存器的状态码，用户可根据读取的状态码，来确认哪部已经执行和决定下一步动作。

在 SI 被清零之前，SCL 低电平周期延迟，传输暂停，该状态对于从机处理接收到的数据非常有用，可以确保准确处理前一数据再接收下一个数据。SI 位被软件清零后，I²C 才会继续下一步：总线继续产生 START 或者 Repeat START 条件，STOP 条件，8 位数据，或者其他控制字节或位，因此在 SI 软件清除前，软件应该准备好合适的寄存器设置。

22.7 仲裁和同步模式

在多主机条件下，同一时间只有一个主机可在总线上传输数据。需要决定由哪个主机可控制总线以及实现传输。时钟同步与仲裁应用于配置多主机传输。时钟同步会在和其他设备同步 SCL 信号时运行。

当同一时间有两个主机要传输时，时钟同步将会在 SCL 由高电平转变成低电平的时候开始。如果主机 1 先将 SCL 切换为低电平，那么主机 1 会将 SCL 锁定在低电平状态直到将其切换成高电平状态。不过，如果有其他主机的 SCL 时序依然保持在低电平状态的话，那么主机 1 将 SCL 时序由低电平切换成高电平状态的过程将不会改变 SCL 的状态，SCL 时序将仍保持在低电平状态。也就是说，SCL 线被有最长低电平周期的主机保持在低电平状态。当所有设备的时钟周期都转换到高电平时，SCL 线将由低转成到高电平状态。在这期间，主 1 将保持在由低电平到高电平的等待状态，之后再继续它的传输。经过时钟同步之后，所有设备的时钟和 SCL 时钟是一样的了。仲裁是用来决定

哪个主机能够通过 SDA 信号来完成它的传输。两个主机可能会在同一时间发出开始信号以及传输数据，导致两者会互相影响。仲裁会强制使得其中一个主机失去总线的控制权。数据传输依然会继续，直到两个主机输出了不同的数据信号。如果其中一个主机传输了高电平状态，而另外一个主机传输了低电平状态，SDA 线会被拉低。输出高电平状态的主机将会检测到 SDA 线上的异常，并失去总线的控制权。输出低电平状态的主机成功获得总线的控制权，并继续它的传输，仲裁的过程中不会丢失数据

22.8 I2C 状态码和应用程序实例

22.8.1 状态码

I²STAT 状态寄存器中,有两种状态码不归属 20 种之前所述状态: F8H 及 00H。

F8H 用以标示在之前的传输中没有有效信息，意味着，SI 标志位为 0 且无 I2C 中断产生。

00H 标示在传输过程中，有总线错误发生。总线错误是指 START 或 STOP 在传输的过程中出现在错误的位置。所有状态码解释如下表：

表 22-1 I2C 状态码

主机模式		从机模式	
状态码	描述	状态码	描述
0x08	开始信号	0xA0	从机接收到 STOP 信号或者接收到重复 START 信号
0x10	主机重复开始信号	0xA8	接收到地址读并应答
0x18	主机发送地址写并接收应答	0xB0	仲裁丢失
0x20	主机发送地址写并无应答	0xB8	从机发送数据并应答
0x28	主机发送数据并接收应答	0xC0	从机发送数据无应答
0x30	主机发送数据并无应答	0xC8	从机发送最后一个字节
0x40	主机发送地址读并接收应答	0x60	从机接收到地址写并应答
0x48	主机发送地址读并无应答	0x68	仲裁丢失
0x50	主机接收数据并应答	0x80	从机接收数据并应答
0x58	主机接收数据并无应答	0x88	从机接收数据无应答
0xF8	释放总线		

22.8.2 I2C 程序示例

```
Void I2C_ISR (void) interrupt I2C_VECTOR // I2C_VECTOR=6
{
    switch (I2CSTA)
    {
        //=====
        //Bus Error, always put in ISR for noise handling
        //=====
        case 0x00: // *00H, 发生总线错误*/
            STO = 1; // 从总线错误中恢复
            break;

        //=====
        //Master Mode
        //=====
        case 0x08: // *08H, 发送 START 信号*/
            STA = 0; // STA 位需要通过软件清除
            I2CDAT = SLA_ADDR; // 加载 SLA+W/R
            break;
        case 0x10: // *10H, 重复发送 START 信号*/
            STA = 0;
            I2CDAT = SLA_ADDR;
            break;

        //=====
        //Master Transmitter Mode
        //=====
        case 0x18: // *18H, 发送 SLA+W, 接收 ACK*/
            I2CDAT = NEXT_SEND_DATA1; // 加载数据
            break;
        case 0x20: // *20H, 发送 SLA+W, 接收 NACK*/
            STO = 1; // 传输停止
            AA = 1;
            break;
        case 0x28: // *28H, 发送数据, 接收 ACK*/
            if (Conti_TX_Data) // 如果继续发送数据
                I2CDAT = NEXT_SEND_DATA2;
            else // 如果没有数据要发送
            {
                STO = 1;
                AA = 1;
            }
            break;
        case 0x30: // *30H, 发送数据, 接收 NACK*/
            STO = 1;
```

```
AA = 1;

    break;

//=====
//Master Receiver Mode
//=====

case 0x40:                                /*40H, 发送 SLA+R, 接收 ACK*/
    AA = 1;                               //接收数据, 返回 ACK。
    break;

case 0x48:                                /*48H, 发送 SLA+R, 接收 NACK*/
    STO = 1;
    AA = 1;
    break;

case 0x50:                                /*50H, 接收数据, 发送 ACK*/
    DATA_RECEIVED1 = I2CDAT;              //存储接收数据
    if (To_RX_Last_Data1)                  //如果最后的数据被接收
        AA = 0;                            //接收数据, 返回 NACK。
    else                                    //如果继续接收数据
        AA = 1;
    break;

case 0x58:                                /*58H, 接收数据, 发送 NACK*/
    DATA_RECEIVED_LAST1 = I2CDAT;
    STO = 1;
    AA = 1;
    break;

//=====
//Slave Receiver and General Call Mode
//=====

case 0x60:                                /*60H, 主机收到 SLA+W, 返回 ACK*/
    AA = 1;
    break;

case 0x68:                                /*68H, 主机在发送 SLA+R/W 时, 仲裁丢失;
                                           并接收到了对应的 SLA+W; 已返回 ACK。*/
    AA = 0;                                //仲裁丢失后, 接收数据, 返回 NACK
    STA = 1;                               //如果总线空闲, 重新尝试发送 START
    break;

case 0x80:                                /*80H, 接收 DATA, 并返回 ACK*/
    DATA_RECEIVED2 = I2CDAT;
    if (To_RX_Last_Data2)
        AA = 0;
    else
        AA = 1;
    break;

case 0x88:                                /*88H, 接收 DATA, 并返回 NACK,
                                           切换至未寻址的从机模式*/
```

```
DATA_RECEIVED_LAST2 = I2CDAT;
    AA = 1;
    break;

//=====
//    Slave Mode
//=====

case 0xA0:                /*A0H, 当从机接收器或者从机发生器仍然被寻址时,
                           接收到了 STOP 结束信号或者重复 START 开始信号*/

    AA = 1;
    break;

//=====
//    Slave Transmitter Mode
//=====

case 0xA8:                /*A8H, 接收 SLA+R, 返回 ACK*/
    I2CDAT = NEXT_SEND_DATA3;
    AA = 1;                //当 AA 置 1, 发送一个字节的数据
    break;

case 0xB0:                /*B0H, 主机在发送 SLA+R/W 时仲裁丢失了;
                           已接收到了对应的 SLA+R; 且已返回 ACK 信号。*/

    I2CDAT = DUMMY_DATA;
    AA = 0;                //当 AA 为 0, 发送最后一个字节的数据
    STA = 1;               //如果总线空闲, 重新尝试发送 START
    break;

case 0xB8:                /*B8H, 数据字节已经发送, 将接收到 ACK 信号。*/
    I2CDAT = NEXT_SEND_DATA4;
    if (To_TX_Last_Data)    //如果最后一个数据被传输
        AA = 0;
    else
        AA = 1;
    break;

case 0xC0:                /*C0H, 数据字节已经发送, 并已接收到 NACK 信号
                           切换至未寻址的从机模式*/

    AA = 1;
    break;

case 0xC8:                /*C8H, 最后一个字节的数据已经发送,
                           并已接收到 ACK 信号, 切换至未寻址的从机模式*/

    break;
}
SI = 0;                    //SI 应该是 I2C ISR 的最后一个命令
while(STO);                //等待传输停止或总线无错误, STO 由硬件清除
}
```

22.9 I2C 寄存器

I2CCON-I2C 接口操作控制寄存器

地址: DCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7、1、0	CR[2:0]	I2C 时钟速率 000: Fcpu/256; 001: Fcpu/224; 010: Fcpu/192; 011: Fcpu/160; 100: Fcpu/960; 101: Fcpu/120; 110: Fcpu/60; 111: (TC2 计数器溢出率)/8
6	ENSI	I2C 使能位 0: 关闭; 1: 使能。
5	STA	START 标志位 0: 没有发送 START 开始信号; 1: 若总线空闲则发送 START 开始信号。 用户需软件清除。
4	STO	STOP 标志位 0: 没有发送 STOP 结束信号; 1: 若 I2C 总线为主机模式, 则发送 STOP 结束信号。 STO 由硬件自动清 0。
3	SI	串行中断标志位 I2C 所有 26 种状态中出现一种, 硬件就会置 1 此位 (F8H 除外), 此时软件根据读取 I2STAT 值, 来确认哪步已经执行和决定下一步动作。 SI 由软件清 0。在 SI 被清 0 之前, SCL 低电平周期延长, 传输暂停, 该状态对于从机处理接收到的数据非常有用, 可以确保准确处理前一数据再接收下一个数据。 SI 位被软件清 0 后, I2C 外设才会继续下一步: SI 清除后, 总线将继续产生 START 和 repeat START 条件, STOP 条件, 8 位数据, 或者其他控制字节或位。因此, 在 SI 软件清除前, 软件应该准备好合适的寄存器设置
2	AA	应答响应 ACK 标志位 当 AA = 1 时, 将在以下情况下返回 “ACK” : - 当 I2C 处于主接收器模式时已接收到一个数据字节 - 当 I2C 处于从接收器模式时已接收到一个数据字节 当 AA = 0 时, 在以下情况下将返回 “NOACK” : - 当 I2C 处于主接收器模式时已接收到一个数据字节 - 当 I2C 处于从接收器模式时已接收到一个数据字节

I2CADR-I2C 从机地址
地址: DBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADR	I2CADR[6:0]							--
R/W	R/W							--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:1	I2CADR[6:0]	I2C 从机地址

I2CDAT-I2C 数据缓存器
地址: DAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CDAT	I2CDAT[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	I2CDAT[7:0]	I2CDAT 寄存器存储的是要通过 I2C 总线发送出去的一字节数据, 或者是刚从 I2C 总线上接收到的一字节数据。当它不在字节移位的过程中时, 控制器可以读写这个 8 位的直接寻址特殊功能寄存器。由于 I2CDAT 寄存器没有缓存和双缓冲, 因此当发生 I2C 中断时, 用户只能读 I2CDAT 寄存器。

I2CSTA-I2C 状态代码
地址: DDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSTA	I2CSTA[7:3]					--	--	--
R/W	R/W					--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:3	I2CSTA[7:3]	I2C 状态代码

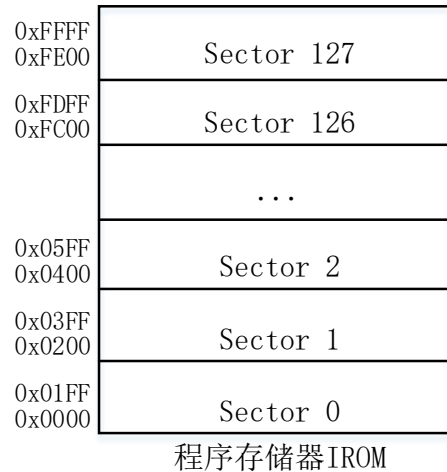
P0M-P0 输入/输出控制寄存器
地址: F9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	P03M	P0.3 的模式选择控制位 0: 输入模式(SDA); 1: 输出模式。
2	P02M	P0.2 的模式选择控制位 0: 输入模式(SCL); 1: 输出模式。

23 在线编程

JSC69F00X 内置 64KB 程序存储 (IROM)，均分为 128 个扇区 (每扇区 512 字节)。在线编程就是使能固件随意的更改每扇区的数据，换句话说，就是存储数据到非易失存储器或者现场升级固件的一个通道。



23.1 扇区擦除

下面是进行扇区擦除的步骤：

1. UNLOCK，配置 PEKEY=0xA5 进行解锁；
2. 配置 PEROMP 地址，选择准备擦除的扇区；
3. 配置 PECMD=0x22，选择 main 区域；
4. 配置 PECMD=0xA5，执行扇区擦除。

23.2 编程

JSC69F00X 支持在所有的程序存储器区域执行字节编程功能，可进行单字节编程。进行一个字节的编程时，配置编程字节的高位和低位进行编程。进行编程步骤如下：

- 1、UNLOCK，配置 PEKEY=0xA5 进行解锁；
- 2、配置 PECMD=0x22 选择 main 区域；
- 3、将编程数据写入 PEROMD 寄存器；
- 4、配置 PEROMA、PEROMP，选定编程区域，必须先配置 PEROMA，再配置 PEROMP。
- 5、配置 PECMD=0xAA 执行编程操作；
- 6、LOCK，配置 PEKEY!=0xA5 进行锁定。

注：进行编程操作，在选择编程地址时，需要按顺序先配置地址低字节（PEROMA），再配置地址高字节（PEROMP）。

23.3 在线编程寄存器

PEROMD-在线编程数据寄存器

地址：97H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEROMD	PEROMD[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	PEROMD[7:0]	ISP 编程数据

PEROMP-在线编程 ROM 地址高字节

地址：96H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEROMP	PEROMP[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	PEROMP[7:0]	ISR 编程 ROM 地址高八位

PEROMA-在线编程 ROM 地址低字节

地址：95H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEROMA	PEROMA[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	PEROMA[7:0]	ISR 编程 ROM 地址低八位

PECMD-在线编程命令寄存器
地址: 94H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PECMD	PECMD[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	PECMD[7:0]	0xA5: 执行扇区擦除,对应的扇区地址为 PEROMP[7:1] 0xAA: 执行编程操作 0x22: 选择 main 区域

PEKEY-在线编程 KEY 寄存器
地址: 9CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PEKEY	PEKEY[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	PEKEY[7:0]	UNLOCK: 配置 PEKEY=0xA5; LOCK: 配置 PEKEY!=0xA5; UNLOCK 状态下才能执行 PECMD 以及通过 JTAG 往 Flash 写数据。

FLCTIM-控制 FLASH 控制器微秒寄存器
地址: 88H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLCTIM	--	--	FLCTIM[5:0]					
R/W	--	--	R/W					
复位值	0	0	0	1	1	1	1	1

位号	符号	描述
5:0	FLCTIM[5:0]	控制 FLASH 控制器微秒的计时时间

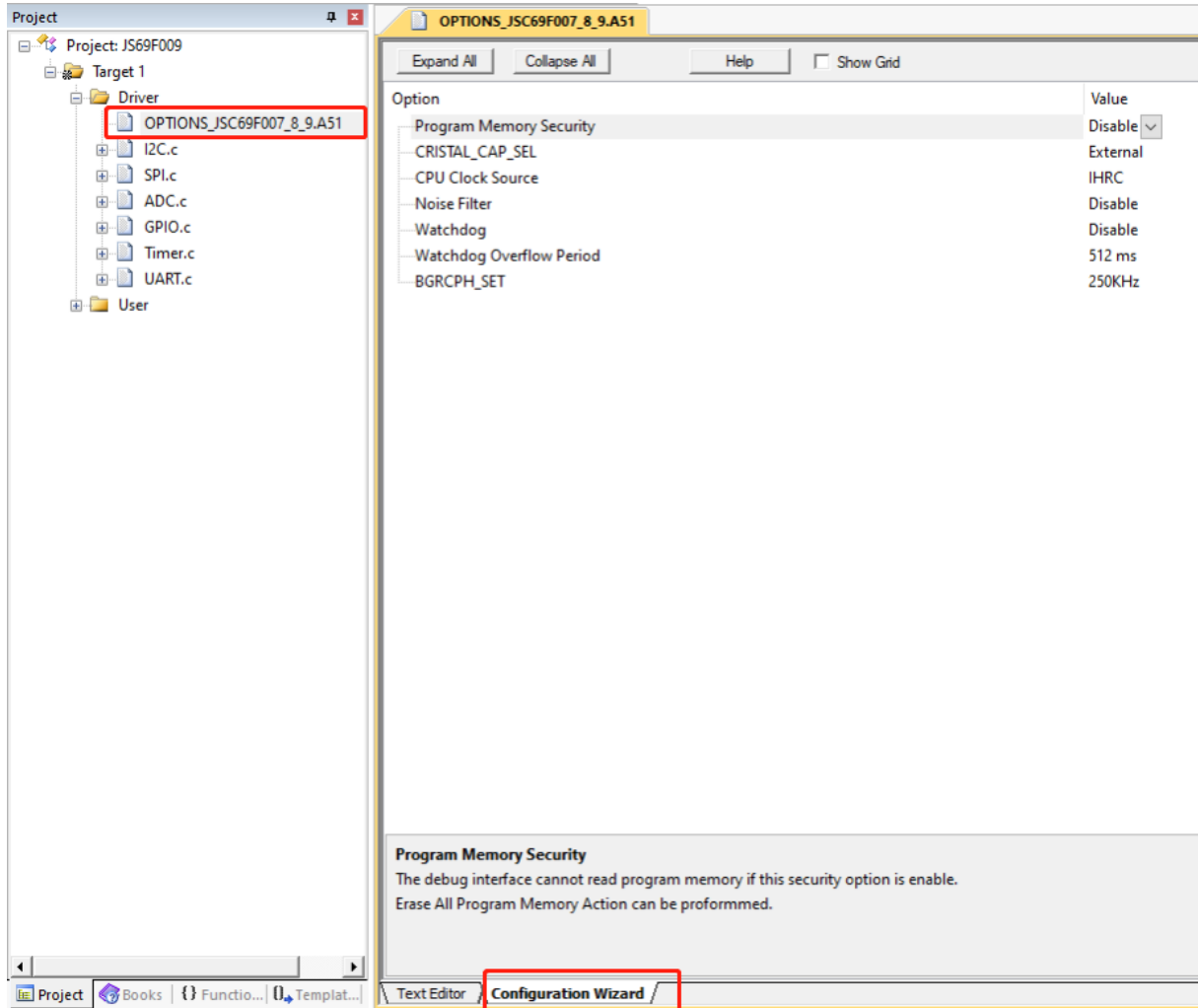
FLCTIM[5:0]的作用是微秒的计数定时, 主要用于写和擦除, 和 MCU 内核的时钟频率 Fcpu 是对应关系:

Fcpu(MHZ)	FLCTIM[5:0]
32M	0x1F
16M	0x0F
8M	0x07
4M	0x03
2M	0x01

24 系统配置 OPTION

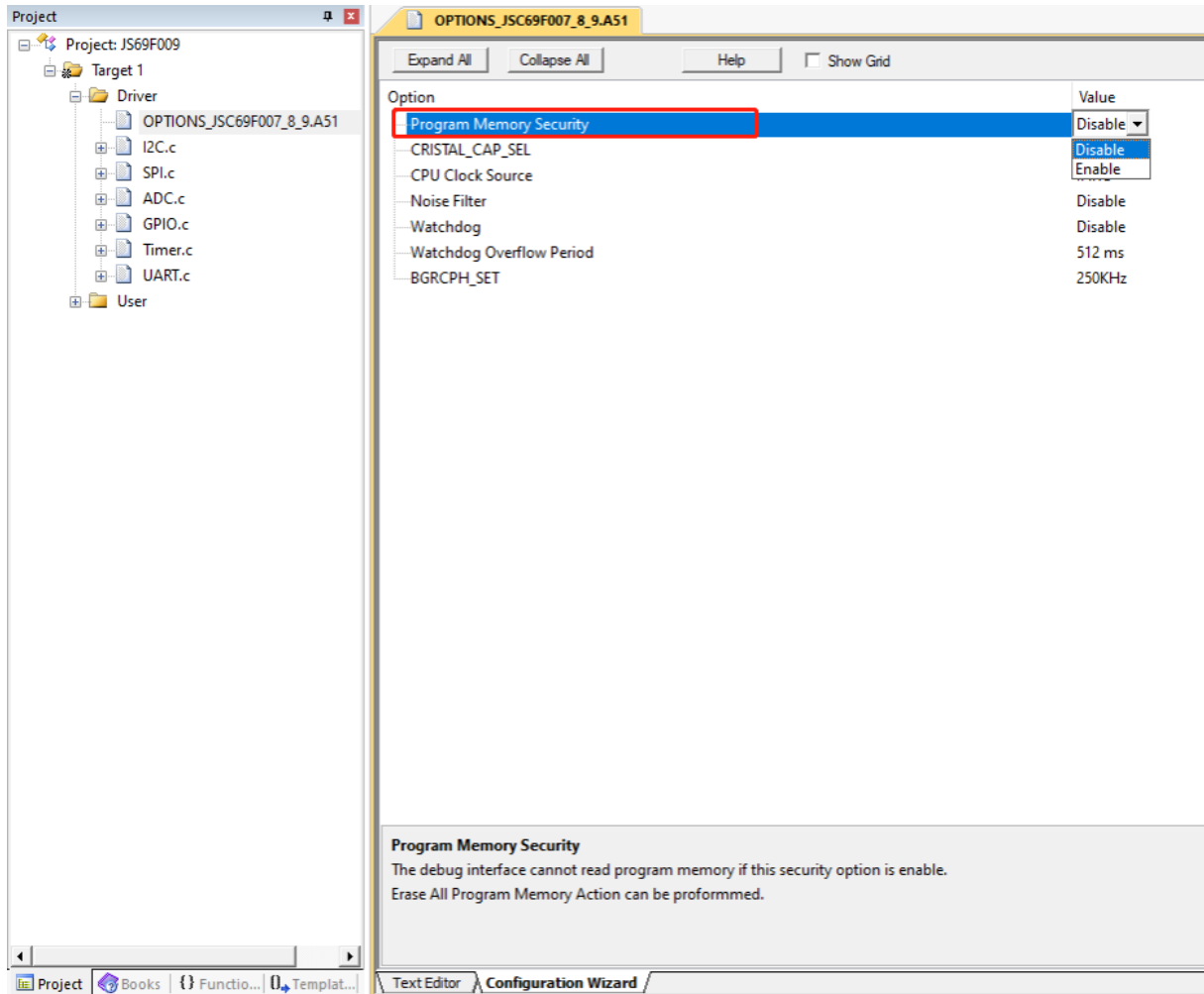
24.1 OPTION 信息

JSC69F00X 带有 OPTIONS_JSC69F007_8_9.A51 文件，用户可根据自身需求更改 OPTION 配置



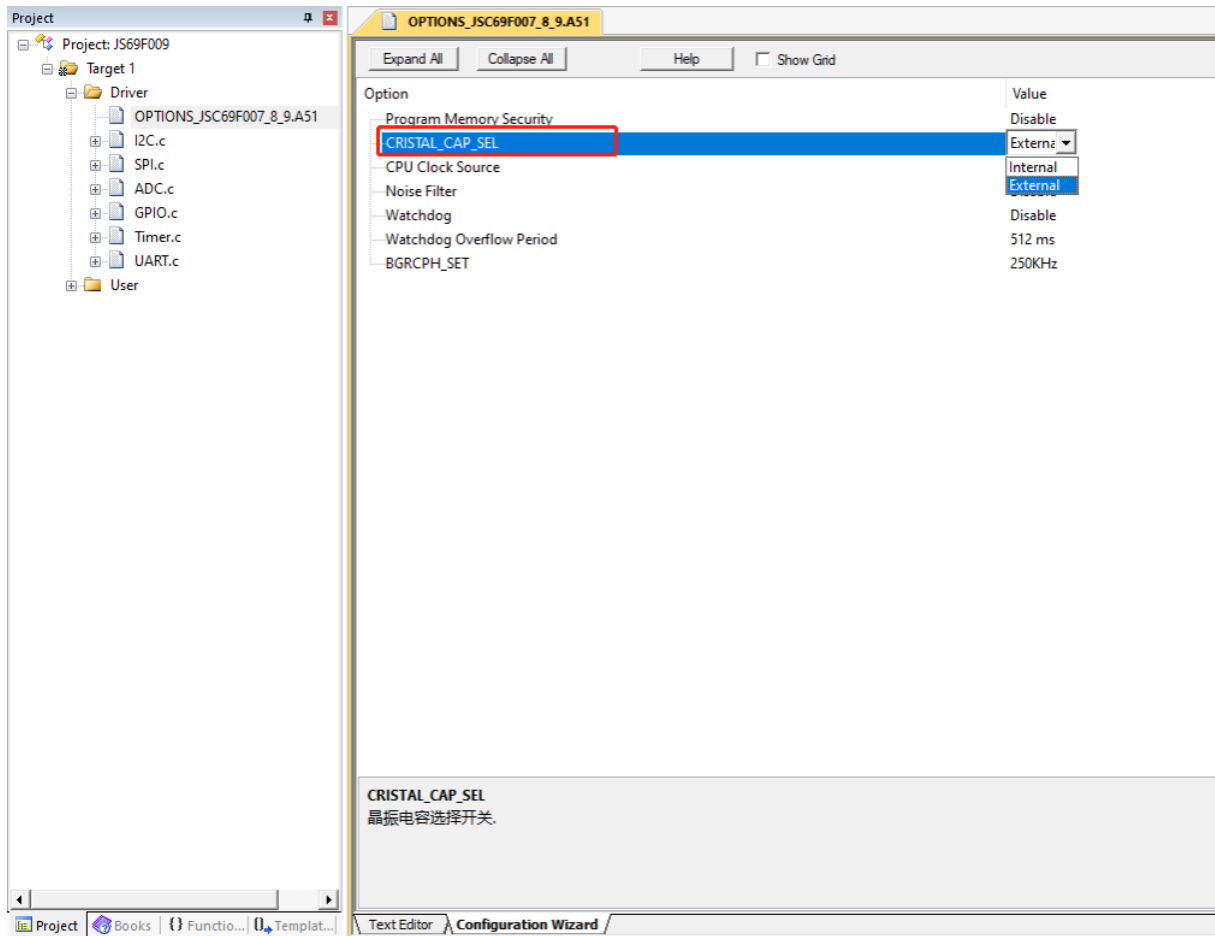
点击任意 OPTION 选项，可在下方看到其具体设置说明。

24.2 程序加密



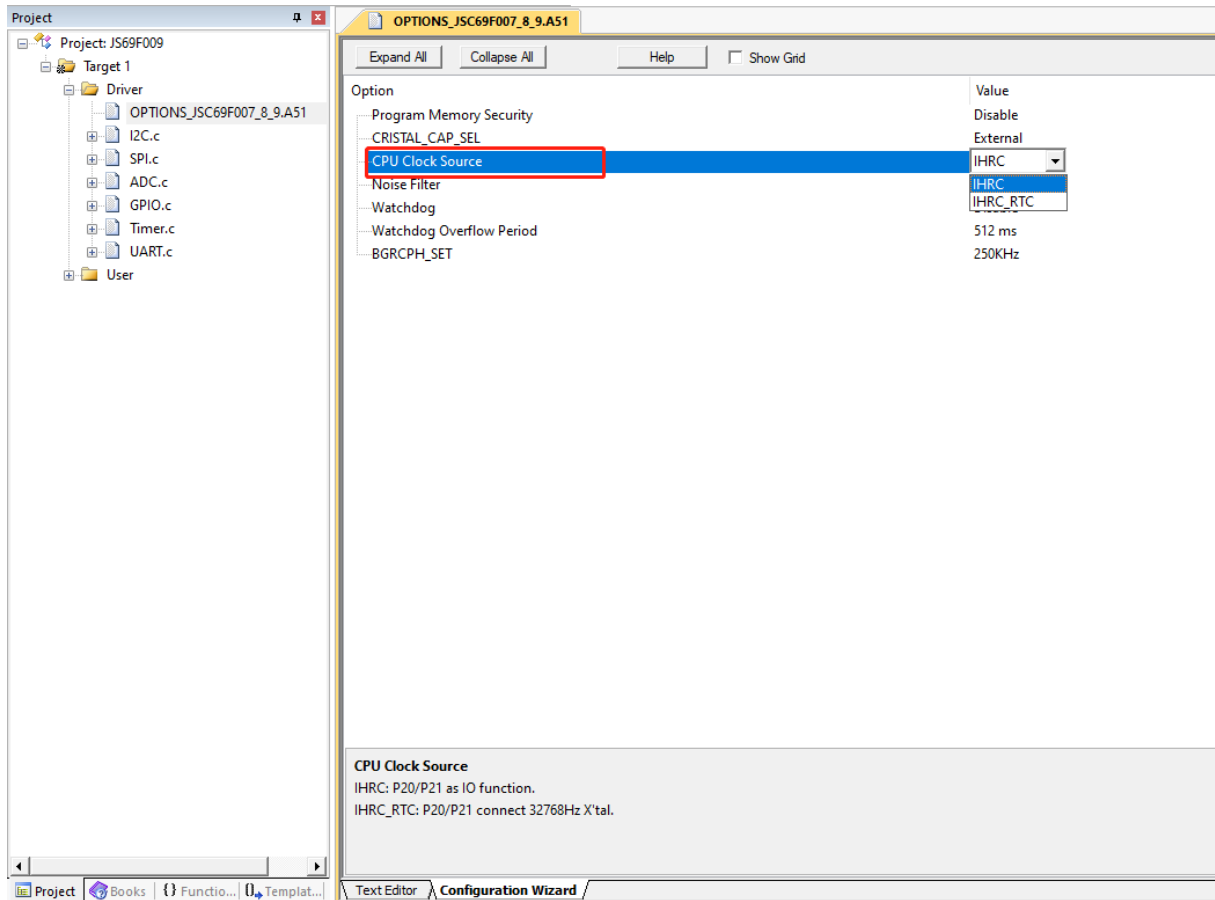
若 Program Memory Security 选择 Enable, 即使能加密功能时, 就无法读出 ROM 里面的内容, 所有的 ROM 地址都只能读到 0x00 的数据。

24.3 晶振电容选择



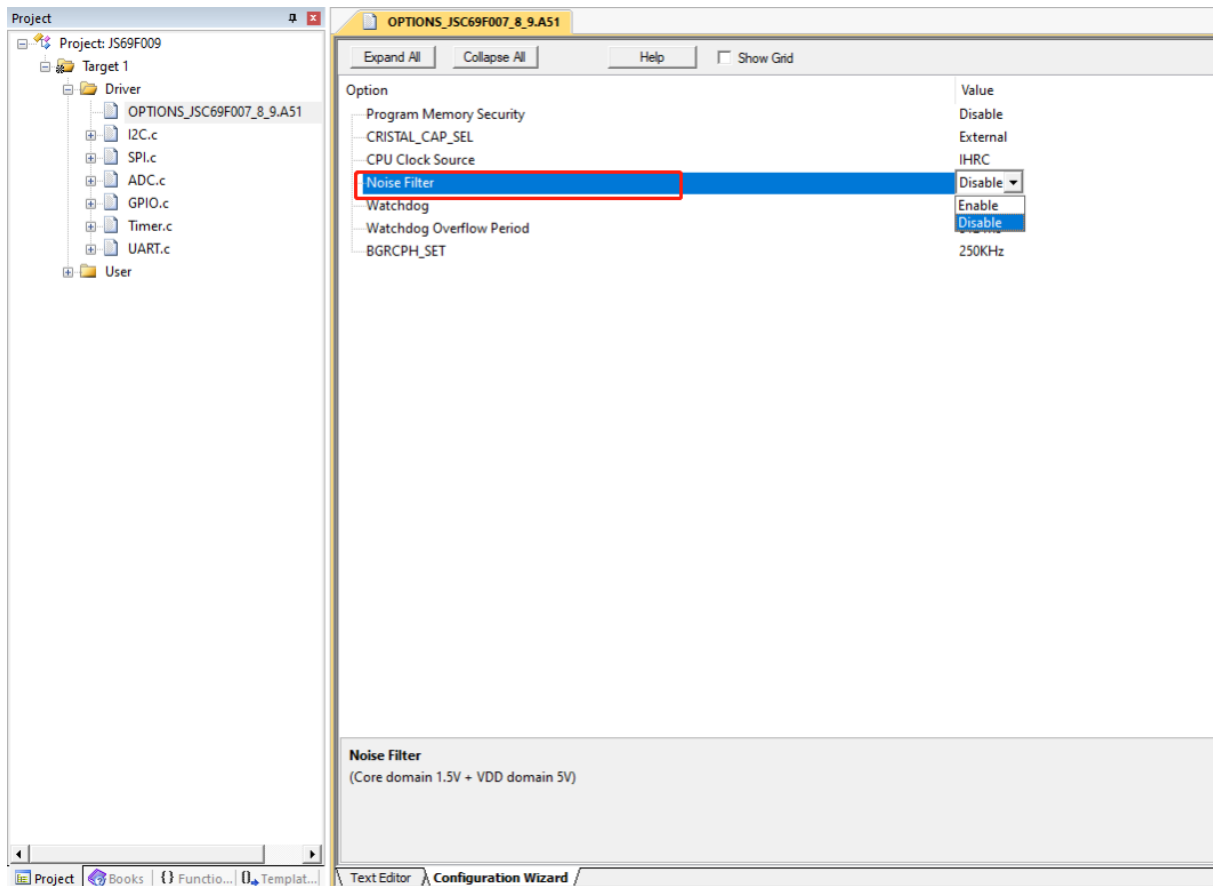
JSC69F009 内置晶振电容，若“CRISTAL_CAP_SEL”选择“Internal”，则选择接内部晶振电容；若选择“External”，则选择接外部晶振电容。

24.4 CPU 时钟源



若 CPU Clock Source 选择 IHRC，此时 CPU 时钟源为内部 32MHz 时钟发生器，且 P20、P21 为 GPIO 模式；若选择 IHRC_RTC，此时 CPU 时钟源为外部 32.768K 晶振,并且 P20、P21 连接 32.768K 晶振。

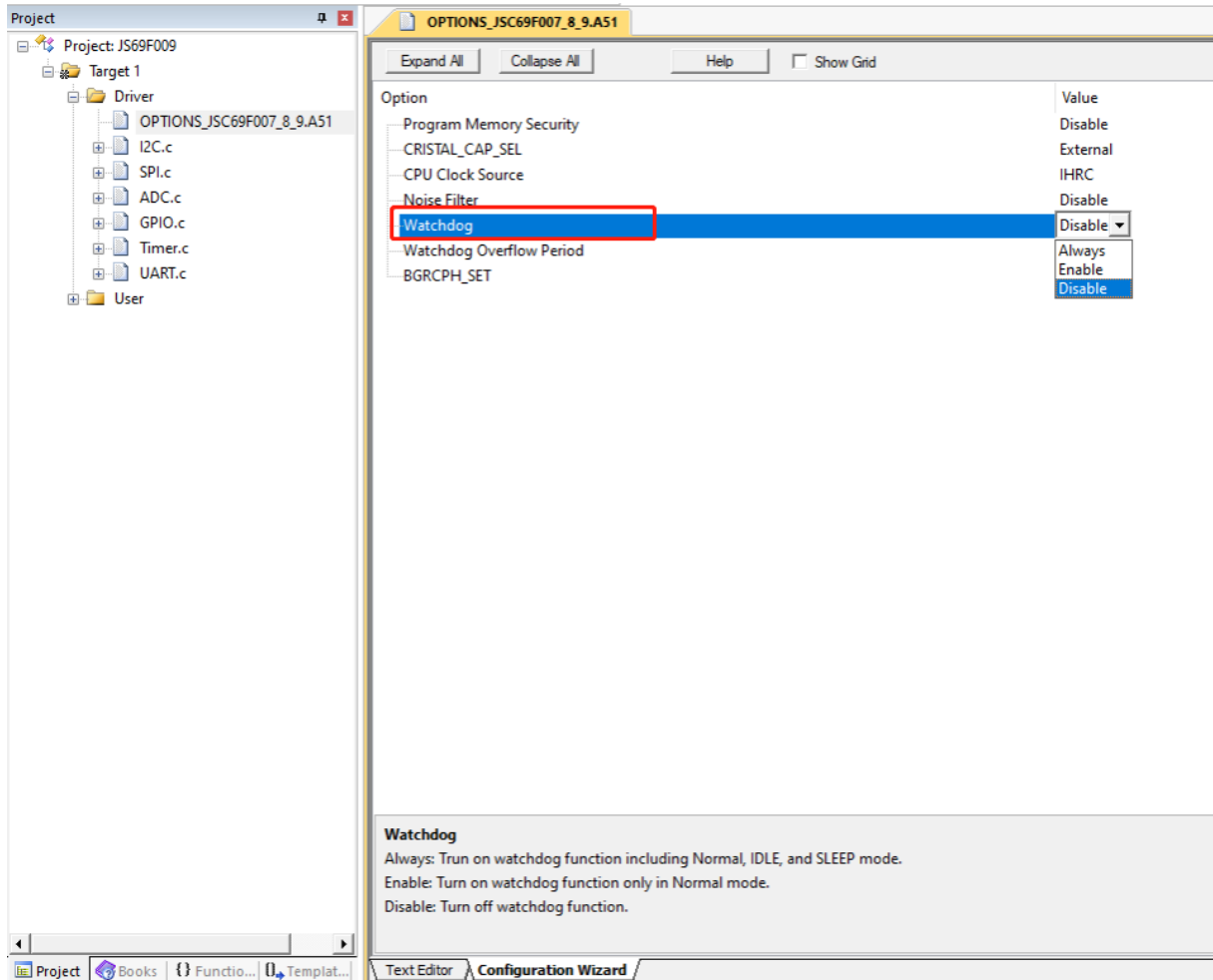
24.5 Noise Filter



MCU 必须设置“Noise filter Enable”在高 EFT 环境中增强抗干扰能力。

24.6 看门狗

24.6.1 看门狗模式选择

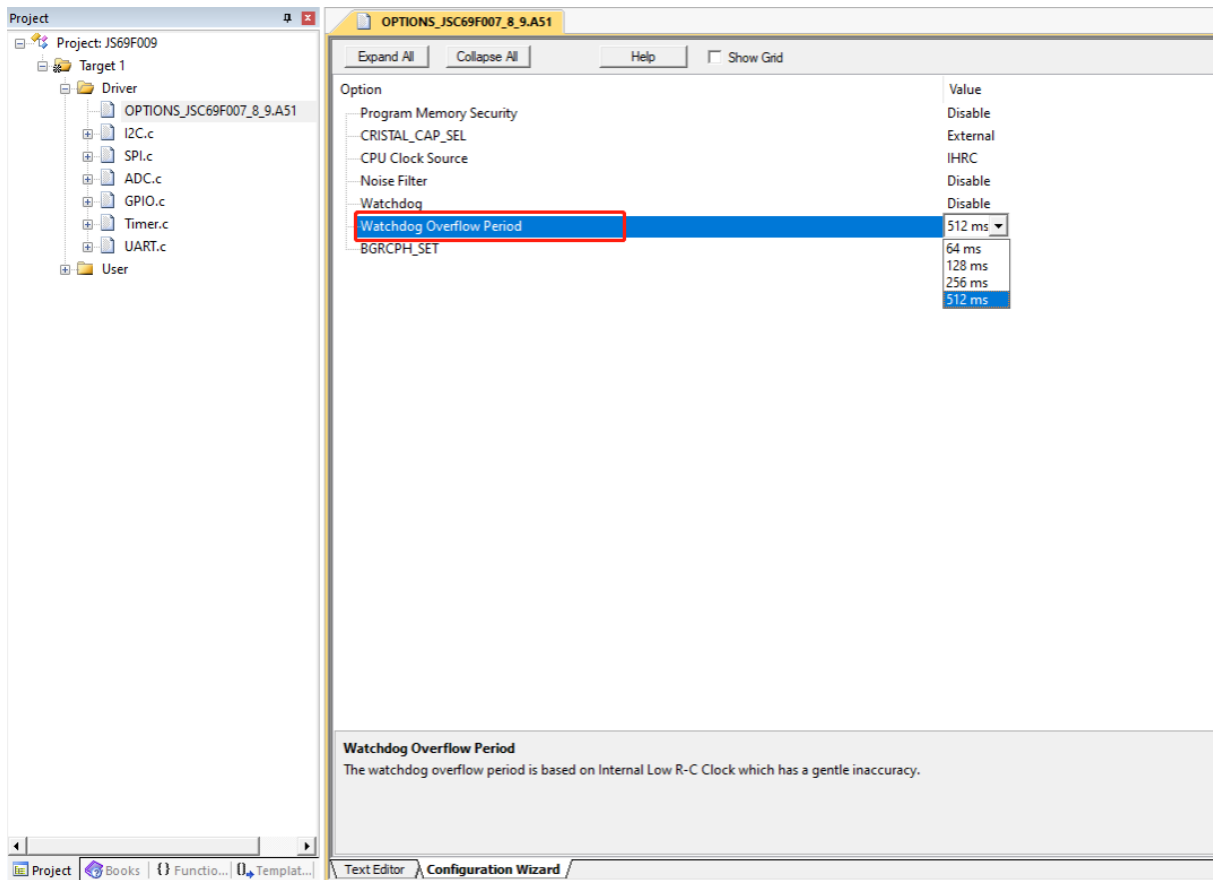


若 Watchdog 选择 Disable, 则处于 Disable mode, 其内部定时器在 CPU 的所有工作模式下都不计数, 在这种模式下不会触发看门狗复位。当看门狗一直处于工作状态时, 系统会消耗额外的功耗。

若 Watchdog 选择 Enable, 其内部定时器只在 CPU 的 Normal 模式下计数, 在 IDLE 和 SLEEP 模式下不会触发看门狗复位。

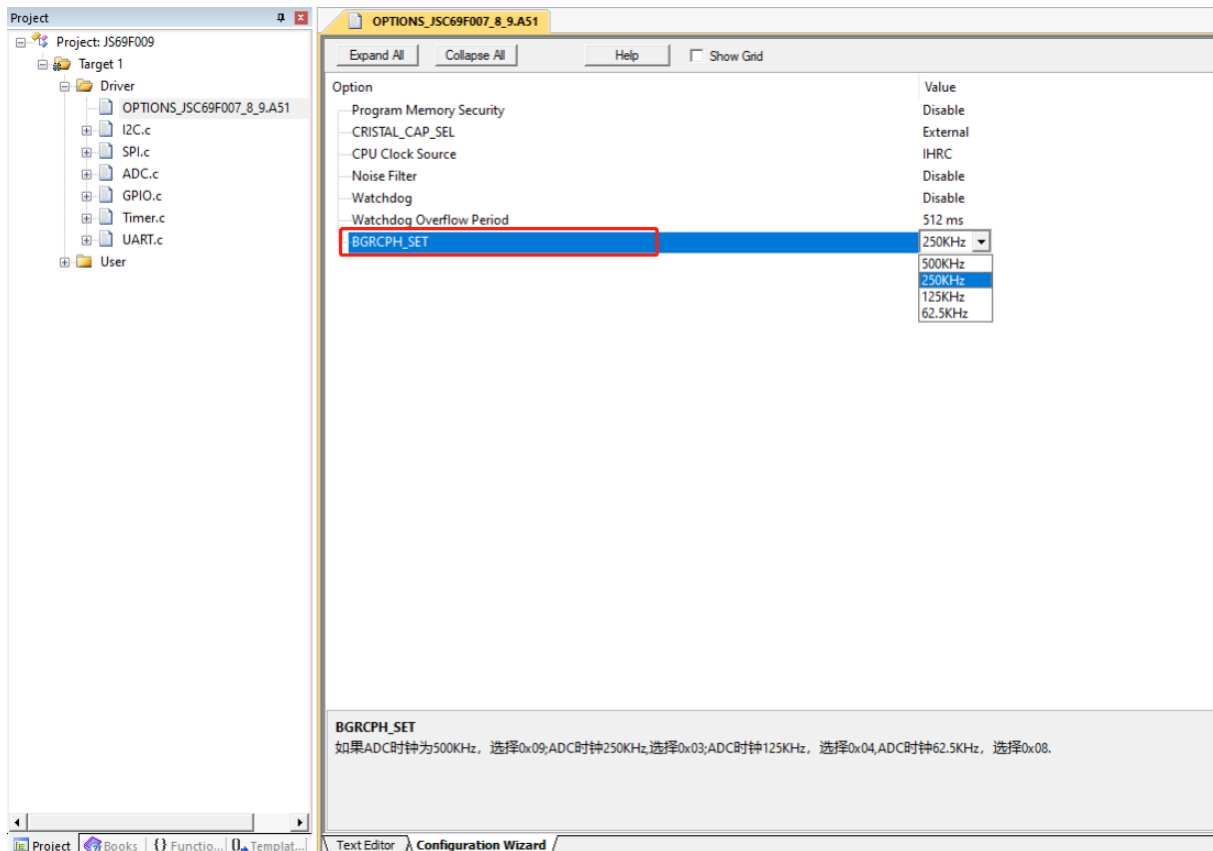
若 Watchdog 选择 Always, 其内部定时器在 CPU 的所有工作模式(NORMAL, IDLE, SLEEP)下都在计数。

24.6.2 看门狗溢出周期



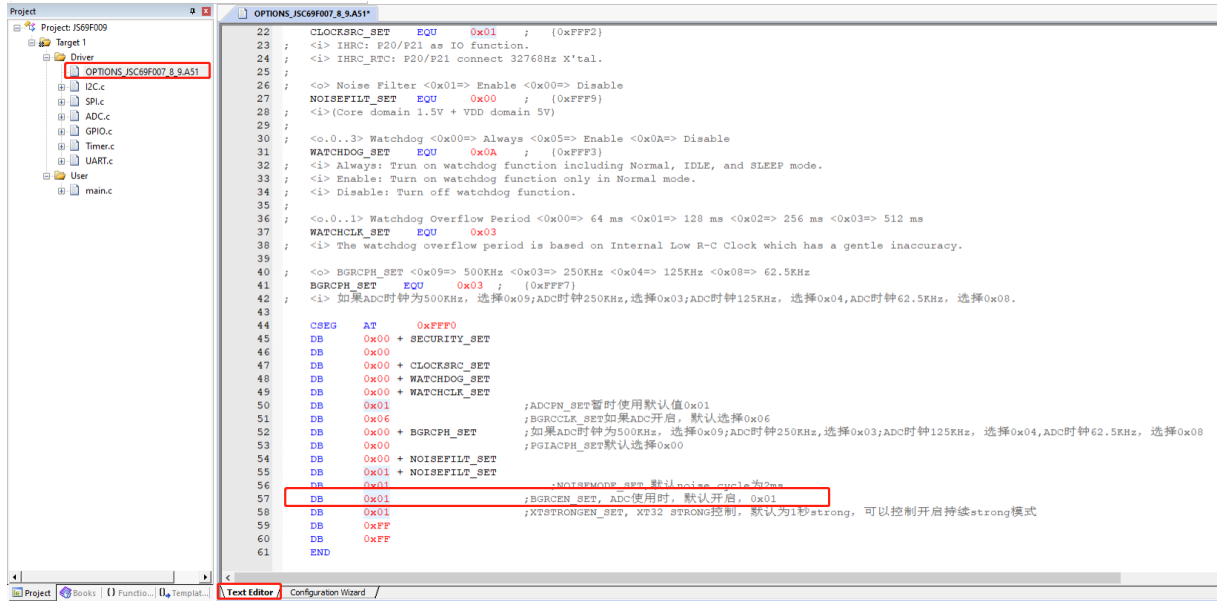
看门狗溢出周期，可根据用户需求选择 64ms、128ms、256ms 以及 512ms。

24.7 Bandgap chop 相位选择



该位需配合 ADC 时钟进行配置。若 ADC 时钟通过寄存器 ADCM2[5:4](ADCKS[1:0])选择 500KHz, 则 BGRCPH_SET 选择 500KHz, 此时该位配为 0x09; 若 ADC 时钟选择 250KHz, 则 BGRCPH_SET 选择 250KHz, 此时该位配为 0x03; 若 ADC 时钟选择 125KHz, 则 BGRCPH_SET 选择 125KHz, 此时该位配为 0x04; 若 ADC 时钟选择 62.5KHz, 则 BGRCPH_SET 选择 62.5KHz, 此时该位配为 0x08;

24.8 Bandgap Chop 时钟



此位为 bandgap chop 时钟使能位，设为 0x01 时，bandgap chop 时钟使能，设为 0x00 时，bandgap chop 时钟关闭。若使能 ADC 时，bandgap chop 需使能，即该位设为 0x01；其余情况下，均设为 0x00。

25 电气参数

25.1 极限参数

VDD 到 VSS 的电压·····	-0.3V to 6.0V
任意引脚到 VSS 电压·····	-0.3V to VDD+0.3V
运行环境温度普通版本·····	-40°C to 85°C
运行环境温度 N 版本·····	-40°C to 125°C
存储环境温度·····	-40°C to 125°C

25.2 系统操作特性

SYM.	Parameter	Test Condition	Min	Typ	Max	UNIT
VDD	运行电压	fcpu=1MHz	2.0		5.5	V
V _{DR}	RAM 数据保留电压		1.3			V
V _{POR}	VDD 上升率*		0.05			V/ms
I _{DD1}	Normal 模式功耗	VDD=3V~5V, fcpu=0.25MHz		1.25		mA
		VDD=3V~5V, fcpu=1MHz		1.35		mA
		VDD=3V~5V, fcpu=2MHz		1.45		mA
		VDD=3V~5V, fcpu=4MHz		1.65		mA
		VDD=3V~5V, fcpu=8MHz		1.95		mA
		VDD=3V~5V, fcpu=16MHz		2.65		mA
		VDD=3V~5V, fcpu=32MHz		3.35		mA
I _{DD2}	STOP 模式功耗	VDD=3V		3		μA
		VDD=5V		4	6	μA
		VDD=3V, RTC enable (32768Hz Crystal)		4.5		μA
I _{DD3}	IDLE 模式功耗 (fcpu=0.25MHz)	VDD=3V		1.2		mA
		VDD=5V		1.21		mA
I _{DD4}	IDLE 模式功耗 (fcpu=32MHz)	VDD=3V		1.95		mA
		VDD=5V		1.96		mA
F _{IHRC}	内部高速时钟发生器	VDD=2.0V~5.5V, 25°C	-1%	32	+1%	MHz
		VDD=2.0V~5.5V -40°C~85°C N: -40°C~125°C	-2%	32	-2%	MHz
V _{LVD16}	LVD16 检测电压	25°C	1.3	1.5	1.7	V
		-40°C~85°C	1.35	1.5	1.75	V
		N: -40°C~125°C				
ESD	Human body mode(HBM)			2	2.5	KV
	Machine mode(MM)			200	300	V
LU	Latch Up	VDD<6V	-400		+400	mA
		6V<VDD<6.5V	-50		+400	mA
		6V<VDD<6.5V, add 10Ω resistor in DVDD path	-400		+400	mA
EFT	电快速瞬变脉冲群(Fcpu= 1mips)	VL+			4500	V
		VL-			4500	
		VN+			4500	
		VN-			4500	
		VLN+			4500	
		VLN-			4500	

*未经验证的设计参考。环境温度是 25°C。

25.3 GPIO 特性

SYM.	Parameter	Test Condition	Min	Typ	Max	UNIT
V _{IL}	低电平输入电压	VDD=5V	VSS		0.3VDD	V
V _{IH}	高电平输入电压	VDD=5V	0.7VDD		VDD	V
I _{LEKG}	I/O 口输入漏电流	VIN=VDD			2	μA
R _{UP}	上拉电阻	VDD=3V	30	80	130	kΩ
		VDD=5V	25	50	75	kΩ
I _{OH}	I/O 输出灌电流	VDD=3V, VO=VDD-0.5V	18	20		mA
I _{OL2}	I/O 拉电流(P0, P10-P13)	VDD=3V, VO=VSS+0.5V	9	10		mA

25.4 ADC 特性

SYM.	DESCRIPTION	Condition	Min	Typ	Max	UNIT
I _{ADC}	工作电流	Run mode @ 2.4V		190		μA
I _{PDN}	睡眠电流	Stop mode @ 2.4V		0.1		μA
F _{SMP}	转换率 (WR)	ADC Clock=62.5KHz, OSR=32768		1.9		Hz
		ADC Clock=500KHz, OSR=64		7.8		kHz
T _{ADCSTL}	ADC 设置时间	3*(1/WR), if WR=61Hz, TADCSTL=3*16.4ms=49.2ms		3		1/WR
V _{REF}	参考电压输入电压	External VREF Input Range (R+~R-)	0.3		1.35	V
		Internal VREF Input Range.	0.3		1.35	V
V _R	ADC 参考信号绝对电压	GR=1, R+ and R- absolutely input Voltage	0.4		AVDDR-1V	V
		GR=0, R+ and R- absolutely input Voltage	0.4		AVDDR-1V	V
V _{AI}	ADC 输入信号绝对电压	GX=1, AI absolutely input Voltage	0.3		AVDDR-1V	V
		GX=0, AI absolutely input Voltage	0		AVDDR-1V	V
V _X	PGIA 输出信号绝对电压	X+ and X- absolutely output Voltage	0.3		AVDDR-1V	
	PGIA 增益比	VDD=5V, PGIA=x128	-3		+3	%
DNL	微分非线性	ADC range±131072 x0.9.(0.9 x Vref, 18-bits)		±2		LSB
INL	积分非线性	ADC range±131072 x0.9.(0.9 x Vref, 18-bits)		±4		LSB
NMC	无丢码	ADC range±131072 x0.9.(0.9 x Vref, 24-bits)		18		bit
NFB	Noise free bits	Gain:1, Vref:0.8V, OSR:32768, Input-short		21.9		bit
		Gain=128, Vref=0.8V, OSR:32768, Input-short		18.29		bit
ENOB	有效位数	Gain:1, Vref:0.8V, OSR:32768, Input-short		24.6		bit
		Gain=128, Vref=0.8V, OSR:32768, Input-short		21		bit
V _{AIN}	ADC 输入微分范围	ADC input signal, signal after PGIA application	0.3		1.44	V

T _{Drift}	ADC 温度漂移	AVDDR=2.7V, PGIA x 16, T=0~50°C		30		PP M/°C
--------------------	----------	---------------------------------	--	----	--	------------

所有测试条件: Input-Short, Vref=0.81V, Gain=PGIA x ADC, Collect 1024 ADC data.

- 1) Noise Free 分辨率=Log2 (Full Scale Range/Peak-Peak Noise)
- 2) where 满量程=2 x Vref/Gain (ex.Vref=0.81V, Gain=1x~128x)
- 3) 有效分辨率=Log2 (Full Scale Range/RMS_Noise)
- 4) RMS Noise=σx LSB_Resolution where LSB_Resolution=Full Scale Range/2^{Bit}, Bit=24
σ=standard deviation of 1024 ADC output data.
- 5) Peak-Peak Noise = 6.6 x RMS Noise, or code variation range x LSB_Resolution where Code variation range=ADC counts max-min of 1024 data.

JSC69F007/8/9 ADC Performance Noise Free Bit vs Output Rate and Gain(ADC 时钟 500KHz)								
OSR	32768	16384	8192	4096	1024	256	128	64
WR	15Hz	31Hz	61Hz	122Hz	488Hz	1.9KHz	3.9KHz	7.8KHz
128x1	14.68	14.32	13.81	13.33	12.36	11.36	10.82	10.3
64x1	15.68	15.29	14.78	14.26	13.34	12.28	11.83	11.07
32x1	16.54	15.94	15.47	14.94	13.97	12.91	12.43	11.5
16x1	17.16	16.74	16.24	15.67	14.57	13.6	12.95	11.83
8x1	17.74	17.26	16.74	16.2	15.13	14.05	13.32	11.95
4x1	17.98	17.5	17.05	16.44	15.31	14.17	13.35	12.12
1x1	18.44	18.07	17.58	16.89	15.81	14.48	13.69	12.3

JSC69F007/8/9 ADC Performance Noise Free Bit vs. Output Rate and Gain(ADC 时钟 250KHz)								
OSR	32768	16384	8192	4096	1024	256	128	64
WR	7.6Hz	15Hz	30.5Hz	61Hz	244Hz	976Hz	1.9KHz	3.9KHz
128x1	18.29	16.64	15.4	14.3	12.79	11.65	11.3	10.44
64x1	19.71	18.19	16.81	15.6	13.89	12.6	12	11.35
32x1	20.32	18.83	17.47	16.3	14.5	13.23	12.63	11.96
16x1	20.68	19.29	18.11	16.9	15.13	13.86	13.25	12.41
8x1	21.14	19.73	18.29	17.12	15.26	14.05	13.39	12.56
4x1	21.34	20	18.74	17.4	15.69	14.32	13.87	12.88
1x1	21.92	20.65	19.34	18.16	16.43	15.1	14.23	13

JSC69F007/8/9 ADC Performance Noise Free Bit vs. Output Rate and Gain(ADC 时钟 125KHz)								
OSR	32768	16384	8192	4096	1024	256	128	64
WR	3.8Hz	7.6Hz	15Hz	30.5Hz	122Hz	488Hz	976Hz	1.9KHz
128x1	19.64	18.22	16.78	15.32	13.19	11.75	11.15	10.5
64x1	21.38	19.92	18.48	16.99	14.48	12.8	12.21	11.48
32x1	21.89	20.62	19.14	17.71	15.22	13.52	12.9	11.96
16x1	22.24	21.2	19.79	18.38	15.91	14.23	13.53	12.67
8x1	22.27	21.59	20.12	18.82	16.23	14.56	13.78	12.8
4x1	22.27	21.73	20.32	18.89	16.44	14.86	14.11	12.78
1x1	22.27	22.05	20.76	19.36	17	15.3	14.46	13.3

JSC69F007/8/9 ADC Performance Noise Free Bit vs. Output Rate and Gain(ADC 时钟 62.5KHz)								
OSR	32768	16384	8192	4096	1024	256	128	64
WR	1.9Hz	3.8Hz	7.6Hz	15Hz	61Hz	244Hz	488Hz	976Hz
128x1	20.78	19.37	17.83	16.36	13.69	11.83	11.16	10.47
64x1	22.12	20.93	19.4	18.02	15.22	13.08	12.32	11.59
32x1	22.27	21.61	20.17	18.77	15.91	13.89	13.06	12.12
16x1	22.27	22.09	20.78	19.34	16.58	14.41	13.69	12.44
8x1	22.27	22.25	21.22	19.79	16.99	14.84	14.12	13.14
4x1	22.27	22.24	21.29	19.84	17	14.97	14.11	13.13
1x1	22.27	22.17	21	19.6	17.23	15.3	14.37	13.05

* Buffer off (GX=0, GR=0)

25.5 Charge Pump Regulator 特性

SYM.	PARAMETER	Condition	Min	Typ	Max	UNIT
V _{oper}	工作电压 (Charge Pump 输入)	使能 Pump, 输入电压范围(AVDD)	2.0		5.5	V
		禁止 Pump, 输入电压范围(AVDD)	2.0		5.5	V
V _{AVDDCP}	Charge pump 输出范围	Charge Pump 关, AVDD 从 2V 到 5.5V AVDDCP=AVDD	2.0		5.5	V
		Busk-boost 模式, AVDD 从 2V 到 5.5V AVDDCP=3.2V~4.1V(AVDDR+0.5V)	AVDDR+0.5V			V
		2x Pump Mode, AVDD from 2V~3.3V AVDDCP=4V~6.6V	2x AVDD			V
		2x Pump Mode, AVDD from 3.3V~5.5V AVDDCP=AVDD.	AVDD			V
I _{AVDDCP}	Charge pump 输出功耗	Charge Pump 打开或关闭		4.5	7	mA
I _{PUMP}	Charge pump 固有功耗	Busk-boost 模式, AVDD=3V		200		uA
I _{DPUMP}	Charge pump 驱动能力	Busk-boost 模式, AVDD=3V		2.5		mA

ACM

SYM.	PARAMETER	Condition	Min	Typ.	Max	UNIT
V _{ACM}	模拟通用电压	输出电压	0.9	1	1.1	V
		输出电压偏移. 温度 $\Delta 10^{\circ}\text{C}$		± 0.1		%
		输出电压偏移. AVDD(2V~6.5V)		± 0.1		%
I _{SRC}	V _{ACM} 驱动能力	仅为 ADC 使用			10	uA
I _{SNK}	V _{ACM} 反向能力	仅为 ADC 使用			1	mA

AVDDR

SYM.	PARAMETER	Condition	Min	Typ.	Max	UNIT
V _{AVDDR}	Regulator 输出电压 AVDDR	AVDDR 输出范围=2.7V, 3.0V, 3.3V, 3.6V	2.7		3.6	V
		AVDDR 设置 2.7V	2.65	2.7	2.75	V
		AVDDR 设置 3.0V	2.95	3.0	3.05	V
		AVDDR 设置 3.3V	3.25	3.3	3.35	V
		AVDDR 设置 3.6V	3.55	3.6	3.65	V
		AVDDR Load regulation $\Delta 5\text{mA}$		AVDD R ± 0.0 5V		V
		输出电压偏移. 温度 $\Delta 10^{\circ}\text{C}$		± 0.1		%
		输出电压偏移. AVDD(2V~6.5V)		± 0.1		%
I _{AVDDR}	AVDDR 驱动/反向电流				10	mA
I _{QI}	静态电流	Pump+AVDDR+ACM		200		uA

25.6 OP-Amp 电气特性

SYM.	PARAMETER	DESCRIPTION	Min	Typ.	Max	UNIT
V _{oper}	工作电压	Power source from AVDDR	2.7		3.6	V
V _{OS}	输入补偿电压			±3		mV
I _{oper}	工作电流	Per OP-Amp		150		uA
V _{IN}	模拟输入电压范围	OP+/OP-	0.05		AVDDR-0.1	V
V _{out}	模拟输出电压范围	OPOUT	0.05		AVDDR-0.1	V
I _{OH} /I _{OL}	输出短路电流	Unit Gain Buffer. Vo=0V~3.2V, AVDDR=3.3V		±5		mA
R _{BUF}	缓冲模式开关	AVDDR 3.3V, VOUT=1.65V.		50		Ω
GB	增益带宽	RL=300KΩ, CL=50pF		1.4		MHz
SR	转换率	10%to90%		0.8		V/uS
T _{ON}	启动时间			20		uS

25.7 比较器电气特性

SYM.	DESCRIPTION	PARAMETER	Min	Typ.	Max	UNIT
V _{ILBT}	内部低电压检测	条件: LBTSEL[3:0]=1000, VLBT=2.2V	2.1	2.2	2.3	V
		条件: LBTSEL[3:0]=1100, VLBT=3V	2.75	3.0	3.15	
		条件: LBTSEL[3:0]=1111, VLBT=3.6V	3.45	3.6	3.75	
V _{ELBT}	外部低电压检测	条件: LBTSEL[3:0]=0xxx, VLBT=P10 input.	1.15	1.25	1.35	
V _{HY}	比较器迟滞窗口			200		mV
I _{COMP}	电流消耗	AVDD=3V		50		uA

25.8 LCD 电气特性

SYM.	DESCRIPTION	PARAMETER	Min	Typ.	Max	UNIT
L _{LCD}	C-Type LCD 工作电流	VDD:3~5V,无面板低功耗模式(BGM=0,DUTY[1:0]=00)		13		uA
		VDD:3~5V,无面板低功耗模式(BGM=0,DUTY[1:0]=01)		14		
		VDD:3~5V,无面板低功耗模式(BGM=0,DUTY[1:0]=10)		15		
		VDD:3~5V,无面板低功耗模式(BGM=0,DUTY[1:0]=11)		19		
		VDD:3~5V,无面板低功耗模式(BGM=1,DUTY[1:0]=xx)		130		
V _{LCD}	C-Type VLCD 输出电压	VDD: 2.0~5.5V, VLCD set 3V (VCP[2:0]=010)	2.8	3.0	3.2	V
		VDD: 2.0~5.5V, VLCD set 3V (VCP[2:0]=010)	4.3	4.5	4.7	

25.9 Flash 存储器特性

SYM.	Parameter	Test Condition	Min	TYP	MAX	UNIT
V _{dd}	工作电压		2.0		5.5	V
T _{en}	endurance	25°C			100K	cycle
I _{wrt}	写功耗	25°C		3	3.5	mA
T _{wrt}	写时间	Write 128 bytes, 25°C		3		ms

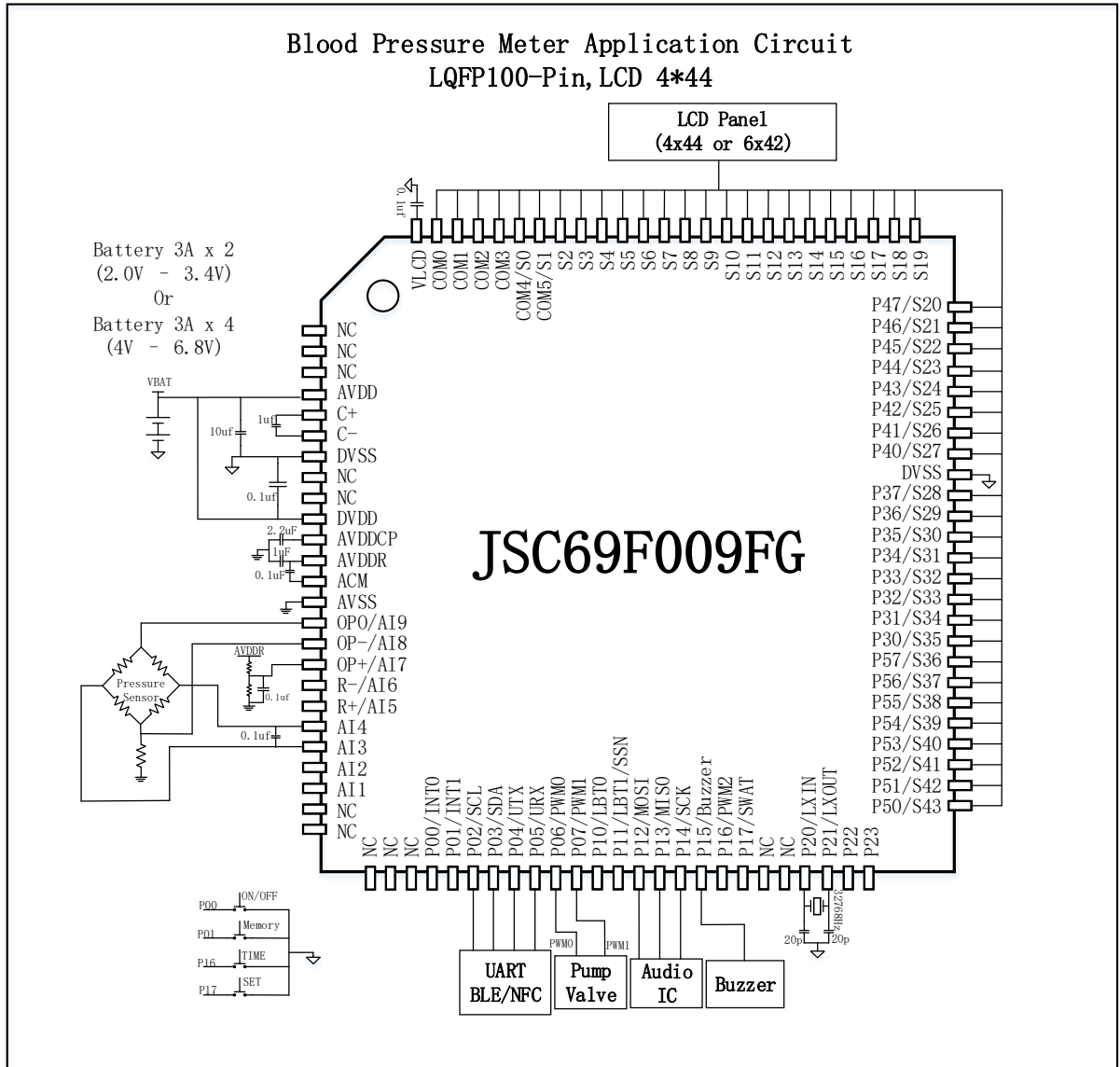
25.10 特殊条件或环境中使用的建议

条件/环境	建议
条件(1) 当MCU I/O 连接到其他电路,它有负瞬时电压出现在 I/O 上,峰值电压-3V,MCU 可能发生延迟现象。	连接 10-Ohm 电阻在电源输入 MCU 的 DVDD pin 之前,这将解决在 I/O 中出现的异常瞬态电压出现的延迟。
条件(2) 当 MCU 电源上有高干扰且 MCU 功能必须好好工作时。(高 EFT 干扰 MCU)	MCU 必须设置“Noise filter Enable”在高 EFT 环境中增强抗干扰能力。

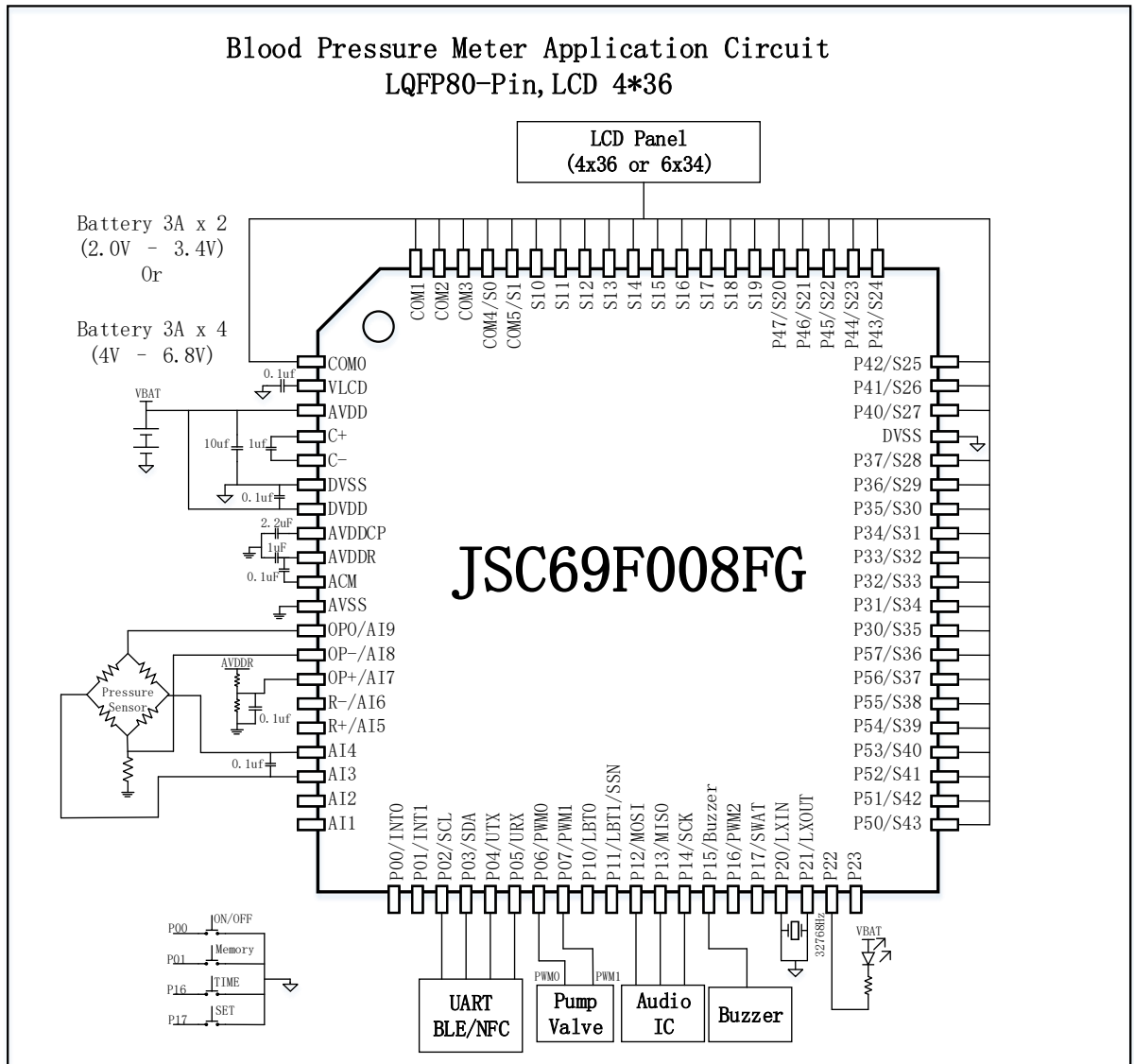
26 系统应用

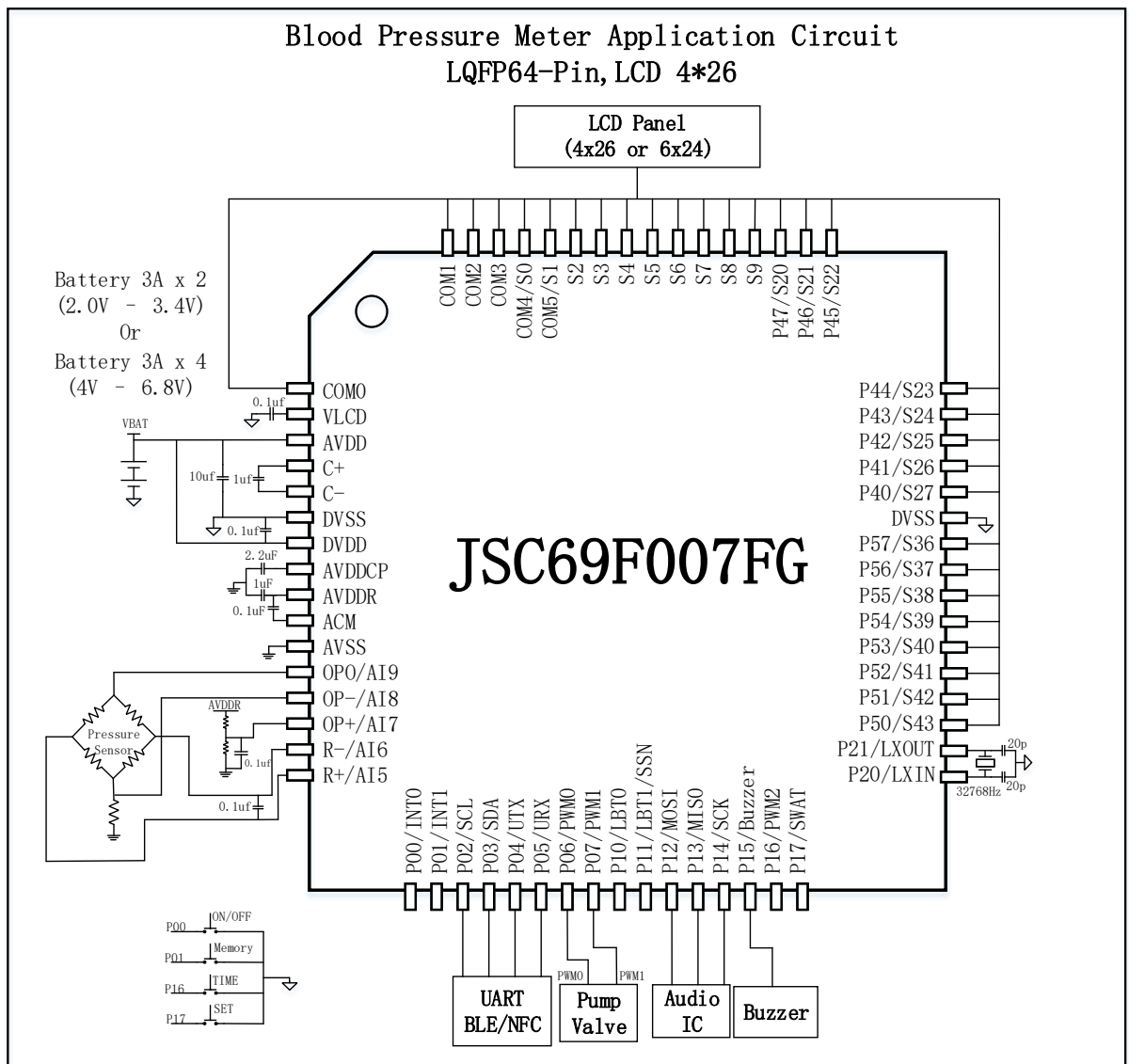
JSC69F007/8/9 芯片的典型应用图如下所示:

血压计应用电路:



注: 晶振负载电容做在芯片内部, 用户可根据需求设置 option 选择带内容大小 (可选 2pF-20pF)





27 指令集

本章对 JSC69F007/8/9 的综合汇编指令进行分类, 总共包括 5 类: 算术运算、逻辑运算、数据传送运算、布尔操作和程序分支指令, 这些指令全部都与标准 8051 兼容。

27.1 符号说明

符号	功能描述
Rn	工作寄存器 R0-R7
direct	直接地址: 128 个内部 RAM 地址之一或特殊功能寄存器地址
@Ri	通过寄存器 R0 或者 R1 来间接寻址内部或外部 RAM 地址
#data	8 位常量 (立即操作数)
#data16	16 位常量 (立即操作数)
bit	位于内部 RAM 中的 128 个位变量之一, 或者是可以位操作的特殊功能寄存器当中的任何标志位
addr16	LCALL 和 LJMP 的目标地址, 可以是 64KB 程序存储空间当中的任意地址
addr11	ACALL 和 AJMP 的目标地址, 与下一条指令的第一个字节在同一个 2KB 区内 (即 16 位地址中的高 5 位地址相同)
rel	SJMP 和所有条件跳转指令中的一个 8 位偏移字节, 相对于下一条指令的第一个字节, 可偏移的范围为+127/-128 字节
A	累加器

27.2 运算指令

指令	功能描述
ADD A, Rn	将累加器与寄存器的内容相加, 结果存回累加器
ADD A, direct	将累加器与直接地址的内容相加, 结果存回累加器
ADD A, @Ri	将累加器与间接地址的内容相加, 结果存回累加器
ADD A, #data	将累加器与常量相加, 结果存回累加器
ADDC A, Rn	将累加器与寄存器的内容及进位 C 相加, 结果存回累加器
ADDC A, direct	将累加器与直接地址的内容及进位 C 相加, 结果存回累加器
ADDC A, @Ri	将累加器与间接地址的内容及进位 C 相加, 结果存回累加器
ADDC A, #data	将累加器与常量及进位 C 相加, 结果存回累加器
SUBB A, Rn	将累加器的值减去寄存器的值减借位 C, 结果存回累加器
SUBB A, direct	将累加器的值减去直接地址的值减借位 C, 结果存回累加器
SUBB A, @Ri	将累加器的值减去间接地址的值减借位 C, 结果存回累加器
SUBB A, #data	将累加器的值减去常量减借位 C, 结果存回累加器
INC A	将累加器的值加 1
INC Rn	将寄存器的值加 1
INC direct	将直接地址的内容加 1
INC @Ri	将间接地址的内容加 1
INC DPTR	将数据指针寄存器的值加 1
DEC A	将累加器的值减 1
DEC Rn	将寄存器的值减 1
DEC direct	将直接地址的内容减 1
DEC @Ri	将间接地址的内容减 1
MUL AB	将累加器的值与 B 寄存器的值相乘, 乘积的低字节存回累加器, 高字节存回 B 寄存器
DIV	将累加器的值除以 B 寄存器的值, 结果的商存回累加器, 余数存回 B 寄存器
DA A	将累加器的值作十进制调整

27.3 逻辑指令

指令	功能描述
ANL A,Rn	将累加器的值与寄存器的值作逻辑与运算, 结果存回累加器
ANL A,direct	将累加器的值与直接地址的内容作逻辑与运算, 结果存回累加器
ANL A,@Ri	将累加器的值与间接地址的内容作逻辑与运算, 结果存回累加器
ANL A,#data	将累加器的值与常量作逻辑与运算, 结果存回累加器
ANL direct,A	将直接地址的内容与累加器的值作逻辑与运算, 结果存回直接地址
ANL direct,#data	将直接地址的内容与常量作逻辑与运算, 结果存回直接地址
ORL A,Rn	将累加器的值与寄存器的值作逻辑或运算, 结果存回累加器
ORL A,direct	将累加器的值与直接地址的内容作逻辑或运算, 结果存回累加器
ORL A,@Ri	将累加器的值与间接地址的内容作逻辑或运算, 结果存回累加器
ORL A,#data	将累加器的值与常量作逻辑或运算, 结果存回累加器
ORL direct,A	将直接地址的内容与累加器的值作逻辑或运算, 结果存回直接地址
ORL direct,#data	将直接地址的内容与常量作逻辑或运算, 结果存回直接地址
XRL A,Rn	将累加器的值与寄存器的值作逻辑异或运算, 结果存回累加器
XRL A,direct	将累加器的值与直接地址的内容作逻辑异或运算, 结果存回累加器
XRL A,@Ri	将累加器的值与间接地址的内容作逻辑异或运算, 结果存回累加器
XRL A,#data	将累加器的值与常量作逻辑异或运算, 结果存回累加器
XRL direct,A	将直接地址的内容与累加器的值作逻辑异或运算, 结果存回直接地址
XRL direct,#data	将直接地址的内容与常量作逻辑异或运算, 结果存回直接地址
CLR A	将累加器的值清零
CPL A	将累加器的值取反
RL A	将累加器的值左移一位
RLC A	将累加器的值带进位 C 左移一位
RR A	将累加器的值右移一位
RRC A	将累加器的值带进位 C 右移一位
SWAP A	将累加器的高 4 位与低 4 位的内容交换

27.4 数据传输指令

指令	功能描述
MOV A,Rn	将寄存器的内容赋值给累加器
MOV A,direct	将直接地址的内容赋值给累加器
MOV A,@Ri	将间接地址的内容赋值给累加器
MOV A, #data	将常量赋值给累加器
MOV Rn,A	将累加器的内容赋值给寄存器
MOV Rn,direct	将直接地址的内容赋值给寄存器
MOV Rn,#data	将常量赋值给寄存器
MOV direct,A	将累加器的内容赋值给直接地址
MOV direct,Rn	将寄存器的内容赋值给直接地址
MOV direct1,direct2	将直接地址 2 的内容赋值给直接地址 1
MOV direct,@Ri	将间接地址的内容赋值给直接地址
MOV direct,#data	将常量赋值给直接地址
MOV @Ri,A	将累加器的内容赋值给间接地址
MOV @Ri,direct	将直接地址的内容赋值给间接地址
MOV @Ri,#data	将常量赋值给间接地址
MOV DPTR, #data16	16 位常量赋值给数据指针寄存器
MOVC A,@A+DPTR	累加器的值加数据指针寄存器的值作为其所指定地址, 将该地址的内容赋值给累加器
MOVC A,@A+PC	累加器的值加程序计数器的值作为其所指定地址, 将该地址的内容赋值给累加器
MOVX A,@Ri	将间接地址所指定外部存储器的内容赋值给累加器 (8 位地址)
MOVX A,@DPTR	将数据指针所指定外部存储器的内容赋值给累加器 (16 位地址)
MOVX @Ri A	将累加器的内容赋值给间接地址所指定的外部存储器 (8 位地址)
MOVX @DPTR,A	将累加器的内容赋值给数据指针所指定的外部存储器 (16 位地址)
PUSH direct	将直接地址的内容压入堆栈区
POP direct	将栈顶的内容弹出至直接地址
XCH A,Rn	将累加器与寄存器的内容互换
XCH A,direct	将累加器与直接地址的内容互换
XCH A,@Ri	将累加器与间接地址的内容互换
XCHD A,@Ri	将累加器的低 4 位与间接地址的低 4 位互换

27.5 布尔运算指令

指令	功能描述
CLR C	将进位 C 清零
CLR bit	将直接寻址位清零
SETB C	设置进位 C 为 1
SETB bit	设置直接寻址位为 1
CPL C	将进位 C 的值取反
CPL bit	将直接寻址位的值取反
ANL C,bit	将进位 C 的值与直接寻址位的值作逻辑与运算, 结果存回进位 C
ANL C,/bit	将进位 C 的值与直接寻址位的反码作逻辑与运算, 结果存回进位 C
ORL C,bit	将进位 C 的值与直接寻址位的值作逻辑或运算, 结果存回进位 C
ORL C,/bit	将进位 C 的值与直接寻址位的反码作逻辑或运算, 结果存回进位 C
MOV C, bit	将直接寻址位的值赋值给进位 C
MOV bit,C	将进位 C 的值赋值给直接寻址位

27.6 程序跳转指令

指令	功能描述
ACALL addr11	绝对调用
LCALL addr16	长调用
RET	从子程序返回
RETI	从中断返回
AJMP addr11	绝对跳转
LJMP addr16	长跳转
SJMP rel	短跳转 (相对地址)
JMP @A+DPTR	跳至累加器的值加数据指针寄存器的值所指定的地址
JZ rel	若累加器的值为 0, 则跳转至 rel 所指定的地址
JNZ rel	若累加器的值不为 0, 则跳转至 rel 所指定的地址
JC rel	若进位 C 的值为 1, 则跳转至 rel 所指定的地址
JNC rel	若进位 C 的值为 0, 则跳转至 rel 所指定的地址
JB bit,rel	若直接寻址位的值为 1, 则跳转至 rel 所指定的地址
JNB bit,rel	若直接寻址位的值为 0, 则跳转至 rel 所指定的地址
JBC bit,rel	若直接寻址位的值为 1, 则跳转至 rel 所指定的地址, 并将该位值清零
CJNE A,direct,rel	将累加器的内容与直接地址的内容比较, 若不相等则跳转至 rel 所指定的地址
CJNE A,#data,rel	将累加器的内容与常量比较, 若不相等则跳转至 rel 所指定的地址
CJNE Rn,#data,rel	将寄存器的内容与常量比较, 若不相等则跳转至 rel 所指定的地址
CJNE @Ri,#data,rel	将间接地址的内容与常量比较, 若不相等则跳转至 rel 所指定的地址
DJNZ Rn,rel	将寄存器的内容减 1, 若不等于 0 则跳转至 rel 所指定的地址
DJNZ direct,rel	将直接地址的内容减 1, 若不等于 0 则跳转至 rel 所指定的地址
NOP	空指令

28 JSC69F007/8/9 Start-Kit

29 模拟设置与应用

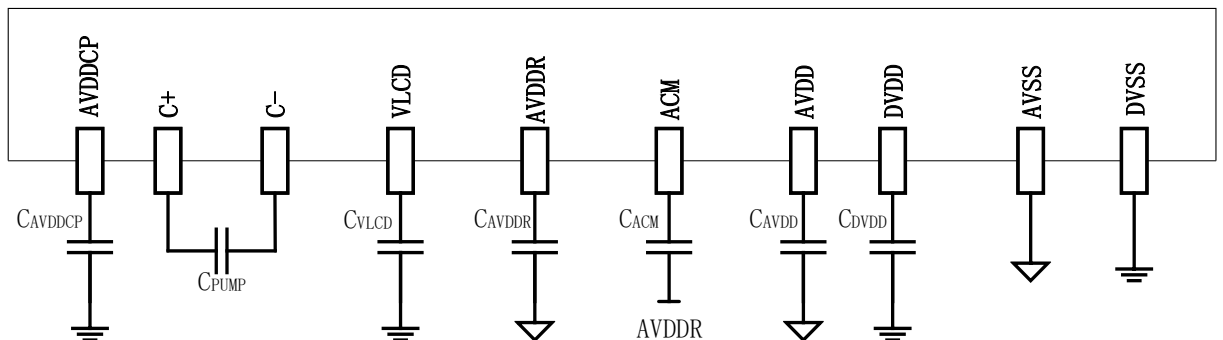
JSC69F007/8/9 应用于许多 DC 测量应用程序, 比如体重秤, 压力测量, 血压测量仪和温度计。

下表显示了不同的应用程序设置, MCU 电源来自 R2032 电池、AA/AAA 干电池或外部调节器。

Charge Pump 使能电容表:

Power	AVDDR	ACM	AVDDCP	C+/-	VLCD	AVDD	DVDD
	C _{AVDDR}	C _{ACM}	C _{AVDDCP}	C _{PUMP}	C _{VLCD}	C _{AVDD}	C _{DVDD}
CR2032 (2.4~3.6V)	0.47uf	0.1uf	2.2uf	1uf	0.1uf	10uf/104	10uf/104
2*AA/AAA Bat (2.4~3.6V)	1uf	0.1uf	2.2uf	1uf	0.1uf	10uf/104	10uf/104
4*AA/AAA Bat (4~5.5V)							

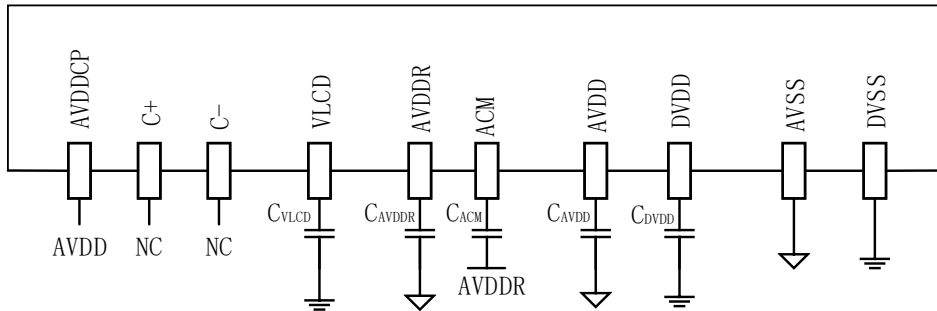
模拟电容连接 (Charge Pump 使能) :



Charge Pump 禁止电容表 (Charge Pump 禁止) :

Power type	AVDDR	ACM	AVDDCP	C+/-	VLCD	AVDD	DVDD
	C _{AVDDR}	C _{ACM}	C _{AVDDCP}	C _{PUMP}	C _{VLCD}	C _{AVDD}	C _{DVDD}
4*AA/AAA Bat (4V~5.5V)	1uf	0.1uf	NC*	NC	0.1uf	10uf/104	10uf/104

模拟电容连接（Charge Pump 禁止）：



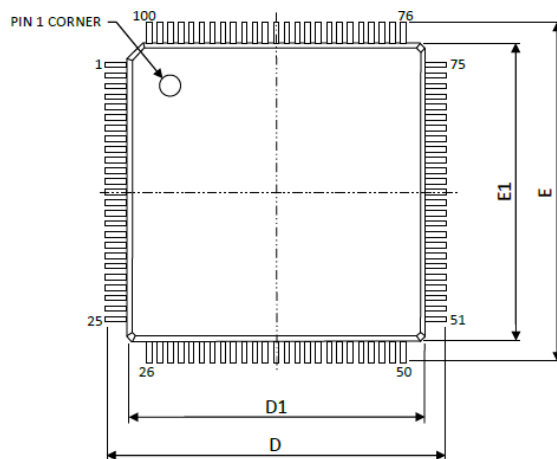
*AVDDCP Pin Short to AVDD

30 订购信息

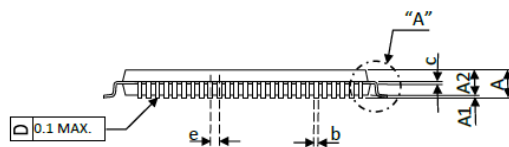
30.1 命名规则

单片机全称	封装类型	适应温度
JSC69F009W	Wafer	-40℃~80℃
JSC69F009H	Dice	-40℃~80℃
JSC69F009FG	LQFP, 100 脚, 绿色封装	-40℃~80℃
JSC69F008FG	LQFP, 80 脚, 绿色封装	-40℃~80℃
JSC69F007FG	LQFP, 64 脚, 绿色封装	-40℃~80℃
JSC69F009LQ	LQFP, 100 脚, 绿色封装	-40℃~105℃
JSC69F007LQ	LQFP, 64 脚, 绿色封装	-40℃~105℃
JSC69F009WN	Wafer	-40℃~125℃
JSC69F009HN	Dice	-40℃~125℃
JSC69F009FGN	LQFP, 100 脚, 绿色封装	-40℃~125℃
JSC69F008FGN	LQFP, 80 脚, 绿色封装	-40℃~125℃
JSC69F007FGN	LQFP, 64 脚, 绿色封装	-40℃~125℃

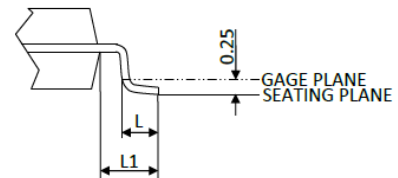
30.2 LQFP100 PIN



TOP VIEW



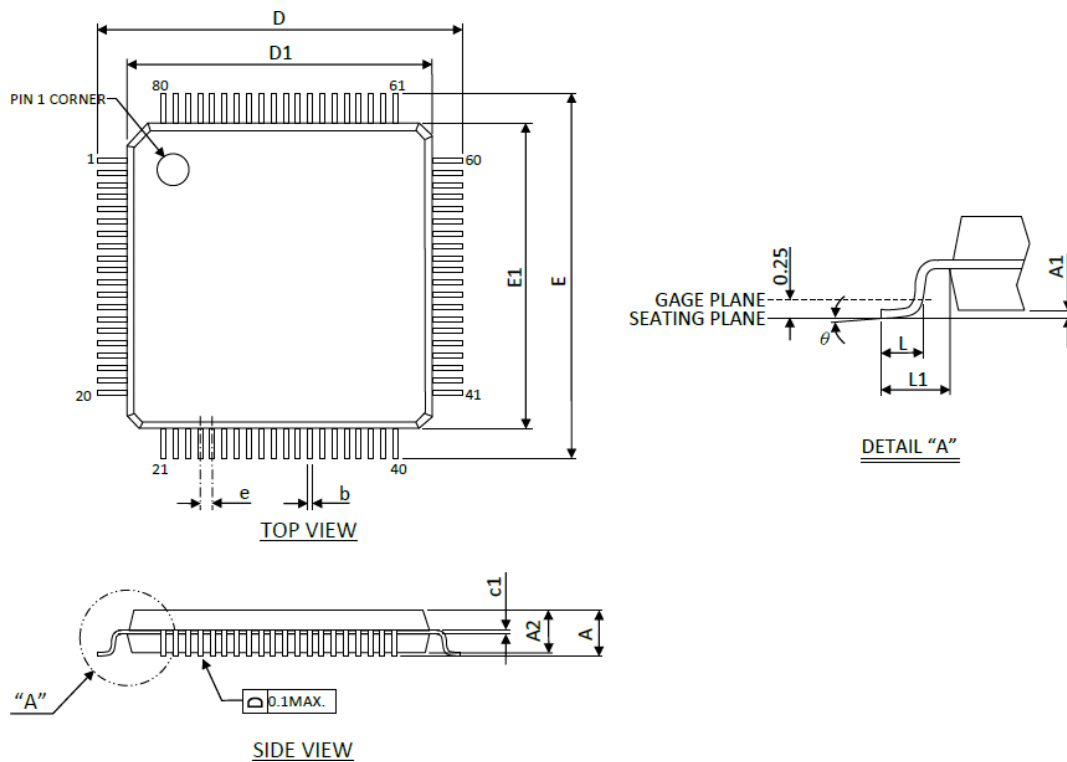
SIDE VIEW



DETAIL "A"

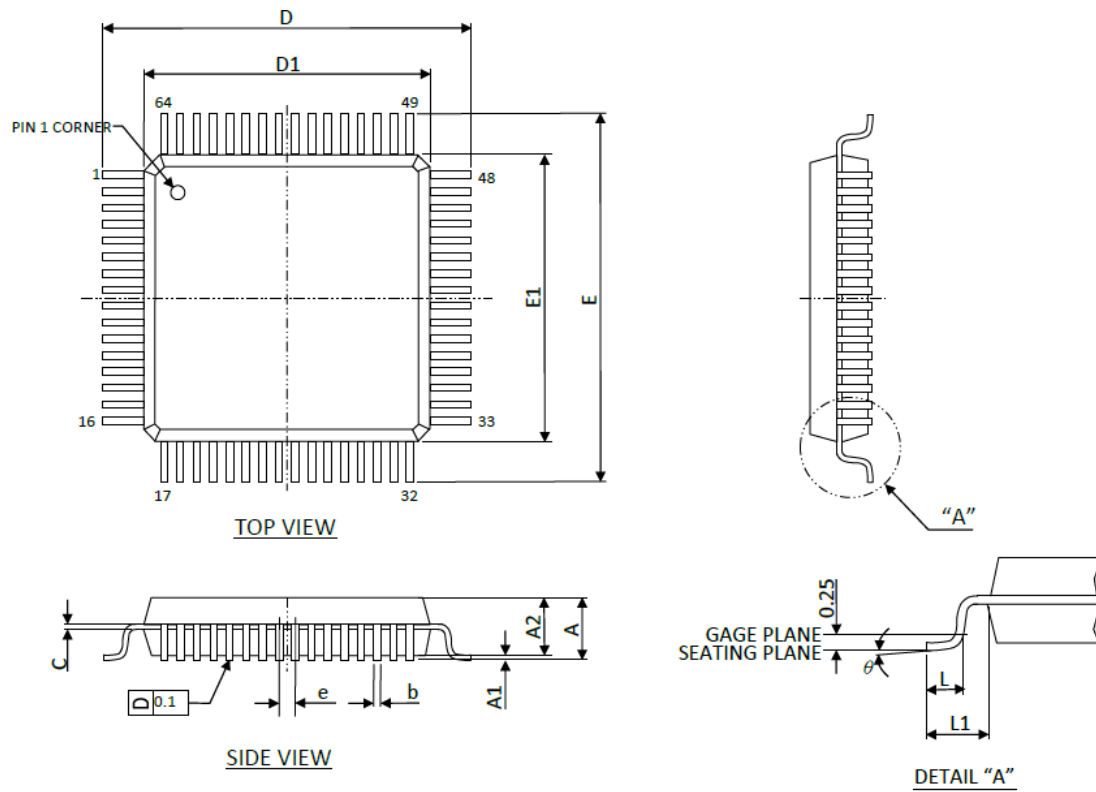
	Min	Typical	Max	Min	Typical	Max
		(inch)			(mm)	
A			0.063			1.6
A1	0.002		0.006	0.05		0.15
A2	0.053		0.057	1.35		1.45
C	0.004		0.008	0.09		0.20
D		0.630BSC			16BSC	
D1		0.551BSC			14BSC	
E		0.630BSC			16BSC	
E1		0.551BSC			14BSC	
e		0.020BSC			0.5BSC	
B	0.007		0.011	0.17		0.27
L	0.018		0.030	0.45		0.75
L1		0.39F			1.00REF	

30.3 LQFP80 PIN



	Min	Typical	Max	Min	Typical	Max
		(inch)			(mm)	
A	0.063			1.6		
A1	0.002		0.006	0.05		0.15
A2	0.053		0.057	1.35		1.45
c1	0.004		0.006	0.09		0.16
D	0.473BSC			12BSC		
D1	0.393BSC			10BSC		
E	0.473BSC			12BSC		
E1	0.393BSC			10BSC		
e	0.016BSC			0.4BSC		
B	0.005		0.009	0.13		0.23
L	0.018		0.030	0.45		0.75
L1	0.039REF			1.00REF		

30.4 LQFP64 PIN



	Min	Typical	Max	Min	Typical	Max
		(inch)			(mm)	
A			0.063			1.6
A1	0.002		0.006	0.05		0.15
A2	0.053		0.057	1.35		1.45
c	0.004		0.008	0.09		0.20
D		0.354BSC			9BSC	
D1		0.276BSC			7BSC	
E		0.354BSC			9BSC	
E1		0.276BSC			7BSC	
e		0.016BSC			0.4BSC	
b	0.005		0.009	0.13		0.23
L	0.018		0.030	0.45		0.75
L1		0.039REF			1.00REF	

31 版本说明

版本号	时间	修改内容
V1.0	2022.4.29	初始版本
V1.1	2022.10.19	1、修改在线编程部分描述 2、删除 SPI、UART、I2C 接口支持 SMBUS
V1.2	2022.10.28	1、修改 SPI 以及 I2C 部分描述
V1.3	2023.2.22	修改电气特性部分参数
V1.4	2023.9.21	1、新增系统配置 OPTIONS 章节 2、新增 LCD 启动描述 3、手册勘误
V1.5	2023.10.17	新增在线编程，选择编程区域描述。
V1.6	2023.11.10	增加 N 版本，温度适应范围：-40℃~125℃
V1.7	2023.12.18	修改 ADC 部分描述，修改系统框图，增加应用框图
V1.8	2024.05.18	增加产品信息
V1.9	2024.07.11	增加定时器时钟源选择内部低速时钟描述