

**HC89F0431A**

**HC89F0421A**

**HC89F0411A**

**数据手册**

**20/16/8 引脚 8 位**

**ADC 型 FLASH 单片机**

**外设功能引脚全映射**

# 目录

<b>1</b>	<b>产品简介 .....</b>	<b>6</b>
1.1	功能特性 .....	6
1.2	应用注意 .....	8
1.3	系统框图 .....	9
1.4	引脚配置 .....	10
1.5	引脚描述 .....	11
1.6	外设功能引脚全映射模块 PTM .....	16
<b>2</b>	<b>CPU .....</b>	<b>18</b>
2.1	CPU 特性 .....	18
2.2	CPU 相关寄存器 .....	18
<b>3</b>	<b>存储器 .....</b>	<b>20</b>
3.1	程序存储器 (FLASH) .....	20
3.2	数据存储器 (RAM) .....	29
3.3	特殊功能寄存器 (SFR) .....	30
<b>4</b>	<b>系统时钟 .....</b>	<b>34</b>
4.1	系统时钟特性 .....	34
4.2	内部时钟 .....	35
4.3	外部时钟 .....	35
4.4	系统时钟相关寄存器 .....	36
<b>5</b>	<b>电源管理 .....</b>	<b>41</b>
5.1	电源管理特性 .....	41
5.2	空闲模式 .....	41
5.3	掉电模式 .....	41
5.4	电源管理相关寄存器 .....	42
<b>6</b>	<b>复位 .....</b>	<b>43</b>
6.1	复位特性 .....	43
6.2	POR (POWER-ON RESET) 复位 .....	43
6.3	BOR (BROWN-OUT RESET) 复位 .....	43
6.4	外部 RST 复位 .....	44
6.5	外部端口低压检测复位 .....	44
6.6	软件复位 .....	44
6.7	看门狗 (WDT) 复位 .....	44
6.8	堆栈溢出复位 .....	44
6.9	复位相关寄存器 .....	45
<b>7</b>	<b>通用及复用 I/O .....</b>	<b>48</b>
7.1	通用及复用 I/O 特性 .....	48
7.2	I/O 模式 .....	48

7.3	I/O 功能框图.....	49
7.4	I/O 端口相关寄存器.....	49
7.5	外设功能引脚全映射控制.....	55
<b>8</b>	<b>中断.....</b>	<b>57</b>
8.1	中断特性.....	57
8.2	中断汇总.....	57
8.3	中断向量.....	58
8.4	中断优先级.....	58
8.5	中断处理.....	59
8.6	中断响应时间.....	59
8.7	外部中断.....	60
8.8	中断相关寄存器.....	61
<b>9</b>	<b>定时器/计数器.....</b>	<b>69</b>
9.1	定时器/计数器特性.....	69
9.2	定时器/计数器 Tx(x=0,1).....	69
9.3	定时器 3.....	74
9.4	定时器/计数器 4.....	77
9.5	定时器 5.....	81
<b>10</b>	<b>脉宽调制 PWM.....</b>	<b>86</b>
10.1	PWM 特性.....	86
10.2	PWM 输出类型.....	87
10.3	PWM 输出模式.....	89
10.4	PWM 相关寄存器.....	92
<b>11</b>	<b>单路 8 位 PWM 模块.....</b>	<b>106</b>
11.1	PWM 特性.....	106
11.2	PWM 模块相关寄存器.....	106
<b>12</b>	<b>看门狗定时器 WDT.....</b>	<b>108</b>
12.1	WDT 特性.....	108
12.2	WDT 相关寄存器.....	109
<b>13</b>	<b>通用异步收发器 UART.....</b>	<b>111</b>
13.1	UART 特性.....	111
13.2	工作方式.....	111
13.3	波特率发生器.....	118
13.4	多机通信.....	119
13.5	帧出错检测.....	120
13.6	UART1 相关寄存器.....	121
13.7	UART2.....	124
<b>14</b>	<b>串行外部设备接口 SPI.....</b>	<b>129</b>
14.1	SPI 特性.....	129
14.2	SPI 信号描述.....	129
14.3	SPI 时钟速率.....	130

14.4	SPI 功能框图 .....	130
14.5	SPI 工作模式 .....	130
14.6	SPI 传送形式 .....	132
14.7	SPI 出错检测 .....	133
14.8	SPI 中断 .....	133
14.9	SPI 配置对照 .....	134
14.10	SPI 相关寄存器 .....	135
<b>15</b>	<b>IIC 总线.....</b>	<b>137</b>
15.1	IIC 特性 .....	137
15.2	IIC 总线工作原理 .....	138
15.3	总线上数据的有效性 .....	138
15.4	总线上的信号 .....	139
15.5	总线上数据初始格式 .....	140
15.6	IIC 总线寻址约定 .....	140
15.7	主机向从机读写 1 个字节数据的过程.....	141
15.8	IIC 工作模式 .....	142
15.9	IIC 总线相关寄存器 .....	147
<b>16</b>	<b>模数转换 ADC.....</b>	<b>150</b>
16.1	ADC 特性 .....	150
16.2	ADC 省电唤醒 .....	151
16.3	ADC 相关寄存器 .....	152
<b>17</b>	<b>低电压检测/比较器 .....</b>	<b>159</b>
17.1	低电压检测/比较器特性 .....	159
17.2	低电压检测/比较器相关寄存器 .....	160
<b>18</b>	<b>软件 LCD.....</b>	<b>163</b>
18.1	LCD 特性 .....	163
18.2	软件流程说明 .....	164
18.3	LCD 帧.....	164
18.4	LCD 相关寄存器 .....	166
<b>19</b>	<b>循环冗余校验 CRC.....</b>	<b>167</b>
19.1	CRC 特性.....	167
19.2	CRC 相关寄存器 .....	168
<b>20</b>	<b>代码选项 .....</b>	<b>170</b>
<b>21</b>	<b>指令表 .....</b>	<b>171</b>
<b>22</b>	<b>电气特性 .....</b>	<b>176</b>
22.1	极限参数 .....	176
22.2	DC 特性.....	176
22.3	AC 特性.....	178
22.4	FLASH 内存特性.....	179
22.5	ADC 特性.....	179
22.6	BOR 检测电压特性 .....	180

---

22.7	LVD/PLVD 检测电压特性 .....	180
22.8	比较器电气特性 .....	180
22.9	系统下电过程功耗 .....	181
22.10	频率-电压特性曲线 .....	182
22.11	频率-温度特性曲线 .....	182
22.12	ADC 内参 2V-温度特性曲线 .....	183
22.13	其他电气特性 .....	183
<b>23</b>	<b>开发工具 .....</b>	<b>184</b>
23.1	HC-LINK 仿真工具 .....	184
23.2	HC-PM51 烧录工具 .....	184
23.3	ISP 串口烧录 .....	184
23.4	软件下载 .....	184
<b>24</b>	<b>封装尺寸 .....</b>	<b>185</b>
24.1	SOP20 .....	185
24.2	TSSOP20 .....	186
24.3	SOP16 .....	187
24.4	SOP8 .....	188
<b>25</b>	<b>版本记录 .....</b>	<b>189</b>

# 1 产品简介

HC89F0431A/0421A/0411A 是一颗采用高速低功耗 CMOS 工艺设计开发的增强型 8 位单片机，内部有 16K Bytes FLASH 程序存储器，256Bytes IRAM+768Bytes XRAM，最多 18 个双向 I/O 口，5 个 16 位定时器/计数器，3 组 12 位带死区控制互补 PWM，1 路 8 位 PWM，2 个 UART，1 个 SPI，1 个 IIC，16 个外部中断，16+2 路 12 位 ADC，1 个低压检测模块，四种系统工作模式和多个中断源。

## 1.1 功能特性

- ◆ CPU
  - 增强型 1T 8051 内核
- ◆ ROM
  - 16K Bytes FLASH/128 Bytes EEPROM
  - IAP 和 ICP 操作
  - 灵活的代码保护模式
- ◆ RAM
  - 256 Bytes IRAM
  - 768 Bytes XRAM
- ◆ 时钟
  - 内部高精度 32MHz RC，可软件微调
  - 内部 44KHz RC
  - 外部高频晶振 4MHz—20MHz
  - 外部低频晶振 32.768KHz
  - 多种时钟输出
- ◆ 多种复位方式
  - 上电复位 (POR)
  - 多级低电压复位 (BOR)
    - 4.2/3.9/3.6/3.0/2.6/2.4/2.0/1.8V
  - 看门狗 (WDT) 复位
  - 软件复位
  - 堆栈溢出复位
  - 外部管脚低电平复位
  - 外部管脚电压 (1.2V) 检测复位
- ◆ I/O
  - 最多 18 个双向 I/O 口
  - 多种模式可配：输入、带上拉输入、带下拉输入、施密特输入、模拟输入、强推挽输出、开漏输出、开漏带上拉输出
  - 外设功能引脚全映射模块 PTM
  - 所有端口可配置 1/2 bias 软件 LCD 驱动
- ◆ 中断
  - 17 个中断源
  - 4 级中断优先级
  - 16 个外部中断
- ◆ 定时器/计数器
  - T0/T1 兼容标准 8051，16 位自动重载
  - T3 可以工作在掉电模式
  - T4 可以使用外部信号触发定时
  - T5 带捕获功能
- ◆ PWM
  - 最多 3 组 12 位带死区控制互补 PWM
    - 可配置为 6 路独立输出
    - 可当定时器使用
    - 具有故障检测功能
    - 可配置边沿对齐或中心对齐
  - 1 路 8 位单输出 PWM
- ◆ 通讯模块
  - 2 个 UART
  - 1 个 SPI
  - 1 个 IIC
- ◆ ADC 检测电路
  - 支持 16+2 ch 12 位 ADC 检测
  - ADC 参考电压可选内部 1.3/2/3/4V、外部 VREF、VDD
  - 软件触发、外部触发、定时器触发
- ◆ 低电压检测模块
  - 比较器功能
  - VDD 多级电压检测，可中断
    - 4.2/3.9/3.6/3.0/2.6/2.4/2.0/1.9V
- ◆ 循环冗余校验 (CRC)
- ◆ 省电模式
  - 空闲模式
  - 掉电模式
- ◆ 工作条件
  - 宽电压 2.0V—5.5V
  - 温度范围 -40°C—105°C
- ◆ 封装类型
  - TSSOP20/SOP20
  - SOP16/SOP8
- ◆ 支持 SWD、JTAG 仿真和下载

## ✓ 选型表

产品型号	ROM Bytes	RAM Bytes	MAX Freq	I/O	ADC	Timer	PWM	INT	WDT
HC89F0431A	16K	256+768	16MHz	18	15+2	5	12Bit*3 组 8Bit*1	16	1
HC89F0431A	16K	256+768	16MHz	18	16+2	5	12Bit*3 组 8Bit*1	16	1
HC89F0421A	16K	256+768	16MHz	14	11+2	5	12Bit*3 组 8Bit*1	12	1
HC89F0411A	16K	256+768	16MHz	6	6+2	5	12Bit*3 组 8Bit*1	6	1

产品型号	Voltag	TEMP	Package	Simulator	Programmer	Datasheet	DemoCode	DemoBoard
HC89F0431A	2.0~5.5V	-40~+105°C	TSSOP20	HC-LINK	HC-PM51	√	√	√
HC89F0431A	2.0~5.5V	-40~+105°C	SOP20	HC-LINK	HC-PM51	√	√	√
HC89F0421A	2.0~5.5V	-40~+105°C	SOP16	HC-LINK	HC-PM51	√	√	√
HC89F0411A	2.0~5.5V	-40~+105°C	SOP8	HC-LINK	HC-PM51	√	√	√

## 1.2 应用注意

1. 为保证系统的稳定性，必须在VDD和GND之间接一电容（容值须大于或等于0.1 $\mu$ F）。
2. 可以直接烧录HC89F0431的烧录文件，但需要将HC-PM51烧录器的固件版本升级到最新版。
3. P2.7引脚出厂时默认为复位管脚，端口模式为施密特输入带上拉，可以通过配置代码选项将此口配置为普通IO引脚。
4. 如果需要进行FLASH IAP操作，请仔细阅读3.1.5.1的注意事项。
5. 在进行IAP操作时，不响应任何中断。
6. ADCEN置1或切换转换通道后，建议延时20 $\mu$ s后再启动ADC转换，如果外部输入阻抗很大时，需要延长这个时间。
7. 当ADC的参考电压为VDD时，ADC转换时钟可以为8MHz，一次转换只需要15个ADC\_CLK，这样可以得到最快的ADC转换速度。
8. P2.7、P2.5、P2.4、P2.3这四个端口的模式配置和其他端口的不一样，请仔细看7.4.6小节。
9. UART2在使用全双工时，需要配置CPU频率为16MHz以上，波特率配置9600或以下，而且在UART2中断服务函数里尽早清零TI或RI，请参考相关例程。
10. HC89F0411A的Pin2管脚兼容P2.0、P2.1两个端口的功能，操作时只能使用其中一个端口功能。其中一个端口在进行操作时，需将另一个端口配置为模拟输入。



### 1.3 系统框图

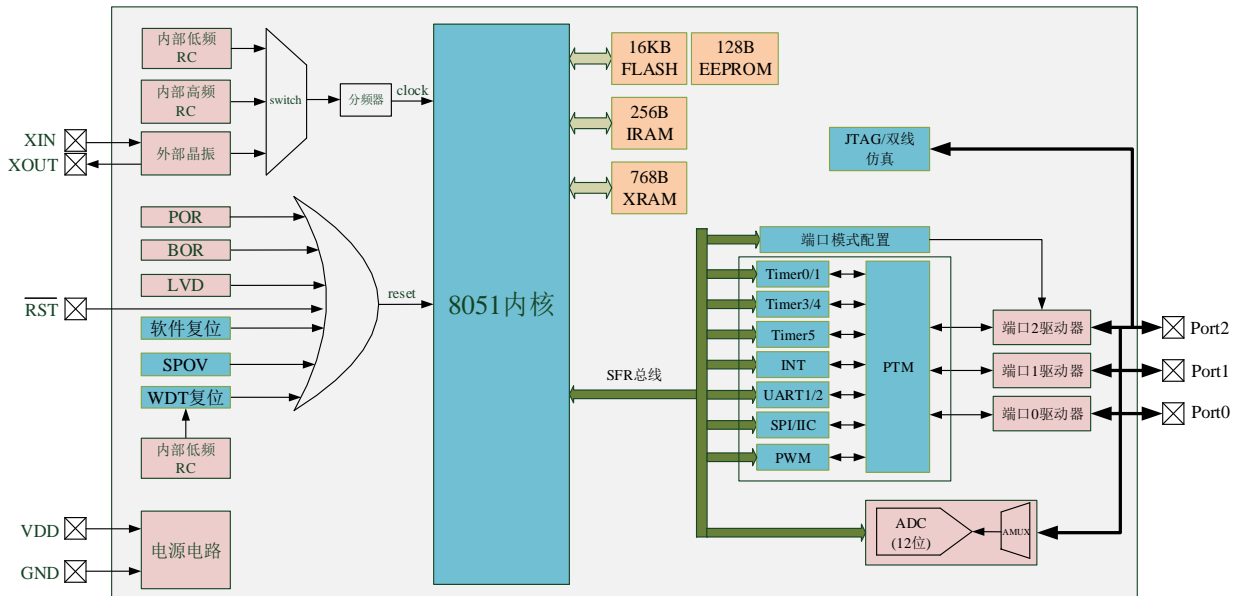


Figure 1-1 系统框图

## 1.4 引脚配置

### 1.4.1 SOP20/TSSOP20 引脚配置

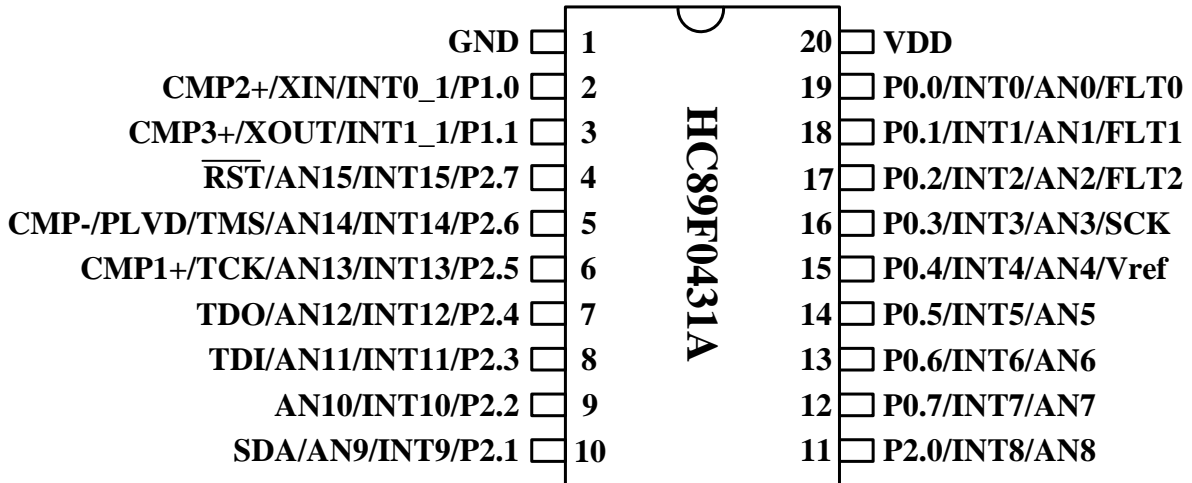


Figure 1-2 SOP20/TSSOP20 引脚配置图

### 1.4.2 SOP16 引脚配置

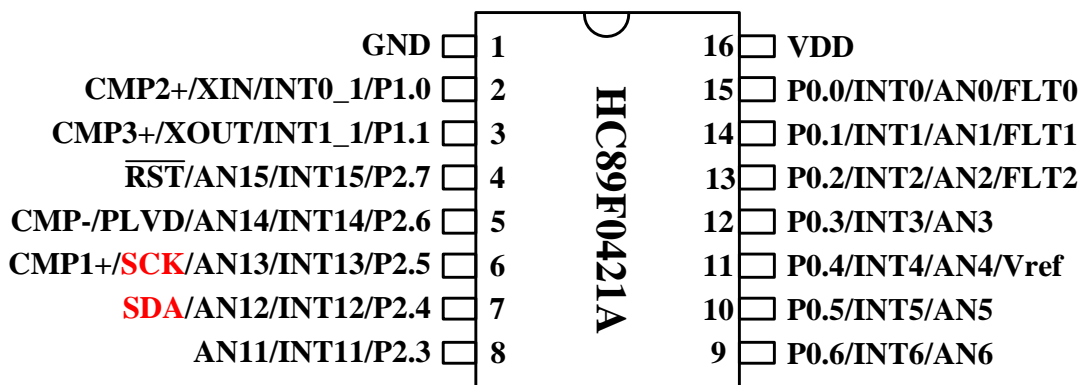


Figure 1-3 SOP16 引脚配置图

### 1.4.3 SOP8 引脚配置

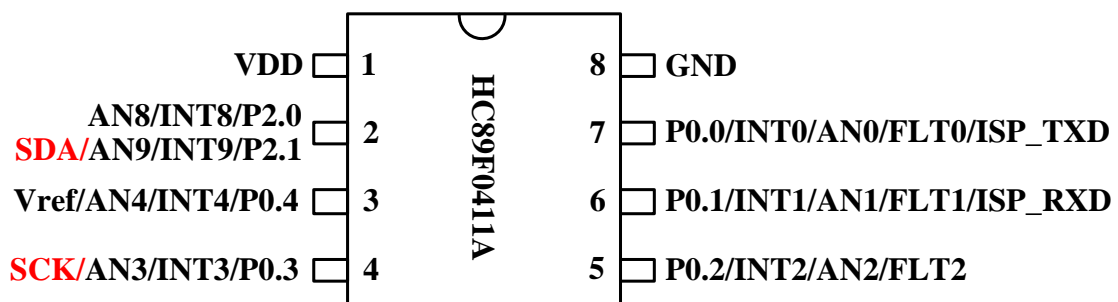


Figure 1-5 SOP8 引脚配置图

## 1.5 引脚描述

### 1.5.1 SOP20/TSSOP20 引脚描述

脚位	名称	类型	说明
1	GND	P	电源地
2	P1.0	I/O	输入/输出口
	XIN	AN	外部晶振输入口
	INT0_1	I	可通过软件配置为外部中断 0 输入口
	CMP2+	AN	比较器正端 2 输入口
3	P1.1	I/O	输入/输出口
	XOUT	AN	外部晶振输出口
	INT1_1	I	可通过软件配置为外部中断 1 输入口
	CMP3+	AN	比较器正端 3 输入口
4	P2.7	I/O	输入/输出口
	AN15	AN	ADC15 输入口
	RST	I	外部复位输入口
	INT15	I	外部中断 15 输入口
5	P2.6	I/O	输入/输出口
	AN14	AN	ADC14 输入口
	TMS	I	JTAG 模式输入
	INT14	I	外部中断 14 输入口
	PLVD	AN	端口低电压检测端口
	CMP-	AN	比较器负端输入口
6	P2.5	I/O	输入/输出口
	AN13	AN	ADC13 输入口
	TCK	I	JTAG 时钟输入
	INT13	I	外部中断 13 输入口
	CMP1+	AN	比较器正端 1 输入口
7	P2.4	I/O	输入/输出口
	AN12	AN	ADC12 输入口
	TDO	O	JTAG 数据输出
	INT12	I	外部中断 12 输入口
8	P2.3	I/O	输入/输出口
	AN11	AN	ADC11 输入口
	TDI	I	JTAG 数据输入
	INT11	I	外部中断 11 输入口
9	P2.2	I/O	输入/输出口
	AN10	AN	ADC10 输入口
	INT10	I	外部中断 10 输入口
10	P2.1	I/O	输入/输出口
	AN9	AN	ADC9 输入口

	INT9	I	外部中断 9 输入口
	SDA	I/O	双线模式数据输入/输出
11	P2.0	I/O	输入/输出口
	AN8	AN	ADC8 输入口
	INT8	I	外部中断 8 输入口
12	P0.7	I/O	输入/输出口
	AN7	AN	ADC7 输入口
	INT7	I	外部中断 7 输入口
13	P0.6	I/O	输入/输出口
	AN6	AN	ADC6 输入口
	INT6	I	外部中断 6 输入口
14	P0.5	I/O	输入/输出口
	AN5	AN	ADC5 输入口
	INT5	I	外部中断 5 输入口
15	P0.4	I/O	输入/输出口
	AN4	AN	ADC4 输入口
	Vref	AN	ADC 外部参考电压输入/输出口
	INT4	I	外部中断 4 输入口
16	P0.3	I/O	输入/输出口
	AN3	AN	ADC3 输入口
	INT3	I	外部中断 3 输入口
	SCK	I	双线模式时钟输入
17	P0.2	I/O	输入/输出口
	AN2	AN	ADC2 输入口
	INT2	I	外部中断 2 输入口
	FLT2	I	PWM2 故障检测输入引脚
18	P0.1	I/O	输入/输出口
	AN1	AN	ADC1 输入口
	INT1	I	外部中断 1 输入口
	FLT1	I	PWM1 故障检测输入引脚
19	P0.0	I/O	输入/输出口
	AN0	AN	ADC0 输入口
	INT0	I	外部中断 0 输入口
	FLT0	I	PWM0 故障检测输入引脚
20	VDD	P	电源输入口

注：I=输入，O=输出，I/O=输入/输出，P=电源，AN=模拟输入输出。

## 1.5.2 SOP16 引脚描述

脚位	名称	类型	说明
1	GND	P	电源地
2	P1.0	I/O	输入/输出口
	XIN	AN	外部晶振输入口
	INT0_1	I	可通过软件配置为外部中断 0 输入口
	CMP2+	AN	比较器正端 2 输入口
3	P1.1	I/O	输入/输出口
	XOUT	AN	外部晶振输出口
	INT1_1	I	可通过软件配置为外部中断 1 输入口
	CMP3+	AN	比较器正端 3 输入口
4	P2.7	I/O	输入/输出口
	$\overline{\text{RST}}$	I	外部复位输入口
	AN15	AN	ADC15 输入口
	INT15	I	外部中断 15 输入口
5	P2.6	I/O	输入/输出口
	AN14	AN	ADC14 输入口
	INT14	I	外部中断 14 输入口
	PLVD	AN	端口低电压检测端口
	CMP-	AN	比较器负端输入口
6	P2.5	I/O	输入/输出口
	SCK	I	双线模式时钟输入
	AN13	AN	ADC13 输入口
	INT13	I	外部中断 13 输入口
	CMP1+	AN	比较器正端 1 输入口
7	P2.4	I/O	输入/输出口
	SDA	I/O	双线模式数据输入/输出
	AN12	AN	ADC12 输入口
	INT12	I	外部中断 12 输入口
8	P2.3	I/O	输入/输出口
	AN11	AN	ADC11 输入口
	INT11	I	外部中断 11 输入口
9	P0.6	I/O	输入/输出口
	AN6	AN	ADC6 输入口
	INT6	I	外部中断 6 输入口
10	P0.5	I/O	输入/输出口
	AN5	AN	ADC5 输入口
	INT5	I	外部中断 5 输入口
11	P0.4	I/O	输入/输出口
	AN4	AN	ADC4 输入口
	Vref	AN	ADC 外部参考电压输入/输出口
	INT4	I	外部中断 4 输入口

12	P0.3 AN3 INT3	I/O AN I	输入/输出口 ADC3 输入口 外部中断 3 输入口
13	P0.2 AN2 INT2 FLT2	I/O AN I I	输入/输出口 ADC2 输入口 外部中断 2 输入口 PWM2 故障检测输入引脚
14	P0.1 AN1 INT1 FLT1	I/O AN I I	输入/输出口 ADC1 输入口 外部中断 1 输入口 PWM1 故障检测输入引脚
15	P0.0 AN0 INT0 FLT0	I/O AN I I	输入/输出口 ADC0 输入口 外部中断 0 输入口 PWM0 故障检测输入引脚
16	VDD	P	电源输入口

注：I=输入，O=输出，I/O=输入/输出，P=电源，AN=模拟输入输出

### 1.5.3 SOP8 引脚描述

脚位	名称	类型	说明
1	VDD	P	电源输入口
2	P2.0	I/O	输入/输出口
	AN8	AN	ADC8 输入口
	INT8	I	外部中断 8 输入口
	P2.1	I/O	输入/输出口
	AN9	AN	ADC9 输入口
	INT9	I	外部中断 9 输入口
	SDA	I/O	双线模式数据输入/输出
3	P0.4	I/O	输入/输出口
	AN4	AN	ADC4 输入口
	Vref	AN	ADC 外部参考电压输入/输出口
	INT4	I	外部中断 4 输入口
4	P0.3	I/O	输入/输出口
	AN3	AN	ADC3 输入口
	INT3	I	外部中断 3 输入口
	SCK	I	双线模式时钟输入
5	P0.2	I/O	输入/输出口
	AN2	AN	ADC2 输入口
	INT2	I	外部中断 2 输入口
	FLT2	I	PWM2 故障检测输入引脚
6	P0.1	I/O	输入/输出口
	AN1	AN	ADC1 输入口
	INT1	I	外部中断 1 输入口
	FLT1	I	PWM1 故障检测输入引脚
	ISP_RXD	I	ISP 下载 RXD 口（此口不影响 UART 正常使用）
7	P0.0	I/O	输入/输出口
	AN0	AN	ADC0 输入口
	INT0	I	外部中断 0 输入口
	FLT0	I	PWM0 故障检测输入引脚
	ISP_TXD	I	ISP 下载 TXD 口（此口不影响 UART 正常使用）
8	GND	P	电源地

注意：固化原厂 ISP 程序的芯片 ISP\_TXD 在 P0.0 上，ISP\_RXD 在 P0.1 上。

注意：Pin2 管脚具备 P2.0、P2.1 两个端口的功能，但操作时只能使用其中一个端口功能。其中一个端口在进行操作时，需将另一个端口设置为默认模拟状态。

注：I=输入，O=输出，I/O=输入/输出，P=电源，AN=模拟输入输出。

## 1.6 外设功能引脚全映射模块 PTM

HC89F0431A/0421A/0411A 内置外设功能引脚全映射模块（PTM），可通过用户软件操作将绝大多数的外设功能引脚配置在任意一个非电源口（VDD、GND）上。

### 1.6.1 PTM 模块特性

- 外设引脚为输入功能（T0/1/3/5 外部输入、RXD 等等）特性时，系统将允许其多对一映射，即将多种输入特性外设功能引脚分配到同一 IO 口上，此举可使用户系统得到更好优化。
- 外设引脚为输出功能（T0/1/4 时钟输出、TXD 等等）特性时，如果将多个输出特性外设功能引脚分配到同一 IO 口，遵循固定的优先级，只能有一个输出有效。
- 软件操作，使用灵活。用户在应用系统设计时，不用考虑外设功能引脚布局问题，从而降低开发成本。
- 用户在开发过程中遇到如 PCB 上外设功能引脚布局错误情况时，可使用该模块对外设功能引脚进行重新分配，从而缩短开发周期。
- 用户应用系统升级如更换外围器件或 MCU 时，改动项可减至最少，从而降低系统维护成本。

### 1.6.2 PTM 可全映射外设功能引脚

外设	名称	类型	说明
定时器	T0	I/O	T0 的外部输入或 T0 时钟分频输出
	T1	I/O	T1 的外部输入或 T1 时钟分频输出
	T3	I	T3 的外部输入
	T4	O	T4 的输出
	T5	I	T5 的外部输入
PWM	PWM0	O	PWM0 输出口
	PWM01	O	PWM01 输出口
	PWM1	O	PWM1 输出口
	PWM11	O	PWM11 输出口
	PWM2	O	PWM2 输出口
	PWM21	O	PWM21 输出口
	PWM3	O	PWM3 输出口
CLK	CLKO	O	时钟输出口
UART	TXD	O	UART1 数据传输口
	RXD	I/O	UART1 接收脚
	TXD2	O	UART2 数据传输口
	RXD2	I	UART2 接收脚
SPI	MOSI	I/O	SPI 的数据口，主机的输出和从机的输入
	MISO	I/O	SPI 的数据口，主机的输入和从机的输出
	SCK	I/O	SPI 的时钟口
	$\overline{SS}$	I	SPI 的片选口
IIC	SCL	I/O	IIC 时钟口
	SDA	I/O	IIC 数据口



### 1.6.3 PTM 不可全映射外设功能引脚

PTM 不可全映射外设功能引脚包括电源口 (VDD、GND)、PWM 故障检测脚 (FLT0/1/2)、ADC 输入、Vref 引脚、INT0-15 功能口、晶振脚 (XIN、XOUT)、外部复位端口 ( $\overline{\text{RST}}$ )、LVD 电压检测端口 (PLVD)，这些功能口是固定的，不能任意映射。

## 2 CPU

### 2.1 CPU 特性

HC89F0431A/0421A/0411A 的 CPU 是一个增强型 1T 兼容 8051 的内核，在同样的系统时钟下，较之传统的 8051 芯片具有运行更快速，性能更优越的特性。

### 2.2 CPU 相关寄存器

#### 2.2.1 程序计数器 PC

程序计数器 PC 在物理上是独立的，不属于 SFR 之列。PC 字长 16 位，是专门用来控制指令执行顺序的寄存器。单片机上电或复位后，PC 的值为 0000H，这样单片机从程序的零地址开始执行程序，假如第二复位向量使能，那上电或复位后，单片机将从第二复位向量指定的地址处开始执行程序。

#### 2.2.2 累加器 ACC

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

#### 2.2.3 寄存器 B

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

#### 2.2.4 程序状态字寄存器 PSW

此寄存器用来保存 ALU 运算结果的特征和处理状态，这些特征和状态可以作为控制程序转移的条件，供程序判别和查询，它的各位定义如下所示：

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	0	0	0	0	0	0
位符号	CY	AC	F0	RS[1:0]		OV	F1	P

位编号	位符号	说明
7	CY	进位/借位标志位 0: 算术运算中，无进位或借位 1: 算术运算中，有进位或借位
6	AC	辅助进位/借位标志位 0: 算术运算中，无辅助进位或借位 1: 算术运算中，有辅助进位或借位
5	F0	用户自定义标志位
4-3	RS[1:0]	工作寄存器组选择位 00: 第 0 组 (00H~07H)

		01: 第 1 组 (08H~0FH) 10: 第 2 组 (10H~17H) 11: 第 3 组 (18H~1FH)
2	OV	溢出标志位 0: 无溢出 1: 有溢出
1	F1	用户自定义标志位
0	P	奇偶标志位 0: ACC 寄存器中 1 的个数为 0 或偶数 1: ACC 寄存器中 1 的个数为奇数

### 2.2.5 堆栈指针 SP

堆栈指针SP是一个8位的专用寄存器，它指示出堆栈顶部在内部RAM中的位置。单片机复位后，SP值为07H，使得堆栈事实上由08H单元开始，考虑到08H~1FH单元分别属于工作寄存器1~3，若在程序设计中要使用到这些区域，最好把SP的值改为较大的值。51单片机的堆栈是向上生成的，例如：SP=30H，CPU执行一条调用指令或响应中断后，PC进栈，PCL保护到31H，PCH保护到32H，SP=32H。

### 2.2.6 数据指针 DPTR

数据指针DPTR是一个16位的专用寄存器，由两个8位的寄存器DPH（高8位）和DPL（低8位）组成。此系列单片机有两个16位的数据指针DPTR0和DPTR1，其共用同一地址空间，可通过设置DPS（INSCON.0）位来选择具体使用的数据指针。

### 2.2.7 数据指针选择寄存器 INSCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R/W	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	EEPROM	-		IAPS	-			DPS

位编号	位符号	说明
7	EEPROM	EEPROM 操作区选择位 0: 不选择 EEPROM 1: 选择 EEPROM，同时需要将 IAPS 设置为 0
6-5	-	保留位（读为 0，写无效）
4	IAPS	MOV C 操作区选择位 0: 对程序区读擦写操作，同时需要将 EEPROM 设置为 0 1: 对 OPTION 区读操作
3-1	-	保留位（读为 0，写无效）
0	DPS	数据指针选择位 0: 数据指针 DPTR0 1: 数据指针 DPTR1

## 3 存储器

### 3.1 程序存储器（FLASH）

#### 3.1.1 FLASH 特性

- 在工作电压范围内都能进行擦除和编程操作
- 在线编程（ICP）操作支持写入、读取和擦除操作
- ICP 操作可设置 32 位密码进行保护
- 在应用编程（IAP）支持用户自定义启动代码
- 灵活的代码保护模式
- 擦写次数至少 10 万次
- 数据保存年限至少 10 年
- 16K Bytes FLASH: 128 字节为 1 个扇区，8 个扇区为 1 页，4 页为一个保护单位
- 128 Bytes EEPROM: 4 字节为 1 个扇区

#### 3.1.2 FLASH 数据安全

FLASH 的操作可以分成两种：第一种是通过 FLASH 编程器对 FLASH 进行读、擦、写操作，这种方式被称为在电路编程模式（ICP），JTAG 就是 ICP 的一种；第二种是用户程序代码在 FLASH 代码区中运行，对 FLASH 存储器其他扇区进行读、擦、写操作，但无法擦除代码自身所在的扇区，这种方式被称为在应用编程模式（IAP）。

##### 3.1.2.1 ICP操作密码保护

用户可以通过上位机软件对 ICP 操作进行密码保护，密码长度为 4 个字节（32 位），一旦用户设置了密码，那么只有输入正确的密码才能进入 JTAG，否则就不能对 FLASH 进行任何操作，这样可以有效保护用户的程序代码。

##### 3.1.2.2 ICP读擦写FLASH保护

ICP 的读保护以 4K 字节为保护单位，当一个 4K 字节空间的读保护使能时，ICP 读这个 4K 字节空间，读出来的数据为全 0，但是仍然可以通过 ICP 操作进行仿真。

ICP 的擦写保护也是以 4K 字节为保护单位，当对应 4K 字节的擦写保护使能时，ICP 将不能擦除和编程这个 4K 字节空间，强写也不允许。

若对应 4K 字节空间读保护被使能，但被允许擦除与写入，则可先擦除后获得该 4K 字节空间的读允许，直至复位或掉电。

ICP 的读擦写保护通过上位机软件来配置，详细情况请参见 HC-51LINK 用户手册。

##### 3.1.2.3 IAP读擦写FLASH保护

IAP 通过 MOV C 指令来读 FLASH，IAP 读保护以 4K 字节为单位，如果一个 4K 字节空间设置了读保护，其他 4K 字节空间的 MOV C 指令读这个 4K 字节空间，读出来的数据为零，但这个 4K 字节空间的 MOV C 指令可以读取自身的数据。

IAP 擦写 FLASH 的步骤详见 [FLASH IAP 操作](#)，IAP 的擦写保护以 4K 字节为单位，IAP 擦写之前需要先看相应扇区的擦写保护是否使能，没有使能擦写保护才能进行 IAP 的擦写。

若对应 4K 字节空间读保护被使能，但被允许擦除与写入，则可先擦除后获得该 4K 字节空间的读允许，直至复位或掉电。

IAP 的读擦写保护通过上位机软件来配置，详细情况请参见 HC-LINK 用户手册。

### 3.1.3 EEPROM

HC89F0431A/0421A/0411A 的 EEPROM 独立于 16K 的 FLASH ROM 外，大小是 128 字节，分成 32 个扇区，即 4 个字节为一个扇区。

在对 EEPROM 进行读擦写操作时，需要将 INSCON 寄存器的第 7 位 EEPROM 置 1，第 4 位 IAPS 置 0，使数据指针指向 EEPROM 操作区域。

写 EEPROM 之前必须先进行一次 IAP 擦除操作，IAP 一次擦除一个扇区（4 字节），IAP 擦除时的地址寄存器可以是扇区里的任意地址。IAP 写 EEPROM 是单字节写，每次写一个字节。

IAP 擦除 EEPROM 一个扇区的时间是 5ms，IAP 写 EEPROM 一个字节的的时间是 68μs。

### 3.1.4 OPTION

在 16K 的 ROM 之外还有一个只读的 OPTION 区域，存放的内容包括：用户定义的一些数据、用户设置的密码、芯片的一些配置、第二复位向量相关的内容。具体地址分配如下表。

地址	名称	地址偏移量	名称	地址	名称	地址偏移量	名称
0x0000	SN_DATA0	0x0020	FLASH_SC0	0x0031	ERST_ENB	0x0100	CHIP_ID0
0x0001	SN_DATA1	0x0021	FLASH_SC1	0x0038	WAIT_TS	0x0101	CHIP_ID1
0x0002	SN_DATA2	0x0022	FLASH_SC2	0x0039	BORVS	0x0102	CHIP_ID2
0x0003	SN_DATA3	0x0023	FLASH_SC3	0x003B	-	0x0103	CHIP_ID3
0x0004	SN_DATA4	-	-	0x003E	RVCFG	0x0104	CHIP_ID4
0x0005	SN_DATA5	-	-	0x003F	nRVCFG	0x0105	CHIP_ID5
0x0006	SN_DATA6	-	-	-	-	0x0106	CHIP_ID6
0x0007	SN_DATA7	-	-	-	-	0x0107	CHIP_ID7
0x0008	ID_DATA0	-	-	-	-	-	-
0x0009	ID_DATA1	-	-	-	-	-	-
0x000A	ID_DATA2	-	-	-	-	-	-
0x000B	ID_DATA3	-	-	-	-	-	-
0x000C	ID_DATA4	-	-	-	-	-	-
0x000D	ID_DATA5	-	-	-	-	-	-
0x000E	ID_DATA6	-	-	-	-	-	-
0x000F	ID_DATA7	-	-	-	-	-	-

HC89F0431A/0421A/0411A 在出厂时都会固化一个 CHIP\_ID，一共 8 个字节，一颗芯片一个 ID，不会重复，也永远不能被擦除，用户可以在程序中通过 MOVC 来读出，也可以通过工具读出。

SN\_DATA 和 ID\_DATA 是用户自定义数据，FLASH\_SC 为客户密码，通过工具软件进行设置，如同设置代码选项一样，它们是可以被擦除和修改的，用户也可以在程序中通过 MOVC 来读出。

注意：1、用户在进行读 OPTION 操作前，需要将寄存器 INSCON[IAPS]位置 1。

2、首字母为“n”位对应数据的反码。

用C语言读CHIP\_ID的程序如下：

//从FLASH里面读取任意长度的数据

```
void Flash_ReadArr(unsigned int fui_Address,unsigned char fuc_Length,unsigned char *fuc_SaveArr)
{
    while(fuc_Length--)
        *(fuc_SaveArr++)=*((unsigned char code*)(fui_Address++));
}
```

```
//读取CHIP_ID的值，并保存到read_chip_id数组里
unsigned char read_chip_id[8];
INSCON |= 0x10;
Flash_ReadArr(0x0100,8,read_chip_id); //CHIP_ID开始地址为0x0100
INSCON &=~ 0x10;
如果需要读SN_DATA或ID_DATA，只需要修改读地址即可。
```

#### 3.1.4.1.1 外部复位使能 ERST0\_ENB

位编号	7	6	5	4	3	2	1	0
位符号	-							ERST0_ENB

位编号	位符号	说明
7-1	-	保留位
0	ERST0_ENB	复位引脚使能位 0: 外部 RST 输入 1: P2.7 为 GPIO

#### 3.1.4.1.2 复位重读 OPTION 后等待时间 WAIT\_TS

位编号	7	6	5	4	3	2	1	0
位符号	-	-	-	-	-	-	WAIT_TS	

位编号	位符号	说明
7-2	-	保留位
1-0	WAIT_TS	复位重读 option 后等待时间选择位 00: 8ms 01: 4ms 11: 16ms

#### 3.1.4.1.3 BOR 检测电压选择 BORVS

位编号	7	6	5	4	3	2	1	0
位符号	-	-	-	-	-	BORVS		

位编号	位符号	说明
7-3	-	保留位
2-0	BORVS	BOR 检测电压点选择位 000: 1.8V 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V

### 3.1.4.1.4 第二复位向量配置 RVCFG

位编号	7	6	5	4	3	2	1	0
位符号	RVSEN	-	-	-	RVADR[3:0]			

位编号	位符号	说明
7	RVSEN	第二复位向量上电使能 0: 禁止第二复位向量 1: 使能第二复位向量
6-4	-	保留位
3-0	RVADR[3:0]	第二复位向量配置值 第二复位向量地址 = {RVADR[3:0], 10'h000} 注: 1. RVADR[3:0]=0 时, 表示第二复位向量地址和 0x0000H 重合; 2. RVADR[3:0]只能配置 1000,1100,1110,1111 四个值 即第二复位向量的空间大小只能为 1K,2K,4K,8K

## 3.1.5 FLASH IAP 操作

HC89F0431A/0421A/0411A 的 FLASH 一共有 128 个扇区, 128 个字节为一个扇区,  $8*128 \text{ Bytes} = 1\text{K Bytes}$  为一页,  $1\text{K Bytes}*4 = 4\text{KBytes}$  为一块。

IAP 写之前必须先进行一次擦除操作, IAP 一次擦除一个扇区 (128 字节), IAP 擦除时的地址寄存器可以是扇区里的任意地址。IAP 写是单字节写, 每次写一个字节。

IAP 擦除一个扇区的时间是 5ms, 在 CPU\_CLK 为 16MHz 时, IAP 写一个字节的的时间是 68 $\mu$ s。

### 3.1.5.1 IAP操作注意事项

HC89F0431A/0421A/0411A 的用户程序代码可对 FLASH 进行读、擦、写操作, 作为用户更新代码或存储数据使用, 为保证用户对 FLASH 操作的安全性, 使用过程中请注意:

1. 在进行 FLASH 的 IAP 擦写之前, 需要配置扩展 SFR 里 `FREQ_CLK` 寄存器, 指明目前 CPU 时钟的频率, `FREQ_CLK` 寄存器配置的值等于 CPU 时钟的频率值, 最小为 1MHz, 假如目前 CPU 的运行频率为 16MHz, 那就配置寄存器 `FREQ_CLK=0x10`。建议在 IAP 擦写之前, 将 CPU 时钟频率分频为整数。当 CPU 时钟频率低于 1MHz 时, 不能进行 FLASH 的 IAP 擦写操作。
2. 系统在进行 IAP 操作时, 不响应任何中断。
3. 系统的正常工作电压是 2.0V~5.5V, 当系统电压低于该电压范围时, 可能会导致 IAP 误操作。建议此时使能 BOR 功能, 且配置合适的 BOR 消抖时间以保证 VDD 存在干扰的情况下 IAP 操作正常。
4. 在 Option 中设置相关的 IAP 擦写保护, 使能用户程序所在扇区保护位, 可以有效保证程序区不会被改写或误擦除。
5. 为防止 MCU 上电瞬间电压不稳导致程序跑飞造成误操作, 建议每次 IAP 操作前采用 ADC 或者 LVD 检测 MCU 当前电压, 如果电源低于 2.0V 则不进行 IAP 操作。
6. IAP 擦写操作前, 建议关闭中断 (`EA=0`), 确保在 IAP 操作期间不会被中断影响, 待 IAP 擦写操作完成后, 再将中断恢复。
7. 在执行 IAP 操作时, 不可避免的会遇到数据擦除结束后, 尚未写数据就掉电的情况, 所以建议采用双区域保存数据的方式, 即使一个区域的数据被擦除, 也可以保证另一个区域的数据被正常读取。

### 3.1.5.2 IAP数据寄存器 IAP\_DATA

位编号	7	6	5	4	3	2	1	0
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
位符号	IAP_DATA[7:0]							

位编号	位符号	说明
7-0	IAP_DATA[7:0]	IAP 数据寄存器

### 3.1.5.3 IAP地址寄存器 IAP\_ADDRL、IAP\_ADDRH

#### IAP\_ADDRL

位编号	7	6	5	4	3	2	1	0
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
位符号	IAP_ADDR[7:0]							

位编号	位符号	说明
7-0	IAP_ADDR[7:0]	IAP 操作时的地址寄存器低八位

#### IAP\_ADDRH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	1	1	1	1
位符号	-		IAP_ADDR[13:8]					

位编号	位符号	说明
7-6	-	保留位
5-0	IAP_ADDR[13:8]	IAP 操作时的地址寄存器高六位

注：必须在解锁后才能修改 IAP 地址寄存器，而且一次操作完成后，IAP 地址自动指向 0x3FFF。



### 3.1.5.4 IAP命令寄存器 IAP\_CMDH、IAP\_CMDL

#### IAP\_CMDH

位编号	7	6	5	4	3	2	1	0
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
位符号	IAP_CMDH[7:0]							

位编号	位符号	说明
7-0	IAP_CMDH[7:0]	操作命令模式选择位 0xF0: 解锁(22个CPU时钟后自动锁定, IAP_CMD[7:0] = 0x00) 0xE1: 触发一次操作 0xD2: 扇区擦除 0xB4: 字节编程 0x87: 软件复位, 复位地址为 0000H, 不重读代码选项 0x78: 软件复位, 复位地址为 0000H, 重读代码选项 其它值: 锁定

#### IAP\_CMDL

位编号	7	6	5	4	3	2	1	0
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
位符号	IAP_CMDL[7:0]							

位编号	位符号	说明
7-0	IAP_CMDL[7:0]	IAP_CMDH[7:0]反码 注: 写入 IAP_CMDL[7:0]数据必须为之前写入 IAP_CMDH[7:0]的反码, 否则将锁定相关操作, 即相关操作会失败。

操作示例:

#### 1、程序空间扇区擦除

```

IAP_CMDH = 0xF0;
IAP_CMDL = 0x0F;
IAP_ADDRH = 0x80;
IAP_ADDRH = 0x00; //选择第1扇区被擦除, 一个扇区为128字节
IAP_CMDH = 0xD2; //选择操作方式, 扇区擦除
IAP_CMDL = 0x2D;
IAP_CMDH = 0xE1; //触发
IAP_CMDL = 0x1E; //触发后 IAP_ADDRH 指向 0xFF, IAP_ADDRH 指向 0x3F, 同时自动锁定
    
```

#### 2、程序空间字节编程

```

IAP_DATA = 0x02; //待编程数据, 写入数据寄存器必须放在解锁之前
IAP_CMDH = 0xF0;
IAP_CMDL = 0x0F;
IAP_ADDRH = 0x00;
    
```

```
IAP_ADDRL = 0x00;  
IAP_CMDH = 0xB4; //选择操作方式, 字节编程  
IAP_CMDL = 0x4B;  
IAP_CMDH = 0xE1; //触发  
IAP_CMDL = 0x1E; //触发后 IAP_ADDRL 指向 0xFF, IAP_ADDRH 指向 0x3F, IAP_DATA 指  
向 0x00, 同时自动锁定
```

注：解锁之后，写地址、选择操作方式、触发这三个步骤之间不能插入任何指令，必须连续操作。

### 3、软件复位(不重读代码选项)

```
IAP_CMDH = 0xF0;  
IAP_CMDL = 0x0F;  
IAP_CMDH = 0x87;  
IAP_CMDL = 0x78;
```

### 4、软件复位(重读代码选项)

```
IAP_CMDH = 0xF0;  
IAP_CMDL = 0x0F;  
IAP_CMDH = 0x78;  
IAP_CMDL = 0x87;
```

### 3.1.6 FLASH ICP 操作

#### 3.1.6.1 JTAG方式

用户可以通过 HC-LINK 仿真器对 MCU 进行编程，当 MCU 已经焊在用户板上后，如果用户采用上电复位的方式进入 JTAG，只需要链接 6 根线，用户系统必须断电，由仿真器提供电源。当用户系统不希望掉电时，可以采用 7 根线进入编程模式，多了一个复位引脚，仿真器更详细的使用说明请参见 HC-LINK 用户手册。

另外，因为编程信号非常敏感，用户需要用 6 个跳线将编程引脚（VDD、TDO、TDI、TMS、TCK、 $\overline{\text{RST}}$ ）从应用电路中分离出来，如下图所示。

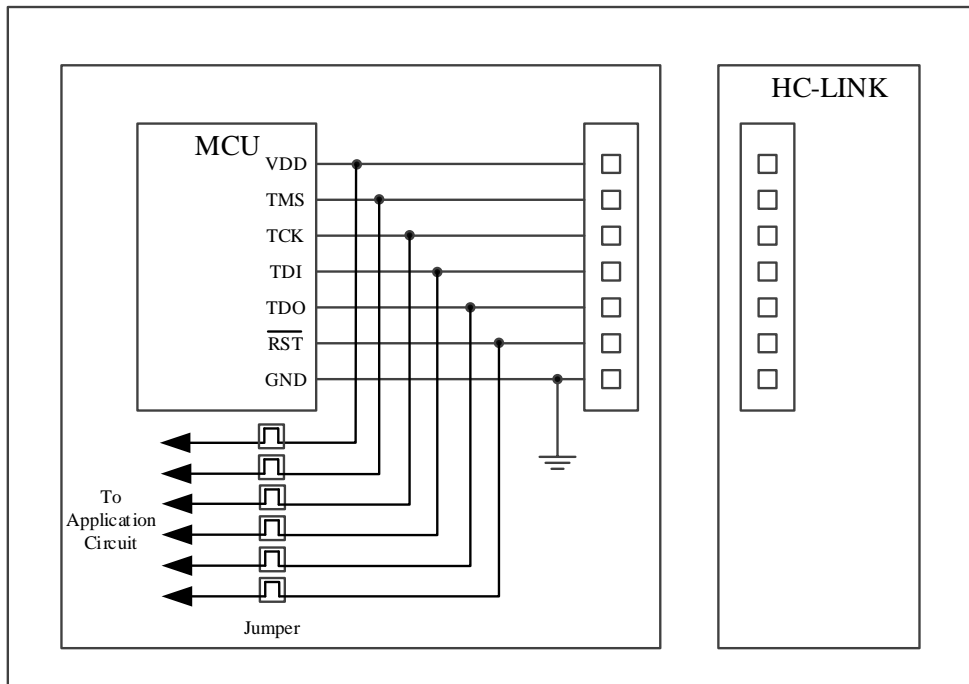


Figure 3-1 HC-LINK 编程硬件连接

### 3.1.6.2 双线方式

用户可以通过 HC-LINK 仿真器的双线方式对 MCU 进行仿真和编程，当 MCU 已经焊在用户板上后，如果用户采用上电复位的方式，只需要连接四根线（VDD、GND、SDA、SCK），用户系统必须断电，由仿真器提供电源。当用户系统不希望掉电时，可以采用五根线进入编程模式，多了一个复位引脚，仿真器更详细的使用说明请参见 HC-LINK 用户手册。

另外，因为编程信号非常敏感，用户需要用 4 个跳线将编程引脚（VDD、SDA、SCK、 $\overline{RST}$ ）从应用电路中分离出来，如下图所示。另外，如果使用外部复位引脚进入，也需要将外部复位引脚进行跳线分离。

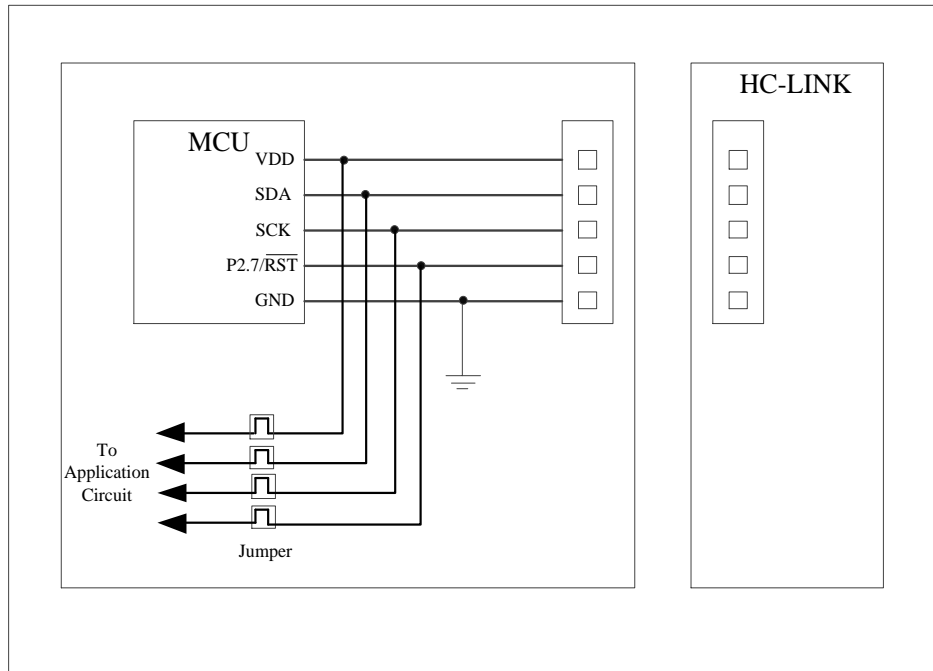


Figure 3-2 HC-LINK 编程硬件连接

当采用 ICP 模式进行操作时，建议按照如下步骤进行操作：

- 1、在开始编程前断开跳线（Jumper），从应用电路中分离编程引脚。
- 2、将芯片编程引脚连接至 Flash 编程器接口，开始编程。
- 3、编程结束后断开 Flash 编程器接口，连接跳线恢复应用电路。

### 3.1.7 第二复位向量操作

如果用户在代码选项中配置了第二复位向量使能和第二复位向量地址，那么芯片上电复位后，PC 会首先指向第二向量地址，开始执行用户的启动程序，用户启动程序的最后需要放置一条不重读代码选项的软件复位程序，那用户就会复位到 0x0000H 处，开始执行用户应用程序。

此功能也可以用来实现 ISP 功能，用户自己编写 ISP 引导程序，然后将 ISP 引导程序下载到第二复位向量地址开始的 FLASH 中，并使能第二复位向量操作。这样用户就可以通过自己编写的 ISP 引导程序完成用户应用程序的更新，这种方式只支持用户应用程序的更新，不支持代码选项的更新。

### 3.2 数据存储器（RAM）

HC89F0431A/0421A/0411A 为用户提供了 256 Bytes 内部 RAM 和 768Bytes 内部扩展 RAM 来作为数据存储器。下图为数据存储器空间分配。

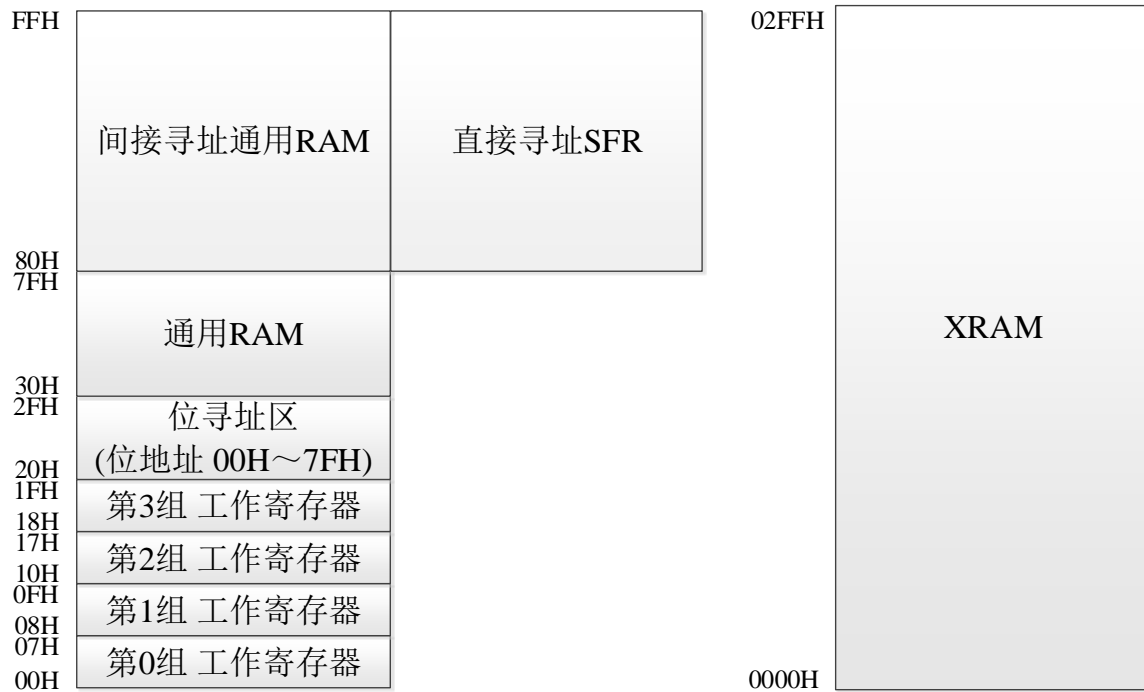


Figure 3-3 数据存储器示意图

内部 RAM 的高 128 Bytes (0x80 ~ 0xFF) 必须采用寄存器间接寻址方式。

内部扩展 RAM (XRAM) 的地址范围是 0x0000~0x02FF，访问内部扩展 RAM 的方法和传统 8051 单片机访问外部扩展 RAM 的方法相同，但是不影响 I/O 口。在汇编语言中，内部扩展 RAM 通过 MOVX 指令访问，即 MOVX @DPTP 或者 MOVX @Ri。

## 3.3 特殊功能寄存器（SFR）

### 3.3.1 特殊功能寄存器列表

#### 3.3.1.1 直接寻址读写SFR

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	RSTFR	IAP_ADDRL	IAP_ADDRH	IAP_DATA	IAP_CMDL	IAP_CMDH	-	-
F0	B	PWM2EN	PWM2PL	PWM2PH	PWM2DL	PWM2DH	PWM2DTL	PWM2DTH
E8	-	PWM1EN	PWM1PL	PWM1PH	PWM1DL	PWM1DH	PWM1DTL	PWM1DTH
E0	ACC	PWM0EN	PWM0PL	PWM0PH	PWM0DL	PWM0DH	PWM0DTL	PWM0DTH
D8	-	-	PWM0C	PWM1C	PWM2C	PWM3C	PWM3P	PWM3D
D0	PSW	LCDCON	-	-	-	-	-	-
C8	-	T3CON	TL3	TH3	T4CON	TL4	TH4	-
C0	-	T5CON	TL5	TH5	RCAP5L	RCAP5H	-	-
B8	IE1	IP2	IP3	LVDC	LVDCMP	WDTC	CRCL	CRCH
B0	-	IP4	-	-	ADCC0	ADCC1	ADCRL	ADCRH
A8	IE	IP0	IP1	SPDAT	SPCTL	SPSTAT	IICDAT	IICADR
A0	P2	-	INSCON	-	-	-	IICCON	IICSTA
98	SCON	SBUF	SADDR	SADEN	-	-	SCON2	-
90	P1				-	-	PINTF0	PINTF1
88	TCON	TMOD	TL0	TL1	TH0	TH1	CLKSWR	CLKCON
80	P0	SP	DPL	DPH	-	-	-	PCON

#### 3.3.1.2 外部扩展XSFR

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

比如写一个地址为 0xFE88 的 XSFR，操作如下：

```
MOV    A, #wdata
MOV    DPTR, #0xFE88
MOVX   @DPTR, A
```

读地址为 0xFE89 的 XSFR，操作如下：

```
MOV    DPTR, #0xFE89
MOVX   A, @DPTR
```

使用 C 语言来编程时，只需要 #define ALLOCATE\_EXTERN，并且 #include "HC89F0431A.h"，就可以像操作直接寻址寄存器一样，直接赋值 XSFR，比如：

```
ADCC2 = 0x4D;
```

**扩展XSFR (基地址0xFE80)**

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x0000	TCON1	0x0010	-	0x0020	WDTCCR	0x0030	PITS0
0x0001	-	0x0011	CLKDIV	0x0021	-	0x0031	PITS1
0x0002	-	0x0012	FREQ_CLK	0x0022	CRCC	0x0032	PITS2
0x0003	-	0x0013	CLKOUT	0x0023	-	0x0033	PITS3
0x0004		0x0014	XTALCFG	0x0024	BORC	0x0034	-
0x0005	T5CON1	0x0015	SPOV_RSTEN	0x0025	BORDBC	0x0035	-
0x0006	T5CON2	0x0016	XTALCFG1	0x0026	-	0x0036	-
0x0007	-	0x0017	PORB_IAP	0x0027	LVDDBC	0x0037	-
0x0008	S2CON	0x0018	ADCWC	0x0028	-	0x0038	PINTE0
0x0009	S2CON2	0x0019	-	0x0029	-	0x0039	PINTE1
0x000A	S2BUF	0x001A	ADCC3	0x002A	RSTDBC	0x003A	-
0x000B	BRTSEL	0x001B	ADCC2	0x002B		0x003B	-
0x000C	-	0x001C	ADCDLYH	0x002C		0x003C	INT01_PINS
0x000D	-	0x001D	ADCDLYL	0x002D		0x003D	TRMEN
0x000E	-	0x001E		0x002E		0x003E	TRMV
0x000F	-	0x001F		0x002F		0x003F	

**扩展XSFR (基地址0xFEC0)**

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x0000	-	0x0010	-	0x0020	-	0x0030	-
0x0001	-	0x0011	-	0x0021	-	0x0031	-
0x0002	-	0x0012	-	0x0022	-	0x0032	-
0x0003	-	0x0013	-	0x0023	-	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	-	0x0015	-	0x0025	-	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	-	0x0018	-	0x0028	-	0x0038	PWMENA
0x0009	-	0x0019	-	0x0029	-	0x0039	PWMCON0
0x000A	-	0x001A	-	0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

**扩展XSFR (基地址0xFF00)**

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x0000	P0M0	0x0010	P2M0	0x0020	-	0x0030	-
0x0001	P0M1	0x0011	P2M1	0x0021	-	0x0031	-
0x0002	P0M2	0x0012	P2M2	0x0022	-	0x0032	-
0x0003	P0M3	0x0013	P2M3	0x0023	-	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	P0LPU	0x0015	-	0x0025	-	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	P1M0	0x0018		0x0028	-	0x0038	-
0x0009	-	0x0019		0x0029	-	0x0039	-
0x000A	-	0x001A		0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

**扩展XSFR (基地址0xFF40)**

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x0000	P00DBC	0x0010	P0OUT	0x0020	COMP0EN	0x0030	-
0x0001	P01DBC	0x0011	P1OUT	0x0021	COMP1EN	0x0031	-
0x0002	P02DBC	0x0012	P2OUT	0x0022	COMP2EN	0x0032	-
0x0003	-	0x0013	-	0x0023	-	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	-	0x0015	-	0x0025	-	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	-	0x0018	-	0x0028	-	0x0038	--
0x0009	-	0x0019	-	0x0029	-	0x0039	-
0x000A	-	0x001A	-	0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-



## 扩展XSFR（基地址0xFF80）

偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称	偏移地址	XSFR 名称
0x0000	T0_MAP	0x0010	PWM0_MAP	0x0020	TXD_MAP	0x0030	-
0x0001	T1_MAP	0x0011	PWM01_MAP	0x0021	RXD_MAP	0x0031	-
0x0002	-	0x0012	-	0x0022	SCL_MAP	0x0032	-
0x0003	T3_MAP	0x0013	-	0x0023	SDA_MAP	0x0033	-
0x0004	T4_MAP	0x0014	PWM1_MAP	0x0024	$\overline{SS}$ _MAP	0x0034	-
0x0005	T5_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0x0006	-	0x0016	-	0x0026	MOSI_MAP	0x0036	-
0x0007	-	0x0017	-	0x0027	MISO_MAP	0x0037	-
0x0008		0x0018	PWM2_MAP	0x0028	TXD2_MAP	0x0038	-
0x0009		0x0019	PWM21_MAP	0x0029	RXD2_MAP	0x0039	-
0x000A		0x001A	-	0x002A	-	0x003A	-
0x000B		0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	PWM3_MAP	0x002C	-	0x003C	-
0x000D		0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	CLKO_MAP	0x001F	-	0x002F	-	0x003F	-

## 4 系统时钟

### 4.1 系统时钟特性

HC89F0431A/0421A/0411A 单片机系统时钟有 4 种时钟源可选：

- 外部高频晶振时钟（4MHz~20MHz）
- 外部低频晶振时钟（32.768KHz）
- 内部高频 RC 时钟（32MHz）
- 内部低频 RC 时钟（44KHz）

用户选择后的系统时钟（如果选择的是内部高频 RC，则经 RC32M\_DIV[1:0]分频后的时钟）记做  $osc\_clk$ ，其频率为  $F_{osc}$ ，周期为  $T_{osc}$ ，主要用于外设模块， $osc\_clk$  可以进行 1-255 之间任意值的分频，分频后的时钟记做 CPU 时钟，其频率为  $F_{cpu}$ ，周期为  $T_{cpu}$ 。

芯片上电复位后，默认选择内部高频 RC 作为系统时钟，其  $F_{osc}$  为 4MHz， $F_{cpu}$  为 2MHz，可以通过配置相关寄存器改变  $osc\_clk$  和  $cpu\_clk$  的频率。

CPU 最高可以运行在 16MHz 频率下，如果所选时钟源频率高于 16MHz，需要对其进行分频，使 CPU 时钟频率等于或低于 16MHz。

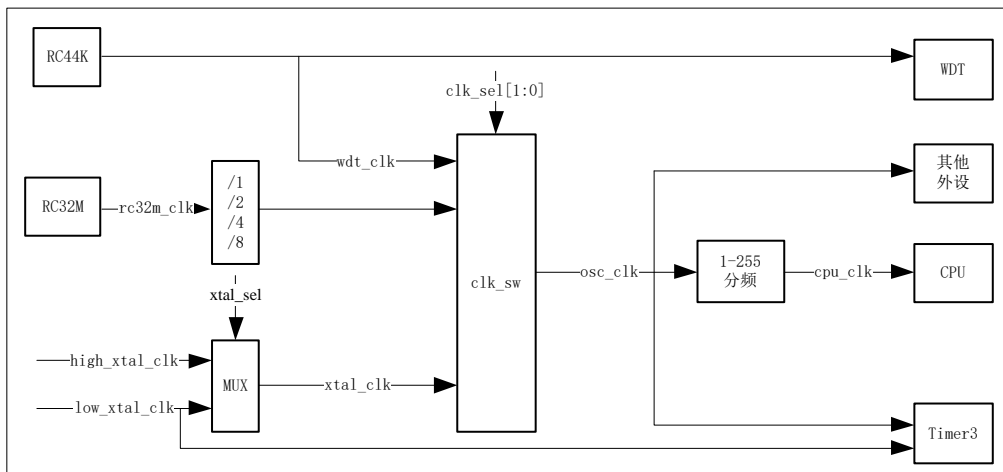


Figure 4-1 系统时钟框图

## 4.2 内部时钟

内部时钟有内部高频 RC（RC32M）和内部低频 RC（RC44K）两种，用户可通过软件进行选择。

内部低频 RC（RC44K）输出的时钟记做 `wdt_clk`，用于看门狗定时器的计数，也可以用于系统时钟；内部高频 RC（RC32M）输出的时钟记做 `rc32m_clk`，可以进行 1/2/4/8 分频。

## 4.3 外部时钟

外部时钟有外部高频晶振时钟（4MHz~20MHz）和外部低频晶振（32.768KHz）两种，用户可通过软件进行选择。经过 XTALCFG 寄存器选择的外部晶振时钟记做 `xtal_clk`。

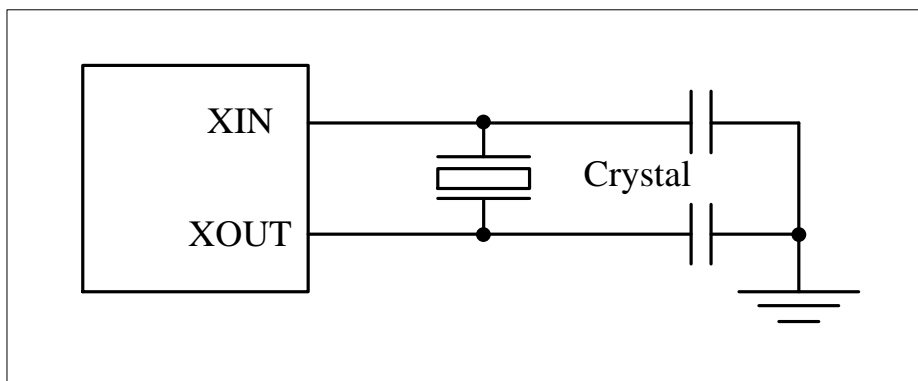


Figure 4-2 外部晶振典型应用

使用注意：

- 1、晶振起振电容推荐值为 20pF，该值可通过晶振基本的起振和运行测试，并非最优值。
- 2、外部晶振和 XIN、XOUT 端口之间的物理距离应在 10mm 以内。
- 3、使用外部晶振前，应充分了解所选晶振相关应用参数和要求，以获得最佳性能。

## 4.4 系统时钟相关寄存器

### 4.4.1 时钟控制寄存器 CLKCON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	R
复位值	0	0	1	1	0	0	1	0
位符号	HXTAL RDY	LXTAL RDY	HSRC RDY	LSRC RDY	-	XTALEN	HSRCEN	-

位编号	位符号	说明
7	HXTALRDY	外部高频晶振状态位 0: 外部高频晶振未准备 1: 外部高频晶振准备就绪 注: 该位硬件自动清 0 或置 1。
6	LXTALRDY	外部低频晶振状态位 0: 外部低频晶振未准备 1: 外部低频晶振准备就绪 注: 该位硬件自动清 0 或置 1。
5	HSRCRDY	内部高频 RC 振荡器状态位 0: 内部高频 RC 未准备 1: 内部高频 RC 准备就绪 注: 该位硬件自动清 0 或置 1。
4	LSRCRDY	内部低频 RC 振荡器状态位 0: 内部低频 RC 未准备 1: 内部低频 RC 准备就绪 注: 该位硬件自动清 0 或置 1。
3	-	保留位
2	XTALEN	外部晶振使能位 0: 外部晶振关闭 1: 外部晶振打开 注: 使能时, 需要软件将对应管脚的 IO 模式设置为模拟通道。
1	HSRCEN	内部高频 RC 振荡器使能位 0: 内部高频 RC 关闭 1: 内部高频 RC 打开
0	-	保留位

#### 4.4.2 时钟选择寄存器 CLKSWR

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
复位值	0	1	0	1	0	0	1	1
位符号	CLKSTA[1:0]		CLKSEL[1:0]		-		RC32M_DIV[1:0]	

位编号	位符号	说明
7-6	CLKSTA[1:0]	系统时钟状态位 00: 当前系统时钟为内部低频 RC 01: 当前系统时钟为内部高频 RC 10: 当前系统时钟为外部低频晶振 11: 当前系统时钟为外部高频晶振 注: 系统根据当前系统时钟自动切换各个状态
5-4	CLKSEL[1:0]	系统时钟选择位 00: 选择系统时钟为内部低频 RC 01: 选择系统时钟为内部高频 RC 1x: 选择系统时钟为外部晶振 注: 系统时钟选择时, 必须对应的时钟源状态位为 1, 否则将延续之前时钟, 切换后, 原时钟不会自动关闭; 选择后的系统时钟记做 $osc\_clk$ , 其频率为 $F_{osc}$ , 周期为 $T_{osc}$ 。
3-2	-	保留位
1-0	RC32M_DIV[1:0]	内部高频 RC 分频系数 00: $rc32m\_clk$ 01: $rc32m\_clk / 2$ 10: $rc32m\_clk / 4$ 11: $rc32m\_clk / 8$ (默认)

#### 4.4.3 时钟分频寄存器 CLKDIV

位编号	7	6	5	4	3	2	1	0
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	1	0
位符号	CLKDIV[7:0]							

位编号	位符号	说明
7-0	CLKDIV[7:0]	CPU 时钟分频系数, 默认为 2 分频 配置值为 0 或 1 时, 时钟不分频; 其他情况下, 配置值等于分频系数; 注: 分频后的时钟为 CPU 时钟, 其频率为 $F_{cpu}$ , 周期为 $T_{cpu}$ 。

#### 4.4.4 时钟输出寄存器 CLKOUT

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-			CLKOEN	-	CLKOSEL[2:0]		

位编号	位符号	说明
7-5	-	保留位
4	CLKOEN	时钟输出使能位 0: 禁止时钟输出 1: 允许时钟输出
3	-	保留位
2-0	CLKOSEL[2:0]	时钟输出选择位 000: 选择 cpu_clk 001: 选择 osc_clk 010: 选择 wdt_clk 011: 选择 xtal_clk 100: 选择 rc32m_clk 101: 选择 rc32m_clk/2 110: 选择 rc32m_clk/4 111: 选择 rc32m_clk/8

#### 4.4.5 外部晶振配置寄存器 XTALCFG

位编号	7	6	5	4	3	2	1	0
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
位符号	HXTALCNT[1:0]		LXTALCNT[1:0]		HXTALMSEL[1:0]		RC_PD_EN	XTALSEL

位编号	位符号	说明
7-6	HXTALCNT[1:0]	外部高频晶振 warmup 计数值选择 00: 2048 01: 256 10: 16384 11: 65536
5-4	LXTALCNT[1:0]	外部低频晶振 warmup 计数值选择 00: 16384 01: 4096 10: 1024 11: 65536
3-2	HXTALMSEL[1:0]	外部高频晶振的选择信号 00: 选择 4M/8M 晶振

		01: 选择 4M/8M 晶振大驱动能力模式, 在低压工作时, 起振时间短, 且功耗适中 11: 选择 16M/20M 晶振
1	RC_PD_EN	系统产生 BOR 复位时, 高频内部 RC 关闭信号 0: 系统产生 BOR 复位时, 不关闭高频内部 RC 1: 系统产生 BOR 复位时, 关闭高频内部 RC 注意: 此位可用于 BOR 使能的情况下, 降低 VDD 下降过程中的系统功耗。
0	XTALSEL	外部晶振选择位 0: 外部低频晶振 32.768KHz 1: 外部高频晶振

#### 4.4.6 时钟频率寄存器 **FREQ\_CLK**

在进行 FLASH 的 IAP 擦写或者系统进入掉电模式之前, 需要配置扩展 SFR 里 FREQ\_CLK 寄存器, 指明目前 CPU 时钟的频率, FREQ\_CLK 寄存器配置的值等于 CPU 时钟的频率值, 最小为 1MHz, 假如目前 CPU 的运行频率为 16MHz, 那就配置寄存器 FREQ\_CLK=0x10。

#### **FREQ\_CLK**

位编号	7	6	5	4	3	2	1	0
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	1	0
位符号	FREQ_CLK[7:0]							

位编号	位符号	说明
7-0	FREQ_CLK[7:0]	当前 CPU 时钟频率寄存器 举例如下: CPU 频率为 16MHz 时, 配置值为 0x10 CPU 频率为 8MHz 时, 配置值为 0x08 CPU 频率为 4MHz 时, 配置值为 0x04 CPU 频率为 2MHz 时, 配置值为 0x02 CPU 频率小于等于 1MHz 时, 配置值为 0x01

#### 4.4.7 内部高频 RC 调整使能寄存器 TRMEN

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号								RCTRMEN

位编号	位符号	说明
7-1	-	保留位
0	RCTRMEN	内部高频 RC 调整使能位 1: 使能内部高频 RC 调整 0: 禁止内部高频 RC 调整 注: 使能该寄存器后, 必须立即配置 TRMV 寄存器, 否则这个使能寄存器再执行完下一条指令后会被清零, 内部高频 RC 调整就会失效。

#### 4.4.8 内部高频 RC 调整配置寄存器 TRMV

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	x	x	x	x	x	x	x
位符号	-	RCTRMV						

位编号	位符号	说明
7	-	保留位
6-0	RCTRMV	内部高频 RC 调整配置值 注: 1. x 表示不确定的值, 此寄存器的上电复位值为出厂的校准值。 2. 在配置这个寄存器值时, 需要先将内部高频 RC 调整使能位配置为 1。 3. 根据校准曲线软件先使能 RCTRMEN, 紧接着就要配置 RCTRMV, 在调整完成后 RCTRMEN 自动清零, 防止重复操作



## 5 电源管理

### 5.1 电源管理特性

- 提供空闲模式（IDLE）和掉电模式（PD），作为省电模式
- 提供多种方式从空闲/掉电模式唤醒
- 提供低频模式（即时钟分频，详见系统时钟章节相关介绍）

### 5.2 空闲模式

空闲模式能够降低系统功耗，在此模式下，程序中止运行，CPU时钟停止，但外部设备时钟可继续运行。空闲模式下，CPU在确定的状态下停止，并在进入空闲模式前所有CPU的状态都被保存，如PC、PSW、SFR、RAM等。

将PCON寄存器中的IDL位置1，使HC89F0431A/0421A/0411A进入空闲模式。IDL位置1是CPU进入空闲模式之前执行的最后一条指令。

两种方式可以退出空闲模式：

(1) 所有的有效中断。HC89F0431A/0421A/0411A在检测到一个有效中断后，CPU时钟立即恢复，硬件清除PCON寄存器的IDL位，然后执行中断服务程序，随后跳转到进入空闲模式指令之后的指令。

(2) 复位信号（外部复位引脚上出现有效电平、WDT复位、BOR复位或外部端口低压检测复位）。HC89F0431A/0421A/0411A在检测到有效复位后，PCON寄存器中的IDL位被复位成零，系统程序也会从复位地址0000H处开始执行，RAM保持不变，SFR的值根据不同功能模块改变。

### 5.3 掉电模式

掉电模式可以使HC89F0431A/0421A/0411A进入功耗非常低的状态。掉电模式将停止CPU和外围设备的所有时钟信号，但如果WDT和TIMER3使能且允许在掉电模式下工作，则WDT和TIMER3模块将继续工作。在进入掉电模式前所有CPU的状态都被保存，如PC、PSW、SFR、RAM等。

在芯片进入掉电模式之前，需要配置扩展SFR里FREQ\_CLK寄存器，指明目前CPU时钟的频率，FREQ\_CLK寄存器配置的值等于CPU时钟的频率值，最小为1MHz，假如目前CPU的运行频率为16MHz，那就配置寄存器FREQ\_CLK=0x10。

将PCON寄存器中的PD位置1，使HC89F0431A/0421A/0411A进入掉电模式。PD位置1是CPU进入掉电模式之前执行的最后一条指令。

注：如果同时设置IDL位和PD位，HC89F0431A/0421A/0411A进入掉电模式。退出掉电模式后，CPU也不会进入空闲模式，从掉电模式退出后硬件会清除IDL及PD位。

多种方式可以退出掉电模式：

(1) 有效外部中断、LVD中断、WDT中断及TIMER3(计数时钟源选择外部低频晶振、外部时钟或RC44K)中断。在有效的外部中断和TIMER3中断发生后，内部高频RC振荡器启动，CPU时钟和外设时钟立即恢复，PCON寄存器中的PD位会被硬件清除，然后程序运行外部中断服务程序。在完成外部中断服务程序之后，跳转到进入掉电模式之后的指令继续运行。

(2) 复位信号（外部复位引脚上出现有效电平、WDT 复位、BOR 复位或外部端口低压检测复位）。有效的复位信号将 PCON 寄存器中的 PD 位复位成零，振荡器重新启动，CPU 时钟和外设时钟立即恢复，系统也会从复位地址 0000H 处开始运行，RAM 保持不变，SFR 的值根据不同功能模块改变。

## 5.4 电源管理相关寄存器

### 5.4.1 电源控制寄存器 PCON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-				GF1	GF0	PD	IDL

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3	GF1	用户通用标志位 1
2	GF0	用户通用标志位 0
1	PD	掉电模式控制位 0: 正常工作模式 1: 进入掉电模式（退出该模式后自动清 0）
0	IDL	空闲模式控制位 0: 正常工作模式 1: 进入空闲模式（退出该模式后自动清 0） 注：若同时置 PD&IDL，系统将进入掉电模式，唤醒后标志被同时清除。

## 6 复位

### 6.1 复位特性

- 提供多种方式复位
- 所有的复位方式都有特定标志

### 6.2 POR（Power-On Reset）复位

HC89F0431A/0421A/0411A单片机在上电过程中，会产生一个POR信号，此信号会复位单片机，同时置位RSTFR寄存器里的PORF位，用户可以判断此标志以来确定是否发生POR复位。

注：POR复位后的RAM值不稳定，建议用户根据需要重新初始化相应RAM；其余复位方式不会对RAM进行复位。

### 6.3 BOR（Brown-Out Reset）复位

当VDD电压下降到 $V_{BOR}$ 以下，且持续时间超过 $T_{BOR}$ 时，系统产生欠压复位。BOR复位时，RSTFR寄存器的BORF位将被置1，用户可以判断此标志以来确定是否发生BOR复位。

HC89F0431A/0421A/0411A可以通过代码选项或者寄存器来选择BOR检测的电压档位。当在代码选项中配置完成BOR档位后，客户也可以在使用时根据情况通过配置寄存器重新配置合适的BOR检测电压。BOR档位：4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V。

BOR电压检测电路有一定的迟滞特性，迟滞电压为0.1V左右。即当VDD电压下降到所选BOR电压档位时BOR复位有效，而VDD电压需要上升到BOR档位电压+0.1V时BOR复位才会解除。

欠压复位示意图如下所示，其中 $T_{BOR}$ 也可以通过寄存器配置，用来进行电压消抖。

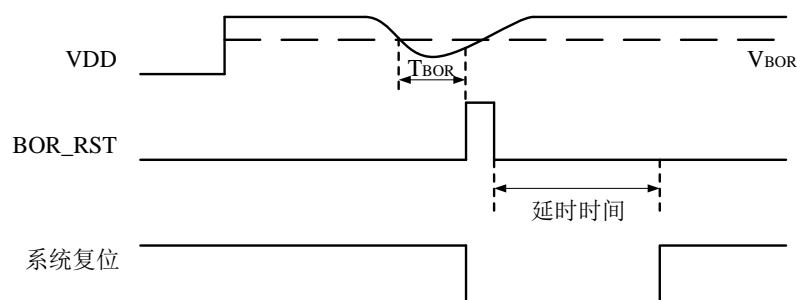


Figure 6-1 BOR 示意图

## 6.4 外部 RST 复位

外部 $\overline{\text{RST}}$ 引脚复位就是从外部向 $\overline{\text{RST}}$ 引脚施加一定宽度的复位脉冲，从而实现单片机的复位，不使用时可以将其配置为I/O口，需要在代码选项中设置。

做 $\overline{\text{RST}}$ 端口时，将 $\overline{\text{RST}}$ 复位管脚拉低并维持至少设定时间（软件配置）后，单片机才会进入复位状态，将 $\overline{\text{RST}}$ 复位管脚拉回高电平后，单片机结束复位状态并从用户程序区的0000H处开始正常工作。RST复位时，RSTFR寄存器的EXRSTF将被置1，用户可以判断此标志以来确定是否发生外部RST复位。

注意：1、P2.7端口作为外部 $\overline{\text{RST}}$ 复位端口时，无法作为普通I/O使用。

2、如果系统使能了外部RST功能且外部 $\overline{\text{RST}}$ 端口处于有效复位状态，此时系统无法进入仿真或烧录模式。

## 6.5 外部端口低压检测复位

当外部电压过低时，无法保证单片机正常工作。此时，可以利用单片机的外部端口低压检测（PLVD）功能对单片机进行复位，外部端口检测的电压为1.2V，此复位功能可以被禁止。PLVD复位时，RSTFR寄存器的PLVRSTF将被置1，用户可以判断此标志以来确定是否发生外部端口低压检测复位。另外，用户也可以通过相关寄存器来对外部端口电压检测进行消抖。

## 6.6 软件复位

对IAP\_CMDH和IAP\_CMDL寄存器按流程写入相关值，系统将产生软件复位，复位后RSTFR寄存器的SWRF将被置1，用户可以判断此标志以来确定是否发生软件复位。具体操作详见FLASH IAP操作章节相关介绍。

软件复位前建议将系统时钟切换到内部高频RC。软件复位不会切换切换系统时钟，但会将CLKSWR寄存器里的RC32M\_DIV[1:0]复位成01B，CLKDIV寄存器复位成08H。

## 6.7 看门狗（WDT）复位

为了防止系统在异常情况下受到干扰，MCU程序跑飞，导致系统长时间异常工作，通常是引进看门狗，如果MCU程序中不在规定的时间内按要求操作看门狗，就认为MCU处于异常状态，看门狗就会强制MCU复位，芯片重新从0000H开始运行。

注：要使WDT复位，必须置WDTRST为1，即允许WDT复位功能，否则即使允许WDT运行，WDT也只会置溢出标志，并不会复位。

## 6.8 堆栈溢出复位

堆栈溢出时，系统将复位，并置SPOVF溢出标志，必须软件清除。

堆栈溢出包含入堆溢出及出栈溢出，入栈溢出是指当前栈顶地址为0xFF，同时又有入栈动作；出栈溢出是指当前栈顶地址等于用户设定的栈底地址，同时又有出栈动作。

堆栈溢出复位配置有使能寄存器，当使能时，堆栈溢出才能复位系统。

## 6.9 复位相关寄存器

### 6.9.1 复位标志寄存器 RSTFR

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
POR复位	1	x	x	x	x	0	0	0
EXRST复位	u	1	u	u	u	0	u	u
BOR复位	u	u	1	u	u	0	u	u
WDT复位	u	u	u	1	u	0	u	u
软复位	u	u	u	u	1	0	u	u
堆栈溢出复位	u	u	u	u	u	0	1	u
PLVD复位	u	u	u	u	u	0	u	1
位符号	PORF	EXRSTF	BORF	WDTRF	SWRF	-	SPOVF	PLVRSTF

注：x表示不确定的值，u表示该值由当前复位方式前的值决定，建议在POR复位后清零一下该寄存器。

位编号	位符号	说明
7	PORF	上电复位标志位 0: 无上电复位 1: 发生上电复位，软件清0
6	EXRSTF	外部 RST 复位标志位 0: 无外部 RST 复位 1: 发生外部 RST 复位，软件清0
5	BORF	欠压复位标志位 0: 无欠压复位 1: 发生欠压复位，软件清0
4	WDTRF	WDT 复位标志位 0: 无 WDT 复位 1: 发生 WDT 复位，软件清0
3	SWRF	软件复位标志位 0: 无软件复位 1: 发生软件复位，软件清0
2	-	保留
1	SPOVF	堆栈溢出标志位 0: 无堆栈溢出复位 1: 堆栈溢出复位，软件清0
0	PLVRSTF	外部端口电压检测复位标志位 0: 外部端口电压检测复位 1: 发生外部端口电压检测复位，软件清0

## 6.9.2 BOR 电压检测控制寄存器 BORC

### BORC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
复位值	1	0	0	0	0	0	0	0
位符号	BOREN	BOR_DBC_EN	BOR_PD_EN	-		BORVS[2:0]		

位编号	位符号	说明
7	BOREN	BOR 使能位 0: 禁止 BOR 1: 允许 BOR
6	BOR_DBC_EN	BOR 消抖使能位 0: 不使能 1: 使能
5	BOR_PD_EN	BOR 电源使能位 0: 不使能 1: 使能 注: 此位使能后, 当 BOR 复位时, 芯片进入 STOP 模式
4-3	-	保留位 (读为 0, 写无效)
2-0	BORVS[2:0]	BOR 检测电压点选择位 000: 1.8V 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V

## 6.9.3 BOR 电压检测去抖控制寄存器 BORDBC

位编号	7	6	5	4	3	2	1	0
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
位符号	BORDBC[7:0]							

位编号	位符号	说明
7-0	BORDBC[7:0]	BOR 消抖控制位 消抖时间 = BORDBC[7:0] * 8T <sub>CPU</sub> + 2 T <sub>CPU</sub> 注: 需要使能 BOR_DBC_EN, 否则 BOR 不消抖。

注: 掉电模式下自动关闭 BOR 消抖功能, 退出掉电模式自动打开。

### 6.9.4 外部 RST 去抖控制寄存器 RSTDBC

位编号	7	6	5	4	3	2	1	0
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
位符号	RSTDBC[7:0]							

位编号	位符号	说明
7-0	RSTDBC[7:0]	外部 RST 消抖控制位 消抖时间 = RSTDBC[7:0] * 8T <sub>CPU</sub> + 2 T <sub>CPU</sub>

注：掉电模式下自动关闭外部 RST 消抖功能，退出掉电模式又自动打开。

### 6.9.5 堆栈溢出复位使能寄存器 SPOV\_RSTEN

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-							SPOV_RSTEN

位编号	位符号	说明
7-1	-	保留位（读为 0，写无效）
0	SPOV_RSTEN	堆栈溢出复位使能位 0：不使能堆栈溢出复位 1：使能堆栈溢出复位

### 6.9.6 PORB\_IAP 寄存器

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-							PORB_IAP

位编号	位符号	说明
7-1	-	保留位
0	PORB_IAP	PORB_IAP 标志位 当此位为 1 时，表示电压高于 1.6V，上电复位结束

## 7 通用及复用I/O

### 7.1 通用及复用 I/O 特性

- 提供最多 18 个双向 I/O 端口
- 多种模式可配

### 7.2 I/O 模式

HC89F0431A/0421A/0411A 所有 I/O 口均可由软件配置成多种工作类型之一，具体为：输入、带上拉输入、带下拉输入、模拟输入、强推挽输出、开漏输出和开漏带上拉输出，并且输入可以配置为施密特输入。

如果 P2.7 被配置为复位脚，其端口是施密特输入上拉状态。

HC89F0431A/0421A/0411A在输入模式时（不包含模拟输入），任何读操作，数据来源都来自引脚电平。而在输出模式时，通过指令来区分读数据来源，采用“读-修改-写”指令时，为读寄存器值，其它指令为读引脚电平。

HC89F0431A/0421A/0411A增加了一组只读寄存器P0OUT、P1OUT、P2OUT，在输出模式时，可以通过读这组寄存器直接获得写到端口数据寄存器的值。

HC89F0431A/0421A/0411A首先将需要修改的寄存器的内容读回ALU，对相应位进行修改，然后再整个写回原来的寄存器地址，完成该功能的指令就叫做“读-修改-写”指令。

“读-修改-写”指令是单片机内部自己执行的，它发生在写IO口的时候，当写IO口的时候它先把IO的当前状态读回来，根据要写的数据修改读回来的数据，再写到IO口；读引脚是直接读引脚的当前状态，当前引脚是高电平，读回来的就是高电平，低电平时读回来的就是低电平。

“读-修改-写”指令包括以下指令：INC direct、DEC direct、ANL direct,A、ANL direct,#data、ORL direct,A、ORL direct,#data、XRL direct,A、XRL direct,#data、DJNZ direct,rel、MOV bit,C、CLR bit、SETB bit、CPL bit、JBC bit,rel。详见第21章节的指令表。



### 7.3 I/O 功能框图

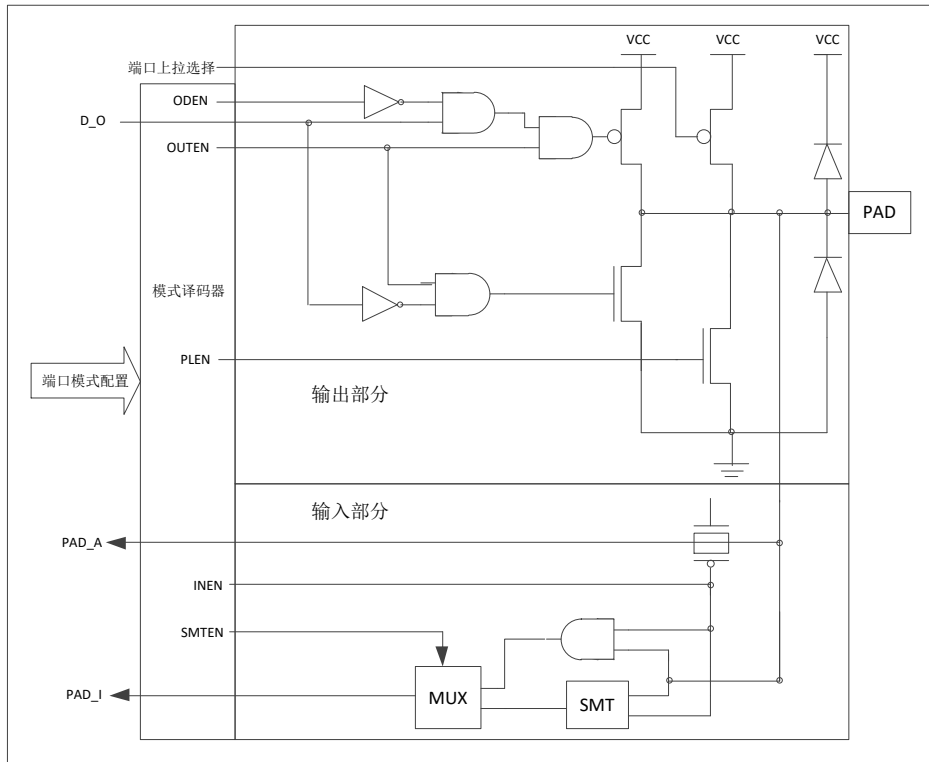


Figure 7-1 I/O 功能框图

### 7.4 I/O 端口相关寄存器

#### 7.4.1 P0 端口数据寄存器 P0

位编号	7	6	5	4	3	2	1	0
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
位符号	P0[7:0]							

位编号	位符号	说明
7-0	P0[7:0]	P0 端口数据寄存器

### 7.4.2 P1 端口数据寄存器 P1

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	-	P1[1:0]	

位编号	位符号	说明
1-0	P1[1:0]	P1 端口数据寄存器

### 7.4.3 P2 端口数据寄存器 P2

位编号	7	6	5	4	3	2	1	0
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
位符号	P2[7:0]							

位编号	位符号	说明
7-0	P2[7:0]	P2 端口数据寄存器

## 7.4.4 P0 端口功能选择寄存器 P0M0、P0M1、P0M2、P0M3

### P0M0

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P01M[3:0]				P00M[3:0]			

### P0M1

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P03M[3:0]				P02M[3:0]			

### P0M2

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P05M[3:0]				P04M[3:0]			

### P0M3

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P07M[3:0]				P06M[3:0]			

位编号	位符号	说明
7-4 3-0	P0xM[3:0] (x = 0..7)	P0.x 端口模式配置位 0000: 输入 (无 SMT) 0001: 带下拉输入 (无 SMT) 0010: 带上拉输入 (无 SMT) 0011: 模拟输入 0100: 输入 (SMT) 0101: 带下拉输入 (SMT) 0110: 带上拉输入 (SMT) 0111: 保留 (模拟输入) 1x00: 推挽输出 1x01: 开漏输出 1x10: 开漏带上拉输出 1x11: 保留 (推挽输出) 注: x 为 0 或 1。

## 7.4.5 P1 端口功能选择寄存器 P1M0

### P1M0

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P11M[3:0]				P10M[3:0]			

位编号	位符号	说明
7-4 3-0	P1xM[3:0] (x = 0...1)	P1.x 端口模式配置位 0000: 输入 (无 SMT) 0001: 带下拉输入 (无 SMT) 0010: 带上拉输入 (无 SMT) 0011: 模拟输入 0100: 输入 (SMT) 0101: 带下拉输入 (SMT) 0110: 带上拉输入 (SMT) 0111: 保留 (模拟输入) 1x00: 推挽输出 1x01: 开漏输出 1x10: 开漏带上拉输出 1x11: 保留 (推挽输出) 注: x 为 0 或 1。

## 7.4.6 P2 端口功能选择寄存器 P2M0、P2M1、P2M2、P2M3

### P2M0

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P21M[3:0]				P20M[3:0]			

### P2M1

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P23M[3:0]				P22M[3:0]			

### P2M2

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P25M[3:0]				P24M[3:0]			

**P2M3**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	0	1	1
位符号	P27M[3:0]				P26M[3:0]			

P2 组端口中 P2.7、P2.5、P2.4、P2.3 支持端口上下拉同时使能功能。P2 组其余端口不支持上述功能。具体配置说明见下：

位编号	位符号	说明
7-4 3-0	P2xM[3:0] (x = 0、1、 2、6)	P2.x 端口模式配置位 0000: 输入（无 SMT） 0001: 带下拉输入（无 SMT） 0010: 带上拉输入（无 SMT） 0011: 模拟输入 0100: 输入（SMT） 0101: 带下拉输入（SMT） 0110: 带上拉输入（SMT） 0111: 保留（模拟输入） 1x00: 推挽输出 1x01: 开漏输出 1x10: 开漏带上拉输出 1x11: 保留（推挽输出） 注：x 为 0 或 1。

位编号	位符号	说明
7-4 3-0	P2xM[3:0] (x = 3、4、 5、7)	P2.x 端口模式配置位 0000、0001、0010: 输入（无 SMT） 0011: 模拟输入 0100: 输入（SMT） 0101: 带下拉输入（SMT） 0110: 带上拉输入（SMT） 0111: 模拟通道，上下拉同时使能 1x00: 推挽输出 1101: 开漏输出 1110: 开漏带上拉输出 其他值: 系统保留，请勿操作 注：x 为 0 或 1。

## 7.4.7 端口上拉电阻选择寄存器

### P0LPU

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-		P02PU[1:0]		-			

位编号	位符号	说明
7-6	-	保留位
5-4	P02PU[1:0]	端口上拉电阻选择位 00: 50 K $\Omega$ 01: 100 K $\Omega$ 10: 150K $\Omega$ 11: 300 K $\Omega$ 注: 阻值为 VDD @5V 时参考值。
3-0	-	保留位

## 7.4.8 端口消抖控制寄存器 P00DBC、P01DBC、P02DBC

位编号	7	6	5	4	3	2	1	0
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
位符号	P0xDBCLK[1:0]			P0xDBCT[5:0]				

位编号	位符号	说明
7-6	P0xDBCLK [1:0]	端口消抖时钟选择 00: $F_{osc}/1$ 01: $F_{osc}/4$ 10: $F_{osc}/16$ 11: $F_{osc}/64$ 注: x 为 0、1 或 2。
5-0	P0xDBCT [5:0]	端口消抖计数时钟个数, 当配置为 00 时, 表示不消抖。 消抖时间是指端口输入时, 其对应端口电平所需要维持的时间, 需要注意的是, 分配到这三个管脚上的功能脚、外部中断输入、故障检测引脚也受消抖控制, 其中 P02DBC[7:0]表示 P0.2 消抖控制寄存器。 注意: $P0xDBCT [5:0]$ 配置的消抖时间是一个范围, $分频系数 * T_{osc} * P0xDBCT [5:0] - T_{osc} < 消抖时间 < 分频系数 * T_{osc} * (P0xDBCT [5:0] + 1) - T_{osc}$ 。

## 7.5 外设功能引脚全映射控制

### 7.5.1 外设功能引脚映射控制寄存器

扩展 SFR 地址	扩展 SFR 名称	扩展 SFR 地址	扩展 SFR 名称	扩展 SFR 地址	扩展 SFR 名称	扩展 SFR 地址	扩展 SFR 名称
0xFF80	T0_MAP	0x0010	PWM0_MAP	0x0020	TXD_MAP	0x0030	-
0xFF81	T1_MAP	0x0011	PWM01_MAP	0x0021	RXD_MAP	0x0031	-
0xFF82	-	0x0012	-	0x0022	SCL_MAP	0x0032	-
0xFF83	T3_MAP	0x0013	-	0x0023	SDA_MAP	0x0033	-
0xFF84	T4_MAP	0x0014	PWM1_MAP	0x0024	$\overline{SS}$ _MAP	0x0034	-
0xFF85	T5_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0xFF86	-	0x0016	-	0x0026	MOSI_MAP	0x0036	-
0xFF87	-	0x0017	-	0x0027	MISO_MAP	0x0037	-
0xFF88	-	0x0018	PWM2_MAP	0x0028	TXD2_MAP	0x0038	-
0xFF89	-	0x0019	PWM21_MAP	0x0029	RXD2_MAP	0x0039	-
0xFF8A	-	0x001A	-	0x002A	-	0x003A	-
0xFF8B	-	0x001B	-	0x002B	-	0x003B	-
0xFF8C	-	0x001C	PWM3_MAP	0x002C	-	0x003C	-
0xFF8D	-	0x001D	-	0x002D	-	0x003D	-
0xFF8E	-	0x001E	-	0x002E	-	0x003E	-
0xFF8F	CLKO_MAP	0x001F	-	0x002F	-	0x003F	-

注：以上 SFR 为外部扩展 XSFR，采用 MOVX 来进行读写。

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	1	1	0	1	1	1
位符号	-		FPORT[1:0]		-	FPIN[2:0]		

位编号	位符号	说明
7-6	-	保留位
5-4	FPORT[1:0]	映射端口选择 00: P0 01: P1 10: P2 11: P3
3	-	保留位
2-0	FPIN[2:0]	映射端口输出脚选择 FPIN[2:0] = x(x = 0...7)，表示选择对应端口名的 x(x = 0...7)脚

注：输出功能，系统将禁止其多对一映射，但输入功能，系统将允许其多对一映射。

上面寄存器的复位值为 0x37，这样复位后 IO 都为 GPIO，用户在使用外设功能脚之前必须先配置上面的寄存器，否则外设功能将无法使用。

使用举例：

将 UART1 的 TXD 和 RXD 分别映射到 P2.1 和 P2.2 上，用户在启动 UART1 之前应该配置下面命令：

```
TXD_MAP = 0x21;    // TXD-->P2.1
```

```
RXD_MAP = 0x22;    // RXD-->P2.2
```

如果用户在下一轮的设计时，需要重新将 UART1 的 TXD 和 RXD 分别映射到 P0.4 和 P0.5 上，那用户需要进行如下的配置：

```
TXD_MAP = 0x04;    // TXD-->P0.4
```

```
RXD_MAP = 0x05;    // RXD-->P0.5
```

多个输出映射到一个端口上时，只能有一个输出有效，下面是默认的优先级

优先级顺序	复用端口功能
1	PWM0
2	PWM01
3	PWM1
4	PWM11
5	PWM2
6	PWM21
7	PWM3
8	CLKO
9	T0_OUT
10	T1_OUT
11	T4_OUT
12	TXD
13	RXD
14	SCK
15	MOSI
16	MISO
17	TXD2
18	SCL
19	SDA

比如：CLKO\_MAP 配置为 0x01 选择 P0.1 口作为 CLKO 的输出口，T4\_MAP 也配置为 0x01，这个时候硬件会按上面的优先级，P0.1 将配置为 CLKO 的输出口，而 T4\_MAP 的配置无效。

当所有的端口映射控制寄存器都不等于 0x01 时，即所有的功能口都不选择 P0.1 作为输入输出口，此时这个端口的输出就是 P0 端口数据寄存器的第 1 位。

输入可以配置为多个功能从一个 PAD 引脚进入，比如：

T0\_MAP 配置为 0x23，则选择 P2.3 作为 T0 的输入口，T5\_MAP 也配置为 0x23，这样从 P2.3 端口进入的信号同时作用于 T3 和 T5。

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

在输入时，无论端口是什么功能，读端口数据寄存器都读芯片引脚上值。



## 8 中断

### 8.1 中断特性

- 17 个中断源
- 4 级中断优先级
- 16 个外部中断

### 8.2 中断汇总

中断源	向量地址	允许位	标志位	查询优先级	中断号 (C 语言)
INT0	0003H	EX0	INT0F	1(最高)	0
T0	000BH	ET0	TF0	2	1
INT1	0013H	EX1	INT1F	3	2
T1	001BH	ET1	TF1	4	3
UART1	0023H	ES1	TI/RI	5	4
WDT	002BH	EWDT	WDTRF	6	5
LVD	0033H	LVDIE	LVDF	7	6
UART2	003BH	ES2	TI/RI	8	7
SPI	0043H	ESPI	SPIF/MODF	9	8
T3	004BH	ET3	TF3	10	9
T4	0053H	ET4	TF4	11	10
PWM	005BH	PWMxIE (x = 0...3)	PWMxIF (x = 0...3)	12	11
T5	0063H	ET5	TF5	13	12
ADC	006BH	EADC	ADCIF/AMWIF	14	13
INT2-INT7	0073H	EINx (x = 2...7)	INTxF (x = 2...7)	15	14
INT8-INT15	007BH	EINx (x = 8...15)	INTxF (x = 8...15)	16	15
IIC	0083H	EIIC	SI	17	16

注：除以上允许位及标志位被置位外，要响应中断必须中断总开关 EA 也被使能，否则不响应中断。

## 8.3 中断向量

当一个中断产生时，程序计数器内容被压栈，相应的中断向量地址被载入程序计数器。中断向量的地址在中断汇总表中详细列出。

## 8.4 中断优先级

每个中断源都可被单独设置为4个中断优先级之一，分别通过IP0，IP1，IP2，IP3，IP4中相应位来实现。中断优先级服务程序描述如下：

响应一个中断服务程序时，可响应更高优先级的中断，但不能响应同优先级或低优先级的另一个中断。

响应最高级中断服务程序时，不响应其它任何中断。如果不同中断优先级的中断源同时申请中断时，响应较高优先级的中断申请。

如果同优先级的中断源在指令周期开始时同时申请中断，那么内部查询优先级确定中断请求响应顺序。查询优先级详细参照中断汇总。

中断优先级	
优先级控制位(x 为功能模块)	优先级
Px[1:0]	
00	优先级 0 (最低)
01	优先级 1
10	优先级 2
11	优先级 3 (最高)

## 8.5 中断处理

中断标志在CPU时钟的上升沿被采样，如果一个标志被置起，那么CPU捕获到后中断系统调用一个长转移指令（LCALL）调用其中断服务程序，但由硬件产生的LCALL会被下列任何条件阻止：

1. 同级或更高级的优先级中断在运行中。
2. 当前的周期不是执行中指令的最后一个周期。换言之，正在执行的指令完成前，任何中断请求都得不到响应。
3. 正在执行的是一条 RETI 或者访问专用寄存器 IE/IE1 或是 IP0/IP1/IP2/IP3/IP4 的指令。换言之，在 RETI 或者读写 IE/IE1 或是 IP0/IP1/IP2/IP3/IP4 之后，不会马上响应中断请求，而至少在执行一条其它指令之后才会响应。

中断服务程序ISR完成和该中断相应的一些操作。ISR以RETI（中断返回）指令结束，将PC值从栈中取回，并恢复原来的中断设置，之后从主程序的断点处继续执行。

当某中断被响应时，被装载到程序计数器PC中的数值称为中断向量，是该中断源相对应的中断服务程序的起始地址。各中断源服务程序的入口地址（即中断向量）明细可参照中断汇总。

由于中断向量入口地址位于程序存储器的开始部分，所以主程序的第1条指令通常为跳转指令，越过中断向量区（LJMP MAIN）。

需要注意的是，不能用RET指令代替RETI指令，RET指令虽然也能控制PC返回到原来中断的地方，但RET指令没有清零中断优先级状态触发器的功能，中断控制系统会认为中断仍在进行，其后果是与此同级或低级的中断请求将不被响应。

若用户在中断服务程序中进行了入栈操作，则在RETI指令执行前应进行相应的出栈操作，即在中断服务程序中PUSH指令与POP指令必须成对使用，否则不能正确返回。

## 8.6 中断响应时间

每一个中断的响应时间都不同，这取决于中断自身特点和发生中断时正在执行的指令。如果检测出一个中断，这个中断的请求标志位就会被置起，内部电路会保持这个标志位，CPU会在第二个时钟周期产生中断。如果响应有效且条件允许，在下一个指令执行的时候硬件LCALL指令将调用请求中断的服务程序，否则中断被挂起。执行LCALL指令需要3个时钟周期。因此，从中断标志置位到开始执行中断服务程序至少需要5个时钟周期。

当中断请求因前述的三个情况受阻时，中断响应时间就会加长。如果同级或更高优先级的中断正在执行，额外的等待时间取决于正执行的中断服务程序的长度。

如果正在执行的指令还没有进行到最后一个周期，假如正在执行RETI指令，则完成正在执行的RETI指令，需要4个时钟周期，加上为完成下一条指令所需的最长时间4个时钟周期，若系统中只有一个中断源，再加上LCALL调用指令3个时钟周期，则最长的响应时间是13个时钟周期。

因此一个简单的中断系统响应时间，总是大于5个时钟周期并且不超过13个时钟周期。

## 8.7 外部中断

HC89F0431A/0421A/0411A有4个外部中断向量入口，外部中断0~1分别有一个独立的中断向量入口，外部中断2~7共用一个中断向量入口，外部中断8~15共用一个中断向量入口，因而共有16个外部中断端口输入，所有的中断都可以设定4种触发方式，分别为上升沿、下降沿、双沿及低电平。

当调用中断服务程序后，外部中断0~15标志必须软件清除。如果中断服务完成后而外部中断仍旧维持，则会产生下一次中断。

外部中断0~2功能端口分别在端口P0.0~P0.2上，在使用外部中断0~2时用户可设置外部中断0~2产生有效中断请求所需要的消抖时间，设置端口P0.0~P0.2的消抖时间即可，具体设置详见端口消抖控制寄存器 P00DBC、P01DBC、P02DBC。

## 8.8 中断相关寄存器

### 8.8.1 中断允许寄存器 IE、IE1

IE

位编号	7	6	5	4	3	2	1	0
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
位符号	EA	ES2	EWDT	ES1	ET1	EX1	ET0	EX0

位编号	位符号	说明
7	EA	CPU 总中断允许控制位 0: 禁止 CPU 中断 1: 允许 CPU 中断
6	ES2	UART2 中断允许位 0: 禁止 UART2 中断 1: 允许 UART2 中断
5	EWDT	WDT 中断允许位 0: 禁止 WDT 中断 1: 允许 WDT 中断
4	ES1	UART1 中断允许位 0: 禁止 UART1 中断 1: 允许 UART1 中断
3	ET1	T1 中断允许位 0: 禁止 T1 中断 1: 允许 T1 中断
2	EX1	外部中断 1 中断允许位 0: 禁止 INT1 中断 1: 允许 INT1 中断
1	ET0	T0 中断允许位 0: 禁止 T0 中断 1: 允许 T0 中断
0	EX0	外部中断 0 中断允许位 0: 禁止 INTO 中断 1: 允许 INTO 中断

**IE1**

位编号	7	6	5	4	3	2	1	0
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
位符号	EX8_15	EX2_7	EADC	ET5	EIIC	ET4	ET3	ESPI

位编号	位符号	说明
7	EX8_15	外部中断 8~15 中断允许位 0: 禁止 INT8~INT15 中断 1: 允许 INT8~INT15 中断 注: INT8~INT15 共用同一中断向量。
6	EX2_7	外部中断 2~7 中断允许位 0: 禁止 INT2~INT7 中断 1: 允许 INT2~INT7 中断 注: INT2~INT7 共用同一中断向量。
5	EADC	ADC 转换完成中断允许位 0: 禁止 ADC 中断 1: 允许 ADC 中断
4	ET5	T5 中断允许位 0: 禁止 T5 中断 1: 允许 T5 中断
3	EIIC	IIC 中断允许位 0: 禁止 IIC 中断 1: 允许 IIC 中断
2	ET4	T4 中断允许位 0: 禁止 T4 中断 1: 允许 T4 中断
1	ET3	T3 中断允许位 0: 禁止 T3 中断 1: 允许 T3 中断
0	ESPI	SPI 中断允许位 0: 禁止 SPI 中断 1: 允许 SPI 中断

## 8.8.2 中断优先级选择寄存器 IP0、IP1、IP2、IP3、IP4

### IP0

位编号	7	6	5	4	3	2	1	0
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
位符号	PT1[1:0]		PX1[1:0]		PT0[1:0]		PX0[1:0]	

位编号	位符号	说明
7-6	PT1[1:0]	T1 中断优先级控制位
5-4	PX1[1:0]	INT1 中断优先级控制位
3-2	PT0[1:0]	T0 中断优先级控制位
1-0	PX0[1:0]	INT0 中断优先级控制位

### IP1

位编号	7	6	5	4	3	2	1	0
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
位符号	PS2[1:0]		PLVD[1:0]		PWDT[1:0]		PS1[1:0]	

位编号	位符号	说明
7-6	PS2[1:0]	UART2 中断优先级控制位
5-4	PLVD[1:0]	LVD 中断优先级控制位
3-2	PWDT[1:0]	WDT 中断优先级控制位
1-0	PS1[1:0]	UART1 中断优先级控制位

### IP2

位编号	7	6	5	4	3	2	1	0
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
位符号	PPWM[1:0]		PT4[1:0]		PT3[1:0]		PSPI[1:0]	

位编号	位符号	说明
7-6	PPWM [1:0]	PWM 中断优先级控制位
5-4	PT4[1:0]	T4 中断优先级控制位
3-2	PT3[1:0]	T3 中断优先级控制位
1-0	PSPI [1:0]	SPI 中断优先级控制位

**IP3**

位编号	7	6	5	4	3	2	1	0
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
位符号	PX8_15 [1:0]		PX2_7[1:0]		PADC[1:0]		PT5[1:0]	

位编号	位符号	说明
7-6	PX8_15 [1:0]	INT8_15 中断优先级控制位
5-4	PX2_7[1:0]	INT2_7 中断优先级控制位
3-2	PADC[1:0]	ADC 中断优先级控制位
1-0	PT5[1:0]	T5 中断优先级控制位

**IP4**

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-						PIIC [1:0]	

位编号	位符号	说明
7-2	-	保留位
1-0	PIIC [1:0]	IIC 中断优先级控制位

中断优先级	
优先级控制位(x 为功能模块)	优先级
Px[1:0]	
00	优先级 0(最低)
01	优先级 1
10	优先级 2
11	优先级 3(最高)



### 8.8.3 外部中断电平选择寄存器 PITSx (x=0~3)

#### PITS0

位编号	7	6	5	4	3	2	1	0
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
位符号	IT3[1:0]		IT2[1:0]		IT1[1:0]		IT0[1:0]	

#### PITS1

位编号	7	6	5	4	3	2	1	0
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
位符号	IT7[1:0]		IT6[1:0]		IT5[1:0]		IT4[1:0]	

#### PITS2

位编号	7	6	5	4	3	2	1	0
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
位符号	IT11[1:0]		IT10[1:0]		IT9[1:0]		IT8[1:0]	

#### PITS3

位编号	7	6	5	4	3	2	1	0
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
位符号	IT15[1:0]		IT14[1:0]		IT13[1:0]		IT12[1:0]	

位编号	位符号	说明
7-6 5-4 3-2 1-0	ITx[1:0] (x = 0...15)	外部中断触发沿选择位 00: 低电平中断或下降沿中断 01: 下降沿中断 10: 上升沿中断 11: 双沿中断

## 8.8.4 外部中断 2-15 使能控制寄存器 PINTE<sub>x</sub> (x=0~1)

### PINTE<sub>0</sub>

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0
位符号	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	-	

位编号	位符号	说明
7-2	EINT <sub>x</sub> (x=2...7)	外部中断控制位(INT2~INT7) 0: 禁止该端口中断 1: 允许该端口中断 注: 只要相应的 EINT <sub>x</sub> (x=2...7)被允许, 对应的中断标志就可能被置 1, 反之, 对应标志不会被置 1。
1-0	-	保留位 (读为 0, 写无效)

### PINTE<sub>1</sub>

位编号	7	6	5	4	3	2	1	0
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
位符号	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8

位编号	位符号	说明
7-0	EINT <sub>x</sub> (x=8...15)	外部中断控制位(INT8~INT15) 0: 禁止该端口中断 1: 允许该端口中断 注: 只要相应的 EINT <sub>x</sub> (x=8...15)被允许, 对应的中断标志就可能被置 1, 反之, 对应标志不会被置 1。

## 8.8.5 外部中断标志寄存器 PINTFx (x=0~1)

### PINTF0

位编号	7	6	5	4	3	2	1	0
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
位符号	INT7F	INT6F	INT5F	INT4F	INT3F	INT2F	INT1F	INT0F

位编号	位符号	说明
7-2	INTxF (x =2...7)	INT2-INT7 中断请求标志位 0: 软件清 0 1: 符合外部中断时, 硬件置 1
1-0	INTxF (x = 0,1)	INT0 和 INT1 中断请求标志位 0: 中断响应时硬件自动清 0, 或软件清 0 1: 符合外部中断时, 硬件置 1

### PINTF1

位编号	7	6	5	4	3	2	1	0
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
位符号	INT15F	INT14F	INT13F	INT12F	INT11F	INT10F	INT9F	INT8F

位编号	位符号	说明
7-0	INTxF (x =8...15)	INT8-INT15 中断请求标志位 0: 软件清 0 1: 符合外部中断时, 硬件置 1

## 8.8.6 外部中断 01 管脚选择寄存器 INT01\_PINS

### INT01\_PINS

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-						INT1_PINS	INT0_PINS

位编号	位符号	说明
7-2	-	保留
1	INT1_PINS	INT1 管脚选择位 0: P0.1 1: P1.1
0	INT0_PINS	INT0 管脚选择位 0: P0.0 1: P1.0

# 9 定时器/计数器

## 9.1 定时器/计数器特性

- 定时器/计数器 T0&T1 是不完全兼容标准 8051，差异主要是在方式 0 的功能定义不同
- 定时器/计数器 T0&T1 支持 16 位自动重载

## 9.2 定时器/计数器 Tx(x = 0,1)

### 9.2.1 定时器/计数器 Tx(x = 0,1)的工作方式

每个定时器的两个数据寄存器 (THx & TLx (x=0,1)) 可作为一个 16 位寄存器来访问，它们由寄存器 TCON 和 TMOD 控制。IE0 寄存器的 ET0 和 ET1 位置 1 能允许定时器 0 和定时器 1 中断。(详见 [中断](#) 章节)。

通过计数器/定时器方式寄存器 (TMOD) 的方式选择位 Mx[1:0]，选择定时器工作方式。

Mx[1:0]	工作方式	描述
00	方式0	16位自动重载定时器/计数器
01	方式1	16位定时器/计数器
10	方式2	8位自动重载定时器/计数器
11	方式3	T0分成两个 (TL0/TH0) 独立的8位定时器/计数器 (T1无此模式)

#### 9.2.1.1 方式0: 16位自动重载定时器/计数器

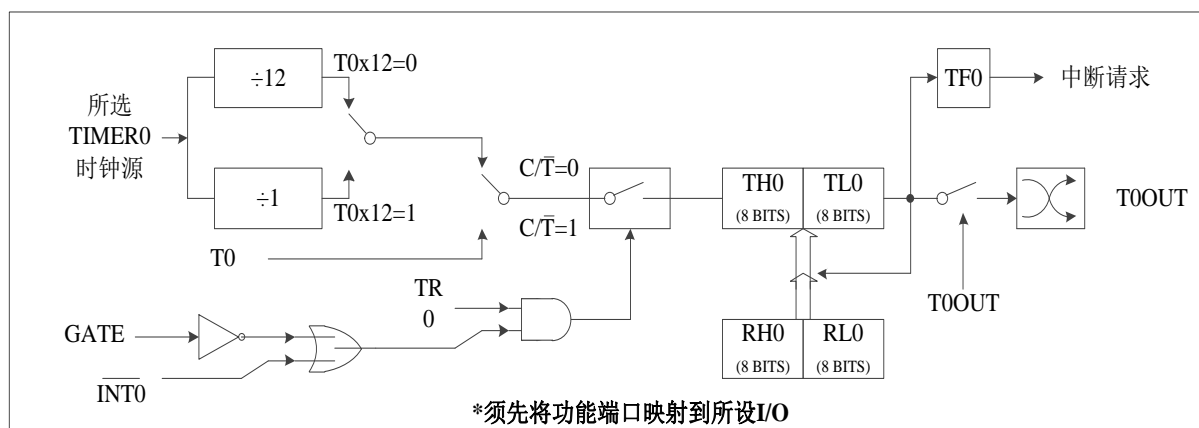


Figure 9-1 TIMER0 方式 0 功能框图

方式 0 与标准 8051 功能不兼容，在此方式下为 16 位自动重载定时器/计数器，当 THx 和 TLx(x = 0,1)被写时，用作定时器重载寄存器，当被读时，被用作计数寄存器。TRx(x = 0,1)为 0 时，按顺序写 THx 和 TLx(x = 0,1)两个寄存器，写的值同时被写进重载寄存器和计数寄存器，TRx(x = 0,1)置 1，计数寄存器开始从写的值递增计数，在计数到 0xFFFF 后，再来一个计数时钟，计数器就会发生溢出，此时 TFx(x = 0,1)被置为 1，同时重载寄存器的 16 位数据被自动重载入计数寄存器中，计数器又开始从这个重载的数值递增计数。

在 TRx(x = 0,1)为 1 时，对 THx & TLx(x = 0,1)的写操作，不会影响计数器的值，只能改变重载寄存

器的值，这个改变后的值在下次溢出时重载进计数寄存器中。只有  $TR_x(x = 0,1)$  为 0 时，对  $TH_x$  和  $TL_x(x = 0,1)$  的写操作，同时会改变计数寄存器和重载寄存器的值。

由于对  $TL_x(x = 0,1)$  和  $TH_x(x = 0,1)$  的写操作需要 2 条指令才能完成，为保证精确计数，对  $TH_x(x = 0,1)$  和  $TL_x(x = 0,1)$  寄存器的写操作都以对  $TL_x(x = 0,1)$  寄存器的写操作作为基准。当写入重载寄存器时，写  $TH_x(x = 0,1)$  寄存器不会立即生效，而是暂存在一个缓冲寄存器中，只有对  $TL_x(x = 0,1)$  寄存器的写操作才会使  $TH_x(x = 0,1)$  和  $TL_x(x = 0,1)$  寄存器同时生效。

因此， $TH_x(x = 0,1)$  和  $TL_x(x = 0,1)$  读写操作遵循以下顺序：

写操作：先高位后低位

读操作：先高位后低位

需要注意的是写操作时，当  $TR_x(x = 0,1)$  为 0，先写高位再写低位，重载数据将直接重载到计数寄存器中，当  $TR_x(x = 0,1)$  为 1，先写高位再写低位，重载数据只会在下次溢出时才会被重载到计数寄存器中。若先写低位再写高位，高位数据将无效（无效：表示发生重载时对应数据不能被更新），直到下次操作写入低位数据，前一次写高位的数据才会有效（有效表示发生重载时对应数据可以被更新）。若只写低位时，低位数据也将有效，例如对 T0 依次做如下操作时：

- (1)  $TH_0 = 0x05$ ;
- (2)  $TL_0 = 0x08$ ; //此时若发生重载，重载到计数器中的数据为  $0x0508$
- (3)  $TH_0 = 0x06$ ; //此时若发生重载，重载到计数器中的数据仍为  $0x0508$
- (4)  $TL_0 = 0x08$ ; //此时若发生重载，重载到计数器中的数据为  $0x0608$
- (5)  $TL_0 = 0x09$ ; //此时若发生重载，重载到计数器中的数据为  $0x0609$

显然只要修改重载数据，低位都必须再写入一次，建议每次修改同时修改。

注：方式 1、2、3 时无此要求。

### 9.2.1.2 方式1: 16位定时器/计数器

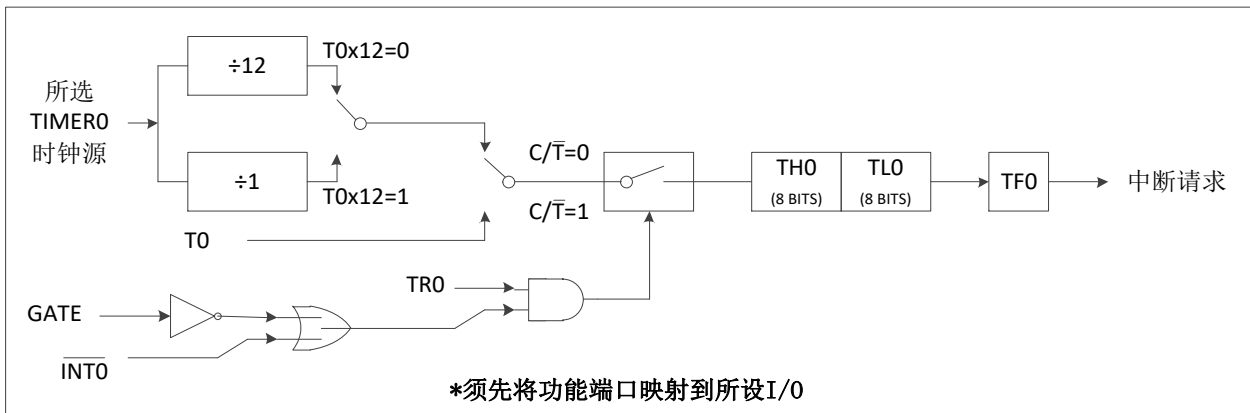


Figure 9-2 TIMER0 方式 1 功能框图

在方式 1 中，定时器  $T_x(x = 0,1)$  为 16 位计数器/定时器。 $TH_x(x = 0,1)$  寄存器存放 16 位计数器/定时器的高 8 位， $TL_x(x = 0,1)$  存放低 8 位。当 16 位定时器寄存器递增溢出时，系统置起定时器溢出标志  $TF_x(x = 0,1)$ 。如果定时器  $x$  中断被允许，将会产生一个中断。

$C/\bar{T}_x(x = 0,1)$  位选择计数器/定时器的功能，如果  $C/\bar{T}_x(x = 0,1) = 1$ ，将工作在外部计数模式，当出现定时器  $T_x(x = 0,1)$  外部计数时钟的下降沿时，将使定时器  $T_x$  数据寄存器加 1。如果  $C/\bar{T}_x(x = 0,1) = 0$ ，选择系统时钟为定时器  $T_x(x = 0,1)$  的时钟源。

当  $GATE_x(x = 0,1) = 0$  时， $TR_x$  置 1 则打开定时器。

当  $GATE_x(x = 0,1) = 1$  时，只有在外部输入信号  $INT_x(x = 0,1)$  为高电平时  $TR_x(x = 0,1)$  才会被置 1，定时器  $T_x$  才会计数，从而可测量  $INT_x(x = 0,1)$  的正脉冲宽度。 $TR_x(x = 0,1)$  位置 1 不强行复位定时器，这意味着如果  $TR_x$  置 1，定时器寄存器将从上次  $TR_x(x = 0,1)$  清 0 时的值开始计数。所以在允许定时器之前，应该设定定时器寄存器的初始值。

### 9.2.1.3 方式2: 8位自动重载定时器/计数器

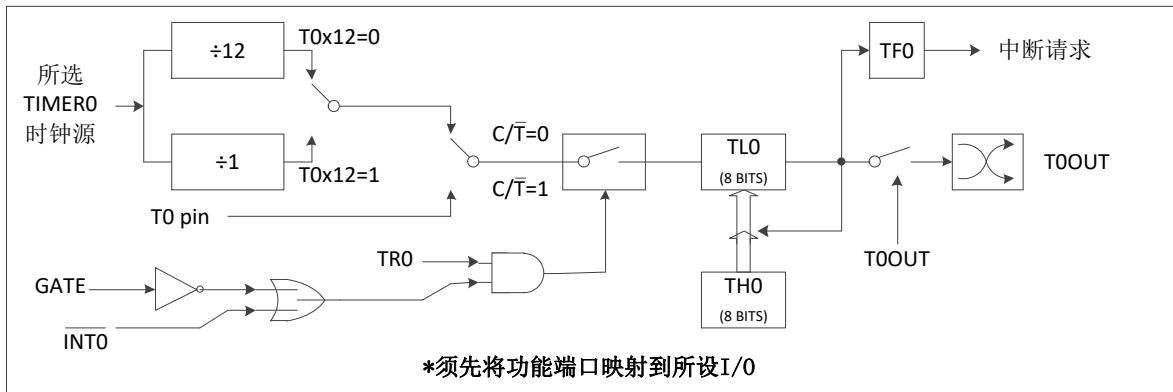


Figure 9-3 TIMER0 方式 2 功能框图

方式2中, 定时器Tx(x = 0,1)是8位自动重载计数器/定时器。TLx(x = 0,1)存放计数值, THx(x = 0,1)存放重载值。当在TLx(x = 0,1)中的计数器溢出至0x00时, 置起定时器溢出标志TFx(x = 0,1), 寄存器THx(x = 0,1)的值被重载入寄存器TLx(x = 0,1)中。如果定时器中断使能, 当TFx(x = 0,1)置1时将产生一个中断。而在THx(x = 0,1)中的重载值不会改变。在允许定时器正确计数开始之前, TLx(x = 0,1)必须初始化为所需的值。

除了自动重载功能外, 方式2中的计数器/定时器的使能和配置与方式1和0是一致的。可配置寄存器 TCON2 中的 TxX12(x = 0,1)位选择系统时钟或系统时钟的1/12作为定时器Tx(x = 0,1)的时钟源。

当作为定时器应用时, 可配置寄存器 TCON1 中的 TxOUT[1:0](x = 0,1)位使定时器 Tx(x = 0,1)溢出时 Tx(x = 0,1)脚自动翻转。

### 9.2.1.4 方式3: 两8位定时器/计数器(T1无此方式)

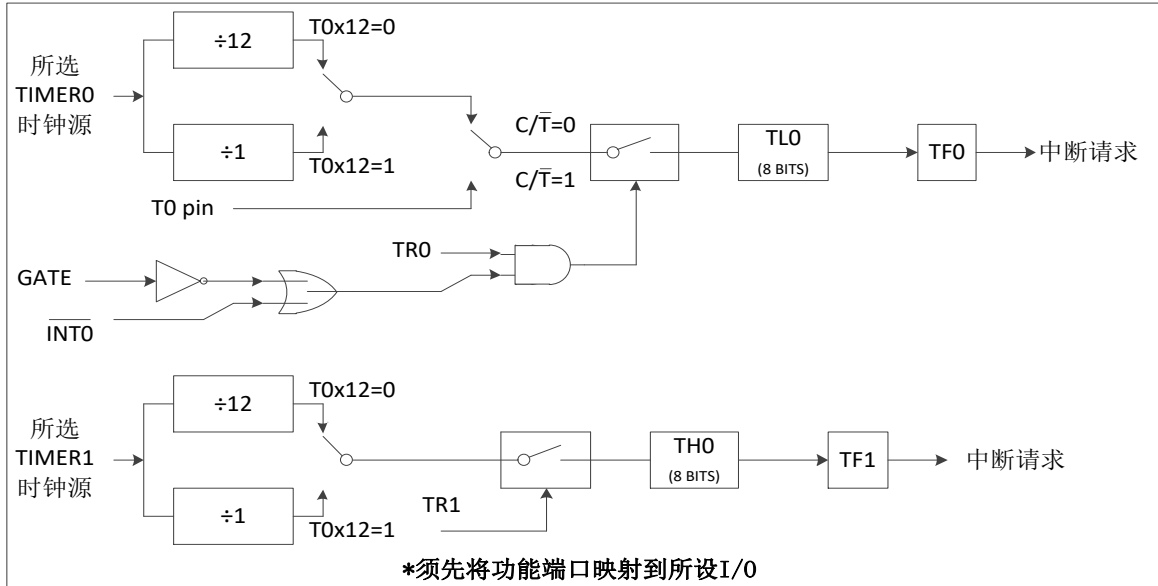


Figure 9-4 TIMER0 方式 3 功能框图

在方式3中, 定时器T0用作两个独立的8位计数器/定时器, 分别由TLO和THO控制。TLO使用定时器0的控制(在TCON中)和状态(在TMOD中)位: TR0, C/T-bar, GATE0和TF0。TLO能用系统时钟或外部输入信号作为时钟源。

THO只能用作定时器功能, 时钟源来自系统时钟。THO由定时器T1的控制位TR1控制使能, 溢出时定时器T1溢出标志TF1置1, 控制定时器T1中断。

定时器0工作在方式3时, 定时器1可以工作在方式0、1或2, 但是不能置TF1标志和产生中断。TH1

和TL1只能用作定时器功能，时钟源来自系统时钟，GATE1位无效。T1输入脚的上拉电阻也无效。定时器1由方式控制使能与否，因为TR1被定时器0占用。定时器1在方式0、1或2时使能，在方式3时被关闭。

可配置寄存器TCON1中的TxX12(x = 0,1)位选择系统时钟或系统时钟的1/12作为定时器Tx(x = 0,1)的时钟源。

## 9.2.2 定时器/计数器 Tx(x = 0,1)相关寄存器

### 9.2.2.1 定时器Tx(x = 0,1)控制寄存器 TCON、TCON1

#### TCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	TF1	TR1	TF0	TR0	-			

位编号	位符号	说明
7,5	TFx (x = 0,1)	Tx(x = 0,1)的溢出标志位 0: 中断响应时硬件自动清0，或软件清0 1: 计数溢出时，硬件置1
6,4	TRx (x = 0,1)	Tx(x = 0,1)运行控制位 0: 停止 Tx 工作 1: 启动 Tx 工作
3-0	-	保留位

#### TCON1

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	T1OUT	T1X12	-	-	T0OUT	T0X12

位编号	位符号	说明
7,6,3,2	-	保留位（读为0，写无效）
5,1	TxOUT (x = 0,1)	Tx(x = 0,1)比较输出功能允许位 0: 禁止定时器 Tx 比较输出功能 1: 允许定时器 Tx 比较输出功能
4,0	TxX12 (x = 0,1)	Tx(x = 0,1)定时器系统时钟源分频选择位 0: Tx 定时器时钟为 $F_{osc}/12$ 1: Tx 定时器时钟为 $F_{osc}$



### 9.2.2.2 定时器Tx(x = 0,1)工作方式寄存器 TMOD

位编号	7	6	5	4	3	2	1	0
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
位符号	GATE1	$\overline{C/T1}$	M1[1:0]		GATE0	$\overline{C/T0}$	M0[1:0]	

位编号	位符号	说明
7,3	GATE <sub>x</sub> (x = 0,1)	T <sub>x</sub> (x = 0,1)门控位 0: 只需软件置 TR <sub>x</sub> 即可启动 T <sub>x</sub> 1: 只有在 INT <sub>x</sub> 端口电平为高电平时 TR <sub>x</sub> 置 1, T <sub>x</sub> 才工作
6,2	$\overline{C/T_x}$ (x = 0,1)	T <sub>x</sub> (x = 0,1)定时/计数功能选择位 0: T <sub>x</sub> 用于内部定时 1: T <sub>x</sub> 用于外部计数
5-4 1-0	M <sub>x</sub> [1:0] (x = 0,1)	T <sub>x</sub> (x = 0,1)工作方式选择位 00: 方式 0 16 位自动重载定时器/计数器 01: 方式 1 16 位定时器/计数器 10: 方式 2 8 位自动重载定时器/计数器 11: 方式 3 T <sub>0</sub> 分成两个(TL <sub>0</sub> /TH <sub>0</sub> )独立的 8 位定时器/计数器; T <sub>1</sub> 停止计数 注: 方式 3 时 TH <sub>0</sub> 占用 T <sub>1</sub> 的 TR <sub>1</sub> 、TF <sub>1</sub> 及中断源, 由于 TR <sub>1</sub> 被 TH <sub>0</sub> 占用, 此时需要关闭 T <sub>1</sub> 可将 T <sub>1</sub> 设为工作方式 3。

### 9.2.2.3 定时器Tx(x = 0,1)数据寄存器 TLx(x = 0,1)、THx(x = 0,1)

#### TLx (x = 0,1)

位编号	7	6	5	4	3	2	1	0
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
位符号	TL <sub>x</sub> [7:0] (x = 0,1)							

位编号	位符号	说明
7-0	TL <sub>x</sub> [7:0] (x = 0,1)	T <sub>x</sub> (x = 0,1)数据寄存器低字节

#### THx (x = 0,1)

位编号	7	6	5	4	3	2	1	0
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
位符号	TH <sub>x</sub> [7:0] (x = 0,1)							

位编号	位符号	说明
7-0	TH <sub>x</sub> [7:0] (x = 0,1)	T <sub>x</sub> (x = 0,1)数据寄存器高字节

### 9.3 定时器 3

定时器 3 是 16 位自动重载定时器，通过两个数据寄存器 TH3 和 TL3 访问，由 T3CON 寄存器控制。IE1 寄存器的 ET3 位置 1 允许定时器 3 中断。

定时器 3 只有一个工作方式：16 位自动重载计数器/定时器，可以设置预分频比，并可以工作在掉电模式。

定时器 3 有一个 16 位计数器/定时器寄存器 (TH3, TL3)。当 TH3 和 TL3 被写时，用作定时器重载寄存器，当被读时，被用做计数寄存器。TR3 位置 1 使定时器 3 开始递增计数，在 0xFFFF 到 0x0000 时发生溢出，溢出会置 TF3 位为 1，同时将重载寄存器的 16 位数据重新载入计数寄存器中。

在 TR3 为 1 时，对 TH3/TL3 的写操作，不会影响计数器的值，只会改变重载寄存器的值，这个改变后的值在下次溢出时会被重载进计数寄存器。只有 TR3 为 0 时，对 TH3/TL3 的写操作会同时会改变计数寄存器和重载寄存器的值。

TH3 和 TL3 读写操作遵循以下顺序：先高位后低位。

如果 T3CLKS[1:0] 为 00，定时器 3 不能工作在掉电模式下。如果 T3CLKS[1:0] 为 01，T3 端口输入外部时钟，定时器 3 可以工作在普通模式或掉电模式，当 T3CLKS[1:0] 为 10 或 11，即定时器 3 的计数时钟源为 32.768KHz 低频晶振或 RC44K 时，定时器 3 也可以工作在普通模式或掉电模式。当需要定时器 3 工作在掉电模式时软件要配置 T3PD\_EN 为 1。

当 T3PD\_EN 为 1，而且 T3CLKS[1:0] 为 10 时，低频晶振在掉电模式下不会关闭，因此定时器 3 可以在掉电模式继续工作。当定时溢出时，会将芯片从掉电模式唤醒，如果中断允许的话，唤醒后的芯片会进入定时器 3 中断。

注意：在读 TH3 和 TL3 时，要确保 TR3 = 0（当 TR=1 时，由于正在计数，读出的 TH3 和 TL3 不准）。

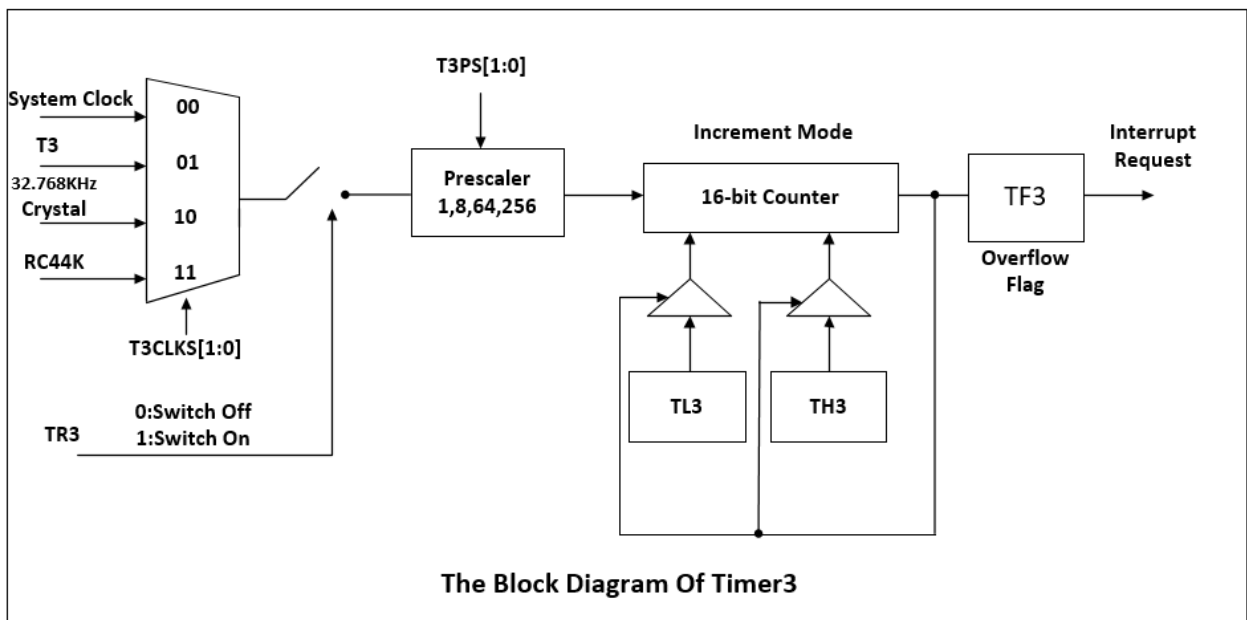


Figure 9-5 TIMER3 功能框图

### 9.3.1 定时器/计数器 T3 相关寄存器

#### 9.3.1.1 定时器T3控制寄存器 T3CON

##### T3CON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	TF3	T3PD_EN	T3PS[1:0]		-	TR3	T3CLKS[1:0]	

位编号	位符号	说明
7	TF3	定时器 3 溢出标志位 0: 无溢出 (硬件清 0) 软件也可以清 0 1: 溢出 (硬件置 1)
6	T3PD_EN	定时器 3 在掉电模式下运行控制位 0: 掉电模式下禁止定时器 3 工作 1: 掉电模式下允许定时器 3 工作, 此时 T3CLKS[1:0]为 01 或 10、11
5-4	T3PS[1:0]	定时器 3 预分频比选择位 00: 1/1 01: 1/8 10: 1/64 11: 1/256
3	-	保留位
2	TR3	定时器 3 允许控制位 0: 禁止定时器 3 1: 允许定时器 3
1-0	T3CLKS[1:0]	定时器 3 计数时钟来源选择位 00: 系统时钟 $F_{osc}$ 01: T3 端口输入外部时钟 10: 外部 32.768KHz 晶体振荡器 11: RC44K

#### 9.3.1.2 定时器T3数据寄存器 TL3、TH3

##### TL3

位编号	7	6	5	4	3	2	1	0
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
位符号	TL3							

位编号	位符号	说明
7-0	TL3	T3 数据寄存器低字节

**TH3**

位编号	7	6	5	4	3	2	1	0
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
位符号	TH3							

位编号	位符号	说明
7-0	TH3	T3 数据寄存器高字节

## 9.4 定时器/计数器 4

定时器 4 是 16 位自动重载定时器。两个数据寄存器 TH4 和 TL4 可作为一个 16 位寄存器来访问，由 T4CON 寄存器控制。IE1 寄存器的 ET4 位置 1 允许定时器 4 中断。

当 TH4 和 TL4 被写时，用作定时器重载寄存器，当被读时，被用作计数寄存器。TR4 位置 1 使定时器 4 开始递增计数。定时器在 0xFFFF 到 0x0000 溢出并置 TF4 位为 1。溢出同时，定时器重载寄存器的 16 位数据重新载入计数寄存器中。

TH4 和 TL4 读写操作遵循以下顺序：先高位后低位。

### 9.4.1 定时器/计数器 T4 的工作方式

定时器 4 有三种工作方式：16 位自动重载定时器，串口波特率发生器和有 T4 边沿触发的 16 位自动重载定时器。这些方式通过 T4CON 寄存器的 T4M[1:0] 设置。

#### 9.4.1.1 方式 0: 16 位自动重载定时器/计数器

定时器 4 在方式 0 为 16 位自动重载定时器。TH4 寄存器存放高 8 位，TL4 寄存器存放低 8 位。TR4 为 0 时，按顺序写 TH4 和 TL4 两个寄存器，写的值同时被写进重载寄存器和计数寄存器；TR4 置 1，计数寄存器开始从写的值递增计数，在计数到 0xFFFF 后，再来一个计数时钟，计数器就会发生溢出，此时 TF4 被置为 1。同时重载寄存器的 16 位数据被自动重载入计数寄存器中，计数器又开始从这个重载的数值递增计数，如果允许定时器 4 中断则产生中断。

在 TR4 为 1 时，对 TH4 和 TL4 的写操作，不会影响计数器的值，只能改变重载寄存器的值，这个改变后的值在下次溢出时重载进计数寄存器中。只有 TR4 为 0 时，对 TH4 和 TL4 的写操作会同时会改变计数寄存器和重载寄存器的值。

T4CON.0 寄存器的 T4CLKS 位选择时钟源。当 T4CLKS = 1 时，定时器 4 的时钟源为外部时钟，预分频后，计数器数据寄存器增加。当 T4CLKS = 0，定时器 4 的时钟源为系统时钟。

在比较方式中，需要软件将 T4 端口设为输出。定时器 4 从 TH4 和 TL4 预设值开始向 0xFFFF 计数，当计数器溢出时，T4 端口输出电平翻转，同时定时器 4 中断标志位被置 1。在比较方式中，定时器 4 必须工作在定时方式 (T4CLKS = 0)。

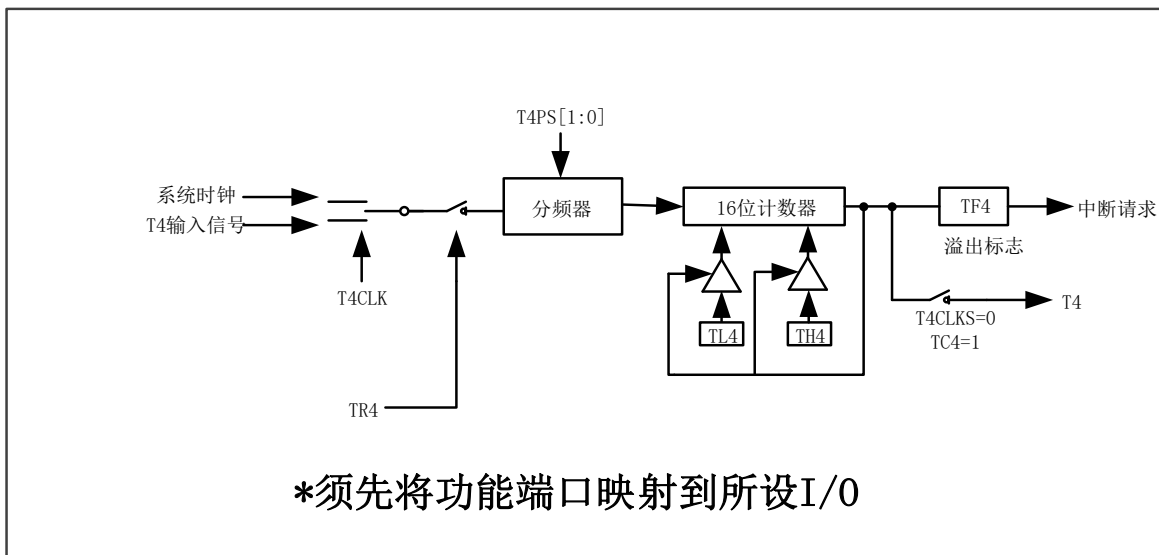


Figure 9-6 TIMER4 方式 0 功能框图

### 9.4.1.2 方式1: 波特率发生器

通过设置 T4CON 寄存器中的 T4M[1:0]为 01 选择定时器 4 作为波特率发生器。该方式与自动重载方式相似。定时器 4 的溢出会引起软件将定时器 4 重载寄存器中的 16 位值载入定时器 4 的计数器中，此时也会产生溢出中断。如果不希望中断发生，可以关闭 ET4。波特率由下列公式计算：

$$\text{BaudRate} = \frac{1}{16} \times \frac{f_{T4}/\text{PRESCALER}}{65536 - [\text{TH4}, \text{TL4}]}$$

用定时器 4 作为波特率发生器。

上式中， $f_{T4}$  为定时器 4 计数所选时钟源，PRESCALER 为定时器 4 预分频比，TH4 和 TL4 为定时器 4 数据寄存器。

如果 TC4=1 时，可以将波特率从 T4 口输出，输出波形的频率为波特率的 1/2，此时定时器 4 必须工作在定时方式（T4CLKS = 0）。

### 9.4.1.3 方式2/3:带边沿触发的16位自动重载定时器

定时器 4 在方式 2/3 为 16 位自动重载定时器。T4CON.0 寄存器的 T4CLKS 位一直为 0，定时器 4 只能选择系统时钟为时钟源，其余设置与方式 0 一致。

方式 2 中，当 TR4 位置 1 后，定时器 4 等待 T4 端口的触发信号（由 T4M[1:0]控制上升/下降沿），一个有效的触发信号使定时器 4 开始运行。当定时器 4 内部 16 位计数器从 0xFFFF 到 0x0000 溢出时，TF4（T4CON.7）会被置起，如定时器 4 的中断使能，将产生定时器 4 中断。溢出同时，定时器重载寄存器的 16 位数据被重新载入内部 16 位计数器 TH4 和 TL4 中，定时器 4 将保持状态并等待下一个触发沿。

如果 TC4 = 0，在定时器 4 计数时，一个触发信号不会中止内部 16 位计数器的计数，内部 16 位计数器溢出后进行重载并保持状态，等待下一个有效触发信号；

如果 TC4 = 1，在定时器 4 计数时，一个触发信号会使重载寄存器的 16 位数据被重新载入内部 16 位计数器 TH4 和 TL4 中，并开始计数，但不会产生中断，内部 16 位计数器溢出后才会产生中断。

TR4 置 1 不清定时器 4 的内部 16 位计数器，在允许定时器之前应该把希望的初始化值写入重载寄存器。

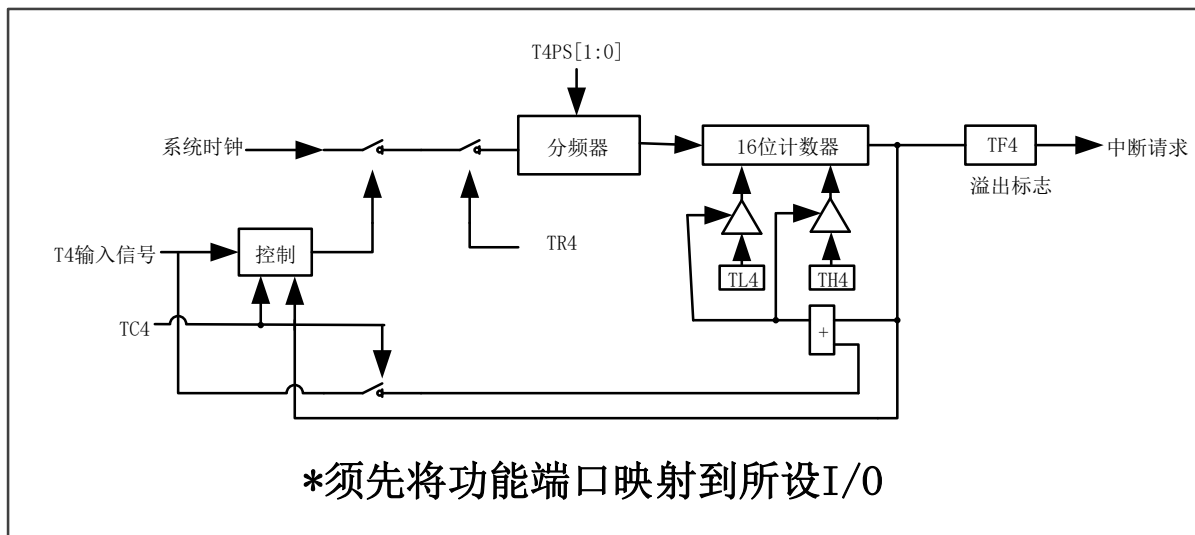


Figure 9-7 TIMER4 方式 2/3 功能框图

注意：

- (1) 当定时器 4 在波特率发生器方式下作为定时器工作时（TR4 = 1），TH4 或 TL4 不能读取或写入。因为定时器在每个状态时间递增，可能导致读取或写入的结果不精确。因此，在访问 TH4/TL4 寄存器之前，定时器 4 必须被关闭（TR4 = 0）。
- (2) 当定时器 4 用作计数器时，T4 引脚的输入信号频率要小于系统时钟的一半。

## 9.4.2 定时器/计数器 T4 相关寄存器

### 9.4.2.1 定时器T4控制寄存器 T4CON、T4CON1

#### T4CON

位编号	7	6	5	4	3	2	1	0
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
位符号	TF4	TC4	T4PS[1:0]		T4M[1:0]		TR4	T4CLKS

位编号	位符号	说明
7	TF4	定时器 4 溢出标志位 0: 无溢出 (硬件清 0) 软件也可以清 0 1: 溢出 (硬件置 1)
6	TC4	比较功能允许位 当 T4M[1:0] = 00 或 01 0: 禁止定时器 4 比较功能 1: 允许定时器 4 比较功能 当 T4M[1:0] = 10 或 11 0: 定时器 4 不能被再触发 1: 定时器 4 可以被再触发
5-4	T4PS[1:0]	定时器 4 预分频比选择位 00: 1/1 01: 1/8 10: 1/64 11: 1/256
3-2	T4M[1:0]	定时器 4 方式选择位 00: 方式 0, 16 位自动重载定时器 01: 方式 1, UART 波特率发生器 10: 方式 2, T4 端口上升沿触发 (只用系统时钟, T4CLKS 无效) 11: 方式 3, T4 端口下降沿触发 (只用系统时钟, T4CLKS 无效)
1	TR4	定时器 4 允许控制位 0: 禁止定时器 4 1: 允许定时器 4
0	T4CLKS	定时器 4 计数时钟来源选择位 0: 系统时钟 F <sub>osc</sub> 1: T4 端口输入外部时钟

### 9.4.2.2 定时器T4数据寄存器 TL4、TH4

#### TL4

位编号	7	6	5	4	3	2	1	0
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
位符号	TL4							

位编号	位符号	说明
7-0	TL4	T4 数据寄存器低字节

#### TH4

位编号	7	6	5	4	3	2	1	0
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
位符号	TH4							

位编号	位符号	说明
7-0	TH4	T4 数据寄存器高字节



## 9.5 定时器 5

定时器 5 是 16 位自动重载定时器。通过两个数据寄存器 TH5 和 TL5 访问，由 T5CON 寄存器控制。IE1 寄存器的 ET5 位置 1 允许定时器 5 中断。

### 9.5.1 定时器 T5 的工作方式

#### 9.5.1.1 方式0: 16位自动重载定时器/计数器

定时器 5 在方式 0 时，TH5 寄存器存放高 8 位，TL5 寄存器存放低 8 位。

当 EXEN5=0，16 位定时寄存器从 0xFFFF 到 0x0000 递增，并溢出时，置起 TF5 位，同时定时器自动将用户软件写好的寄存器 RCAP5H 和 RCAP5L 的 16 位值装入 TH5 和 TL5 寄存器，如果允许定时器 5 中断则产生中断。

如果 EXEN5=1，溢出或在外部输入 T5 都能触发一次 16 位重载，同时也会置起 EXF5 位。如果 ET5 被使能，TF5 和 EXF5 位都能产生中断。（T5 的沿可以引发重载中断）

T5CON.1 寄存器的 TR5 位置 1 使能定时器 5，且不清定时器 5 的计数器。在允许定时器 5 之前，将希望的初始值写入定时器重载寄存器中。

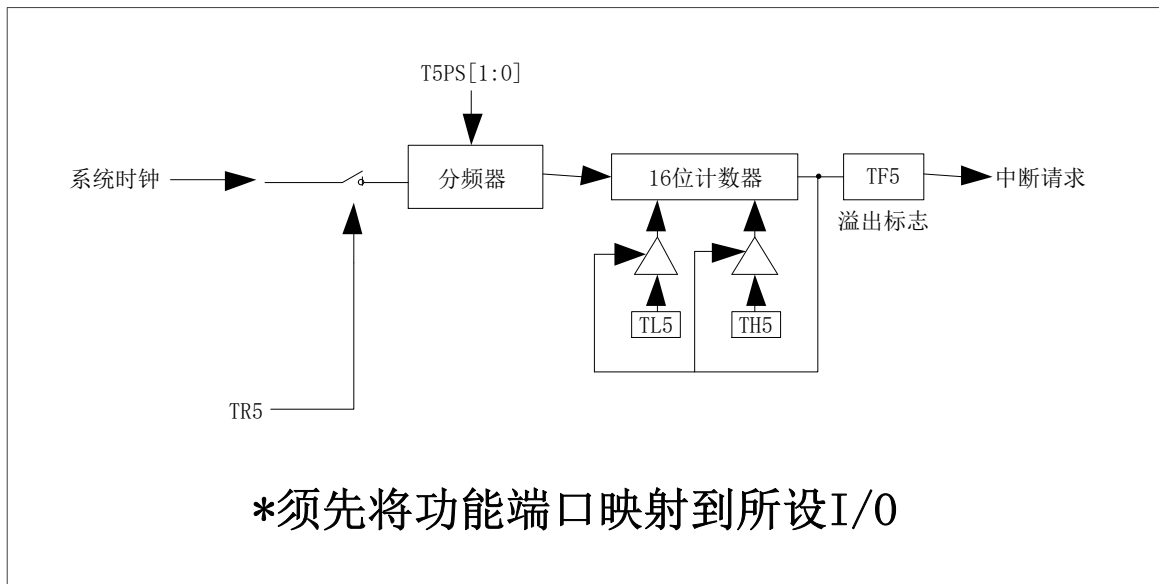


Figure 9-8 TIMER5 方式 0 功能框图

#### 9.5.1.2 方式1: 波特率发生器

通过设置 T5CON 寄存器中的 T5M 为 01，选择定时器 5 作为波特率发生器。该方式与自动重载方式相似。定时器 5 的溢出会将定时器 5 重载寄存器中的 16 位值载入定时器 5 的计数器中，但溢出不会产生中断。此时如果 EXEN5=1，则 T5 引脚上的一个下降沿/上升沿会置位 EXF5，但不会引起重载。因此当定时器 5 作为波特率发生器时，T5 引脚可作为一个额外的外部中断。

波特率由下列公式计算：

$$\text{BaudRate} = \frac{1}{16} \times \frac{f_{T5}/\text{PRESCALER}}{65536 - [\text{RCAP5H}, \text{RCAP5L}]}, \text{ 用定时器 5 作为波特率发生器。}$$

上式中， $f_{T5}$  为 Fosc，PRESCALER 为定时器 5 预分频比，RCAP5H 和 RCAP5L 为定时器 5 的重载捕获寄存器。

### 9.5.1.3 方式2: 16位上升沿捕获

在捕获方式中, T5CON 的 EXEN5 位有两个选项。

如果 EXEN5 = 0, 定时器 T5 作为 16 位定时器, 如果 ET5 被允许的话, 定时器 T5 能设置 TF5 溢出产生一个中断。溢出后计数不会被重载, 从 0 开始计数。

如果 EXEN5 = 1, 定时器 T5 执行相同操作, 但是在外部输入 T5 (可通过 T5CON1 来选择捕获输入的类型) 的上升沿能引起 TH5 和 TL5 的当前值分别被捕获到 RCAP5H 和 RCAP5L 中, 此外, 在 T5 上的上升沿也能引起在 T5CON 中的 EXF5 被设置。如果 ET5 被允许, EXF5 位也像 TF5 一样也产生一个中断。

### 9.5.1.4 方式3: 16位下降沿捕获

在捕获方式中, T5CON 的 EXEN5 位有两个选项。

如果 EXEN5 = 0, 定时器 T5 作为 16 位定时器, 如果 ET5 被允许的话, 定时器 T5 能设置 TF5 溢出产生一个中断。溢出后计数不会被重载, 从 0 开始计数。

如果 EXEN5 = 1, 定时器 T5 执行相同操作, 但是在外部输入 T5 (可通过 T5CON1 来选择捕获输入的类型) 的下降沿能引起 TH5 和 TL5 的当前值分别被捕获到 RCAP5H 和 RCAP5L 中, 此外, 在 T5 上的下降沿也能引起在 T5CON 中的 EXF5 被设置。如果 ET5 被允许, EXF5 位也像 TF5 一样也产生一个中断。

### 9.5.1.5 方式4: 16位双沿捕获

在捕获方式中, T5CON 的 EXEN5 位有两个选项。

如果 EXEN5 = 0, 定时器 T5 作为 16 位定时器, 如果 ET5 被允许的话, 定时器 T5 能设置 TF5 溢出产生一个中断。溢出后计数不会被重载, 从 0 开始计数。

如果 EXEN5 = 1, 定时器 T5 执行相同操作, 但是在外部输入 T5 (可通过 T5CON1 来选择捕获输入的类型) 的下降沿或上升沿能引起 TH5 和 TL5 的当前值分别被捕获到 RCAP5H 和 RCAP5L 中, 此外, 在 T5 上的下降沿或上升沿也能引起在 T5CON 中的 EXF5 被设置。如果 ET5 被允许, EXF5 位也像 TF5 一样也产生一个中断。

## 9.5.2 定时器/计数器 T5 相关寄存器

### 9.5.2.1 定时器T5控制寄存器 T5CON、T5CON1、T5CON2

#### T5CON

位编号	7	6	5	4	3	2	1	0
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
位符号	TF5	EXF5	T5PS[1:0]		T5M[1:0]		TR5	EXEN5

位编号	位符号	说明
7	TF5	定时器 5 溢出标志位 0: 无溢出 1: 溢出时硬件置 1, 必须软件清 0
6	EXF5	T5 引脚外部输入事件发生标志位 0: 无外部输入事件发生, 必须软件清 0 1: 检测到外部输入事件且 EXEN5=1 时, 硬件置 1, 也做中断请求标志位
5-4	T5PS[1:0]	定时器 5 预分频比 (PRESCALER) 选择位 00: 1/1 01: 1/8

		10: 1/64 11: 1/256
3-2	T5M[1:0]	定时器 5 方式选择位 00: Mode0, 16 位自动重载定时器 01: Mode1, 波特率发生器 10: Mode2, 16 位上升沿捕获 11: Mode3, 16 位下降沿捕获
1	TR5	定时器 5 允许控制位 0: 禁止定时器 5 1: 允许定时器 5
0	EXEN5	T5 引脚上的外部事件输入用作重载/捕获触发器允许/禁止控制位 0: 忽略 T5 引脚上的事件 1: 根据模式的不同, EXEN5 有不同的作用: 当定时器 5 用作 16 位自动重载定时器时, EXEN5=1 时, T5 引脚上一个下降沿或上升沿会产生一次重载。EXF5 置位, 如果中断使能, 会发生 T5 中断; 当定时器 5 用作波特率发生器时, EXEN5=1 时, T5 引脚上的一个下降沿或上升沿会置位 EXF5, 如果中断使能, 会发生 T5 中断, 可以做一个额外的外部中断; 当定时器 5 用作上升沿捕获时, T5 引脚上的一个上升沿会产生一次捕获, 同时会置位 EXF5, 如果中断使能, 会发生 T5 中断; 当定时器 5 用作下降沿捕获时, T5 引脚上的一个下降沿会产生一次捕获, 同时会置位 EXF5, 如果中断使能, 会发生 T5 中断; 当定时器 5 用作上升沿和下降沿捕获时, T5 引脚上的一个上升沿和下降沿都会产生一次捕获, 同时会置位 EXF5, 如果中断使能, 会发生 T5 中断; 注: 当捕获内部低频 RC 或 RXD 引脚时, 也需要使能 EXEN5, 同时还要配置上升沿捕获、下降沿捕获或双沿捕获。

### T5CON1

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-						CAPM[1:0]	

位编号	位符号	说明
7-2	-	保留位
1-0	CAPM[1:0]	定时器 5 捕获类型选择位 00: T5 脚的变化沿 01: 内部低频 RC, 即看门狗的计数时钟 10: UART1 的 RXD1 脚 11: UART2 的 RXD2 脚

**T5CON2**

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-					T5_MODE	-	T5CAPCR

位编号	位符号	说明
7-3	-	保留位（读为0，写无效）
2	T5_MODE	T5 的扩展模式选择 0: 由 T5M[1:0]决定 1: 上升沿或下降沿捕获
1	-	保留位（读为0，写无效）
0	T5CAPCR	T5 端口捕获自动清除 该位使能，当捕获事件发生，在 TH5 和 TL5 内的数据移入捕获寄存器后，硬件将自动清除 TH5 及 TL5 计数寄存器的值。注意：当该位置 1，如果有一个捕获事件发生，仅清除 TH5、TL5 的值。 0: T5 端口捕获事件发生后定时器 5 计数值按之前计数值继续累加 1: T5 端口捕获事件发生后定时器 5 计数值自动清 0

**9.5.2.2 定时器T5数据寄存器 TL5、TH5**
**TL5**

位编号	7	6	5	4	3	2	1	0
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
位符号	TL5							

位编号	位符号	说明
7-0	TL5	T5 数据寄存器低字节

**TH5**

位编号	7	6	5	4	3	2	1	0
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
位符号	TH5							

位编号	位符号	说明
7-0	TH5	T5 数据寄存器高字节

### 9.5.2.3 定时器T5重载捕获寄存器 RCAP5L、RCAP5H

#### RCAP5L

位编号	7	6	5	4	3	2	1	0
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
位符号	RCAP5L							

位编号	位符号	说明
7-0	RCAP5L	T5 重载捕获寄存器低字节

#### RCAP5H

位编号	7	6	5	4	3	2	1	0
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
位符号	RCAP5H							

位编号	位符号	说明
7-0	RCAP5H	T5 重载捕获寄存器高字节

# 10 脉宽调制PWM

## 10.1 PWM 特性

- 3 组带死区互补 PWM 或 6 路独立 PWM 输出
- 12 位 PWM
- 提供每个 PWM 周期溢出中断，但中断共用同一向量入口
- 输出极性可选择
- 提供出错帧测功能可紧急关闭 PWM 输出
- PWM 工作时钟源可设定时钟分频比
- PWM 可做定时器/计数器使用

HC89F0431A/0421A/0411A 集成了三个 12 位 PWM 模块 PWM0、PWM1 和 PWM2，三个模块各有一个计数器，PWM0 的计数器由 PWM0\_EN 来控制，只要使能 PWM0\_EN，计数器就会启动，计数器的时钟源通过 PWM0C 控制寄存器里的 CK0 来选择。

当需要从芯片管脚输出 PWM 波形时，还需要使能 PWM0\_OEN 或 PWM01\_OEN，同时需要将端口设置为强推挽模式。如果不希望从芯片管脚上输出 PWM 波形，可以不用使能 PWM0\_OEN 或 PWM01\_OEN，这时候 PWM0 的计数器可以当一个定时器来使用，当计数器溢出时，如果中断允许也会产生 PWM 中断。

如果 EFLT0 置 1，PWM0 输出和其互补输出可由 FLT0 引脚输入信号变化自动关闭。一旦检测到 FLT0 引脚输入有效电平，PWM 输出会立即关闭，但 PWM 内部计数器仍在继续运行，这样方便在 FLT0 引脚错误去除后继续 PWM 输出。在 FLT0 输入信号有效期间，FLT0S 位无法清除。只有当 FLT0 输入信号消失后，才能软件清除 FLT0S 状态位，此时 PWM 恢复正常输出。

PWM0 故障检测端口 FLT0（P0.0 端口）具有消抖功能，用户可根据需要设置合适的消抖时间。配置 FLT0 的消抖时间即配置端口 P0.0 的消抖时间，设置方法详见端口消抖控制寄存器 P00DBC、P01DBC、P02DBC 章节。

三个 PWM 模块的功能及操作完全一样，用户可通过控制寄存器产生 3 路带死区互补的 PWM 或 6 路单独的 PWM 输出。

3 个 PWM 模块共用一个中断向量入口，但有各自的控制位及标志位，方便用户定时修改 PWM 模块下一次循环的周期或占空比。

## 10.2 PWM 输出类型

HC89F0431A/0421A/0411A 的 PWM 输出包含两种类型：边沿对齐和中心对齐。通过设置 TYPx (PWMCON0[7:5]) (x=0,1,2)来决定。

### 10.2.1 边沿对齐模式

边沿对齐模式下，模块产生边沿对齐的 PWM 信号。PWM 输出信号的周期由[PWM0PH:PWM0PL]决定，其占空比由相应的占空比寄存器（PWM 独立输出时，PWM01、PWM11、PWM21 的占空比寄存器是其死区时间寄存器）。

12 位计数器采用单周期模式，从 0000H 开始向上计数，所有使能的 PWM 输出在 PWM 周期开始时被驱动为有效状态。当计数器的值与 PWM 占空比寄存器的值匹配时，PWM 输出被驱动为无效状态。计数器直到与[PWM0PH:PWM0PL]匹配，然后重新从 0000H 开始向上计数。

$$PWM_x \text{ 周期} = [PWM0PH : PWM0PL] * PWM0 \text{ 工作时钟源周期}$$

$$PWM_x \text{ 占空比} = [PWM_xDH : PWM_xDL] * PWM0 \text{ 工作时钟周期}$$

$$PWM_{x1} \text{ 占空比} = [PWM_xDTH : PWM_xDTL] * PWM0 \text{ 工作时钟周期}$$

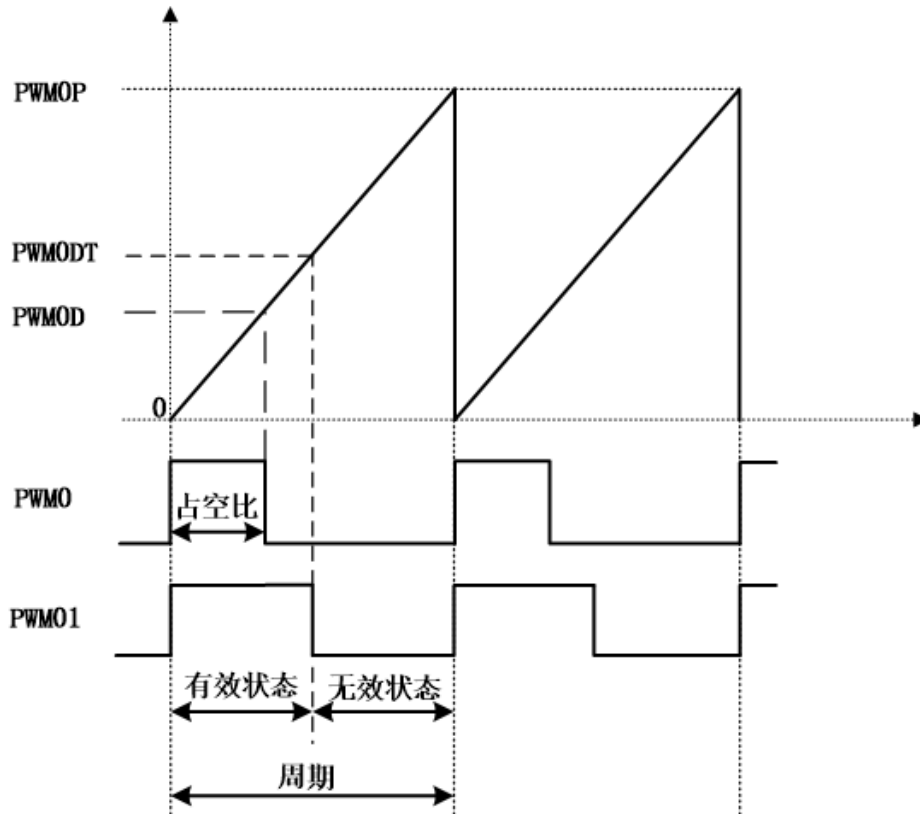


Figure 10-1 PWM 边沿对齐波形

### 10.2.2 中心对齐模式

中心对齐模式下，模块产生中心对齐的 PWM 信号。PWM 输出信号的周期由[PWM0PH:PWM0PL]决定，其占空比由相应的占空比寄存器（PWM 独立输出时，PWM01、PWM11、PWM21 的占空比寄存器是其死区时间寄存器）。

12 位计数器采用双周期模式，先从 0000H 开始向上计数至[PWM0PH:PWM0PL]，然后由[PWM0PH:PWM0PL]向下计数至 0000H，此为一个 PWM 的完整周期。所有使能的 PWM 输出在 PWM 周期开始时被驱动为有效状态。在向上计数的过程中，当计数器的值与 PWM 占空比寄存器的值匹配时，PWM 输出被驱动为无效状态，直到计数器转换为向下计数且计数器值与 PWM 占空比寄存器的值匹配时，PWM 输出才被驱动为有效状态。

$$PWM_x \text{ 周期} = [PWM0PH:PWM0PL] * PWM0 \text{ 工作时钟源周期} * 2$$

$$PWM_x \text{ 占空比} = [PWM_xDH:PWM_xDL] * PWM0 \text{ 工作时钟源周期} * 2$$

$$PWM_{x1} \text{ 占空比} = [PWM_xDTH:PWM_xDTL] * PWM0 \text{ 工作时钟源周期} * 2$$

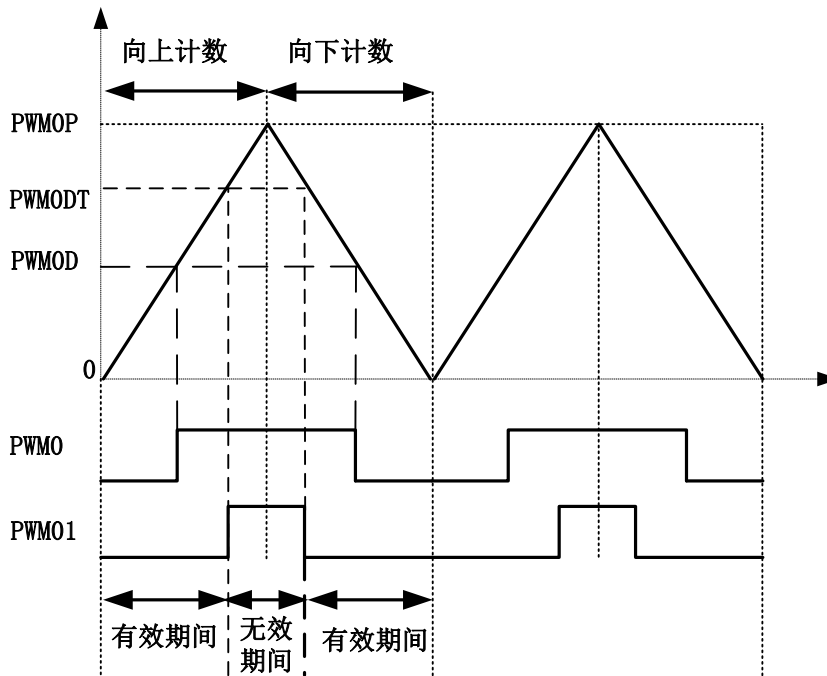


Figure 10-2 PWM 中心沿对齐波形



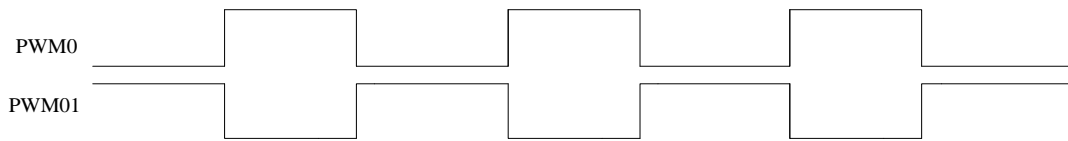
## 10.3 PWM 输出模式

PWM 模块包含 3 个独立的波形发生模块, 对应的 3 对 PWM 输出为 PWM0/PWM01、PWM1/PWM11、PWM2/PWM21, 通过控制相关寄存器可使每对 PWM 输出配置成互补输出模式或独立输出模式。

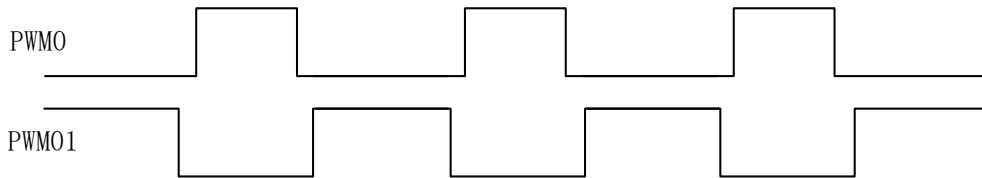
### 10.3.1 互补输出模式

当  $PWMxM(x=0,1,2)$  置 0: PWM 将工作在互补输出模式, 互补输出模式时, 通常使能  $PWMx\&PWMx(x=0,1,2)$  输出, 此时可以控制对应的周期寄存器、占空比寄存器及死区时间寄存器, 从而控制互补波形的输出。互补输出时可选择  $PWMx\&PWMx1(x=0,1,2)$  输出极性, 方便用户各种电平驱动需求。

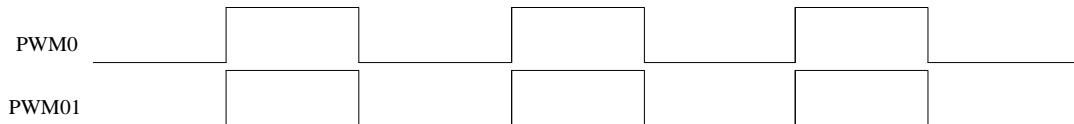
$PWM0S=00\& PWM0M=0$ : PWM0和PWM01工作于互补模式且均为高有效



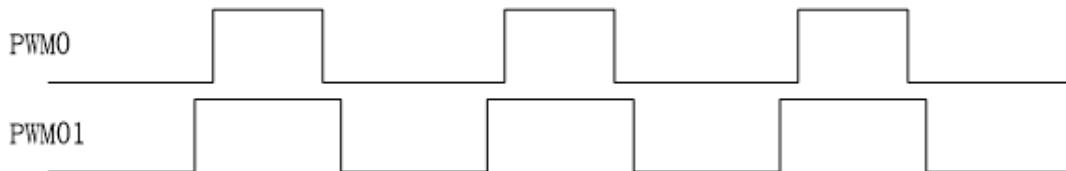
$PWM0S=00\& PWM0M=0$ : PWM0 和 PWM01 工作于互补模式（带死区）且均为高有效



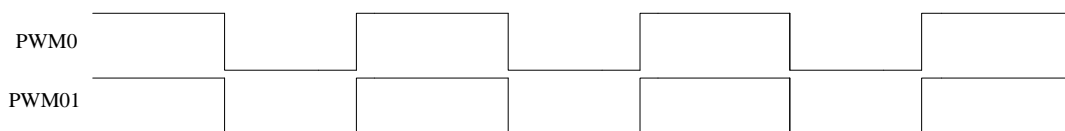
$PWM0S=01\& PWM0M=0$ : PWM0和PWM01工作于互补模式且PWM0为高有效、PWM01为低有效



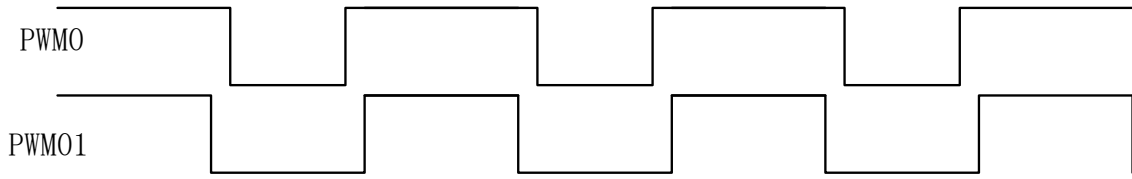
$PWM0S=01\& PWM0M=0$ : PWM0 和 PWM01 工作于互补模式（带死区）且 PWM0 为高有效、PWM01 为低有效



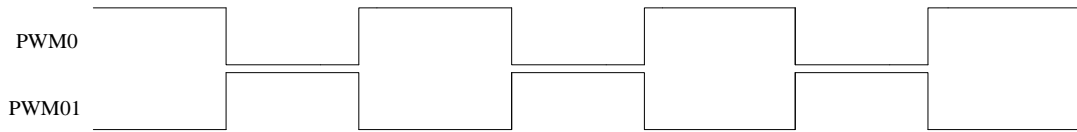
$PWM0S=10\& PWM0M=0$ : PWM0和PWM01工作于互补模式且PWM0为低有效、PWM01为高有效



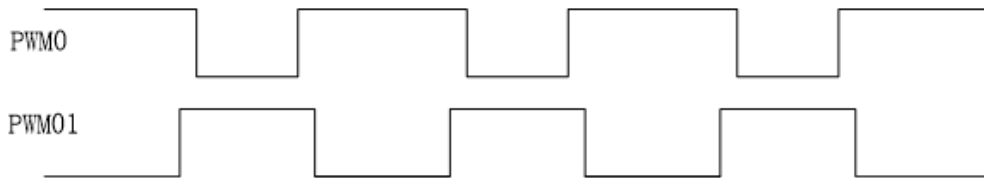
PWM0S=10& PWM0M=0: PWM0 和 PWM01 工作于互补模式（带死区）且 PWM0 为低有效、PWM01 为高有效



PWM0S=11& PWM0M=0: PWM0和PWM01工作于互补模式且PWM0和PWM01均为低有效



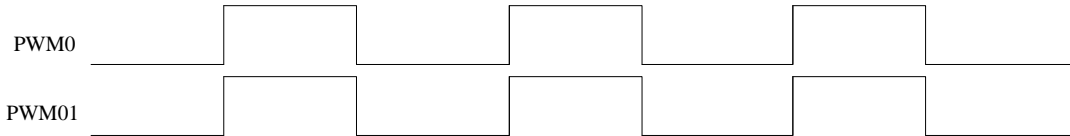
PWM0S=11& PWM0M=0: PWM0 和 PWM01 工作于互补模式（带死区）且 PWM0 和 PWM01 均为低有效



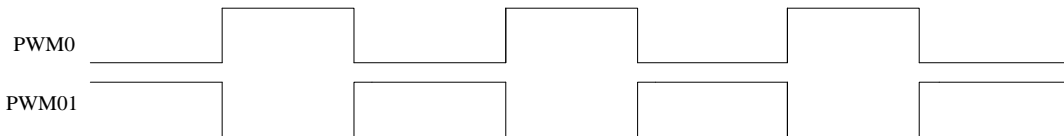
### 10.3.2 独立输出模式

当 PWMxM 置 1: PWM 将工作在独立输出模式, 独立输出模式时, 可以控制相关寄存器使能对应 PWM 端口单一输出或同时输出, 同时让 PWMx&PWMx1 输出时, 其周期相同但占空比可单独设置。当为独立输出模式时占空比寄存器将控制 PWMx 的占空比, 死区时间控制寄存器将控制 PWMx1 的占空比, 独立输出时也可控制 PWMx&PWMx1 输出极性, 方便用户各种电平驱动需求。(x = 0,1,2)

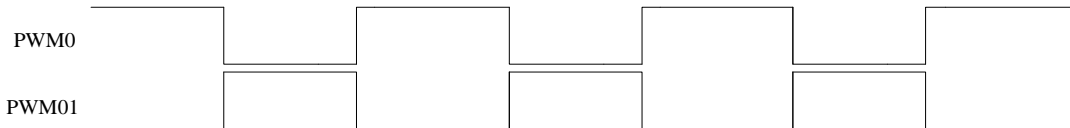
PWM0S=00& PWM0M=1: PWM0和PWM01工作于独立模式且均为高有效



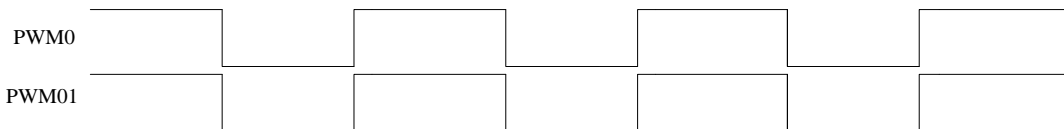
PWM0S=01& PWM0M=1: PWM0和PWM01工作于独立模式且PWM0为高有效、PWM01为低有效



PWM0S=10& PWM0M=1: PWM0和PWM01工作于独立模式且PWM0为低有效、PWM01为高有效



PWM0S=11& PWM0M=1: PWM0和PWM01工作于独立模式且PWM0和PWM01均为低有效



## 10.4 PWM 相关寄存器

### 10.4.1.1 PWM控制寄存器PWMCON0

#### PWMCON0

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	1	1	1	0	0
位符号	TYP2	TYP1	TYP0	RLOAD2	RLOAD1	RLOAD0	-	PENCTRL

位编号	位符号	说明
7	TYP2	PWM2 输出类型选择位 0: PWM2 边沿对齐 1: PWM2 中心对齐
6	TYP1	PWM1 输出类型选择位 0: PWM1 边沿对齐 1: PWM1 中心对齐
5	TYP0	PWM0 输出类型选择位 0: PWM0 边沿对齐 1: PWM0 中心对齐
4	RLOAD2	PWM2 自动重载使能位 0: 禁止自动重载 1: 使能自动重载 注: 默认值为 1, 默认状态下修改参数后, 参数自动载入, 并在下一周期修改 PWM2 周期、占空比、死区。 在修改参数前禁止自动重载, 修改参数之后使能, 可实现多组 PWM 间的同步, 在禁止期间输出之前状态。 <b>适合在三组 PWM 周期相同, 后面想修改 PWM 周期或占空比的情况, 如果三组 PWM 的周期不同, 那么 pwm_ov 信号就不同</b>
3	RLOAD1	PWM1 自动重载使能位 0: 禁止自动重载 1: 使能自动重载
2	RLOAD0	PWM0 自动重载使能位 0: 禁止自动重载 1: 使能自动重载
1	-	保留位 (读为 0, 写无效)
0	PENCTRL	PWM0/1/2 模块使能控制位 0: 由 PWM0_EN、PWM1_EN、PWM2_EN 控制 1: 由 PWMENA 寄存器控制 PWM0/1/2 的模块使能和输出, 且 PWM0_EN、PWM1_EN、PWM2_EN 位不会受此影响改变状态。 注: 关闭时, PWM0 计数停止, 输出立即关闭。打开时, PWM0 计数器都重新从 1 开始计数, 输出受 PWM0_OEN 和 PWM01_OEN 控制。

### 10.4.1.2 PWM使能寄存器PWMEANA

#### PWMEANA

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	PWM21EN	PWM11EN	PWM01EN	-	PWM2EN	PWM1EN	PWM0EN

位编号	位符号	说明
7	-	保留位
6	PWM21EN	PWM21 输出控制位 0: 禁止 PWM21 输出 1: 允许 PWM21输出
5	PWM11EN	PWM11 输出控制位 0: 禁止 PWM11 输出 1: 允许 PWM11输出
4	PWM01EN	PWM01 输出控制位 0: 禁止 PWM01 输出 1: 允许 PWM01输出
3	-	保留位
2	PWM2EN	PWM2 输出控制位 0: 禁止 PWM2 输出 1: 允许 PWM2输出
1	PWM1EN	PWM1 输出控制位 0: 禁止 PWM1 输出 1: 允许 PWM1 输出
0	PWM0EN	PWM0 输出控制位 0: 禁止 PWM0 输出 1: 允许 PWM0输出 注: PWM允许输出(输出时对应端口必须设为输出模式), 必须在PWM管脚映射管脚, 否则为PWM输出关闭状态(输出电平状态与输出模式选择位有关); 即使都禁止输出, 只要相关位被使能, PWM都可以溢出中断, 即此时PWM可以作为定时器使用, 此控制位修改立即生效。

## 10.4.2 PWM0 模块

### 10.4.2.1 PWM0使能寄存器PWM0EN

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	FLT0_MODE	EFLT0	PWM0M	PWM01_OEN	PWM0_OEN	PWM0_EN	

位编号	位符号	说明
7	-	保留位
6-5	FLT0_MODE	PWM0 故障输出预定状态选择位 00: PWM0&PWM01故障期间均为低电平 01: PWM0故障期间低电平, PWM01故障期间高电平 10: PWM0故障期间高电平, PWM01故障期间低电平 11: PWM0&PWM01故障期间均为高电平
4	EFLT0	PWM0 FLT0 控制引脚使能位 0: 禁止故障检测, GPIO 功能或其它功能 1: 允许故障检测, PWM0 故障检测输入引脚 注: 互补输出模式及独立输出模式都可受故障检测脚控制。
3	PWM0M	PWM0 工作模式选择位 0: PWM0&PWM01 工作于互补输出模式 1: PWM0&PWM01 工作于独立输出模式 注: 修改 PWM0 工作模式时建议先关闭 PWM0 模块。 注: 工作在独立模式时, PWM0 的占空比寄存器要大于 PWM01 的占空比寄存器。
2	PWM01_OEN	PWM01 输出控制位 0: 禁止 PWM01 输出 1: 允许 PWM01 输出
1	PWM0_OEN	PWM0 输出控制位 0: 禁止 PWM0 输出 1: 允许 PWM0 输出 注: PWM0 允许输出, 必须在 PWM0_EN 置 1 下才有效, 否则为 PWM0 输出关闭状态(输出时对应端口必须设为输出模式); 即使都禁止输出, 只要 PWM0_EN 位被使能, PWM0 都可以溢出中断, 即此时 PWM0 可以作为定时器使用, 此控制位修改立即生效。
0	PWM0_EN	PWM0 模块使能控制位 0: 关闭 PWM0 模块 1: 打开 PWM0 模块(重新计数) 注: 关闭时, PWM0 计数停止, 输出立即关闭。打开时, PWM0 计数器都重新从 1 开始计数, 输出受 PWM0_OEN 和 PWM01_OEN 控制。

### 10.4.2.2 PWM0控制寄存器PWM0C

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM0IE	PWM0IF	FLT0S	FLT0C	PWM0S		CK0	

位编号	位符号	说明
7	PWM0IE	PWM0 中断允许位 0: 禁止 PWM0 中断 1: 允许 PWM0 中断
6	PWM0IF	PWM0 中断标志位 0: 软件清 0 1: PWM0 周期计数器溢出, 由硬件置 1
5	FLT0S	PWM0 FLT 状态位 0: PWM 正常状态, 软件清 0 1: PWM 输出关闭, 硬件置 1
4	FLT0C	PWM0 FLT 引脚配置位 0: FLT0 为低电平时, PWM 输出关闭 1: FLT0 为高电平时, PWM 输出关闭
3-2	PWM0S	PWM0 和 PWM01 输出模式选择位 00: PWM0和PWM01均为高有效 01: PWM0为高有效, PWM01为低有效 10: PWM0为低有效, PWM01为高有效 11: PWM0 和 PWM01 均为低有效 注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM0 的有效期间为占空比期间, PWM01 的有效期间为占空比的互补期间。
1-0	CK0	PWM0 时钟源选择位 00: $F_{osc}/1$ 01: $F_{osc}/8$ 10: $F_{osc}/32$ 11: $F_{osc}/128$

### 10.4.2.3 PWM0周期寄存器PWM0PL、PWM0PH

#### PWM0PL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM0PL[7:0]							

位编号	位符号	说明
7-0	PWM0PL[7:0]	PWM0 周期寄存器低 8 位

#### PWM0PH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0PH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3-0	PWM0PH[3:0]	PWM0 周期寄存器高 4 位

注：修改PWM0周期时先修改高位，后修改低位，读时不受限制，例如

- (1) PWM0PH = 0x05;
- (2) PWM0PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508
- (3) PWM0PH = 0x06; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508
- (4) PWM0PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0608
- (5) PWM0PL = 0x09; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0609

显然只要修改 PWM 周期，无论低位寄存器是否需要修改，低位都必须写入一次，且周期修改都只会在下一个 PWM 周期才会生效。

PWM0周期 = [ PWM0PH : PWM0PL ] \* PWM0工作时钟源周期



### 10.4.2.4 PWM0占空比寄存器PWM0DL、PWM0DH

#### PWM0DL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM0DL[7:0]							

位编号	位符号	说明
7-0	PWM0DL[7:0]	PWM0 占空比寄存器低 8 位

#### PWM0DH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0DH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3-0	PWM0DH[3:0]	PWM0 占空比寄存器高 4 位

注：修改 PWM0 占空比寄存器，操作类似修改 PWM0 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

PWM0 占空比 = [ PWM0DH : PWM0DL ] \* PWM0 工作时钟周期

### 10.4.2.5 PWM0死区时间寄存器PWM0DTL、PWM0DTH

#### PWM0DTL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM0DTL[7:0]							

位编号	位符号	说明
7-0	PWM0DTL[7:0]	PWM0 死区时间寄存器低 8 位

#### PWM0DTH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0DTH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3-0	PWM0DTH[3:0]	PWM0 死区时间寄存器高 4 位

当 PWM0M=1 时, PWM0 工作在 2 路独立模式, 此时的死区时间寄存器被用来当做 PWM01 的占空比寄存器, 即独立模式的 PWM0 可以产生 2 路周期相同, 但占空比可以不同的 PWM 波形。

互补模式下: PWM0 死区时间 = [ PWM0DTH : PWM0DTL ] \* PWM0 工作时钟周期。

互补模式下: 死区时间必须小于占空比时间, 死区时间与占空比时间的和必须小于 PWM0 周期。

独立模式下: PWM01 占空比时间 = [ PWM0DTH : PWM0DTL ] \* PWM0 工作时钟周期。

## 10.4.3 PWM1 模块

### 10.4.3.1 PWM1 使能寄存器 PWM1EN

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	FLT1_MODE	EFLT1	PWM1M	PWM11_OEN	PWM1_OEN	PWM1_EN	

位编号	位符号	说明
7	-	保留位
6-5	FLT1_MODE	PWM1 故障输出预定状态选择位 00: PWM1&PWM11故障期间均为低电平 01: PWM1故障期间低电平, PWM11故障期间高电平 10: PWM1故障期间高电平, PWM11故障期间低电平 11: PWM1&PWM11故障期间均为高电平
4	EFLT1	PWM1 FLT1 控制引脚使能位 0: 禁止故障检测, GPIO 功能或其它功能 1: 允许故障检测, PWM1 故障检测输入引脚 注: 互补输出模式及独立输出模式都可受故障检测脚控制。
3	PWM1M	PWM1 工作模式选择位 0: PWM1&PWM11 工作于互补输出模式 1: PWM1&PWM11 工作于独立输出模式 注: 修改 PWM1 工作模式时建议先关闭 PWM1 模块。
2	PWM11_OEN	PWM11 输出控制位 0: 禁止 PWM11 输出 1: 允许 PWM11 输出
1	PWM1_OEN	PWM1 输出控制位 0: 禁止 PWM1 输出 1: 允许 PWM1 输出 注: PWM1 允许输出, 必须在 PWM1_EN 置 1 下才有效, 否则为 PWM1 输出关闭状态(输出时对应端口必须设为输出模式); 即使都禁止输出, 只要 PWM1_EN 位被使能, PWM1 都可以溢出中断, 即此时 PWM1 可以作为定时器使用, 此控制位修改立即生效。
0	PWM1_EN	PWM1 模块使能控制位 0: 关闭 PWM1 模块 1: 打开 PWM1 模块(重新计数) 注: 关闭时, PWM1 计数停止, 输出立即关闭。打开时, PWM1 计数器都重新从 1 开始计数, 输出受 PWM1_OEN 和 PWM11_OEN 控制。

### 10.4.3.2 PWM1控制寄存器PWM1C

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM1IE	PWM1IF	FLT1S	FLT1C	PWM1S		CK1	

位编号	位符号	说明
7	PWM1IE	PWM1 中断允许位 0: 禁止 PWM1 中断 1: 允许 PWM1 中断
6	PWM1IF	PWM1 中断标志位 0: 软件清 0 1: PWM1 周期计数器溢出, 由硬件置 1
5	FLT1S	PWM1 FLT 状态位 0: PWM 正常状态, 软件清 0 1: PWM 输出关闭, 硬件置 1
4	FLT1C	PWM1 FLT 引脚配置位 0: FLT1 为低电平时, PWM 输出关闭 1: FLT1 为高电平时, PWM 输出关闭
3-2	PWM1S	PWM1 和 PWM11 输出模式选择位 00: PWM1和PWM11均为高有效 01: PWM1为高有效, PWM11为低有效 10: PWM1为低有效, PWM11为高有效 11: PWM1 和 PWM11 均为低有效 注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM1 的有效期间为占空比期间, PWM11 的有效期间为占空比的互补期间。
1-0	CK1	PWM1 时钟源选择位 00: $F_{osc}/1$ 01: $F_{osc}/8$ 10: $F_{osc}/32$ 11: $F_{osc}/128$

### 10.4.3.3 PWM1周期寄存器PWM1PL、PWM1PH

#### PWM1PL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM1PL[7:0]							

位编号	位符号	说明
7-0	PWM1PL[7:0]	PWM1 周期寄存器低 8 位

#### PWM1PH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM1PH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3-0	PWM1PH[3:0]	PWM1 周期寄存器高 4 位

注：修改PWM1周期时先修改高位，后修改低位，读时不受限制，例如

(6) PWM1PH = 0x05;

(7) PWM1PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508

(8) PWM1PH = 0x06; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508

(9) PWM1PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0608

(10) PWM1PL = 0x09; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0609

显然只要修改 PWM 周期，无论低位寄存器是否需要修改，低位都必须写入一次，且周期修改都只会在下一个 PWM 周期才会生效。

PWM1周期 = [ PWM1PH : PWM1PL ] \* PWM1工作时钟源周期

### 10.4.3.4 PWM1占空比寄存器PWM1DL、PWM1DH

#### PWM1DL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM1DL[7:0]							

位编号	位符号	说明
7-0	PWM1DL[7:0]	PWM1 占空比寄存器低 8 位

### PWM1DH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM1DH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为0，写无效）
3-0	PWM1DH[3:0]	PWM1 占空比寄存器高 4 位

注：修改 PWM1 占空比寄存器，操作类似修改 PWM1 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

PWM1 占空比 = [ PWM1DH : PWM1DL ] \* PWM1 工作时钟周期

### 10.4.3.5 PWM1死区时间寄存器PWM1DTL、PWM1DTH

#### PWM1DTL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM1DTL[7:0]							

位编号	位符号	说明
7-0	PWM1DTL[7:0]	PWM1 死区时间寄存器低 8 位

#### PWM1DTH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-				PWM1DTH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为0，写无效）
3-0	PWM1DTH[3:0]	PWM1 死区时间寄存器高 4 位

当 PWM1M=1 时，PWM1 工作在 2 路独立模式，此时的死区时间寄存器被用来当做 PWM11 的占空比寄存器，即独立模式的 PWM1 可以产生 2 路周期相同，但占空比可以不同的 PWM 波形。

互补模式下：PWM1 死区时间 = [ PWM1DTH : PWM1DTL ] \* PWM1 工作时钟周期；

互补模式下：死区时间必须小于占空比时间，死区时间与占空比时间的和必须小于 PWM1 周期；

独立模式下：PWM11 占空比时间 = [ PWM1DTH : PWM1DTL ] \* PWM1 工作时钟周期；

## 10.4.4 PWM2 模块

### 10.4.4.1 PWM2使能寄存器PWM2EN

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	FLT2_MODE		EFLT2	PWM2M	PWM21_OEN	PWM2_OEN	PWM2_EN

位编号	位符号	说明
7	-	保留位
6-5	FLT2_MODE	PWM2 故障输出预定状态选择位 00: PWM2&PWM21故障期间均为低电平 01: PWM2故障期间低电平, PWM21故障期间高电平 10: PWM2故障期间高电平, PWM21故障期间低电平 11: PWM2&PWM21故障期间均为高电平
4	EFLT2	PWM2 FLT2 控制引脚使能位 0: 禁止故障检测, GPIO 功能或其它功能 1: 允许故障检测, PWM2 故障检测输入引脚 注: 互补输出模式及独立输出模式都可受故障检测脚控制。
3	PWM2M	PWM2 工作模式选择位 0: PWM2&PWM21 工作于互补输出模式 1: PWM2&PWM21 工作于独立输出模式 注: 修改 PWM2 工作模式时建议先关闭 PWM2 模块。
2	PWM21_OEN	PWM21 输出控制位 0: 禁止 PWM21 输出 1: 允许 PWM21 输出
1	PWM2_OEN	PWM2 输出控制位 0: 禁止 PWM2 输出 1: 允许 PWM2 输出 注: PWM2 允许输出, 必须在 PWM2_EN 置 1 下才有效, 否则为 PWM2 输出关闭状态 (输出时对应端口必须设为输出模式); 即使都禁止输出, 只要相 PWM2_EN 位被使能, PWM2 都可以溢出中断, 即此时 PWM2 可以作为定时器使用, 此控制位修改立即生效。
0	PWM2_EN	PWM2 模块使能控制位 0: 关闭 PWM2 模块 1: 打开 PWM2 模块 (重新计数) 注: 关闭时, PWM2 计数停止, 输出立即关闭。打开时, PWM2 计数器都重新从 1 开始计数, 输出受 PWM2_OEN 和 PWM21_OEN 控制。

### 10.4.4.2 PWM2控制寄存器PWM2C

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM2IE	PWM2IF	FLT2S	FLT2C	PWM2S		CK2	

位编号	位符号	说明
7	PWM2IE	PWM2 中断允许位 0: 禁止 PWM2 中断 1: 允许 PWM2 中断
6	PWM2IF	PWM2 中断标志位 0: 软件清 0 1: PWM2 周期计数器溢出, 由硬件置 1
5	FLT2S	PWM2 FLT 状态位 0: PWM 正常状态, 软件清 0 1: PWM 输出关闭, 硬件置 1
4	FLT2C	PWM2 FLT 引脚配置位 0: FLT2 为低电平时, PWM 输出关闭 1: FLT2 为高电平时, PWM 输出关闭
3-2	PWM2S	PWM2 和 PWM21 输出模式选择位 00: PWM2和PWM21均为高有效 01: PWM2为高有效, PWM21为低有效 10: PWM2为低有效, PWM21为高有效 11: PWM2 和 PWM21 均为低有效 注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM2 的有效期间为占空比期间, PWM21 的有效期间为占空比的互补期间。
1-0	CK2	PWM2 时钟源选择位 00: $F_{osc}/1$ 01: $F_{osc}/8$ 10: $F_{osc}/32$ 11: $F_{osc}/128$

### 10.4.4.3 PWM2周期寄存器PWM2PL、PWM2PH

#### PWM2PL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM2PL[7:0]							

位编号	位符号	说明
7-0	PWM2PL[7:0]	PWM2 周期寄存器低 8 位

**PWM2PH**

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM2PH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为0，写无效）
3-0	PWM2PH[3:0]	PWM2 周期寄存器高 4 位

注：修改PWM2周期时先修改高位，后修改低位，读时不受限制，例如

(11) PWM2PH = 0x05;

(12) PWM2PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508

(13) PWM2PH = 0x06; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508

(14) PWM2PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0608

(15) PWM2PL = 0x09; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0609

显然只要修改 PWM 周期，无论低位寄存器是否需要修改，低位都必须写入一次，且周期修改都只会在下一个 PWM 周期才会生效。

PWM2周期 = [ PWM2PH : PWM2PL ] \* PWM2工作时钟源周期

**10.4.4.4 PWM2占空比寄存器PWM2DL、PWM2DH**
**PWM2DL**

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM2DL[7:0]							

位编号	位符号	说明
7-0	PWM2DL[7:0]	PWM2 占空比寄存器低 8 位

**PWM2DH**

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-				PWM2DH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为0，写无效）
3-0	PWM2DH[3:0]	PWM2 占空比寄存器高 4 位

注：修改 PWM2 占空比寄存器，操作类似修改 PWM2 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

PWM2 占空比 = [ PWM2DH : PWM2DL ] \* PWM2 工作时钟周期



### 10.4.4.5 PWM2死区时间寄存器PWM2DTL、PWM2DTH

#### PWM2DTL

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM2DTL[7:0]							

位编号	位符号	说明
7-0	PWM2DTL[7:0]	PWM2 死区时间寄存器低 8 位

#### PWM2DTH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-				PWM2DTH[3:0]			

位编号	位符号	说明
7-4	-	保留位（读为 0，写无效）
3-0	PWM2DTH[3:0]	PWM2 死区时间寄存器高 4 位

当 PWM2M=1 时，PWM2 工作在 2 路独立模式，此时的死区时间寄存器被用来当做 PWM21 的占空比寄存器，即独立模式的 PWM2 可以产生 2 路周期相同，但占空比可以不同的 PWM 波形。

互补模式下：PWM2 死区时间 = [ PWM2DTH : PWM2DTL ] \* PWM2 工作时钟周期；

互补模式下：死区时间必须小于占空比时间，死区时间与占空比时间的和必须小于 PWM2 周期；

独立模式下：PWM21 占空比时间 = [ PWM2DTH : PWM2DTL ] \* PWM2 工作时钟周期；

# 11 单路8位PWM模块

## 11.1 PWM 特性

- 8 位 PWM 输出
- 提供 PWM 周期溢出中断，但中断与互补 12 位 PWM 共用同一向量
- 输出极性可选择
- PWM 可做定时器/计数器使用，即周期寄存器写入时做定时器使用，读时做计数器使用

## 11.2 PWM 模块相关寄存器

### 11.2.1 PWM3 模块

#### 11.2.1.1 PWM3控制寄存器PWM3C

##### PWM3C

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM3EN	PWM3IE	PWM3IF	PWM3OEN	PWM3S	PTCK3[2:0]		

位编号	位符号	说明
7	PWM3EN	PWM3 模块使能控制位 0: 关闭 PWM3 模块 1: 打开 PWM3 模块（重新计数） 注：关闭时，PWM 计数停止，输出立即关闭。 打开时，PWM 计数器都重新从 1 开始计数，输出受 PWM3OEN 控制。
6	PWM3IE	PWM3 中断允许位 0: 禁止 PWM3 中断 1: 允许 PWM3 中断
5	PWM3IF	PWM3 中断标志位 0: 软件清 0 1: 硬件置 1，仅在 PWM3 计数器溢出（大于 PWM3P 时）才置 1
4	PWM3OEN	PWM3 输出使能位 0: PWM3 禁止输出 1: PWM3 允许输出 注：PWM 允许输出，必须在 PWM3EN 置 1 下才有效，否则为 PWM 输出关闭状态（输出时对应端口必须设为输出模式）；即使都禁止输出，只要相关位被使能，PWM 都可以溢出中断，即此时 PWM 可以作为定时器/计数器使用，此控制位修改立即生效。
3	PWM3S	PWM3 输出极性选择位

		0: PWM3 有效期间为高电平 1: PWM3 有效期间为低电平 注: 修改此控制位, 立即生效, 有效期间是指占空比期间
2-0	PTCK3[2:0]	PWM3 工作时钟源选择位 000: $F_{osc}/1$ 001: $F_{osc}/2$ 010: $F_{osc}/4$ 011: $F_{osc}/8$ 100: $F_{osc}/16$ 101: $F_{osc}/32$ 110: $F_{osc}/64$ 111: $F_{osc}/128$ 注: 修改此控制位, 立即生效, 不建议在输出过程中修改。

### 11.2.1.2 PWM3周期寄存器PWM3P

#### PWM3P

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM3P[7:0]							

位编号	位符号	说明
7-0	PWM3P[7:0]	PWM3 周期寄存器

### 11.2.1.3 PWM3D占空比寄存器PWM3D

#### PWM3D

位编号	7	6	5	4	3	2	1	0
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
位符号	PWM3D[7:0]							

位编号	位符号	说明
7-0	PWM3D[7:0]	PWM3 占空比寄存器 PWM3P $\leq$ PWM3D 时, 占空比 100% PWM3D = 0X00 时, 占空比 0%

## 12 看门狗定时器WDT

### 12.1 WDT 特性

- 可配置是否溢出复位
- 可配置在空闲/掉电模式下是否允许
- 可灵活配置溢出时间

HC89F0431A/0421A/0411A看门狗定时器是一个递增计数器，其时钟源内部低频RC，可以通过寄存器选择在空闲/掉电模式下是否运行。WDT溢出时，芯片是否复位可通过RSTFR寄存器里的WDTRF位来判断。如果WDTRST为1，则WDT溢出时会复位系统，如果WDTRST为0，而且WDT的中断使能，则会产生WDT中断。

由于内部低频RC振荡器频率随工艺角有偏差，可通过定时器5进行捕获测量内部低速RC振荡器的真实频率，然后根据实际的频率值进行溢出时间的计算。

HC89F0431A/0421A/0411A看门狗定时器溢出后有溢出标志，复位有专用的复位标志，可设分频，可设计数溢出值，清WDT操作只需置相应控制位即可，操作灵活。

## 12.2 WDT 相关寄存器

### 12.2.1 WDT 控制寄存器 WDTC

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	W	R/W	R/W	R/W	R/W
复位值	0	1	0	0	1	1	1	1
位符号	-	WDTRST	WDTF	WDTCLR	WDTPD	WDTPS[2:0]		

位编号	位符号	说明
7	-	保留位
6	WDTRST	WDT 复位允许位 0: 禁止 WDT 复位 1: 允许 WDT 复位 注: 禁止 WDT 复位时, WDT 计数溢出依旧可以置中断请求标志。
5	WDTF	WDT 中断请求标志位 0: 无 WDT 计数溢出, 中断响应时软件清 0 1: WDT 计数溢出, WDTF 硬件置 1, 可用于中断请求
4	WDTCLR	看门狗清零位 置 1 会清零 WDT 计数器, 硬件自动清 0 该位
3	WDTPD	WDT 空闲/掉电模式下运行控制位 0: 空闲/掉电模式下允许 WDT 运行, 如果复位使能打开则会复位唤醒系统, 如果复位使能关闭, 而且 EA=1, EWDT=1, 会中断唤醒系统。 1: 空闲/掉电模式下禁止 WDT 运行
2-0	WDTPS[2:0]	看门狗定时器时钟源分频选择位 000: /8 001: /16 010: /32 011: /64 100: /128 101: /256 110: /512 111: /1024

## 12.2.2 WDT 计数比较寄存器 WDTCCR

位编号	7	6	5	4	3	2	1	0
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
位符号	WDTCCR[7:0]							

位编号	位符号	说明
7-0	WDTCCR[7:0]	WDT 计数比较寄存器 注：WDT 计数器与 WDTCCR[7:0]匹配时，溢出并且计数器清 0 重新计数，写入 00 时，将关闭 WDT 功能（但不关闭内部低频 RC），即相当于禁止 WDT。写入非 0 数据时，将启动 WDT。

下面以 44KHz 为例进行计算，真实的内部低频 RC 频率可通过定时器 5 进行捕获测量得到。

溢出时间 = WDT分频系数 \* (WDTCCR[7:0]+1)/实际内部低频RC频率。

WDTCCR[7:0] = 0xFF 看门狗溢出时间如下表。

PS2	PS1	PS0	WDT分频系数	调整步进值	WDT最大溢出时间@44K
0	0	0	8	0.182ms	46.55ms
0	0	1	16	0.364ms	93.09ms
0	1	0	32	0.728ms	186.18ms
0	1	1	64	1.456ms	372.36ms
1	0	0	128	2.912ms	744.73ms
1	0	1	256	5.824ms	1489.45ms
1	1	0	512	11.648 ms	2978.91ms
1	1	1	1024	23.296ms	5957.82ms

# 13 通用异步收发器UART

## 13.1 UART 特性

- 2 个 UART
- 多种工作方式
- 多种错误检测

## 13.2 工作方式

UART有4种工作方式，在四种方式中，任何将SBUF作为目标寄存器的写操作都会启动发送。在方式0中由条件RI = 0和REN = 1初始化接收。这会在TXD引脚上产生一个时钟信号，然后在RXD引脚上移出8位数据。在其它方式中由输入的起始位初始化接收（如果RI = 0和REN = 1）。外部发送器通信以发送起始位开始。在发送之前TXD引脚必须被设置为输出高电平。

SM0	SM1	工作方式	类型	波特率
0	0	方式0	同步	波特率是 $F_{osc}/12 \times 6^{UX6}$
0	1	方式1	异步	定时器4的溢出率/16
1	0	方式2	异步	$(2^{SMOD}/64) \times F_{osc}$
1	1	方式3	异步	定时器4的溢出率/16

### 13.2.1 方式 0：同步半双工通讯

方式0支持与外部设备的同步通信，在RXD引脚上收发串行数据，TXD引脚发送移位时钟。HC89F0431A/0421A/0411A提供TXD引脚上的移位时钟，因此这种方式是串行通信的半双工方式。在这个方式中，每帧收发8位，低位先接收或发送。

通过置UX6位为0或1，波特率固定为 $F_{osc}$ 的1/12或1/2。当UX6位等于0时，串行端口以 $F_{osc}$ 的1/12运行，当UX6位等于1时，串行端口以 $F_{osc}$ 的1/2运行。与标准8051唯一不同的是，HC89F0431A/0421A/0411A在方式0中有可变波特率。

功能块框图如下图所示，数据通过RXD引脚移入和移出串行端口，移位时钟由TXD引脚输出。

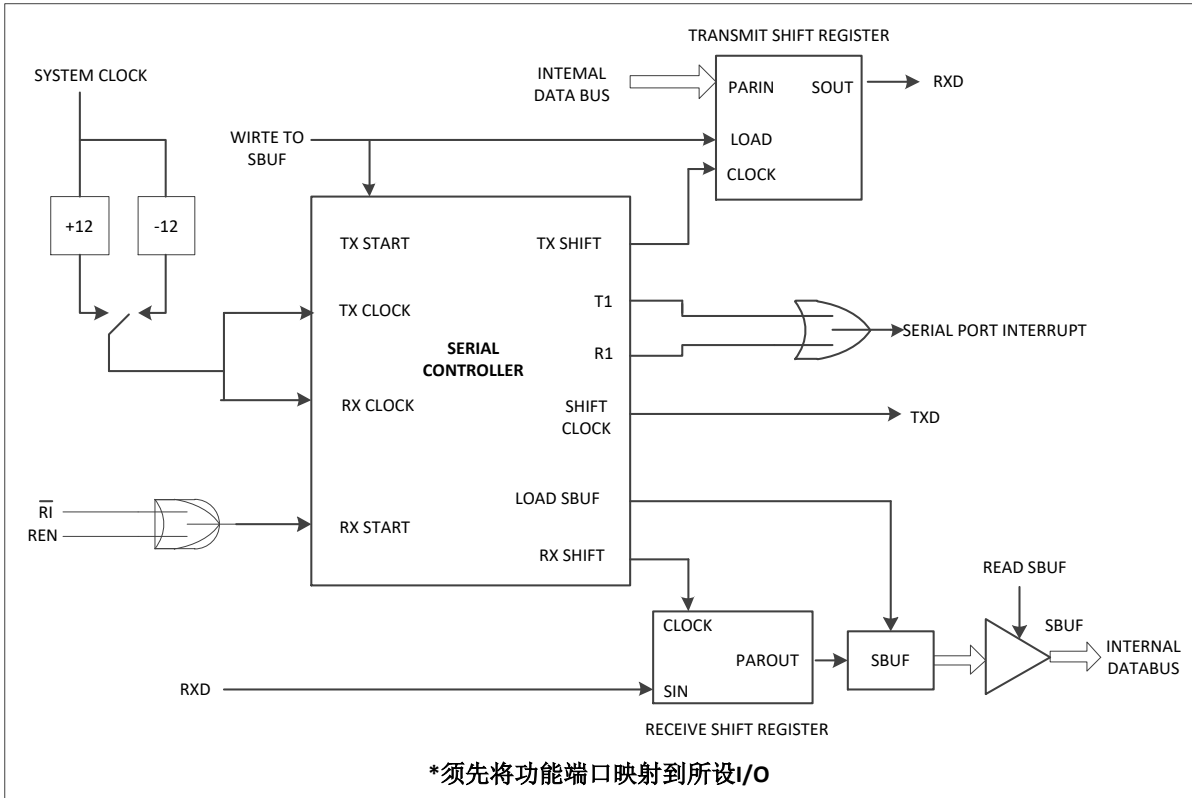


Figure 13-1 UART 模式 0 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送。下一个系统时钟 TX 控制块开始发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置 0。当移位寄存器中的所有 8 位都发送后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TI 位置 1。

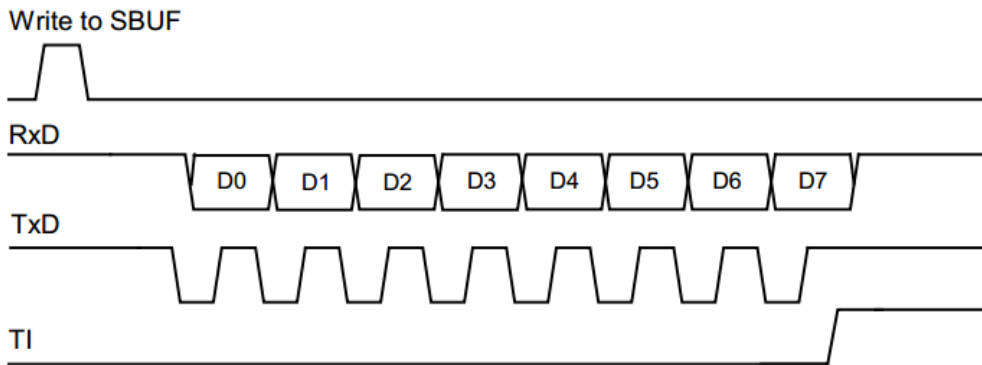


Figure 13-2 模式 0 数据发送时序框图



REN 位置 1 和 RI 位清 0 初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有 8 位数据都移到移位寄存器中后，RX 控制块停止接收，在下一个系统时钟的上升沿 RI 置位，直到被软件清零才允许下一次接收。

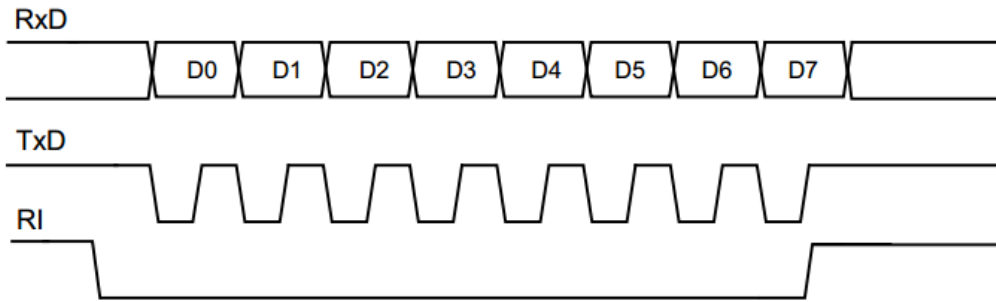


Figure 13-3 模式 0 数据接收时序框图

### 13.2.2 方式 1: 8 位 UART, 可变波特率, 异步全双工

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0），8 个数据位（低位在前）和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中而停止位储存在 RB8 中。方式 1 中的波特率为定时器 4 的溢出率/16。

功能块框图如下图所示：

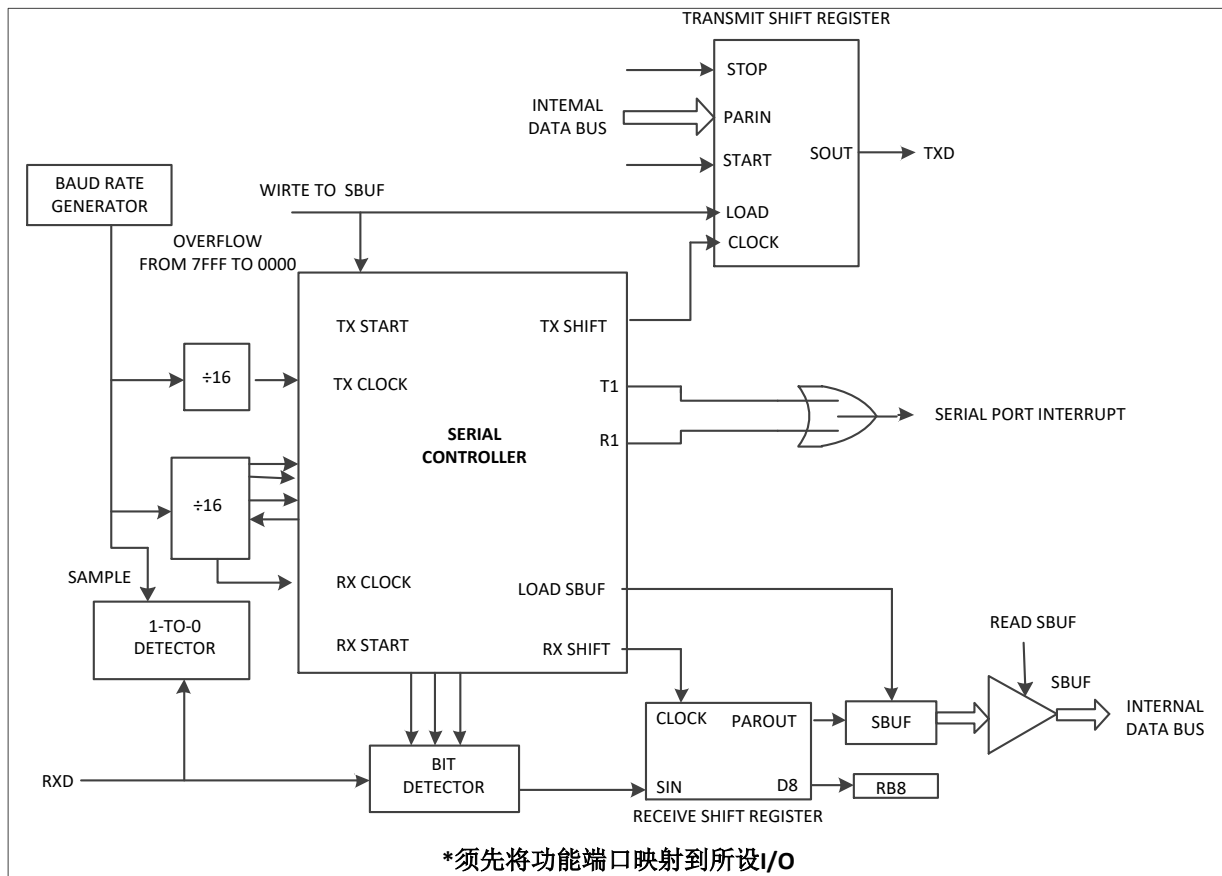


Figure 13-4 UART 模式 1 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送，实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后，停止位在 TXD 引脚上移出，在停止位发出的同时 TI 标志置位。

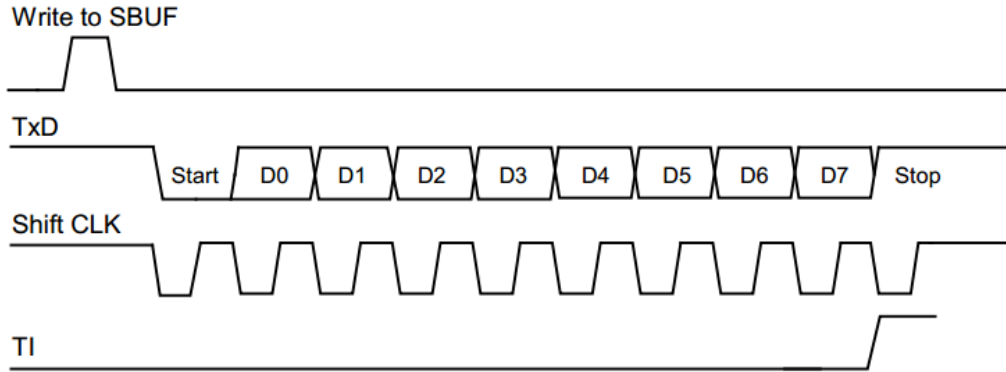


Figure 13-5 模式 1 数据发送时序框图

只有REN置1时才允许接收。当RXD引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位，这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8个数据位和1个停止位（包含错误的停止位，详见寄存器SM2位说明）移入之后，移位寄存器的内容和停止位(包含错误的停止位)被分别装入SBUF和RB8中，RI置1，但必须满足下列条件：

- (1) RI = 0
- (2) SM2 = 0或者接收的停止位= 1

如果这些条件被满足，那么停止位（包含错误的停止位）装入 RB8，8 个数据位装入 SBUF，RI 被置位。否则接收的帧会丢失。这时，接收器将重新去探测 RXD 端是否有另一个下降沿。用户必须用软件清零 RI，然后才能再次接收。

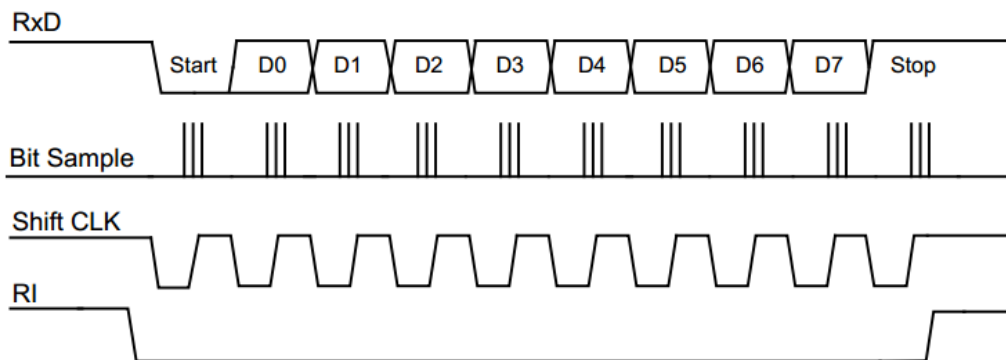


Figure 13-6 模式 1 数据接收时序框图

### 13.2.3 方式 2: 9 位 UART, 固定波特率, 异步全双工

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位在前），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 2 支持多机通信和硬件地址识别（详见多机通信章节）。在数据传送时，第 9 数据位（TB8 位）可以写 0 或 1，例如，可写入 PSW 中的奇偶位 P，或用作多机通信中的数据/地址标志位。当接收到数据时，第 9 数据位移入 RB8 而停止位不保存。SMOD 位选择波特率为系统工作频率的 1/32 或 1/64。功能块框图如下所示。

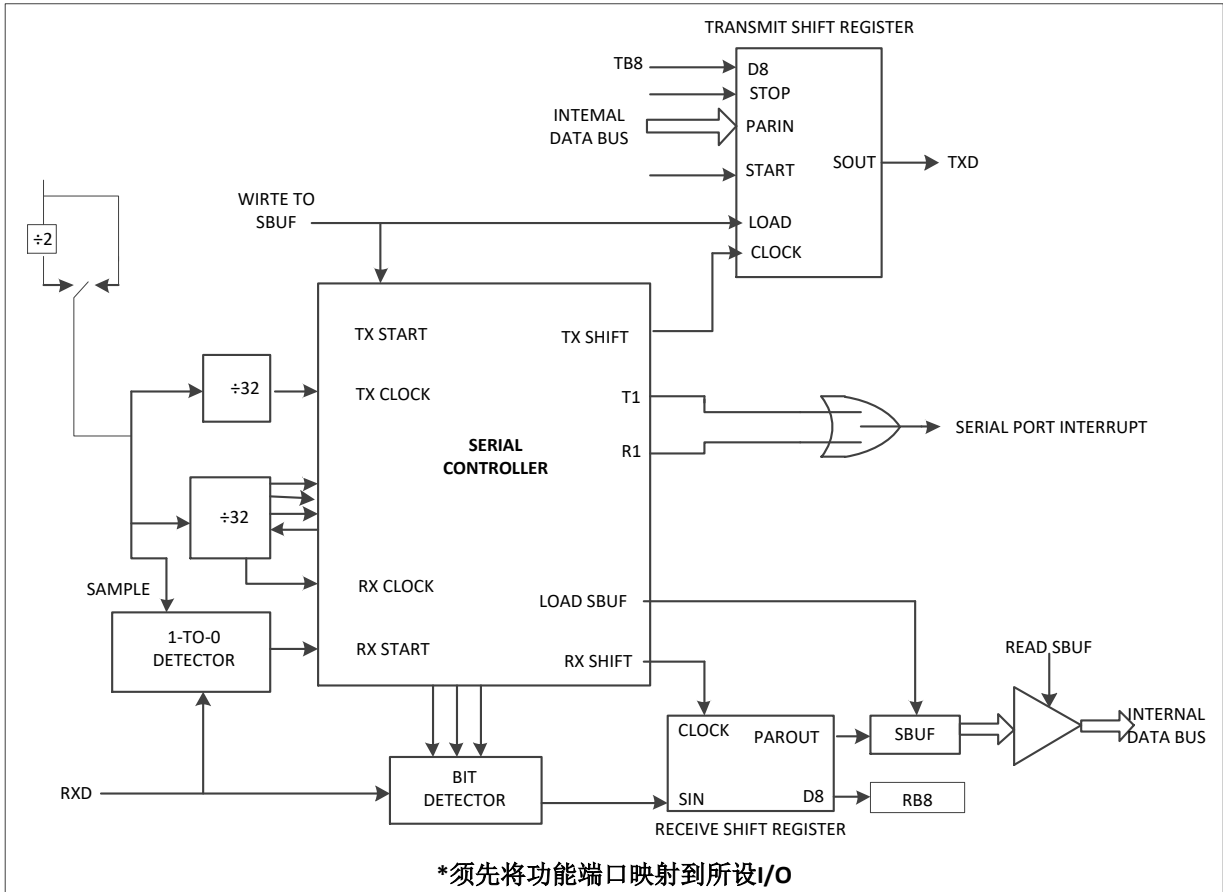


Figure 13-7 UART 模式 2 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TXD 引脚上移出，在停止位开始发送时 TI 标志置位。

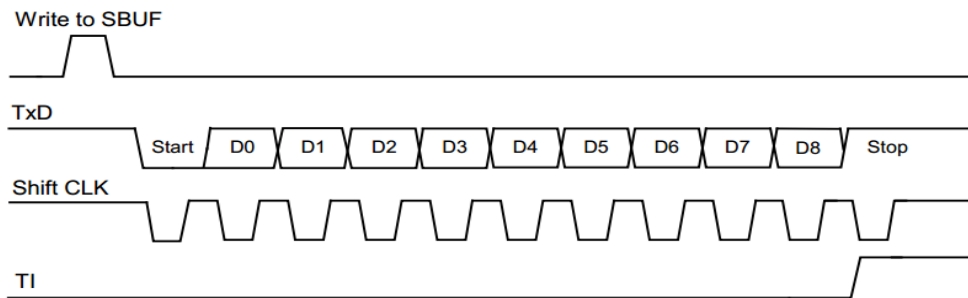


Figure 13-8 模式 2 数据发送时序框图

只有REN置位时才允许接收。当RXD引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位。这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9个数据位和1个停止位移入之后，移位寄存器的内容被分别装入SBUF和RB8中，RI置1，但必须满足下列条件：

- (1) RI = 0
- (2) SM2 = 0或者接收的第9位= 1，且接收的字节符合约定从机地址

如果这些条件被满足，那么第9位移入RB8，8位数据移入SBUF，RI被置位。否则接收的数据帧会丢失。

在停止位的当中，接收器回到寻找RXD引脚上的另一个下降沿。用户必须用软件清除RI，然后才能再次接收。

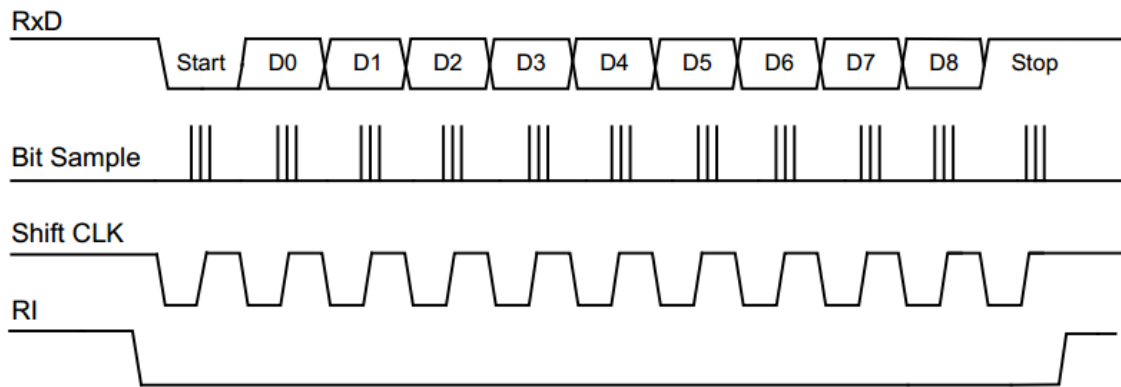


Figure 13-9 模式 2 数据接收时序框图

### 13.2.4方式 3: 9 位 UART, 可变波特率, 异步全双工

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式。

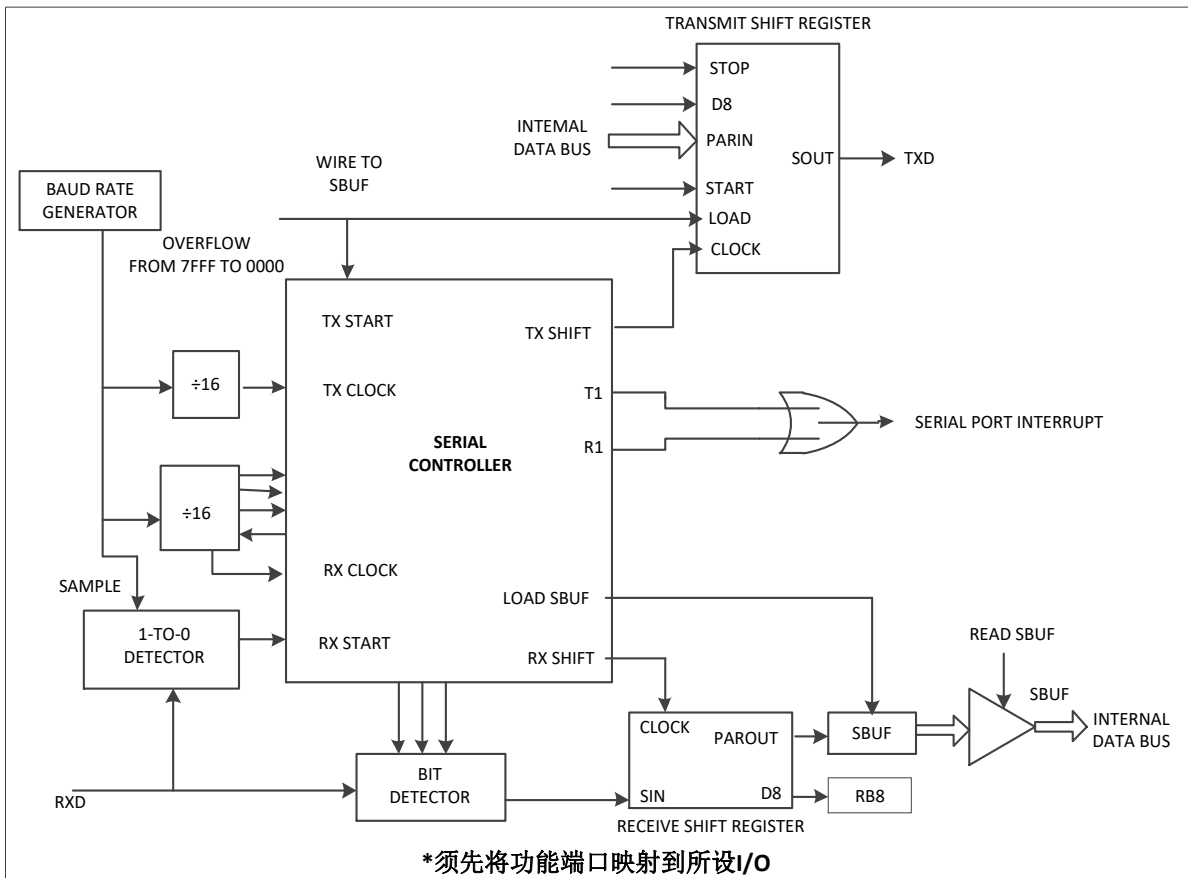


Figure 13-10 UART 模式 3 功能框图

## 13.3 波特率发生器

定时器 4 可以作为 UART1/UART2 的波特率发生器。选择定时器 4 的工作方式 1 作为波特率发生器。该方式与自动重载方式相似。定时器 4 的溢出会引起软件将定时器 4 重载寄存器中的 16 位值载入定时器 4 的计数器中，此时也会产生溢出中断。如果不希望中断发生，可以关闭 ET4。波特率由下列公式计算：

$$\text{BaudRate} = \frac{1}{16} \times \frac{f_{T4}/\text{PRESCALER}}{65536 - [\text{TH4}, \text{TL4}]}, \text{ 用定时器 4 作为波特率发生器。}$$

上式中，TH4 和 TL4 为定时器 4 数据寄存器。

下表为常用Fosc频率与常用波特率所对应的定时器4的计数值：

常用波特率	Fosc			
	4MHz	8MHz	16MHz	32MHz
1200	FF30	FE5F	FCBF	F97D
2400	FF98	FF30	FE5F	FCBF
4800	FFCC	FF98	FF30	FE5F
9600	FFE6	FFCC	FF98	FF30
19200	FFF3	FFE6	FFCC	FF98
38400	/	FFF3	FFE6	FFCC
57600	/	/	FFEF	FFDD
115200	/	/	/	FFEF

## 13.4 多机通信

### 13.4.1 软件地址识别

方式2和方式3具有适用于多机通讯功能。在这两个方式下，接收的是9位数据，第9位移入RB8中，之后是停止位。可以这样设定UART：当接收到停止位，且RB8 = 1时，串行口中断有效（请求标志RI置位）。此时置位SM2位，UART工作在多机通讯模式。

在多机通讯系统中，按如下所述来使用这一功能。当主机要发送一数据块给几个从机中的一个时，先发送一地址字节，以寻址目标从机。地址字节与数据字节可用第9数据位来区别，地址字节的第9位为1，数据字节的第9位为0。

如果从机SM2为1，则不会响应数据字节中断。地址字节可以使所有从机产生中断，每一个从机都检查所接收到的地址字节，以判别本机是不是目标从机。被寻到的从机对SM2位执行清零操作，并准备接收即将到来的数据字节。当接收完毕时，从机再一次将SM2置位。没有被寻址的从机，则保持SM2位为1，不响应数据字节。

注意：在方式1中，SM2用来检测停止位是否有效，如果SM2 = 1，接收中断不会响应直到接收到一个有效的停止位。

### 13.4.2 自动（硬件）地址识别

在方式2和方式3中，SM2置位，UART运行状态如下：接收到停止位，RB8的第9位为1（地址字节），且接收到的数据字节符合UART的从机地址，UART产生一个中断。从机将SM2清零，接收后续数据字节。

第9位为1表明该字节是地址而非数据。当主机要发送一组数据给几个从机中的一个时，必须先发送目标从机地址。所有从机等待接收地址字节，为了确保仅在接收地址字节时产生中断，SM2位必须置位。自动地址识别的特点是只有地址匹配的从机才能产生中断，硬件完成地址比较。

中断产生后，地址匹配的从机清零SM2，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。全部信息接收完毕后，地址匹配的从机应该再次把SM2置位，忽略所有传送的非地址字节，直到接收到下一个地址字节。

使用自动地址识别功能时，主机可以通过调用给定的从机地址选择与一个或多个从机通信。主机使用广播地址可以寻址所有从机。有两个特殊功能寄存器，从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个8位的字节，存于SADDR寄存器中。SADEN用于定义SADDR各位的有效与否，如果SADEN中某一位为0，则SADDR中相应位被忽略，如果SADEN中某一位置位，则SADDR中相应位将用于产生约定地址。这可以使用户在不改变SADDR寄存器中的从机地址的情况下灵活地寻址多个从机。

	从机1	从机2
SADDR	10100100	10100111
SADEN	11111010	11111001
约定地址	10100x0x	10100xx1
广播地址	1111111x	11111111

从机1和从机2的约定地址最低位是不同的。从机1忽略了最低位，而从机2的最低位是1。因此只与从机1通讯时，主机必须发送最低位为0的地址（10100000）。类似地，从机1的第1位为0，从机2的第1位被忽略。因此，只与从机2通讯时，主机必须发送第1位为1的地址（10100011）。如果主机需要同时与两从机通讯，则第0位为1，第1位为0，第2位被两从机都忽略，两个不同的地址用于选定两个从机（1010 0001和1010 0101）。

主机可以通过广播地址与所有从机同时通讯。这个地址等于SADDR和SADEN的位或，结果中的0表示该位被忽略。多数情况下，广播地址为0xFF，该地址可被所有从机应答。

系统复位后，SADDR和SADEN两个寄存器初始化为0，这两个结果设定了约定地址和广播地址为xxxxxxx（所有位都被忽略）。这有效地去除了多从机通讯的特性，禁止了自动寻址方式。这样的UART将对任何地址都产生应答，兼容了不支持自动地址识别的8051控制器。用户可以按照上面提到的方法实现软件地址识别的多机通讯。

## 13.5 帧出错检测

3个错误标志位被置位后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

### 13.5.1 发送冲突

如果在一个数据发送正在进行时，用户软件写数据到SBUF寄存器时，发送冲突位（TXCOL位）置1。如果发生了冲突，新数据会被忽略，不能被写入发送缓冲器（即不影响传送）。

### 13.5.2 接收溢出

RI置1，接收缓冲器中的数据未被读取，RI被清0，又开始新的数据接收，若在新的数据接收完成前（RI置1）还未读取之前接收缓冲区中的数据，在那么接收溢出位（RXROV位）置位。如果发生了接收溢出，接收缓冲器中原来的数据不影响，后面的数据则丢失。

### 13.5.3 帧出错

如果检测到一个无效（低）停止位，那么帧出错位（FE位）置1。



## 13.6 UART1 相关寄存器

### 13.6.1 UART1 控制寄存器 SCON、SCON2

#### SCON

位编号	7	6	5	4	3	2	1	0
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
位符号	FE	RXROV	TXCOL	REN	TB8	RB8	TI	RI

位编号	位符号	说明
7	FE	帧错误检测位 0: 无帧错误或软件清 0 1: 有帧错误, 硬件置 1
6	RXROV	接收溢出标志位 0: 无接收溢出或软件清 0 1: 接收溢出, 硬件置 1
5	TXCOL	发送冲突标志位 0: 无发送冲突或软件清 0 1: 有发送冲突, 硬件置 1
4	REN	串行接收使能控制位 0: 禁止串行接收 1: 允许串行接收
3	TB8	方式 2/方式 3 时, 为要发送的第 9 位数据, 由软件置 1 或清 0
2	RB8	方式 2/方式 3 时, 为接收到的第 9 位数据, 作为奇偶校验位或地址帧/数据帧的标志位
1	TI	发送中断请求中断标志位 0: 软件清 0 1: 方式 0 时, 当串行发送数据第 8 位结束时, 由硬件自动置 1, 其它方式时, 在停止位开始发送时由硬件置 1
0	RI	接收中断请求中断标志位 0: 软件清 0 1: 方式 0 时, 当串行接收数据第 8 位结束时, 由硬件自动置 1, 其它方式时, 串行接收到停止位开始时刻由硬件置 1

**SCON2**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R/W	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	SMOD	-	UX6	-		SM0	SM1	SM2

位编号	位符号	说明
7	SMOD	波特率加倍控制位 0: 在方式2中, 波特率为系统时钟 $F_{osc}$ 的 1/64 1: 在方式2中, 波特率为系统时钟 $F_{osc}$ 的 1/32
6	-	保留位 (读为0, 写无效)
5	UX6	串口模式0的通信速度设置位 0: 串口模式0时钟为 $F_{osc}/12$ 1: 串口模式0时钟为 $F_{osc}/2$
4-3	-	保留位 (读为0, 写无效)
2-1	SM0:SM1	串口工作方式选择位, 详细见下表
0	SM2	多机通信使能控制位(第九位“1”校验器) 0: 在方式1时, 不检测停止位, 停止位无论是0还是1都会置位RI 在方式2和3时, 不检测第9位, 任何字节都会置位RI 1: 在方式1时, 允许停止位确认检验, 只有有效的停止位“1”才能置位RI 在方式2和3时, 只有地址字节 (第9位=“1”)才能置位RI

SM0	SM1	工作方式	功能说明	波特率
0	0	方式0	同步移位串行方式: 移位寄存器	当UX6 = 0时, 波特率是 $F_{osc}/12$ 当UX6 = 1时, 波特率是 $F_{osc}/2$
0	1	方式1	8位UART, 波特率可变	定时器4/5的溢速率/16
1	0	方式2	9位UART	$(2^{SMOD}/64) \times F_{osc}$
1	1	方式3	9位UART, 波特率可变	定时器4/5的溢速率/16

**13.6.2 UART1 数据缓冲寄存器 SBUF**

位编号	7	6	5	4	3	2	1	0
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
位符号	SBUF[7:0]							

位编号	位符号	说明
7-0	SBUF[7:0]	串口缓冲寄存器: 写为需要发送的数据, 读为接收到的数据 实际上该地址上有2个独立寄存器, 一个用于接收数据, 一个用于发送数据。当数据写入SBUF, 这是个发送寄存器并且移位进行串口发送。当数据从SBUF读出, 这是个接收寄存器。

### 13.6.3 UART1 自动地址识别 SADDR、SADEN

#### 从机地址寄存器 SADDR

位编号	7	6	5	4	3	2	1	0
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
位符号	SADDR[7:0]							

位编号	位符号	说明
7-0	SADDR[7:0]	从机地址寄存器

#### 从机地址掩码寄存器 SADEN

位编号	7	6	5	4	3	2	1	0
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
位符号	SADEN [7:0]							

位编号	位符号	说明
7-0	SADEN [7:0]	从机地址掩码寄存器

### 13.6.4 波特率选择寄存器 BRTSEL

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号							UART2_BRT_SEL	UART1_BRT_SEL

位编号	位符号	说明
7-2	-	保留位
1	UART2_BRT_SEL	UART2 波特率选择位 0: 定时器 5 的溢出率 1: 定时器 4 的溢出率
0	UART1_BRT_SEL	UART1 波特率选择位 0: 定时器 4 的溢出率 1: 定时器 5 的溢出率

## 13.7 UART2

UART2的控制和工作方式与UART1相同，寄存器请参考UART1  
不同点：

1. UART2的寄存器存放在扩展SFR中；
2. UART2只有两种工作方式；
3. UART2没有错误检测；
4. UART2没有自动硬件地址识别。

### 13.7.1 UART2 工作方式

#### 13.7.1.1 方式0：8位UART，可变波特率，异步全双工

方式0提供10位全双工异步通信，10位由一个起始位（逻辑0），8个数据位（低位在前）和一个停止位（逻辑1）组成。在接收时，这8个数据位存储在S2BUF中而停止位储存在RB8中。方式0中的波特率为定时器5溢出率的1/16。

任何将S2BUF作为目标寄存器的写操作都会启动发送，实际上发送是从16分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与16分频计数器是同步的，与对S2BUF的写操作不同步。起始位首先在TXD引脚上移出，然后是8位数据位，在发送移位寄存器中的所有8位数据都发送完后，停止位在TXD引脚上移出，在停止位发出的同时TI标志置位。

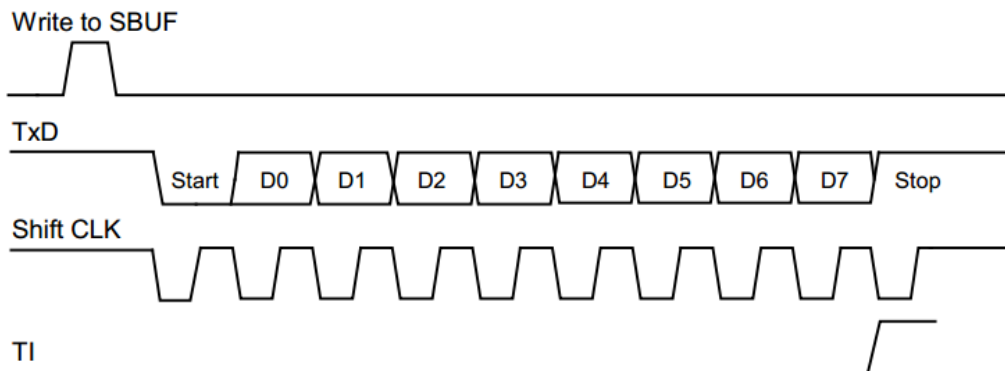


Figure 13-11 Send Timing of Mode 1

只有REN置1时才允许接收。当RXD引脚检测到下降沿时串口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位，这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8个数据位和1个停止位（包含错误的停止位，详细见寄存器SM2位说明）移入之后，移位寄存器的内容和停止位（包含错误的停止位）被分别装入SBUF和RB8中，RI置1，但必须满足下列条件：

- (1) RI = 0
- (2) SM2 = 0不判断停止位或者SM2=1判断停止位，且停止位必须为1

如果这些条件被满足，那么停止位（包含错误的停止位）装入RB8，8个数据位装入SBUF，RI被置位。否则接收的帧会丢失。这时，接收器将重新去探测RXD端是否另一个下降沿。用户必须用软件清零RI，然后才能再次接收。

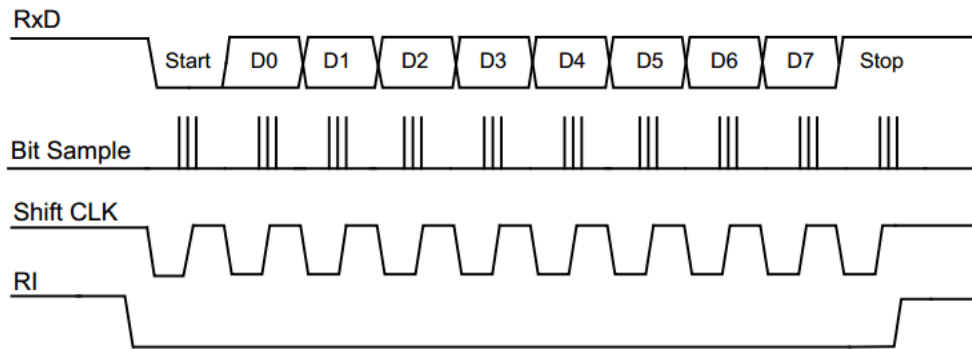


Figure 13-12 Receive Timing of Mode 1

### 13.7.1.2 方式1: 9位UART, 可变波特率, 异步全双工

这个方式使用异步全双工通信中的11位。一帧由一个起始位（逻辑0），8个数据位（低位在前），一个可编程的第9数据位和一个停止位（逻辑1）组成。方式1支持多机通信，在数据传送时，第9数据位（TB8位）可以写0或1，例如，可写入PSW中的奇偶位P，或用作多机通信中的数据/地址标志位。当接收到数据时，第9数据位移入RB8而停止位不保存。

任何将SBUF作为目标寄存器的写操作都会启动发送，同时也将TB8载入到发送移位寄存器的第9位中。实际上发送是从16分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与16分频计数器是同步的，与对SBUF的写操作不同步。起始位首先在TXD引脚上移出，然后是9位数据。在发送转换寄存器中的所有9位数据都发送完后，停止位在TXD引脚上移出，在停止位开始发送时TI标志置位。

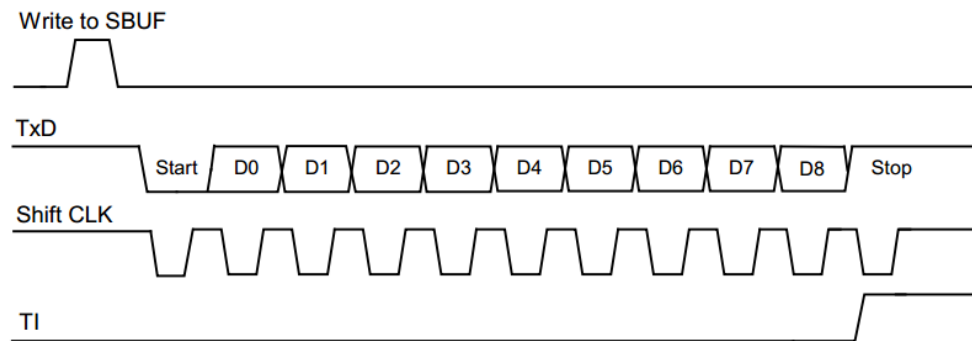


Figure 13-13 Send Timing of Mode 2

只有REN置位时才允许接收。当RXD引脚检测到下降沿时串口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位。这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9个数据位移入之后，移位寄存器的内容被分别装入SBUF和RB8中，但必须满足下列条件：

- (3) RI = 0
- (4) SM2 = 0

如果这些条件被满足，那么第9位移入RB8，8位数据移入SBUF。但还需要检测停止位，只有停止位为1，才能置位RI，如果停止位为0，则RI不会置位。

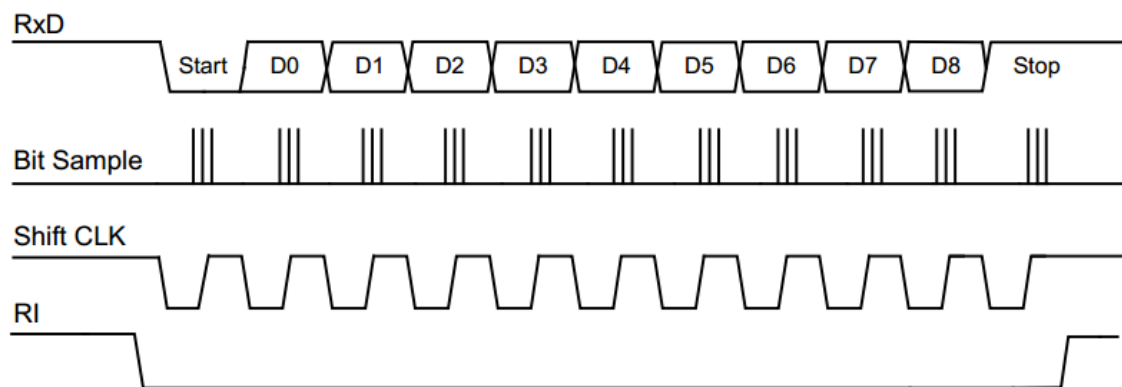


Figure 13-14 Receive Timing of Mode 2

## 13.7.2 UART2 控制寄存器 S2CON、S2CON2

### S2CON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	FE	-		REN	TB8	RB8	TI	RI

位编号	位符号	说明
7	FE	帧错误检测位 0: 无帧错误或软件清 0 1: 有帧错误, 硬件置 1
6-5	-	保留位 (读为 0, 写无效)
4	REN	串行接收使能控制位 0: 禁止串行接收 1: 允许串行接收
3	TB8	方式 1 时, 为要发送的第 9 位数据, 由软件置 1 或清 0
2	RB8	方式 1 时, 为接收到的第 9 位数据, 作为奇偶校验位或地址帧/数据帧的标志位
1	TI	发送中断请求中断标志位 0: 软件清 0 1: 在停止位开始发送时由硬件置 1
0	RI	接收中断请求中断标志位 0: 软件清 0 1: 串行接收到停止位开始时刻由硬件置 1

### S2CON2

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-						SM1	SM2

位编号	位符号	说明
7-2	-	保留位 (读为 0, 写无效)
1	SM1	0: 8 位 UART, 定时器 5 的溢出率/16 1: 9 位 UART, 定时器 5 的溢出率/16 注: 1、UART2 的波特率设置详见定时器 5 方式 1: 波特率发生器章节。 2、UART2 的波特率的配置值可以参考 UART1。
0	SM2	第九位检测使能位 0: 在方式 0 时, 不检测停止位, 停止位无论是 0 还是 1 都会置位 RI 在方式 1 时, 不检测第 9 位, 第 9 位无论是 0 还是 1 都会置位 RI 1: 在方式 0 时, 只有停止位为 1 才能置位 RI 在方式 1 时, 只有第 9 位为 1 才能置位 RI

### 13.7.3 UART2 数据缓冲寄存器 S2BUF

位编号	7	6	5	4	3	2	1	0
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
位符号	S2BUF[7:0]							

位编号	位符号	说明
7-0	S2BUF[7:0]	串口缓冲寄存器：写为需要发送的数据，读为接收到的数据 实际上该地址上有 2 个独立寄存器，一个用于接收数据，一个用于发送数据。当数据写入 S2BUF，这是个发送寄存器并且移位进行串口发送。 当数据从 S2BUF 读出，这是个接收寄存器。



# 14 串行外部设备接口SPI

## 14.1 SPI 特性

- 全双工，三/四线同步传输
- 主从机操作
- 4级可编程主时钟频率
- 极性相位可编程的串行时钟
- 可选择数据传输方向
- 写冲突及接收溢出标志
- 带MCU中断的主模式模式冲突检测
- 带MCU中断的传输结束标志
- 主模式支持高达8Mbps的通信速率（ $F_{osc}=32\text{MHz}$ ），从模式下通信速率须在 $F_{osc}/16$ 及 $F_{osc}/16$ 以下

## 14.2 SPI 信号描述

主输出从输入（MOSI）：该信号连接主设备和一个从设备，数据通过 MOSI 从主设备串行送到从设备，主设备输出，从设备输入。

主输入从输出（MISO）：该信号连接主设备和一个从设备。数据通过 MISO 从从设备串行传入主设备，从设备输出，主设备输入。若该设备为从设备且未被选时，从设备的 MISO 引脚处于高阻状态。

串行时钟（SCK）：该信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动，每 8 个时钟周期 MOSI 和 MISO 线上传送一个字节，如果从设备未被选中，SCK 信号将被此设备忽略。注意：只有主设备才能产生 SCK 信号。

从设备选择引脚（ $\overline{SS}$ ）：每个从属外围设备由一个从选择引脚 $\overline{SS}$ 选择，当引脚信号为低电平时，表明该从设备被选中。主设备可以通过软件控制连接于从设备 $\overline{SS}$ 引脚的端口电平选择每个从设备，很明显，只有一个主设备可以驱动通讯网络。为了防止 MISO 总线冲突，同一时间只允许一个从设备与主设备通讯。在主设备模式中， $\overline{SS}$ 引脚状态关联 SPI 状态寄存器 SPSTAT 中 MODF 标志位以防止多个主设备驱动 MOSI 和 SCK。

下列情况， $\overline{SS}$ 引脚可以作为普通端口或其它功能使用：

（1）设备作为主设备，SPI 控制寄存器 SPCTL 寄存器的 SSIG 位置 1。这种配置仅仅存在于通讯网络中只有一个主设备的情况，因此，SPI 状态寄存器 SPSTA 中 MODF 标志位不会被置 1。

（2）设备配置为从设备，SPI 控制寄存器 SPCTL 的 CPHA 位和 SSIG 位置 1。这种配置情况存在于只有一个主设备一个从设备的通讯网络中，因此，设备总是被选中的，主设备也不需要控制从设备的 $\overline{SS}$ 引脚选择其作为通讯目标。

从设备的 $\overline{SS}$ 引脚被使能时，其它主设备可通过使该引脚维持低电平，从而选中该从设备。为防止 MISO 总线冲突，原则上不允许两个及以上的从设备被选中。

主设备的 $\overline{SS}$ 引脚被使能时，若 $\overline{SS}$ 被拉低将置模式错误标志 MODF（可中断），且 MSTR 位也将被清 0，从而使该设备强制切换到从设备。

当 MSTR = 0（从模式）及 CPHA = 0 时，SSIG 必须为 0，因为此时数据传送需要 $\overline{SS}$ 引脚配合，才能完成多数据传送。

### 14.3 SPI 时钟速率

在主模式下，SPI 的速率有 4 级选择，分别是内部时钟的 4、16、64 或 128 分频，可通过 SPCTL 寄存器的 SPR[1:0]位进行选择。

### 14.4 SPI 功能框图

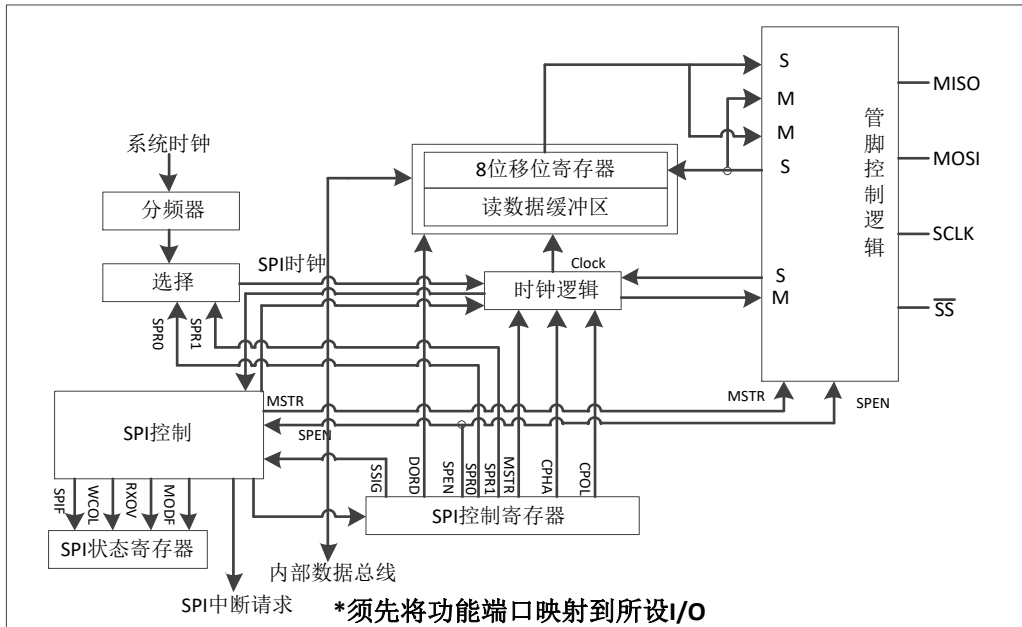


Figure 14-1 SPI 功能方框图

### 14.5 SPI 工作模式

SPI 可配置为主模式或从模式中的一种。SPI 模块的配置和初始化通过设置相关寄存器来完成。进一步设置相关寄存器即可完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出，串行时钟线（SCK）使两条串行数据线（MOSI&MISO）上数据的移动和采样保持同步。从设备选择线（SS）可以独立地选择从属设备；如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为相应，从而实现在同一时钟下数据发送与接收的同步全双工传输。发送移位寄存器和接收寄存器使用相同的 SFR 地址，对 SPI 数据寄存器 SPDAT 进行写操作将写入发送移位寄存器，对 SPDAT 寄存器进行读操作将获得接收移位寄存器的数据。

注意：写入的数据不会影响到需要读出的数据。

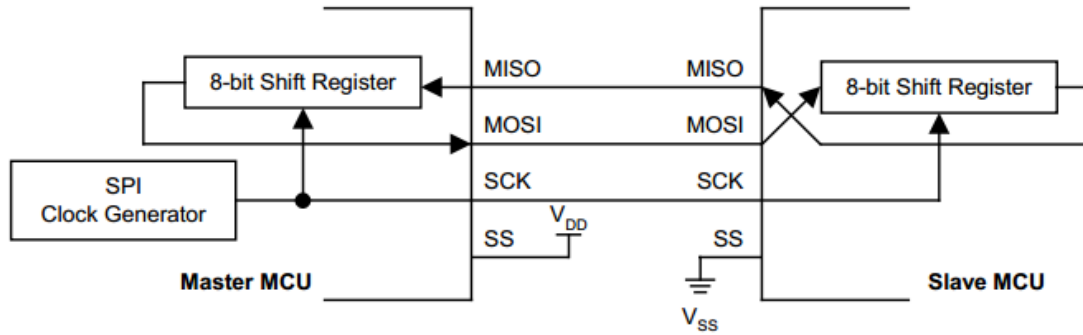


Figure 14-2 全双工主从互联图

### 主模式

#### (1) 模式启动

SPI 主设备控制 SPI 总线上的所有数据传送的启动。一个 SPI 总线中只允许一个主设备可以启动传送。

#### (2) 发送

在 SPI 主模式下，写一个字节数据到 SPI 数据寄存器 SPDAT，数据将会写入发送移位缓冲器。如果发送移位寄存器中已经存在一个数据或正在传送一个数据，那么主 SPI 将产生一个 WCOL 信号以表明写入太快。但是发送移位寄存器中的数据不会受到影响，发送也不会中断。

#### (3) 接收

当主设备通过 MOSI 线传送数据到从设备时，同时对应的从设备也可以通过 MISO 线将其发送移位寄存器的数据传送给主设备的接收移位寄存器，实现全双工操作。故 SPIF 标志置 1 即表示数据发送完成也表示数据接收完成。本 SPI 模块接收为双缓冲器，即数据可以在 SPIF 置 1 后读出，但必须在下一字节数据接收完成前读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

### 从模式

#### (1) 模式启动

将 MSTR 置 0（若  $\overline{SS}$  被使能则必须拉低）时，设备处于从模式下运行，数据传送过程中设备模式不能改变（ $\overline{SS}$  引脚必须维持低电平），否则数据传送将失败（SPIF 不会被置 1）。

#### (2) 发送

SPI 从设备下不能启动数据传送，所以 SPI 从设备必须在主设备开始一次新的数据传送之前将要传送给主设备的数据写入发送移位寄存器。若发送前未写入数据到发送移位寄存器，从设备将传送数据“0x00”给主设备。若写入数据时发送移位寄存器已经存在数据（或发生在传送过程中），那么 SPI 从设备的 WCOL 标志位将置 1，表示发生写 SPDAT 冲突。但是移位寄存器的数据不受影响，传送也不会被中断，传送完成 SPIF 将被置 1。

#### (3) 接收

从模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引进移入，当计数器计数 SCK 边缘数到 8 时，表示一个字节数据接收完毕，SPIF 将置 1，数据可以通过此时读取 SPDAT 寄存器获得，但必须在下一数据接收完成前被读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

## 14.6 SPI 传送形式

通过软件设置寄存器的 CPOL 位和 CPHA 位,用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性,即空闲时的电平状态。CPHA 位定义时钟相位,即定义允许数据移位采样的时钟边沿。在通信的两个主从设备中,时钟极性相位设置应当保持一致。

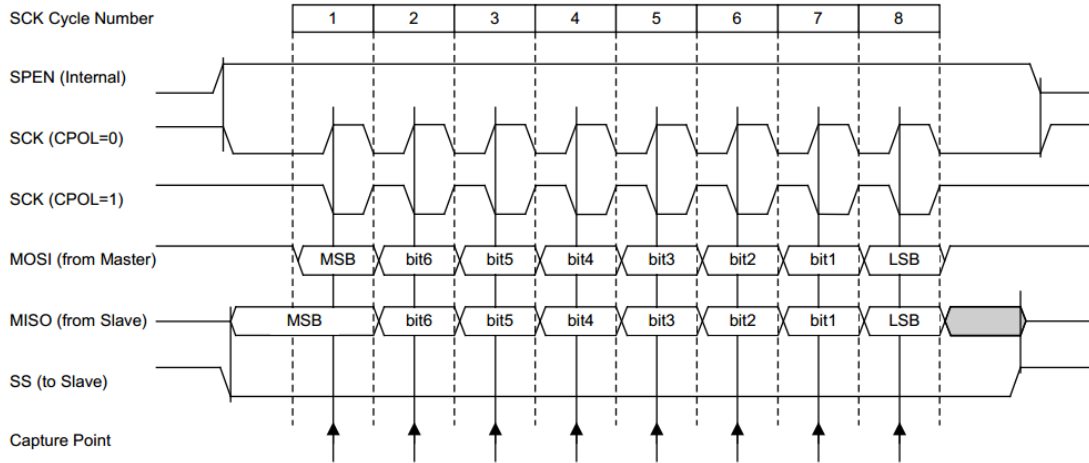


Figure 14-3 数据传送形式 (CPHA=0)

如果 CPHA = 0; 数据在 SCK 的第一沿就被捕获, 所以从设备必须在 SCK 的第一个沿之前就准备好数据, 因此,  $\overline{SS}$  引脚的下降沿从设备就开始数据。 $\overline{SS}$  引脚在每次传送完一个字节后必须拉高, 在发送下一字节之前重新又被拉低, 故 CPHA = 0 时, SSIG 位无效, 即  $\overline{SS}$  脚被强制使能。

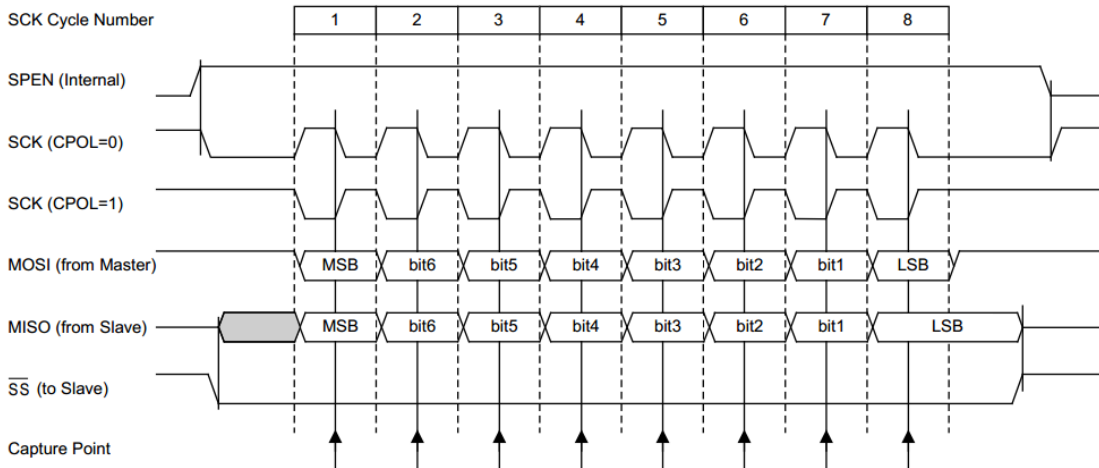


Figure 14-4 数据发送形式 (CPHA=1)

如果 CPHA = 1, 主设备在 SCK 的第一个沿将数据输出到 MOSI 线上, 从设备把 SCK 的第一个沿作为开始发送信号。用户必须在第一个 SCK 的前 2 个沿内完成对 SPDAT 完成写操作。传送过程中彼此模式不能改变, 否则数据发送接收将失败, 模式被改变的寄存器数据 (发送数据) 及状态 (接收为空) 不变。这种数据传送形式为单一主从设备间通信的首先形式。

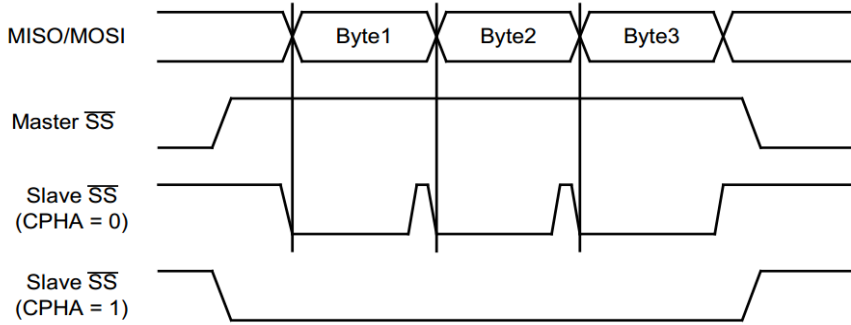


Figure 14-5 CPHA/ $\overline{SS}$ 时序

## 14.7 SPI 出错检测

SPSTA 寄存器中的一些标志位表示 SPI 通信中的通信错误情况：

### (1) 模式故障 (MODF)

SPI 主模式下的模式故障出错表明 $\overline{SS}$ 引脚上的电平状态与实际设备模式不一致，MODF 标志位将被置 1（可产生中断），以此来表明 SPI 控制系统中存在多主设备的冲突情况，此时硬件将自动清除 SPEN 位，即先关闭 SPI 模块；同时硬件也将自动清除 MSTR 位。需要重启 SPI 模块时，MODF 必须先软件写 1 清 0，再置 SPEN 位。

### (2) 写冲突 (WCOL)

在数据未发送或发送期间继续对 SPDAT 做写入操作会引起写冲突，WCOL 位会被置 1，但发送不会终止。需软件写 1 清 0

### (3) 接收溢出 (RXOV)

在接收第二数据完成前仍未清除之前接收数据产生的 SPIF 标志，将置接收溢出标志 RXOV，SPIF 被置 1 时，后面的数据将不会被传入接收寄存器，故接收的数据存入 SPDAT 前必须清除 SPIF，RXOV 位需软件写 1 清 0。

## 14.8 SPI 中断

两种 SPI 状态标志 SPIF&MODF 都能产生一个 CPU 中断请求。

串行数据传输完成标志 SPIF：完成一个字节数据发送/接收后由硬件置 1。

故障模式标志 MODF：该位被置 1 是指设备模式（主机）与 $\overline{SS}$ 引脚电平不一致，SSIG 位为 1（ $\overline{SS}$ 未被使能）时，无 MODF 中断请求。

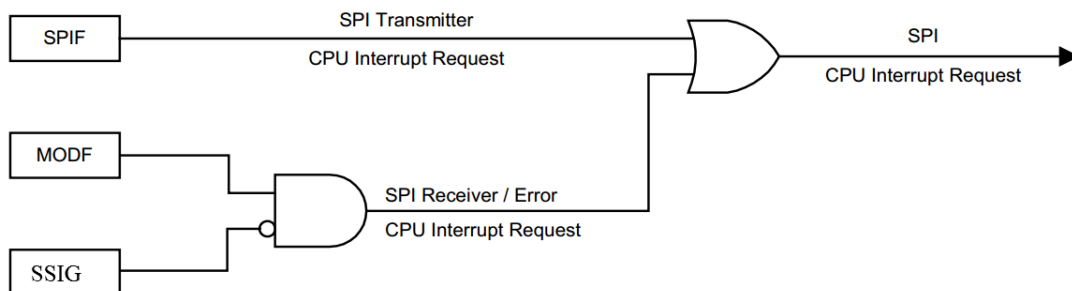


Figure 14-6 SPI 中断请求的产生

## 14.9 SPI 配置对照

SPEN	SSIG	$\overline{SS}$	MSTR	主或从模式	MISO	MOSI	SCK	备注
0	x	I/O	x	SPI功能禁止	I/O	I/O	I/O	SPI禁止
1	0	0	0	从机模式	输出	输入	输入	选择从机
1	0	1	0	从机模式未被选中	高阻	输入	输入	未被选中。MISO为高阻，以避免总线冲突
1→0	0	0	1→0	关闭SPI	输出	输入	输入	SS配置为输入，SSIG为0。如果SS被驱动为低电平。则被选择作为从机。此时MSTR将清零，并置模式错误标志MODE，可用于请求中断。
1	0	1	1	主（空闲）	输入	高阻	高阻	当主机空闲时MOSI和SCK为高阻态以避免总线冲突。用户必须将SCK上拉或下拉（根据CPOL的取值）以避免SCK出现悬浮状态。
				主（激活）		输出	输出	作为主机激活时，MOSI和SCK为推挽输出。
1	1	I/O	0	从	输出	输入	输入	CPHA不能为0
1	1	I/O	1	主	输入	输出	输出	-

## 14.10 SPI 相关寄存器

### 14.10.1 SPI 控制寄存器 SPCTL

位编号	7	6	5	4	3	2	1	0
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
位符号	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]	

位编号	位符号	说明
7	SSIG	$\overline{SS}$ 脚使能位 0: $\overline{SS}$ 脚被使能用于确定器件为主机还是从机 1: MSTR 确定器件为主机还是从机, $\overline{SS}$ 脚作为普通 I/O 使用
6	SPEN	SPI 使能位 0: 禁止 SPI 模块, 相关管脚为普通 I/O(建议 I/O 时设为高阻) 1: 使能 SPI 模块, 相关管脚为 SPI 通信管脚
5	DORD	传送方向选择位 0: MSB先发送 1: LSB先发送
4	MSTR	主/从机模式选择位 0: 从机模式 1: 主机模式
3	CPOL	SPI 时钟极性选择位 0: SCK 空闲时为低电平 1: SCK 空闲时为高电平
2	CPHA	SPI 时钟相位选择位 0: 数据在 SPI 时钟的第一个边沿采样 1: 数据在 SPI 时钟的第二个边沿采样 注: SSIG = 0&CPHA = 0 时, 数据在 $\overline{SS}$ 为低被驱动; CPHA = 1 时, 数据在 SCK 的前时钟沿驱动。
1-0	SPR[1:0]	SPI 时钟速率选择控制位 00: $F_{osc} / 4$ 01: $F_{osc} / 16$ 10: $F_{osc} / 64$ 11: $F_{osc} / 128$

### 14.10.2 SPI 状态寄存器 SPSTAT

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	SPIF	WCOL	RXOV	MODF	-			

位编号	位符号	说明
7	SPIF	SPI 传输完成标志位 0: 软件写 1 清 0 1: 一次传送完成时, 硬件置 1, 也做中断请求标志位
6	WCOL	SPI 写冲突标志位 0: 软件写 1 清 0 1: 传送过程中对 SPDAT 执行写操作硬件置 1 (正在传送的数据不受影响)
5	RXOV	SPI 接收溢出标志位 0: 软件写 1 清 0 1: 发生接收溢出, 硬件置 1 注意: 接收为双 BUFF, 接收溢出发生在第二个数据传送完成前仍未清除之前接收数据产生的 SPIF 标志, 故每次准备接收下一个数据前必须先清除 SPIF, 否则 RXOV 将置 1, RXOV 置 1 不会影响 SPI 接收。
4	MODF	模式故障标志位 0: 软件写 1 清 0 1: SS 引脚电平与 SPI 模式不一致时, 硬件置 1 (且立即切成从机模式), 也做中断请求标志位
3-0	-	保留位 (读为 0, 写无效)

### 14.10.3 SPI 数据寄存器 SPDAT

位编号	7	6	5	4	3	2	1	0
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
位符号	SPDAT[7:0]							

位编号	位符号	说明
7-0	SPDAT[7:0]	SPI 数据寄存器



# 15 IIC总线

## 15.1 IIC 特性

- 双线通信
- 支持主机模式及从机模式
- 支持多主机通信时仲裁功能
- 支持地址可编程
- 支持标准速率（最多 100kbps）和快速（最多 400kbps）

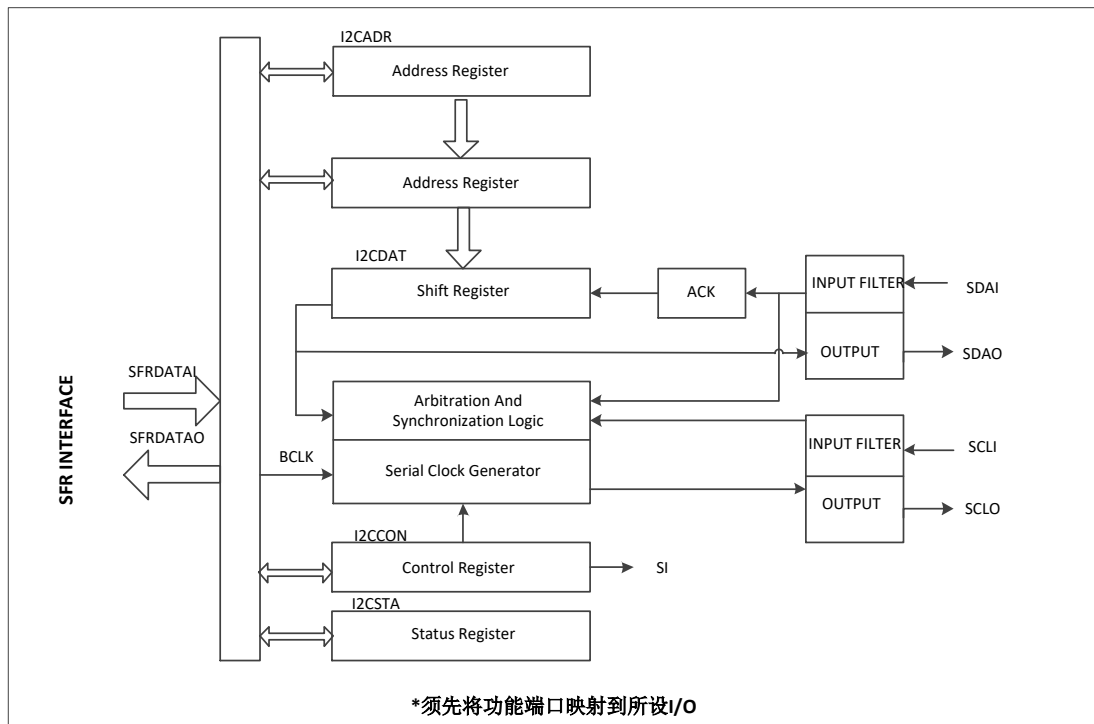


Figure 15-1 IIC 功能框图

## 15.2 IIC 总线工作原理

物理结构上，IIC 系统由一条串行数据线 SDA 和一条串行时钟线 SCL 组成。主机按一定的通信协议向从机寻址和进行信息传输，在数据传输时，由主机初始化一次数据传输，主机使数据在 SDA 线上传输的同时还通过 SCL 线传输时钟。信息传输的对象和方向以及信息传输的开始和终止均由主机决定。

每个器件都有一个唯一的地址，而且可以是单接收的器件或者可以接收也可以发送的器件。发送器或接收器可以在主模式或从模式下操作，这取决于芯片是否必须启动数据的传输还是仅仅被寻址。

下图是最常用、最典型的 IIC 总线连接方式。

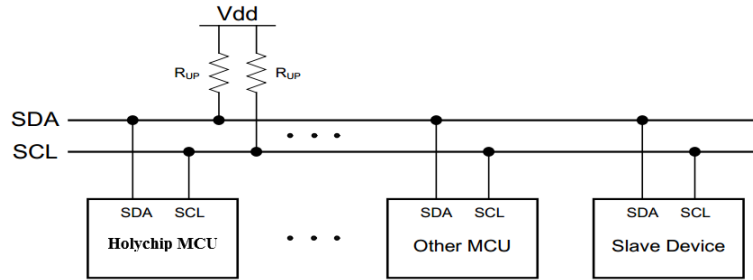


Figure 15-2 IIC 总线连接图

## 15.3 总线上数据的有效性

IIC 总线是以串行方式传输数据，从数据字节的最高位开始传送，每一个数据位在 SCL 上都有一个时钟脉冲相对应。在时钟线高电平期间数据线上必须保持稳定的逻辑电平状态，高电平为数据 1，低电平为数据 0。只有在时钟线为低电平时，才允许数据线上的电平状态变化，如 Figure15-3 所示。

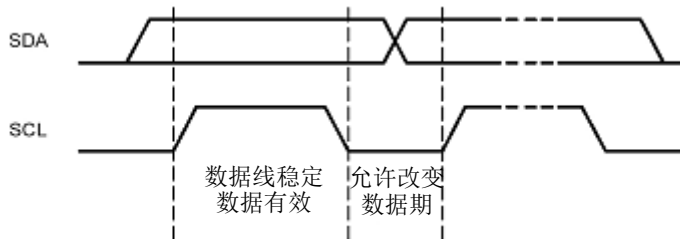


Figure 15-3 IIC 总线上数据的有效性

## 15.4 总线上的信号

IIC 总线在传送数据过程中共有四种类型信号，它们分别是：开始信号、停止信号、重新开始信号和应答信号。

**开始信号 (START):** 如 Figure 15-4 所示，当 SCL 为高电平时，SDA 由高电平向低电平跳变，产生开始信号。当总线空闲的时候，例如，没有主动设备在使用总线 (SDA 和 SCL 都处于高电平)，主机通过发送开始 (START) 信号建立通信。

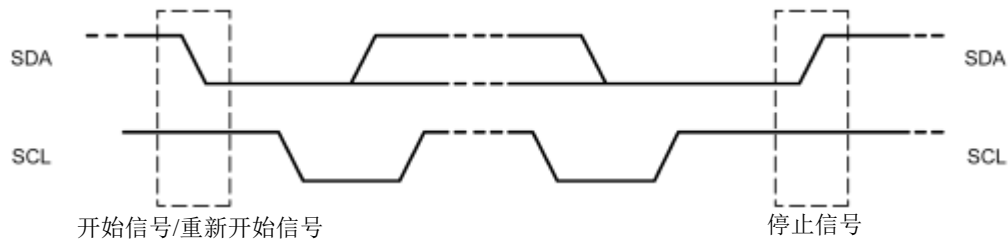


Figure 15-4 开始、重新开始、停止信号

**停止信号 (STOP):** 如 Figure15-4 所示，当 SCL 为高电平时，SDA 由低电平向高电平跳变，产生停止信号。主机通过发送停止信号，结束数据通信。

**重新开始信号 (Repeated START):** 在 IIC 总线上，由主机发送一个开始信号启动一次通信后，在首次发送停止信号之前，主机通过发送重新开始信号，可以转换与当前从机的通信模式，或是切换到与另一个从机通信。如 Figure15-5 所示，当 SCL 为高电平时，SDA 由高电平向低电平跳变，产生重新开始信号，它的本质就是一个开始信号。

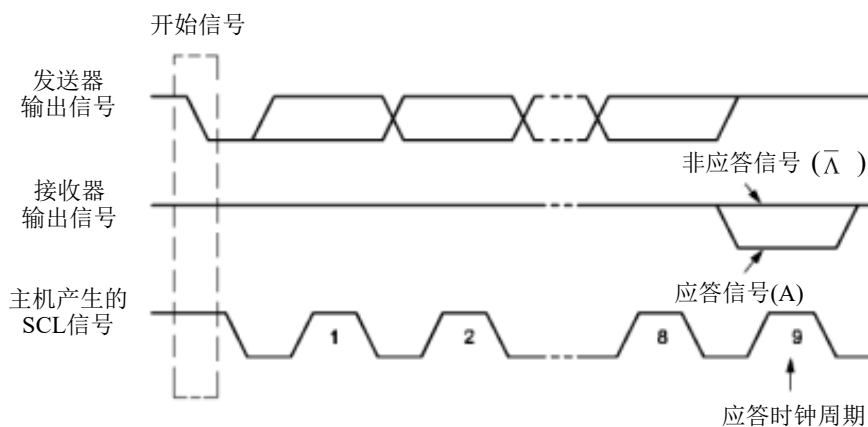


Figure 15-5 IIC 总线的应答信号

**应答信号 (A):** 接收数据的 IIC 在接收到 8 位数据后，向发送数据的 IC 发出的特定的低电平脉冲。每一个数据字节后面都要跟一位应答信号，表示已收到数据。应答信号在第 9 个时钟周期出现，这时发送器必须在这一时钟位上释放数据线，由接收设备拉低 SDA 电平来产生应答信号，由接收设备保持 SDA 的高电平来产生非应答信号 ( $\bar{A}$ )，如 Figure15-5 所示。所以，一个完整的字节数据传输需要 9 个时钟脉冲。如果从机作为接收方向主机发送非应答信号，这样，主机方就认为此次数据传输失败；如果是主机作为接收方，在从机发送器发送完一个字节数据后，发送了非应答信号，从机就认为数据传输结束，并释放 SDA 线。不论是以上哪种情况都会终止数据传输，这时，主机或是产生停止信号释放总线，或是产生重新开始信号，开始一次新的通信。开始信号、重新开始信号和停止信号都是由主控制器产生，应答信号由接收器产生，总线上带有 IIC 总线接口的器件很容易检测到这些信号。

## 15.5 总线上数据初始格式

一般情况下，一个标准的 IIC 通信由四部分组成：开始信号、从机地址传输、数据传输、停止信号。

由主机发送一个开始信号，启动一次 IIC 通信；在主机对从机寻址后，再在总线上传输数据。IIC 总线上传送的每一个字节均为 8 位，首先发送的数据位为最高位，每传送一个字节后都必须跟随一个应答位，每次通信的数据字节数是没有限制的；在全部数据传送结束后，由主机发送停止信号，结束通信。

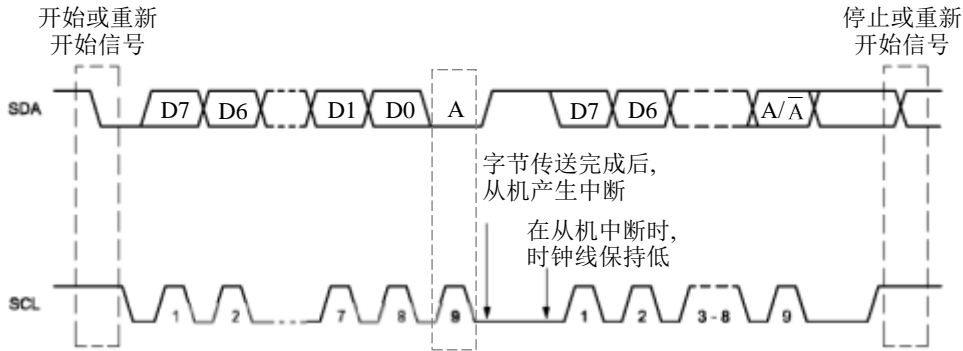


Figure 15-6 IIC 总线的数据传输格式

如 Figure 15-6 所示，时钟线为低电平时数据传输将停止进行。这种情况可以用于当接收器接收到一个字节数据后要要进行一些其它工作而无法立即接收下一个数据时，迫使总线进入等待状态，直到接收器准备好接收新数据时，接收器再释放时钟线使数据传输得以继续正常进行。例如，当接收器接收完主控制器的一个字节数据后，产生中断信号并进行中断处理，中断处理完毕才能接收下一个字节数据，这时接收器在中断处理时将钳住 SCL 为低电平，直到中断处理完毕才释放 SCL。

## 15.6 IIC 总线寻址约定

IIC 总线系统中挂接的所有外围器件，一般均拥有一个专用的 7 位从器件地址码。由于 7 位从器件地址码，其编码空间最多只有 128 个，后来在原有的 7 位地址码格式基础上，又发展了 10 位地址码格式。10 位地址格式仍然符合总线协议。

“广播呼叫”是个例外，它可以通过将第一个字节的数据全部赋值为 0 来寻址所有器件。广播呼叫用于当主机希望发送相同信息到几个从机时情况。当该地址在使用时，其他器件根据软件配置可能响应应答或忽略。如果器件响应广播呼叫，其操作就像从机接收器模式。

## 15.7 主机向从机读写 1 个字节数据的过程

如 Figure 15-7 所示，主机要向从机写 1 个字节数据时，主机首先产生 START 信号，然后紧跟着发送一个从机地址，这个地址共有 7 位，紧接着的第 8 位是数据方向位(R/W)，0 表示主机发送数据(写)，1 表示主机接收数据(读)，这时候主机等待从机的应答信号(A)，当主机收到应答信号时，发送要访问的地址，继续等待从机的应答信号，当主机收到应答信号时，发送 1 个字节的数据，继续等待从机的应答信号，当主机收到应答信号时，产生停止信号，结束传送过程。

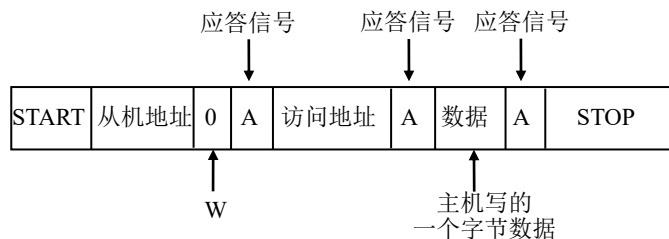


Figure 15-7 主机向从机写数据

如 Figure 15-8 所示，主机要从从机读 1 个字节数据时，主机首先产生 START 信号，然后紧跟着发送一个从机地址，注意此时该地址的第 8 位为 0，表明是向从机写命令，这时候主机等待从机的应答信号(A)，当主机收到应答信号时，发送要访问的地址，继续等待从机的应答信号，当主机收到应答信号后，主机要改变通信模式(主机将由发送变为接收，从机将由接收变为发送)所以主机发送重新开始信号，然后紧跟着发送一个从机地址，注意此时该地址的第 8 位为 1，表明将主机设置成接收模式开始读取数据，这时候主机等待从机的应答信号，当主机收到应答信号时，就可以接收 1 个字节的数据，当接收完成后，主机发送非应答信号，表示不在接收数据，主机进而产生停止信号，结束传送过程。

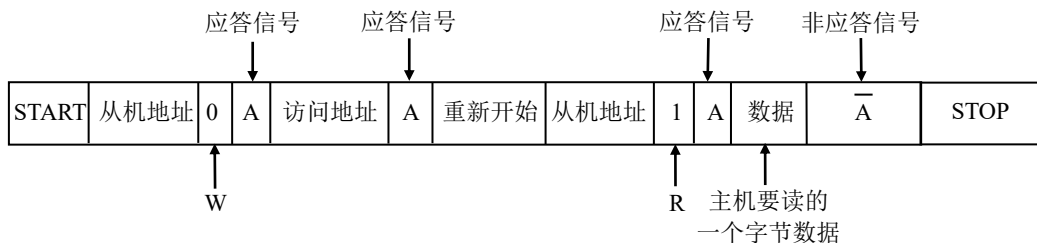


Figure 15-8 主机向从机读写数据 1 个字节数据

## 15.8 IIC 工作模式

### 15.8.1 主机发送模式

在主机发送模式下，向从机接收器发送几个数据字节。主机通过 CR[2:0]设置期望时钟速率并向 IICEN 位写 1 使能 IIC 总线，设置 STA 位为 1 进入主机发送模式，只要总线空闲，硬件将测试总线并产生起始信号，成功产生起始信号后，SI 标志位将置位且 IICSTA 的状态码为 08H，之后就是给 IICDAT 载入目标从机地址和数据方向“写”（SLA+W），SLA+W 开始传输时 SI 位必须清零。

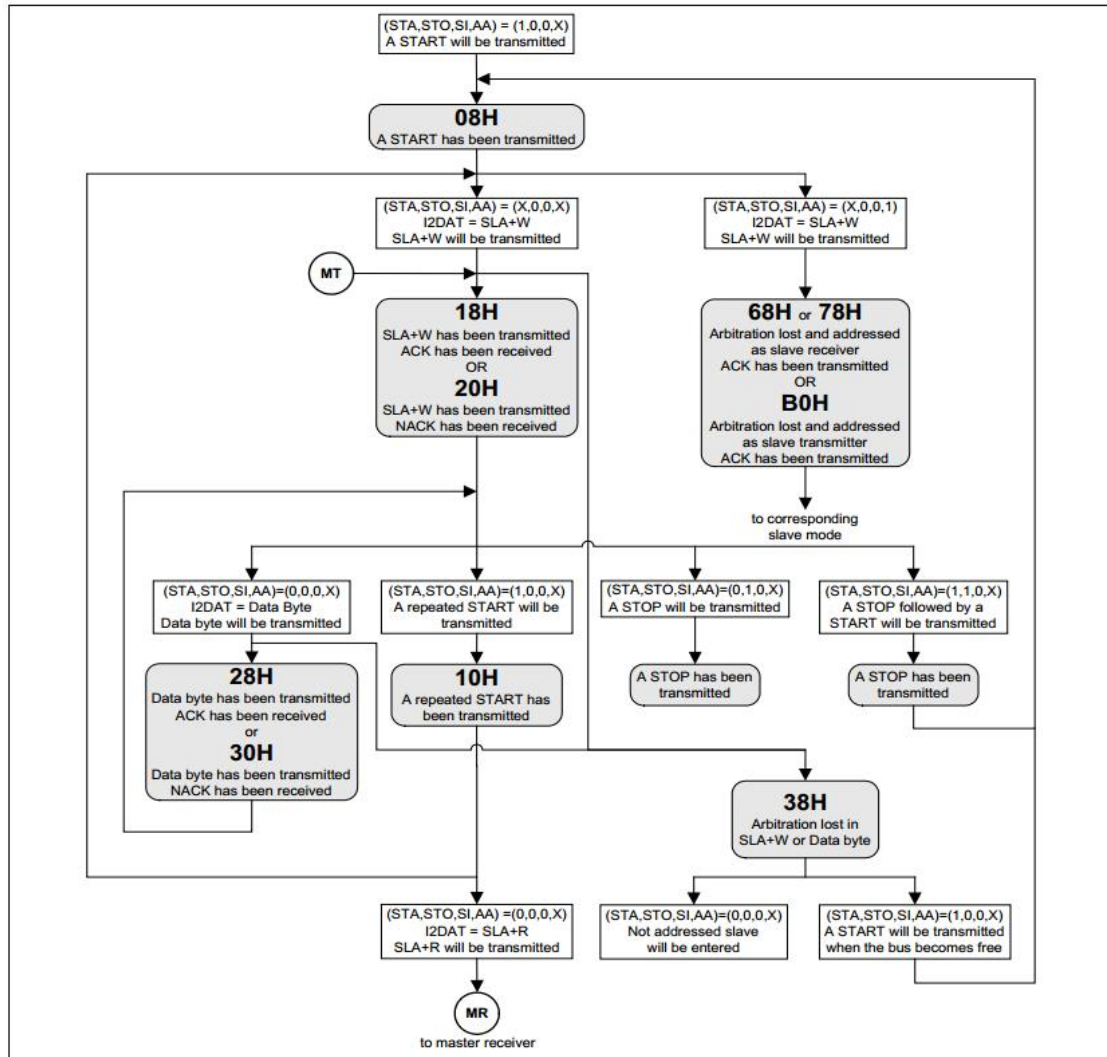


Figure 15-9 主机发送模式流程与状态

### 15.8.2 主机接收模式

在主机接收模式下，从从机发送器接收几个字节的数据。传输开始与主机发送模式相似，在起始信号之后，IICDAT 应该加载目标从机地址和数据方向位“读”（SLA+R），SLA+R 字节发送后，且返回应答位，重新置位 SI 标志且 IICSTA 读出为 40H，SI 标志应该被清零以便接收从机发送过来的数据，如果 AA 标志位置位，主机接收器将应答从机发送器，如果清零 AA，主机接收器将不会应答从机，并释放从机发送器为不被寻址的从机，然后主机产生停止信号或重复起始信号中止传输或开始另一次传输。

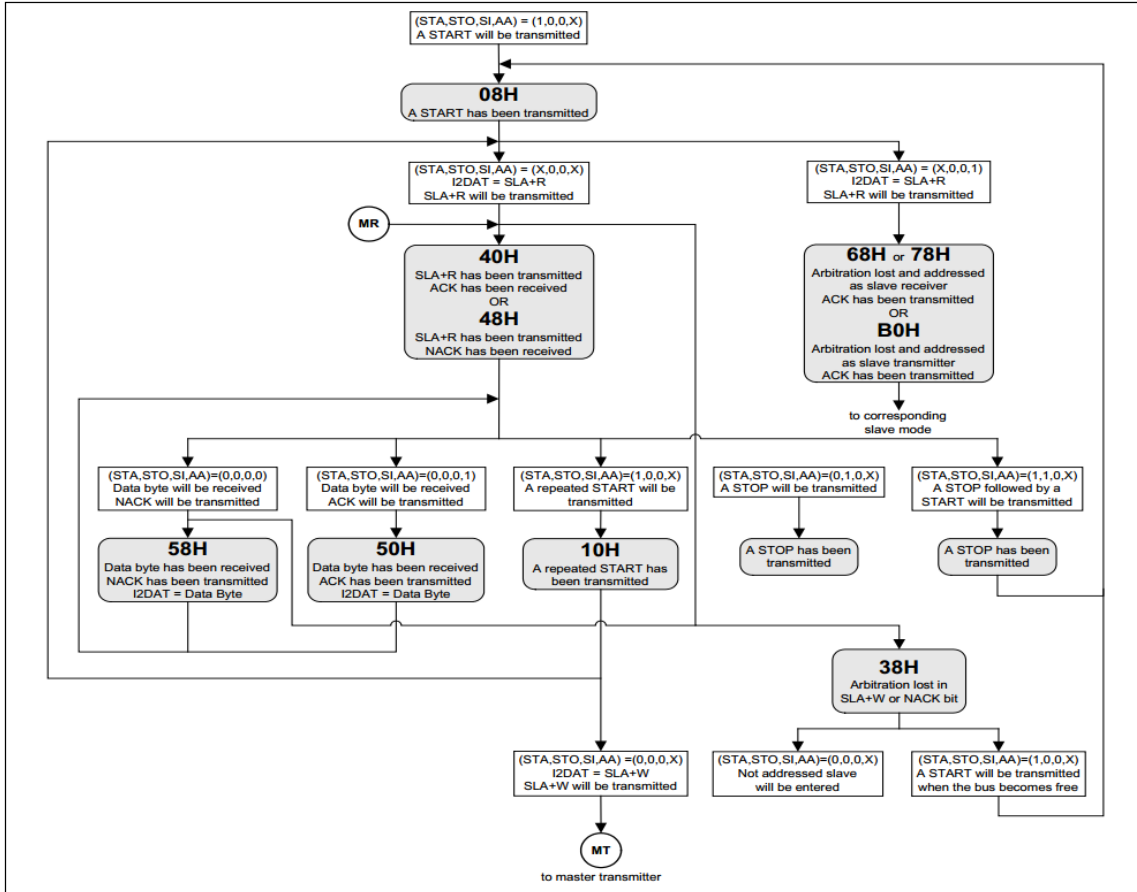


Figure 15-10 主机接收模式流程与状态

### 15.8.3 从机发送模式

在从机发送模式下，发送几个字节数据到主机接收器。确定 IICADR 和 IICCON 的值之后，IIC 等待自己的地址被寻址“读”（SLA+R）。如果仲裁失败后，也可以进入从机发送模式。

在从机被 SLA+W 寻址后，应该清 SI 标志以便传输数据到主机发送器，通常主机接收器将在从机发送每个字节数据之后返回应答，如果没有接收到应答，如果继续传输将发送全“1”，就成为不被寻址的从机，如果在传输中清了 AA 标志，从机发送最后一个字节数据，下一次传输数据全为“1”，从机成为不被寻址。

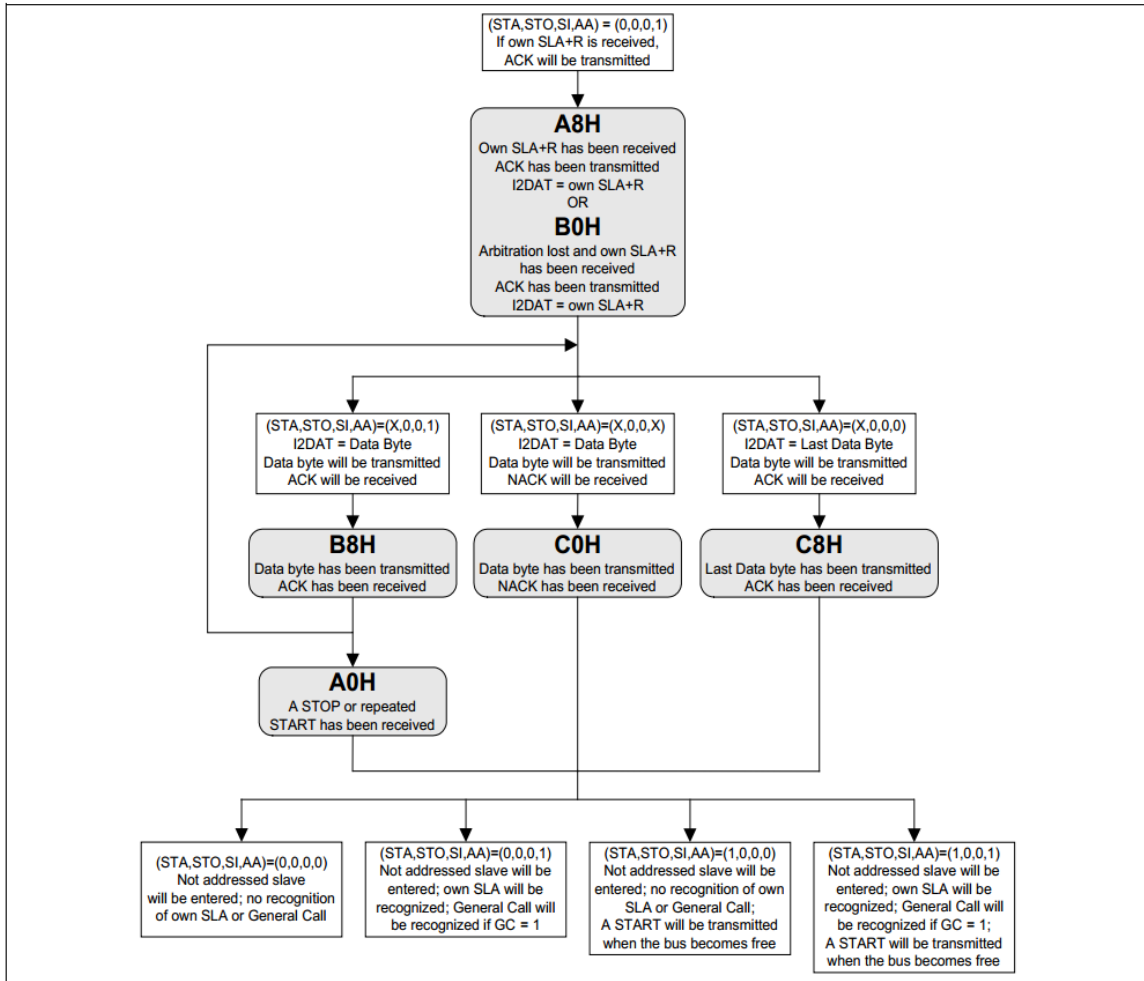


Figure 15-11 从机发送模式流程与状态



### 15.8.4 从机接收模式

在从机接收模式下，从主机发送器接收几个字节数据。发送开始之前，IICADR 必须装载响应器件的地址，以让主机寻址，AA 位必须设置使能应答自身从机地址或广播呼叫，完成以上初始过程后，IIC 等待自身地址被寻址与数据方向位“写”(SLA+W)或被广播呼叫寻址。如果在仲裁失败时，也可以进入从机接收模式。

在从机被 SLA+W 寻址后，应该清 SI 标志以便接收主机发送过来的数据，传输期间，如果 AA 位为 0，从机将在下一次接收到的数据字节之后返回无应答 (non-acknowledge)，从机也不被寻址并与主机分离，不能接收 IICDAT 的任何字节，而保持当前接收到的数据字节。

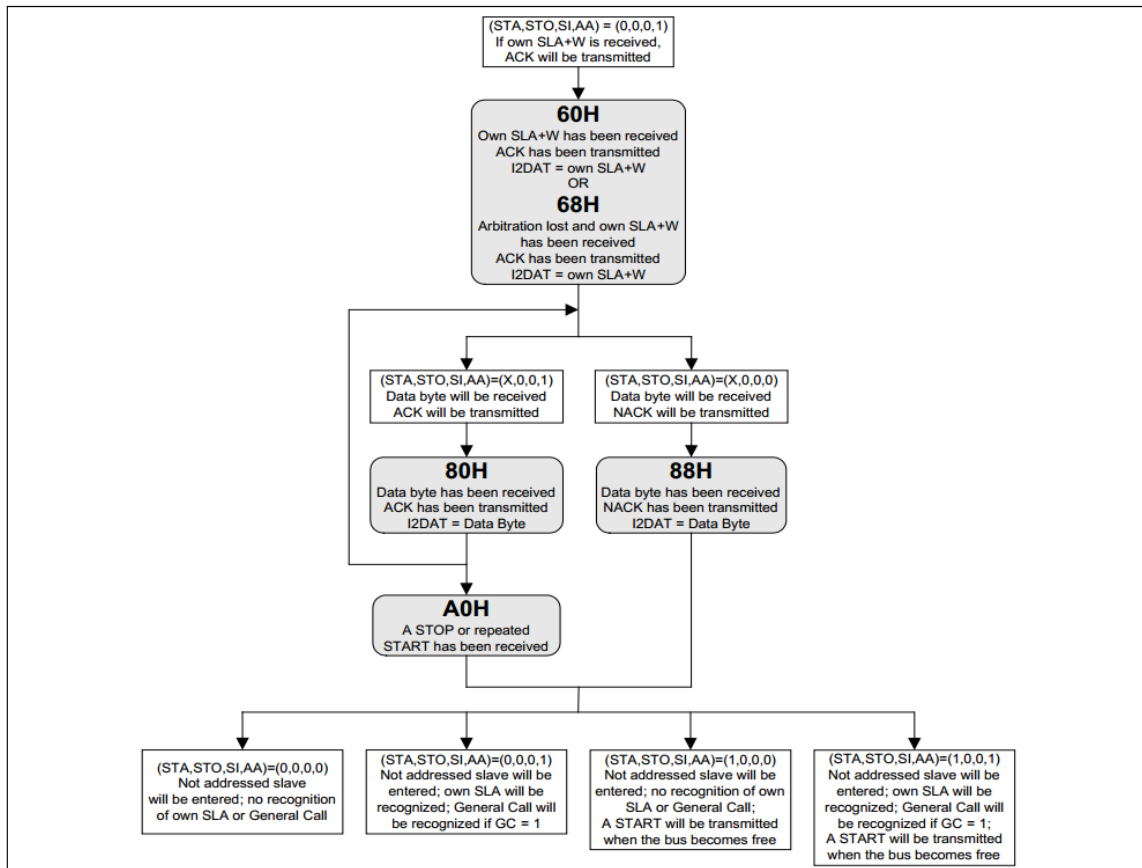


Figure 15-12 从机接收模式流程与状态

### 15.8.5 广播呼叫

广播呼叫是从机接收模式的一种特殊情况，即从机地址和数据方向位全为 0，被广播呼叫寻址的从机在正常从机接收模式的 IICSTA 里有不同状态码，如果仲裁失败，也可以产生广播呼叫。

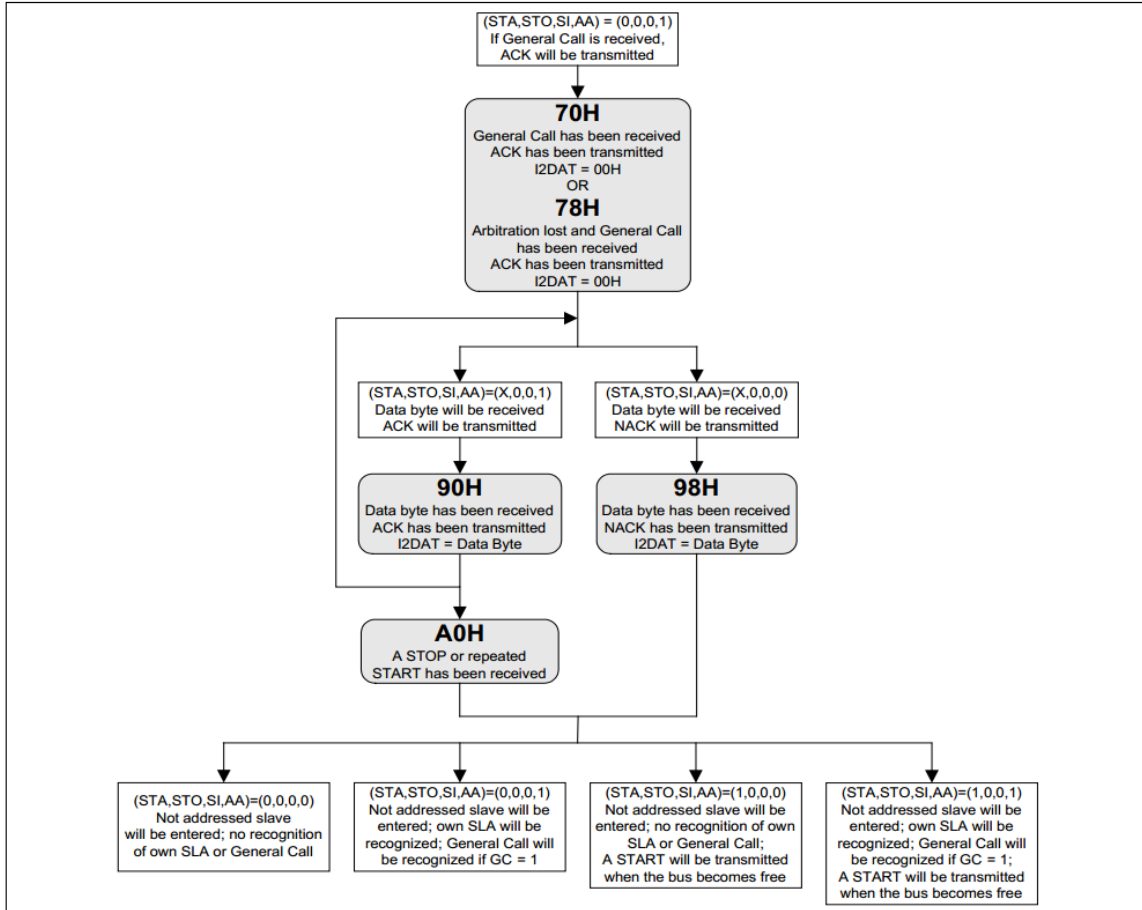


Figure 15-13 广播呼叫模式流程与状态

### 15.8.6 其他状态

有两个 IICSTA 状态码与 24 个定义状态不一致，即前面提到的 0F8H 和 00H 状态。

第一个状态码 0F8H 表示在每次传输期间没有得到相关信息，同时，SI 标志为 0 且没有 IIC 中断请求。

另一个标志码 00H 意味在传输过程中发生错误，总线错误是由 START 或停止信号暂时出现在一个非法的位置，如地地址字节里第 2 位换到第 8 位，或数据字节包括应答位，当出现总线错误时，SI 标志立即置位，当在 IIC 总线上检测到总线错误，工作器件立即切换到不被寻址从机模式，释放 SDA 和 SCL 总，置位 SI 标志，将 00H 载入 IICSTA。要从总线错误恢复，STO 位必须设置为逻辑 1 且 SI 必须清零，然后，STO 由硬件清零且在停止信号就释放 IIC 总线。

特例：如果没有成功产生 START 或重复起始信号，IIC 总线被 SDA 的低电平阻挡，如一个从 CPU 时钟件没有位同步，可以通过在 SCL 总线上发送额外时钟脉冲解决这个问题。当 STA 位置位时，IIC 硬件发送额外时钟脉冲，但是由于 SDA 被拉低，不能产生起始信号，当 SDA 总线最终被释放，发送一个普通的 START 条件，进入状态 08H，继续进行串行传输。当 SDA 为低，如果发送重复起始信号，IIC 硬件也执行以上相同的动作。此情况下，在成功发送起始信号后，进入状态 08H，而不是进入 10H。

注：软件不能解决这类总线问题。

## 15.9 IIC 总线相关寄存器

### 15.9.1 IIC 控制寄存器 IICCON

位编号	7	6	5	4	3	2	1	0
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
位符号	CR2	IICEN	STA	STO	SI	AA	CR1	CR0

位编号	位符号	说明
7	CR2	IIC 通信时钟选择位 2
6	IICEN	IIC 模块使能位 0: 禁止 IIC 模块 1: 启动 IIC 模块
5	STA	起始位 0: 不发送起始信号 1: 总线空闲时产生起始信号。忙时, 等待停止信号后产生一个起始信号。主机模式下, IIC 准备好发送或接收一个或多个字节时, 置 1 产生一个重复的起始信号
4	STO	停止位 0: 不发送停止信号 1: 主机模式时产生停止信号, 当检测到总线上出现停止信号。IIC 硬件清除 STO 标志。STO 标示的设置也用于将 IIC 设备从错误状态 (IICSTA 为 00H) 恢复, 此条件下, 没有停止信号发送 IIC 总线上。若 STA 和 STO 都置 1, 且在主机模式下设备为原始的, IIC 总线将产生停止信号并立即伴随着起始信号。如果设备为从机模式, 置 STO 恢复到非寻址从机, STO 将会硬件清 0。
3	SI	IIC 串行中断标志位 0: 没有 IIC 串行中断发生 1: 产生 IIC 通信状态码中除 0F8H 之外的状态码时置 1。必须软件清 0
2	AA	应答标志位 0: 回复 NACK (SDA 上为高电平) 1: 回复 ACK (SDA 上为低电平)
1	CR1	IIC 通信时钟选择位 1
0	CR0	IIC通信时钟选择位0

CR[2:0] IIC 通信时钟选择位:

CR2	CR1	CR0	F <sub>osc</sub>				分频系数	
			6MHz	12 MHz	16 MHz	24 MHz		
0	0	0	23KHz	47KHz	63KHz	92KHz	256	
0	0	1	27KHz	54KHz	71KHz	108KHz	224	
0	1	0	31KHz	63KHz	83KHz	124KHz	192	
0	1	1	37KHz	75KHz	100KHz	148KHz	160	
1	0	0	6.25KHz	12.5KHz	17KHz	25KHz	960	
1	0	1	50KHz	100KHz	133KHz	200KHz	120	
1	1	0	100KHz	200KHz	266KHz	400KHz	60	
1	1	1	T5 溢出率/8 (需要配置定时器 5 为波特率发生器)					

### 15.9.2 IIC 状态寄存器 IICSTA

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
复位值	1	1	1	1	1	0	0	0
位符号	IICSTA[7:3]						-	

位编号	位符号	说明
7-3	IICSTA[7:3]	IIC 状态码, 共有 26 个可能的状态码, 状态码中除 0F8H 外都可置 SI 标志
2-0	-	保留位

### 15.9.3 IIC 数据寄存器 IICDAT

位编号	7	6	5	4	3	2	1	0
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
位符号	IICDAT[7:0]							

位编号	位符号	说明
7:0	IICDAT[7:0]	IIC 数据 IICDAT 包含一个字节的将被发送或刚接收到的 IIC 数据。只要 SI 为逻辑 1, IICDAT 中的数据保持不变, 在 IIC 发送接收过程中, 读或写 IICDAT 的结果都是不确定的。 当 IICDAT 的数据被移出, 总线上的数据同步被移入以更新 IICDAT。IICDAT 常显示当前 IIC 总线上的最后字节。因此失去仲裁, 在传输之后的 IICDAT 原始值被改变。

### 15.9.4 IIC 地址寄存器 IICADR

位编号	7	6	5	4	3	2	1	0
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
位符号	IICADR[7:1]							GC

位编号	位符号	说明
7-1	IICADR[7:1]	从机模式：IIC 设备自身从机地址 主机模式：该数据无影响
0	GC	广播呼叫位 0：广播呼叫常被忽略 1：如果 AA 标志为 1，广播呼叫被识别；如果 AA 为 0，忽略广播呼叫 注：该位只在从机模式有效，主机模式无影响。做从机时，置 AA 标志，在空闲模式下，若总线其它主机寻址地址与本从机地址匹配，则将唤醒本从机。

# 16 模数转换ADC

## 16.1 ADC 特性

- 多达 16 个外部通道及 2 个内部通道（包括 GND）的 12/10 位 ADC 检测
- 参考电压可选内部 1.3V、2V、3V、4V、VDD 及外部 VREF
- 支持外部触发 ADC
- 可选择转换数据对齐方向
- 可选择转换数据位数
- ADC 转换完成可中断
- 单通道（P0.2 端口）ADC 唤醒中断

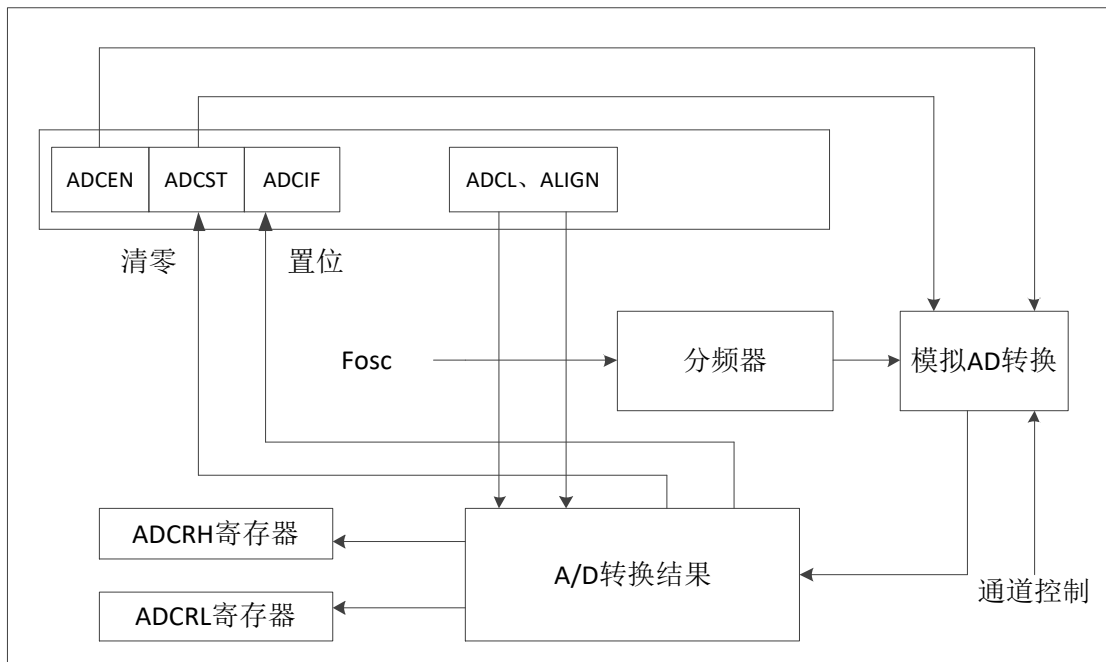


Figure 16-1 ADC 功能框图

## 16.2 ADC 省电唤醒

在芯片进入省电模式（IDLE 或 PD）后，可通过使能 ADC 省电唤醒功能将芯片从省电模式唤醒，这个功能可以实现用一个 IO 口检测多个按键。

具体操作如下：

1. 通过配置 ADC 唤醒控制寄存器 ADCWC 允许唤醒；
2. 将 P0.2 配置为模拟通道；
3. 通过 P0.2 端口上拉电阻选择寄存器配置唤醒电阻（比如配置 P0.2 上拉电阻为 50 K $\Omega$ ）；
4. 使能 IE 和 IE1 寄存器里的 EA 和 EADC；
5. 进入省电模式（IDLE 或 PD）；
6. 如果 Figure16-3 中的某个键被按下，端口上就会有一个电压值，当这个电压值小于 4.2V（@VDD=5V）时，AMWIF 标志位会被拉高。芯片也会从省电模式醒来，同时进入 ADC 中断服务程序，软件清零 AMWIF 标志；
7. 开启 ADC 功能，采集 P0.2 上的电压值，根据不同的电压值判断不同的按键。

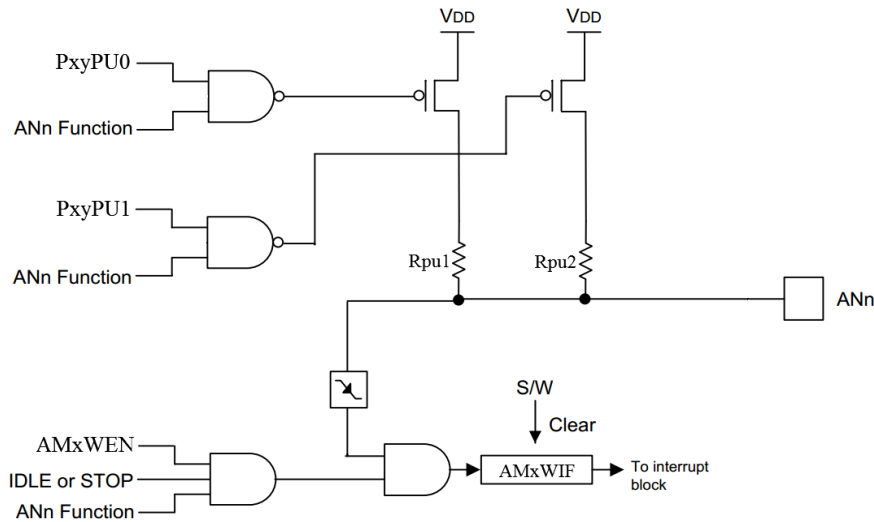


Figure 16-2 ADC 省电唤醒功能框图

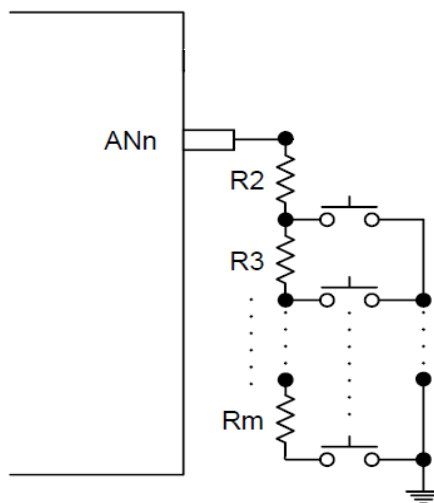


Figure 16-3 ADC 按键输入串联电阻应用参考图

## 16.3 ADC 相关寄存器

### 16.3.1 ADC 控制寄存器 ADCC0、ADCC1

#### ADCC0

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	1	1
位符号	ADCEN	ADCST	ADCIF	-	VREFO	VREFS	INREF_S[1:0]	

位编号	位符号	说明
7	ADCEN	ADC 模块电源控制位 0: 关闭 ADC 转换电源 1: 打开 ADC 转换电源 注意: 1. 在掉电模式下, ADCEN 强制为 0。 2. ADCEN 置 1 或切换转换通道后, 建议延时 20us 后再启动 ADC 转换。 3. 启动 ADC 转换时, 需要关闭 ADC 省电唤醒功能。
6	ADCST	ADC 启动控制位 0: 转换结束后, 硬件自动清 0, 在转换过程中, 软件清 0 将终止转换。 1: 启动转换 注: 启动转换时, ADCIF 需要先清 0, ADCIF 位为 1 时, 置 ADCST 不能启动新的转换。
5	ADCIF	ADC 中断标志位 0: 无 ADC 转换中断 1: 转换结束后, 硬件置 1, 可用于中断请求 (必须软件清 0)
4	-	保留位 (读为 0, 写无效)
3	VREFO	VREF 输出使能位 0: VREF 不输出 1: 从引脚 P0.4 输出内部 VREF, 此时需要设置 P0.4 为模拟输入, 而且 VREFS 必须为 0 注: VREF 输出驱动能力较弱, 仅供测试使用。
2	VREFS	VREF 选择 0: 选内部 VREF 1: 选外部 VREF (此时 P0.4 只做 ADC 参考电压输入, 且设为模拟输入)
1-0	INREF_S	ADC 内部参考电压选择位 00: VDD 01: 内部 4V 10: 内部 3V 11: 内部 2V 注意: 内部参考电压选择为 2V 时, VDD 电压须高于 2.7V; 内部参考电压选择 3/4V 时, VDD 须高于内部参考电压 0.5V 以上。 系统进入掉电模式前, 建议将 ADC 参考电压选择非 VDD, 可以进一步降低系统功耗。



**ADCC1**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	ICHS[1:0]		-		XCHS[3:0]			

位编号	位符号	说明
7-6	ICHS[1:0]	ADC 内部输入通道选择 00: 禁止内部通道接入 01: 1/4VDD 作为 ADC 输入通道 10: 保留位 11: GND 接入
5-4	-	保留位（读为 0，写无效）
3-0	XCHS[3:0]	ADC 外部输入通道选择 $XCHS[3:0] = x(x = 0 \dots 15)$ ，表示当前检测通道为 $An_x$ ，如 $XCHS[3:0] = 2$ ，表示当前检测通道为外部通道 AN2。 注：外部通道除设置 XCHS[3:0]，还需设置对应管脚的功能为模拟输入。

## 16.3.2 ADC 控制寄存器 ADCC2、ADCC3

### ADCC2

位编号	7	6	5	4	3	2	1	0
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
位符号	ADCL	ALIGN	ADCTS[2:0]			ADCS[2:0]		

位编号	位符号	说明
7	ADCL	ADC 转换数据长度控制位 0: ADC 转换结果为 12 位数据 1: ADC 转换结果为 10 位数据（取 12 位数据的高 10 位）
6	ALIGN	ADC 数据对齐方向控制位，见下面的 ADC 转换数据格式说明表
5-3	ADCTS [2:0]	ADC 时钟为 4MHZ 时，配置此 3bits 为 000；一次转换需要 22 个 ADC_CLK ADC 时钟为 2MHZ&1MHZ 时，配置此 3bits 为 001 或者 010；一次转换需要 19 个 ADC_CLK ADC 时钟为 <1MHZ 时，配置此 3bits 为 011 或 100 或 101 或 110 或 111；一次转换需要 15 个 ADC_CLK 注：此配置位只针对选择内部参考电压 1.3/2/3/4V，如果选择 VDD 作参考电压，无论 ADC 时钟为多少，都可以选择 15 个 ADC_CLK
2-0	ADCS[2:0]	ADC 时钟选择位 000: $F_{osc}/2$ 001: $F_{osc}/4$ 010: $F_{osc}/6$ 011: $F_{osc}/8$ 100: $F_{osc}/12$ 101: $F_{osc}/16$ 110: $F_{osc}/24$ 111: $F_{osc}/32$

ADC转换数据格式说明表:

ADCL	ALIGN	ADCRH								ADCRL							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	D11	D10	D9	D8	D7	D6	D5	D4	/	/	/	/	D3	D2	D1	D0
0	1	/	/	/	/	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1	0	D11	D10	D9	D8	D7	D6	D5	D4	/	/	/	/	/	/	D3	D2
1	1	/	/	/	/	/	/	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2

**ADCC3**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	IVREFS	FCLKEN	-	TRIGSEL[4:0]				

位编号	位符号	说明
7	IVREFS	ADC 1.3V 内参选择位 0: 有 ADCC0 寄存器里的 INREF_S 来决定 1: 内参为 1.3V
6	FCLKEN	选择内部参考电压时, ADC 转换的加速使能 0: 使用内参的 ADC 可以最快工作在 2MHz 的频率 1: 使用内参的 ADC 可以最快工作在 4MHz 的频率
5	-	保留位
4-0	TRIGSEL[4:0]	ADC 触发信号选择位 00000: ADC 转换启动仅由 ADCST(ADCC0.6)控制 00001: PWM0 上升沿 00010: PWM0 下降沿 00011: PWM0 中点 00100: PWM0 终点 00101: PWM1 上升沿 00110: PWM1 下降沿 00111: PWM1 中点 01000: PWM1 终点 01001: PWM2 上升沿 01010: PWM2 下降沿 01011: PWM2 中点 01100: PWM2 终点 10001: 定时器 5 的 T5 通道发生一次输入捕获事件 其他值: 保留 注意: 1、PWM 中点触发仅适用于中心对齐模式的 PWM 输出 2、当 ADCST 为 1 时 (正在转换), 外部触发信号不会影响 ADC 直到本次 ADC 转换结束即 ADCST 被硬件清 0

### 16.3.3 ADC 触发延时计时器 ADCDLYH、ADCDLYL

#### ADCDLYH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-				ADCDLY[11:8]			

位编号	位符号	说明
7-4	-	保留位
3-0	ADCDLY[11:8]	ADC 外部触发延时启动计时器的高 4 位

#### ADCDLYL

位编号	7	6	5	4	3	2	1	0
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
位符号	ADCDLY[7:0]							

位编号	位符号	说明
7-0	ADCDLY[7:0]	ADC 外部触发延时启动计时器的低 8 位, 用于在外部触发启动 ADC 之前插入一段延时, 在延时计时结束时开始 ADC 转换 延时时间=ADCDLY[11:0]* ADC 时钟

### 16.3.4 ADC 唤醒控制寄存器 ADCWC

#### ADCWC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	AMWEN	AMWIF	-					

位编号	位符号	说明
7	AMWEN	ADC 唤醒模块允许位 0: 禁止 ADC 唤醒模块 1: 允许 ADC 唤醒模块 注: 允许 ADC 唤醒模块时, 需要关闭 ADC。
6	AMWIF	ADC 唤醒模块中断标志位 (与 ADC 转换共用中断向量) 0: 无 ADC 唤醒模块中断, 软件清 0 1: 发生 ADC 唤醒模块中断, 硬件置 1 注: 相应唤醒模块被禁止时, 即使条件满足, 对应标志也不会被置 1。 需要使能 EADC。
5-0	-	保留位

注: 唤醒电阻通过端口上拉电阻来配置。

### 16.3.5 ADC 转换结果寄存器 ADCRL、ADCRH

#### ADCRL

位编号	7	6	5	4	3	2	1	0
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
位符号	ADCRL[7:0]							

#### ADCRH

位编号	7	6	5	4	3	2	1	0
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
位符号	ADCRH[7:0]							

位编号	位符号	说明
7-0	ADCRH[7:0]	ALIGN: 0 时 ADCRH[7:0]为 ADC 转换的高 8 位, ADCRL[3:0]为 ADC 转换的低 4/2 位
7-0	ADCRL[7:0]	ALIGN: 1 时 ADCRH[3:0]为 ADC 转换的高 4/2 位, ADCRL[7:0]为 ADC 转换的低 8 位

启动 ADC 转换步骤:

- (1) 使能 ADC 模块;
- (2) 选择模拟输入通道、参考电压、转换时钟、转换结果对齐方式等;
- (3) ADCST 置 1 开始 ADC 转换;
- (4) 等待 ADCST=0 或者 ADCIF=1, 如果 ADC 中断使能, 则 ADC 中断将会产生, 用户需要软件清零 ADCIF;
- (5) 从 ADCRH/ADCRL 获得转换数据;
- (6) 重复步骤 3-5 开始另一次转换

# 17 低电压检测/比较器

## 17.1 低电压检测/比较器特性

- 支持内部 VDD 多档位电压检测，并且可以产生中断
- 支持端口电压检测，并可以产生中断或复位
- LVD 档位：4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.9V
- 支持比较器功能

内部电压检测和 BOR 一样，也是检测 VDD 电压，但独立于 BOR，因此可以检测比 BOR 设定电压更高的各档检测电压点，可通过寄存器设定其检测电压点、其是否工作、其是否允许中断，方便客户需求。

LVD 电压检测电路有一定的迟滞特性，迟滞电压为 0.1V 左右。即当待检测电压下降到所选 LVD 电压档位时 LVD 会产生中断请求或复位，而待检测电压需要上升到 LVD 档位电压+0.1V 时 LVD 中断请求或复位才会解除。

LVD 检测端口 P2.6 引脚上的电压，当低于电压检测电压点 1.2V 时，置相应标志，若中断允许，则可产生中断请求；若中断不允许，端口电压检测将产生复位。端口电压检测产生的有效中断和复位可以将芯片从 PD 模式和 IDLE 模式唤醒。

LVD检测VDD电压时，不能产生复位，可通过有效中断将芯片从PD模式和IDLE模式唤醒。

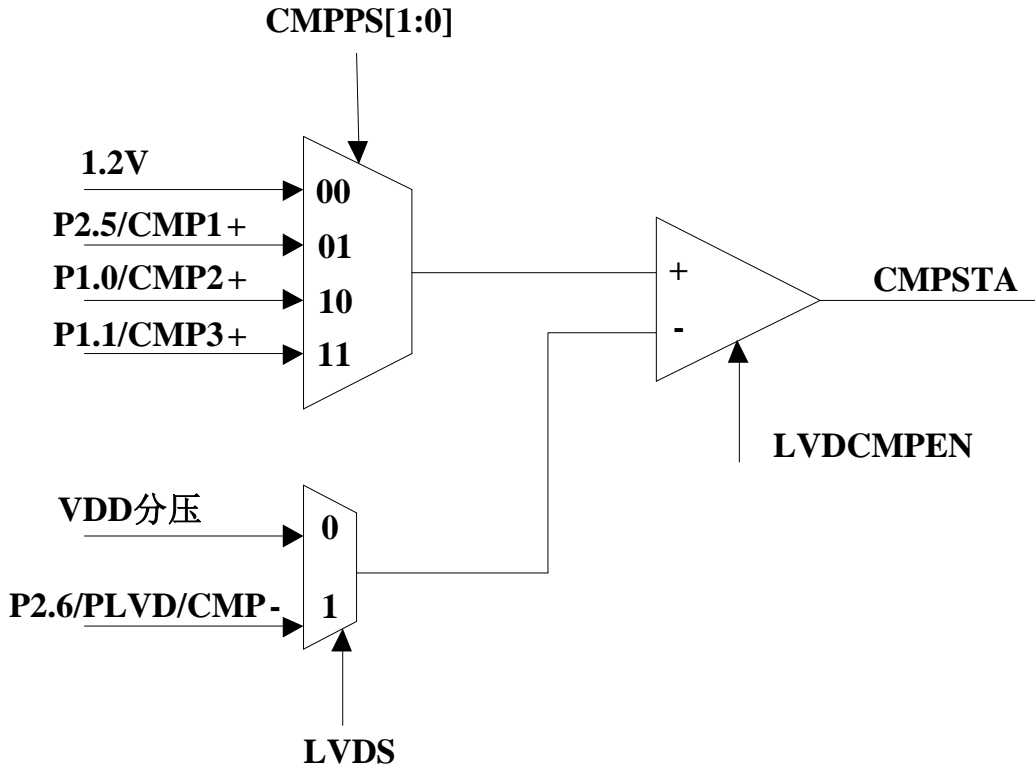


Figure 17-1 LVD/比较器框图

## 17.2 低电压检测/比较器相关寄存器

### 17.2.1 控制寄存器 LVDC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	LVDCMPEN	LVDS	LVDIE	-	LVDF	LVDV		

位编号	位符号	说明
7	LVDCMPEN	LVD/比较器使能位 0: 禁止 LVD/比较器 1: 允许 LVD/比较器
6	LVDS	LVD 检测选择位 0: 检测 VDD 电压 1: 检测 P2.6 端口电压
5	LVDIE	LVD 中断允许位 0: 禁止 LVD 中断 1: 允许 LVD 中断 注意: 禁止时, 只要允许检测, LVDF 也可被置 1, 但即使此时 EA 被置 1, 也不会产生中断请求。 当 LVDS 为 1, 检测 P2.6 端口电压时: LVDIE=0: 端口电压检测复位 LVDIE=1: 端口电压检测中断
4	-	保留位
3	LVDF	低电压检测标志位 0: 必须软件清 0 1: VDD 电压低于检测选择电压时, 硬件置 1, 也做中断请求 注意: VDD 电压低于检测电压时间大于 LVDDBC 寄存器设置的消抖时间后才会置 LVDF, 高于检测电压时, 不会自动清除该位, 必须软件清除, 只有在 VDD 电压持续高于检测电压时, 软件清除才能起作用, 如果 VDD 电压持续持续低于检测电压, 软件是无法清除 LVDF 的。
2-0	LVDV[2:0]	VDD 电压检测电压点选择位 000: 1.9V 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V 注意: LVD 检测电压设置在 BOR 电压之上才有意义。



## 17.2.2 比较功能控制寄存器 LVDCMP

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	1	0	0	0	0	0
位符号	-	-	DBEN	CMPSTA	CMPIM[1:0]		CMPPS[1:0]	

位编号	位符号	说明
7-6	-	保留位
5	DBEN	消抖使能 0: 不消抖 1: 消抖 注意: 1. LVD 和 CMP 都适用; 2. 掉电模式和空闲模式下自动不消抖, 退出掉电模式和空闲模式有 DBEN 来控制是否消抖
4	CMPSTA	比较器输出状态 0: 比较器正端电压小于负端电压 1: 比较器正端电压大于负端电压
3-2	CMPIM[1:0]	当 CMPPS[1:0]不选择 00 时, 需要配置这个寄存器: 00: 不置位 LVDF 01: CMP+从小于 CMP-到大于 CMP-后会置位 LVDF; 10: CMP+从大于 CMP-到小于 CMP-后会置位 LVDF; 11: CMP+从小于 CMP-到大于 CMP-或 CMP+从大于 CMP-到小于 CMP-都置位 LVDF 当作为 PWM FLT 来源时, 需要配置此寄存器为 01 或 10; 01: CMP+的电压大于 CMP-的电压, 会产生一个 PWM FLT 高的标志; 10: CMP+的电压小于 CMP-的电压, 会产生一个 PWM FLT 高的标志;
1-0	CMPPS[1:0]	比较器正端选择位 00: 1.2V 01: CMP1 管脚为比较器的正端输入 10: CMP2 管脚为比较器的正端输入 11: CMP3 管脚为比较器的正端输入

### 17.2.3 去抖控制寄存器 LVDDBC

位编号	7	6	5	4	3	2	1	0
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
位符号	LVDDBC[7:0]							

位编号	位符号	说明
7-0	LVDDBC[7:0]	LVD 消抖控制位 消抖时间： $LVDDBC[7:0] * 8T_{cpu} + 2T_{cpu}$

注：掉电模式和空闲模式下自动关闭，退出掉电模式和空闲模式又自动打开。

# 18 软件LCD

## 18.1 LCD 特性

- 支持 1/2Bias 的 LCD 点阵
- 驱动能力可配置
- COM 口数量和 SEG 口数量可任意配置
- LCD 控制信号（COM 和 SEG）由软件程序实现
- 可以通过关闭 LCDENU 或 LCDEND 来得到一个精准的下拉电阻或上拉电阻
- 可以通过同时关闭 LCDENU 和 LCDEND 来实现多个端口的内部连接

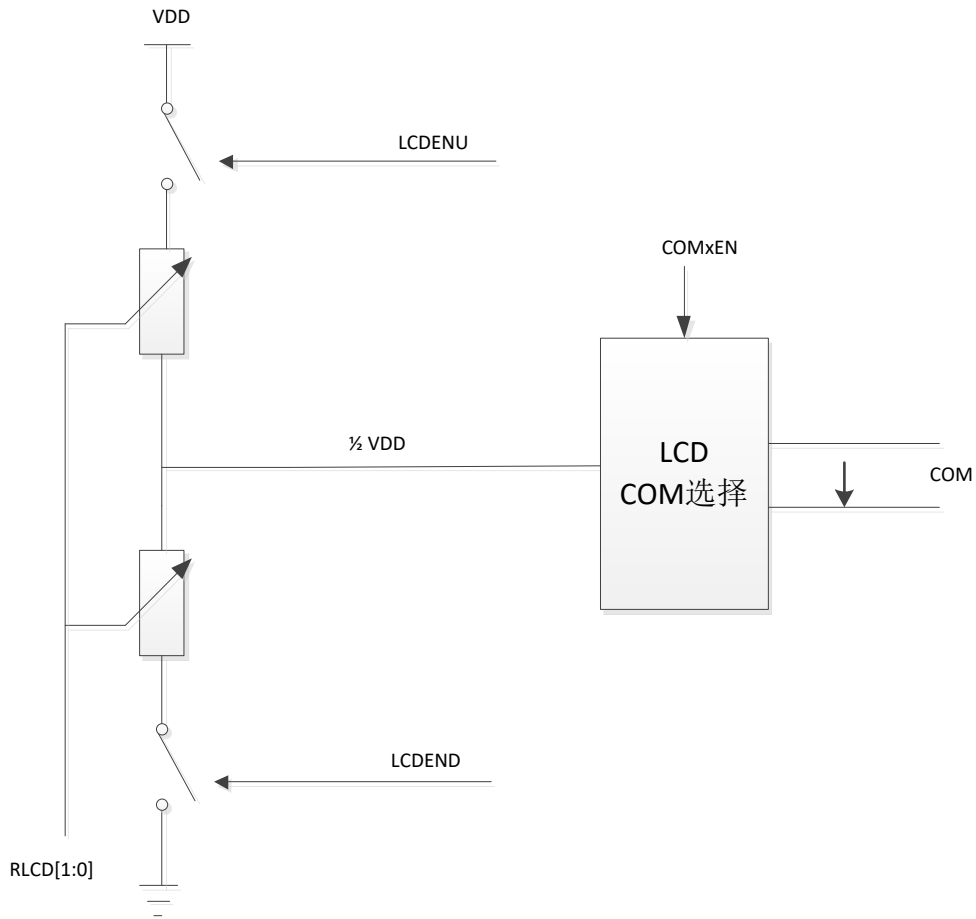


Figure 18-1 LCD 系统框图

## 18.2 软件流程说明

### 18.3 LCD 帧

一个完整的LCD波形周期包含两个Frame，即Frame0和Frame1。

#### Frame 0

在Frame0中，COM信号输出可以是VDD，或是1/2VDD；

在Frame0中，SEG信号输出可以是GND，或是VDD。

COM口和SEG口的电压差为VDD时，LCD被点亮。

#### Frame 1

在Frame1中，COM信号输出可以是GND，或是1/2VDD；

在Frame1中，SEG信号输出可以是VDD，或是GND。

COM口和SEG口的电压差为VDD时，LCD被点亮。

COM口可以输出VDD，GND或1/2VDD。其中，1/2VDD是通过LCD模块的分压输出到IO上，此时IO的模式配置为模拟输入，相应的COMPxEN使能，而VDD、GND是通过IO口的强推挽输出1和0来实现。

通过软件来决定SEG口输出VDD或GND。VDD、GND是通过IO口的强推挽输出1和0来实现。

下面的波形图显示了一个利用应用程序产生的典型1/2Bias LCD波形。COMn和SEGm引脚上所产生COM和SEG信号极性（0或1）通过软件来产生。

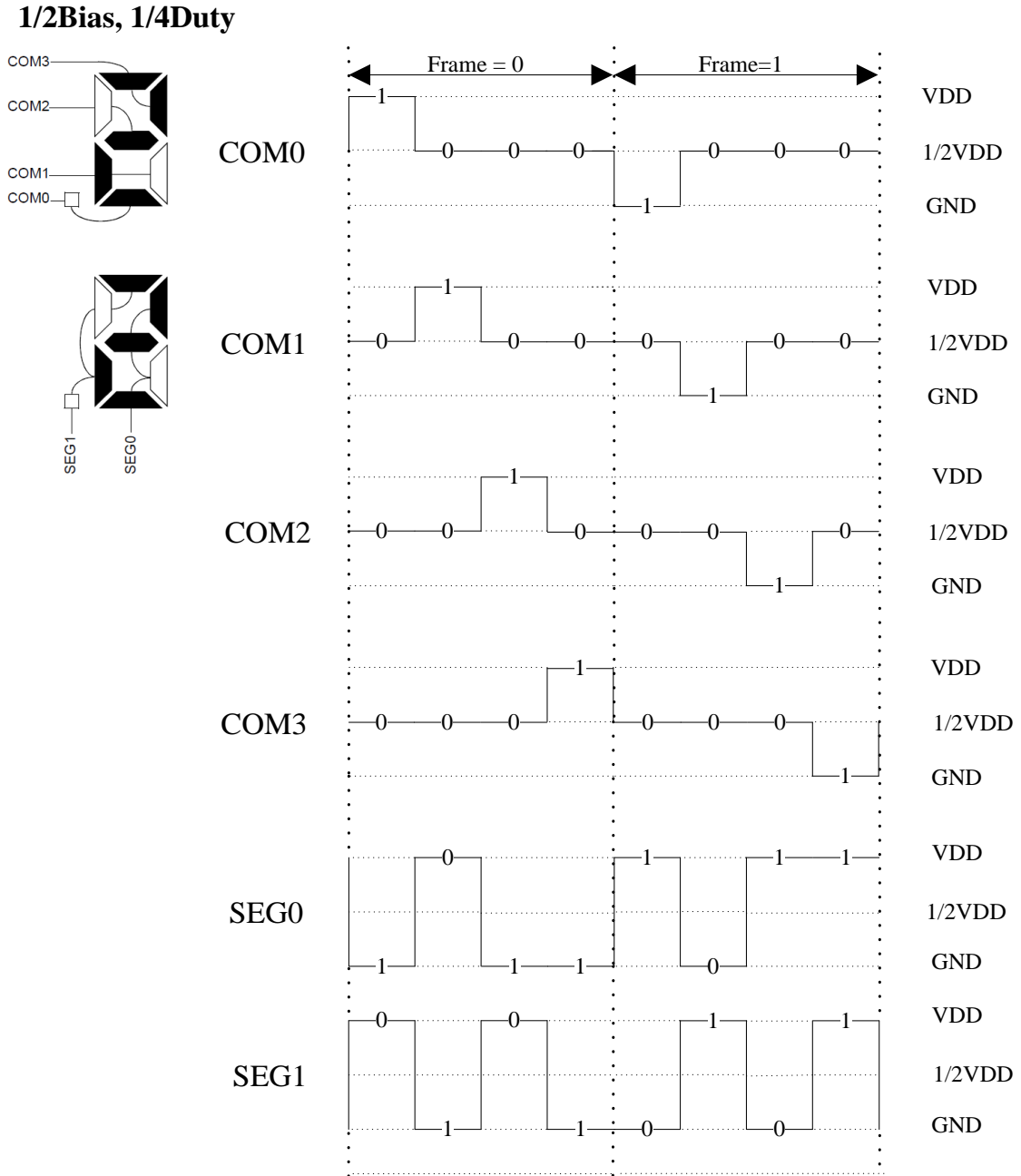


Figure 18-2 1/2bias LCD 波形图

## 18.4 LCD 相关寄存器

### 18.4.1 LCD 控制寄存器 LCDCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	LCDENU	LCDEND	RLCD1	RLCD0	-			

位编号	位符号	说明
7	LCDENU	LCD 上使能控制位 0: 禁止 1: 使能
6	LCDEND	LCD 下使能控制位 0: 禁止 1: 使能
5-4	RLCD[1:0]	软件 LCD 电阻选择位 00: 10kΩ 01: 25kΩ 10: 75kΩ 11: 175kΩ
3-0	-	保留

### 18.4.2 COM 口使能控制寄存器 COMP0EN-COMP2EN

位编号	7	6	5	4	3	2	1	0
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
位符号	COMPxEN[7:0]							

位编号	位符号	说明
7-0	COMPxENy	软件 LCD COM 功能使能位 0: 禁止, 普通 IO 1: 使能 注: x = 0~2 y = 0-7

# 19 循环冗余校验CRC

## 19.1 CRC 特性

- 16 位 CRC
- CRC 生成校验遵从 CRC-CCITT 多项式，即 0x1021
- 初值可设为 0x0000 或 0xFFFF
- 计算与结果共用同一寄存器

每一次写入数据寄存器CRCL，其计算结果是前一次CRC计算结果和新计算结果的组合。

每一次读出数据寄存器[CRCH:CRCL]，其值都为最后的CRC计算结果。

可通过设置寄存器 CRCC 的 CRCRSV 位来选择计算初值，但不会影响之前的 CRC 计算数据，只有置寄存器 CRCC 的 CRCRST 位后，才会复位 CRC 计算器，后写入的数据将以新的初值计算 CRC 结果。

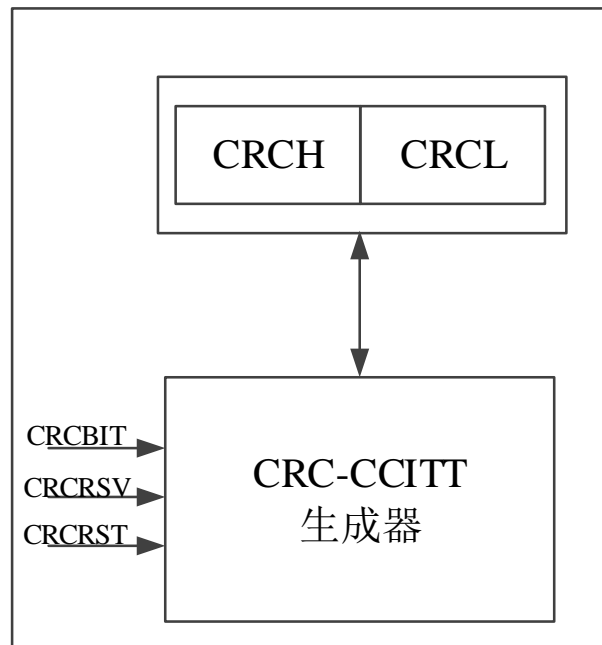


Figure 19-1 CRC 功能框图

## 19.2 CRC 相关寄存器

### 19.2.1 CRC 控制寄存器 CRCC

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	W
复位值	0	0	0	0	0	0	0	0
位符号	-					CRCBIT	CRCRSV	CRCRST

位编号	位符号	说明
7-3	-	保留位（读为 0，写无效）
2	CRCBIT	CRC BIT 翻转控制位 0: MSB first 1: LSB first
1	CRCRSV	CRC 复位初值选择位 0: 复位初值为 0x0000 1: 复位初值为 0xFFFF
0	CRCRST	CRC 计算器复位控制位 置 1 复位 CRC 计算器，硬件自动清 0



## 19.2.2 CRC 数据寄存器 CRCL、CRCH

### CRCL

位编号	7	6	5	4	3	2	1	0
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
位符号	CRCL[7:0]							

位编号	位符号	说明
7-0	CRCL[7:0]	写入数据时为 CRC 计算器的输入数据 读出数据时为 CRC 计算结果的低字节 注：写入数据时，自动启动 CRC 计算，完后自动关闭

### CRCH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	CRCH[7:0]							

位编号	位符号	说明
7-0	CRCH[7:0]	对该寄存器写入数据无效 读出数据时为 CRC 计算结果的高字节

注：每一次写入待计算数据，其计算结果是由与前一次计算结果共同生成。

## 20 代码选项

### 1. 外部复位使能

- P2.7 为外部复位引脚（默认）。该口作为外部复位引脚时，无法作为普通 I/O 使用
- P2.7 为普通 IO 引脚

### 2. BOR 检测电压

- 1.8V（默认）
- 2.0V
- 2.4V
- 2.6V
- 3.0V
- 3.6V
- 3.9V
- 4.2V

### 3. 复位后等待时间

- 4ms
- 8ms（默认）
- 16ms

### 4. 第二复位向量配置

用户可以通过此选择来配置自定义启动代码的首地址，配置值必须以 1K 字节为单位，即第二复位向量地址的最低 10 位必须为全零，默认不使能第二复位向量。

## 21 指令表

助记符		指令说明	字节	周期	代码
<b>数据传递类指令</b>					
MOV	A, Rn	寄存器送到累加器	1	1	0xE8-0xEF
MOV	A, direct	直接地址数据送到累加器	2	2	0xE5
MOV	A, @Ri	间接 RAM 数据送到累加器	1	2	0xE6-0xE7
MOV	A, #data	立即数送到累加器	2	2	0x74
MOV	Rn, A	累加器送到寄存器	1	1	0xF8-0xFF
MOV	Rn, direct	直接地址数据送到寄存器	2	2	0xA8-0xAF
MOV	Rn, #data	立即数送到寄存器	2	2	0x78-0x7F
MOV	direct, Rn	寄存器送到直接地址	2	2	0x88-0x8F
MOV	direct, direct	直接地址数据送到直接地址	3	3	0x85
MOV	direct, A	累加器送到直接地址	2	2	0xF5
MOV	direct, @Ri	间接 RAM 数据送到直接地址	2	2	0x86-0x87
MOV	direct, #data	立即数送到直接地址	3	3	0x75
MOV	@Ri, A	累加器送到间接 RAM	1	1	0xF6-0xF7
MOV	@Ri, direct	直接地址数据送到间接 RAM	2	2	0xA6-0xA7
MOV	@Ri, #data	立即数送到间接 RAM	2	2	0x76-0x77
MOV	DPTR, #data16	16 位立即数加载到数据指针	3	3	0x90
MOVC	A, @A+DPTR	代码字节送到累加器	1	3	0x93
MOVC	A, @A+PC	代码字节送到累加器	1	3	0x83
MOVX	A, @Ri	外部 RAM(8 位地址)送到累加器	1	3	0xE2-0xE3
MOVX	A, @DPTR	外部 RAM(16 位地址)送到累加器	1	3	0xE0
MOVX	@Ri, A	累加器送到外部 RAM(8 位地址)	1	3	0xF2-0xF3
MOVX	@DPTR, A	累加器送到外部 RAM(16 位地址)	1	3	0xF0
PUSH	direct	直接地址数据压入堆栈	2	2	0xC0
POP	direct	直接地址数据弹出堆栈	2	2	0xD0
XCH	A,Rn	寄存器和累加器交换	1	1	0xC8-0xCF

XCH	A, direct	直接地址数据和累加器交换	2	2	0xC5
XCH	A, @Ri	间接 RAM 和累加器交换	1	2	0xC6-0xC7
XCHD	A, @Ri	间接 RAM 和累加器交换低 4 位字节	1	2	0xD6-0xD7
<b>算术运算类指令</b>					
INC	A	累加器加 1	1	1	0x04
INC	Rn	寄存器加 1	1	1	0x08-0x0F
INC	direct	直接地址数据加 1	2	2	0x05
INC	@Ri	间接 RAM 加 1	1	2	0x06-0x07
INC	DPTR	数据指针加 1	1	1	0xA3
DEC	A	累加器减 1	1	1	0x14
DEC	Rn	寄存器减 1	1	1	0x18-0x1F
DEC	direct	直接地址数据减 1	2	2	0x15
DEC	@Ri	间接 RAM 减 1	1	2	0x16-0x17
MUL	AB	累加器和 B 寄存器相乘	1	4	0xA4
DIV	AB	累加器除以 B 寄存器	1	4	0x84
DA	A	累加器十进制调整	1	1	0xD4
ADD	A,Rn	寄存器与累加器求和	1	1	0x28-0x2F
ADD	A,direct	直接地址数据与累加器求和	2	2	0x25
ADD	A,@Ri	间接 RAM 与累加器求和	1	2	0x26-0x27
ADD	A,#data	立即数与累加器求和	2	2	0x24
ADDC	A,Rn	寄存器与累加器求和（带进位）	1	1	0x38-0x3F
ADDC	A,direct	直接地址数据与累加器求和（带进位）	2	2	0x35
ADDC	A,@Ri	间接 RAM 与累加器求和（带进位）	1	2	0x36-0x37
ADDC	A,#data	立即数与累加器求和（带进位）	2	2	0x34
SUBB	A,Rn	累加器减去寄存器（带借位）	1	1	0x98-0x9F
SUBB	A,direct	累加器减去直接地址数据（带借位）	2	2	0x95
SUBB	A,@Ri	累加器减去间接 RAM（带借位）	1	2	0x96-0x97
SUBB	A,#data	累加器减去立即数（带借位）	2	2	0x94
<b>逻辑运算类指令</b>					

ANL	A,Rn	寄存器“与”到累加器	1	1	0x58-0x5F
ANL	A,direct	直接地址数据“与”到累加器	2	2	0x55
ANL	A,@Ri	间接 RAM “与”到累加器	1	2	0x56-0x57
ANL	A,#data	立即数“与”到累加器	2	2	0x54
ANL	direct,A	累加器“与”到直接地址	2	2	0x52
ANL	direct, #data	立即数“与”到直接地址	3	3	0x53
ORL	A,Rn	寄存器“或”到累加器	1	1	0x48-0x4F
ORL	A,direct	直接地址数据“或”到累加器	2	2	0x45
ORL	A,@Ri	间接 RAM “或”到累加器	1	2	0x46-0x47
ORL	A,#data	立即数“或”到累加器	2	2	0x44
ORL	direct,A	累加器“或”到直接地址	2	2	0x42
ORL	direct, #data	立即数“或”到直接地址	3	3	0x43
XRL	A,Rn	寄存器“异或”到累加器	1	1	0x68-0x6F
XRL	A,direct	直接地址数据“异或”到累加器	2	2	0x65
XRL	A,@Ri	间接 RAM “异或”到累加器	1	2	0x66-0x67
XRL	A,#data	立即数“异或”到累加器	2	2	0x64
XRL	direct,A	累加器“异或”到直接地址	2	2	0x62
XRL	direct, #data	立即数“异或”到直接地址	3	3	0x63
CLR	A	累加器清零	1	1	0xE4
CPL	A	累加器求反	1	1	0xF4
RL	A	累加器循环左移	1	1	0x23
RLC	A	带进位累加器循环左移	1	1	0x33
RR	A	累加器循环右移	1	1	0x03
RRC	A	带进位累加器循环右移	1	1	0x13
SWAP	A	累加器高、低 4 位交换	1	1	0xC4
<b>控制转移类指令</b>					
JMP	@A+DPTR	相对 DPTR 的无条件间接转移	1	2	0x73
JZ	rel	累加器为 0 则转移	2	3	0x60
JNZ	rel	累加器为 1 则转移	2	3	0x70

CJNE	A,direct,rel	比较直接地址和累加器，不相等转移	3	4	0xB5
CJNE	A,#data,rel	比较立即数和累加器，不相等转移	3	4	0xB4
CJNE	Rn,#data,rel	比较寄存器和立即数，不相等转移	2	4	0xB8-0xBF
CJNE	@Ri,#data,rel	比较立即数和间接 RAM，不相等转移	3	5	0xB6-0xB7
DJNZ	Rn,rel	寄存器减 1，不为 0 则转移	2	3	0xD8-0xDF
DJNZ	direct,rel	直接地址数据减 1，不为 0 则转移	3	4	0xD5
NOP		空操作，用于短暂延时	1	1	0x00
ACALL	add11	绝对调用子程序	2	2	xxx10001b
LCALL	add16	长调用子程序	3	3	0x12
RET		从子程序返回	1	4	0x22
RETI		从中断服务子程序返回	1	4	0x32
AJMP	add11	无条件绝对转移	2	2	xxx00001b
LJMP	add16	无条件长转移	3	3	0x02
SJMP	rel	无条件相对转移	2	3	0x80
<b>布尔指令</b>					
CLR	C	清零进位位	1	1	0xC3
CLR	bit	清零直接寻址位	2	2	0xC2
SETB	C	置位进位位	1	1	0xD3
SETB	bit	置位直接寻址位	2	2	0xD2
CPL	C	取反进位位	1	1	0xB3
CPL	bit	取反直接寻址位	2	2	0xB2
ANL	C,bit	直接寻址位“与”到进位位	2	2	0x82
ANL	C, /bit	直接寻址位的反码“与”到进位位	2	2	0xB0
ORL	C,bit	直接寻址位“或”到进位位	2	2	0x72
ORL	C, /bit	直接寻址位的反码“或”到进位位	2	2	0xA0
MOV	C,bit	直接寻址位送到进位位	2	2	0xA2
MOV	bit, C	进位位送到直接寻址	2	2	0x92
JC	rel	如果进位位为 1 则转移	2	3	0x40
JNC	rel	如果进位位为 0 则转移	2	3	0x50

JB	bit, rel	如果直接寻址位为 1 则转移	3	4	0x20
JNB	bit, rel	如果直接寻址位为 0 则转移	3	4	0x30
JBC	bit, rel	直接寻址位为 1 则转移并清除该位	3	4	0x10

## 22 电气特性

除非另外说明，以下数据测试条件均为：VDD=5.0V，GND=0V，25°C。

### 22.1 极限参数

参数	符号	最小值	典型值	最大值	单位
直流供电电压	VDD	-0.3	-	+6.0	V
输入/输出电压	V <sub>I</sub> /V <sub>O</sub>	GND-0.3	-	VDD+0.3	V
工作环境温度	T <sub>ORG</sub>	-40	-	+105	°C
存储温度	T <sub>STG</sub>	-55	-	+125	°C

注：（1）流过 VDD 的最大电流值在 5.0V，25°C 下须小于 100mA。

（2）流过 GND 的最大电流值在 5.0V，25°C 下须小于 150mA。

### 22.2 DC 特性

参数	符号	条件 (VDD: 5V)	最小值	典型值	最大值	单位
工作电压	VDD	F <sub>CPU</sub> =16MHz 或 44KHz, ADC 模块关闭	2.0	5.0	5.5	V
工作电流	I <sub>OP1</sub>	Fosc=32MHz, F <sub>CPU</sub> =16MHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	2.8	-	mA
		Fosc=32MHz, F <sub>CPU</sub> =8MHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	2.2	-	
		Fosc=32MHz, F <sub>CPU</sub> =4MHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	1.7	-	
		Fosc=32MHz, F <sub>CPU</sub> =2MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	1.5	-	
		Fosc=32MHz, F <sub>CPU</sub> =1MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	1.3	-	
		Fosc=32MHz, F <sub>CPU</sub> =500KHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	1.2	-	
		Fosc=16MHz, F <sub>CPU</sub> =16MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	2.2	-	
		Fosc=16MHz, F <sub>CPU</sub> =8MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	1.5	-	
		Fosc=16MHz, F <sub>CPU</sub> =4MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭	-	1.1	-	



	Fosc=16MHz, Fcpu =2MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.9		
	Fosc=16MHz, Fcpu =1MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.8		
	Fosc=16MHz, Fcpu =500KHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.8		
	Fosc=8MHz, Fcpu =8MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		1.3		
	Fosc=8MHz, Fcpu =4MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.9		
	Fosc=8MHz, Fcpu =2MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.7		
	Fosc=8MHz, Fcpu =1MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.6		
	Fosc=8MHz, Fcpu =500KHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.5		
	Fosc=4MHz, Fcpu =4MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.8		
	Fosc=4MHz, Fcpu =2MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.6		
	Fosc=4MHz, Fcpu =1MHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.5		
	Fosc=4MHz, Fcpu =500KHz, 无负载, 无浮动输入, 执行 NOP, 其它模块关闭		0.4		
I <sub>OP2</sub>	Fosc =44KHz, Fcpu=44KHz, 无负载, 无浮动, 执行 NOP 指令, 其它模块关闭	-	70	-	μA
I <sub>PD1</sub>	进入掉电模式, 无负载, 无浮动输入管脚, BOR 打开, 其它模块关闭	-	12	-	μA
I <sub>PD2</sub>	进入掉电模式, 无负载, 无浮动输入管脚, BOR 关闭, 其它模块关闭		5.5		μA
I <sub>IDLE1</sub>	Fosc =32MHz, Fcpu =16MHz, 进入空闲模式, 无负载, 无浮动输入, 所有模块关闭	-	1.2	-	mA
I <sub>IDLE2</sub>	Fosc =16MHz, Fcpu =16MHz, 进入空闲模式, 无负载, 无浮动输入, 所有模块关闭		0.7		mA
I <sub>IDLE3</sub>	Fosc =8MHz, Fcpu =8MHz, 进入空闲模式, 无负载无浮动输入, 所有模块关闭	-	0.5	-	mA
I <sub>IDLE4</sub>	Fosc =4MHz, Fcpu =4MHz, 进入空闲模式, 无负载, 无浮动输入, 所有模块关闭	-	0.4	-	mA
I <sub>IDLE4</sub>	Fosc =44KHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭, 内部高频 RC 时钟关闭	-	63	-	μA

掉电定时中断唤醒电流	$I_{PW}$	$F_{CPU}=16\text{MHz}$ , 关闭 BOR, TIMER3 计数时钟源选择 RC44K, 系统进入掉电, TIMER3 定时 1S 中断唤醒的平均电流	-	10	-	$\mu\text{A}$
输入低电压 1	$V_{IL1}$	I/O 端口非施密特输入	GND	-	$0.3*V_{DD}$	V
输入高电压 1	$V_{IH1}$	I/O 端口非施密特输入	$0.7*V_{DD}$	-	VDD	V
输入低电压 2	$V_{IL2}$	I/O 端口施密特输入	GND	-	$0.2*V_{DD}$	V
输入高电压 2	$V_{IH2}$	I/O 端口施密特输入	$0.8*V_{DD}$	-	VDD	V
输入漏电流	$I_{ILC}$	I/O 端口输入模式, $V_{IN}$ : VDD或GND	-1	0	1	$\mu\text{A}$
输出漏电流	$I_{OLC}$	I/O 端口输出模式, $V_{OUT}$ : VDD或GND	-1	0	1	$\mu\text{A}$
灌电流	$I_{OL1}$	$V_{out}=GND+0.6$	-	25	-	mA
拉电流	$I_{OH1}$	$V_{out}=V_{DD}-0.6$	-	21	-	
上拉电阻	$R_{PU1}$	P0.2 端口, $V_{IN}=GND$	-	50	-	k $\Omega$
	$R_{PU2}$	P0.2 端口, $V_{IN}=GND$	-	100	-	
	$R_{PU3}$	P0.2 端口, $V_{IN}=GND$	-	150	-	
	$R_{PU4}$	P0.2 端口, $V_{IN}=GND$	-	300	-	
	$R_{PU5}$	普通端口, $V_{IN}=GND$	-	50	-	
下拉电阻	$R_{PD}$	普通端口, $V_{IN}=V_{DD}$	-	50	-	k $\Omega$
ADC 唤醒电压	$V_{AW1}$	常温, $V_{DD}=5\text{V}$	4.0	4.2	4.4	V
	$V_{AW2}$	常温, $V_{DD}=3\text{V}$	2.3	2.5	2.6	
RAM 保持电压	$V_{RAM}$	-	-	0.7	-	

注意: 除非另外说明, 以上数据测试条件均为: VDD: 5.0V, GND: 0V, 25°C。

## 22.3 AC 特性

参数	符号	条件	最小值	典型值	最大值	单位
内部 RC32M 启动时间	Tset1	常温, $V_{DD}=5\text{V}$	-	-	5	$\mu\text{s}$
内部 RC44K 启动时间	Tset2	常温, $V_{DD}=5\text{V}$	-	-	150	$\mu\text{s}$
外部高频振荡器启动时间	Tset3	16MHz, 常温, $V_{DD}=5\text{V}$	-	200	-	$\mu\text{s}$
外部高频振荡器工作电压	Vset3	16MHz	2.5	-	5.5	V
外部低频振荡器启动时间	Tset4	常温, $V_{DD}=5\text{V}$	-	2	-	s
频率精度	FIRC1	$V_{DD}=2\text{V}\sim 5.5\text{V}$ , 25°C	32(1-1%)	32	32(1+1%)	MHz
	FIRC2	$V_{DD}=5.0\text{V}$ , -20°C ~+85°C	32(1-2%)	32	32(1+2%)	MHz
	FIRC3	$V_{DD}=5.0\text{V}$ , -40°C ~+105°C	32(1-4%)	32	32(1+4%)	MHz
	FWRC	-	31	44	58	KHz

## 22.4 FLASH 内存特性

参数	符号	条件	最小值	典型值	最大值	单位
读写测试	NENDUR	-	100000	-	-	Cycle
数据保存时间	TRET	T=25°C	-	10	-	year
FLASH 扇区擦除时间	TERASE1	1 个扇区 (128 字节)	-	5	-	ms
EEPROM 扇区擦除时间	TERASE2	1 个扇区 (4 字节)	-	5	-	ms
字节写入时间	TPROG	1 个字节	-	68	-	μs
读取耗电流	IDD1	Fcpu=16MHz	-	4	-	mA
写入耗电流	IDD2	-	-	4	-	mA
擦除耗电流	IDD3	-	-	2	-	mA

## 22.5 ADC 特性

参数	符号	条件	最小值	典型值	最大值	单位
供电电压	VAD	-	2.0	5.0	5.5	V
精度	NR	GND≤VAIN≤Vref	-	10	12	bit
ADC 输入电压	VAIN	-	GND	-	Vref	V
ADC 输入电阻	RAIN	VAIN: 5V	2	-	-	MΩ
模拟电压源推荐阻抗	ZAIN	-	-	-	10	kΩ
ADC 转换电流	IAD	ADC 模块打开, VDD: 5.0V	-	0.6	1	mA
ADC 输入电流	IADIN	VDD: 5.0V	-	-	10	μA
微分非线性误差	DLE	VDD: 5.0V	-	-	±2	LSB
积分非线性误差 (1MHz 转换频率)	ILE	VDD=5.0V, Vref=1.3V	-	-	-5~2	LSB
		VDD=5.0V, Vref=2V	-	-	-5~2	
		VDD=5.0V, Vref=3V	-	-	-4~2	
		VDD=5.0V, Vref=4V	-	-	-3~2	
		VDD=5.0V, Vref=VDD	-	-	±2	
VDD=5.0V, Vref=外参	-	-	±2			
满刻度误差	EF	VDD=5.0V	-	-	±5	LSB
偏移量误差	EZ	VDD=5.0V	-	-	±3	LSB
总绝对误差	EAD	VDD=5.0V	-	-	±5	LSB
总转换时间 1	TCON1	VDD=5.0V Vref=2/3/4V	10	-	-	μs
总转换时间 2	TCON2	VDD=5.0V Vref=VDD	2	-	-	μs
内部参考电压	VADREF	VDD=5.0V, Vref=2V	2(1-1%)	2	2(1+1%)	V

## 22.6 BOR 检测电压特性

参数	符号	条件	最小值	典型值	最大值	单位
BOR 设定电压 1	VBOR1	BOR 使能, VDD: 2V~5.5V	1.7	1.8	1.9	V
BOR 设定电压 2	VBOR2		1.9	2.0	2.1	V
BOR 设定电压 3	VBOR3		2.3	2.4	2.5	V
BOR 设定电压 4	VBOR4		2.5	2.6	2.7	V
BOR 设定电压 5	VBOR5		2.9	3.0	3.1	V
BOR 设定电压 6	VBOR6		3.5	3.6	3.7	V
BOR 设定电压 7	VBOR7		3.8	3.9	4.0	V
BOR 设定电压 8	VBOR8		4.1	4.2	4.3	V

## 22.7 LVD/PLVD 检测电压特性

参数	符号	条件	最小值	典型值	最大值	单位
LVD 设定电压 0	VPLVD	LVD 使能, VDD=2V~5.5V	-	1.2	-	V
LVD 设定电压 1	VLVD1		1.8	1.9	2.0	V
LVD 设定电压 2	VLVD2		1.9	2.0	2.1	V
LVD 设定电压 3	VLVD3		2.3	2.4	2.5	V
LVD 设定电压 4	VLVD4		2.5	2.6	2.7	V
LVD 设定电压 5	VLVD5		2.9	3.0	3.1	V
LVD 设定电压 6	VLVD6		3.5	3.6	3.7	V
LVD 设定电压 7	VLVD7		3.8	3.9	4.0	V
LVD 设定电压 8	VLVD8		4.1	4.2	4.3	V

## 22.8 比较器电气特性

(VDD=2V~5.5V, 温度范围: -40°C~105°C, 室温为: 25°C, 除非另有说明)

符号	参数	测试条件			最小值	典型值	最大值	单位
		VDD	温度	条件				
V <sub>OS</sub>	输入失调电压	5V	-	-	-	±2	±4	mV
V <sub>ICM</sub>	输入共模电压范围	5V	-	-	0	-	VDD-1.2V	V
I <sub>COMP</sub>	比较器工作功耗	-	-	-	-	200	400	nA
Tr <sub>S1</sub>	小信号响应时间	5V	-	CMPP=1.25V CMPN=1.2V	-	1	2	us
Tr <sub>S2</sub>	大信号响应时间	5V	-	CMPP=3V CMPN=1V	-	0.3	0.5	us
R <sub>in</sub>	信号源输出阻抗	5V	-	-	-	-	1	MΩ

## 22.9 系统下电过程功耗

### 1、系统关闭 BOR，进入掉电模式

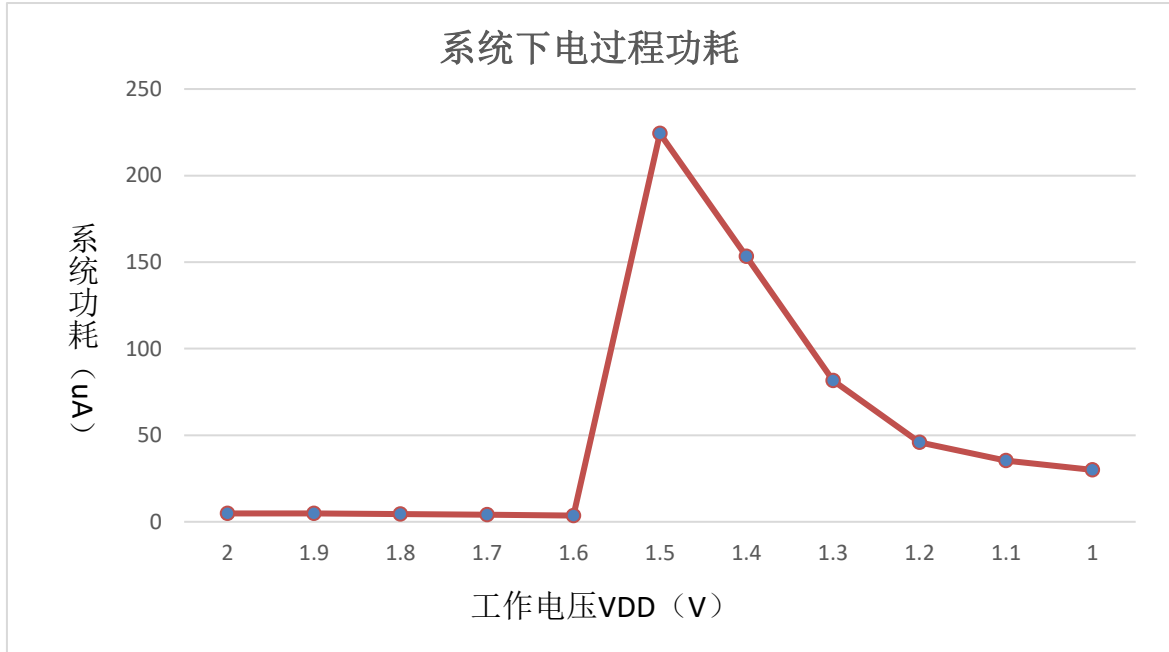


Figure 22-1 系统下电过程功耗

### 2、系统使能BOR，进入掉电模式，使能/禁止BOR\_PD\_EN

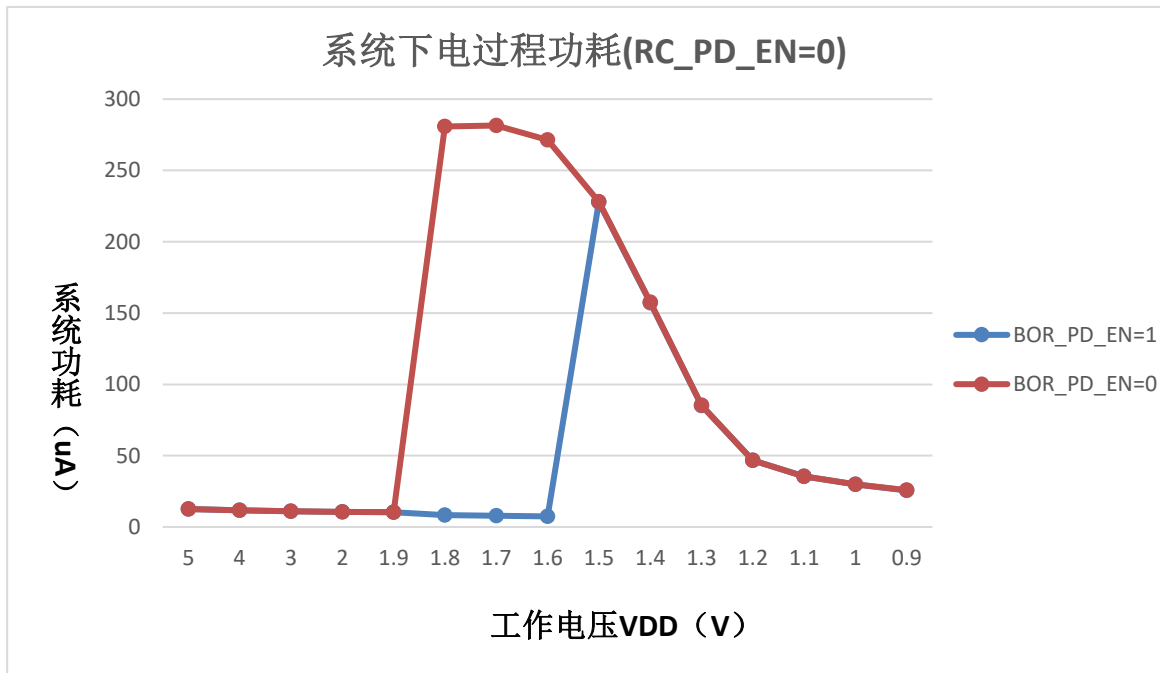


Figure 22-2 系统下电过程功耗

## 22.10 频率-电压特性曲线

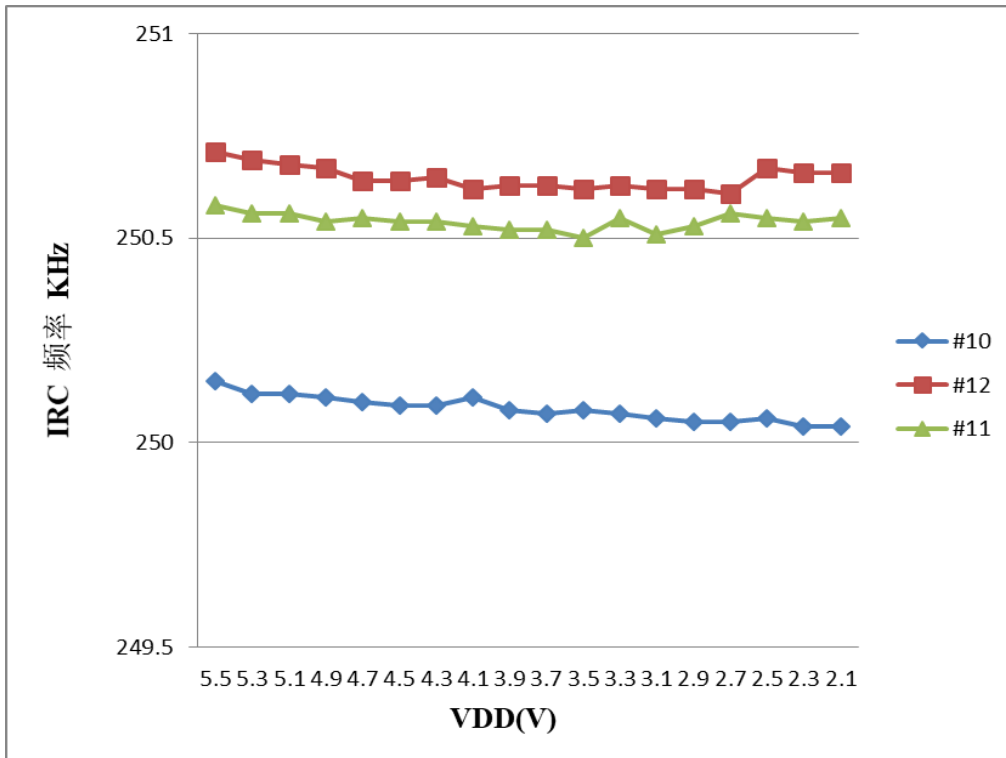


Figure 22-3 内部高频/128-电压特性曲线图

## 22.11 频率-温度特性曲线

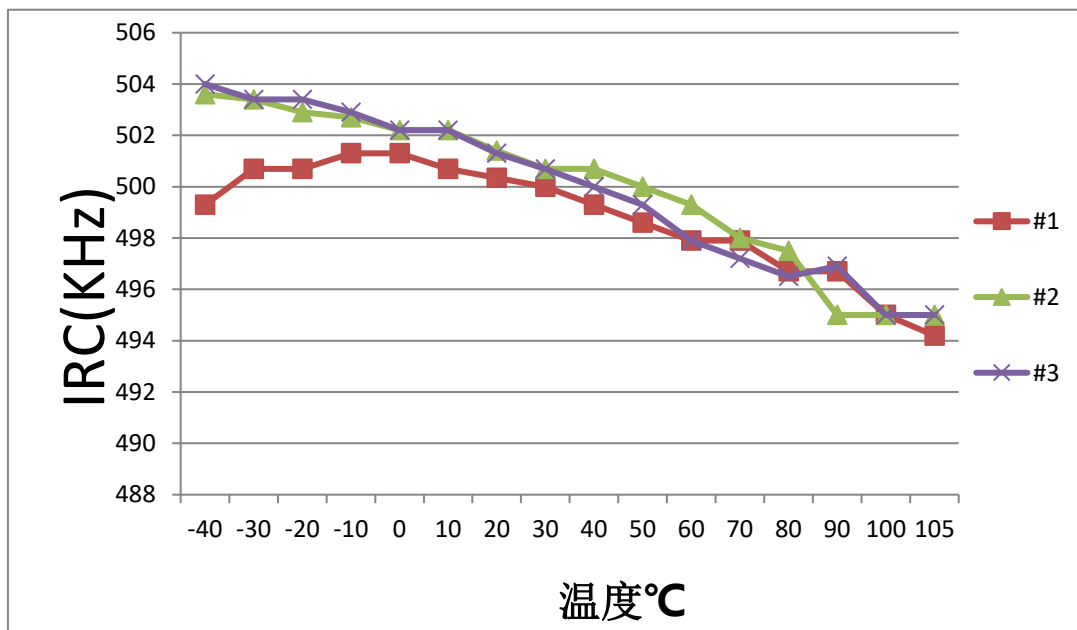


Figure 22-4 内部高频RC32M/500-温度特性曲线图

## 22.12 ADC 内参 2V-温度特性曲线

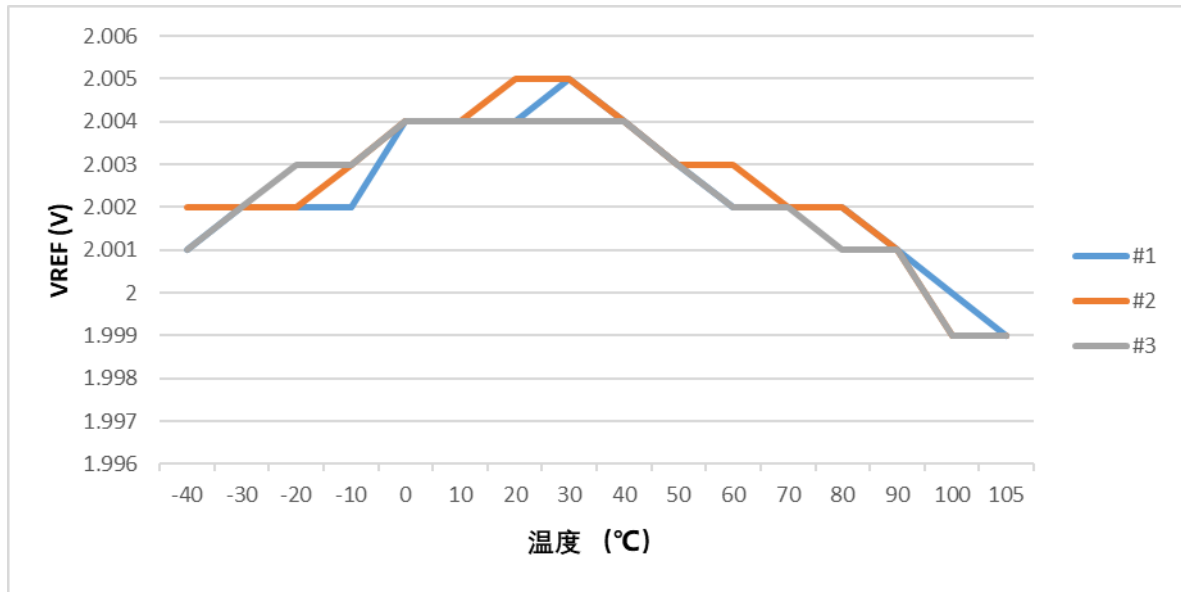


Figure 22-5 内部参考电压2V-温度特性曲线图

## 22.13 其他电气特性

- 1、ESD (HBM): Class3A ( $\geq 5500V$ )
- 2、ESD (MM): Class C ( $\geq 400V$ ) 最高等级
- 3、ESD (CDM): Class C3 ( $\geq 1000V$ ) 最高等级
- 4、Latch\_up: 800mA
- 5、EFT:  $\pm 4800V$

## 23 开发工具

### 23.1 HC-LINK 仿真工具

HC89F0431A/0421A/0411A 使用 HC-LINK 仿真器进行程序的下载和仿真，HC-LINK 通过 JTAG/SWD 接口可以对芯圣所有的增强型 8051 内核单片机（非固化 ISP）实现下载和仿真。关于 HC-LINK 的使用，请参见 HC-LINK 用户手册。

HC-LINK 特性：

- 支持 Keil C51 集成编译环境（uVision4.0 及以上版本）
- 支持所有的芯圣 8051（非固化 ISP）芯片
- 可以对 FLASH 进行擦除、编程和校验
- 可以对加密位以及代码选项进行编程
- 直接从 USB 供电，不需要外接电源

### 23.2 HC-PM51 烧录工具

HC-PM51 烧录器是芯圣新一代量产烧录工具，适用于芯圣 8051 内核系列的 Flash MCU 的烧录。关于 HC-PM51 的使用，请参见 HC-PM51 工具用户手册。

HC-PM51 特性：

- 采用 USB 方式连接
- 支持单路脱机烧录

### 23.3 ISP 串口烧录

HC-LINK V4.0 和 HC-PM51 支持脱机固化 ISP 程序，当把 ISP 程序固化到 HC89F0431A/0421A/0411A 之后，用户就可以使用 TXD/RXD 两个引脚进行程序的下载和更新。

用户可以使用 HC-LINK V4.0 工具，配合上位机软件 HC-ISP，通过串口对固化 ISP 程序的 FLASH 单片机实现一键下载的功能。另外，用户也可以使用 HC-PM51 的 ISP 模式进行用户程序的脱机下载。

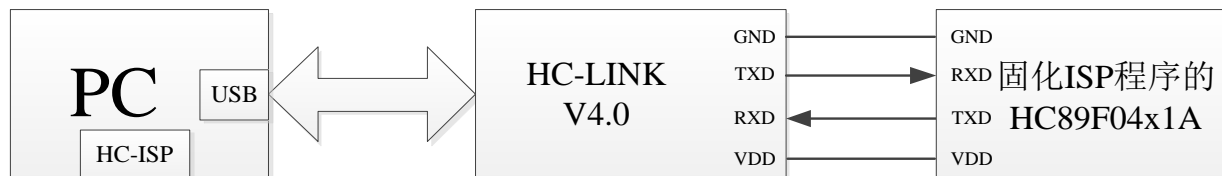


Figure 23-1 ISP 串口烧录框图

### 23.4 软件下载

直接在浏览器地址栏中输入地址：[www.holychip.cn](http://www.holychip.cn) 进行软件下载。



# 24 封装尺寸

## 24.1 SOP20

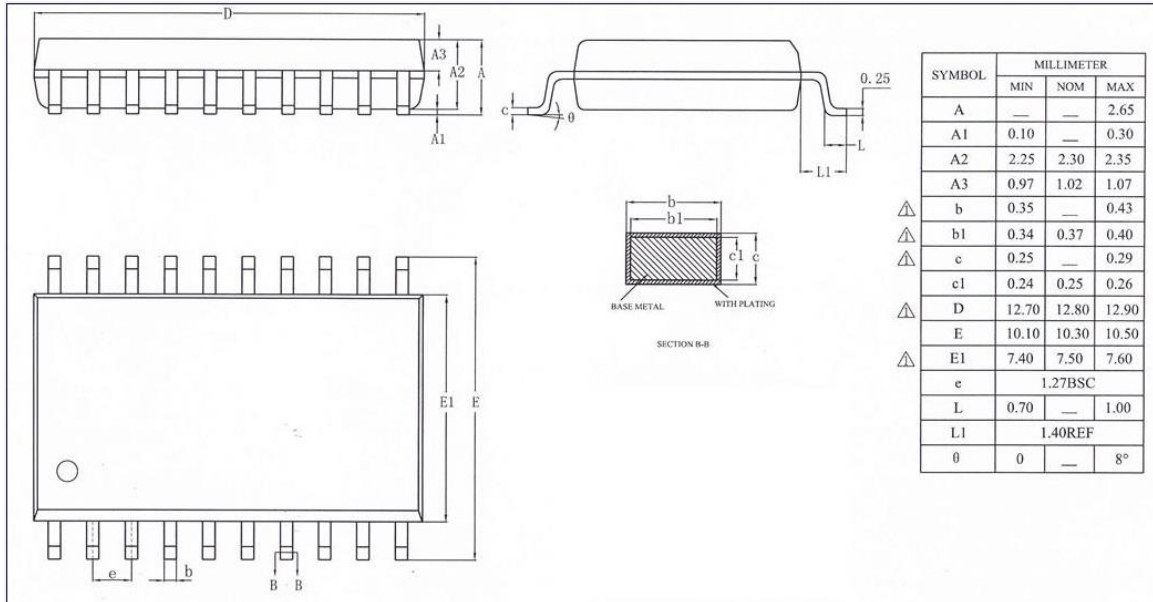


Figure 24-1 SOP20 封装尺寸

## 24.2 TSSOP20

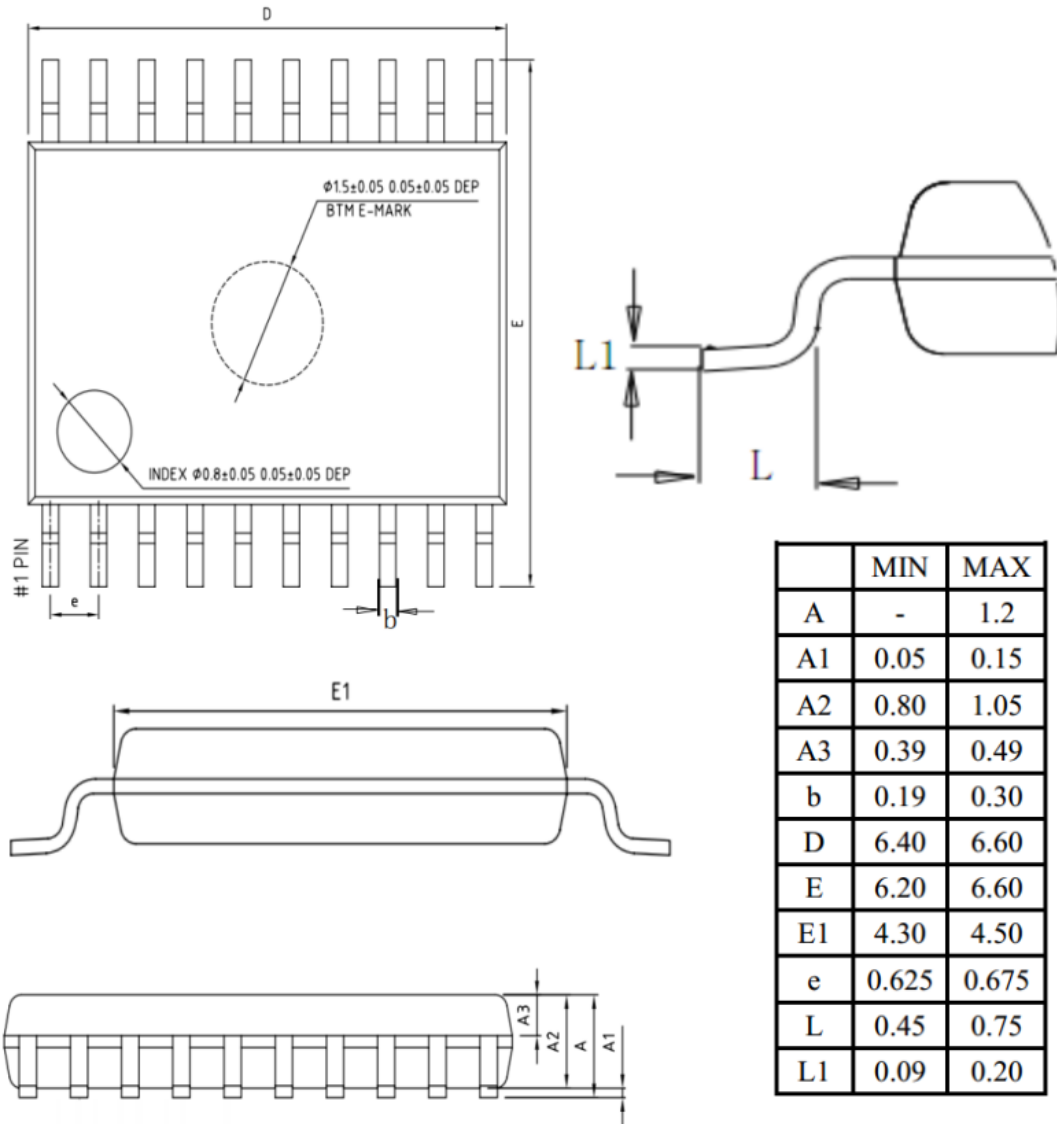


Figure 24-2 TSSOP20 封装尺寸



## 24.4 SOP8

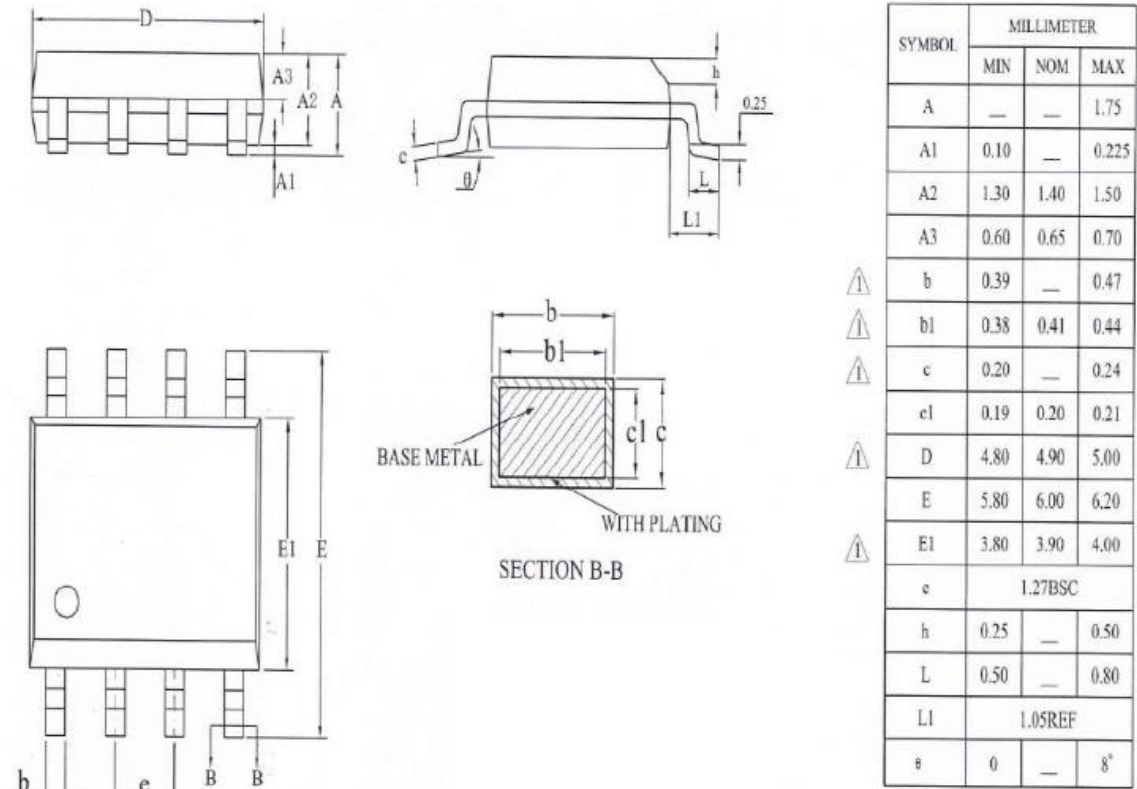


Figure 24-4 SOP8 封装尺寸

## 25 版本记录

版本	日期	描述
Ver1.00	2021-07-29	第一版
Ver1.01	2021-10-29	1. 完善部分描述； 2. HC89F0421A 的烧录脚开放双线方式，关闭 JTAG 方式； 3. RC32M 的全温度范围精度修改；
Ver1.02	2021-12-16	1.完善部分描述； 2.删除 HC89F0431A-QFN20 的封装； 3. RC32M 的全温度范围精度修改；

HOLYCHIP 公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。HOLYCHIP 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP 的产品不是专门设计来应用于外科植入、生命维持和任何 HOLYCHIP 产品产生的故障会对个体造成伤害甚至死亡的领域。如果将 HOLYCHIP 的产品用于上述领域，即使这些是由 HOLYCHIP 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证 HOLYCHIP 及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子

2021 年 7 月