

**HC89S105C8**

**HC89S105S8**

**HC89S105K8**

**数据手册**

**48/44/32 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 .....	7
1.3	系统框图 .....	8
1.4	引脚配置 .....	9
1.5	引脚描述 .....	12
1.6	外设功能引脚全映射模块 PTM .....	18
<b>2</b>	<b>CPU .....</b>	<b>20</b>
2.1	CPU 特性 .....	20
2.2	CPU 相关寄存器.....	20
<b>3</b>	<b>存储器 .....</b>	<b>22</b>
3.1	程序存储器（FLASH） .....	22
3.2	数据存储器（RAM） .....	27
3.3	特殊功能寄存器（SFR） .....	28
<b>4</b>	<b>系统时钟.....</b>	<b>31</b>
4.1	系统时钟特性 .....	31
4.2	系统时钟相关寄存器 .....	32
<b>5</b>	<b>电源管理.....</b>	<b>36</b>
5.1	电源管理特性 .....	36
5.2	空闲模式 .....	36
5.3	掉电模式 .....	36
5.4	电源管理相关寄存器 .....	37
<b>6</b>	<b>复位 .....</b>	<b>38</b>
6.1	复位特性 .....	38
6.2	POR(POWER-ON RESET) 复位 .....	38
6.3	BOR（BROWN-OUT RESET）复位.....	38
6.4	外部 RST 复位 .....	38
6.5	外部端口低压检测复位 .....	错误!未定义书签。
6.6	软件复位 .....	39
6.7	看门狗（WDT）复位 .....	39
6.8	复位相关寄存器 .....	40
<b>7</b>	<b>通用及复用 I/O.....</b>	<b>43</b>
7.1	通用及复用 I/O 特性 .....	43
7.2	I/O 模式.....	43
7.3	I/O 功能框图.....	44
7.4	I/O 端口相关寄存器 .....	44
7.5	外设功能引脚全映射控制 .....	54

<b>8</b>	<b>中断</b>	<b>56</b>
8.1	中断特性	56
8.2	中断汇总	57
8.3	中断向量	57
8.4	中断优先级	57
8.5	中断处理	错误!未定义书签。
8.6	外部中断	59
8.7	中断相关寄存器	59
<b>9</b>	<b>定时器/计数器</b>	<b>65</b>
9.1	定时器/计数器特性	65
9.2	定时器/计数器 Tx(x=0,1)	65
<b>10</b>	<b>脉宽调制 PWM</b>	<b>70</b>
10.1	PWM 特性	70
10.2	PWM 相关寄存器	71
<b>11</b>	<b>可编程计数阵列 PCA</b>	<b>81</b>
11.1	PCA 特性	81
11.2	PCA 工作模式	81
11.3	PCA 相关寄存器	84
<b>12</b>	<b>实时时钟 RTC</b>	<b>89</b>
12.1	RTC 特性	89
12.2	RTC 相关寄存器	90
<b>13</b>	<b>看门狗定时器 WDT</b>	<b>91</b>
13.1	WDT 特性	91
13.2	WDT 相关寄存器	91
<b>14</b>	<b>通用异步收发器 UART</b>	<b>94</b>
14.1	UART 特性	94
14.2	工作方式	94
14.3	波特率发生器	100
14.4	多机通信	101
14.5	帧出错检测	102
14.6	UART1 相关寄存器	102
14.7	UART2	105
<b>15</b>	<b>串行外部设备接口 SPI</b>	<b>109</b>
15.1	SPI 特性	109
15.2	SPI 信号描述	109
15.3	SPI 时钟速率	110
15.4	SPI 功能框图	110
15.5	SPI 工作模式	110
15.6	SPI 传送形式	112
15.7	SPI 出错检测	113
15.8	SPI 中断	113

15.9	SPI 配置对照.....	114
15.10	SPI 相关寄存器 .....	114
<b>16</b>	<b>IIC 总线.....</b>	<b>116</b>
16.1	IIC 特性.....	116
16.2	IIC 总线工作原理 .....	116
16.3	总线上数据的有效性 .....	117
16.4	总线上的信号 .....	117
16.5	总线上数据初始格式 .....	118
16.6	IIC 总线寻址约定 .....	119
16.7	主机向从机读写 1 个字节数据的过程.....	119
16.8	IIC 工作模式.....	121
16.9	IIC 总线相关寄存器.....	126
<b>17</b>	<b>模数转换 ADC.....</b>	<b>129</b>
17.1	ADC 特性.....	129
17.2	ADC 省电唤醒.....	129
17.3	ADC 相关寄存器.....	131
<b>18</b>	<b>低电压检测 LVD .....</b>	<b>135</b>
18.1	LVD 特性 .....	135
18.2	LVD 相关寄存器 .....	135
<b>19</b>	<b>循环冗余校验 CRC.....</b>	<b>137</b>
19.1	CRC 特性.....	137
19.2	CRC 相关寄存器 .....	138
<b>20</b>	<b>代码选项 .....</b>	<b>139</b>
<b>21</b>	<b>指令表 .....</b>	<b>140</b>
<b>22</b>	<b>电气特性 .....</b>	<b>145</b>
22.1	极限参数 .....	145
22.2	DC 特性.....	145
22.3	AC 特性.....	147
22.4	ADC 特性.....	147
22.5	FLASH 特性.....	148
22.6	BOR 检测电压特性 .....	148
22.7	LVD/PLVD 检测电压特性 .....	148
22.8	其他电气特性 .....	149
<b>23</b>	<b>开发工具 .....</b>	<b>150</b>
23.1	HC-LINK 仿真工具 .....	150
23.2	HC-PM51 烧录工具.....	150
23.3	软件下载 .....	150
<b>24</b>	<b>封装尺寸 .....</b>	<b>151</b>
24.1	LQFP48 .....	151
24.2	LQFP44 .....	151

---

24.3	LQFP32 .....	152
25	版本记录 .....	<b>153</b>

# 1 产品简介

HC89S105xx 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，内部最多 64K Bytes FLASH 程序存储器，256 Bytes IRAM 和 2K Bytes XRAM，6 组双向 I/O 口，1 个外设功能引脚全映射模块 PTM，2 个 16 位定时器/计数器，2 个 PCA 模块，3 组 12 位带死区控制互补 PWM，2 个串行通信口，1 个 IIC 通信口，1 个 SPI 通信口，1 个 RTC 电路，最多 23+2 路 12 位 ADC，四种系统工作模式（正常、低速、掉电和空闲）和 16 个中断源。

## 1.1 功能特性

- ◆ CPU
  - 增强型 1T 8051 内核
- ◆ ROM
  - 64K Bytes FLASH
  - IAP 和 ICP 操作
  - 灵活的代码保护模式
- ◆ RAM
  - 256 Bytes IRAM
  - 2K Bytes XRAM
- ◆ 时钟
  - 内部高精度 ( $\pm 1\%$ ) 32MHz RC
  - 外部高频晶振 4MHz—16MHz
  - 外部低频晶振 32.768KHz
  - 外设模块时钟可单独开关
  - 系统时钟最高可跑 16MHz
- ◆ 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
  - 外部管脚低电平复位
- ◆ I/O
  - 41/29 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
  - 2 级端口驱动电流可选
  - 外设功能引脚全映射模块 PTM
- ◆ 中断
  - 多达 16 个中断源
  - 4 级中断优先级
  - 15/13 个外部中断
- ◆ 定时器/计数器
  - 2 个 16 位 Timer(T0/T1)
  - 支持 16 位自动重载
- ◆ PWM
  - 最多 3 组 12 位带死区控制互补 PWM
  - 具有故障检测功能
- ◆ PCA
  - 16 位捕获/比较/软件定时
  - 8 位 PWM
- ◆ RTC
- ◆ 通讯模块
  - 2 个 UART
  - 1 个 SPI
  - 1 个 IIC
- ◆ ADC 检测电路
  - 支持最多 23+2 通道 12 位 ADC 检测
  - ADC 参考电压可选内部 Vref、外部 Vref
  - 具有省电唤醒功能
- ◆ 低电压检测模块
  - 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
  - 外部管脚电压 (1.2V) 检测，可中断
- ◆ 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—+85°C
- ◆ 封装类型
  - LQFP48
  - LQFP44
  - LQFP32

## ✓ 选型表

产品型号	ROM	RAM	I/O	Timer	PWM	A/D	INT	PCA	IIC	UART	SPI	WDT	Voltage	TEMP	封装形式
HC89S105C8	64K	256+2K	41	16bit*2	12bit*3 组	23+2	15	2	1	2	1	1	2.0~ 5.5V	-40~ +85°C	LQFP48
HC89S105S8	64K	256+2K	41	16bit*2	12bit*3 组	23+2	15	2	1	2	1	1	2.0~ 5.5V	-40~ +85°C	LQFP44
HC89S105K8	64K	256+2K	29	16bit*2	12bit*3 组	17+2	13	2	1	2	1	1	2.0~ 5.5V	-40~ +85°C	LQFP32

产品型号	Simulator	Programmer	Datasheet	DemoCode	DemoBoard
HC89S105xx	HC-LINK	HC-PM51/HC-ISP	√	√	√

## 1.2 应用注意

1. P4.7端口出厂时默认为复位管脚，可以通过配置代码选项将此口配置为普通IO引脚。
2. P4.7端口作为外部RST复位端口时，无法作为普通I/O使用。
3. 如果P4.7引脚为外部复位脚且引脚为复位电平，那么芯片处于外部复位状态，无法进行仿真或烧录。
4. 在 IAP 擦写和进入掉电模式之前，需要配置 `FREQ_CLK` 寄存器，指明目前 CPU 时钟的频率。
5. 如果需要进行FLASH IAP操作，请仔细阅读3.1.3.1的注意事项。
6. ADCEN 置 1 或切换转换通道后，建议延时 20us 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. 当使用 ADC 模块时，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。
8. 在进行 IAP 操作时，CPU 进入空闲模式，此时不响应任何中断。
9. HC89S105xx 上电复位后，在 8ms+option 选择的复位后等待时间（1/4/8/16ms）内，TMS（P2.3）、TDI（P2.2）、TCK（P2.1）端口模式为输入上拉，TDO（P2.4）端口强推挽输出一个不定态（0 或 1 电平），过了这段时间，这四个端口的模式会变为模拟输入。
10. 如果使用下降沿外部中断唤醒掉电模式，那么在进入掉电模式前必须使外部中断引脚上保持高电平，同时需要打开中断使能和中断允许寄存器。
11. 如果 RTCEN 为 1，RTC 模块就开始工作，在掉电模式下仍然会继续工作，当 RTC 计数溢出后，RTC 中断会唤醒掉电模式，但唤醒之后必须工作 30μs 后才能进入下一次掉电，如果不需要 RTC 在掉电模式下工作需要使 RTCEN 为 0。

### 1.3 系统框图

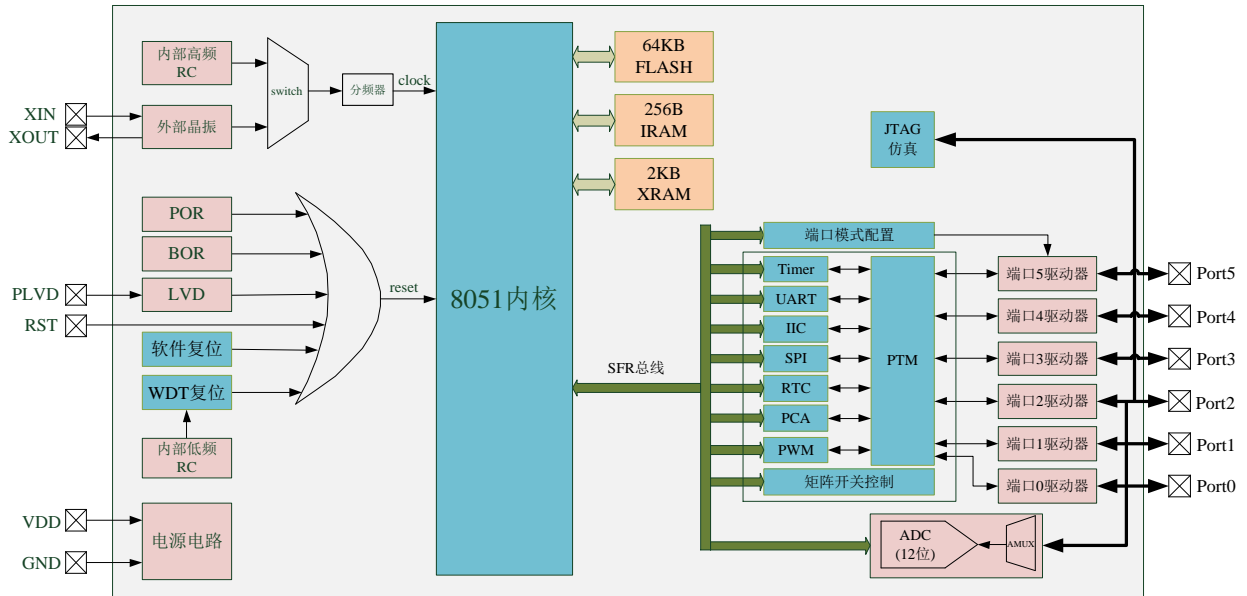


Figure 1-1 系统框图



## 1.4 引脚配置

### 1.4.1 HC89S105xx 引脚配置

#### 1.4.1.1 LQFP48引脚配置

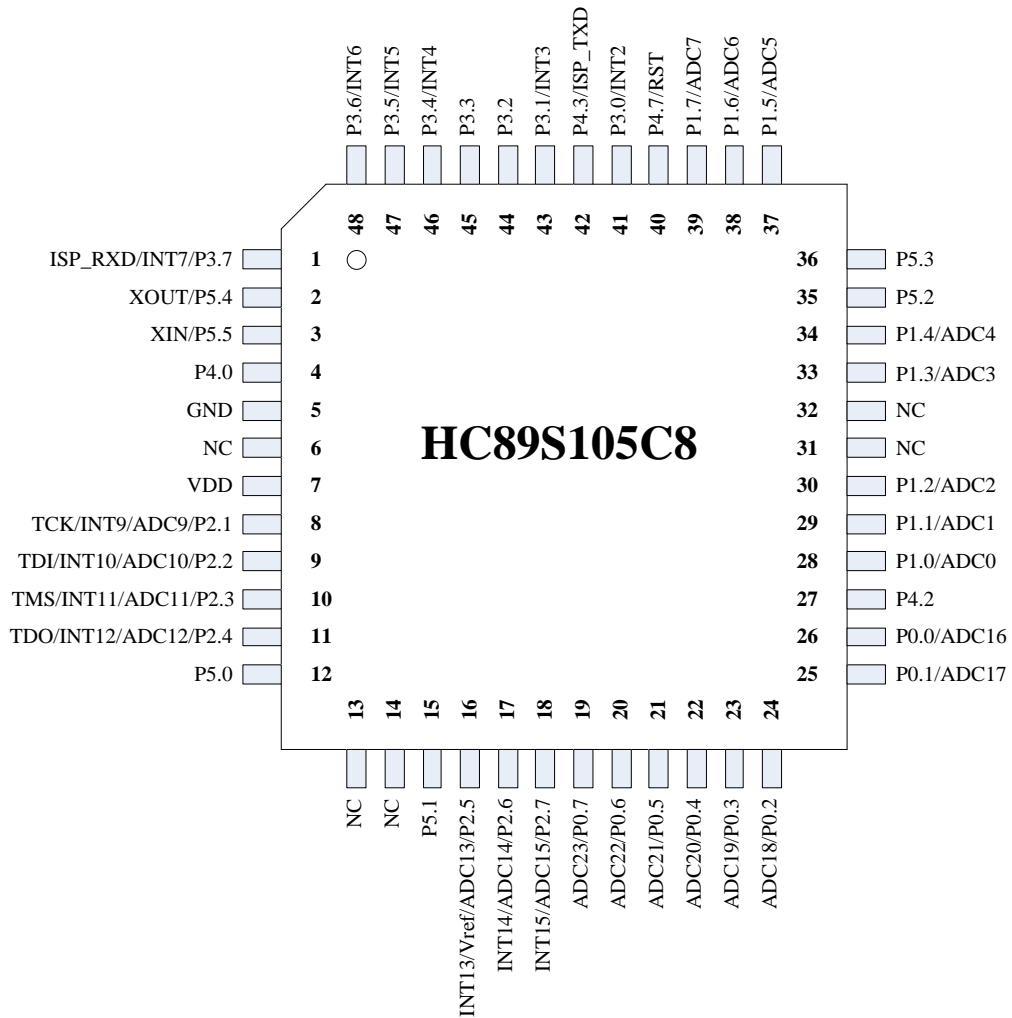


Figure 1-2 LQFP48 引脚配置图

1.4.1.2 LQFP44引脚配置

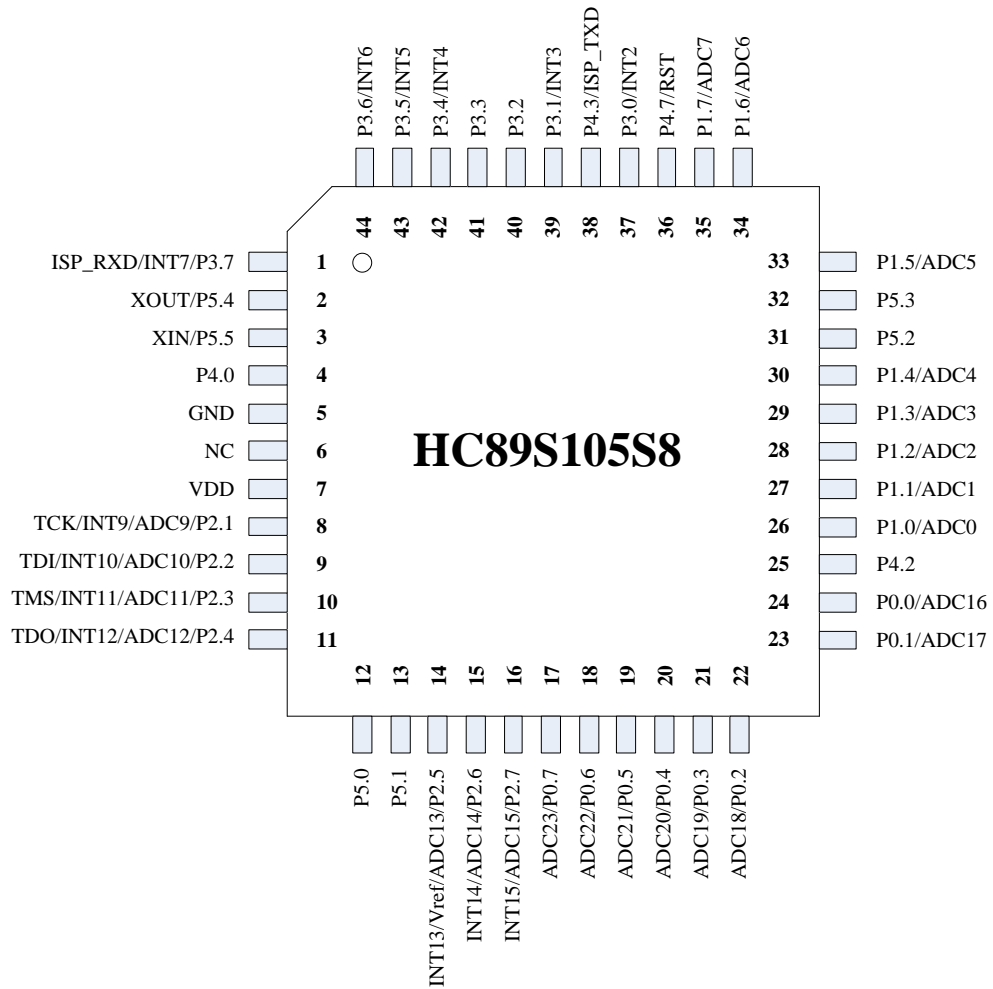


Figure 1-3 LQFP44 引脚配置图

1.4.1.3 LQFP32引脚配置

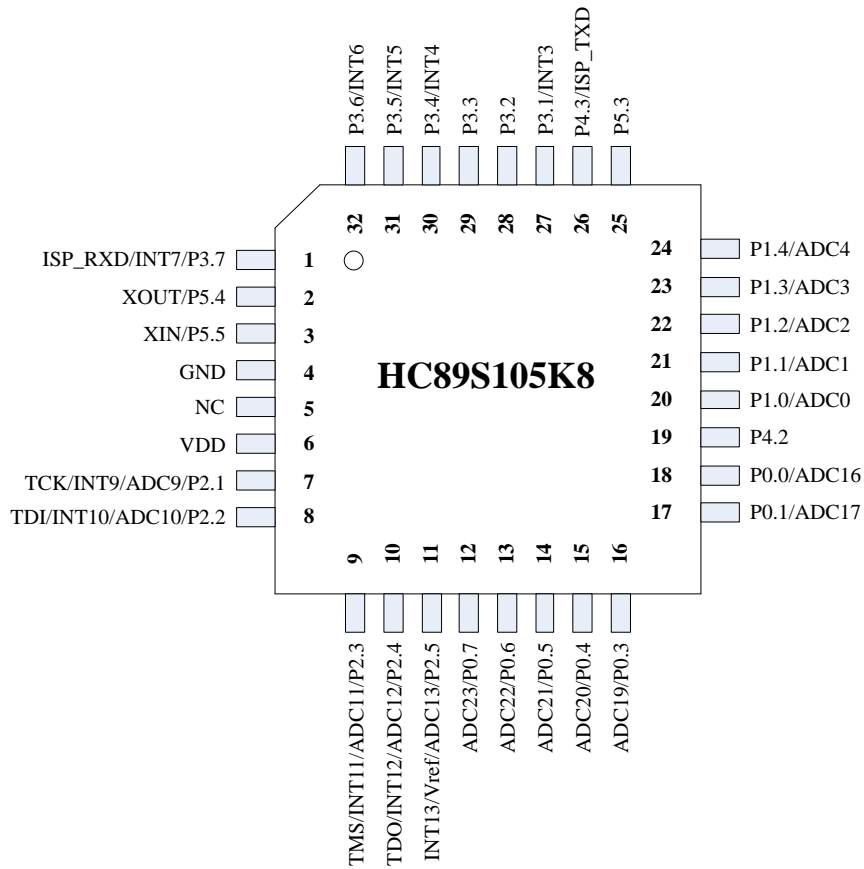


Figure 1-4 LQFP32 引脚配置图

## 1.5 引脚描述

### 1.5.1 HC89S105xx 引脚描述

#### 1.5.1.1 LQFP48引脚描述

脚位	名称	类型	说明
1	P3.7	I/O	输入/输出口
	INT7	I	外部中断 7 输入口
	ISP_RXD	I	ISP 下载 RXD 口（此口不影响 UART 正常使用）
2	P5.4	I/O	输入/输出口
	XOUT	AN	外部晶振输出口
3	P5.5	I/O	输入/输出口
	XIN	AN	外部晶振输入口
4	P4.0	I/O	输入/输出口
5	GND	P	电源地
6	NC	-	悬空脚
7	VDD	P	电源输入口
8	P2.1	I/O	输入/输出口
	ADC9	AN	ADC 输入口
	INT9	I	外部中断 9 输入口
	TCK	I	JTAG 时钟输入
9	P2.2	I/O	输入/输出口
	ADC10	AN	ADC 输入口
	INT10	I	外部中断 10 输入口
	TDI	I	JTAG 数据输入
10	P2.3	I/O	输入/输出口
	ADC11	AN	ADC 输入口
	INT11	I	外部中断 11 输入口
	TMS	I	JTAG 模式输入
11	P2.4	I/O	输入/输出口
	ADC12	AN	ADC 输入口
	INT12	I	外部中断 12 输入口
	TDO	O	JTAG 数据输出
12	P5.0	I/O	输入/输出口
13	NC	-	悬空脚
14	NC	-	悬空脚
15	P5.1	I/O	输入/输出口
16	P2.5	I/O	输入/输出口
	ADC13	AN	ADC 输入口
	INT13	I	外部中断 13 输入口
	Vref	AN	ADC 外部参考电压输入口
17	P2.6	I/O	输入/输出口
	ADC14	AN	ADC 输入口
	INT14	I	外部中断 14 输入口

18	P2.7 ADC15 INT15	I/O AN I	输入/输出口 ADC 输入口 外部中断 15 输入口
19	P0.7 ADC23	I/O AN	输入/输出口 ADC 输入口
20	P0.6 ADC22	I/O AN	输入/输出口 ADC 输入口
21	P0.5 ADC21	I/O AN	输入/输出口 ADC 输入口
22	P0.4 ADC20	I/O AN	输入/输出口 ADC 输入口
23	P0.3 ADC19	I/O AN	输入/输出口 ADC 输入口
24	P0.2 ADC18	I/O AN	输入/输出口 ADC 输入口
25	P0.1 ADC17	I/O AN	输入/输出口 ADC 输入口
26	P0.0 ADC16	I/O AN	输入/输出口 ADC 输入口
27	P4.2	I/O	输入/输出口
28	P1.0 ADC0	I/O AN	输入/输出口 ADC 输入口
29	P1.1 ADC1	I/O AN	输入/输出口 ADC 输入口
30	P1.2 ADC2	I/O AN	输入/输出口 ADC 输入口
31	NC	-	悬空脚
32	NC	-	悬空脚
33	P1.3 ADC3	I/O AN	输入/输出口 ADC 输入口
34	P1.4 ADC4	I/O AN	输入/输出口 ADC 输入口
35	P5.2	I/O	输入/输出口
36	P5.3	I/O	输入/输出口
37	P1.5 ADC5	I/O AN	输入/输出口 ADC 输入口
38	P1.6 ADC6	I/O AN	输入/输出口 ADC 输入口
39	P1.7 ADC7	I/O AN	输入/输出口 ADC 输入口
40	P4.7 RST	I/O I	输入/输出口 外部复位输入口
41	P3.0 INT2	I/O I	输入/输出口 外部中断 2 输入口
42	P4.3 ISP_TXD	I/O O	输入/输出口 ISP 下载 TXD 口（此口不影响 UART 正常使用）

43	P3.1	I/O	输入/输出口
	INT3	I	外部中断 3 输入口
44	P3.2	I/O	输入/输出口
45	P3.3	I/O	输入/输出口
46	P3.4	I/O	输入/输出口
	INT4	I	外部中断 4 输入口
47	P3.5	I/O	输入/输出口
	INT5	I	外部中断 5 输入口
48	P3.6	I/O	输入/输出口
	INT6	I	外部中断 6 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。

### 1.5.1.2 LQFP44引脚描述

脚位	名称	类型	说明
1	P3.7	I/O	输入/输出口
	INT7	I	外部中断 7 输入口
	ISP_RXD	I	ISP 下载 RXD 口（此口不影响 UART 正常使用）
2	P5.4	I/O	输入/输出口
	XOUT	AN	外部晶振输出口
3	P5.5	I/O	输入/输出口
	XIN	AN	外部晶振输入口
4	P4.0	I/O	输入/输出口
5	GND	P	电源地
6	NC	-	悬空脚
7	VDD	P	电源输入口
8	P2.1	I/O	输入/输出口
	ADC9	AN	ADC 输入口
	INT9	I	外部中断 9 输入口
9	TCK	I	JTAG 时钟输入
	P2.2	I/O	输入/输出口
	ADC10	AN	ADC 输入口
10	INT10	I	外部中断 10 输入口
	TDI	I	JTAG 数据输入
	P2.3	I/O	输入/输出口
11	ADC11	AN	ADC 输入口
	INT11	I	外部中断 11 输入口
	TMS	I	JTAG 模式输入
12	P2.4	I/O	输入/输出口
	ADC12	AN	ADC 输入口
	INT12	I	外部中断 12 输入口
13	TDO	O	JTAG 数据输出
	P5.0	I/O	输入/输出口
14	P5.1	I/O	输入/输出口
14	P2.5	I/O	输入/输出口
	ADC13	AN	ADC 输入口
	INT13	I	外部中断 13 输入口

	Vref	AN	ADC 外部参考电压输入口
15	P2.6	I/O	输入/输出口
	ADC14	AN	ADC 输入口
	INT14	I	外部中断 14 输入口
16	P2.7	I/O	输入/输出口
	ADC15	AN	ADC 输入口
	INT15	I	外部中断 15 输入口
17	P0.7	I/O	输入/输出口
	ADC23	AN	ADC 输入口
18	P0.6	I/O	输入/输出口
	ADC22	AN	ADC 输入口
19	P0.5	I/O	输入/输出口
	ADC21	AN	ADC 输入口
20	P0.4	I/O	输入/输出口
	ADC20	AN	ADC 输入口
21	P0.3	I/O	输入/输出口
	ADC19	AN	ADC 输入口
22	P0.2	I/O	输入/输出口
	ADC18	AN	ADC 输入口
23	P0.1	I/O	输入/输出口
	ADC17	AN	ADC 输入口
24	P0.0	I/O	输入/输出口
	ADC16	AN	ADC 输入口
25	P4.2	I/O	输入/输出口
26	P1.0	I/O	输入/输出口
	ADC0	AN	ADC 输入口
27	P1.1	I/O	输入/输出口
	ADC1	AN	ADC 输入口
28	P1.2	I/O	输入/输出口
	ADC2	AN	ADC 输入口
29	P1.3	I/O	输入/输出口
	ADC3	AN	ADC 输入口
30	P1.4	I/O	输入/输出口
	ADC4	AN	ADC 输入口
31	P5.2	I/O	输入/输出口
32	P5.3	I/O	输入/输出口
33	P1.5	I/O	输入/输出口
	ADC5	AN	ADC 输入口
34	P1.6	I/O	输入/输出口
	ADC6	AN	ADC 输入口
35	P1.7	I/O	输入/输出口
	ADC7	AN	ADC 输入口
36	P4.7	I/O	输入/输出口
	RST	I	外部复位输入口
37	P3.0	I/O	输入/输出口
	INT2	I	外部中断 2 输入口

38	P4.3 ISP_TXD	I/O O	输入/输出口 ISP 下载 TXD 口（此口不影响 UART 正常使用）
39	P3.1 INT3	I/O I	输入/输出口 外部中断 3 输入口
40	P3.2	I/O	输入/输出口
41	P3.3	I/O	输入/输出口
42	P3.4 INT4	I/O I	输入/输出口 外部中断 4 输入口
43	P3.5 INT5	I/O I	输入/输出口 外部中断 5 输入口
44	P3.6 INT6	I/O I	输入/输出口 外部中断 6 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。

### 1.5.1.3 LQFP32引脚描述

脚位	名称	类型	说明
1	P3.7	I/O	输入/输出口
	INT7	I	外部中断 7 输入口
	ISP_RXD	I	ISP 下载 RXD 口（此口不影响 UART 正常使用）
2	P5.4	I/O	输入/输出口
	XOUT	AN	外部晶振输出口
3	P5.5	I/O	输入/输出口
	XIN	AN	外部晶振输入口
4	GND	P	电源地
5	NC	-	悬空脚
6	VDD	P	电源输入口
7	P2.1	I/O	输入/输出口
	ADC9	AN	ADC 输入口
	INT9	I	外部中断 9 输入口
	TCK	I	JTAG 时钟输入
8	P2.2	I/O	输入/输出口
	ADC10	AN	ADC 输入口
	INT10	I	外部中断 10 输入口
	TDI	I	JTAG 数据输入
9	P2.3	I/O	输入/输出口
	ADC11	AN	ADC 输入口
	INT11	I	外部中断 11 输入口
	TMS	I	JTAG 模式输入
10	P2.4	I/O	输入/输出口
	ADC12	AN	ADC 输入口
	INT12	I	外部中断 12 输入口
	TDO	O	JTAG 数据输出
11	P2.5	I/O	输入/输出口
	ADC13	AN	ADC 输入口
	INT13	I	外部中断 13 输入口
	Vref	AN	ADC 外部参考电压输入口



12	P0.7 ADC23	I/O AN	输入/输出口 ADC 输入口
13	P0.6 ADC22	I/O AN	输入/输出口 ADC 输入口
14	P0.5 ADC21	I/O AN	输入/输出口 ADC 输入口
15	P0.4 ADC20	I/O AN	输入/输出口 ADC 输入口
16	P0.3 ADC19	I/O AN	输入/输出口 ADC 输入口
17	P0.1 ADC17	I/O AN	输入/输出口 ADC 输入口
18	P0.0 ADC16	I/O AN	输入/输出口 ADC 输入口
19	P4.2	I/O	输入/输出口
20	P1.0 ADC0	I/O AN	输入/输出口 ADC 输入口
21	P1.1 ADC1	I/O AN	输入/输出口 ADC 输入口
22	P1.2 ADC2	I/O AN	输入/输出口 ADC 输入口
23	P1.3 ADC3	I/O AN	输入/输出口 ADC 输入口
24	P1.4 ADC4	I/O AN	输入/输出口 ADC 输入口
25	P5.3	I/O	输入/输出口
26	P4.3 ISP_TXD	I/O O	输入/输出口 ISP 下载 TXD 口（此口不影响 UART 正常使用）
27	P3.1 INT3	I/O I	输入/输出口 外部中断 3 输入口
28	P3.2	I/O	输入/输出口
29	P3.3	I/O	输入/输出口
30	P3.4 INT4	I/O I	输入/输出口 外部中断 4 输入口
31	P3.5 INT5	I/O I	输入/输出口 外部中断 5 输入口
32	P3.6 INT6	I/O I	输入/输出口 外部中断 6 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。

## 1.6 外设功能引脚全映射模块 PTM

HC89S105xx 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 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 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
PCA	ECI	I	PCA 外部输入
	PCA0	I/O	PCA0 输入/输出口
	PCA1	I/O	PCA1 输入/输出口
PWM	FLT0	I	PWM0 故障检测输入口
	PWM0	O	PWM0 输出口
	PWM01	O	PWM01 输出口
	FLT1	I	PWM1 故障检测输入口
	PWM1	O	PWM1 输出口
	PWM11	O	PWM11 输出口
	FLT2	I	PWM2 故障检测输入口
	PWM2	O	PWM2 输出口
RTC	PWM21	O	PWM21 输出口
	RTCO	O	RTC 时钟分频输出
UART	TXD	O	UART 数据传输口
	RXD	I/O	UART 接收脚
	TXD2	O	UART2 数据传输口
	RXD2	I/O	UART2 接收脚
	BRTO	O	BRT 时钟分频输出
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

外部中断	INT0	I	外部中断 0
	INT1	I	外部中断 1

### 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)、ADC 输入口、Vref 引脚、外部晶振引脚、外部中断 2-15 输入口、外部复位端口 ( $\overline{\text{RST}}$ )。

## 2 CPU

### 2.1 CPU 特性

HC89S105xx 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	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	-	-	DPS

位编号	位符号	说明
7-1	-	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 64 位密码进行保护
- 在应用编程（IAP）支持用户自定义启动代码和类 EEPROM 区
- 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

#### 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 8 个字节（64 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 1K 字节为保护单位，当一个 1K 字节空间的读保护使能时，ICP 读这个 1K 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 1K 字节为保护单位，当对应 1K 字节的擦写保护使能时，ICP 将不能擦除和编程这个 1K 字节空间，强写也不允许。

若对应 1K 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 1K 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-LINK 用户手册。

##### 3.1.2.3 IAP读擦写FLASH保护

IAP 通过 MOVC 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 MOVC 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 MOVC 指令可以读取自身的数据。

IAP 擦写 FLASH 的步骤详见 FLASH IAP 操作章节，IAP 的擦写保护以 1K 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 1K 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 1K 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 FLASH IAP 操作

HC89S105xx 的 FLASH 一共有 512 个扇区，128 个字节为一个扇区，8\*128 Bytes = 1K Bytes 为一页，1K Bytes\*4 = 4KBytes 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 23μs，在 CPU\_CLK 为 2MHz 时，IAP 写一个字节的的时间是 37μs。在进行 IAP 擦写期间，CPU 进入空闲模式。

#### 3.1.3.1 IAP操作注意事项

HC89S105xx 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、IAP 操作只能对其他扇区进行读、擦、写，无法擦除 IAP 操作代码自身所在的扇区。
- 2、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。
- 3、在进行 IAP 操作时，CPU 进入空闲模式，此时不响应任何中断。
- 4、芯片的正常工作电压是 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。
- 5、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，可以有效的保证程序区不会被改写或误擦除。
- 6、为防止 MCU 上电瞬间电压不稳导致程序跑飞造成误操作，所以建议每次 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。
- 7、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。
- 8、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。

#### 3.1.3.2 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

### 3.1.3.3 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]							

#### IAP\_ADDRH

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[15:8]							

位编号	位符号	说明
7-0	IAP_ADDR[15:8]	IAP 操作时的地址寄存器高八位
7-0	IAP_ADDR[7:0]	IAP 操作时的地址寄存器低八位

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 0xFFFF。

### 3.1.3.4 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	0	0	0	0	0	0	0	0
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&IAP_ADDRH 指向 0xFF，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_ADDRH = 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_ADDRH&IAP_ADDRH 指向 0xFF,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;
    
```

### 3.1.4 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。

### 3.1.5 FLASH ICP 操作

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 5 个跳线将编程引脚（VDD、TDO、TDI、TMS、TCK）从应用电路中分离出来，如下图所示。

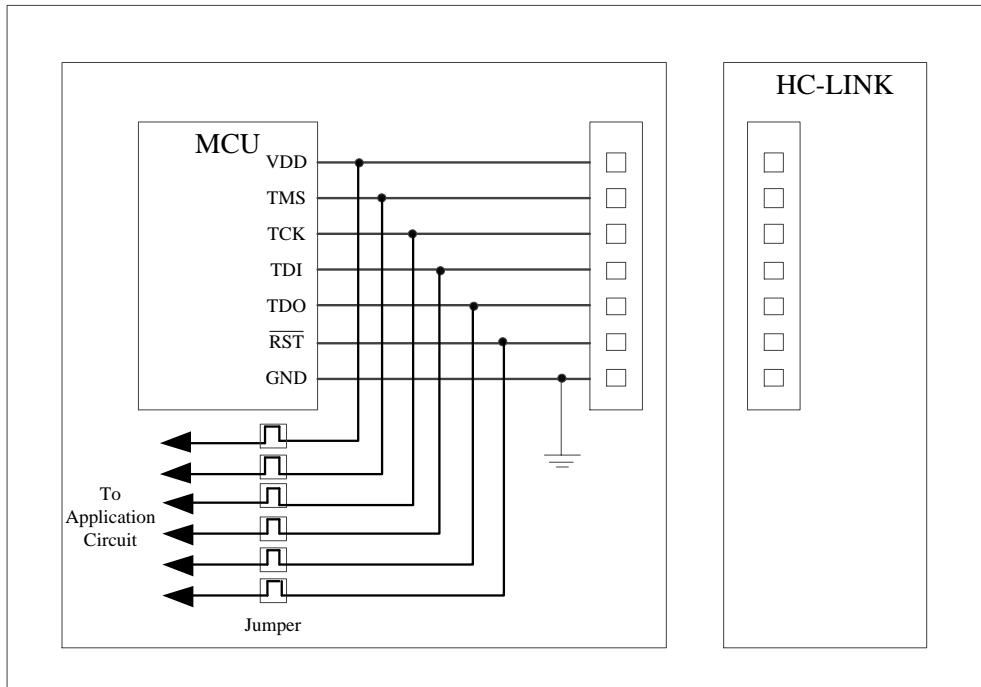


Figure 3-1 HC-LINK 编程硬件连接

### 3.2 数据存储器（RAM）

HC89S105xx 为用户提供了 256 Bytes 内部 RAM 和 2K Bytes 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 3-2 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 0x000~0x7FF，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	-	PWM2C	PWM2PL	PWM2PH	PWM2DL	PWM2DH	PWM2DT
E8	-	PWMFLT	PWM1C	PWM1PL	PWM1PH	PWM1DL	PWM1DH	PWM1DT
E0	ACC	PWMEN	PWM0C	PWM0PL	PWM0PH	PWM0DL	PWM0DH	PWM0DT
D8	-	-	-	-	-	-	-	-
D0	PSW	-	-	-	-	-	-	-
C8	P5	PCACLK	PCAMOD0	PCAMOD1	CCAPL0	CCAPH0	CCAPL1	CCAPH1
C0	P4	PCACON	PCACL	PCACH	-	-	-	-
B8	IE1	IP2	IP3	LVDC	RTCC	WDTC	CRCL	CRCH
B0	P3	IP4	-	-	ADCC0	ADCC1	ADCRL	ADCRH
A8	IE	IP0	IP1	SPDAT	SPCTL	SPSTAT	IICDAT	IICADR
A0	P2	-	-	INSCON	-	-	IICCON	IICSTA
98	SCON	SBUF	SADDR	SADEN	SBRTL	SBRTH	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 "HC89S105xx.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	-	0x0023	-	0x0033	PITS3
0x0004	PCA_PWM0	0x0014	-	0x0024	BORC	0x0034	-
0x0005	PCA_PWM1	0x0015	-	0x0025	BORDBC	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	LVDDBC	0x0037	-
0x0008	S2CON	0x0018	ADCWC0	0x0028	-	0x0038	PINTE0
0x0009	S2CON2	0x0019	ADCWC1	0x0029	-	0x0039	PINTE1
0x000A	S2BUF	0x001A	-	0x002A	RSTDBC	0x003A	-
0x000B	S2ADDR	0x001B	ADCC2	0x002B	-	0x003B	-
0x000C	S2ADEN	0x001C	PWM0DBC	0x002C	CLKPCKEN0	0x003C	-
0x000D	S2BRTH	0x001D	PWM1DBC	0x002D	CLKPCKEN1	0x003D	-
0x000E	S2BRTL	0x001E	PWM2DBC	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	P4M0	0x0030	-
0x0001	P0M1	0x0011	P2M1	0x0021	P4M1	0x0031	-
0x0002	P0M2	0x0012	P2M2	0x0022	-	0x0032	-
0x0003	P0M3	0x0013	P2M3	0x0023	P4M3	0x0033	-
0x0004	P0HPU	0x0014	P2HPU	0x0024	P4HPU	0x0034	-
0x0005	P0LPU	0x0015	P2LPU	0x0025	P4LPU	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	P1M0	0x0018	P3M0	0x0028	P5M0	0x0038	-
0x0009	P1M1	0x0019	P3M1	0x0029	P5M1	0x0039	-
0x000A	P1M2	0x001A	P3M2	0x002A	P5M2	0x003A	-
0x000B	P1M3	0x001B	P3M3	0x002B	-	0x003B	-
0x000C	P1HPU	0x001C	P3HPU	0x002C	P5HPU	0x003C	-
0x000D	P1LPU	0x001D	P3LPU	0x002D	P5LPU	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	INT0_MAP
0x0001	T1_MAP	0x0011	PWM01_MAP	0x0021	RXD_MAP	0x0031	INT1_MAP
0x0002	-	0x0012	FLT0_MAP	0x0022	SCL_MAP	0x0032	-
0x0003	-	0x0013	-	0x0023	SDA_MAP	0x0033	-
0x0004	RTCO_MAP	0x0014	PWM1_MAP	0x0024	$\overline{SS}$ _MAP	0x0034	-
0x0005	BRTO_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0x0006	-	0x0016	FLT1_MAP	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	ECl_MAP	0x001A	FLT2_MAP	0x002A	-	0x003A	-
0x000B	PCA0_MAP	0x001B	-	0x002B	-	0x003B	-
0x000C	PCA1_MAP	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

只读：

**扩展XSFR（基地址0xFFC0）**

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	SN_DATA0	0x0010	CHIP_ID0	0x0020	-	0x0030	-
0x0001	SN_DATA1	0x0011	CHIP_ID1	0x0021	-	0x0031	-
0x0002	SN_DATA2	0x0012	CHIP_ID2	0x0022	-	0x0032	-
0x0003	SN_DATA3	0x0013	CHIP_ID3	0x0023	-	0x0033	-
0x0004	SN_DATA4	0x0014	CHIP_ID4	0x0024	-	0x0034	-
0x0005	SN_DATA5	0x0015	CHIP_ID5	0x0025	-	0x0035	-
0x0006	SN_DATA6	0x0016	CHIP_ID6	0x0026	-	0x0036	-
0x0007	SN_DATA7	0x0017	CHIP_ID7	0x0027	-	0x0037	-
0x0008	ID_DATA0	0x0018	-	0x0028	-	0x0038	-
0x0009	ID_DATA1	0x0019	-	0x0029	-	0x0039	-
0x000A	ID_DATA2	0x001A	-	0x002A	-	0x003A	-
0x000B	ID_DATA3	0x001B	-	0x002B	-	0x003B	-
0x000C	ID_DATA4	0x001C	-	0x002C	-	0x003C	-
0x000D	ID_DATA5	0x001D	-	0x002D	-	0x003D	-
0x000E	ID_DATA6	0x001E	-	0x002E	-	0x003E	-
0x000F	ID_DATA7	0x001F	-	0x002F	-	0x003F	-

HC89S105xx 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，用户可以在程序中像读 XSFR 一样可以直接读出。

SN\_DATA 和 ID\_DATA 是用户自定义数据，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像读 XSFR 一样可以直接读出。

# 4 系统时钟

## 4.1 系统时钟特性

HC89S105xx 单片机系统时钟有 3 种时钟源可选：外部高频晶振时钟（4MHz~16MHz）、外部低频晶振时钟（32.768KHz）、内部高频 RC 时钟（32 MHz），内部低频 RC 时钟（38KHz）只能用作 WDT 的时钟，不作为系统时钟使用。其中，内部高频 RC 在-40°C ~+85°C 范围误差不超过 1%。系统选择后的时钟记做 Fosc，Fosc 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，即 Fcpu。

如果使用内部 32MHz RC 做系统时钟，必须通过分频将系统时钟降到 16MHz 或 16MHz 以下。如果系统时钟高于 16MHz 可能会导致系统工作不正常。

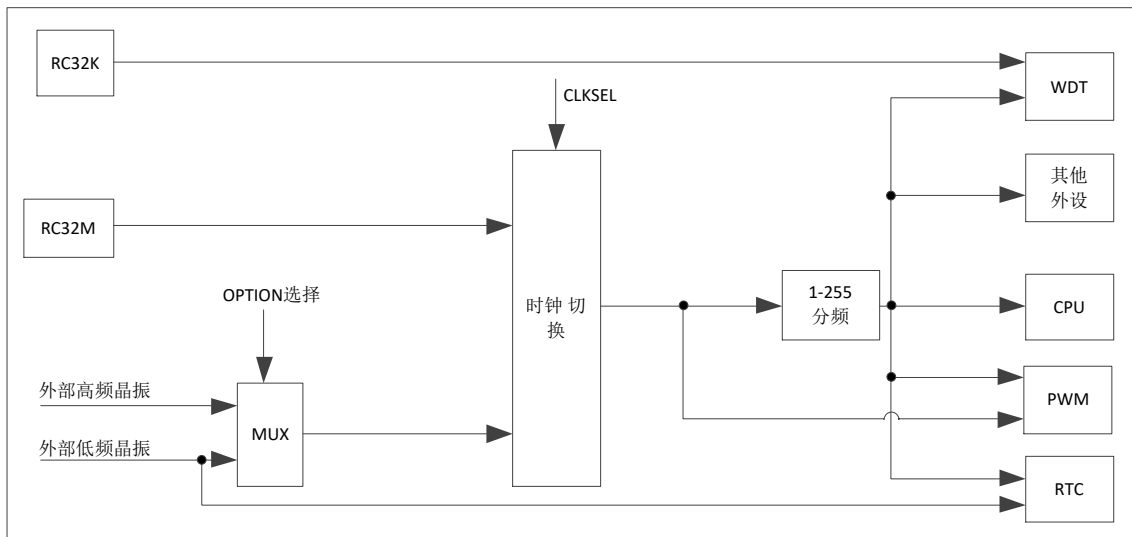


Figure 4-1 系统时钟框图

## 4.2 系统时钟相关寄存器

### 4.2.1 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	高速内部 RC 振荡器状态位 0: 高速内部 RC 未准备 1: 高速内部 RC 准备就绪 注: 该位硬件自动清 0 或置 1。
4	LSRCRDY	低速内部 RC 振荡器状态位 0: 低速内部 RC 未准备 1: 低速内部 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, 上电默认选择低频晶振, 如果想使用高频晶振, 需要配置代码选项。
1	HSRCEN	高速内部 RC 振荡器使能位 0: 高速内部 RC 关闭 1: 高速内部 RC 打开
0	-	保留位



## 4.2.2 时钟选择寄存器 CLKSWR

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

位编号	位符号	说明
7-6	CLKSTA[1:0]	系统时钟状态位 x0: 当前系统时钟为高速内部 RC 01: 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 系统据当前系统时钟自动切换各个状态
5	-	保留位
4	CLKSEL	系统时钟选择位 0: 系统时钟为高速内部 RC 1: 系统时钟为外部晶振 注: 1. 系统时钟选择时, CLKCON 寄存器中对应的时钟源状态位必须为 1, 否则将延续之前时钟。 2. 建议用户在切换系统时钟之前先打开要切换时钟的使能, 然后查看时钟源状态位是否为 1, 为 1 后再配置该位就可以完成时钟切换。
3-0	-	保留位

## 4.2.3 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	1	0	0	0	0
位符号	CLKDIV[7:0]							

位编号	位符号	说明
7-0	CLKDIV[7:0]	时钟分频系数, 默认为 16 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: 分频后的时钟即 $F_{CPU}$ 。

## 4.2.4 外设时钟门控寄存器 CLKPCKEN0、CLKPCKEN1

关闭未使用外设的时钟可降低功耗, 外设的时钟门控寄存器使用户可在运行模式下随时打开或关闭系统时钟与外设的连接。当用户关闭某个外设的时钟后, 此模块就会被禁止掉, 操作这个模块的寄存器都将不起作用。

系统复位后, 所有外设时钟均处于开的状态, 用户可通过清除 CLKPCKEN0 或 CLKPCKEN1 中的外设时钟控制位来关闭相应的外设时钟。

**CLKPCKEN0**

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	UART2_ CLKEN	UART1_ CLKEN	WDT_ CLKEN	RTC_ CLKEN	PWM_ CLKEN	PCA_ CLKEN	T1_ CLKEN	T0_ CLKEN

位编号	位符号	说明
7	UART2_CLKEN	UART2 时钟使能位 0: 禁止外设时钟与 UART2 连接 1: 使能外设时钟与 UART2 连接
6	UART1_CLKEN	UART1 时钟使能位 0: 禁止外设时钟与 UART1 连接 1: 使能外设时钟与 UART1 连接
5	WDT_CLKEN	WDT 时钟使能位 0: 禁止外设时钟与 WDT 连接 1: 使能外设时钟与 WDT 连接
4	RTC_CLKEN	RTC 时钟使能位 0: 禁止外设时钟与 RTC 连接 1: 使能外设时钟与 RTC 连接
3	PWM_CLKEN	PWM 时钟使能位 0: 禁止外设时钟与 PWM 连接 1: 使能外设时钟与 PWM 连接
2	PCA_CLKEN	PCA 时钟使能位 0: 禁止外设时钟与 PCA 连接 1: 使能外设时钟与 PCA 连接
1	T1_CLKEN	定时器 1 时钟使能位 0: 禁止外设时钟与定时器 1 连接 1: 使能外设时钟与定时器 1 连接
0	T0_CLKEN	定时器 0 时钟使能位 0: 禁止外设时钟与定时器 0 连接 1: 使能外设时钟与定时器 0 连接

**CLKPCKEN1**

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	1	1
位符号	-						SPI_CLKEN	IIC_CLKEN

位编号	位符号	说明
7-2	-	保留位
1	SPI_CLKEN	SPI 时钟使能位 0: 禁止外设时钟与 SPI 连接 1: 使能外设时钟与 SPI 连接
0	IIC_CLKEN	IIC 时钟使能位 0: 禁止外设时钟与 IIC 连接 1: 使能外设时钟与 IIC 连接

**4.2.5 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。建议在 IAP 擦写之前，将系统时钟频率分频为整数。当系统时钟频率低于 1MHz 时，不能进行 FLASH 的 IAP 擦写操作。

**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

## 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 (STOP), 作为省电模式
- 提供多种方式从空闲/掉电模式唤醒
- 提供低速模式 (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HC89S105xx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。HC89S105xx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)。HC89S105xx 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HC89S105xx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使能, WDT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。

将PCON寄存器中的PD位置1, 使HC89S105xx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, HC89S105xx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) 有效外部中断及RTC中断。在有效的外部中断和RTC中断发生后振荡器启动, 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。运行30 $\mu$ s后才能进入下一次掉电模式。

(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

HC89S105xx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。

HC89S105xx可以通过寄存器来选择BOR检测的电压档位，客户在使用时可根据情况选择合适的BOR检测电压。BOR档位：4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V

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。

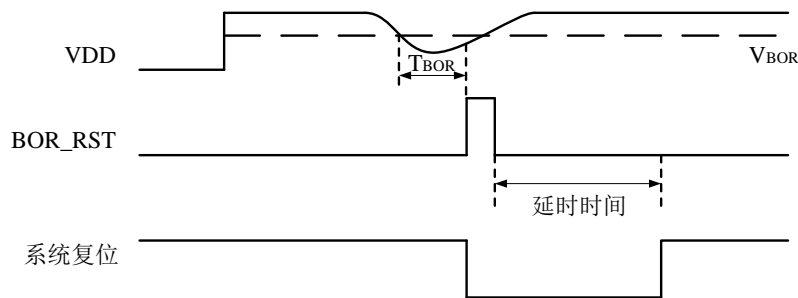


Figure 6-1 BOR 示意图

### 6.4 外部 RST 复位

外部RST引脚复位就是从外部向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，外部复位的有效电平也可以通过代码选项来设置，芯片出厂默认为高电平复位。

做RST端口时，将RST复位管脚拉高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RST复位管脚拉回低电平后，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：1、P4.7端口作为外部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端口且外部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 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 SWBF 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。

## 6.6 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，看门狗就会强迫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。

## 6.7 复位相关寄存器

### 6.7.1 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	R
POR复位	1	x	x	x	x	0	0	0
EXRST复位	u	1	u	u	u	0	0	0
BOR复位	u	u	1	u	u	0	0	0
WDT复位	u	u	u	1	u	0	0	0
软复位	u	u	u	u	1	0	0	0
位符号	PORF	EXRSTF	BORF	WDTRF	SWRF	-	-	-

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 WD 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
2-0	-	保留



### 6.7.2 BOR 电压检测控制寄存器 BORC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	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	-			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
5-3	-	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

### 6.7.3 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 BOR_DBC_EN 打开，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。

### 6.7.4 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。

## 7 通用及复用I/O

### 7.1 通用及复用 I/O 特性

- 提供 41/29 个双向 I/O 端口
- I/O 端口与其它复用功能端口共用，且部分复用功能端口可任意映射到任意 I/O 端口，原则上不允许多种复用功能端口映射到同一端口。ADC、INT2-15 等复用功能口不能映射。

### 7.2 I/O 模式

HC89S105xx 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。

另外，P3.0-P3.7 端口提供大灌电流能力，详见电气特性章节。

HC89S105xx 上电复位后，在 8ms+option 选择的复位后等待时间（1/4/8/16ms）内，TMS（P2.3）、TDI（P2.2）、TCK（P2.1）端口的模式为输入上拉，TDO（P2.4）端口强推挽输出一个不定态（0 或 1 电平），过了这段时间，这四个端口的模式会变为模拟输入。

如果 P4.7 被配置为复位脚，其端口模式是施密特输入上拉。

HC89S105xx 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。

HC89S105xx 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

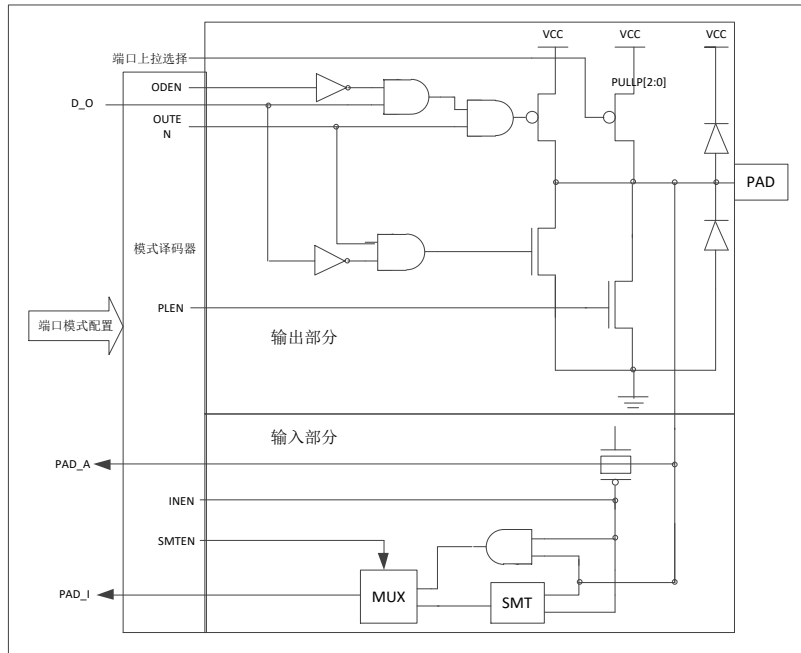


Figure 7-1 I/O 功能框图

### 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

注：对于 HC89S105K8，不能操作 P0[2]。

#### 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/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	P1[7:0]							

位编号	位符号	说明
7-0	P1[7:0]	P1 端口数据寄存器

注：对于 HC89S105K8，只能操作 P1[4:0]，其它请不要操作。

### 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[7:1]							-

位编号	位符号	说明
7-1	P2[7:1]	P2 端口数据寄存器
0	-	保留位

注：对于 HC89S105C8/S8，只能操作 P2[7:1]，其它请不要操作。

注：对于 HC89S105K8，只能操作 P2[5:1]，其它请不要操作。

### 7.4.4 P3 端口数据寄存器 P3

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	P3[7:0]							

位编号	位符号	说明
7-0	P3[7:0]	P3 端口数据寄存器

### 7.4.5 P4 端口数据寄存器 P4

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	P4[3:2]		-	P4[0]

位编号	位符号	说明
7-4	-	保留位
3-2	P4[3:2]	P4 端口数据寄存器
1	-	保留位
0	P4[0]	P4 端口数据寄存器

注：对于 HC89S105C8/S8，只能操作 P4[0]和 P4[3:2]，其它请不要操作。

注：对于 HC89S105K8，只能操作 P4[2]，其它请不要操作。

### 7.4.6 P5 端口数据寄存器 P5

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	P5[5:0]							

位编号	位符号	说明
7-6	-	保留位
5-0	P5[5:0]	P5 端口数据寄存器

注：对于 HC89S105K8，只能操作 P5[5:3]，其它请不要操作。

## 7.4.7 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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010 : 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011 : 保留 1100 : 推挽输出 (IO 驱动电流 Low Drive Mode)

		1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110: 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111: 保留
--	--	-------------------------------------------------------------------------------------------

注: 对于 HC89S105K8, 不能操作 P0M1[3:0]。

## 7.4.8 P1 端口功能选择寄存器 P1M0、P1M1、P1M2、P1M3

### 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]			

### P1M1

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	P13M[3:0]				P12M[3:0]			

### P1M2

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	P15M[3:0]				P14M[3:0]			

### P1M3

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	P17M[3:0]				P16M[3:0]			

位编号	位符号	说明
7-4 3-0	P1xM[3:0] (x = 0...7)	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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010: 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011: 保留 1100: 推挽输出 (IO 驱动电流 Low Drive Mode)

		1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110: 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111: 保留
--	--	-------------------------------------------------------------------------------------------

注: 对于 HC89S105K8, 不能操作 P1M2[7:4]以及 P1M3[7:0]。

## 7.4.9 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	R	R	R
复位值	0	0	1	1	0	0	1	1
位符号	P21M[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]			

位编号	位符号	说明
7-4 3-0	P2xM[3:0] (x = 1...7)	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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010: 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011: 保留 1100: 推挽输出 (IO 驱动电流 Low Drive Mode)



		1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110: 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111: 保留
--	--	-------------------------------------------------------------------------------------------

注: 对于 HC89S105C8/S8, 不能操作 P2M0[3:0]。

注: 对于 HC89S105K8, 不能操作 P2M0[3:0]和 P2M3[7:0]。

### 7.4.10 P3 端口功能选择寄存器 P3M0、P3M1、P3M2、P3M3

#### P3M0

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	P31M[3:0]				P30M[3:0]			

#### P3M1

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	P33M[3:0]				P32M[3:0]			

#### P3M2

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	P35M[3:0]				P34M[3:0]			

#### P3M3

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	P37M[3:0]				P36M[3:0]			

位编号	位符号	说明
7-4 3-0	P3xM[3:0] (x = 0...7)	P3.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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010: 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011: 保留

		1100 : 推挽输出 (IO 驱动电流 Low Drive Mode) 1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110 : 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111 : 保留
--	--	-------------------------------------------------------------------------------------------------------------------------------------

### 7.4.11 P4 端口功能选择寄存器 P4M0、P4M1

#### P4M0

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	1	1	0	0	1	1
位符号	-				P40M[3:0]			

#### P4M1

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	P43M[3:0]				P42M[3:0]			

位编号	位符号	说明
7-4 3-0	P4xM[3:0] (x = 0...4)	P4.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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010 : 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011 : 保留 1100 : 推挽输出 (IO 驱动电流 Low Drive Mode) 1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110 : 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111 : 保留

注: 对于 HC89S105C8/S8, 只能操作 P4M0[3:0]和 P4M1[7:0], 其它请不要操作。

注: 对于 HC89S105K8, 只能操作 P4M1[3:0], 其它请不要操作。

## 7.4.12 P5 端口功能选择寄存器 P5M0、P5M1、P5M2

### P5M0

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	P51M[3:0]				P50M[3:0]			

### P5M1

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	P53M[3:0]				P52M[3:0]			

### P5M2

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	P55M[3:0]				P54M[3:0]			

位编号	位符号	说明
7-4 3-0	P5xM[3:0] (x = 0...5)	P5.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 1000: 推挽输出 (IO 驱动电流 High Drive Mode) 1001: 开漏输出 (IO 驱动电流 High Drive Mode) 1010 : 开漏带上拉输出 (IO 驱动电流 High Drive Mode) 1011 : 保留 1100 : 推挽输出 (IO 驱动电流 Low Drive Mode) 1101: 开漏输出 (IO 驱动电流 Low Drive Mode) 1110 : 开漏带上拉输出 (IO 驱动电流 Low Drive Mode) 1111 : 保留

注：对于 HC89S105K8，只能操作 P5M1[7:4]，其它请不要操作。

### 7.4.13 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/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	P03PU[1:0]		P02PU[1:0]		P01PU[1:0]		P00PU[1:0]	

注：对于 HC89S105K8，不能操作 P0LPU[5:4]。

#### P0HPU

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	P07PU[1:0]		P06PU[1:0]		P05PU[1:0]		P04PU[1:0]	

#### P1LPU

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	P13PU[1:0]		P12PU[1:0]		P11PU[1:0]		P10PU[1:0]	

#### P1HPU

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	P17PU[1:0]		P16PU[1:0]		P15PU[1:0]		P14PU[1:0]	

注：对于 HC89S105K8，只能操作 P1HPU[1:0]，其它请不要操作。

#### P2LPU

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	P23PU[1:0]		P22PU[1:0]		P21PU[1:0]		-	-

#### P2HPU

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	P27PU[1:0]		P26PU[1:0]		P25PU[1:0]		P24PU[1:0]	

注：对于 HC89S105K8，只能操作 P2HPU[3:0]，其它请不要操作。

#### P3LPU

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	P33PU[1:0]		P32PU[1:0]		P31PU[1:0]		P30PU[1:0]	

**P3HPU**

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	P37PU[1:0]		P36PU[1:0]		P35PU[1:0]		P34PU[1:0]	

**P4LPU**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	P43PU[1:0]		P42PU[1:0]		-		P40PU[1:0]	

注：对于 HC89S105C8/S8，只能操作 P4LPU[1:0]和 P4LPU [7:4]，其它请不要操作。

注：对于 HC89S105K8，只能操作 P4LPU [5:4]，其它请不要操作。

**P5LPU**

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	P53PU[1:0]		P52PU[1:0]		P51PU[1:0]		P50PU[1:0]	

注：对于 HC89S105K8，只能操作 P5LPU [7:6]，其它请不要操作。

**P5HPU**

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	-	-	-	-	P55PU[1:0]		P54PU[1:0]	

位编号	位符号	说明
7-6, 5-4, 3-2, 1-0	PxyPU[1:0] (x = 0,1,2,3,4,5) (y = 0...7)	端口上拉电阻选择位 00: 30 kΩ 01: 50 kΩ 10: 120kΩ 11: 230 kΩ 注：阻值为 VDD =5V 时参考值。

## 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	0xFF90	PWM0_MAP	0xFFA0	TXD_MAP	0xFFB0	INT0_MAP
0xFF81	T1_MAP	0xFF91	PWM01_MAP	0xFFA1	RXD_MAP	0xFFB1	INT1_MAP
0xFF82	-	0xFF92	FLT0_MAP	0xFFA2	SCL_MAP	0xFFB2	-
0xFF83	-	0xFF93	-	0xFFA3	SDA_MAP	0xFFB3	-
0xFF84	RTCO_MAP	0xFF94	PWM1_MAP	0xFFA4	$\overline{SS}$ _MAP	0xFFB4	-
0xFF85	BRTO_MAP	0xFF95	PWM11_MAP	0xFFA5	SCK_MAP	0xFFB5	-
0xFF86	-	0xFF96	FLT1_MAP	0xFFA6	MOSI_MAP	0xFFB6	-
0xFF87	-	0xFF97	-	0xFFA7	MISO_MAP	0xFFB7	-
0xFF88	-	0xFF98	PWM2_MAP	0xFFA8	TXD2_MAP	0xFFB8	-
0xFF89	-	0xFF99	PWM21_MAP	0xFFA9	RXD2_MAP	0xFFB9	-
0xFF8A	ECI_MAP	0xFF9A	FLT2_MAP	0xFFAA	-	0xFFBA	-
0xFF8B	PCA0_MAP	0xFF9B	-	0xFFAB	-	0xFFBB	-
0xFF8C	PCA1_MAP	0xFF9C	-	0xFFAC	-	0xFFBC	-
0xFF8D	-	0xFF9D	-	0xFFAD	-	0xFFBD	-
0xFF8E	-	0xFF9E	-	0xFFAE	-	0xFFBE	-
0xFF8F	-	0xFF9F	-	0xFFAF	-	0xFFBF	-

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/W	R/W	R/W	R/W
复位值	0	0	1	1	1	1	1	1
位符号	-		FPORT[2:0]			FPIN[2:0]		

位编号	位符号	说明
7-6	-	保留位
5-3	FPORT[2:0]	映射端口选择 000: P0 001: P1 010: P2 011: P3 100: P4 101: P5
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 0xFF，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 P3.1 和 P3.2 上，用户在启动 UART1 的功能前，应该配置下面命令：

```
TXD_MAP = 0x19;    //TXD-->P3.1
```

```
RXD_MAP = 0x1A;    //RXD-->P3.2
```

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 P4.4 和 P4.5 上，那用户需要进行如下的配置：

```
TXD_MAP = 0x14;    //TXD-->P4.4
```

```
RXD_MAP = 0x15;    //RXD-->P4.5
```

多个输出映射到一个端口上时，复用功能默认的优先级：

优先级顺序	复用端口功能
1	T0
2	T1
3	RTCO
4	BRTO
5	PCA0
6	PCA1
7	PWM0
8	PWM01
9	PWM1
10	PWM11
11	PWM2
12	PWM21
13	TXD
14	RXD
15	SCL
16	SDA
17	SCK
18	MOSI
19	MISO
20	TXD2
21	RXD2

比如：RTCO\_MAP 低 6 位配置为 000001 选择 P0.1 口作为 RTCO 的输出口，BRTO\_MAP 的低 6 位也配置为 000001，这个时候硬件会按上面的优先级，P0.1 将配置为 RTCO 的输出口，而 BRTO\_MAP 的配置无效。

当所有的端口映射控制寄存器低 6 位都不等于 000001 时，即所有的功能口都不选择 P0.1 作为输出口，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 低 6 位配置为 010011，则选择 P2.3 作为 T0 的输入口，FLT0\_MAP 低 6 位也配置为 010011，这样从 P2.3 端口进入的信号也会作用于 FLT0。

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

- 16 个中断源
- 4 级中断优先级
- 12 个外部中断

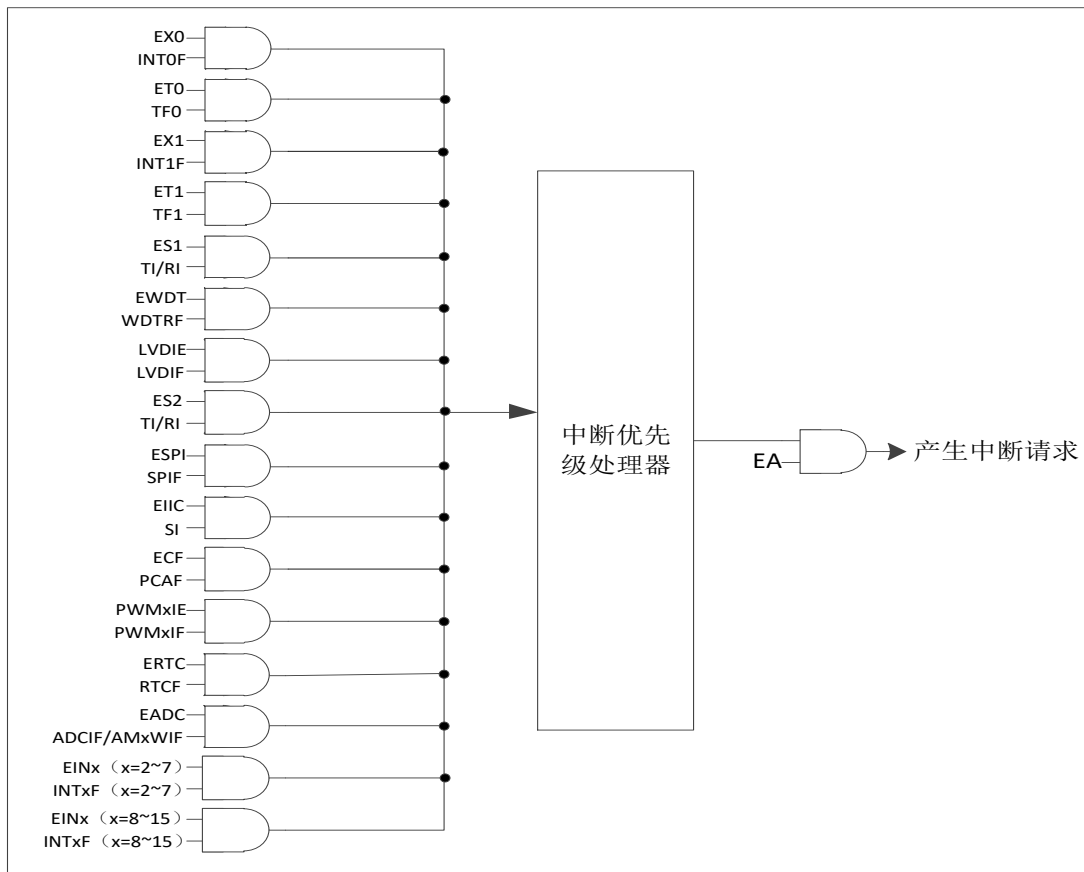


Figure 8-1 中断功能框图



## 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/模式冲突	9	8
IIC	004BH	EIIC	SI	10	9
PCA	0053H	ECF	PCAF	11	10
PWM	005BH	PWM <sub>x</sub> IE (x=0/1/2)	PWM <sub>x</sub> IF (x=0/1/2)	12	11
RTC	0063H	ERTC	RTCF	13	12
ADC	0073H	EADC	ADCIF/AM <sub>x</sub> WIF (x=0,1)	14	14
INT2-INT7	007BH	EIN <sub>x</sub> (x=2...7)	INT <sub>x</sub> F (x=2...7)	15	15
INT9-INT15	0083H	EIN <sub>x</sub> (x=9...15)	INT <sub>x</sub> F (x=9...15)	16	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，否则不响应任何中断。

## 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。中断向量的入口地址详见中断汇总章节。

## 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

HC89S105xx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9~15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。

外部中断引脚都是每个CPU时钟周期采样一次，如果外部中断为下降沿触发，外部中断源应当将中断输入脚至少保持1个CPU时钟周期高电平，然后至少保持1个CPU时钟周期低电平。这样就确保了边沿能够被检测到以使对应标志位置1。当调用中断服务程序后，外部中断0~1标志会被硬件自动清0，但外部中断2~15标志必须软件清除。

上升沿及双沿中断触发与下降沿触发类似。

如果外部中断为低电平触发，外部中断源必须一直保持请求有效，直到产生所请求的中断为止，此过程需要2个系统时钟周期。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。

如果使用下降沿外部中断唤醒掉电模式，那么在进入掉电模式前必须使外部中断引脚上保持高电平，同时需要打开中断使能和中断允许寄存器。

## 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 INT0 中断 1: 允许 INT0 中断

**IE1**

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	-	EX9_15	EX2_7	EADC	-	ERTC	EIIC	ESPI

位编号	位符号	说明
7	-	保留位
6	EX9_15	外部中断 9~15 中断允许位 0: 禁止 INT9~INT15 中断 1: 允许 INT9~INT15 中断 注: INT9~INT15 共用同一中断向量。
5	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。
4	EADC	A/D 转换完成中断允许位 0: 禁止 A/D 转换完成中断 1: 允许 A/D 转换完成中断
3	-	保留位
2	ERTC	RTC 中断允许位 0: 禁止 RTC 中断 1: 允许 RTC 中断
1	EIIC	IIC 中断允许位 0: 禁止 IIC 中断 1: 允许 IIC 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]		PWDT[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]		PPCA[1:0]		PIIC[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	PPCA [1:0]	PCA 中断优先级控制位
3-2	PIIC [1:0]	IIC 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	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PX2_7[1:0]		PADC[1:0]		-		PRTC[1:0]	

位编号	位符号	说明
7-6	PX2_7[1:0]	INT2_7 中断优先级控制位
5-4	PADC[1:0]	ADC 中断优先级控制位
3-2	-	保留位
1-0	PRTC [1:0]	RTC 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	-						PX9_15[1:0]	

位编号	位符号	说明
7-2	-	保留位
1-0	PX9_15 [1:0]	INT9_15 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 PITS0、PITS1、PITS2、PITS3

#### 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	R
复位值	0	0	0	0	0	0	0	0
位符号	IT11[1:0]		IT10[1:0]		IT9[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...7)	外部中断触发沿选择位 00: 低电平中断 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 PINTE0、PINTE1**
**PINTE0**

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	EINTx (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 EINTx(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)

**PINTE1**

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	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	-

位编号	位符号	说明
7-0	EINTx (x = 9...15)	外部中断控制位(INT9~INT15) 0: 禁止该端口中断 1: 允许该端口中断 注: 只要相应的 EINTx(x = 9...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 PINTF0、PINTF1

### 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
复位值	0	0	0	0	0	0	0	0
位符号	INT15F	INT14F	INT13F	INT12F	INT11F	INT10F	INT9F	-

位编号	位符号	说明
7-0	INTxF (x = 9...15)	INT9-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



# 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

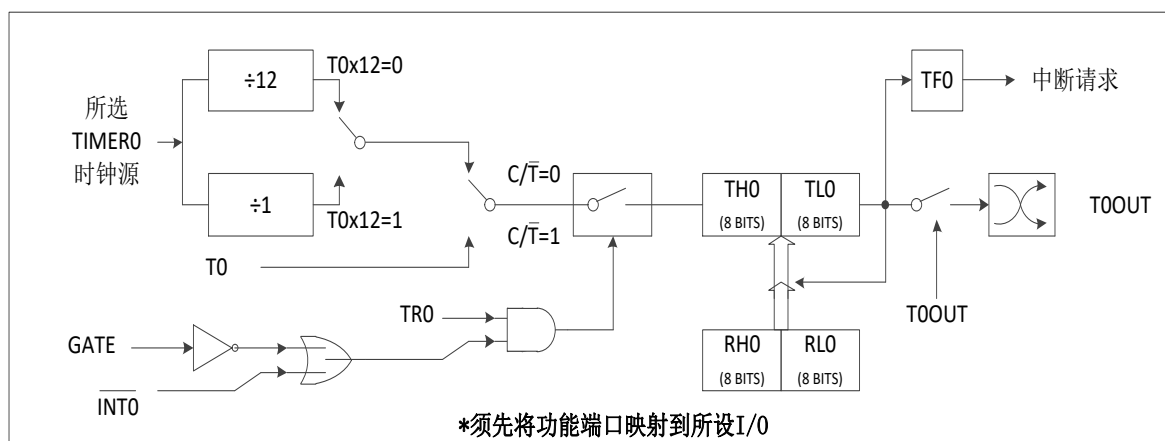


Figure 9-1 TIMER0 方式 0 功能框图

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 & 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 & 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

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  $TH_x$  &  $TL_x(x=0,1)$  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

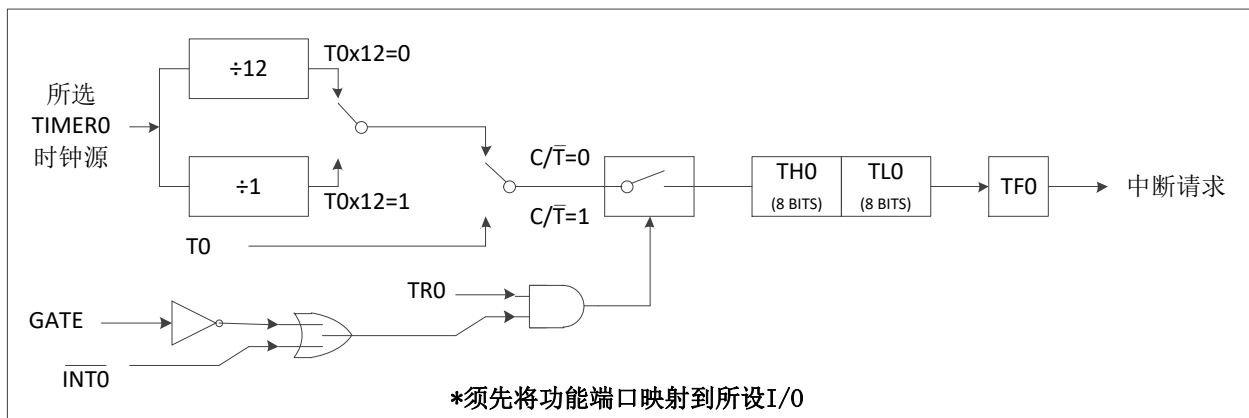


Figure 9-2 TIMER0 方式 1 功能框图

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TRx置1，定时器寄存器将从上次TRx(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

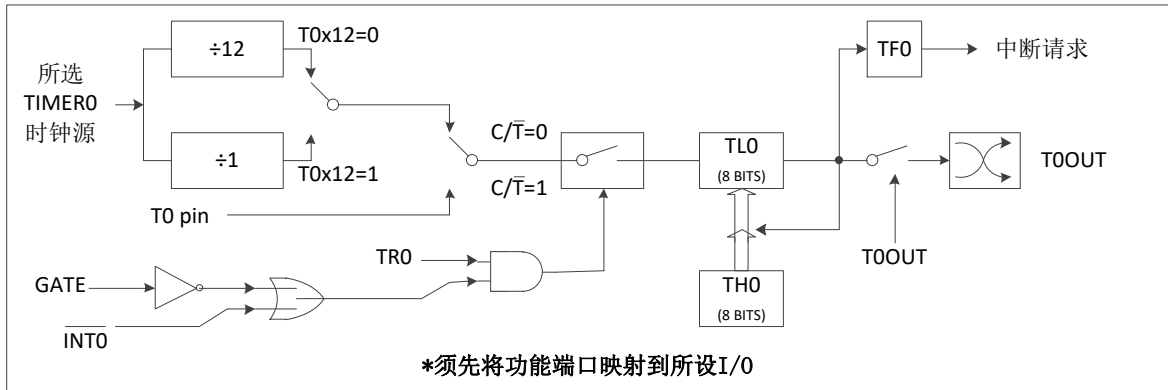


Figure 9-3 TIMER0 方式 2 功能框图

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)中的计数器递增至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)

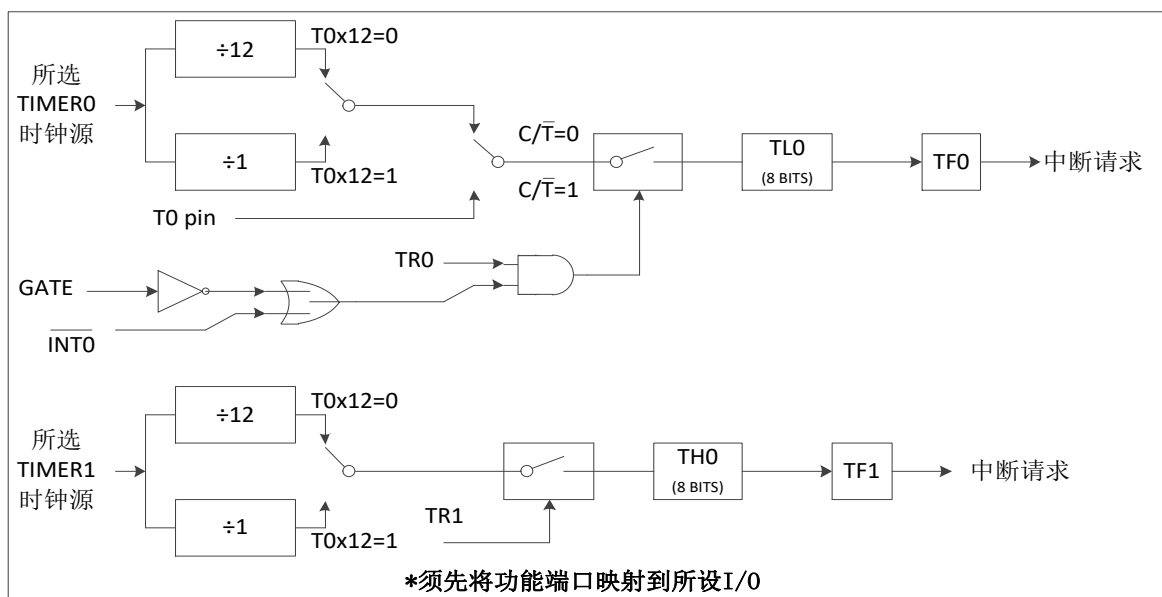


Figure 9-4 TIMER0 方式 3 功能框图

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/T0, GATE0和TFO。TLO能用系统时钟或外

部输入信号作为时钟源。

TH0只能用作定时器功能，时钟源来自系统时钟。TH0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。可以用来产生串口的波特率。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。

当作为定时器应用时，可配置寄存器 TCON1 中的 T0OUT 位使定时器 T0 溢出时 T0 脚自动翻转。

## 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	TF <sub>x</sub> (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	TR <sub>x</sub> (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 <sub>CPU</sub> /12 1: Tx 定时器时钟为 F <sub>CPU</sub>

### 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	C/T1	M1[1:0]		GATE0	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	C/T <sub>x</sub> (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 位自动重装初值定时器/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 T <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 T <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

# 10 脉宽调制PWM

## 10.1 PWM 特性

- 3 组带死区互补 PWM 或 3 路独立 PWM 输出
- 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
- 提供出错侦测功能可紧急关闭 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

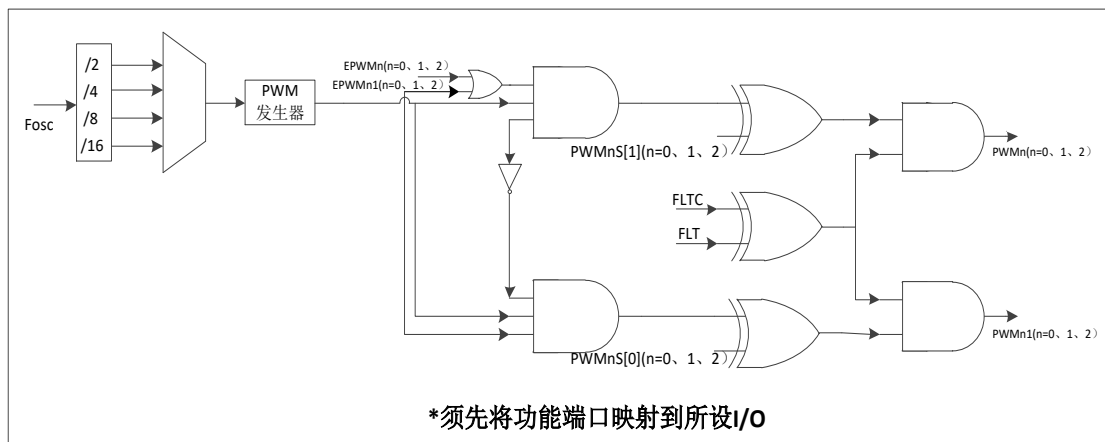


Figure 10-1 PWM 功能框图

HC89S105xx 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 EPWM0 或 EPWM01 来控制，只要它们中的任何一个使能，都可以启动计数器，计数器的时钟源通过 PWM0C 控制寄存器里的 CK0 来选择。

如果 EPWM0 使能了，而没有通过功能引脚映射寄存器进行 PWM0 的映射，这样也不会从芯片管脚上输出 PWM0，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。

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 3 路单独的 PWM 输出。

如果 EFLT0/EFLT1/EFLT2 置 1，PWM0/PWM1/PWM2 输出和其互补输出可由 FLT0/FLT1/FLT2 引脚输入信号变化自动关闭。一旦检测到 FLT 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 FLT 引脚错误去除后继续 PWM 输出。在 FLT 输入信号有效期间，FLTS 位无法清除。只有当 FLT 输入信号消失后，才能软件清除 FLTS 状态位，此时 PWM 恢复正常输出。

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 相关寄存器

### 10.2.1 PWMMEN

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	PWM0FLT_EN	EPWM21	EPWM11	EPWM01	-	EPWM2	EPWM1	EPWM0

位编号	位符号	说明
7	PWM0FLT_EN	PWM0 故障检测使能位 0: 禁止故障检测 1: 允许故障检测, 需要先进行 FLT0 的引脚映射和 IO 模式设置
6	EPWM21	PWM21 使能控制位 0: 禁止 PWM21 输出 1: 允许 PWM21 输出, 同时需要配置 PWM21 的引脚映射
5	EPWM11	PWM11 使能控制位 0: 禁止 PWM11 输出 1: 允许 PWM11 输出, 同时需要配置 PWM11 的引脚映射
4	EPWM01	PWM01 使能控制位 0: 禁止 PWM01 输出 1: 允许 PWM01 输出, 同时需要配置 PWM01 的引脚映射
3	-	保留位
2	EPWM2	PWM2 使能控制位 0: 禁止 PWM2 输出 1: 允许 PWM2 输出, 同时需要配置 PWM2 的引脚映射
1	EPWM1	PWM1 使能控制位 0: 禁止 PWM1 输出 1: 允许 PWM1 输出, 同时需要配置 PWM1 的引脚映射
0	EPWM0	PWM0 使能控制位 0: 禁止 PWM0 输出 1: 允许 PWM0 输出, 同时需要配置 PWM0 的引脚映射

### 10.2.2 PWMFLT

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	PWM2FLT_EN	PWM1FLT_EN	PWM2_FLT_MODE		PWM1_FLT_MODE		PWM0_FLT_MODE	

位编号	位符号	说明
7	PWM2FLT_EN	PWM2 故障检测使能位 0: 禁止故障检测 1: 允许故障检测, 需要先进行 FLT2 的引脚映射和 IO 模式设置
6	PWM1FLT_EN	PWM1 故障检测使能位 0: 禁止故障检测 1: 允许故障检测, 需要先进行 FLT1 的引脚映射和 IO 模式设置
5-4	PWM2_FLT_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
3-2	PWM1_FLT_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
1-0	PWM0_FLT_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

## 10.2.3 PWM0 模块

### 10.2.3.1 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	PWM0FLTS	PWM0FLTC	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	PWM0FLTS	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	PWM0FLTC	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 占空比输出模式选择位 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
1-0	CK0	PWM0 时钟源选择位 00: Fosc/2 01: Fosc/4 10: Fosc/8 11: Fosc/16 注: Fosc 为未进行任何分频的系统时钟。

### 10.2.3.2 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 (PWM 关闭时修改不受限制)。

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

### 10.2.3.3 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。

### 10.2.3.4 PWM0死区时间寄存器PWM0DT

#### PWM0DT

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	PWM0DT[7:0]							

位编号	位符号	说明
7-0	PWM0DT [7:0]	PWM0 死区时间寄存器 注：其时钟源为没有进行任何分频的 Fosc。

### 10.2.3.5 PWM0故障输入脚消抖控制寄存器PWM0DBC

#### PWM0DBC

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	PWM0DBCLK[1:0]		PWM0DBCT[5:0]					

位编号	位符号	说明
7-6	PWM0DBCLK[1:0]	端口消抖时钟选择 00: $F_{CPU}/1$ 01: $F_{CPU}/4$ 10: $F_{CPU}/16$ 11: $F_{CPU}/64$
5-0	PWM0DBCT[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 = 分频系数 \*  $T_{CPU}$  \* PWM0DBCT [5:0]

注: 消抖时间不准确, 真正的消抖个数在配置值减一和配置值之间。

## 10.2.4 PWM1 模块

### 10.2.4.1 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	PWM1FLTS	PWM1FLTC	PWM1S		CK0	

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	PWM1FLTS	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	PWM1FLTC	PWM1 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	PWM1S	PWM1 和 PWM11 占空比输出模式选择位 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
1-0	CK0	PWM1 时钟源选择位 00: $F_{osc}/2$ 01: $F_{osc}/4$ 10: $F_{osc}/8$ 11: $F_{osc}/16$ 注: $F_{osc}$ 为未进行任何分频的系统时钟。

### 10.2.4.2 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

- (1) PWM1PH = 0x05;
- (2) 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
- (3) 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
- (4) 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
- (5) 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（PWM 关闭时修改不受限制）。

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

### 10.2.4.3 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。

**10.2.4.4 PWM1死区时间寄存器PWM1DT**
**PWM1DT**

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	PWM1DT[7:0]							

位编号	位符号	说明
7-0	PWM1DT [7:0]	PWM1 死区时间寄存器 注：其时钟源来没有进行任何分频的 Fosc。

**10.2.4.5 PWM1故障输入脚消抖控制寄存器PWM1DBC**
**PWM1DBC**

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	PWM1DBCLK[1:0]		PWM1DBCT[5:0]					

位编号	位符号	说明
7-6	PWM1DBCLK[1:0]	端口消抖时钟选择 00: F <sub>CPU</sub> /1 01: F <sub>CPU</sub> /4 10: F <sub>CPU</sub> /16 11: F <sub>CPU</sub> /64
5-0	PWM1DBCT[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 = 分频系数 \* T<sub>CPU</sub> \* PWM1DBCT [5:0]

注：消抖时间不准确，真正的消抖个数在配置值减一和配置值之间。

## 10.2.5 PWM2 模块

### 10.2.5.1 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	PWM2FLTS	PWM2FLTC	PWM2S		CK0	

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	PWM2FLTS	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	PWM2FLTC	PWM2 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	PWM2S	PWM2 和 PWM21 占空比输出模式选择位 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
1-0	CK0	PWM2 时钟源选择位 00: Fosc/2 01: Fosc/4 10: Fosc/8 11: Fosc/16 注: Fosc 为未进行任何分频的系统时钟。

### 10.2.5.2 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

- (1) PWM2PH = 0x05;
- (2) 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
- (3) 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
- (4) 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
- (5) 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（PWM 关闭时修改不受限制）。

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

**10.2.5.3 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。

### 10.2.5.4 PWM2死区时间寄存器PWM2DT

#### PWM2DT

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	PWM2DT[7:0]							

位编号	位符号	说明
7-0	PWM2DT [7:0]	PWM2 死区时间寄存器 注：其时钟源来没有进行任何分频的 Fosc。

### 10.2.5.5 PWM2故障输入脚消抖控制寄存器PWM2DBC

#### PWM2DBC

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	PWM2DBCLK[1:0]		PWM2DBCT[5:0]					

位编号	位符号	说明
7-6	PWM2DBCLK[1:0]	端口消抖时钟选择 00: F <sub>CPU</sub> /1 01: F <sub>CPU</sub> /4 10: F <sub>CPU</sub> /16 11: F <sub>CPU</sub> /64
5-0	PWM2DBCT[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 = 分频系数 \* T<sub>CPU</sub> \* PWM2DBCT [5:0]

注：消抖时间不准确，真正的消抖个数在配置值减一和配置值之间。



# 11 可编程计数阵列PCA

## 11.1 PCA 特性

HC89S105xx 单片机有 2 路可编程计数阵列 PCA，PCA 含有一个特殊的 16 位定时器，有 2 个 16 位的捕获/比较模块与之相连，每个模块可编程工作在 4 种模式下：捕获、软件定时器、高速输出和可调制脉冲输出。

## 11.2 PCA 工作模式

### 11.2.1 捕获模式

要使一个 PCA 模块工作在捕获模式，寄存器 PCAMODn(n=0,1)的两位 (CAPNn 和 CAPPn) 或其中任何一位必须置 1。PCA 模块工作于捕获模式时，对模块的外部 PCAn(n=0,1)输入的跳变进行采样，当采样到有效跳变时，PCA 模块就将 PCA 计数寄存器 (PCACH 和 PCACL) 的值装载到模块的捕获寄存器 CCAPHn 和 CCAPLn(n=0,1)中。

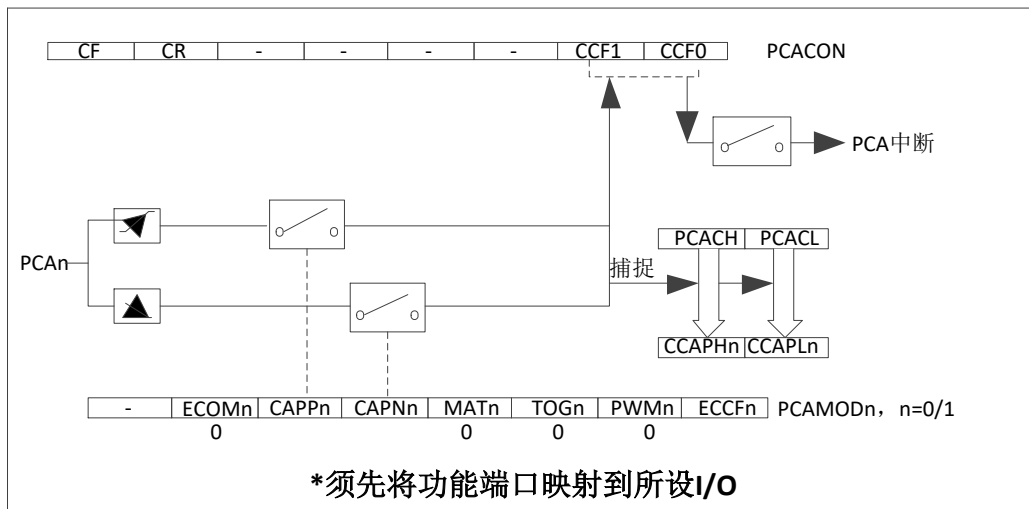


Figure 11-1 PCA 捕获模式图

如果 PCACON 寄存器中的位 CCFn(n=0,1)和 PCAMODn(n=0,1)寄存器中的位 ECCFn(n=0,1)位被置位，将产生中断。可在中断服务程序中判断哪一个模块产生了中断，并注意中断标志位的软件清零问题。

### 11.2.2 软件定时器模式

16 位软件定时器模式结构如下图所示：

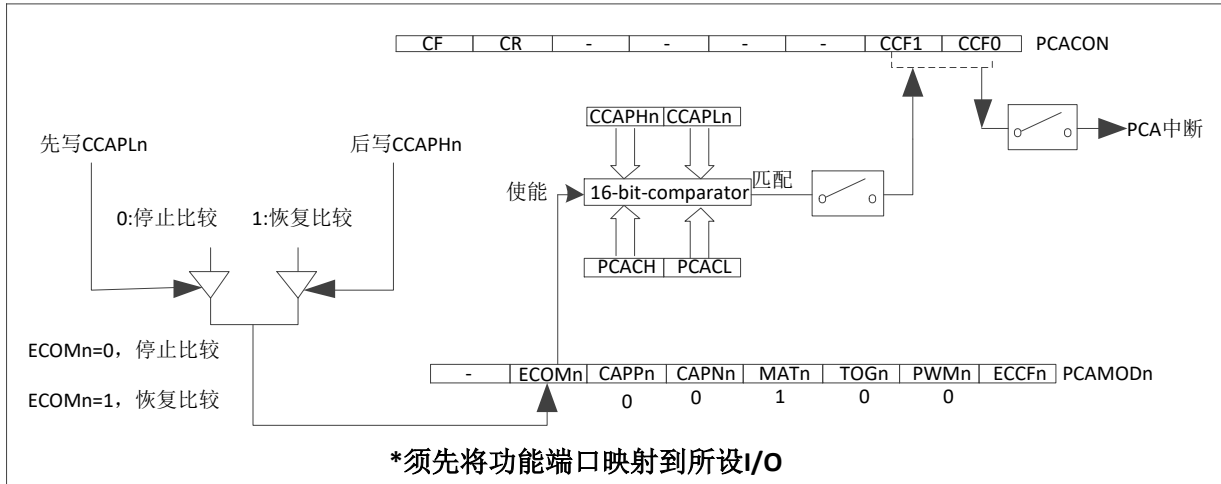


Figure 11-2 PCA 模块的 16 位软件定时器模式/PCA 比较模式

通过置位 PCAMODn(n=0,1)寄存器的 ECOM 和 MAT 位，可使 PCA 模块用作软件定时器，PCA 定时器的值与捕获寄存器的值相比较，当两者相等时，如果位 CCFn（在 PCACON 寄存器中，n=0,1）和位 ECCFn（PCAMODn 寄存器中，n=0,1）都置位，将产生中断。

[PCACH,PCACL]每隔一定的时间自动加 1，时间间隔取决于选择的时钟源。例如，当选择的时钟源为 SYSclk/12，每 12 个时钟周期[PCACH,PCACL]加 1，当[PCACH,PCACL]增加到 [CCAPHn, CCAPLn] (n=0,1)时，CCFn=1，产生中断请求。如果每次 PCA 模块中断后，在中断服务程序中给 [CCAPHn, CCAPLn] (n=0,1)增加一个相同的数值，那么下次中断来临的间隔时间 T 也是相同的，从而实现了定时功能。定时时间的长短取决于时钟源的选择以及 PCA 计数器计数值的设置。下面举例说明 PCA 计数器计数值的计算方法。

假设，系统时钟频率 SYSclk =18.432MHz，选择的时钟源为 SYSclk/12，定时时间 T 为 5ms，则 PCA 计数器计数值为：

PCA 计数器的计数值=  $T / ((1/SYSclk) \times 12) = 0.005 / ((1/18432000) \times 12) = 7680$  (10 进制数)=1E00H (16 进制数)

也就是说，PCA 计时器计数 7680 次，定时时间才是 5ms，这也就是每次给[CCAPHn,CCAPLn]增加的数值（步长）。

在操作[CCAPHn,CCAPLn]，必须先写 CCAPLn，再写 CCAPHn (n=0,1)。

### 11.2.3 高速输出模式

该模式中见下图，当 PCA 计数器的计数值与捕获寄存器的值相匹配时，PCA 模块的 PCAn(n=0,1) 输出将发生翻转。要激活高速输出模式，PCAMODn 寄存器的 TOGn(n=0,1)，MATn 和 ECOMn(n=0,1) 位必须置位。

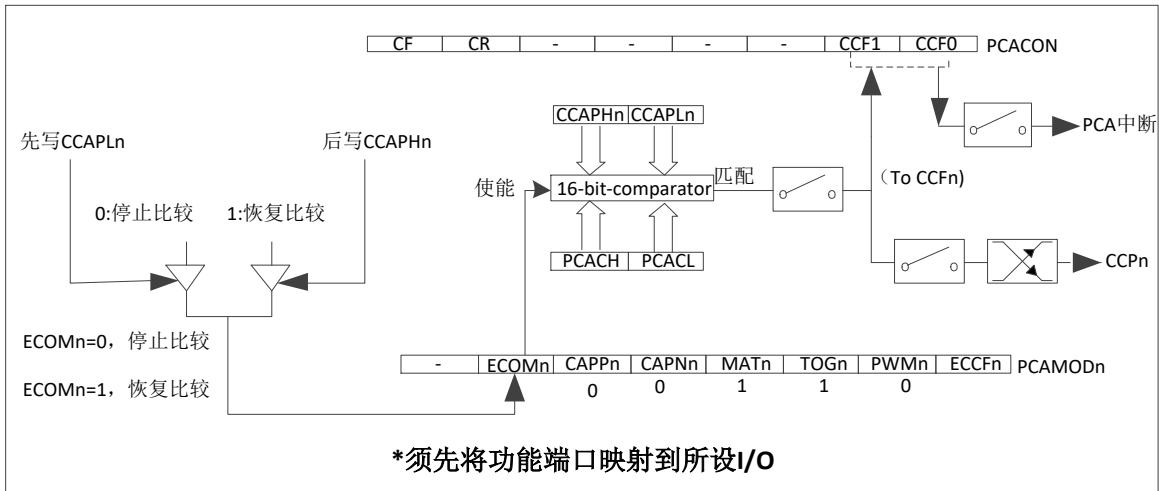


Figure 11-3 PCA 高速输出模式

CCAPLn、CCAPHn(n=0,1)的值决定了 PCA 模块 n 的输出脉冲频率。当 PCA 时钟源是 SYSclk/2 时，输出脉冲的频率 F 为： $f = \text{SYSclk} / ([\text{CCAPHn}, \text{CCAPLn}] - [\text{PCACH}, \text{PCACL}])$

其中，SYSclk 为系统时钟频率。由此，可以得到 CCAPLn、CCAPHn(n=0,1)的值。

如果计算出的结果不是整数，则进行四舍五入取整。

例如，假设 SYSclk = 20MHz，[PCACH,PCACL]从 0x0000 开始计数。要求 PCA 高速脉冲输出频率为 125KHz 的方波，则 [CCAPHn,CCAPLn] (n=0,1) 中的值应为： $[\text{CCAPHn}, \text{CCAPLn}] = (1000000/125000) * 20/2 = 80 = 50\text{H}$ 。

即设置[CCAPHn,CCAPLn]=0x0050，[PCACH,PCACL]从 0x0000 开始计数，当两者匹配时，CCPn 端口出现电平变换。因为[PCACH,PCACL]在不停地加 1 动作，为实现稳定的 125KHZ 的 PWM 输出，需要在匹配发生时使[CCAPHn,CCAPLn]在原来[CCAPHn,CCAPLn]的基础上增加 0x0050 或者清零 [PCACH,PCACL]，即可输出要求的 PWM 波形。

在操作[CCAPHn,CCAPLn]，必须先写 CCAPLn，再写 CCAPHn(n=0,1)。

### 11.2.4 脉宽调制模式 (PWM)

PCA 模块通过配置寄存器可使其工作在 8 位 PWM 模式。

8 位 PWM 模式如下图：

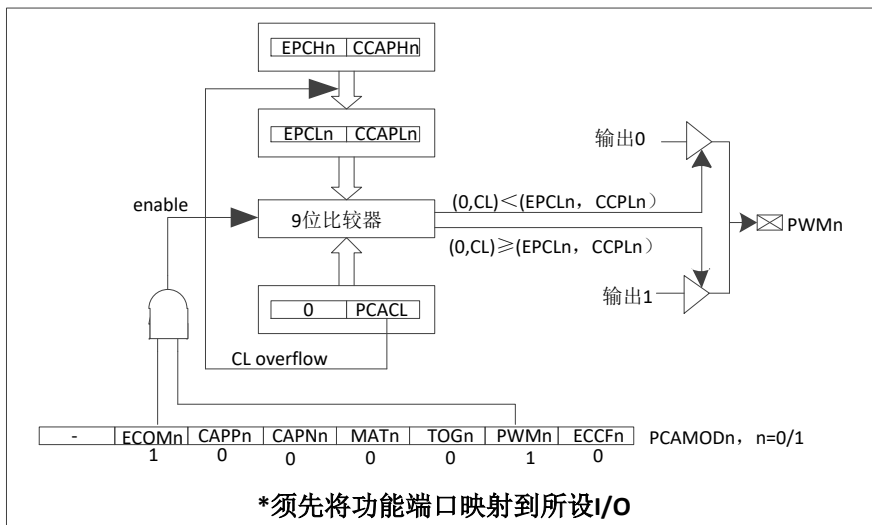


Figure 11-4 PCA PWM 模式

所有 PCA 模块都可用作 PWM 输出，输出频率取决 PCA 定时器的时源。使用相同一个 PCA 定时器的模块的输出频率相同，如果使用不同的 PCA 定时器，可以设置为不同的输出频率。各个模块的输出占空比是独立变化的，与使用的捕获寄存器[EPCLn,CCAPLn] (n=0,1)有关。

当寄存器 PCACL 的值小于[EPCLn, CCAPLn] (n=0,1)时，输出为低；当寄存器 PCACL 的值大于或等于 [EPCLn,CCAPLn]时，输出为高。当 PCACL 的值由 FF 变为 00 溢出时，[EPCHn,CCAPHn]的内容装载到[EPCLn,CCAPLn]中。这样就可实现无干扰地更新 PWM。要使能 PWM 模式，模块 PCAMODn 寄存器的 PWMn(n=0,1)和 ECOMn(n=0,1)位必须置位。

由于 PWM 是 8 位的，所以 PWM 的频率=PCA 时钟输入源频率÷256

PCA 时钟输入源可以从以下 8 种中选择一种：SYSclk、SYSclk/2、SYSclk/4、SYSclk/6、SYSclk/8、SYSclk/12、定时器 0 的溢出、ECI 输入。

举例：要求 PWM 输出频率为 38KHz，选 SYSclk 为 PCA 时钟输入源，求出 SYSclk 的值。由计算公式  $38000 = \text{SYSclk} / 256$ ，得到外部时钟频率  $\text{SYSclk} = 38000 \times 256 = 9728000$

如果可实现可调频率的 PWM 输出，可选择定时器 0 的溢出率或者 ECI 脚的输入作为 PCA 的时钟输入源。

当 EPCLn=0 及 CCAPLn = 00H 时，PWM 固定输出高，当 EPCLn=1 及 CCAPLn =0FFH 时，PWM 固定输出低。

## 11.3 PCA 相关寄存器

### 11.3.1 PCA 控制寄存器

#### PCACON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	CF	CR	-				CCF1	CCF0

位编号	位符号	说明
7	CF	PCA 计数器阵列溢出标志位 0: 软件清 0 1: 硬件置 1, PCA 计数器溢出时置 1 如果 CMOD 寄存器的 ECF 位置位，则 CF 标志可用来产生中断。
6	CR	PCA 计数器阵列运行控制位 0: 关闭 PCA 计数器 1: 启动 PCA 计数器
5-2	-	保留位（读为 0，写无效）
1	CCF1	PCA 模块 1 中断标志 0: 软件清 0 1: 硬件置 1, 当 PCA 模块 1 出现匹配或捕获时该位置 1
0	CCF0	PCA 模块 0 中断标志 0: 软件清 0 1: 硬件置 1, 当 PCA 模块 0 出现匹配或捕获时该位置 1

### 11.3.2 PCA 时钟寄存器

#### PCACLK

位编号	7	6	5	4	3	2	1	0
R/W	R/W	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	CIDL	-			CPS[2:0]			ECF

位编号	位符号	说明
7	CIDL	空闲模式下是否停止 PCA 计数的控制位 0: 空闲模式下 PCA 计数器继续工作 1: 空闲模式下 PCA 计数器停止工作
5-4	-	保留位（读为 0，写无效）
3-1	CPS[2:0]	PCA 计数脉冲源选择控制位 000: SYSclk/12 001: SYSclk/2 010: 定时器 0 的溢出脉冲 011: ECI 脚输入的外部时钟（最大速率=SYSclk/2） 100: SYSclk 101: SYSclk/4 110: SYSclk/6 111: SYSclk/8
0	ECF	PCA 计数溢出中断使能位 0: 禁止寄存器 PCACON 中 CF 位的中断 1: 允许寄存器 PCACON 中 CF 位的中断

### 11.3.3 PCA 模式寄存器

#### PCAMODn (n=0,1)

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	-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

位编号	位符号	说明
7	-	保留位（读为 0，写无效）
6	ECOMn	比较器功能控制位 0: 禁止比较器功能 1: 允许比较器功能
5	CAPPn	正捕获控制位 0: 禁止上升沿捕获 1: 允许上升沿捕获

4	CAPNn	负捕获控制位 0: 禁止下降沿捕获 1: 允许下降沿捕获
3	MATn	匹配控制位 0: 禁止匹配 1: PCA 计数值与模块的比较/捕获寄存器的值的匹配将置位 PCACON 寄存器的中断标志位 CCFn
2	TOGn	翻转控制位 0: 禁止翻转 1: 工作在 PCA 高速输出模式, PCA 计数器的值与模块的比较/捕获寄存器的值的匹配将使 PCAn 脚翻转
1	PWMn	脉宽调节模式 0: 禁止 PWM 1: 允许 CEXn 脚用作脉宽调节输出
0	ECCFn	CCFn 中断使能位 0: 禁止寄存器 PCACON 中比较/捕获标志 CCFn 的中断 1: 允许寄存器 PCACON 中比较/捕获标志 CCFn 的中断

ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	模块功能
0	0	0	0	0	0	0	无此操作
1	0	0	0	0	1	0	8位PWM, 无中断
1	1	0	0	0	1	1	8位PWM输出, 由低变高可产生中断
1	0	1	0	0	1	1	8位PWM输出, 由高变低可产生中断
1	1	1	0	0	1	1	8位PWM输出, 由低变高可产生, 由高变低都可产生中断
x	1	0	0	0	0	x	16位捕获模式, 由PCAn/PCAn的上升沿触发
x	0	1	0	0	0	x	16位捕获模式, 由PCAn/PCAn的下降沿触发
x	1	1	0	0	0	x	16位捕获模式, 由PCAn/PCAn的跳变触发
1	0	0	1	0	0	x	16位软件定时器
1	0	0	1	1	0	x	16位高速输出

### 11.3.4 PCA 计数寄存器

#### PCACL

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	PCACL[7:0]							

位编号	位符号	说明
7-0	PCACL[7:0]	计数寄存器低字节

#### PCACH

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	PCACH[7:0]							

位编号	位符号	说明
7-0	PCACH[7:0]	计数寄存器高字节

### 11.3.5 PCA 捕捉/比较寄存器 CCAPLn(n = 0,1)、CCAPHn (n = 0,1)

#### CCAPLn (n = 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	CCAPLn[7:0] (n = 0,1)							

位编号	位符号	说明
7-0	CCAPLn [7:0] (n = 0,1)	PCAn(n = 0,1)捕捉/比较寄存器低字节

#### CCAPHn (n = 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	CCAPHn[7:0] (n = 0,1)							

位编号	位符号	说明
7-0	CCAPHn [7:0] (n = 0,1)	PCAn(n = 0,1)捕捉/比较寄存器高字节

### 11.3.6 PCA 模块 PWM 寄存器 PCA\_PWMn

PCA\_PWMn (n = 0,1)

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	-						EPCHn	EPCLn

位编号	位符号	说明
7-2	-	保留位
1	EPCHn	在 PWM 模式下，与 CCAPHn 组成 9 位数。
0	EPCLn	在 PWM 模式下，与 CCAPLn 组成 9 位数。



# 12 实时时钟 RTC

## 12.1 RTC 特性

此系列单片机内建有一个 RTC 模块，要使 RTC 模块工作，必须要使能外部低频晶体时钟 32.768KHz，否则 RTC 无效，使能外部低频晶振需要两步操作，先通过代码选项选择外部低频晶振，然后配置时钟控制寄存器 CLKCON 里的 XTALEN 为 1。使能低频晶振后，如果 RTCEN 为 1，RTC 模块就开始工作，在掉电模式下仍然会继续工作，当 RTC 计数溢出后，RTC 中断会唤醒掉电模式，但唤醒之后必须工作 30μs 后才能进入下一次掉电，如果不需要 RTC 在掉电模式下工作需要使 RTCEN 为 0。可以通过设置 RTC 的控制寄存器产生固定的周期性中断及连续的固定时钟输出。

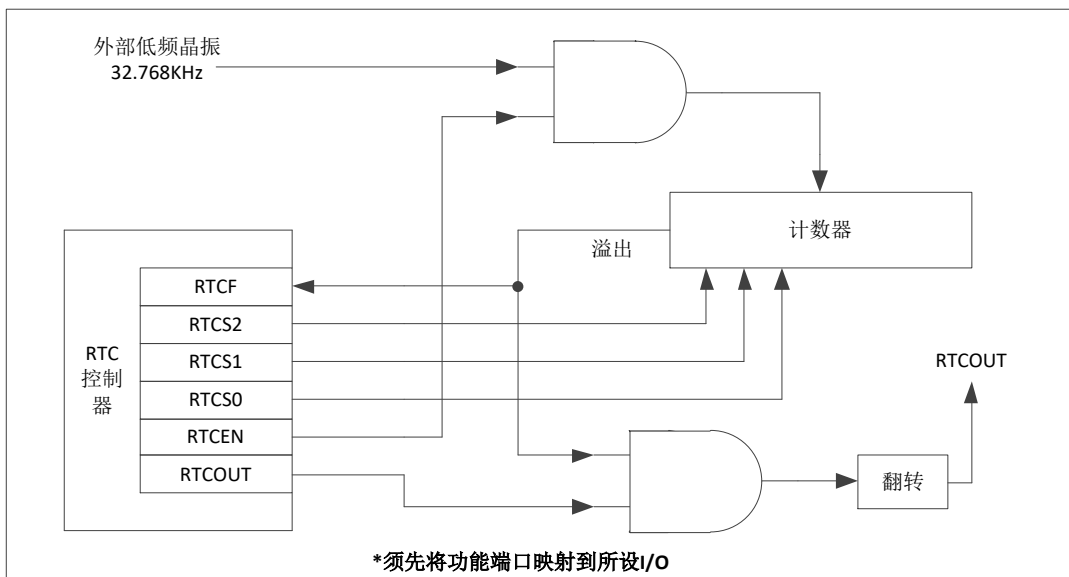


Figure 12-1 RTC 功能框图

## 12.2 RTC 相关寄存器

### 12.2.1 RTC 控制寄存器 RTCC

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	RTCF	-		RTCS[2:0]			RTCEN	RTCOUT

位编号	位符号	说明
7	RTCF	RTC 中断标志位 0: 软件清 0 1: RTC 计数器溢出时, 硬件置 1
6-5	-	保留位 (读为 0, 写无效)
4-2	RTCS[2:0]	RTC 中断时间选择位 000: 0.0625s 001: 0.125s 010: 0.25s 011: 0.5s 100: 1.0s 101: 2.0s 110: 4.0s 111: 8.0s
1	RTCEN	RTCEN 运行控制位 0: 停止 RTC 工作 1: 启动 RTC 工作 (重新计数)
0	RTCOUT	RTC 时钟输出使能位 0: 禁止 RTC 时钟输出 1: 允许 RTC 时钟输出 (应事先将该端口设为输出模式)

# 13 看门狗定时器WDT

## 13.1 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工作
- 可灵活配置溢出时间

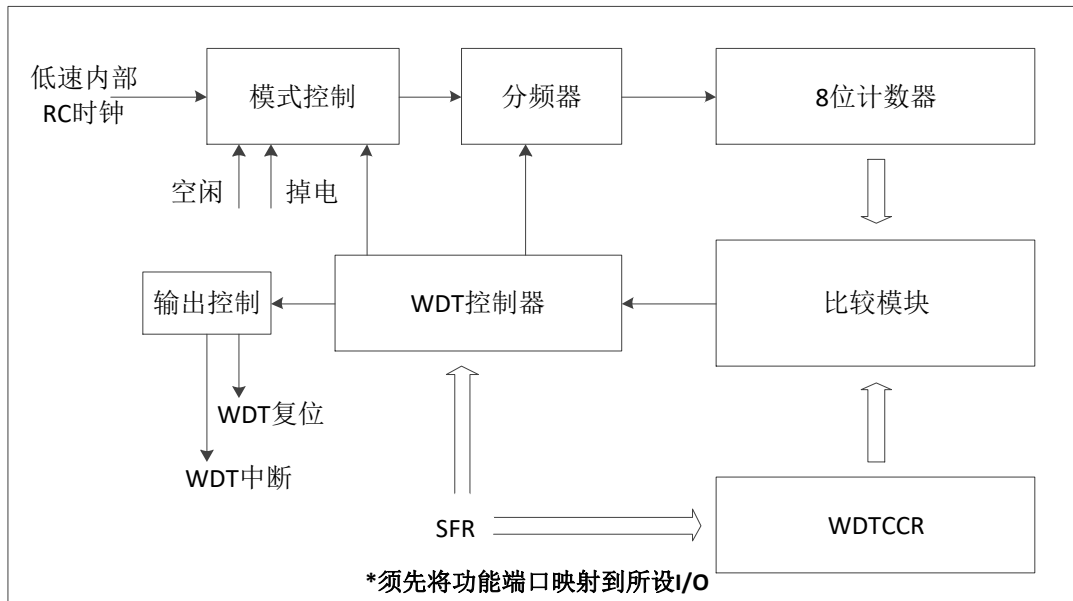


Figure 13-1 WDT 功能框图

HC89S105xx看门狗定时器是一个递增计数器，独立内建38KHz RC振荡器作为其时钟源，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，芯片是否复位也可通过寄存器来配置。

HC89S105xx看门狗定时器溢出后有溢出标志，复位有专用的复位标志，WDT溢出时间可设，清WDT操作只需置相应控制位即可，操作灵活。

## 13.2 WDT 相关寄存器

### 13.2.1 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	-	WDRST	WDTF	WDTCLR	WDTPD	WDTPS[2:0]		

位编号	位符号	说明
7	-	保留位（读为 0，写无效）
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 WDT 计数器清 0，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，需要使能 WDTRST。 即只有看门狗复位才能唤醒掉电模式。 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

### 13.2.2 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，WDT 溢出并且计数器清 0 重新计数，写入 00 时，将关闭 WDT 功能（但不关闭低速内部 RC），即相当于禁止 WDT。写入非 0 数据时，将启动 WDT。

溢出时间 = WDT分频系数 \* (WDTCCR[7:0]+1)/38K。

WDTCCR[7:0] = 0xFF 看门狗溢出时间如下表：

PS2	PS1	PS0	WDT分频系数	WDT最大溢出时间@38K
0	0	0	8	54 ms
0	0	1	16	108 ms
0	1	0	32	216 ms
0	1	1	64	432 ms
1	0	0	128	864 ms
1	0	1	256	1728 ms
1	1	0	512	3456 ms
1	1	1	1024	6912 ms

# 14 通用异步收发器UART

## 14.1 UART 特性

- 2 个自带波特率发生器的 UART
- 波特率发生器为一个 16 位向上的计数器
- UART 有四种工作方式
- UART 增加帧错误，接收溢出及写冲突检测
- UART 增加地址自动识别

## 14.2 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 $F_{CPU} / 12 \times 6^{UX6}$
0	1	方式1	异步	BRT独立波特率发生器的溢出率/16
1	0	方式2	异步	$(2^{SMOD} / 64) \times F_{CPU}$
1	1	方式3	异步	BRT独立波特率发生器的溢出率/16

### 14.2.1 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。HC89S105xx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 $F_{CPU}$ 的1/12或1/2。当UX6位等于0时，串行端口以 $F_{CPU}$ 的1/12运行，当UX6位等于1时，串行端口以 $F_{CPU}$ 的1/2运行。与标准8051唯一不同的是，HC89S105xx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。

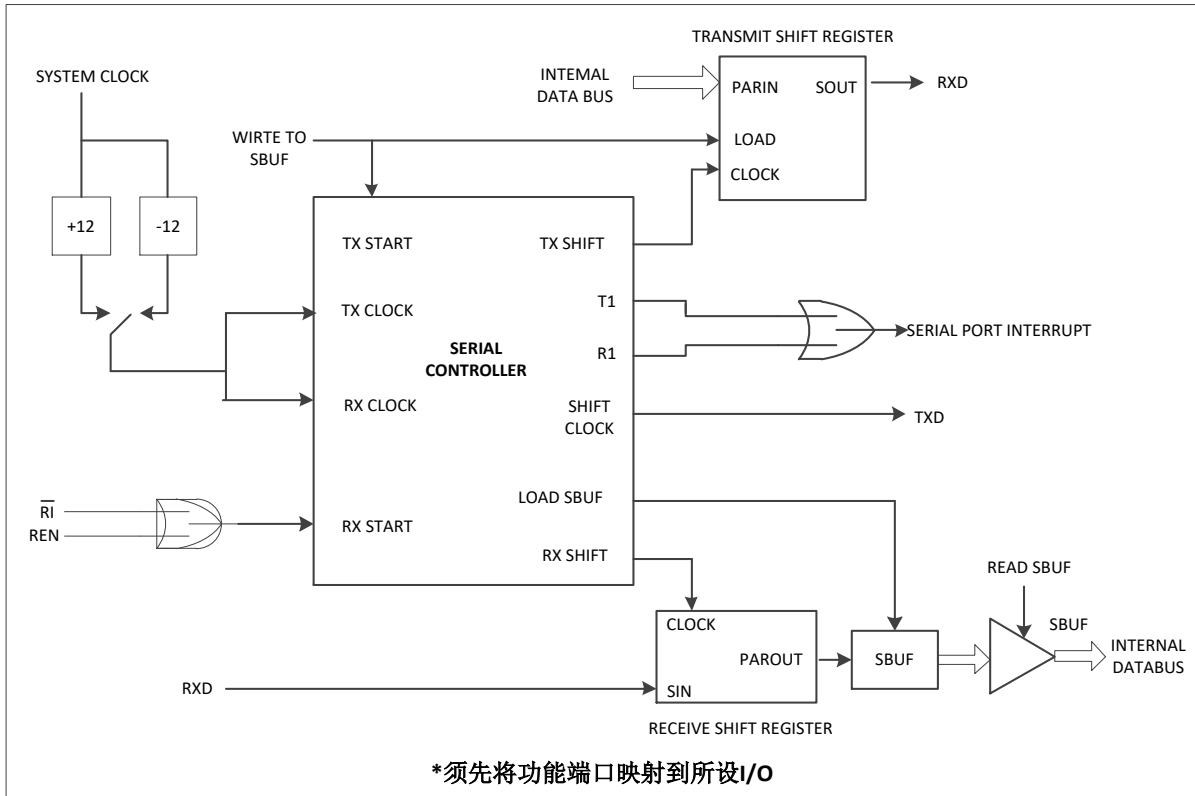


Figure 14-1 UART 模式 0 功能框图

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。

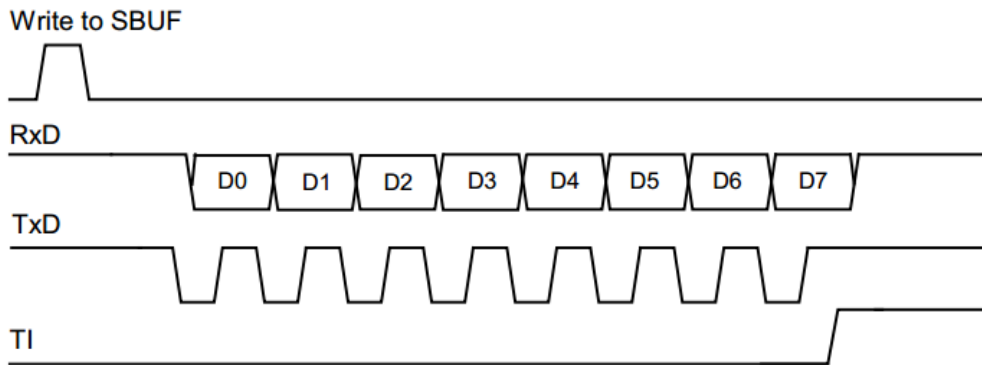


Figure 14-2 模式 0 数据发送时序框图

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。

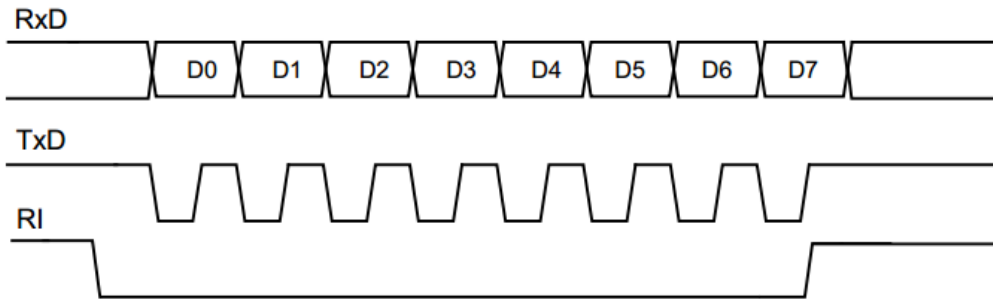


Figure 14-3 模式 0 数据接收时序框图

### 14.2.2 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 中的波特率固定为自带波特率发生器溢出率的 1/16。

功能块框图如下图所示:

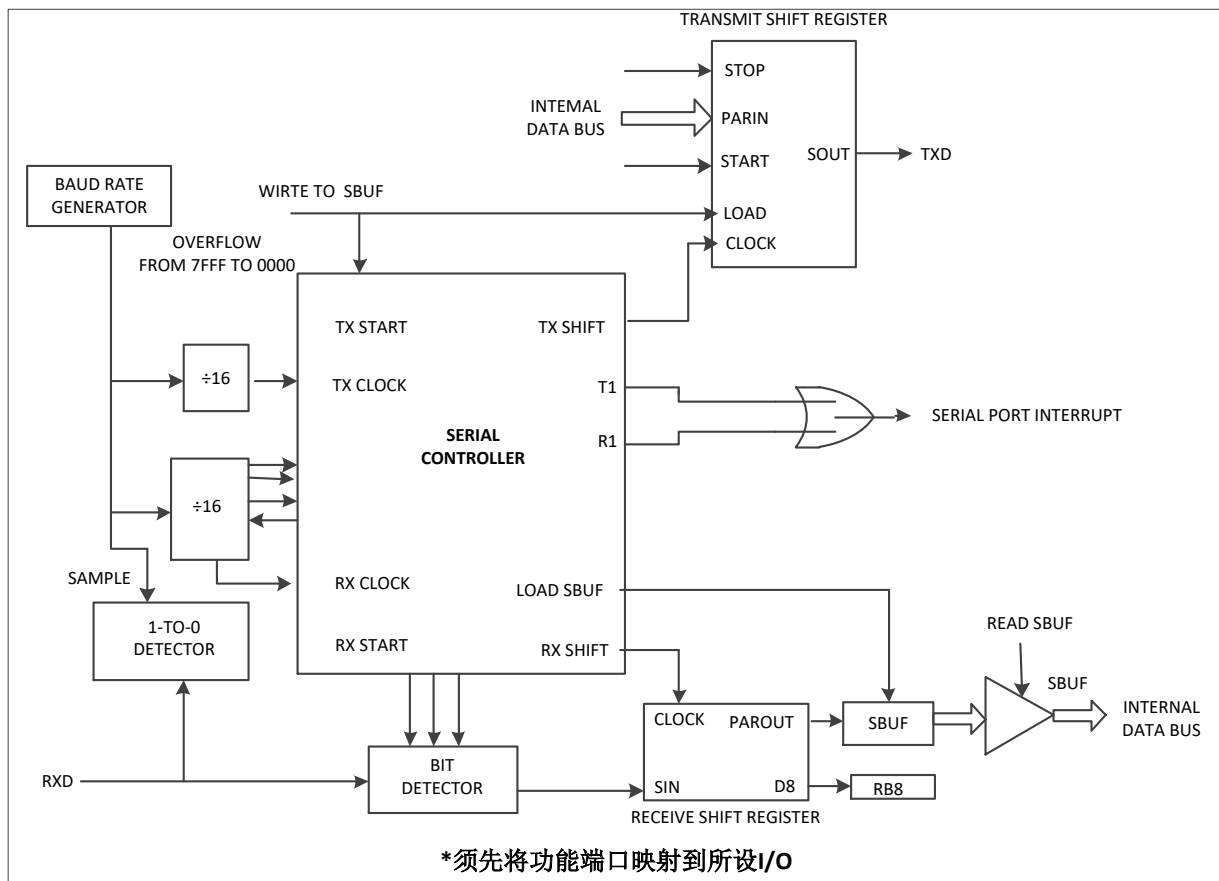


Figure 14-4 UART 模式 1 功能框图

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。



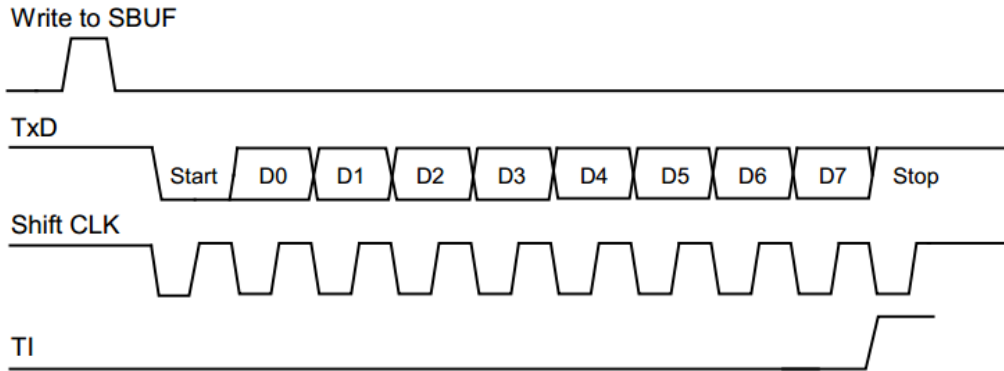


Figure 14-5 模式 1 数据发送时序框图

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。

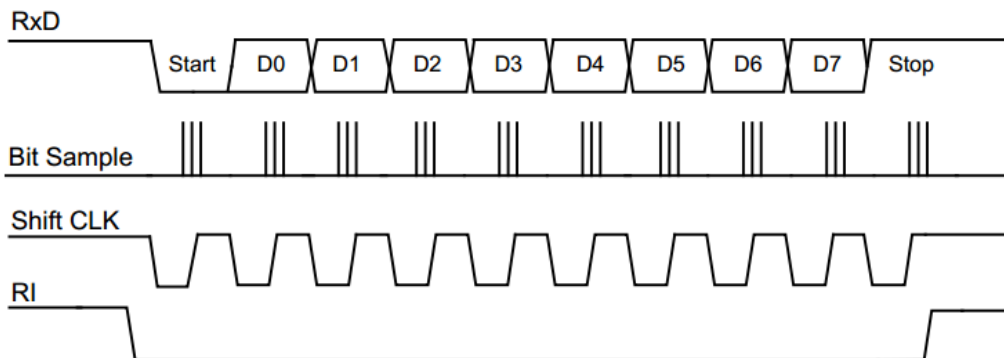


Figure 14-6 模式 1 数据接收时序框图

### 14.2.3 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。

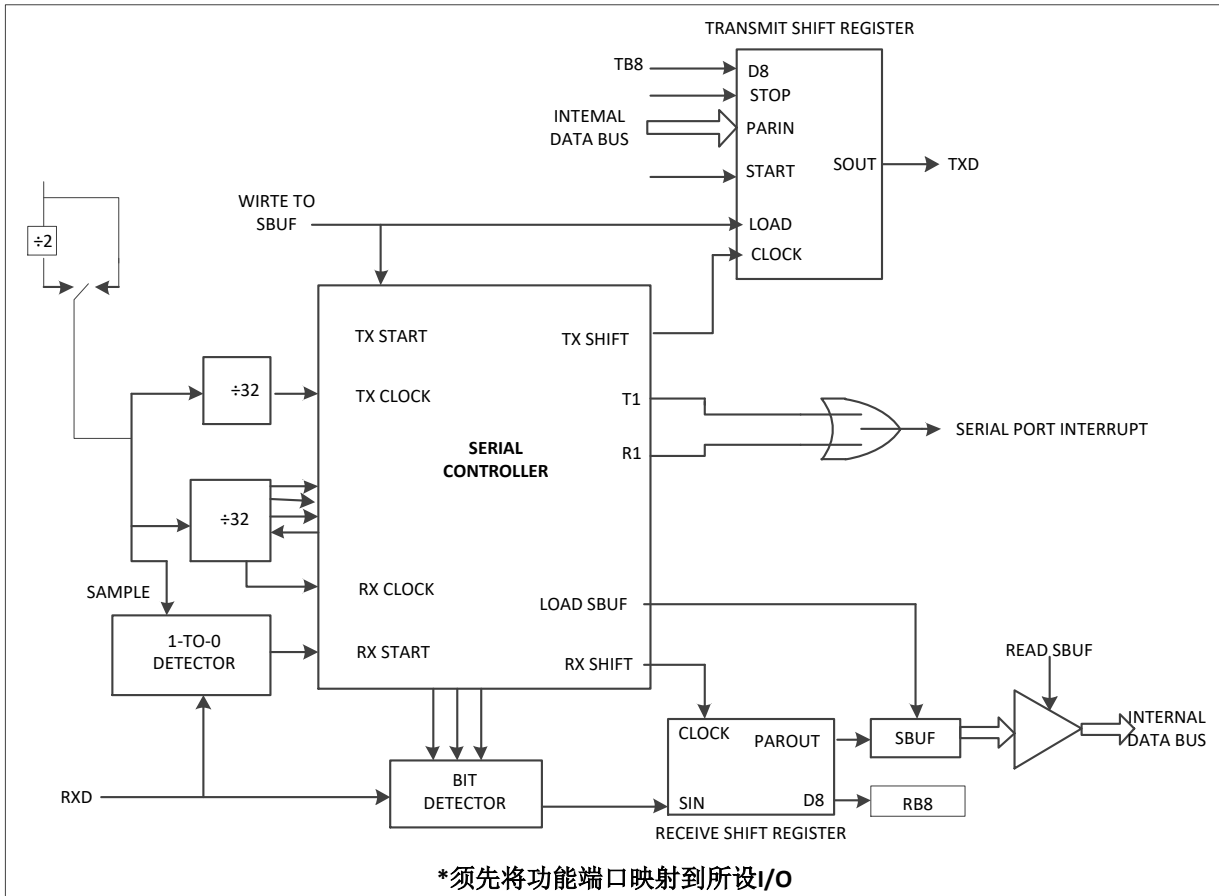


Figure 14-7 UART 模式 2 功能框图

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。

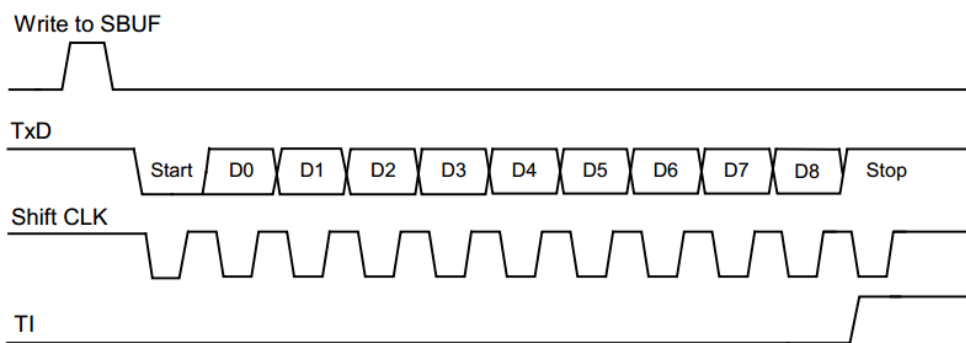


Figure 14-8 模式 2 数据发送时序框图

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。

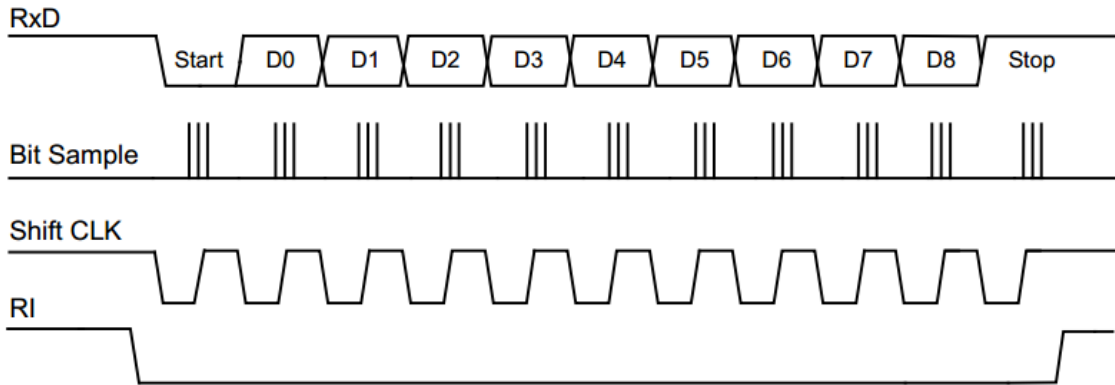


Figure 14-9 模式 2 数据接收时序框图

### 14.2.4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。

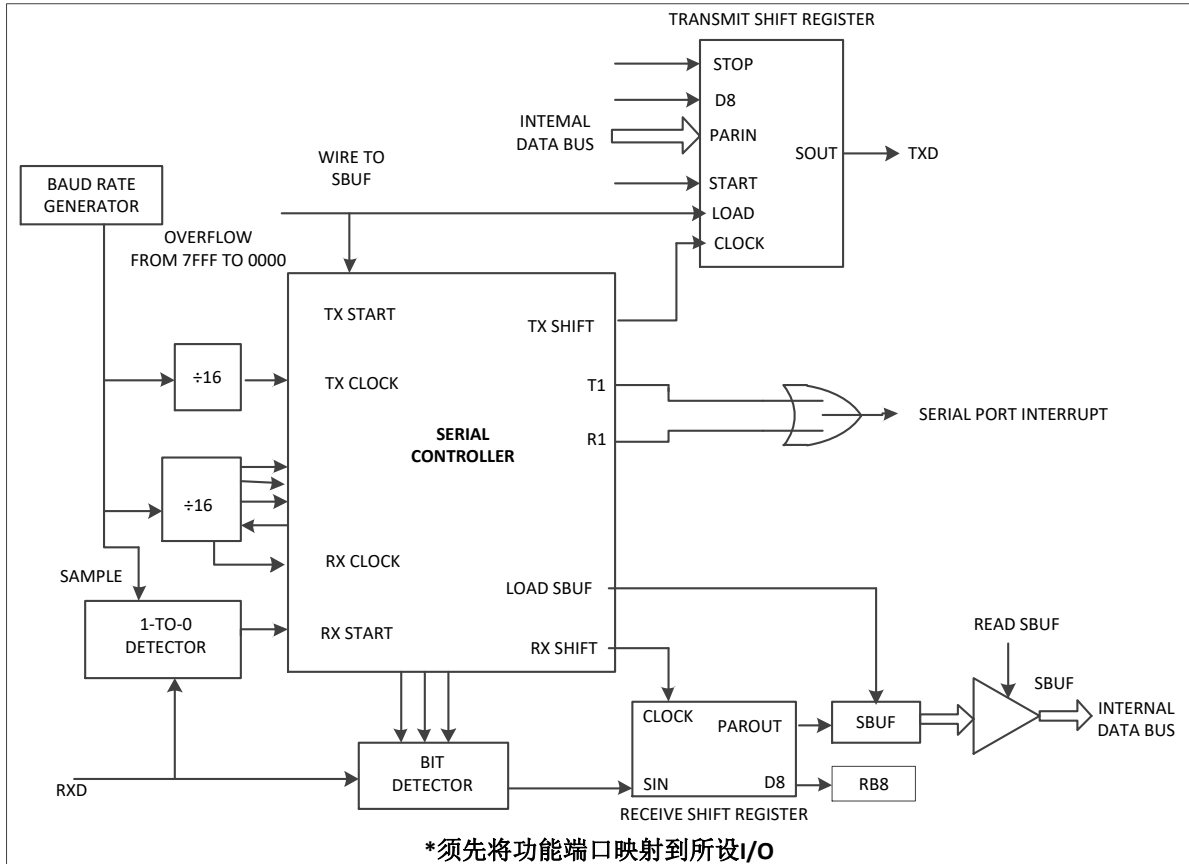


Figure 14-10 UART 模式 3 功能框图

## 14.3 波特率发生器

UART 自带一个波特率发生器，它实质上是一个 16 位递增计数器。

在方式 0 中，波特率可编程为系统时钟的 1/12 或 1/2，由 UX6 位决定。当 UX6 为 0 时，串行端口在 F<sub>CPU</sub> 的 1/12 下运行。当 UX6 为 1 时，串行端口在 F<sub>CPU</sub> 的 1/2 下运行。

在方式 2 中，波特率固定为系统时钟的 1/32 或 1/64，由 SMOD 位中决定。当 SMOD 位为 0 时，UART 以 F<sub>CPU</sub> 的 1/64 运行。当 SMOD 位为 1 时，UART 以 F<sub>CPU</sub> 的 1/32 运行。

$$\text{Baud} = 2^{\text{SMOD}} \times \left(\frac{F_{\text{CPU}}}{64}\right)$$

在方式 1 和方式 3 中，波特率公式如下：

$$\text{Baud} = \frac{F_{\text{CPU}}}{16 \times (65536 - \text{SBRT})} , \text{SBRT} = [\text{SBRTH} : \text{SBRTL}]$$

下表为常用CPU频率与常用波特率所对应的波特率发生器的重载值：

常用 波特率	F <sub>CPU</sub>			
	2MHz	4MHz	8MHz	16MHz
1200	FF98	FF2F	FE5F	FCBE
2400	FFCC	FF97	FF2F	FE5F
4800	FFE6	FFCB	FF97	FF2F
9600	FFF3	FFE5	FFCB	FF97
19200	FFF9	FFF2	FFE5	FFCB
38400	FFFC	FFF9	FFF2	FFE5
56000	FFFD	FFFB	FFF7	FFEE

## 14.4 多机通信

### 14.4.1 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：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。

### 14.4.2 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（10100001和10100101）。

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。

## 14.5 帧出错检测

2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。

### 14.5.1 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）。

### 14.5.2 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。

## 14.6 UART1 相关寄存器

### 14.6.1 UART1 控制寄存器 SCON、SCON2

#### SCON

位编号	7	6	5	4	3	2	1	0
R/W	R	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	-	RXROV	TXCOL	REN	TB8	RB8	TI	RI

位编号	位符号	说明
7	-	保留位
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/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	SMOD	-	UX6	BRTR	BRTOUT	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, 波特率为系统时钟的 1/64 1: 在方式 2 中, 波特率为系统时钟的 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 $F_{CPU}/12$ 1: 串口模式 0 时钟为 $F_{CPU}/2$
4	BRTR	独立波特率发生器 BRT 运行控制位 0: 停止独立波特率发生器 BRT 工作 1: 启动独立波特率发生器 BRT 工作
3	BRTOUT	独立波特率发生器 BRT 输出使能位 0: 禁止独立波特率发生器 BRT 的时钟输出 1: 允许独立波特率发生器 BRT 的时钟输出
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下, 禁止停止位确认检验, 任何停止位都会置位 RI 在方式2和3下, 任何字节都会置位RI 1: 在方式1下, 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下, 只有地址字节(第9位=“1”)才能置位RI
---	-----	---------------------------------------------------------------------------------------------------------------------------------------------------------------

UART 工作方式选择表:

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 $F_{CPU}/12$ 当UX6 = 1时, 波特率是 $F_{CPU}/2$
0	1	方式1	8位UART, 波特率可变	BRT独立波特率发生器的溢出率/16
1	0	方式2	9位UART	$(2^{SMOD}/64) \times F_{CPU}$
1	1	方式3	9位UART, 波特率可变	BRT独立波特率发生器的溢出率/16

### 14.6.2 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						

### 14.6.3 UART1 独立波特率发生器寄存器 SBRTL、SBRTH

#### SBRTL

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	SBRTL[7:0]							

位编号	位符号	说明
7-0	SBRTL[7:0]	波特率发生器寄存器 BRT 低 8 位, 用于保存重装时间常数

#### SBRTH

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	SBRTH[7:0]							

位编号	位符号	说明
7-0	SBRTH[7:0]	波特率发生器寄存器 BRT 高 8 位, 用于保存重装时间常数

注: 修改SBRTL & SBRTH, 类似T0/T1方式0时重载数据修改。



## 14.6.4 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

## 14.7 UART2

UART2的控制和工作方式与UART1基本相同。

不同点如下：

1. UART2的寄存器在扩展SFR中；
2. URAT2的波特率时钟不能输出。

### 14.7.1 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/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	侦错误检测位 0: 无侦错误或软件清 0 1: 有侦错误, 硬件置 1 注: 侦错误是指无效停止位。
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

**S2CON2**

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	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	SMOD	-	UX6	BRTR	-	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, 波特率为系统时钟的 1/64 1: 在方式 2 中, 波特率为系统时钟的 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 $F_{CPU}/12$ 1: 串口模式 0 时钟为 $F_{CPU}/2$
4	BRTR	独立波特率发生器 BRT 运行控制位 0: 停止独立波特率发生器 BRT 工作 1: 启动独立波特率发生器 BRT 工作
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 下, 禁止停止位确认检验, 任何停止位都会置位 RI 在方式 2 和 3 下, 任何字节都会置位 RI 1: 在方式 1 下, 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 下, 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 $F_{CPU} / 12$ 当UX6 = 1时，波特率是 $F_{CPU} / 2$
0	1	方式1	8位UART，波特率可变	BRT独立波特率发生器的溢出率/16
1	0	方式2	9位UART	$(2^{SMOD} / 64) \times F_{CPU}$
1	1	方式3	9位UART，波特率可变	BRT独立波特率发生器的溢出率/16

### 14.7.2 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

### 14.7.3 UART2 独立波特率发生器寄存器 S2BRTL、S2BRTH

#### S2BRTL

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	SBRTL[7:0]							

位编号	位符号	说明
7-0	SBRTL[7:0]	波特率发生器寄存器 BRT 低 8 位，用于保存重装时间常数

#### S2BRTH

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	SBRTH[7:0]							

位编号	位符号	说明
7-0	SBRTH[7:0]	波特率发生器寄存器 BRT 高 8 位，用于保存重装时间常数

注：修改S2BRTL & S2BRTH，应先修改高位S2BRTH，再修改低位S2BRTL。

## 14.7.4 UART2 自动地址识别 S2ADDR、S2ADEN

### 从机地址寄存器 S2ADDR

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	S2ADDR[7:0]							

位编号	位符号	说明
7-0	S2ADDR[7:0]	从机地址寄存器

### 从机地址掩码寄存器 S2ADEN

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	S2ADEN [7:0]							

位编号	位符号	说明
7-0	S2ADEN [7:0]	从机地址掩码寄存器

# 15 串行外部设备接口SPI

## 15.1 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

## 15.2 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 从主设备串行传送到从设备, 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 从从设备串行传送到主设备, 从设备输出, 主设备输入。若该设备为从设备且未被选中, 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。

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。

(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 SPSTAT 的 SPIF 标志位将置位 (可中断), 且 MSTR 位也将被清 0, 从而使该设备强制切换成从设备。因此, 用户软件必须一直对 MSTR 位进行检测, 如果该为被一个从机选择所清零而用户想继续将 SPI 作为主机, 这时就必须重新置位 MSTR, 否则就进入从机模式。

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。

### 15.3 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。

### 15.4 SPI 功能框图

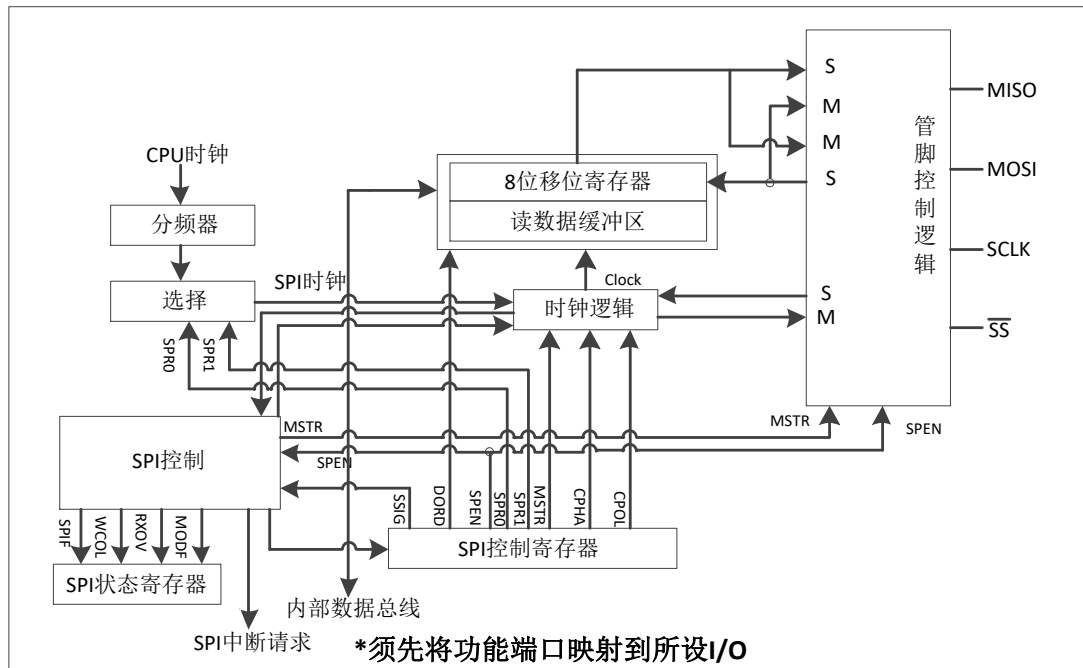


Figure 15-1 SPI 功能方框图

### 15.5 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：写入的数据不会影响到需要读出的数据。

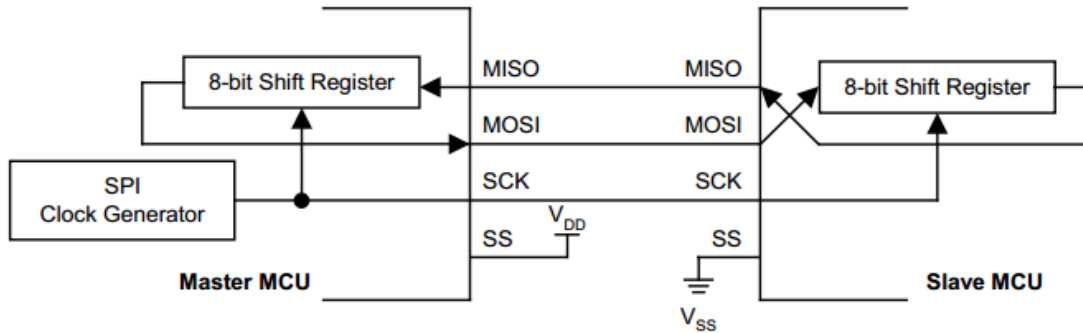


Figure 15-2 全双工主从互联图

### 主模式

#### (1) 模式启动

SPI 主设备控制控制 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。

## 15.6 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。

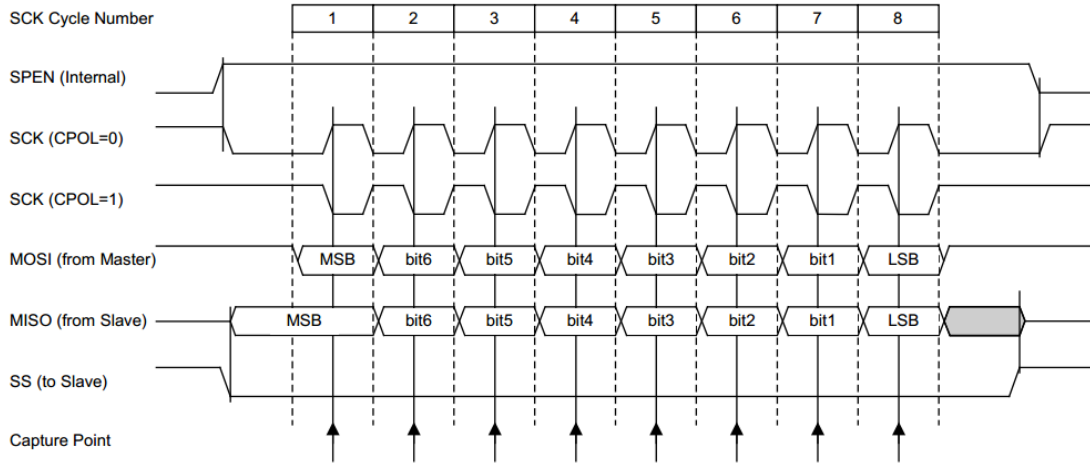


Figure 15-3 数据传送形式 (CPHA=0)

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。

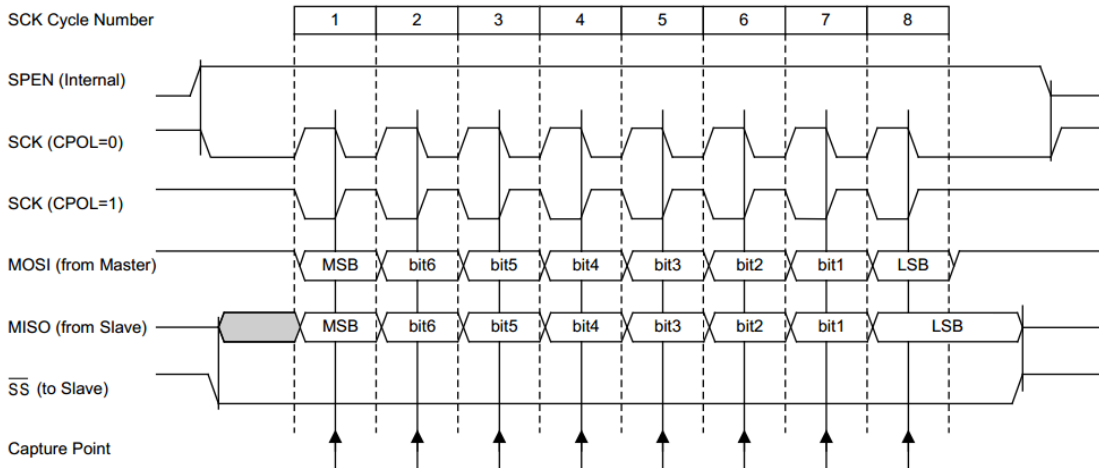


Figure 15-4 数据发送形式 (CPHA=1)

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。



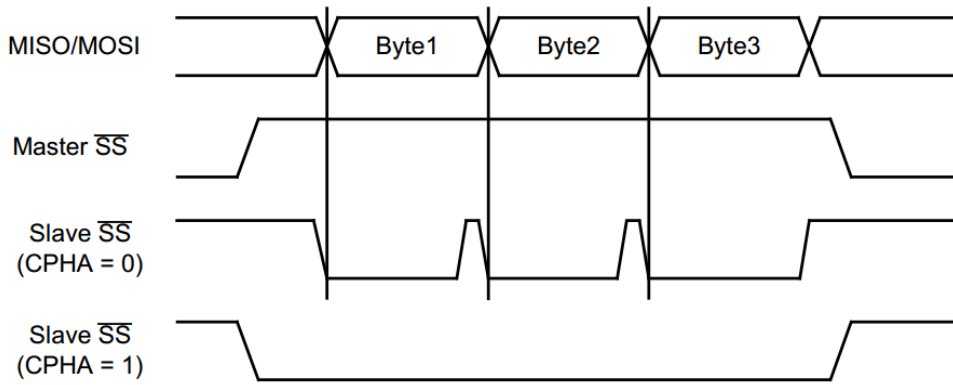


Figure 15-5 CPHA/ $\overline{SS}$ 时序

## 15.7 SPI 出错检测

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。需软件清 0

## 15.8 SPI 中断

SPIF 置位或者模式冲突都能产生一个 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。

模式冲突：主设备的  $\overline{SS}$  引脚被使能时，若  $\overline{SS}$  被拉低，此时将出现争夺总线的情况。SPSTAT 的 SPIF 标志位将置位（可中断），且 MSTR 位也将被清 0，从而使该设备强制切换成从设备。因此，用户软件必须一直对 MSTR 位进行检测，如果该为被一个从机选择所清零而用户想继续将 SPI 作为主机，这时就必须重新置位 MSTR，否则就进入从机模式。

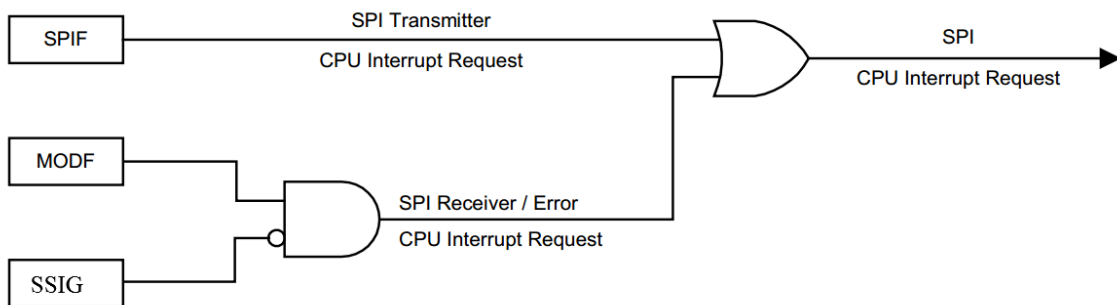


Figure 15-6 SPI 中断请求的产生

## 15.9 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	$\overline{SS}$ 配置为输入，SSIG为0。如果 $\overline{SS}$ 被驱动为低电平。则被选择作为从机。此时MSTR将清零，可产生中断请求。
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	-

## 15.10 SPI 相关寄存器

### 15.10.1 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	$\overline{SS}$ 引脚使能位 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	主/从机模式选择位

		0: MSB先发送 1: LSB先发送
4	MSTR	传送方向选择位 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: $F_{CPU}/4$ 01: $F_{CPU}/16$ 10: $F_{CPU}/64$ 11: $F_{CPU}/128$

### 15.10.2 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	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	SPIF	WCOL	-					

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)
5-0	-	保留位 (读为 0, 写无效)

### 15.10.3 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

# 16 IIC总线

## 16.1 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）

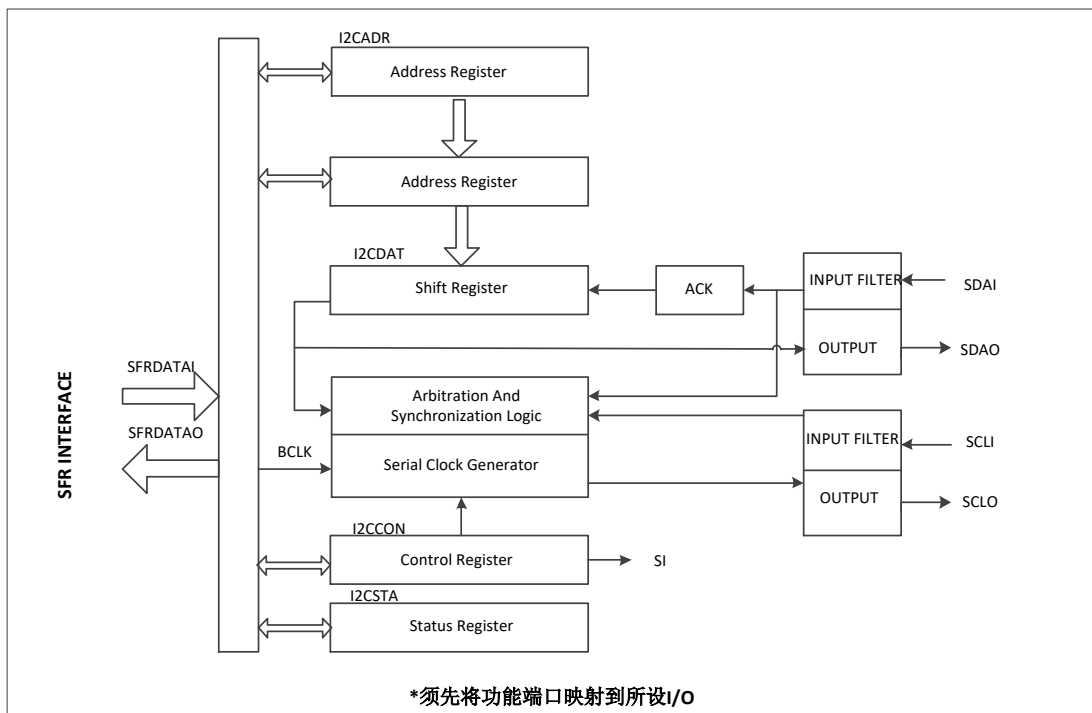


Figure 16-1 IIC 功能框图

## 16.2 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。

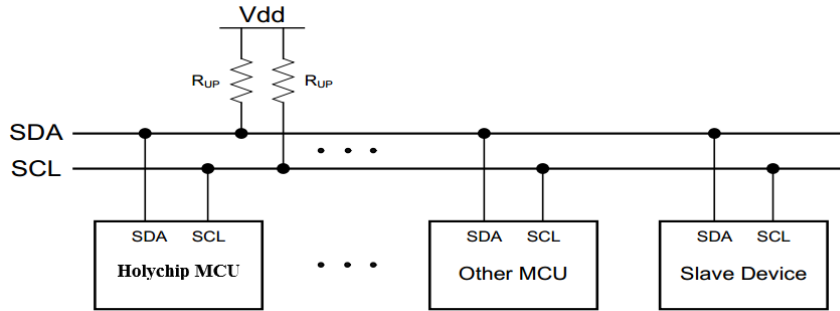


Figure 16-2 IIC 总线连接图

### 16.3 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 Figure16-3 所示。

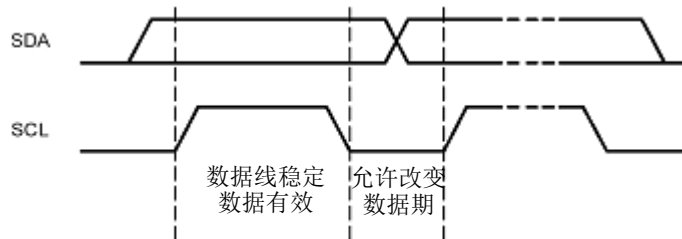


Figure 16-3 IIC 总线上数据的有效性

### 16.4 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 16-4 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。

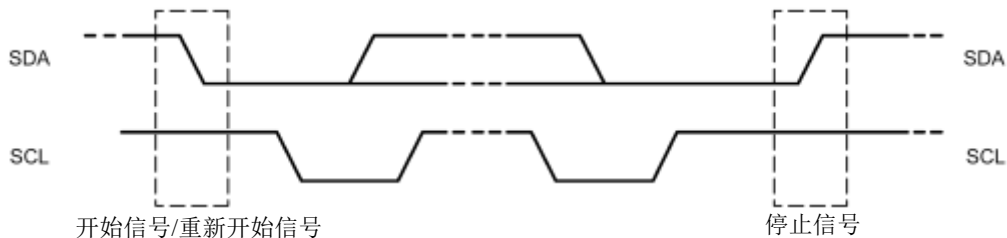


Figure 16-4 开始、重新开始、停止信号

停止信号 (STOP): 如 Figure16-4 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 Figure16-5 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。

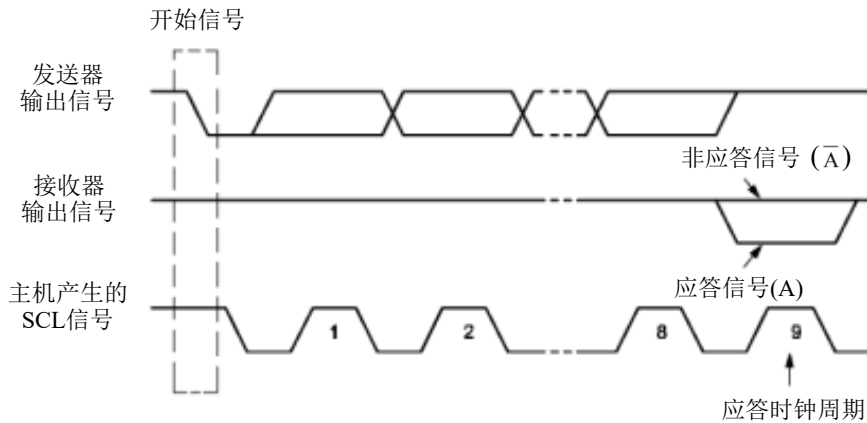


Figure 16-5 IIC 总线的应答信号

**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 Figure16-5 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。

## 16.5 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。

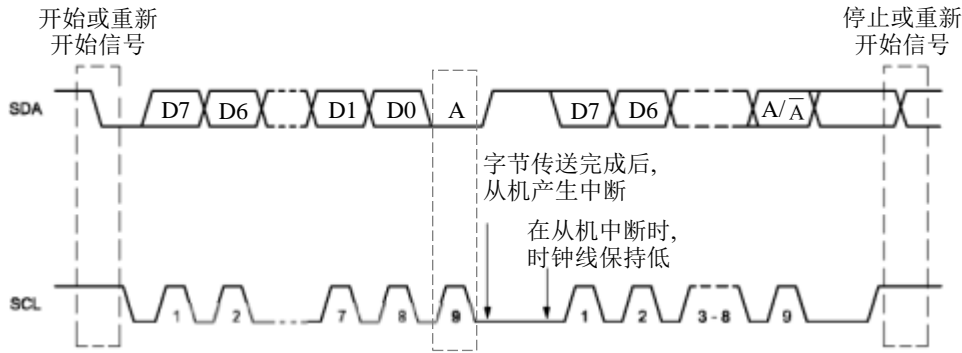


Figure 16-6 IIC 总线的数据传输格式

如 Figure 16-6 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送得以继续正常进行。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。

## 16.6 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。

## 16.7 主机向从机读写 1 个字节数据的过程

如 Figure 16-7 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，然后紧接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。

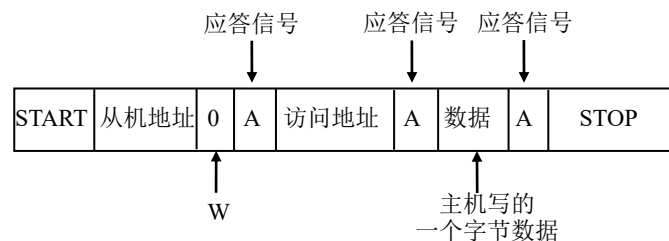


Figure 16-7 主机向从机写数据

如 Figure 16-7 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，然后紧接着

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。

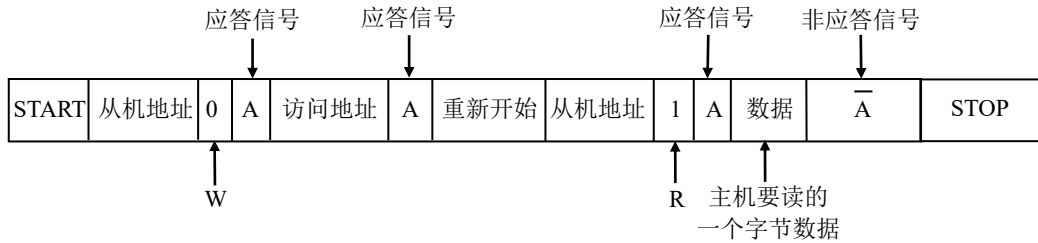


Figure 16-8 主机向从机读写数据 1 个字节数据



## 16.8 IIC 工作模式

### 16.8.1 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。

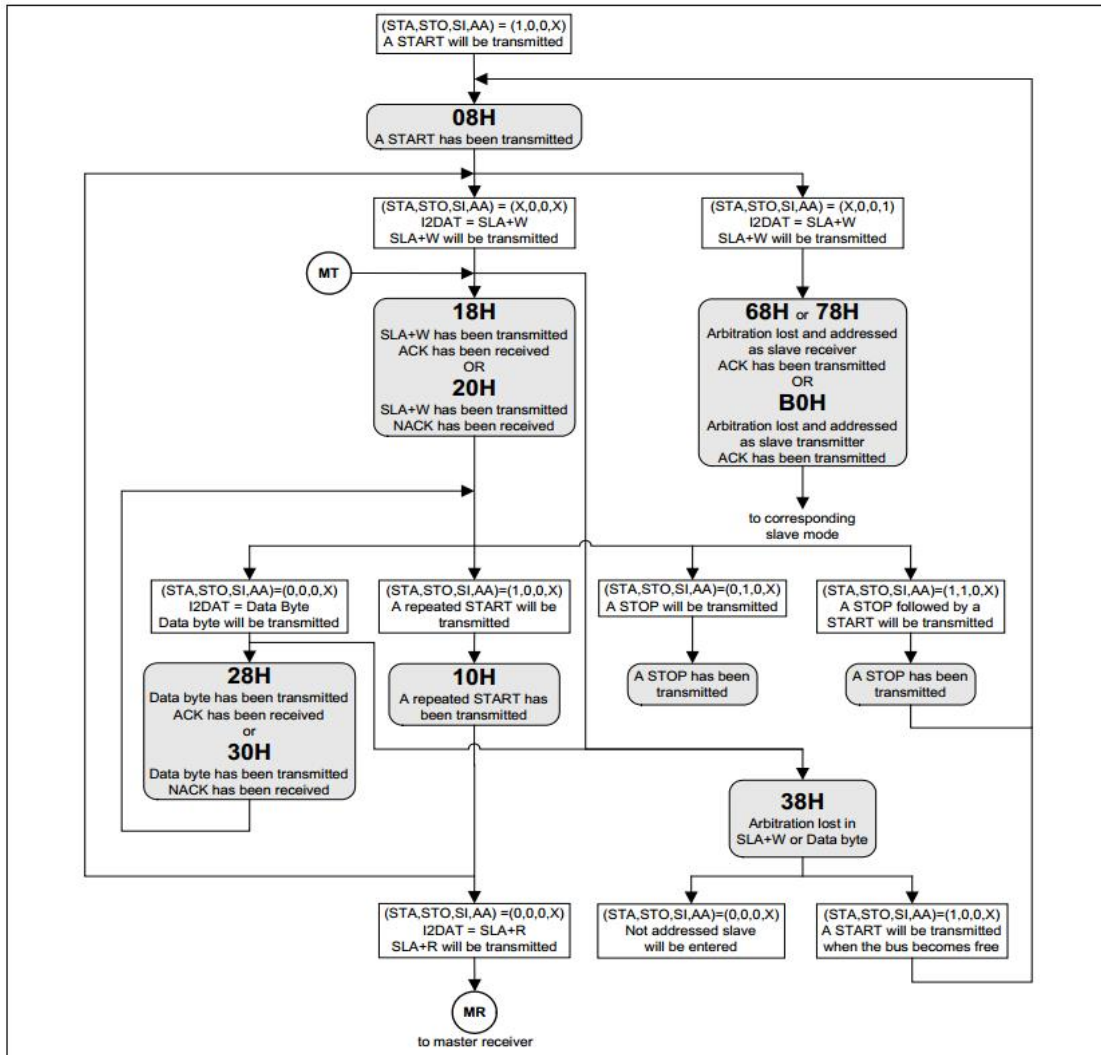


Figure 16-9 主机发送模式流程与状态

### 16.8.2 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 读数为 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。

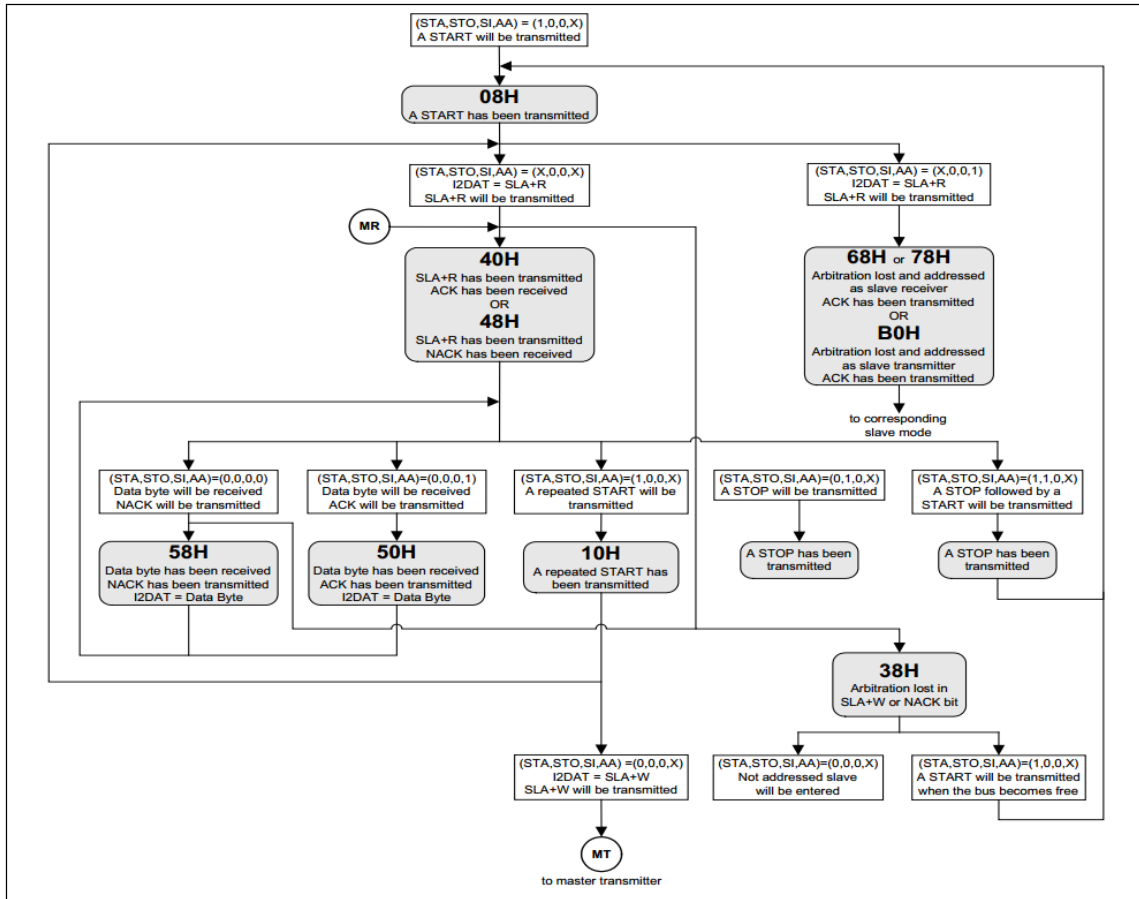


Figure 16-10 主机接收模式流程与状态

### 16.8.3 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。

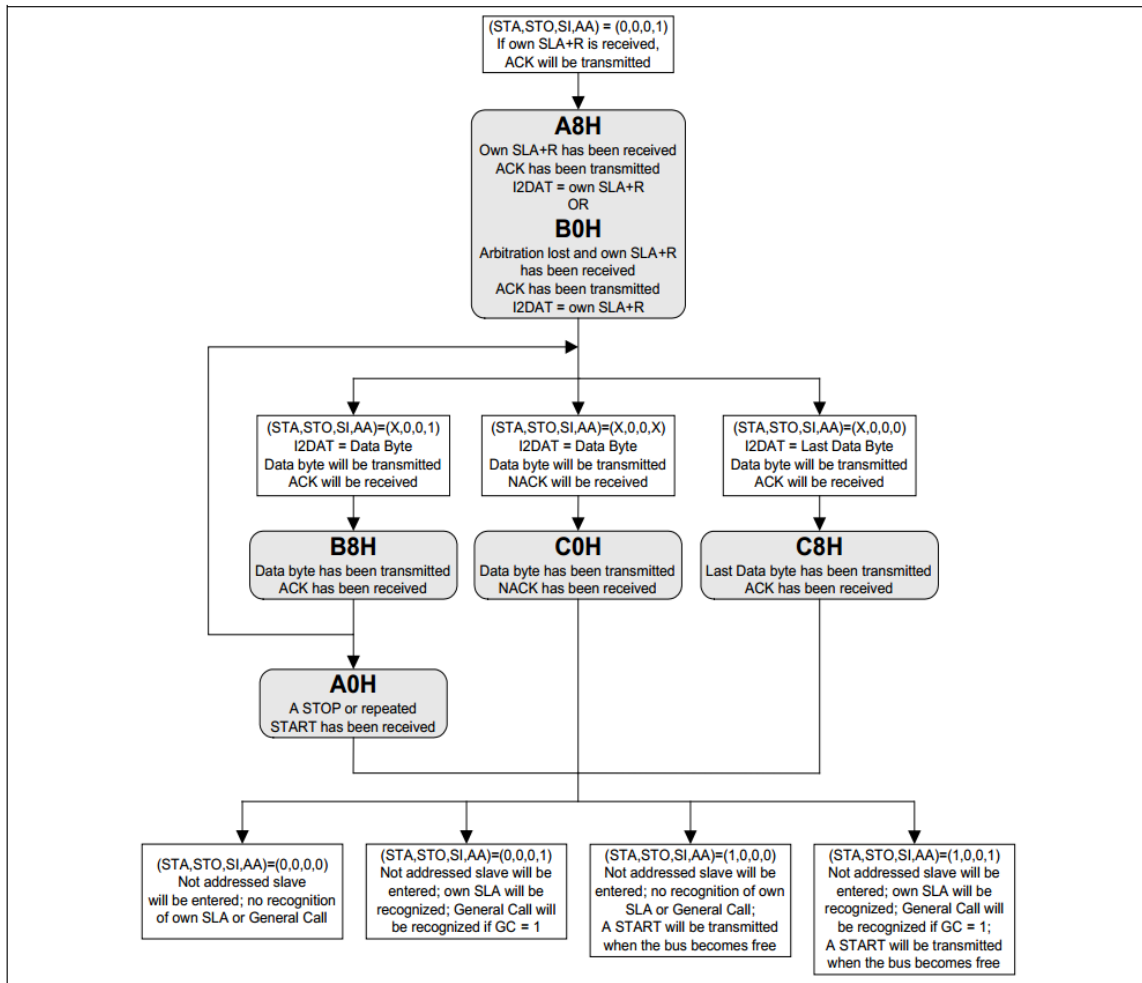


Figure 16-11 从机发送模式流程与状态

### 16.8.4 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。

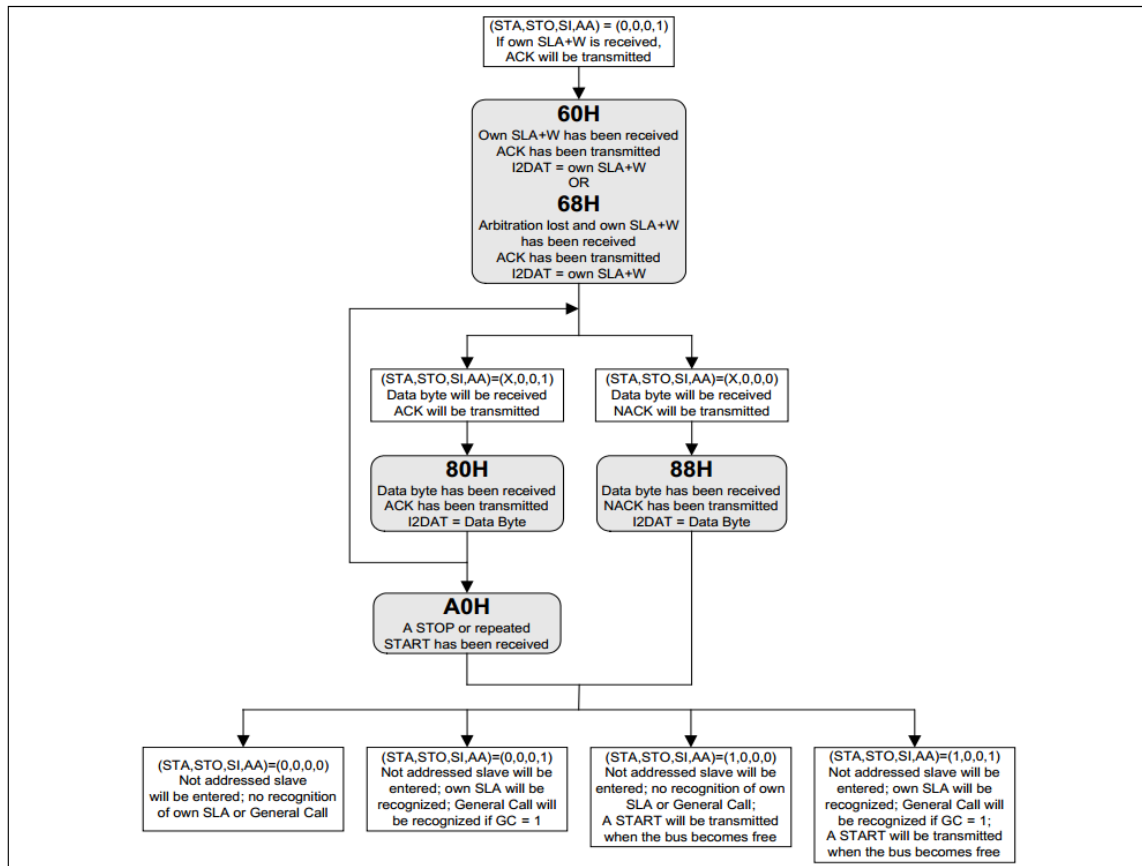


Figure 16-12 从机接收模式流程与状态

### 16.8.5 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。

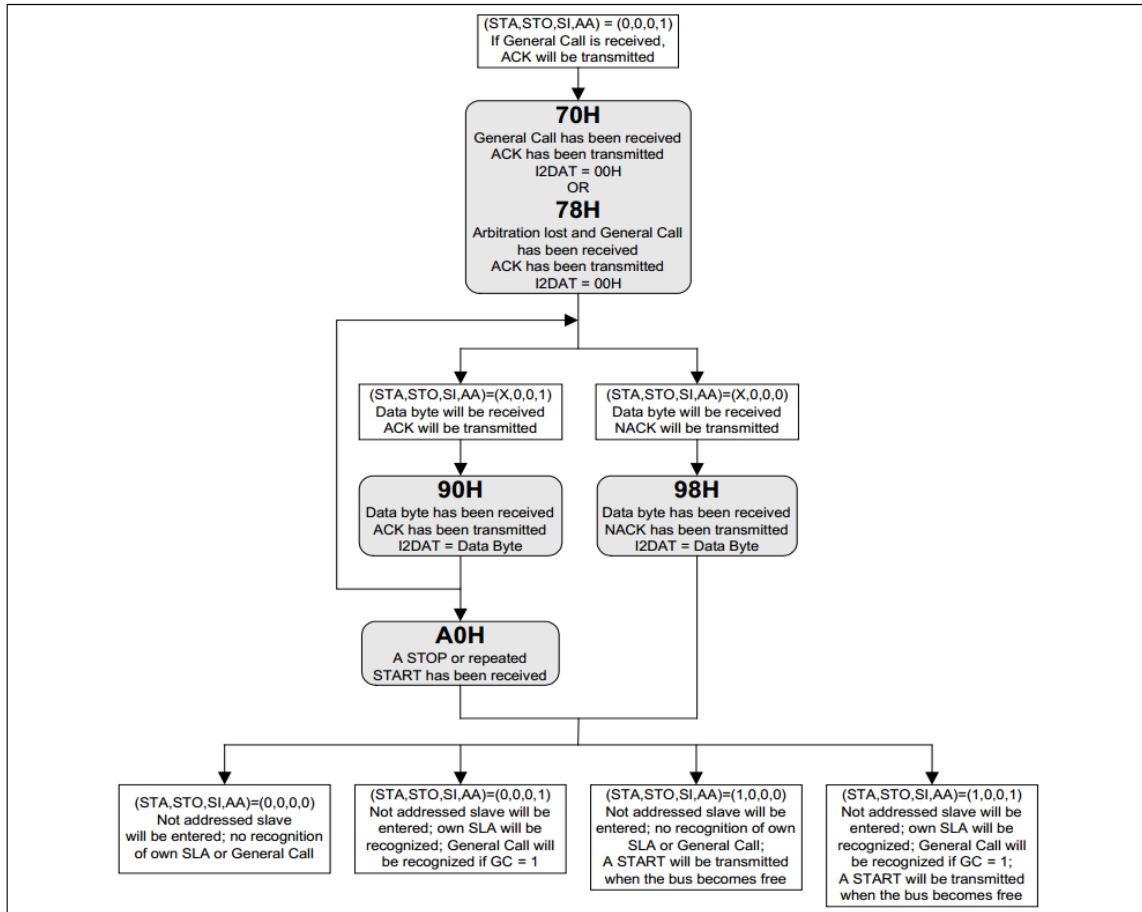


Figure 16-13 广播呼叫模式流程与状态

### 16.8.6 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 由硬件清零且在没有停止信号就释放 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。

## 16.9 IIC 总线相关寄存器

### 16.9.1 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>CPU</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	UART2 的 BRT 独立波特率发生器的溢出率/8					

### 16.9.2 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

### 16.9.3 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。

## 16.9.4 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。



# 17 模数转换ADC

## 17.1 ADC 特性

- 最多 23 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 2V、3V、4V、VDD 及外部 Vref
- 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
- ADC 唤醒中断

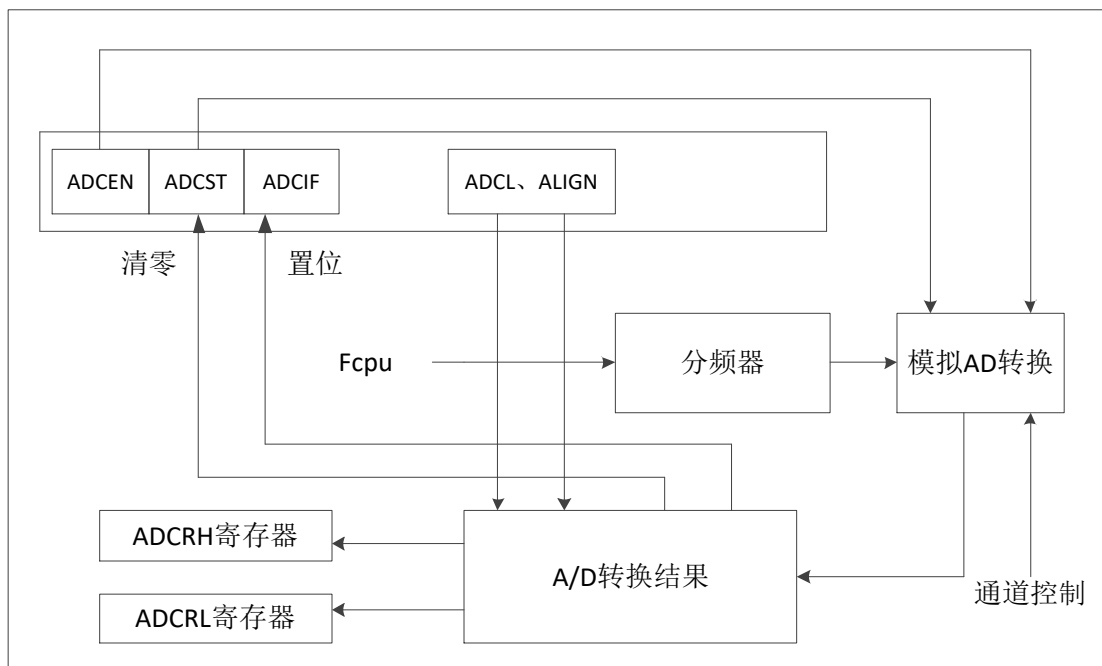


Figure 17-1 ADC 功能框图

## 17.2 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 KΩ）；
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 Figure17-3 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。

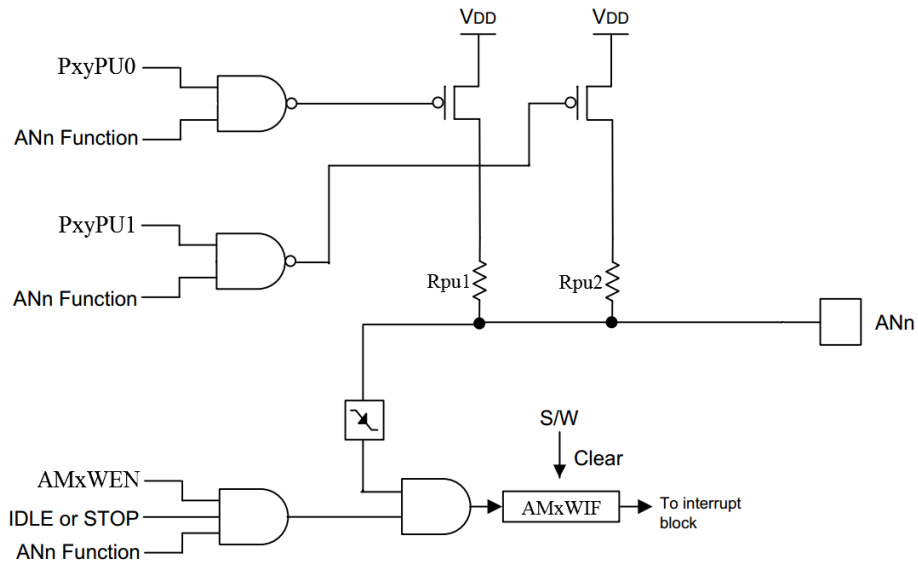


Figure 17-2 ADC 省电唤醒功能框图

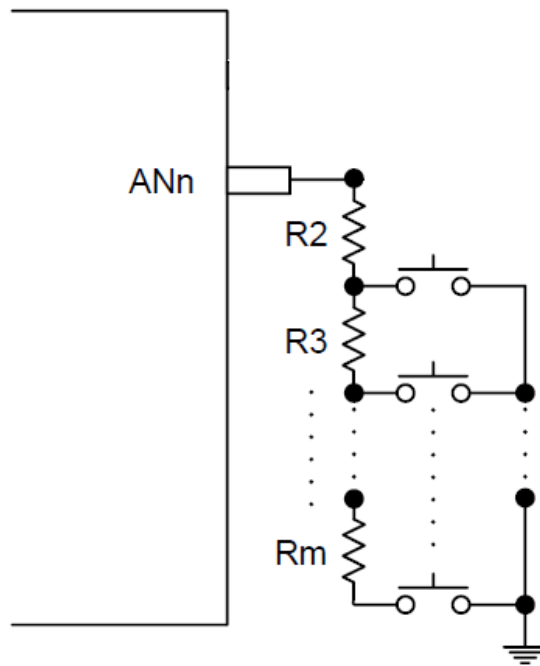


Figure 17-3 ADC 按键输入串联电阻应用参考图

## 17.3 ADC 相关寄存器

### 17.3.1 ADC 控制寄存器 ADCC0、ADCC1、ADCC2

#### ADCC0

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	0	0	0	0	0	0	0	0
位符号	ADCEN	ADCST	ADCIF	-		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: 1. ADCEN 置 1 后, 建议延时 20us 再启动 ADC 转换; 2. 在掉电模式下, ADCEN 强制为 0; 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) 注意: 软件清零 ADCIF 和 ADCST 置 1 不可同时操作。
4-3	-	保留位 (读为 0, 写无效)
2	VREFS	Vref 选择 0, 选内部 Vref 1, 选外部 VREF_H。 (此时 P2.5 只做 ADC 参考电压输入, 且此端口必须设为模拟输入)
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: 内部参考电压选择为 2/3/4V 时, VDD 需要高于内部参考电压 0.5V 以上。

**ADCC1**

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	ICHS[1:0]		-	XCHS[4:0]				

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 注: 在进行内部通道选择时, 外部通道选择 XCHS[4:0]应配置为 11111, 否则可能会造成内部通道和外部通道同时打开的情况。
5	-	保留位 (读为 0, 写无效)
4-0	XCHS[4:0]	ADC 外部输入通道选择 XCHS[4:0] = x(x = 0...23), 表示当前检测通道为 AN <sub>x</sub> , 如 XCHS[4:0] = 3, 表示当前检测通道为外部通道 AN3。外部通道除设置 XCHS[4:0], 还需设置对应管脚的功能为模拟输入。

**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 注意: 1. 当参考电压为内部 2V、3V、4V 时, 为保证 ADC 转换精度, 建议 ADC 时钟在 2MHz 及 2MHz 以下, 同时需要配置这三位。 2. 当参考电压为 VDD 时, ADC 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。
2-0	ADCS[2:0]	ADC 时钟选择位 000: Fcpu/2 001: Fcpu/4

		010: Fcpu/6 011: Fcpu/8 100: Fcpu/12 101: Fcpu/16 110: Fcpu/24 111: Fcpu/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

### 17.3.2 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。

### 17.3.3 ADC 唤醒控制寄存器 ADCWC0、ADCWC1

#### ADCWC0

位编号	7	6	5	4	3	2	1	0
R/W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0
位符号	AM0WEN	AM0WIF	-	CHSW0[4:0]				

#### ADCWC1

位编号	7	6	5	4	3	2	1	0
R/W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0
位符号	AM1WEN	AM1WIF	-	CHSW1[4:0]				

位编号	位符号	说明
7	AM <sub>x</sub> WEN (x = 0,1)	ADC 唤醒模块 x(x = 0,1)允许位 0: 禁止 ADC 唤醒模块 x 1: 允许 ADC 唤醒模块 x 注: 允许 ADC 唤醒模块 x 时, 需要关闭 ADC 转换功能。
6	AM <sub>x</sub> WIF (x = 0,1)	ADC 唤醒模块 x(x = 0,1)中断标志位 (与 ADC 转换共用中断向量) 0: 无 ADC 唤醒模块 x 中断, 软件清 0 1: 发生 ADC 唤醒模块 x 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。
5	-	保留位
4-0	CHSW <sub>x</sub> [4:0]	ADC 唤醒模块 x(x = 0,1)输入通道选择位 CHSW <sub>x</sub> [4:0] = y(y = 0...23), 表示唤醒模块 0 唤醒通道选择为 AN <sub>y</sub> , 如 CHSW0[4:0] = 3, 表示当前唤醒模块 0 唤醒通道为 AN3。 注: 1. 通道选择必须允许唤醒模块才有效, 且需要使用的唤醒通道端口必须预先设定为模拟输入端口。 2. 唤醒电阻通过端口上拉电阻来配置。

注: 这两个寄存器只能写, 不能读。

# 18 低电压检测LVD

## 18.1 LVD 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，并且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

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中断不能将芯片从掉电模式唤醒，但中断可以唤醒IDLE模式。

## 18.2 LVD 相关寄存器

### 18.2.1 LVD 控制寄存器 LVDC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	LV DEN	-	LV DIE	LV DM	LV DF	LV DV		

位编号	位符号	说明
7	LV DEN	LVD 使能位 0: 禁止 LVD 1: 允许 LVD，掉电模式下不会自动关闭
6	-	系统使用，请勿操作。
5	LV DIE	LVD 中断允许位 0: 禁止 LVD 中断 1: 允许 LVD 中断 注：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。
4	LV DM	LVD 检测模式选择位 0: VDD 电压或端口电压小于检测点时，LVDF 硬件置 1 1: VDD 电压或端口电压大于检测点时，LVDF 硬件置 1
3	LV DF	低电压检测标志位 0: 必须软件清 0 1: VDD 电压或端口电压低于（或高于）检测选择电压时，硬件置 1，也做中断请求 注：LVDM = 0: VDD 电压或端口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 电压或端口电压持续高于检测电压时，软件清除才能起作用，如果 VDD 电压或端口电压持续持续低于检测电压，软件是无法清除 LVDF 的。 LVDM = 1，同理。
2-0	LV DV[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：LVD 检测电压设置在 BOR 电压之上才有意义。

## 18.2.2 LVD 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 <sub>CPU</sub> + 2T <sub>CPU</sub>

注：掉电模式下自动关闭，退出掉电模式又自动打开。



# 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。

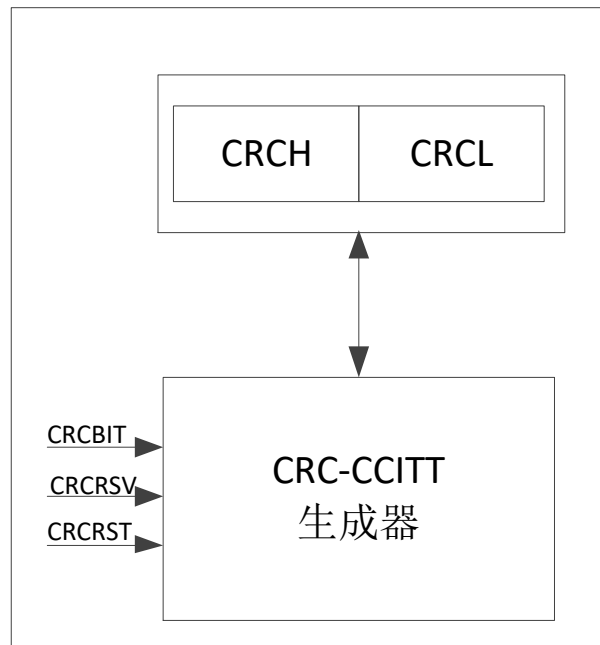


Figure 19-1 CRC 功能框图

## 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

- 0: P4.7 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
- 1: P4.7 为普通 IO 引脚

### 2. 外部复位电平选择

- 0: 高电平复位（默认）
- 1: 低电平复位

### 3. 外部晶振选择

- 0: 低频晶振 32.768KHz（默认）
- 1: 高频晶振

### 4. 复位后等待时间

- 0: 1ms
- 1: 4ms
- 2: 8ms（默认）
- 3: 16ms

### 5. 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

### 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>OTG</sub>	-40	-	+85	°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 32KHz	2.0	5.0	5.5	V
工作电流	I <sub>OP1</sub>	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	-	5.0	-	mA
		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	-	3.2	-	
		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	-	2.1	-	
		F <sub>CPU</sub> =2MHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	1.5	-	
		F <sub>CPU</sub> =1MHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	1.2	-	
		F <sub>CPU</sub> =500KHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	1.1	-	
	I <sub>OP2</sub>	F <sub>OSC</sub> =32KHz, 无负载, 无浮动输入管脚, 执行 NOP 指令, 其它模块关闭	-	1.0	-	mA
	I <sub>ID1</sub>	F <sub>CPU</sub> =16MHz, 无负载, 无浮动输入管脚, 其它模块关闭	-	7.0	-	mA
	I <sub>ID2</sub>	F <sub>OSC</sub> =32KHz, 无负载, 无浮动输入管脚, 其它模块关闭	-	1.0	-	mA
	I <sub>PD</sub>	进入掉电模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	7.0	-	μA
I <sub>IDLE1</sub>	F <sub>CPU</sub> =16MHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	2	-	mA	

	I <sub>IDLE2</sub>	F <sub>CPU</sub> =8MHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	1.3	-	mA
	I <sub>IDLE3</sub>	F <sub>CPU</sub> =4MHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	1.0	-	mA
	I <sub>IDLE4</sub>	F <sub>CPU</sub> =2MHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	0.8	-	mA
	I <sub>IDLE5</sub>	F <sub>CPU</sub> =1MHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	0.7	-	mA
	I <sub>IDLE6</sub>	F <sub>CPU</sub> =125KHz, 进入空闲模式, 无负载, 无浮动输入管脚, 所有模块关闭	-	0.7	-	mA
	I <sub>RTC</sub>	RTC 使能中断, 进入掉电模式, 无浮动输入管脚, ADC 参考电压选择非 VDD, 其他模块关闭	-	50	-	μA
WDT 电流	I <sub>WDT</sub>	VDD = 5V	-	1.2	-	μA
LVD 电流	I <sub>LVD</sub>	VDD = 5V	-	11.0	-	
RTC 电流	I <sub>RTC</sub>	VDD = 5V, 系统进入掉电模式	-	11.0	-	μA
输入低电压 1	V <sub>IL1</sub>	I/O 端口非施密特输入	GND	-	0.3*VDD	V
输入高电压 1	V <sub>IH1</sub>	I/O 端口非施密特输入	0.7*VDD	-	VDD	V
输入低电压 2	V <sub>IL2</sub>	I/O 端口施密特输入	GND	-	0.2*VDD	V
输入高电压 2	V <sub>IH2</sub>	I/O 端口施密特输入	0.8*VDD	-	VDD	V
输入漏电流	I <sub>ILC1</sub>	I/O端口输入模式, V <sub>IN</sub> = VDD 或GND	-1	0	1	μA
输出漏电流	I <sub>OLC1</sub>	I/O端口输出模式, V <sub>OUT</sub> = VDD 或GND	-1	0	1	μA
灌电流	I <sub>OL1</sub>	V <sub>out</sub> =GND+0.6, High Drive Mode	-	25	-	mA
	I <sub>OL2</sub>	V <sub>out</sub> =GND+0.6, Low Drive Mode	-	8	-	
	I <sub>OL3</sub>	V <sub>out</sub> =GND+1.5, High Drive Mode 只限 P3.0-P3.7(+1.5V@5V)	-	100	-	
	I <sub>OL4</sub>	V <sub>out</sub> =GND+1.5, Low Drive Mode 只限 P3.0-P3.7(+1.5V@5V)	-	35	-	
拉电流	I <sub>OH1</sub>	V <sub>out</sub> =VDD-0.6, High Drive Mode	-	15	-	
	I <sub>OH2</sub>	V <sub>out</sub> =VDD-0.6, Low Drive Mode	-	5	-	
上拉电阻	R <sub>PU1</sub>	普通端口, V <sub>IN</sub> =GND	-	30	-	kΩ
	R <sub>PU2</sub>	普通端口, V <sub>IN</sub> =GND	-	50	-	
	R <sub>PU3</sub>	普通端口, V <sub>IN</sub> =GND	-	120	-	
	R <sub>PU4</sub>	普通端口, V <sub>IN</sub> =GND	-	230	-	
	R <sub>PU5</sub>	外部复位端口低有效, V <sub>IN</sub> =GND	-	30	-	
下拉电阻	R <sub>PD1</sub>	普通端口, V <sub>IN</sub> =VDD	-	45	-	
	R <sub>PD2</sub>	外部复位端口高有效, V <sub>IN</sub> =GND	-	45	-	
ADC 唤醒电压	V <sub>AW1</sub>	常温, VDD = 5V	4.0	4.2	4.4	V
	V <sub>AW2</sub>	常温, VDD = 3V	2.3	2.5	2.6	
RAM 保持电压	V <sub>RAM</sub>	-	-	0.7	-	

注：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, VDD=5V	-	-	5	μs
内部 RC38K 启动时间	Tset2	常温, VDD=5V	-	-	150	μs
外部高频振荡器 启动时间	Tset3	16MHz, 常温, VDD=5V	-	200	-	μs
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, VDD=5V	-	2	-	s
频率精度	FIRC1	VDD=2V~5.5V, 25°C	32(1-1%)	32	32(1+1%)	MHz
	FIRC2	VDD=5.0V, -40°C ~+85°C	32(1-1%)	32	32(1+1%)	MHz
	FWRC	-	19	38	57	KHz

## 22.4 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	-	2.5	5.0	5.5	V
精度	NR	$GND \leq VAIN \leq Vref$	-	12	-	bit
A/D 输入电压	VAIN	-	GND	-	Vref	V
A/D 输入电阻	RAIN	VAIN=5V	2	-	-	MΩ
模拟电压源推荐阻抗	ZAIN	-	-	-	1	kΩ
A/D 转换电流	IAD	ADC 模块打开, VDD=5.0V	-	1	3	mA
A/D 输入电流	IADIN	VDD=5.0V	-	-	10	μA
微分非线性误差	DLE	VDD=5.0V	-	-	±1	LSB
积分非线性误差	ILE	VDD=5.0V, Vref =2V	-	-	±4	LSB
		VDD=5.0V, Vref =3V	-	-	±5	
		VDD=5.0V, Vref =4V	-	-	±6	
		VDD=5.0V, Vref =VDD	-	-	±7	
		VDD=5.0V, Vref =外参	-	-	±7	
满刻度误差	EF	VDD=5.0V	-	-	±10	LSB
偏移量误差	EZ	VDD=5.0V	-	-	±5	LSB
总绝对误差	EAD	VDD=5.0V	-	-	±10	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

## 22.5 FLASH 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
扇区擦除时间	TERASE	1 个扇区 (128 字节)	-	5	-	ms
字节写入时间	TPROG	1 个字节, Fcpu=16MHz	-	23	-	us
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

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

## 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 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 1	VLVD1	LVD 使能, VDD=2V~5.5V	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 其他电气特性

- 1、ESD (HBM): CLASS 3A ( $\geq 4000V$ )
- 2、ESD (MM): CLASS 2 ( $\geq 200V$ )
- 3、Latch\_up: CLASS I (200mA)

## 23 开发工具

### 23.1 HC-LINK 仿真工具

HC89S105xx 使用 HC-LINK 仿真器进行程序的下载和仿真，HC-LINK 通过 JTAG 接口可以对芯圣所有的增强型 8051 内核单片机实现下载和仿真。关于 HC-LINK 的使用，请参见 HC-LINK 用户手册。

- 支持 Keil C51 集成编译环境（uVision2.34 及以上版本）
- 支持所有的芯圣 8051 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，使用 JTAG 方式烧录，适用于芯圣 8051 内核系列的 Flash MCU（非固化 ISP）的烧录。关于 HC-PM51 的使用，请参见 HC-PM51 工具用户手册。

- 采用 USB 方式连接
- 支持单路脱机烧录

### 23.3 ISP 串口烧录

HC89S105xx 系列芯片支持 ISP 串口烧录的功能，用户可以使用 HC-LINK 或 HC-PM51 完成 ISP 固件程序的烧录，详见相关应用手册

HC-LINK V4.0 支持 USB 转串口的功能，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，无需冷启动，非常方便用户程序的下载。关于 HC-ISP 的使用，请参见 HC-ISP 工具使用说明。

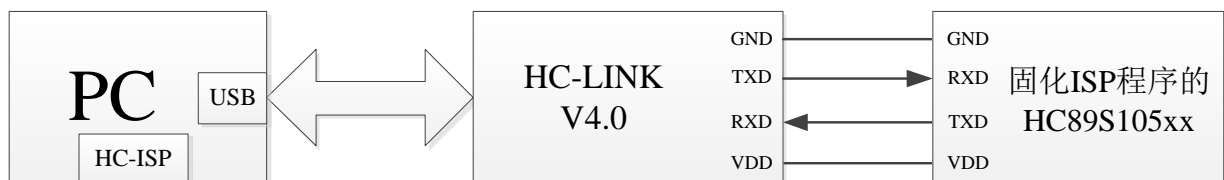


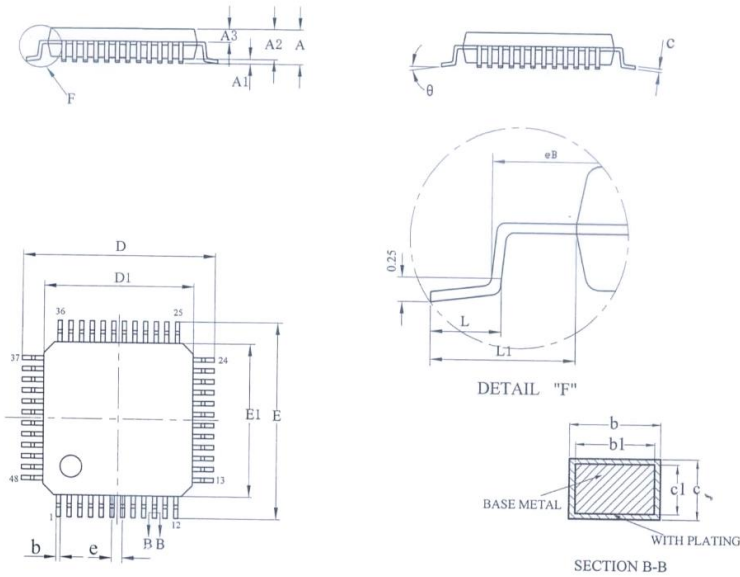
Figure 23-1 ISP 串口烧录框图

### 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

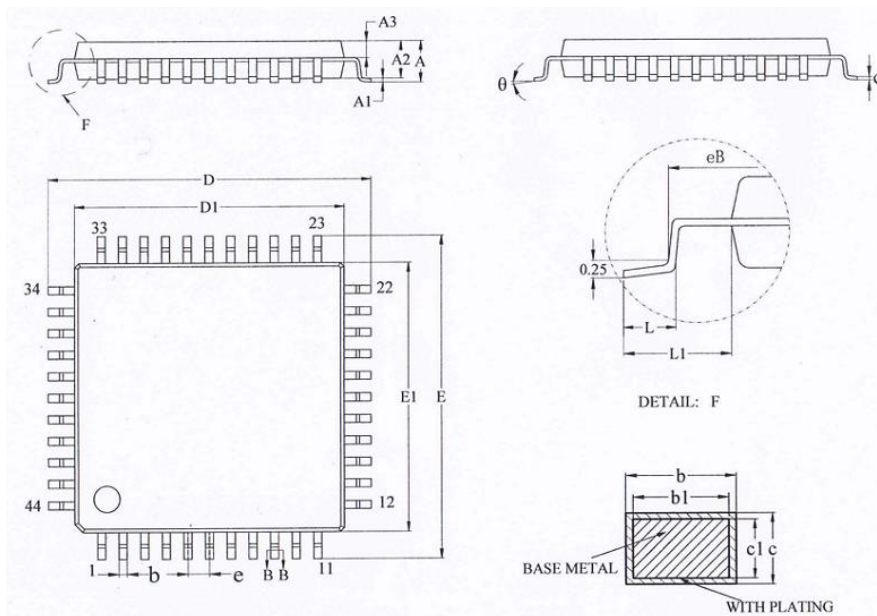
## 24.1 LQFP48



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.50BSC		
L	0.45	—	0.75
L1	1.00BSC		
θ	0	—	7°

Figure 24-1 LQFP48 封装尺寸

## 24.2 LQFP44



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.28	—	0.36
b1	0.27	0.30	0.33
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e	0.80BSC		
eB	11.05	—	11.25
L	0.45	—	0.75
L1	1.00REF		
θ	0	—	7°

Figure 24-2 LQFP44 封装尺寸

### 24.3 LQFP32

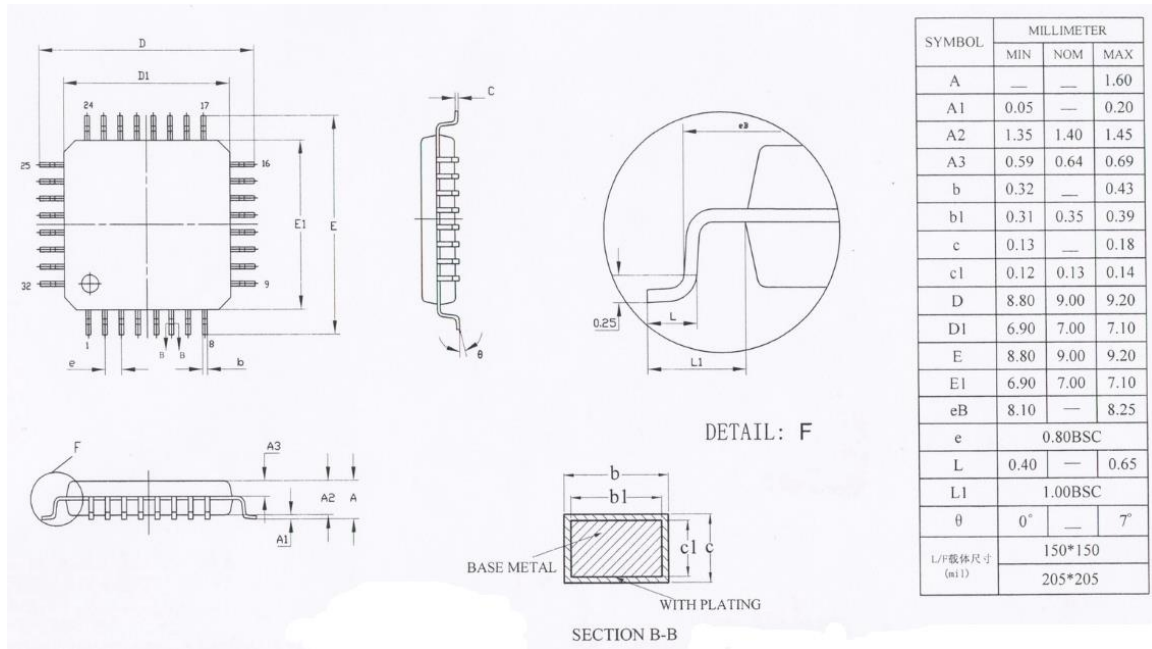


Figure 24-3 LQFP32 封装尺寸



## 25 版本记录

版本	日期	描述
Ver1.00	2017-09-12	第一版
Ver1.01	2017-11-28	增加了 ISP 烧录口描述
Ver1.02	2017-12-05	1、增加 1.5 章节 2、修改 7.2 章节端口上电复位后描述
Ver1.03	2018-05-10	删除 HC89S105C4、HC89S105K4
Ver1.04	2018-12-27	1、增加指令表章节 2、增加 FLASH 电气特性 3、将应用注意事项单独写成一个小节，并增加了一些注意事项
Ver1.05	2019-08-16	1、增加 LQFP44 封装 2、完善指令表 3、完善部分描述

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

2018 年 5 月