

STC12C5628AD 系列
STC12C5410AD 系列
STC12C2052AD 系列
单片机技术参考手册

目录

1	单片机基础概述.....	1
1.1	数制与编码.....	1
1.1.1	数制转换.....	1
1.1.2	原码、反码及补码.....	4
1.1.3	常用编码.....	4
1.2	几种常用的逻辑运算及其图形符号.....	5
1.3	STC12C5628AD 系列单片机性能概述.....	8
1.4	STC12C5410AD 系列单片机性能概述.....	8
2	特性、价格及管脚.....	9
2.1	STC12C5628AD 系列.....	9
2.2	STC12C5410AD 系列.....	14
3	封装尺寸图.....	24
3.1	SOP16 封装尺寸图.....	24
3.2	SOP20 封装尺寸图.....	25
3.3	SOP28 封装尺寸图.....	26
3.4	LQFP32 封装尺寸图（9mm*9mm）.....	27
3.5	LQFP44 封装尺寸图（12mm*12mm）.....	错误!未定义书签。
3.6	LQFP48 封装尺寸图（9mm*9mm）.....	错误!未定义书签。
4	ISP 下载.....	28
4.1	ISP 下载流程及典型应用线路图.....	28
4.1.1	ISP 下载流程图.....	28
4.1.2	使用一箭双雕之 USB 转串口工具下载.....	29
4.1.3	使用 USB 转双串口/TTL 下载（有外部晶振）.....	30
4.1.4	使用 USB 转双串口/TTL 下载（无外部晶振）.....	31
4.1.5	使用 USB 转双串口/TTL 下载（自动停电/上电）.....	32
4.1.6	使用 RS-232 转换器下载，也可支持仿真.....	33
4.1.7	使用 PL2303-GL 下载，也可支持仿真.....	34
4.1.8	单片机电源控制参考电路.....	35
5	复位、省电模式与系统电源管理.....	36
5.1	系统复位.....	36
5.1.1	看门狗复位（WDT_CONTR）.....	37
5.1.2	软件复位（IAP_CONTR）.....	38
5.1.3	高电平上电复位参考电路.....	39
5.2	时钟停振/省电模式与系统电源管理.....	40
5.2.1	电源控制寄存器（PCON）.....	40
6	存储器.....	41
6.1	程序存储器.....	41
6.2	数据存储器.....	42
6.2.1	内部 RAM.....	42
6.2.2	程序状态寄存器（PSW）.....	43

6.2.3	堆栈指针 (SP)	43
6.2.4	内部扩展 RAM, XRAM, XDATA	43
6.2.5	辅助寄存器 (AUXR)	44
6.2.6	外部扩展 RAM, XRAM, XDATA	45
6.2.7	8051 中可位寻址的数据存储器	46
7	特殊功能寄存器	48
7.1	特殊功能寄存器列表	49
8	I/O 口	51
8.1	I/O 口相关寄存器	51
8.1.1	端口数据寄存器 (Px)	51
8.1.2	端口模式配置寄存器 (PxM0, PxM1)	51
8.2	配置 I/O 口	53
8.3	I/O 的结构图	54
8.3.1	准双向口 (弱上拉)	54
8.3.2	推挽输出	54
8.3.3	高阻输入	55
8.3.4	开漏输出	55
8.4	一种典型三极管控制电路	56
8.5	典型发光二极管控制电路	56
8.6	混合电压供电系统 3V/5V 器件 I/O 口互连	57
8.7	如何让 I/O 口上电复位时为低电平	58
8.8	利用 74HC595 驱动 8 个数码管(串行扩展,3 根线)的线路图	59
8.9	I/O 口直接驱动 LED 数码管应用线路图	60
9	指令系统	61
10	中断系统	65
10.1	中断源	65
10.2	中断列表	66
10.3	中断相关寄存器	67
10.3.1	中断使能寄存器 (中断允许位)	67
10.3.2	中断请求寄存器 (中断标志位)	69
10.3.3	中断优先级寄存器	70
11	定时器/计数器	71
11.1	定时器的相关寄存器	72
11.2	定时器 0/1	73
11.2.1	定时器 0/1 控制寄存器 (TCON)	73
11.2.2	定时器 0/1 模式寄存器 (TMOD)	73
11.2.3	定时器 0 模式 0	74
11.2.4	定时器 0 模式 1	75
11.2.5	定时器 0 模式 2	76
11.2.6	定时器 0 模式 3	76
11.2.7	定时器 1 模式 0	77
11.2.8	定时器 1 模式 1	78
11.2.9	定时器 1 模式 2	78
11.2.10	定时器 0 计数寄存器 (TL0, TH0)	79

11.2.11	定时器 1 计数寄存器 (TL1, TH1)	79
11.2.12	辅助寄存器 1 (AUXR)	79
11.2.13	中断与时钟输出控制寄存器 (INT_CLKO)	80
12	串口通信	81
12.1	串口相关寄存器	81
12.2	串口 1	82
12.2.1	串口 1 控制寄存器 (SCON)	82
12.2.2	串口 1 数据寄存器 (SBUF)	82
12.2.3	电源管理寄存器 (PCON)	83
12.2.4	辅助寄存器 1 (AUXR)	83
12.2.5	串口 1 模式 0, 模式 0 波特率计算公式	83
12.2.6	串口 1 模式 1, 模式 1 波特率计算公式	84
12.2.7	串口 1 模式 2, 模式 2 波特率计算公式	87
12.2.8	串口 1 模式 3, 模式 3 波特率计算公式	87
12.2.9	自动地址识别	88
12.2.10	串口 1 从机地址控制寄存器 (SADDR, SADEN)	88
13	IAP/EEPROM/DATA-FLASH	90
13.1	EEPROM 相关的寄存器	90
13.1.1	EEPROM 数据寄存器 (IAP_DATA)	90
13.1.2	EEPROM 地址寄存器 (IAP_ADDR)	90
13.1.3	EEPROM 命令寄存器 (IAP_CMD)	91
13.1.4	EEPROM 触发寄存器 (IAP_TRIG)	91
13.1.5	EEPROM 控制寄存器 (IAP_CONTR)	92
14	ADC 模数转换	93
14.1	ADC 相关的寄存器	93
14.1.1	ADC 控制寄存器 (ADC_CONTR), PWM 触发 ADC 控制	93
14.1.2	ADC 转换结果寄存器 (ADC_RES, ADC_RES1)	94
14.2	10 位 ADC 静态特性	94
15	PCA/CCP/PWM 应用	95
15.1	PCA 相关的寄存器	95
15.1.1	PCA 控制寄存器 (CCON)	96
15.1.2	PCA 模式寄存器 (CMOD)	96
15.1.3	PCA 计数器寄存器 (CL, CH)	96
15.1.4	PCA 模块模式控制寄存器 (CCAPMn)	97
15.1.5	PCA 模块模式捕获值/比较值寄存器 (CCAPnL, CCAPnH)	97
15.1.6	PCA 模块 PWM 模式控制寄存器 (PCA_PWMn)	97
15.2	PCA 工作模式	98
15.2.1	捕获模式	98
15.2.2	软件定时器模式	98
15.2.3	高速脉冲输出模式	99
15.2.4	PWM 脉宽调制模式及频率计算公式	99
15.3	利用 CCP/PCA/PWM 模块实现 8~16 位 DAC 的参考线路图	101
16	同步串行外设接口 SPI	102
16.1	SPI 相关的寄存器	102

16.1.1	SPI 状态寄存器 (SPSTAT)	102
16.1.2	SPI 控制寄存器 (SPCTL), SPI 速度控制.....	102
16.1.3	SPI 数据寄存器 (SPDAT)	103
16.2	SPI 通信方式	104
16.2.1	单主单从.....	104
16.2.2	互为主从.....	104
16.2.3	单主多从.....	105
16.3	配置 SPI.....	106
16.4	数据模式.....	108
附录 A	电气特性.....	109
A.1	绝对最大额定值	109
A.2	直流特性 (3.3V)	110
A.3	直流特性 (5.0V)	111

1 单片机基础概述

——无微机原理的用户请从本章开始学习

这一章主要讲述的内容有：①在数字设备中进行算术运算的基本知识——数制和编码；②数字电路中一些常用逻辑运算及其图形符号。它们是学习单片机这门课程的基础。对于没有微机原理基础的用户和同学，请从这章开始学习。

1.1 数制与编码

数制是人们利用符号进行计数的科学方法。

数制有很多种，常用的数制有：二进制，十进制和十六进制。

进位计数制是把数划分为不同的位数，逐位累加，加到一定数量之后，再从零开始，同时向高位进位。进位计数制有三个要素：数码符号、进位规律和计数基数。下表是各常用数制的总体介绍。

常用的数制	表示符号	数码符号	进制规律	计数基数
二进制	B	0、1	逢二进一	2
十进制	D	0、1、2、3、4、5、6、7、8、9	逢十进一	10
十六进制	H	0、1、2、3、4、5、6、7、8、9、 A、B、C、D、E、F	逢十六进一	16

我们日常生活中计数一般采用十进制。计算机中采用的是二进制，因为二进制具有运算简单，易实现且可靠，为逻辑设计提供了有利的途径、节省设备等优点。为区别于其它进制数，二进制数的书写通常在数的右下方注上基数 2，或加后面加 B 表示。二进制数中每一位仅有 0 和 1 两个可能的数码，所以计数基数为 2。二进制数的加法和乘法运算如下：

$$\begin{array}{lll} 0 + 0 = 0 & 0 + 1 = 1 + 0 = 1 & 1 + 1 = 10 \\ 0 \times 0 = 0 & 0 \times 1 = 1 \times 0 = 0 & 1 \times 1 = 1 \end{array}$$

由于二进制数在使用中位数太长,不容易记忆，为了便于描述，又常用十六进制作为二进制的缩写。十六进制通常在表示时用尾部标志 H 或下标 16 以示区别。

1.1.1 数制转换

现在我们来介绍这些常用数制之间的转换。

一：二进制 — 十进制转换

方法：将二进制数按权(如下式)展开，然后将各项的数值按十进制数相加，就得到相应的等值十进制数。

例如：N=(1101.101)B，那么 N 所对应的十进制数时多少呢？

$$\text{按权展开 } N=1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 = (13.625)D$$

二：十进制 — 二进制转换

方法：分两部分进行即整数部分和小数部分。

①整数部分转换(基数除法)：

- ★ 把我们要转换的数除以二进制的基数(二进制的基数为 2)，把余数作为二进制的最低位；
- ★ 把上一次得的商在除以二进制基数(即 2)，把余数作为二进制的次低位；
- ★ 继续上一步,直到最后的商为零,这时的余数就是二进制的最高位.

②小数部分转换(基数乘法)：

- ★ 把要转换数的小数部分乘以二进制的基数(二进制的基数为 2)，把得到的整数部分作为二进制小数部分的最高位；
- ★ 把上一步得的小数部分再乘以二进制的基数(即 2)，把整数部分作为二进制小数部分的次高位；
- ★ 继续上一步，直到小数部分变成零为止。或者达到预定的要求也可以。

例如：将 $(213.8125)_{10}$ 化为二进制数可按如下进行：
先化整数部分：

2		213	-----	余数=1= k_0
2		106	-----	余数=0= k_1
2		53	-----	余数=1= k_2
2		26	-----	余数=0= k_3
2		13	-----	余数=1= k_4
2		6	-----	余数=0= k_5
2		3	-----	余数=1= k_6
2		1	-----	余数=1= k_7
		0		

于是整数部分 $(213)_{10}=(11010101)_2$

再化小数部分：

0.8125		
×	2	
-----	1.6250	----- 整数部分=1= k_{-1}
0.6250		
×	2	
-----	1.2500	----- 整数部分=1= k_{-2}
0.2500		
×	2	
-----	0.5000	----- 整数部分=0= k_{-3}
0.5000		
×	2	
-----	1.0000	----- 整数部分=1= k_{-4}

于是小数部分 $(0.8125)_{10}=(0.1101)_2$

综上所述，十进制数 $213.8125=(11010101.1101)_2=(11010101.1101)_B$

三：二进制 — 十六进制转换

方法：二进制和十六进制之间满足 24 的关系，因此把要转换的二进制从低位到高位每 4 位一组，高位不足时在有效位前面添“0”，然后把每组二进制数转换成十六进制即可。

例如：将(010111011110.10110010)B 转换为十六进制数：

$$\begin{array}{ccccccc} & 0101 & 1101 & 1110 & . & 1011 & 0010 \\ & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ = & (5 & D & E & & B & 2) \end{array} \text{H}$$

于是：(010111011110.10110010)B=(5DE.B2)H

四：十六进制 — 二进制转换

方法：十六进制转换为二进制时，把上面二进制转换十六进制的过程逆过来，即转换时只需将十六进制的每一位用等值的 4 位二进制代替就行了。

例如：将(C1B.C6)H 转换为二进制数：

$$\begin{array}{ccccccc} & C & 1 & B & . & C & 6 \\ & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ = & (1100 & 0001 & 1011 & & 1100 & 0110) \end{array} \text{B}$$

于是：(C1B.C6)H=(110000011011.11000110)B

五：十六进制 — 十进制转换

方法：将十六进制数按权(如下式)展开，然后将各项的数值按十进制数相加，就得到相应的等值十进制数。

例如：N=(2A.7F)H，那么 N 所对应的十进制数时多少呢？

$$\text{按权展开 } N=2 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2} = 32 + 10 + 0.4375 + 0.05859375 = (42.49609375) \text{D}$$

于是：(2A.7F)H=(42.49609375)D

六：十进制 — 十六进制转换

方法：将十进制数转换为十六进制数时，可以先将十进制数转换为二进制数，然后再将得到的二进制数转换为等值的十六进制数。

1.1.2 原码、反码及补码

在生活中,数有正负之分,在计算机中是怎样表示数的正负符号呢?

在生活中表示数的时候一般都是把正数前面加一个“+”，负数前面加一个“-”，但是计算机是不认识这些的，通常在二进制数前面增加一位符号位。符号位为“0”表示“+”，符号位为“1”表示“-”。这种形式的二进制数称为原码。如果原码为正数，则原码的反码和补码都与原码相同。如果原码为负数，则将原码(除符号位外)按位取反，所得的新二进制数称为原码的反码，反码加 1 为其补码。

原码、反码、补码这三种形式的总结如下表所示：

	真值	原码	反码	补码
正数	+N	0N	0N	0N
负数	-N	1N	$(2^n-1)+N$	2^n+N

例 1：求+18 和-18 八位原码、反码、补码形式。

真值	原码	反码	补码
+18	00010010	00010010	00010010
-18	10010010	11101101	11101110

1.1.3 常用编码

指定某一组二进制数去代表某一指定的信息，就称为编码。

一：十进制编码

用二进制码表示的十进制数，称为十进制编码。它具有二进制的形式，还具有十进制的特点它可作为人们与数字系统的联系的一种间表示。十进制编码有很多种，最常用的一种是 BCD 码，又称 8421 码。

下面我们用表列出几种常见的十进制编码：

编码种类 十进制数	8421 码 (BCD 码)	余 3 码	2421 码	5211 码	7321 码
0	0000	0011	0000	0000	0000
1	0001	0100	0001	0001	0001
2	0010	0101	0010	0100	0010
3	0011	0110	0011	0101	0011
4	0100	0111	0100	0111	0101
5	0101	1000	1011	1000	0110
6	0110	1001	1100	1001	0111
7	0111	1010	1101	1100	1000
8	1000	1011	1110	1101	1001
9	1001	1100	1111	1111	1010
权	8421		2421	5211	7321

十进制编码分为有权和无权编码。有权编码是指每一位十进制数符均用一组四位二进制码来表示，而且二进制码的每一位都有固定权值。无权编码是指二进制码中每一位都没有固定的权值。上表中 8421 码(即 BCD 码)、2421 码、5211 码、7321 码都是有权编码，而余 3 码是无权编码。

二：奇偶校验码

在数据的存取、运算和传送过程中，难免会发生错误，把“1”错成“0”或把“0”错成“1”。奇偶校验码是一种能检验这种错误的代码。它分为两部分；信息位和奇偶校验位。有奇数个“1”称为奇校验，有偶数个“1”则称为偶校验。

1.2 几种常用的逻辑运算及其图形符号

逻辑代数中常用的运算有：与(AND)、或(OR)、非(NOT)、与非(NAND)、或非(NOR)、与或非(AND-NOR)、异或(EXCLUSIVE OR)、同或(EXCLUSIVE NOR)等。其中与(AND)、或(OR)、非(NOT)运算时三种最基本的运算。

一：与运算及与门

与运算：决定事件结果的全部条件同时具备时，事件才发生。

逻辑变量 A 和 B 进行与运算时可写成： $Y=A \cdot B$

真值表		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

与门：实行与逻辑运算的单元电路。

与门图形符号：

二：或运算及或门

或运算：决定事件结果的各条件中只要有任何一个满足，事件就会发生。

逻辑变量 A 和 B 进行或运算时可写成： $Y=A+B$

真值表		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

或门：实行或逻辑运算的单元电路。

或门图形符号：

三：非运算及非门

非运算：条件具备时，事件不会发生；条件不具备时，事件才会发生。

逻辑变量 A 进行非运算时可写成： $Y=A'$

真值表	
A	Y
0	1
1	0

非门：实行非逻辑运算的单元电路。

非门图形符号：

四：与非运算及与非图形符号

与非运算：先进行与运算，然后将结果求反，最后得到的即为与非运算结果。

逻辑变量 A 和 B 进行与非运算时可写成： $Y=(A \cdot B)'$

真值表		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

与非图形符号：

五：或非运算及或非图形符号

或非运算：先进行或运算，然后将结果求反，最后得到的即为或非运算结果。

逻辑变量 A 和 B 进行或非运算时可写成： $Y=(A+B)'$

真值表		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

或非图形符号：

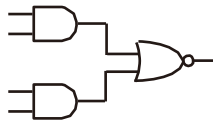
六：与或非运算及与或非图形符号

与或非运算：在与或非逻辑运算中有 4 个逻辑变量 A、B、C、D。假设 A 和 B 为一组，C 和 D 为一组，A、B 之间以及 C、D 之间都是与的关系，只要 A、B 或 C、D 任何一组同时为 1，输出 Y 就是 0。只有当每一组输入都不全是 1 时，输出 Y 才是 1。

逻辑变量 A 和 B 进行或非运算时可写成： $Y=(A \cdot B + C \cdot D)'$

真值表				
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

与或非图形符号：



七：异或运算及异或图形符号

异或运算：当 A、B 不同时，输出 Y 为 1；而当 A、B 相同时，输出 Y 为 0。逻辑变量 A 和 B 进行异或运算时可写成： $Y = A \oplus B = (A \cdot B') + (A' \cdot B)$

真值表		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

异或图形符号：

八：同或运算及同或图形符号

同或运算：当 A、B 不同时，输出 Y 为 0；而当 A、B 相同时，输出 Y 为 1。逻辑变量 A 和 B 进行同或运算时可写成： $Y = A \odot B = (A \cdot B) + (A' \cdot B')$

真值表		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

同或图形符号：

1.3 STC12C5628AD 系列单片机性能概述

STC12C5628AD 系列单片机是宏晶科技生产的单时钟 / 机器周期(1T)的单片机，是高速 / 低功耗 / 超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051,但速度快 8-12 倍，内部集成 MAX810 专用复位电路。4 路 PWM,8 路高速 10 位 A/D 转换,针对电机控制，强干扰场合。

1.4 STC12C5410AD 系列单片机性能概述

STC12C5410AD 系列单片机是宏晶科技生产的单时钟 / 机器周期(1T)的单片机，是高速 / 低功耗 / 超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051,但速度快 8-12 倍，内部集成 MAX810 专用复位电路。4 路 PWM,8 路高速 10 位 A/D 转换,针对电机控制，强干扰场合。

1.5 STC12C2052AD 系列单片机性能概述

STC12C2052AD 系列单片机是宏晶科技生产的单时钟 / 机器周期(1T)的单片机，是高速 / 低功耗 / 超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051,但速度快 8-12 倍，内部集成 MAX810 专用复位电路。4 路 PWM,8 路高速 10 位 A/D 转换,针对电机控制，强干扰场合。

STC12C2052AD 系列只有 2 路 PWM,8 路高速 8 位 A/D 转换。

2 特性、价格及管脚

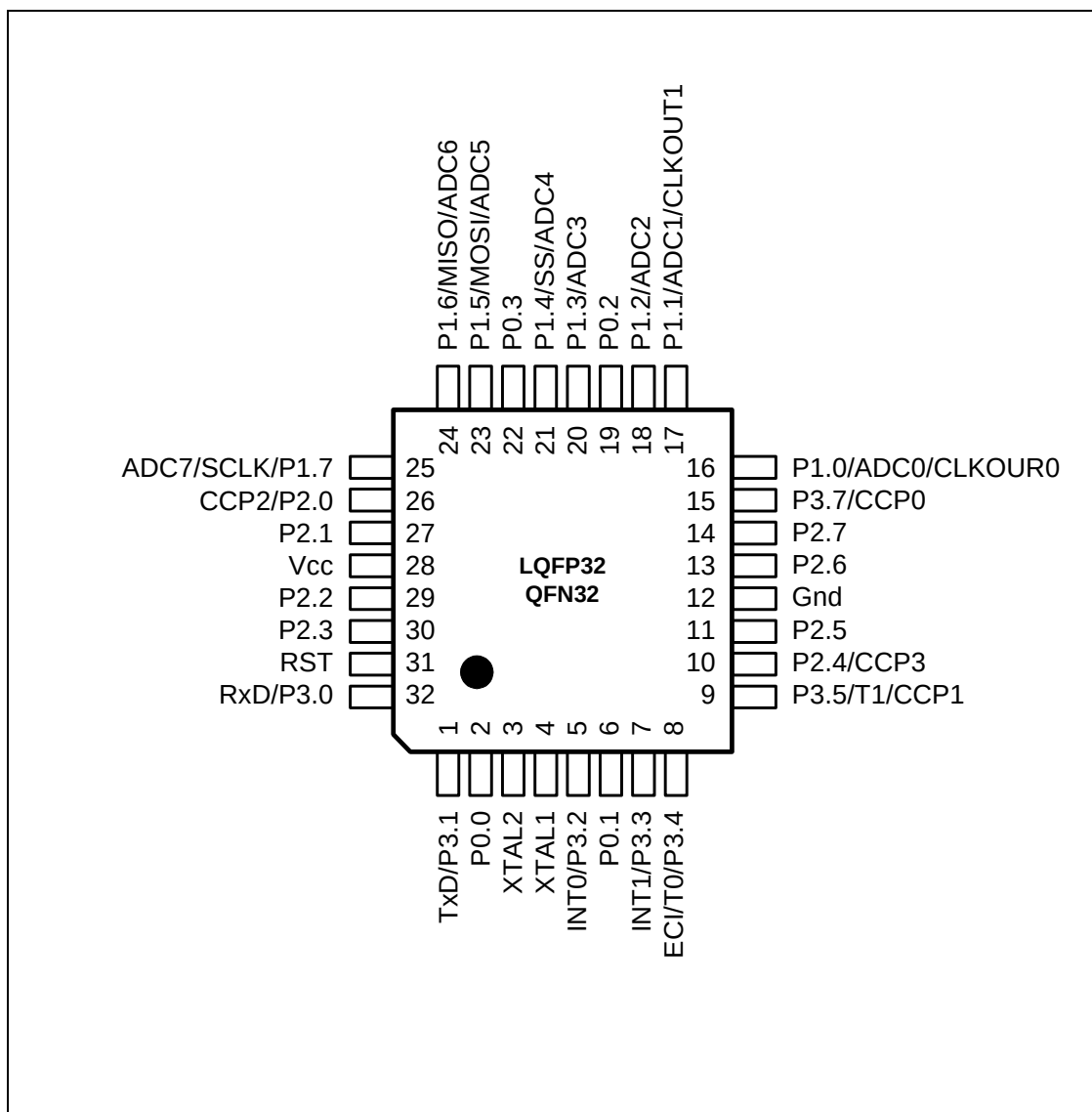
2.1 STC12C5628AD 系列

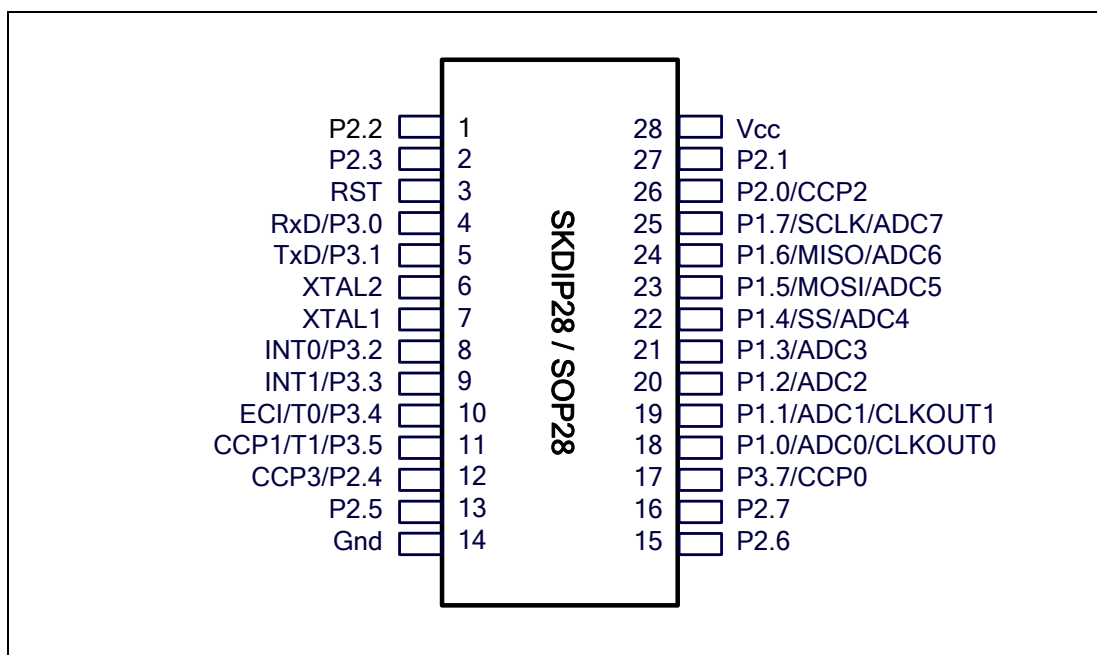
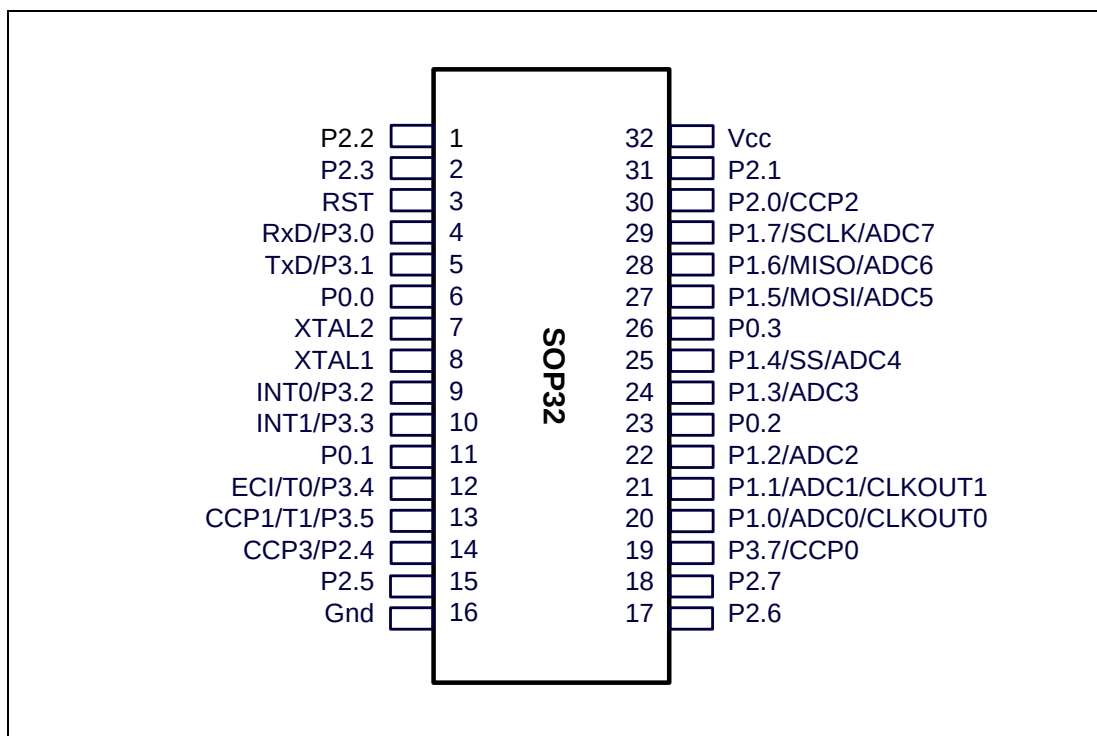
单片机型号	工作电压 (V)	Flash 程序存储器 10 万次 字节	内部大容量扩展 SRAM 字节	EEPROM 10 万次 字节	串口	SPI	定时器 计数器	PCA/CCP/PWM	8 路高速 ADC	看门狗 复位定时器	价格及封装						
											LQFP32	SOP32	SOP28	SKDIP28	SOP20	TSSOP20	SIP20
12C5601AD	3.5-5.5	1K	768	有	1	有	6	4	10 位	有							
12C5602AD	3.5-5.5	2K	768	有	1	有	6	4	10 位	有							
12C5604AD	3.5-5.5	4K	768	有	1	有	6	4	10 位	有							
12C5605AD	3.5-5.5	5K	768	IAP	1	有	6	4	10 位	有							
12C5608AD	3.5-5.5	8K	768	有	1	有	6	4	10 位	有							
12C5612AD	3.5-5.5	12K	768	IAP	1	有	6	4	10 位	有							
12C5616AD	3.5-5.5	16K	768	有	1	有	6	4	10 位	有							
12C5620AD	3.5-5.5	20K	768	有	1	有	6	4	10 位	有							
12C5624AD	3.5-5.5	24K	768	有	1	有	6	4	10 位	有							
12C5628AD	3.5-5.5	28K	768	IAP	1	有	6	4	10 位	有							
12C5630AD	3.5-5.5	30K	768	IAP	1	有	6	4	10 位	有							
12C5601	3.5-5.5	1K	768	有	1	有	6	4		有							
12C5602	3.5-5.5	2K	768	有	1	有	6	4		有							
12C5604	3.5-5.5	4K	768	有	1	有	6	4		有							
12C5605	3.5-5.5	5K	768	IAP	1	有	6	4		有							
12C5608	3.5-5.5	8K	768	有	1	有	6	4		有							
12C5612	3.5-5.5	12K	768	IAP	1	有	6	4		有							
12C5616	3.5-5.5	16K	768	有	1	有	6	4		有							
12C5620	3.5-5.5	20K	768	有	1	有	6	4		有							
12C5624	3.5-5.5	24K	768	有	1	有	6	4		有							
12C5628	3.5-5.5	28K	768	IAP	1	有	6	4		有							
12C5630	3.5-5.5	30K	768	IAP	1	有	6	4		有							
12LE5601AD	2.2-3.6	1K	768	有	1	有	6	4	10 位	有							
12LE5602AD	2.2-3.6	2K	768	有	1	有	6	4	10 位	有							
12LE5604AD	2.2-3.6	4K	768	有	1	有	6	4	10 位	有							
12LE5605AD	2.2-3.6	5K	768	IAP	1	有	6	4	10 位	有							
12LE5608AD	2.2-3.6	8K	768	有	1	有	6	4	10 位	有							
12LE5612AD	2.2-3.6	12K	768	IAP	1	有	6	4	10 位	有							

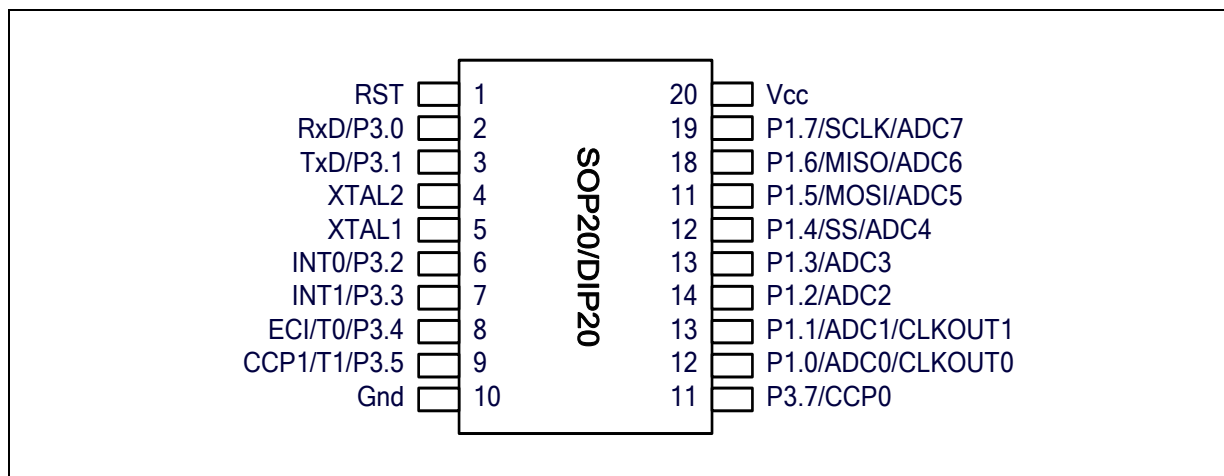
12LE5616AD	2.2-3.6	16K	768	有	1	有	6	4	10 位	有							
12LE5620AD	2.2-3.6	20K	768	有	1	有	6	4	10 位	有							
12LE5624AD	2.2-3.6	24K	768	有	1	有	6	4	10 位	有							
12LE5628AD	2.2-3.6	28K	768	IAP	1	有	6	4	10 位	有							
12LE5630AD	2.2-3.6	30K	768	IAP	1	有	6	4	10 位	有							
12LE5601	2.2-3.6	1K	768	有	1	有	6	4		有							
12LE5602	2.2-3.6	2K	768	有	1	有	6	4		有							
12LE5604	2.2-3.6	4K	768	有	1	有	6	4		有							
12LE5605	2.2-3.6	5K	768	IAP	1	有	6	4		有							
12LE5608	2.2-3.6	8K	768	有	1	有	6	4		有							
12LE5612	2.2-3.6	12K	768	IAP	1	有	6	4		有							
12LE5616	2.2-3.6	16K	768	有	1	有	6	4		有							
12LE5620	2.2-3.6	20K	768	有	1	有	6	4		有							
12LE5624	2.2-3.6	24K	768	有	1	有	6	4		有							
12LE5628	2.2-3.6	28K	768	IAP	1	有	6	4		有							
12LE5630	2.2-3.6	30K	768	IAP	1	有	6	4		有							

1. 增强型 8051 CPU， 1T， 单时钟 / 机器周期， 指令代码完全兼容传统 8051
2. 工作电压： 5.5V - 3.5V（ 5V 单片机） / 3.6V - 2.2V（ 3V 单片机）
3. 工作频率范围： 0 - 35 MHz， 相当于普通 8051 的 0~ 420MHz
4. 用户应用程序空间 30K / 28K / 24K / 20K / 16K / 12K / 8K / 4K 字节
5. 片上集成 768 字节 RAM
6. 通用 I/O 口（ 27/23/15 个）， 复位后为： 准双向口 / 弱上拉（ 普通 8051 传统 I/O 口）
可设置成四种模式： 准双向口 / 弱上拉， 推挽 / 强上拉， 仅为输入 / 高阻， 开漏
每个 I/O 口驱动能力均可达到 20mA， 但整个芯片最大不得超过 55mA
7. ISP（ 在系统可编程） / IAP（ 在应用可编程）， 无需专用编程器， 无需专用仿真器可通过串口（ P3.0/P3.1） 直接下载用户程序， 数秒即可完成一片
8. EEPROM 功能
9. 看门狗
10. 内部集成 MAX810 专用复位电路（ 外部晶体 20M 以下时， 可省外部复位电路）
11. 时钟源： 外部高精度晶体 / 时钟， 内部 R/C 振荡器
用户在下载用户程序时， 可选择是使用内部 R/C 振荡器还是外部晶体 / 时钟
常温下内部 R/C 振荡器频率为： 5.2MHz ~ 6.8MHz
精度要求不高时， 可选择使用内部时钟， 但因为有制造误差和温漂， 应认为是 4MHz ~ 8MHz
12. 共 6 个 16 位定时器 / 计数器，
两个专用 16 位定时器 T0 和 T1
PCA 模块可实现 4 个 16 位定时器
2 个时钟输出口， 可由 T0 的溢出在 P1.0 输出时钟， 可由 T1 的溢出在 P1.1 输出时钟
13. 外部中断 9 路， 下降沿中断或低电平触发中断， PCA 模式可分别或同时支持上升沿中断 / 下降沿中断， Power Down 模式可由外部中断唤醒
INT0/P3.2， INT1/P3.3， P3.4/T0， P3.5/T1， P3.0/RxD， PCA0/P3.7， PCA1/P3.5， PCA2/P2.0， PCA3/P2.4
14. PWM(4 路) / PCA（ 可编程计数器阵列， 4 路）
--- 也可用来当 4 路 D/A 使用

- 也可用来再实现 4 个定时器
- 也可用来再实现 4 个外部中断(上升沿中断 / 下降沿中断均可分别或同时支持)
- 15. A/D 转换, 10 位精度 ADC, 共 8 路
- 16. 通用全双工异步串行口(UART), 由于 STC12 系列是高速的 8051, 也可再用定时器软件实现多串口
- 17. SPI 同步通信口, 主模式 / 从模式
- 18. 工作温度范围: 0 - 75°C / -40 - +85°C
- 19. 封装: LQFP-32, SOP-32/28/20, SKDIP-28, PDIP-20, TSSOP-20(超小封状 6.4mm × 6.4mm, 定货), LQFP/SOP32 有 27 个 I/O 口, SOP28/SKDIP28 有 23 个 I/O 口, SOP20/TSSOP20/PDIP20 有 15 个 I/O 口,
I/O 口不够时, 可用 2 到 3 根普通 I/O 口线外接 74HC164/165/595(均可级联) 来扩展 I/O 口, 还可用 A/D 做按键扫描来节省 I/O 口, 或用双 CPU, 三线通信, 还多了串口。





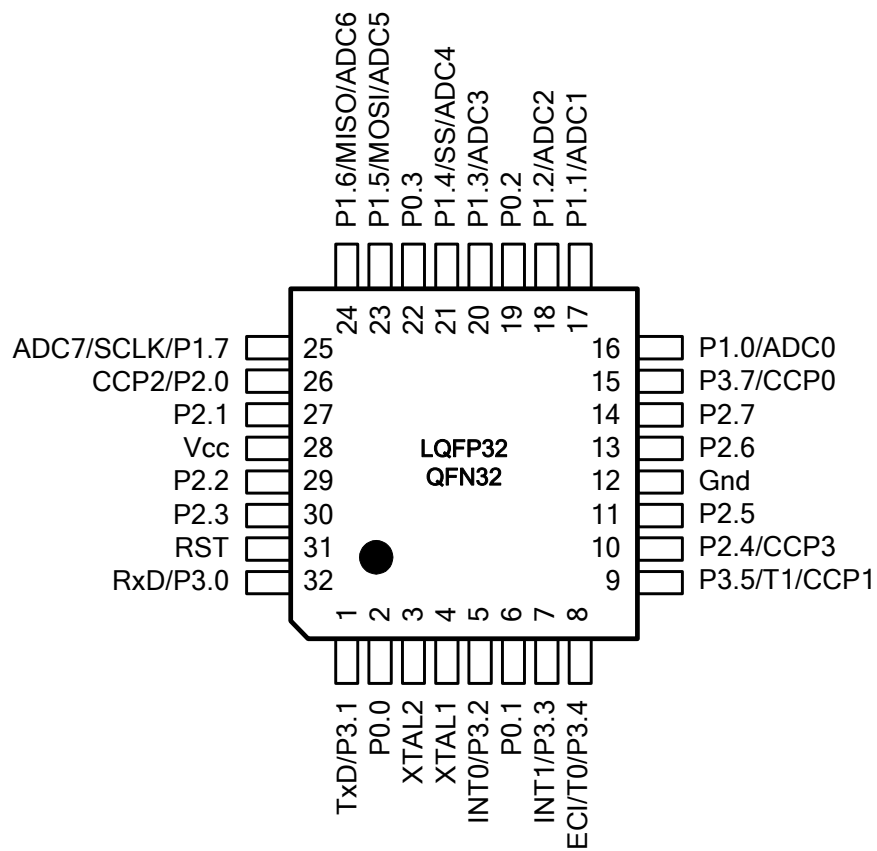


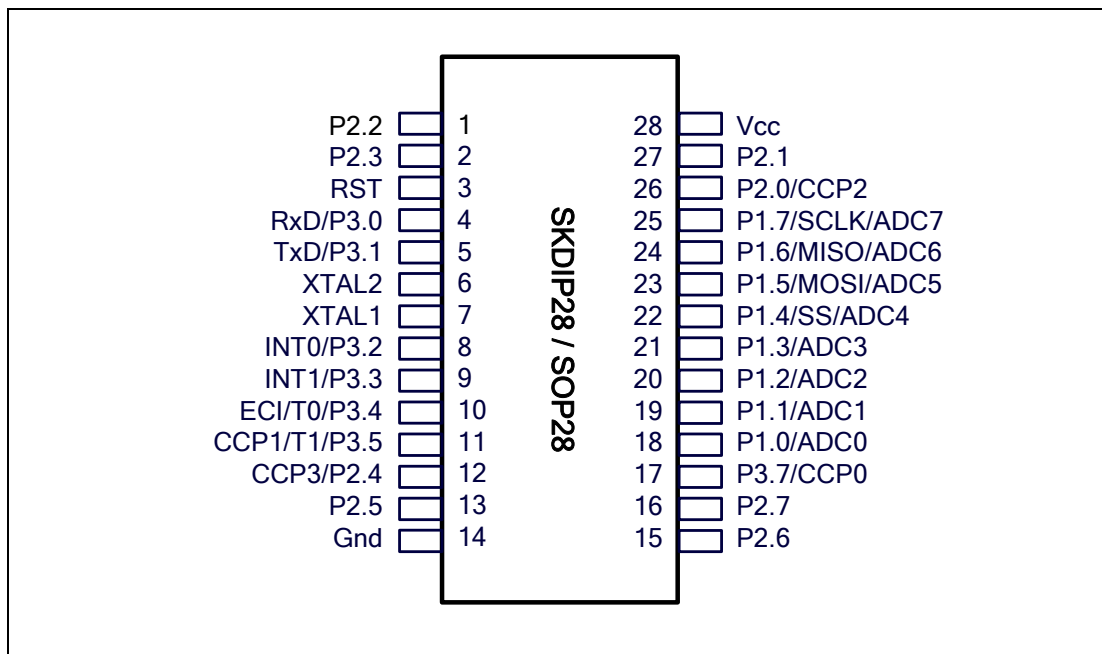
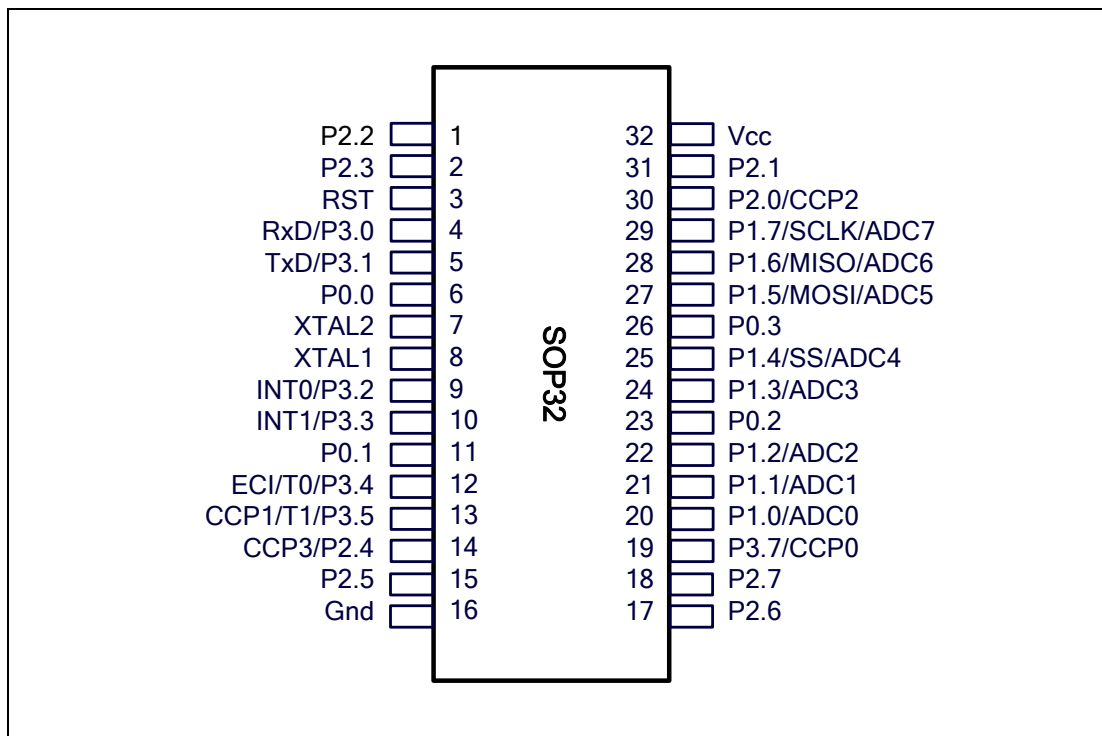
2.2 STC12C5410AD 系列

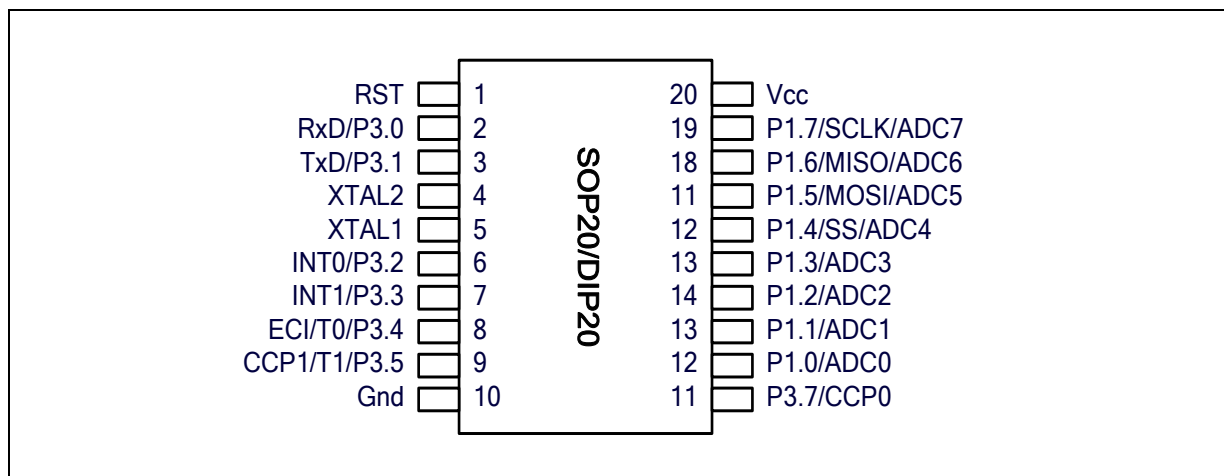
单片机型号	工作电压 (V)	Flash 程序存储器 10 万次 字节	内部大容量扩展 SRAM 字节	EEPROM 10 万次 字节	串口	SPI	定时器/计数器	PCA/CCP/PWM	8 路高速 ADC	看门狗 复位定时器	价格及封装					
											LQFP32	SOP32	SOP28	SKDIP28	SOP20	TSSOP20
12C5402AD	3.5-5.5	2K	512	有	1	有	6	4	10 位	有						
12C5404AD	3.5-5.5	4K	512	有	1	有	6	4	10 位	有						
12C5406AD	3.5-5.5	6K	512	有	1	有	6	4	10 位	有						
12C5408AD	3.5-5.5	8K	512	有	1	有	6	4	10 位	有						
12C5410AD	3.5-5.5	10K	512	有	1	有	6	4	10 位	有						
12C5412AD	3.5-5.5	12K	512	有	1	有	6	4	10 位	有						
12C5402	3.5-5.5	2K	512	有	1	有	6	4		有						
12C5404	3.5-5.5	4K	512	有	1	有	6	4		有						
12C5406	3.5-5.5	6K	512	有	1	有	6	4		有						
12C5408	3.5-5.5	8K	512	有	1	有	6	4		有						
12C5410	3.5-5.5	10K	512	有	1	有	6	4		有						
12C5412	3.5-5.5	12K	512	有	1	有	6	4		有						
12LE5402AD	2.2-3.8	2K	512	有	1	有	6	4	10 位	有						
12LE5404AD	2.2-3.8	4K	512	有	1	有	6	4	10 位	有						
12LE5406AD	2.2-3.8	6K	512	有	1	有	6	4	10 位	有						
12LE5408AD	2.2-3.8	8K	512	有	1	有	6	4	10 位	有						
12LE5410AD	2.2-3.8	10K	512	有	1	有	6	4	10 位	有						
12LE5412AD	2.2-3.8	12K	512	有	1	有	6	4	10 位	有						
12LE5402	2.2-3.8	2K	512	有	1	有	6	4		有						
12LE5404	2.2-3.8	4K	512	有	1	有	6	4		有						
12LE5406	2.2-3.8	6K	512	有	1	有	6	4		有						
12LE5408	2.2-3.8	8K	512	有	1	有	6	4		有						
12LE5410	2.2-3.8	10K	512	有	1	有	6	4		有						
12LE5412	2.2-3.8	12K	512	有	1	有	6	4		有						

1. 增强型 8051 CPU， 1T， 单时钟 / 机器周期， 指令代码完全兼容传统 8051
2. 工作电压： 5.5V - 3.5V (5V 单片机) / 3.6V - 2.2V (3V 单片机)
3. 工作频率范围： 0 - 35 MHz， 相当于普通 8051 的 0~ 420MHz
4. 用户应用程序空间 30K / 28K / 24K / 20K / 16K / 12K / 8K / 4K 字节
5. 片上集成 768 字节 RAM
6. 通用 I/O 口 (27/23/15 个)， 复位后为： 准双向口 / 弱上拉 (普通 8051 传统 I/O 口)

- 可设置成四种模式：准双向口 / 弱上拉，推挽 / 强上拉，仅为输入 / 高阻，开漏
每个 I/O 口驱动能力均可达到 20mA，但整个芯片最大不得超过 55mA
7. ISP（在系统可编程）/ IAP（在应用可编程），无需专用编程器，无需专用仿真器可通过串口（P3.0/P3.1）直接下载用户程序，数秒即可完成一片
 8. EEPROM 功能
 9. 看门狗
 10. 内部集成 MAX810 专用复位电路（外部晶体 20M 以下时，可省外部复位电路）
 11. 时钟源：外部高精度晶体 / 时钟，内部 R/C 振荡器
用户在下载用户程序时，可选择是使用内部 R/C 振荡器还是外部晶体 / 时钟
常温下内部 R/C 振荡器频率为：5.2MHz ~ 6.8MHz
精度要求不高时，可选择使用内部时钟，但因为有制造误差和温漂，应认为是 4MHz ~ 8MHz
 12. 共 6 个 16 位定时器 / 计数器，
两个专用 16 位定时器 T0 和 T1
PCA 模块可实现 4 个 16 位定时器
2 个时钟输出口，可由 T0 的溢出在 P1.0 输出时钟，可由 T1 的溢出在 P1.1 输出时钟
 13. 外部中断 9 路，下降沿中断或低电平触发中断，PCA 模式可分别或同时支持上升沿中断 / 下降沿中断，Power Down 模式可由外部中断唤醒
INT0/P3.2，INT1/P3.3，P3.4/T0，P3.5/T1，P3.0/RxD，PCA0/P3.7，PCA1/P3.5，PCA2/P2.0，PCA3/P2.4
 14. PWM(4 路) / PCA（可编程计数器阵列，4 路）
--- 也可用来当 4 路 D/A 使用
--- 也可用来再实现 4 个定时器
--- 也可用来再实现 4 个外部中断(上升沿中断 / 下降沿中断均可分别或同时支持)
 15. A/D 转换，10 位精度 ADC，共 8 路
 16. 通用全双工异步串行口(UART)，由于 STC12 系列是高速的 8051，也可再用定时器软件实现多串口
 17. SPI 同步通信口，主模式 / 从模式
 18. 工作温度范围：0 - 75℃ / -40 - +85℃
 19. 封装：LQFP-32，SOP-32/28/20，SKDIP-28，PDIP-20，TSSOP-20(超小封装 6.4mm × 6.4mm，定货)，LQFP/SOP32 有 27 个 I/O 口，SOP28/SKDIP28 有 23 个 I/O 口，SOP20/TSSOP20/PDIP20 有 15 个 I/O 口，
I/O 口不够时，可用 2 到 3 根普通 I/O 口线外接 74HC164/165/595（均可级联）来扩展 I/O 口，还可利用 A/D 做按键扫描来节省 I/O 口，或用双 CPU，三线通信，还多了串口。







2.3 STC12C2052AD 系列

单片机型号	工作电压 (V)	Flash 程序存储器 10 万次 字节	内部大容量扩展 SRAM 字节	EEPROM 10 万次 字节	串口	SPI	定时器/计数器	PCA/CCP/PWM	8 路高速 ADC	看门狗 复位定时器	价格及封装					
											LQFP32	SOP32	SOP28	SKDIP28	SOP20	TSSOP20
12C1052AD	3.5-5.5	1K	256	有	1	有	4	2	8 位	有						
12C2052AD	3.5-5.5	2K	256	有	1	有	4	2	8 位	有						
12C4052AD	3.5-5.5	4K	256	有	1	有	4	2	8 位	有						
12C5052AD	3.5-5.5	5K	256	有	1	有	4	2	8 位	有						
12C1052	3.5-5.5	1K	256	有	1	有	4	2		有						
12C2052	3.5-5.5	2K	256	有	1	有	4	2		有						
12C4052	3.5-5.5	4K	256	有	1	有	4	2		有						
12C5052	3.5-5.5	5K	256	有	1	有	4	2		有						
12LE1052AD	2.2-3.8	1K	256	有	1	有	4	2	8 位	有						
12LE2052AD	2.2-3.8	2K	256	有	1	有	4	2	8 位	有						
12LE4052AD	2.2-3.8	4K	256	有	1	有	4	2	8 位	有						
12LE5052AD	2.2-3.8	5K	256	有	1	有	4	2	8 位	有						
12LE1052	2.2-3.8	1K	256	有	1	有	4	2		有						
12LE2052	2.2-3.8	2K	256	有	1	有	4	2		有						
12LE4052	2.2-3.8	4K	256	有	1	有	4	2		有						
12LE5052	2.2-3.8	5K	256	有	1	有	4	2		有						

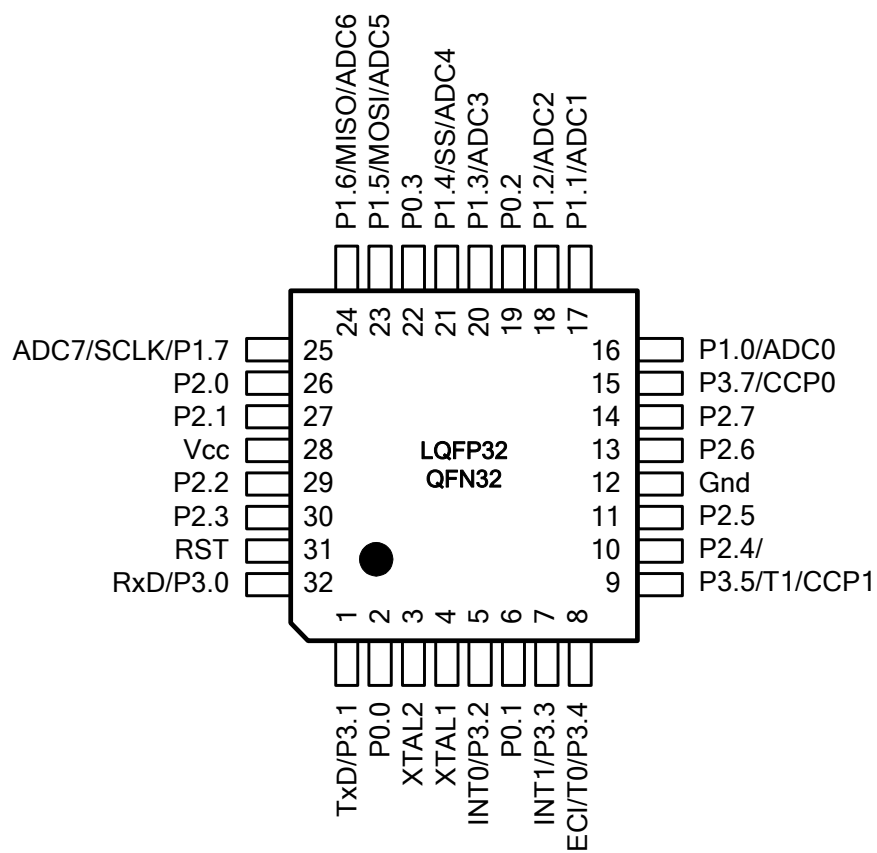
1. 增强型 8051 CPU， 1T， 单时钟 / 机器周期， 指令代码完全兼容传统 8051
2. 工作电压： 5.5V - 3.5V (5V 单片机) / 3.6V - 2.2V (3V 单片机)
3. 工作频率范围： 0 - 35 MHz， 相当于普通 8051 的 0 ~ 420MHz
4. 用户应用程序空间 5K / 4K / 2K / 1K 字节
5. 片上集成 256 字节 RAM
6. 通用 I/O 口 (27/23/15 个)， 复位后为： 准双向口 / 弱上拉 (普通 8051 传统 I/O 口)
可设置成四种模式： 准双向口 / 弱上拉， 推挽 / 强上拉， 仅为输入 / 高阻， 开漏
每个 I/O 口驱动能力均可达到 20mA， 但整个芯片最大不得超过 55mA
7. ISP (在系统可编程) / IAP (在应用可编程)， 无需专用编程器， 无需专用仿真器可通过串口 (P3.0/P3.1) 直接下载用户程序， 数秒即可完成一片
8. EEPROM 功能
9. 看门狗
10. 内部集成 MAX810 专用复位电路 (外部晶体 20M 以下时， 可省外部复位电路)
11. 时钟源： 外部高精度晶体 / 时钟， 内部 R/C 振荡器

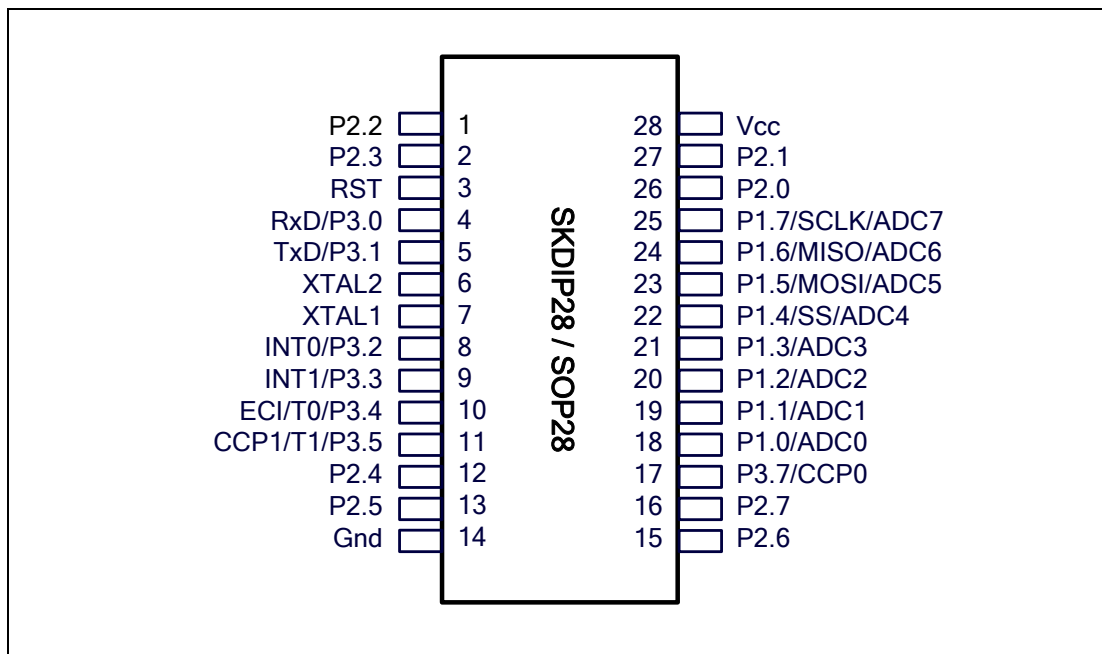
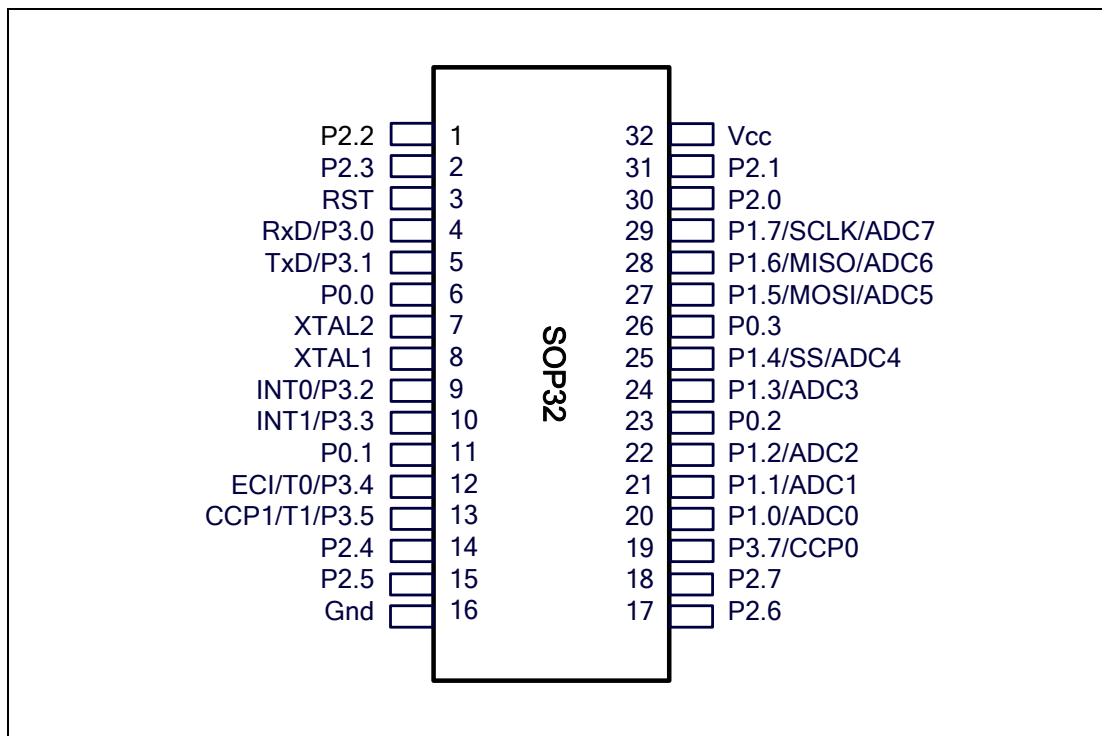
用户在下载用户程序时，可选择是使用内部 R/C 振荡器还是外部晶体 / 时钟

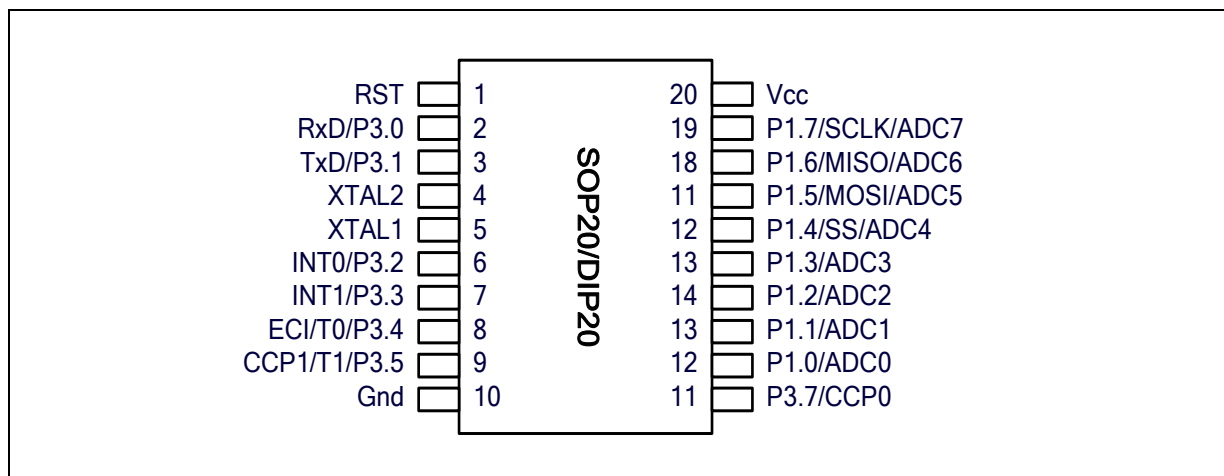
常温下内部 R/C 振荡器频率为：5.2MHz ~ 6.8MHz

精度要求不高时，可选择使用内部时钟，但因为有制造误差和温漂，应认为是 4MHz ~ 8MHz

12. 共 4 个 16 位定时器 / 计数器，
两个专用 16 位定时器 T0 和 T1
PCA 模块可实现 2 个 16 位定时器
2 个时钟输出口，可由 T0 的溢出在 P1.0 输出时钟，可由 T1 的溢出在 P1.1 输出时钟
13. 外部中断 9 路，下降沿中断或低电平触发中断，PCA 模式可分别或同时支持上升沿中断 / 下降沿中断，Power Down 模式可由外部中断唤醒
INT0/P3.2，INT1/P3.3，P3.4/T0，P3.5/T1，P3.0/RxD，PCA0/P3.7，PCA1/P3.5，PCA2/P2.0，PCA3/P2.4
14. PWM(4 路) /PCA (可编程计数器阵列，4 路)
--- 也可用来当 4 路 D/A 使用
--- 也可用来再实现 4 个定时器
--- 也可用来再实现 4 个外部中断(上升沿中断 / 下降沿中断均可分别或同时支持)
15. A/D 转换，10 位精度 ADC，共 8 路
16. 通用全双工异步串行口(UART)，由于 STC12 系列是高速的 8051，也可再用定时器软件实现多串口
17. SPI 同步通信口，主模式 / 从模式
18. 工作温度范围：0 - 75℃ / -40 - +85℃
19. 封装：LQFP-32, SOP-32/28/20, SKDIP-28, PDIP-20, TSSOP-20(超小封装 6.4mm × 6.4mm, 定货)，LQFP/SOP32 有 27 个 I/O 口，SOP28/SKDIP28 有 23 个 I/O 口，SOP20/TSSOP20/PDIP20 有 15 个 I/O 口，
I/O 口不够时，可用 2 到 3 根普通 I/O 口线外接 74HC164/165/595 (均可级联) 来扩展 I/O 口，还可利用 A/D 做按键扫描来节省 I/O 口，或用双 CPU，三线通信，还多了串口。

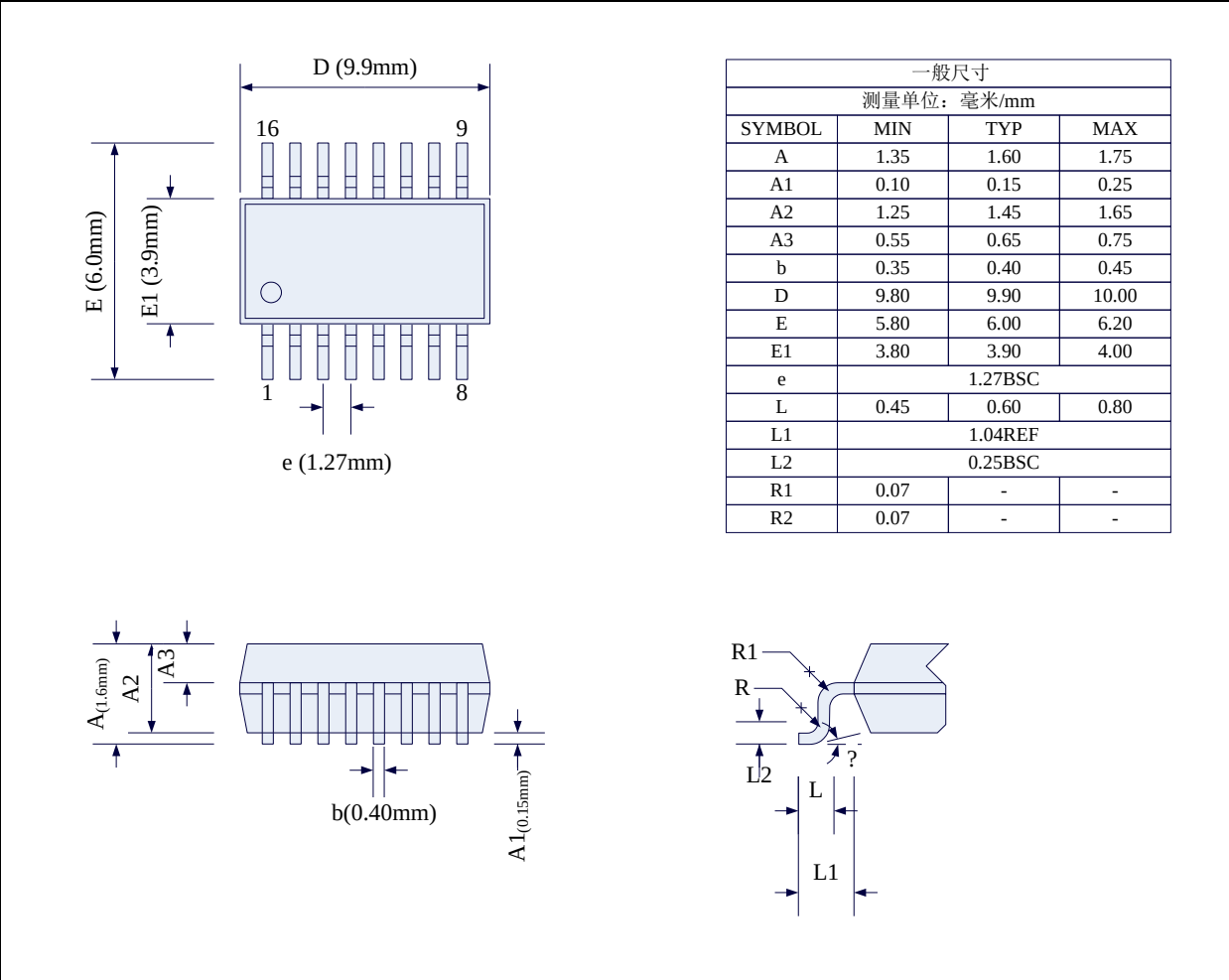




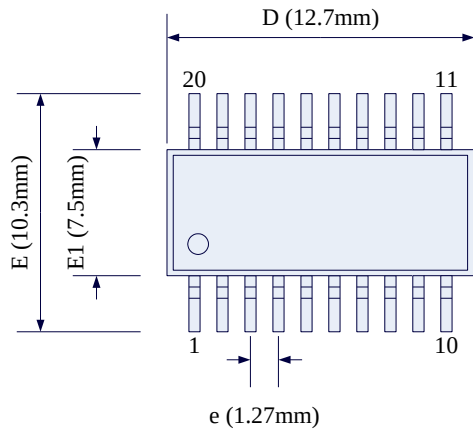


3 封装尺寸图

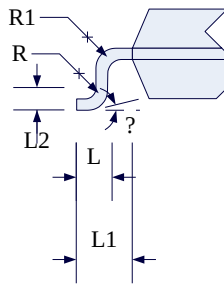
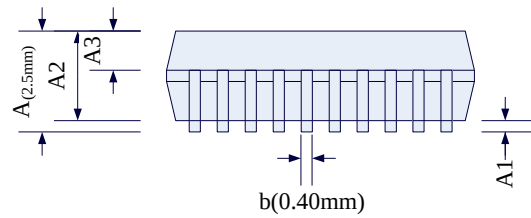
3.1 SOP16 封装尺寸图



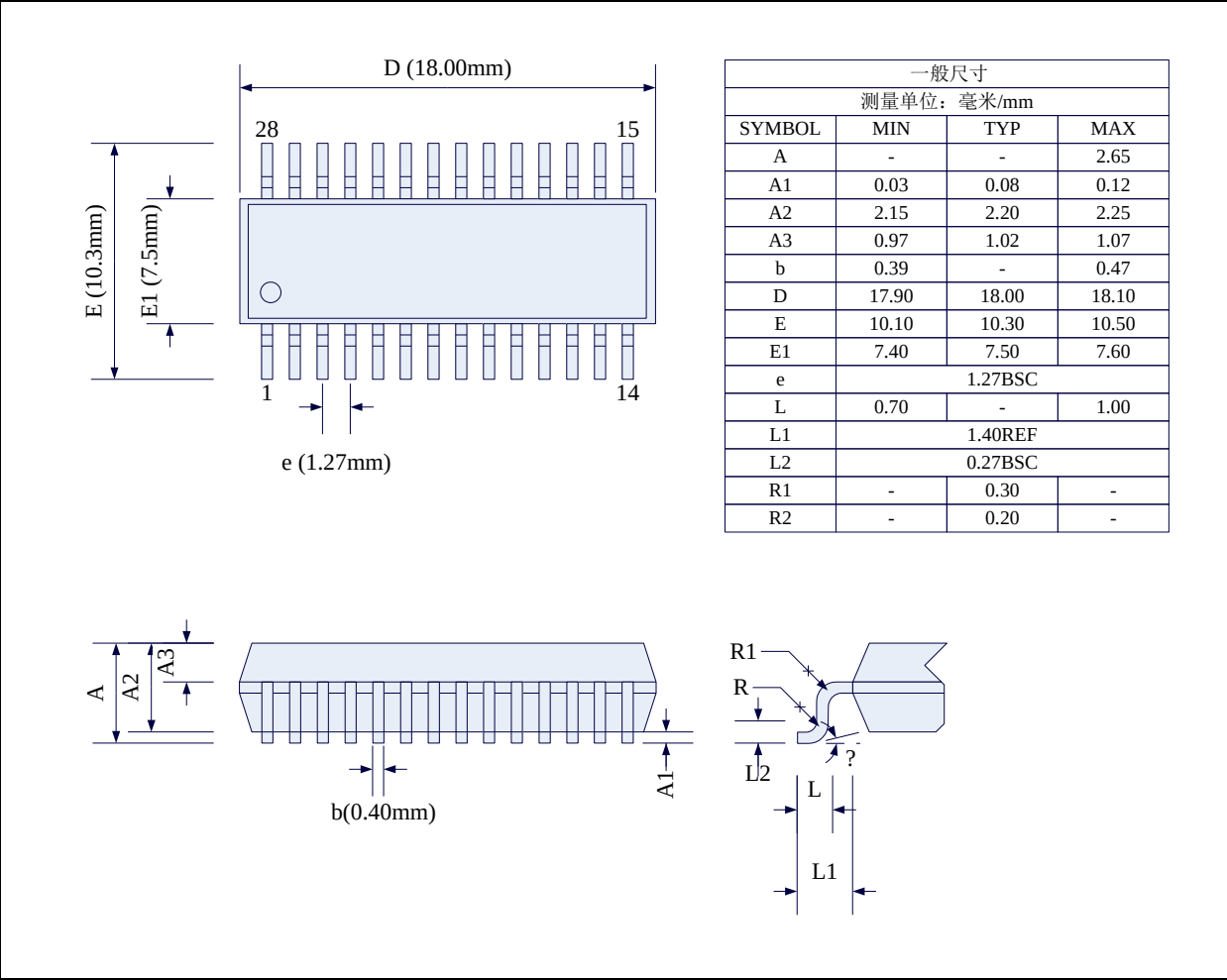
3.2 SOP20 封装尺寸图



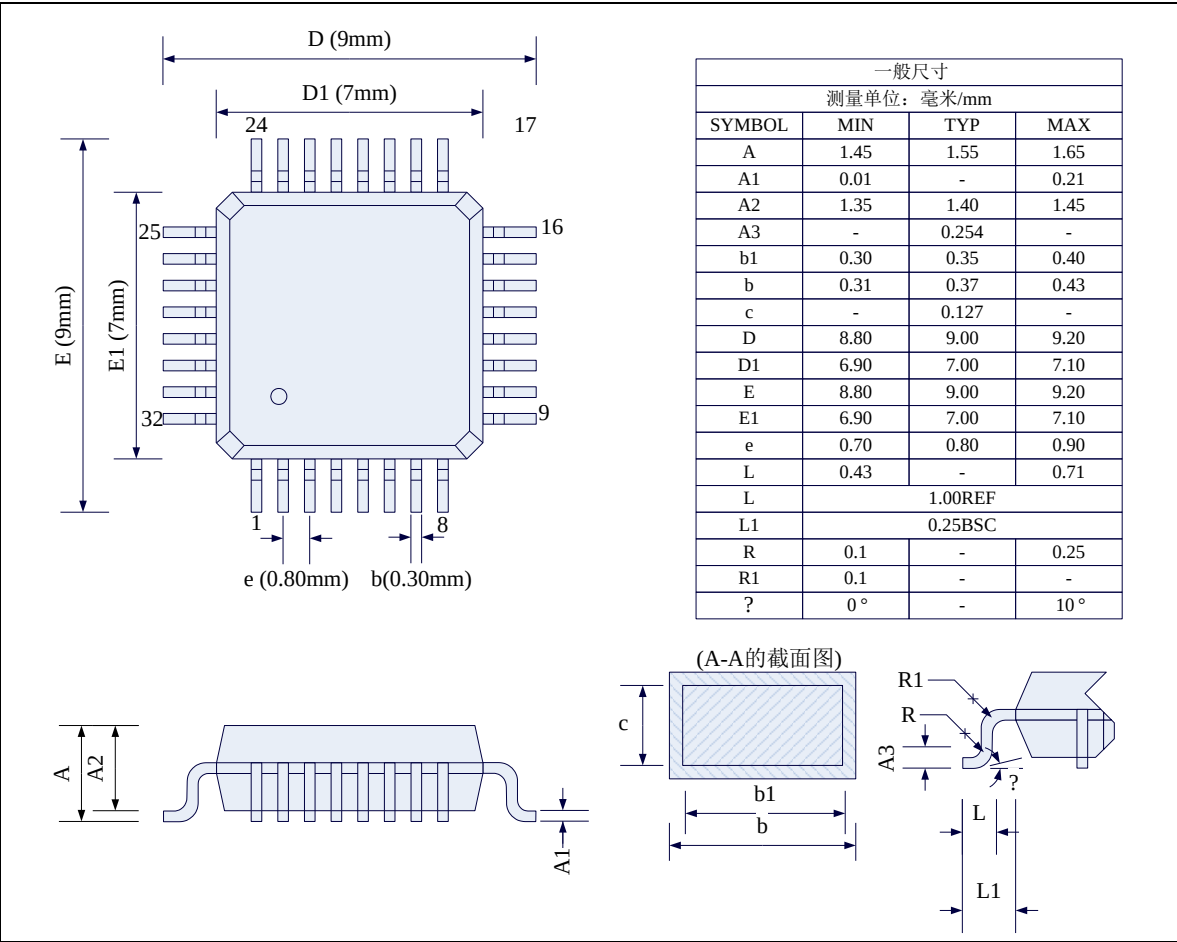
一般尺寸			
测量单位: 毫米/mm			
SYMBOL	MIN	TYP	MAX
A	2.465	2.515	2.565
A1	0.10	0.15	0.20
A2	2.10	2.30	2.50
A3	0.55	0.65	0.75
b	0.356	0.406	0.456
D	12.50	12.70	12.90
E	10.20	10.30	10.40
E1	7.45	7.50	7.55
e	1.27BSC		
L	0.80	0.86	0.90
L1	1.40REF		
L2	0.27BSC		
R1	-	0.30	-
R2	-	0.20	-



3.3 SOP28 封装尺寸图



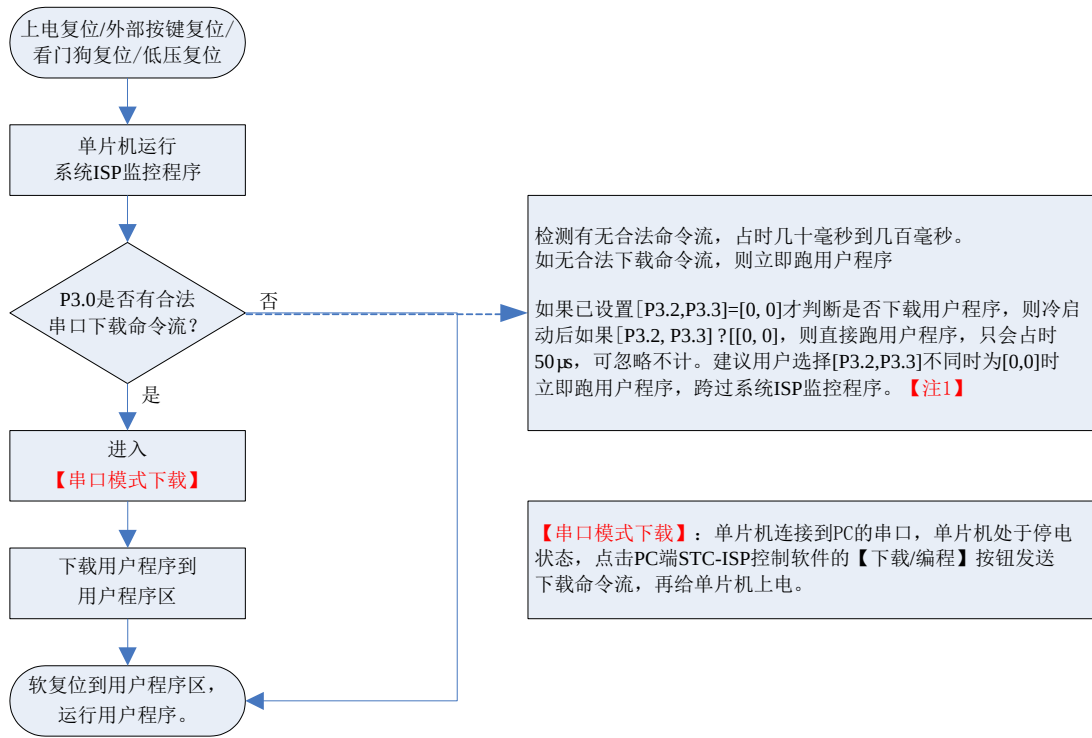
3.4 LQFP32 封装尺寸图 (9mm*9mm)



4 ISP 下载

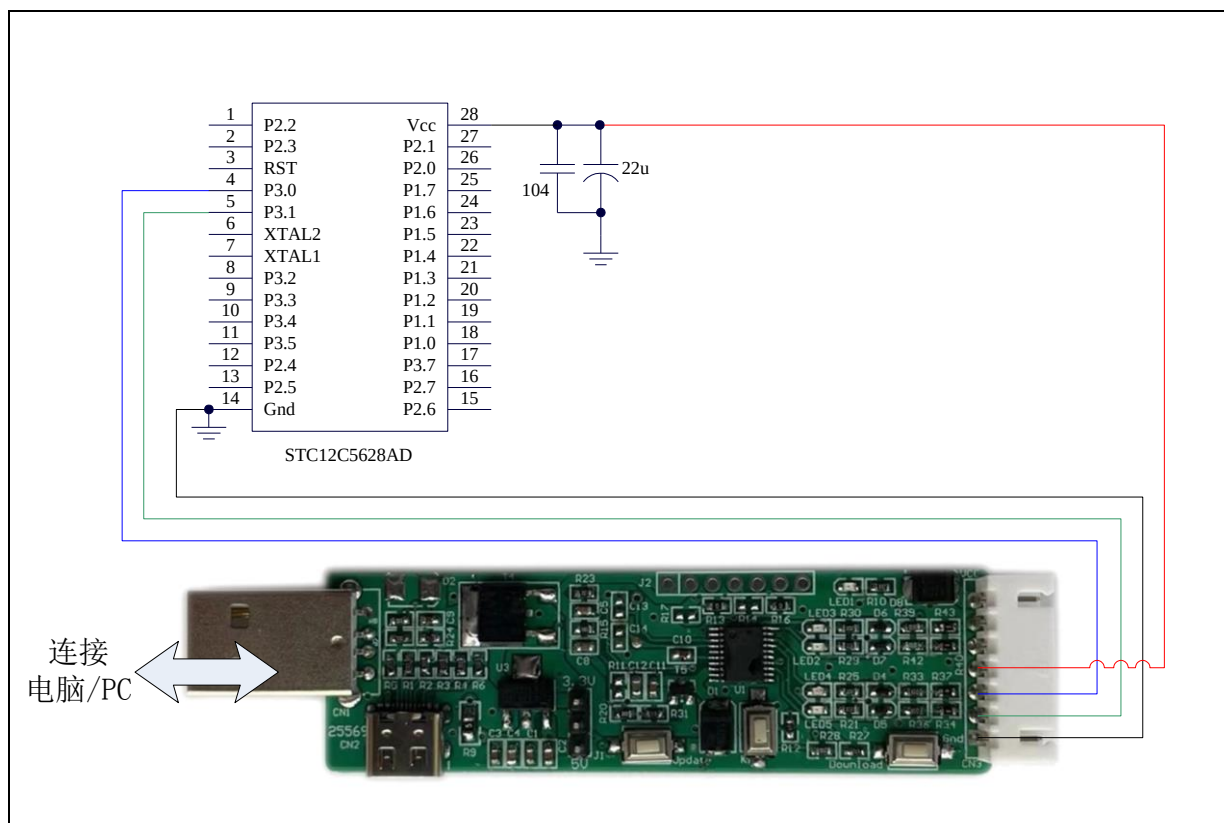
4.1 ISP 下载流程及典型应用线路图

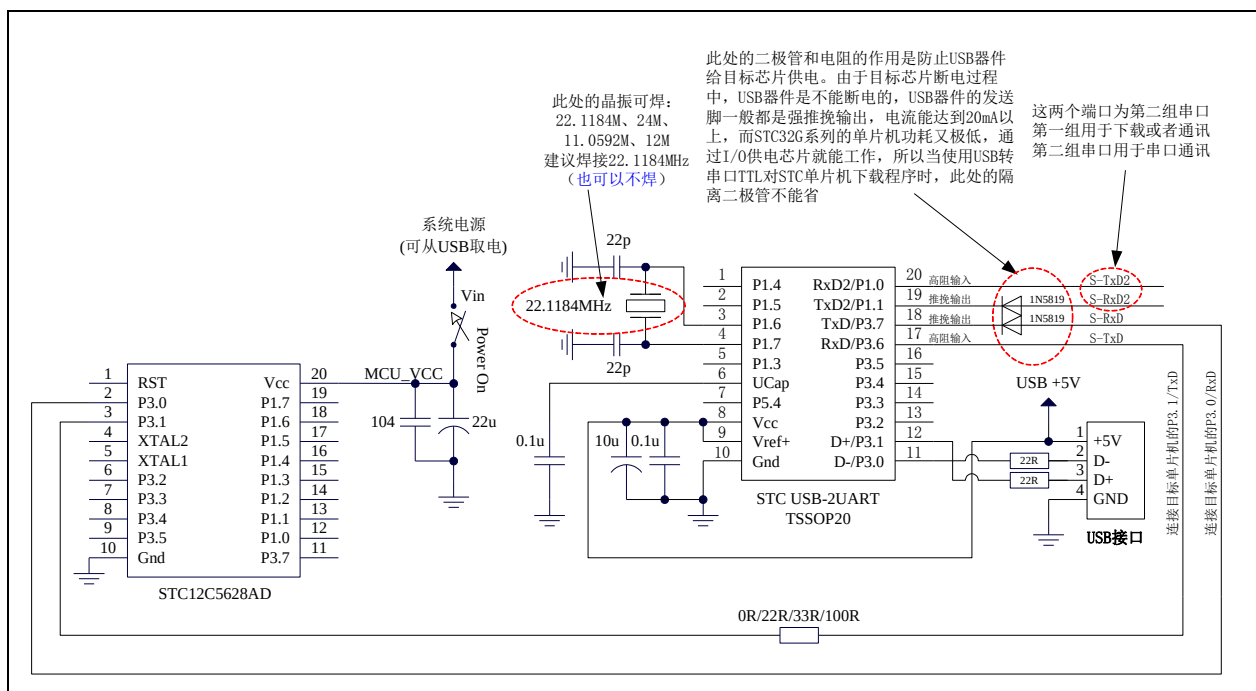
4.1.1 ISP 下载流程图



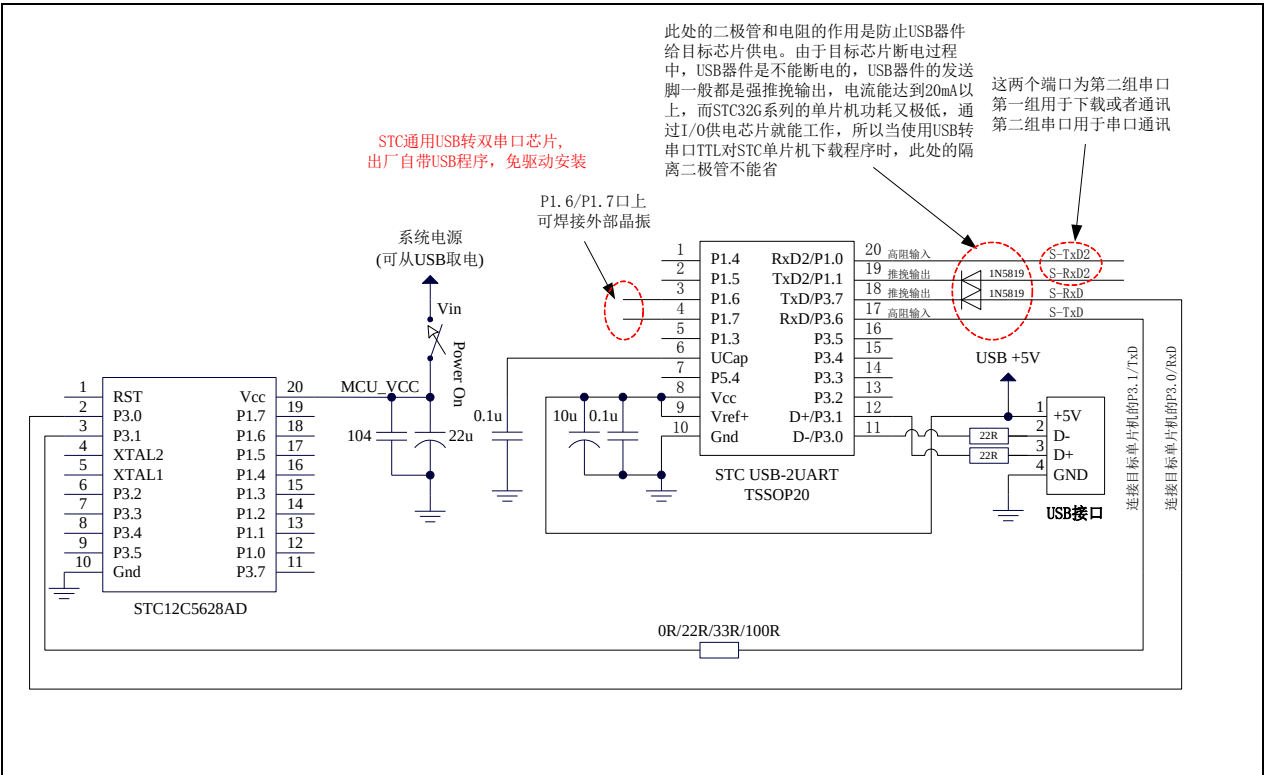
注意：因 [P3.0, P3.1] 作下载/仿真用(下载/仿真接口仅可用 [P3.0, P3.1])，故建议用户将串口 1 放在 P3.6/P3.7 或 P1.6/P1.7，若用户不想切换，坚持使用 P3.0/P3.1 工作或作为串口 1 进行通信，则务必在下载程序时，在软件上勾选“下次冷启动时，P3.2/P3.3 为 0/0 时才可以下载程序”。

4.1.2 使用一箭双雕之 USB 转串口工具下载



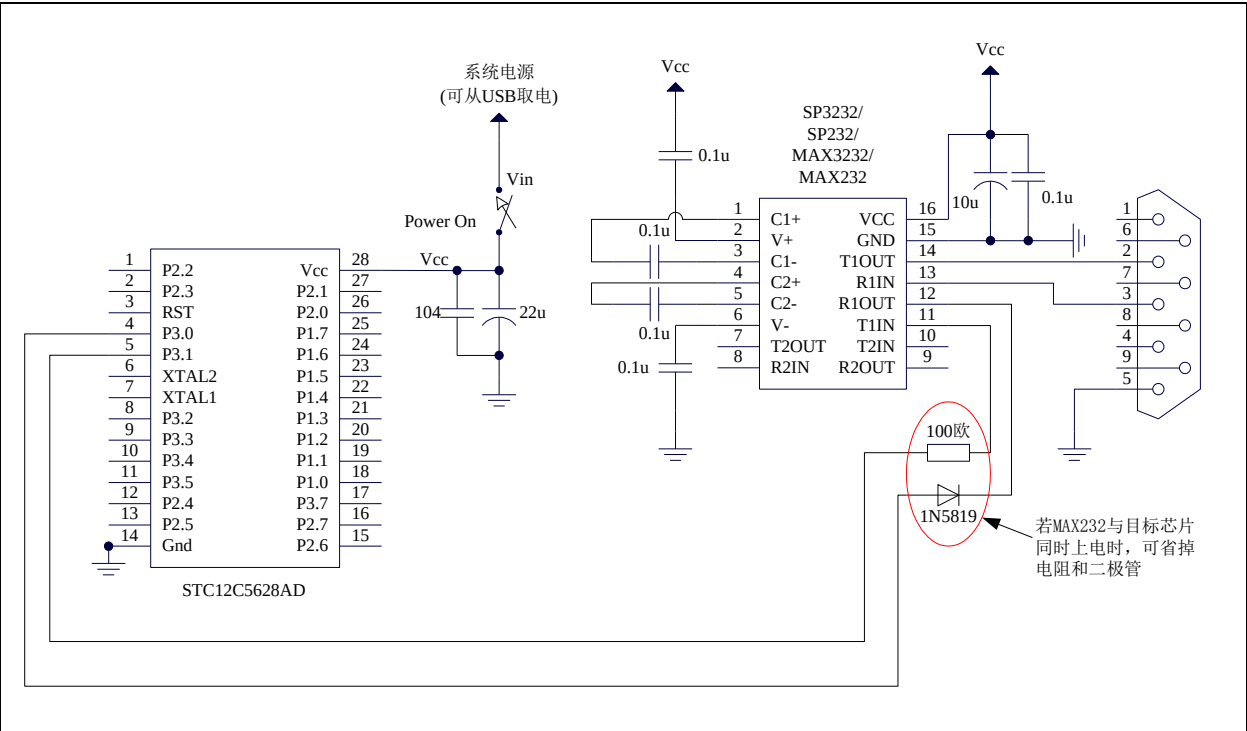


4.1.4 使用 USB 转双串口/TTL 下载（无外部晶振）

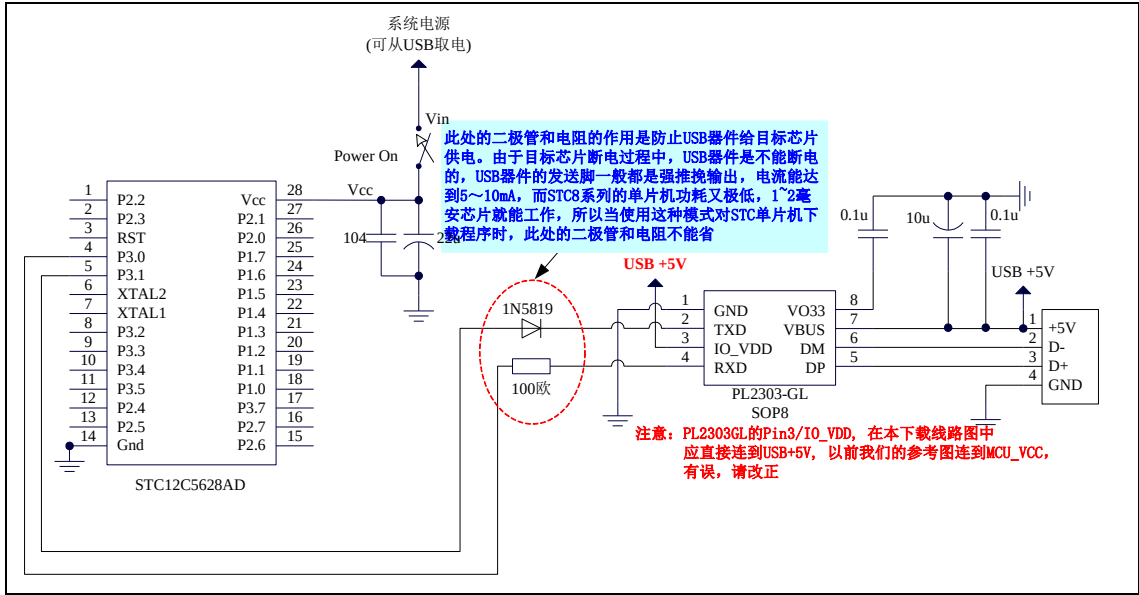




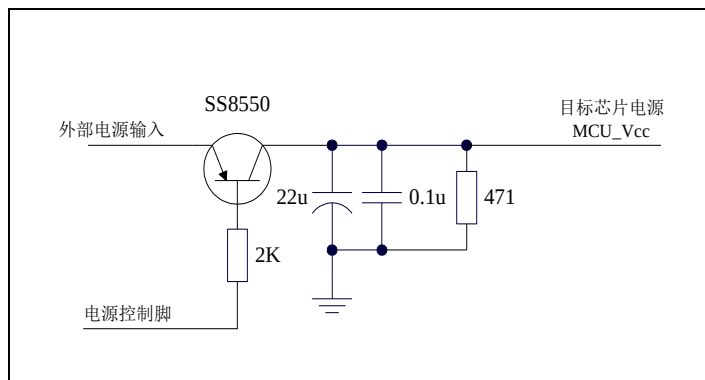
4.1.6 使用 RS-232 转换器下载，也可支持仿真



4.1.7 使用 PL2303-GL 下载，也可支持仿真



4.1.8 单机电源控制参考电路



5 复位、省电模式与系统电源管理

5.1 系统复位

单片机的复位分为硬件复位和软件复位两种。

硬件复位时，所有的寄存器的值会复位到初始值，系统会重新读取所有的硬件选项。同时根据硬件选项所设置的上电等待时间进行上电等待。硬件复位主要包括：

- 上电复位
- 低压复位
- 复位脚复位
- 看门狗复位

软件复位时，除与时钟相关的寄存器保持不变外，其余的所有寄存器的值会复位到初始值，软件复位不会重新读取所有的硬件选项。软件复位主要包括：

- 写 IAP_CONTR 的 SWRST 所触发的复位

相关寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
WDT_CONTR	看门狗控制寄存器	C1H	WDT_FLAG	-	EN_WDT	CLR_WDT	IDL_WDT	WDT_PS[2:0]			0x00,0000
IAP_CONTR	IAP 控制寄存器	C7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT[2:0]-			0000,x000

5.1.1 看门狗复位（WDT_CONTR）

在工业控制/汽车电子/航空航天等需要高可靠性的系统中, 为了防止“系统在异常情况下, 受到干扰, MCU/CPU 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/CPU 不在规定的时间内按要求访问看门狗, 就认为 MCU/CPU 处于异常状态, 看门狗就会强制 MCU/CPU 复位, 使系统重新从头开始执行用户程序。

看门狗复位是热启动复位中的硬件复位之一。看门狗复位状态结束后, 系统固定从 ISP 监控程序区启动, 与看门狗复位前 IAP_CONTR 寄存器的 SWBS 无关

WDT_CONTR（看门狗控制寄存器）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
WDT_CONTR	C1H	WDT_FLAG	-	EN_WDT	CLR_WDT	IDL_WDT	WDT_PS[2:0]		

WDT_FLAG: 看门狗溢出标志

看门狗发生溢出时, 硬件自动将此位置 1, 需要软件清零。

EN_WDT: 看门狗使能位

0: 对单片机无影响

1: 启动看门狗定时器。

CLR_WDT: 看门狗定时器清零

0: 对单片机无影响

1: 清零看门狗定时器, 硬件自动将此位复位

IDL_WDT: IDLE 模式时的看门狗控制位

0: IDLE 模式时看门狗停止计数

1: IDLE 模式时看门狗继续计数

WDT_PS[2:0]: 看门狗定时器时钟分频系数

WDT_PS[2:0]	分频系数	12M 主频时的溢出时间	20M 主频时的溢出时间
000	2	≈ 65.5 毫秒	≈ 39.3 毫秒
001	4	≈ 131 毫秒	≈ 78.6 毫秒
010	8	≈ 262 毫秒	≈ 157 毫秒
011	16	≈ 524 毫秒	≈ 315 毫秒
100	32	≈ 1.05 秒	≈ 629 毫秒
101	64	≈ 2.10 秒	≈ 1.26 秒
110	128	≈ 4.20 秒	≈ 2.52 秒
111	256	≈ 8.39 秒	≈ 5.03 秒

看门狗溢出时间计算公式如下:

$$\text{看门狗溢出时间} = \frac{12 \times 32768 \times 2^{(\text{WDT_PS}+1)}}{\text{SYSclk}}$$

5.1.2 软件复位（IAP_CONTR）

IAP_CONTR（IAP 控制寄存器）

对 IAP 控制寄存器写 60H，可达到对单片机冷启动的效果

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_CONTR	C7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT[2:0]		

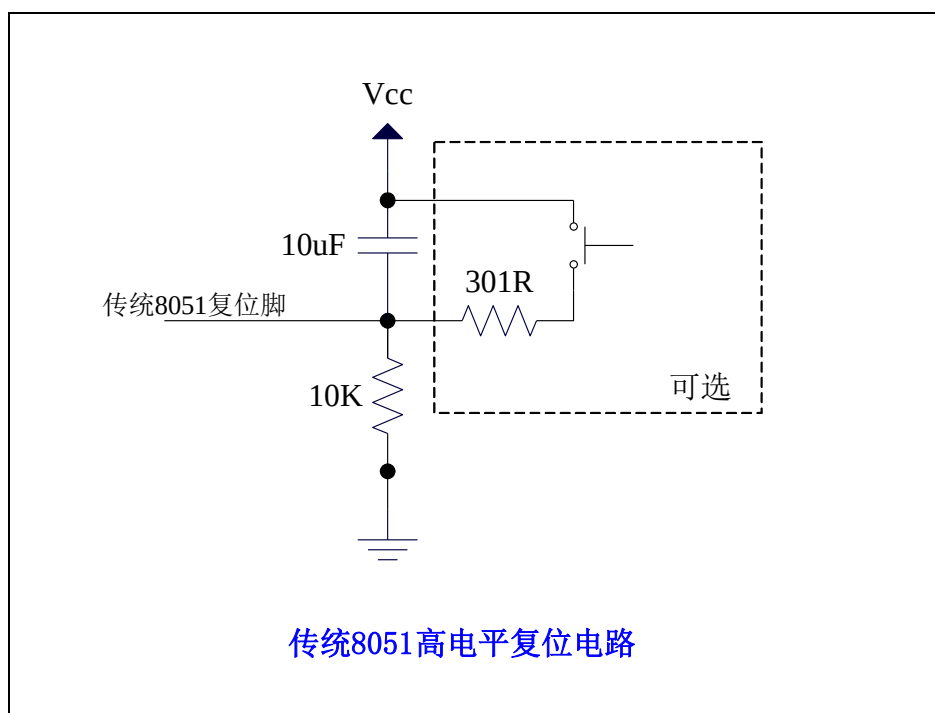
SWBS：软件复位启动选择

- 0：软件复位后从用户程序区开始执行代码。用户数据区的数据保持不变。
- 1：软件复位后从系统 ISP 区开始执行代码。用户数据区的数据会被初始化。

SWRST：软件复位触发位

- 0：对单片机无影响
- 1：触发软件复位

5.1.3 高电平上电复位参考电路



5.2 时钟停振/省电模式与系统电源管理

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
PCON	电源控制寄存器	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000

5.2.1 电源控制寄存器（PCON）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL

LVDF：低压检测标志位。当系统检测到低压事件时，硬件自动将此位置 1，并向 CPU 提出中断请求。此位需要用户软件清零。

POF：上电复位标志位。MCU 每次重新上电后，硬件自动将此位置 1，可软件将此位清零。

PD：时钟停振模式/掉电模式/停电模式控制位

0：无影响

1：单片机进入时钟停振模式/掉电模式/停电模式，CPU 以及全部外设均停止工作。唤醒后硬件自动清零。

IDL：IDLE（空闲）模式控制位

0：无影响

1：单片机进入 IDLE 模式，只有 CPU 停止工作，其他外设依然在运行。唤醒后硬件自动清零

6 存储器

程序存储器和数据存储器是各自独立编址的。由于没有提供访问外部程序存储器的总线，单片机的所有程序存储器都是片上 Flash 存储器，不能访问外部程序存储器。

单片机内部集成了大容量的数据存储器。单片机内部的数据存储器在物理和逻辑上都分为两个地址空间:内部 RAM(256 字节)和内部扩展 RAM。其中内部 RAM 的高 128 字节的数据存储器与特殊功能寄存器(SFRs)地址重叠，实际使用时通过不同的寻址方式加以区分。

6.1 程序存储器

程序存储器用于存放用户程序、数据以及表格等信息。

单片机复位后，程序计数器(PC)的内容为 0000H，从 0000H 单元开始执行程序。另外中断服务程序的入口地址(又称中断向量)也位于程序存储器单元。在程序存储器中，每个中断都有一个固定的入口地址，当中断发生并得到响应后，单片机就会自动跳转到相应的中断入口地址去执行程序。外部中断 0 (INT0) 的中断服务程序的入口地址是 0003H，定时器/计数器 0 (TIMER0) 中断服务程序的入口地址是 000BH，外部中断 1 (INT1) 的中断服务程序的入口地址是 0013H，定时器/计数器 1 (TIMER1) 的中断服务程序的入口地址是 001BH 等。更多的中断服务程序的入口地址(中断向量)请参考中断介绍章节。

由于相邻中断入口地址的间隔区间仅仅有 8 个字节，一般情况下无法保存完整的中断服务程序，因此在中断响应的地址区域存放一条无条件转移指令，指向真正存放中断服务程序的空间去执行。

单片机中都包含有 Flash 数据存储器 (EEPROM)。以字节为单位进行读/写数据，以 512 字节为页单位进行擦除，可在线反复编程擦写 10 万次以上，提高了使用的灵活性和方便性。

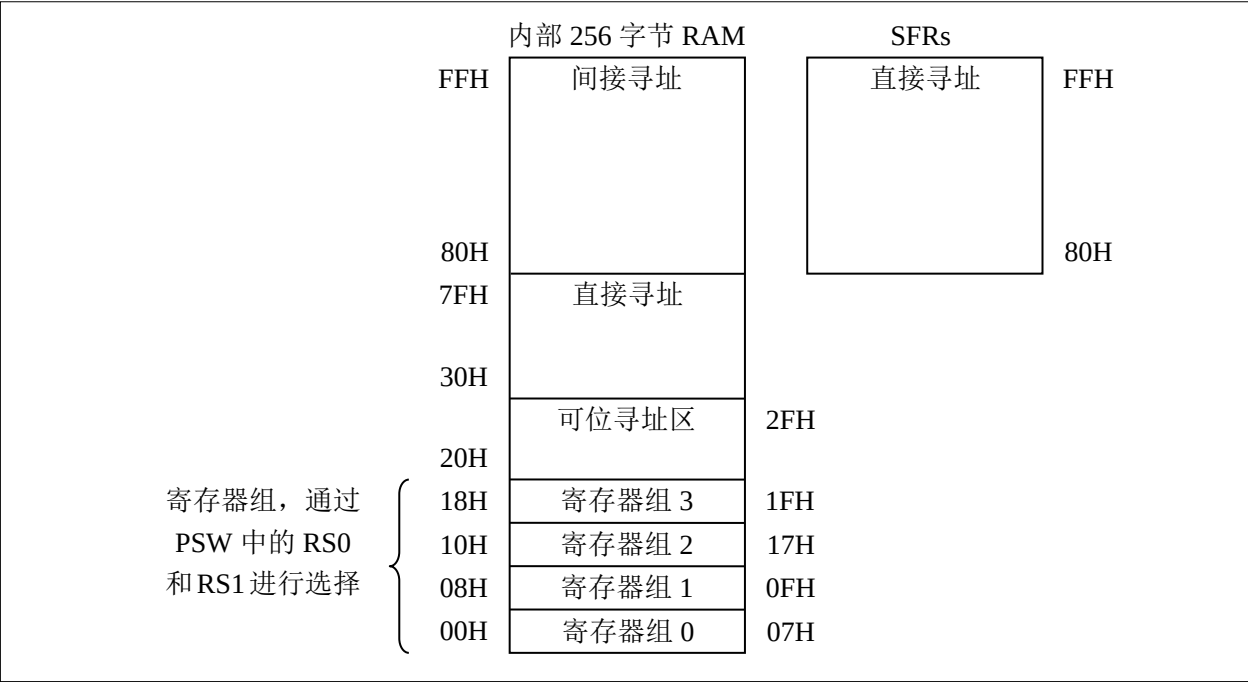
6.2 数据存储器

单片机内部集成的 RAM 可用于存放程序执行的中间结果和过程数据。

6.2.1 内部 RAM

内部 RAM 共 256 字节，可分为 2 个部分：低 128 字节 RAM 和高 128 字节 RAM。低 128 字节的数据存储器与传统 8051 兼容，既可直接寻址也可间接寻址。高 128 字节 RAM（在 8052 中扩展了高 128 字节 RAM）与特殊功能寄存器区共用相同的逻辑地址，都使用 80H~FFH，但在物理上是分别独立的，使用时通过不同的寻址方式加以区分。高 128 字节 RAM 只能间接寻址，特殊功能寄存器区只可直接寻址。

内部 RAM 的结构如下图所示：



低 128 字节 RAM 也称通用 RAM 区。通用 RAM 区又可分为工作寄存器组区，可位寻址区，用户 RAM 区和堆栈区。工作寄存器组区地址从 00H~1FH 共 32 字节单元，分为 4 组，每一组称为一个寄存器组，每组包含 8 个 8 位的工作寄存器，编号均为 R0~R7，但属于不同的物理空间。通过使用工作寄存器组，可以提高运算速度。R0~R7 是常用的寄存器，提供 4 组是因为 1 组往往不够用。程序状态字 PSW 寄存器中的 RS1 和 RS0 组合决定当前使用的工作寄存器组，见下面 PSW 寄存器的介绍。

6.2.2 程序状态寄存器（PSW）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
PSW	D0H	CY	AC	F0	RS1	RS0	OV	F1	P

CY: 进/借位标志位。

AC: 辅组进/借位标志位。

F0: 用户标志位 0。

RS1, RS0: 工作寄存器选择位

RS1	RS0	工作寄存器组（R0~R7）
0	0	第 0 组（00H~07H）
0	1	第 1 组（08H~0FH）
1	0	第 2 组（10H~17H）
1	1	第 3 组（18H~1FH）

OV: 溢出标志位。

F1: 用户标志位 1。

P: 奇偶校验标志位。

可位寻址区的地址从 20H~2FH 共 16 个字节单元。20H~2FH 单元既可像普通 RAM 单元一样按字节存取，也可以对单元中的任何一位单独存取，共 128 位，所对应的逻辑位地址范围是 00H~7FH。位地址范围是 00H~7FH，内部 RAM 低 128 字节的地址也是 00H~7FH，从外表看，二者地址是一样的，实际上二者具有本质的区别：位地址指向的是一个位，而字节地址指向的是一个字节单元，在程序中使用不同的指令区分。

内部 RAM 中的 30H~FFH 单元是用户 RAM 和堆栈区。一个 8 位的堆栈指针(SP)，用于指向堆栈区。单片机复位后，堆栈指针 SP 为 07H，指向了工作寄存器组 0 中的 R7，因此，用户初始化程序都应对 SP 设置初值，一般设置在 80H 以后的单元为宜。

6.2.3 堆栈指针（SP）

堆栈指针是一个 8 位专用寄存器。它指示出堆栈顶部在内部 RAM 块中的位置。系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 单元开始，考虑 08H~1FH 单元分别属于工作寄存器组 1~3，若在程序设计中用到这些区，则最好把 SP 值改变为 80H 或更大的值为宜。单片机的堆栈是向上生长的，即将数据压入堆栈后，SP 内容增大。

6.2.4 内部扩展 RAM，XRAM，XDATA

单片机片内除了集成 256 字节的内部 RAM 外，还集成了内部的扩展 RAM。访问内部扩展 RAM 的方法与传统 8051 单片机访问外部扩展 RAM 的方法相同，但是不影响 P0 口(数据总线和高八位地址总线)、P2 口(低八位地址总线)、以及 RD、WR 和 ALE 等端口上的信号。

在汇编语言中，内部扩展 RAM 通过 MOVX 指令访问，

```

MOVX    A,@DPTR
MOVX    @DPTR,A
MOVX    A,@Ri
MOVX    @Ri,A

```

在 C 语言中，可使用 xdata 声明存储类型即可。如：

unsigned char xdata i;

注：pdata 即为 xdata 的低 256 字节，用 MOVX @Ri,A 和 MOVX A,@Ri 进行访问。但读写 pdata 类型的变量比 xdata 类型要慢，所有建议用户代码中均统一使用 xdata 在扩展 RAM 中声明变量。

单片机内部扩展 RAM 是否可以访问，受辅助寄存器 AUXR 中的 EXTRAM 位控制。

6.2.5 辅助寄存器（AUXR）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
AUXR	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

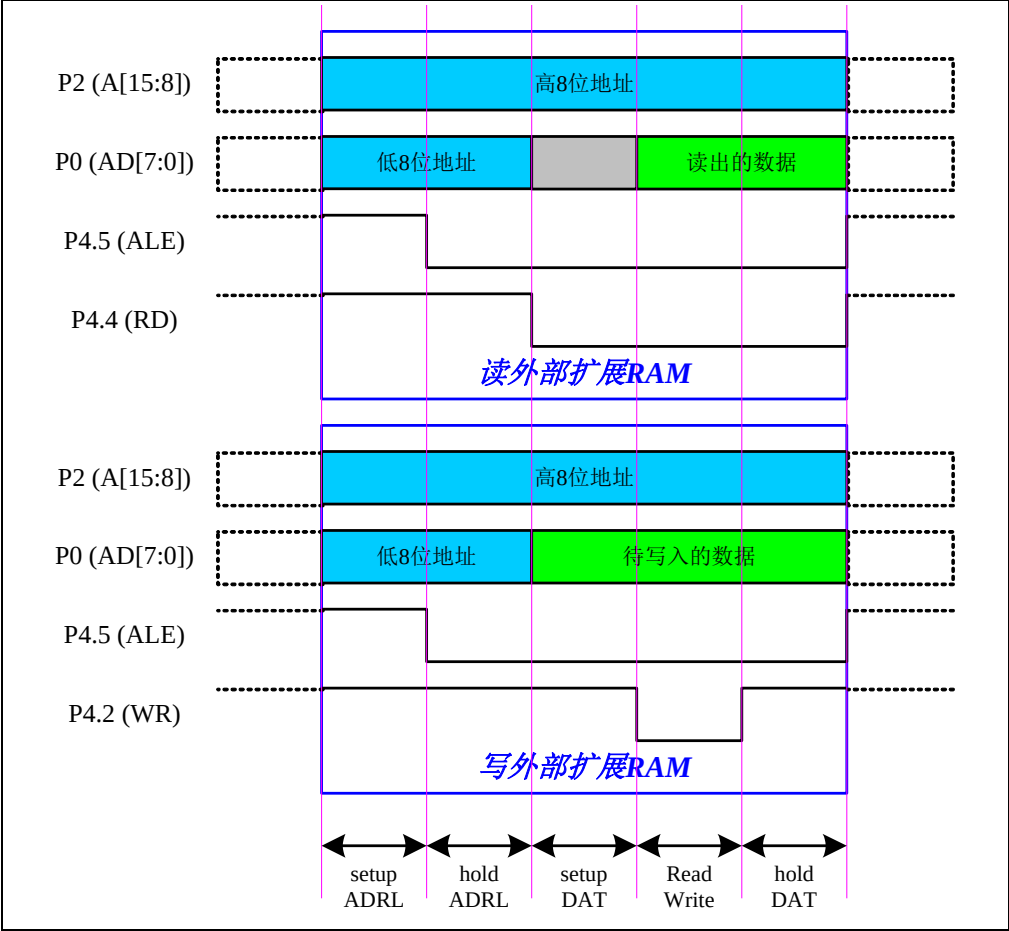
EXTRAM: 扩展 RAM 访问控制

0: 访问内部扩展 RAM。

1: 内部扩展 RAM 被禁用。

6.2.6 外部扩展 RAM，XRAM，XDATA

单片机具有扩展 64KB 外部数据存储器的能力。访问外部数据存储器期间，WR/RD/ALE 信号要有效。S 读写外部扩展 RAM 时序如下图所示：



6.2.7 8051 中可位寻址的数据存储器

8051 单片机内部可位寻址的数据存储器包括两部分：第一部分的地址范围为 00H~7FH，第二部分的地址范围是 80H~FFH。00H~7FH 的位寻址区域是数据区 20H~2FH 这 16 个字节的映射；而 80H~FFH 的位寻址区域则是所有的特殊功能寄存器中地址能被 8 整除的 16 个特殊功能寄存器（包括 80H、88H、90H、98H、A0H、A8H、B0H、B8H、C0H、C8H、D0H、D8H、E0H、E8H、F0H、F8H）的映射。

数据存储器地址	位寻址地址							
	B7	B6	B5	B4	B3	B2	B1	B0
F8H (P7)	FFH F8H. 7	FEH F8H. 6	FDH F8H. 5	FCH F8H. 4	FBH F8H. 3	FAH F8H. 2	F9H F8H. 1	F8H F8H. 0
F0H (B)	F7H F0H. 7	F6H F0H. 6	F5H F0H. 5	F4H F0H. 4	F3H F0H. 3	F2H F0H. 2	F1H F0H. 1	F0H F0H. 0
E8H (P6)	EFH E8H. 7	EEH E8H. 6	EDH E8H. 5	ECH E8H. 4	EBH E8H. 3	EAH E8H. 2	E9H E8H. 1	E8H E8H. 0
E0H (ACC)	E7H E0H. 7	E6H E0H. 6	E5H E0H. 5	E4H E0H. 4	E3H E0H. 3	E2H E0H. 2	E1H E0H. 1	E0H E0H. 0
D8H (CCON)	DFH D8H. 7	DEH D8H. 6	DDH D8H. 5	DCH D8H. 4	DBH D8H. 3	DAH D8H. 2	D9H D8H. 1	D8H D8H. 0
D0H (PSW)	D7H D0H. 7	D6H D0H. 6	D5H D0H. 5	D4H D0H. 4	D3H D0H. 3	D2H D0H. 2	D1H D0H. 1	D0H D0H. 0
C8H (P5)	CFH C8H. 7	CEH C8H. 6	CDH C8H. 5	CCH C8H. 4	CBH C8H. 3	CAH C8H. 2	C9H C8H. 1	C8H C8H. 0
C0H (P4)	C7H C0H. 7	C6H C0H. 6	C5H C0H. 5	C4H C0H. 4	C3H C0H. 3	C2H C0H. 2	C1H C0H. 1	C0H C0H. 0
B8H (IP)	BFH B8H. 7	BEH B8H. 6	BDH B8H. 5	BCH B8H. 4	BBH B8H. 3	BAH B8H. 2	B9H B8H. 1	B8H B8H. 0
B0H (P3)	B7H B0H. 7	B6H B0H. 6	B5H B0H. 5	B4H B0H. 4	B3H B0H. 3	B2H B0H. 2	B1H B0H. 1	B0H B0H. 0
A8H (IE)	AFH A8H. 7	AEH A8H. 6	ADH A8H. 5	ACH A8H. 4	ABH A8H. 3	AAH A8H. 2	A9H A8H. 1	A8H A8H. 0
A0H (P2)	A7H A0H. 7	A6H A0H. 6	A5H A0H. 5	A4H A0H. 4	A3H A0H. 3	A2H A0H. 2	A1H A0H. 1	A0H A0H. 0
98H (SCON)	9FH 98H. 7	9EH 98H. 6	9DH 98H. 5	9CH 98H. 4	9BH 98H. 3	9AH 98H. 2	99H 98H. 1	98H 98H. 0
90H (P1)	97H 90H. 7	96H 90H. 6	95H 90H. 5	94H 90H. 4	93H 90H. 3	92H 90H. 2	91H 90H. 1	90H 90H. 0
88H (TCON)	8FH 88H. 7	8EH 88H. 6	8DH 88H. 5	8CH 88H. 4	8BH 88H. 3	8AH 88H. 2	89H 88H. 1	88H 88H. 0
80H (P0)	87H 80H. 7	86H 80H. 6	85H 80H. 5	84H 80H. 4	83H 80H. 3	82H 80H. 2	81H 80H. 1	80H 80H. 0
2FH	7FH 2FH. 7	7EH 2FH. 6	7DH 2FH. 5	7CH 2FH. 4	7BH 2FH. 3	7AH 2FH. 2	79H 2FH. 1	78H 2FH. 0
2EH	77H 2EH. 7	76H 2EH. 6	75H 2EH. 5	74H 2EH. 4	73H 2EH. 3	72H 2EH. 2	71H 2EH. 1	70H 2EH. 0
2DH	6FH 2DH. 7	6EH 2DH. 6	6DH 2DH. 5	6CH 2DH. 4	6BH 2DH. 3	6AH 2DH. 2	69H 2DH. 1	68H 2DH. 0
2CH	67H 2CH. 7	66H 2CH. 6	65H 2CH. 5	64H 2CH. 4	63H 2CH. 3	62H 2CH. 2	61H 2CH. 1	60H 2CH. 0
2BH	5FH 2BH. 7	5EH 2BH. 6	5DH 2BH. 5	5CH 2BH. 4	5BH 2BH. 3	5AH 2BH. 2	59H 2BH. 1	58H 2BH. 0
2AH	57H 2AH. 7	56H 2AH. 6	55H 2AH. 5	54H 2AH. 4	53H 2AH. 3	52H 2AH. 2	51H 2AH. 1	50H 2AH. 0

29H	4FH 29H. 7	4EH 29H. 6	4DH 29H. 5	4CH 29H. 4	4BH 29H. 3	4AH 29H. 2	49H 29H. 1	48H 29H. 0
28H	47H 28H. 7	46H 28H. 6	45H 28H. 5	44H 28H. 4	43H 28H. 3	42H 28H. 2	41H 28H. 1	40H 28H. 0
27H	3FH 27H. 7	3EH 27H. 6	3DH 27H. 5	3CH 27H. 4	3BH 27H. 3	3AH 27H. 2	39H 27H. 1	38H 27H. 0
26H	37H 26H. 7	36H 26H. 6	35H 26H. 5	34H 26H. 4	33H 26H. 3	32H 26H. 2	31H 26H. 1	30H 26H. 0
25H	2FH 25H. 7	2EH 25H. 6	2DH 25H. 5	2CH 25H. 4	2BH 25H. 3	2AH 25H. 2	29H 25H. 1	28H 25H. 0
24H	27H 24H. 7	26H 24H. 6	25H 24H. 5	24H 24H. 4	23H 24H. 3	22H 24H. 2	21H 24H. 1	20H 24H. 0
23H	1FH 23H. 7	1EH 23H. 6	1DH 23H. 5	1CH 23H. 4	1BH 23H. 3	1AH 23H. 2	19H 23H. 1	18H 23H. 0
22H	17H 22H. 7	16H 22H. 6	15H 22H. 5	14H 22H. 4	13H 22H. 3	12H 22H. 2	11H 22H. 1	10H 22H. 0
21H	0FH 21H. 7	0EH 21H. 6	0DH 21H. 5	0CH 21H. 4	0BH 21H. 3	0AH 21H. 2	09H 21H. 1	08H 21H. 0
20H	07H 20H. 7	06H 20H. 6	05H 20H. 5	04H 20H. 4	03H 20H. 3	02H 20H. 2	01H 20H. 1	00H 20H. 0

7 特殊功能寄存器

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8H		CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H		
F0H	B		PCA_PWM0	PCA_PWM1	PCA_PWM2	PCA_PWM3		
E8H		CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L		
E0H	ACC	WDT_CONTR	IAP_DATA	IAP_ADDRH	IAP_ADDRL	IAP_CMD	IAP_TRIG	IAP_CONTR
D8H	CCON	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3		
D0H	PSW							
C8H								
C0H						ADC_CONTR	ADC_RES	CLK_DIV
B8H	IP	SADEN					ADC_RESL	
B0H	P3	P3M1	P3M0					IPH
A8H	IE	SADDR						
A0H	P2							
98H	SCON	SBUF						
90H	P1	P1M1	P1M0	P0M1	P0M0	P2M1	P2M0	
88H	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	WAKE_CLKO
80H	P0	SP	DPL	DPH	SPSTAT	SPCTL	SPDAT	PCON

可位寻址

不可位寻址

注意：寄存器地址能够被 8 整除的才可进行位寻址，不能被 8 整除的则不可位寻址

7.1 特殊功能寄存器列表

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
P0	P0 端口	80H	P07	P06	P05	P04	P03	P02	P01	P00	1111,1111
SP	堆栈指针	81H									0000,0111
DPL	数据指针（低字节）	82H									0000,0000
DPH	数据指针（高字节）	83H									0000,0000
SPSTAT	SPI 状态寄存器	84H	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
SPCTL	SPI 控制寄存器	85H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]		0000,0100
SPDAT	SPI 数据寄存器	86H									0000,0000
PCON	电源控制寄存器	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
TCON	定时器控制寄存器	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
TMOD	定时器模式寄存器	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000,0000
TL0	定时器 0 低 8 位寄存器	8AH									0000,0000
TL1	定时器 1 低 8 位寄存器	8BH									0000,0000
TH0	定时器 0 高 8 位寄存器	8CH									0000,0000
TH1	定时器 1 高 8 位寄存器	8DH									0000,0000
AUXR	辅助寄存器 1	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2	0000,0001
WAKE_CLKO	中断与时钟输出控制	8FH	-	EX4	EX3	EX2	MCKO_S2	T2CLKO	T1CLKO	T0CLKO	x000,0000
P1	P1 端口	90H	P17	P16	P15	P14	P13	P12	P11	P10	1111,1111
P1M1	P1 口配置寄存器 1	91H	P17M1	P16M1	P15M1	P14M1	P13M1	P12M1	P11M1	P10M1	1111,1111
P1M0	P1 口配置寄存器 0	92H	P17M0	P16M0	P15M0	P14M0	P13M0	P12M0	P11M0	P10M0	0000,0000
P0M1	P0 口配置寄存器 1	93H	P07M1	P06M1	P05M1	P04M1	P03M1	P02M1	P01M1	P00M1	1111,1111
P0M0	P0 口配置寄存器 0	94H	P07M0	P06M0	P05M0	P04M0	P03M0	P02M0	P01M0	P00M0	0000,0000
P2M1	P2 口配置寄存器 1	95H	P27M1	P26M1	P25M1	P24M1	P23M1	P22M1	P21M1	P20M1	1111,1111
P2M0	P2 口配置寄存器 0	96H	P27M0	P26M0	P25M0	P24M0	P23M0	P22M0	P21M0	P20M0	0000,0000
SCON	串口 1 控制寄存器	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
SBUF	串口 1 数据寄存器	99H									0000,0000
P2	P2 端口	A0H	P27	P26	P25	P24	P23	P22	P21	P20	1111,1111
IE	中断允许寄存器	A8H	EA	ELVD	EADC	ES	ET1	EX1	ET0	EX0	0000,0000
SADDR	串口 1 从机地址寄存器	A9H									0000,0000
P3	P3 端口	B0H	P37	P36	P35	P34	P33	P32	P31	P30	1111,1111
P3M1	P3 口配置寄存器 1	B1H	P37M1	P36M1	P35M1	P34M1	P33M1	P32M1	P31M1	P30M1	1111,1100
P3M0	P3 口配置寄存器 0	B2H	P37M0	P36M0	P35M0	P34M0	P33M0	P32M0	P31M0	P30M0	0000,0000
IP	中断优先级控制寄存器	B8H		PPCA_LVD	PADC_SPI	PS	PT1	PX1	PT0	PX0	0000,0000
IP	中断优先级控制寄存器	B8H		PLVD_LVDH	PADC_SPIH	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
SADEN	串口 1 从机地址屏蔽寄存器	B9H									0000,0000
ADC_RES1	ADC 转换结果低位寄存器	BEH									0000,0000
ADC_CONTR	ADC 控制寄存器	C5H	ADC_POWER	SPEED[1:0]		ADC_FLAG	ADC_START	ADC_CHS[2:0]			0000,0000
ADC_RES	ADC 转换结果高位寄存器	C6H									0000,0000
CLK_DIV	时钟分频寄存器	C7H									0000,0000
PSW	程序状态字寄存器	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	0000,0000

CCON	PCA 控制寄存器	D8H	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,0000
CMOD	PCA 模式寄存器	D9H	CIDL	-	-	-	CPS[2:0]			ECF	0xxx,0000
CCAPM0	PCA 模块 0 模式控制寄存器	DAH	-	ECOM0	CCAPP0	CCAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	PCA 模块 1 模式控制寄存器	DBH	-	ECOM1	CCAPP1	CCAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	PCA 模块 2 模式控制寄存器	DCH	-	ECOM2	CCAPP2	CCAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	PCA 模块 3 模式控制寄存器	DDH	-	ECOM3	CCAPP3	CCAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000
ACC	累加器	E0H									0000,0000
WDT_CONTR	看门狗控制寄存器	E1H	WDT_FLAG	-	EN_WDT	CLR_WDT	IDL_WDT	WDT_PS[2:0]			0xn0,nnnn
IAP_DATA	IAP 数据寄存器	E2H									1111,1111
IAP_ADDRH	IAP 高地址寄存器	E3H									0000,0000
IAP_ADDRL	IAP 低地址寄存器	E4H									0000,0000
IAP_CMD	IAP 命令寄存器	E5H	-	-	-	-	-	-	CMD[1:0]		xxxx,xx00
IAP_TRIG	IAP 触发寄存器	E6H									0000,0000
IAP_CONTR	IAP 控制寄存器	E7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT[2:0]			0000,x000
CCAP0L	PCA 模块 0 低字节	EAH									0000,0000
CCAP1L	PCA 模块 1 低字节	EBH									0000,0000
CCAP2L	PCA 模块 2 低字节	ECH									0000,0000
CCAP3L	PCA 模块 3 低字节	EDH									0000,0000
B	B 寄存器	F0H									0000,0000
PCA_PWM0	PCA0 的 PWM 模式寄存器	F2H			-	-	-	-	EPC0H	EPC0L	00xx,xx00
PCA_PWM1	PCA1 的 PWM 模式寄存器	F3H				-	-	-	EPC1H	EPC1L	00xx,xx00
PCA_PWM2	PCA2 的 PWM 模式寄存器	F4H			-	-	-	-	EPC2H	EPC2L	00xx,xx00
PCA_PWM3	PCA3 的 PWM 模式寄存器	F5H				-	-	-	EPC3H	EPC3L	00xx,xx00
CH	PCA 计数器高字节	F9H									0000,0000
CCAP0H	PCA 模块 0 高字节	FAH									0000,0000
CCAP1H	PCA 模块 1 高字节	FBH									0000,0000
CCAP2H	PCA 模块 2 高字节	FCH									0000,0000
CCAP3H	PCA 模块 3 高字节	FDH									0000,0000

8 I/O 口

所有的 I/O 口均有 4 种工作模式：准双向口/弱上拉（标准 8051 输出口模式）、推挽输出/强上拉、高阻输入（电流既不能流入也不能流出）、开漏输出。可使用软件对 I/O 口的工作模式进行容易配置。

8.1 I/O 口相关寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
P0	P0 端口	80H	P07	P06	P05	P04	P03	P02	P01	P00	1111,1111
P1	P1 端口	90H	P17	P16	P15	P14	P13	P12	P11	P10	1111,1111
P2	P2 端口	A0H	P27	P26	P25	P24	P23	P22	P21	P20	1111,1111
P3	P3 端口	B0H	P37	P36	P35	P34	P33	P32	P31	P30	1111,1111
P0M1	P0 口配置寄存器 1	93H	P07M1	P06M1	P05M1	P04M1	P03M1	P02M1	P01M1	P00M1	0000,0000
P0M0	P0 口配置寄存器 0	94H	P07M0	P06M0	P05M0	P04M0	P03M0	P02M0	P01M0	P00M0	0000,0000
P1M1	P1 口配置寄存器 1	91H	P17M1	P16M1	P15M1	P14M1	P13M1	P12M1	P11M1	P10M1	0000,0000
P1M0	P1 口配置寄存器 0	92H	P17M0	P16M0	P15M0	P14M0	P13M0	P12M0	P11M0	P10M0	0000,0000
P2M1	P2 口配置寄存器 1	95H	P27M1	P26M1	P25M1	P24M1	P23M1	P22M1	P21M1	P20M1	0000,0000
P2M0	P2 口配置寄存器 0	96H	P27M0	P26M0	P25M0	P24M0	P23M0	P22M0	P21M0	P20M0	0000,0000
P3M1	P3 口配置寄存器 1	B1H	P37M1	P36M1	P35M1	P34M1	P33M1	P32M1	P31M1	P30M1	0000,0000
P3M0	P3 口配置寄存器 0	B2H	P37M0	P36M0	P35M0	P34M0	P33M0	P32M0	P31M0	P30M0	n000,0000

8.1.1 端口数据寄存器（Px）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
P0	80H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2	A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3	B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

读写端口状态

写 0：输出低电平到端口缓冲区

写 1：输出高电平到端口缓冲区

读：直接读端口管脚上的电平

8.1.2 端口模式配置寄存器（PxM0，PxM1）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
P0M0	94H	P07M0	P06M0	P05M0	P04M0	P03M0	P02M0	P01M0	P00M0
P0M1	93H	P07M1	P06M1	P05M1	P04M1	P03M1	P02M1	P01M1	P00M1
P1M0	92H	P17M0	P16M0	P15M0	P14M0	P13M0	P12M0	P11M0	P10M0
P1M1	91H	P17M1	P16M1	P15M1	P14M1	P13M1	P12M1	P11M1	P10M1
P2M0	96H	P27M0	P26M0	P25M0	P24M0	P23M0	P22M0	P21M0	P20M0

P2M1	95H	P27M1	P26M1	P25M1	P24M1	P23M1	P22M1	P21M1	P20M1
P3M0	B2H	P37M0	P36M0	P35M0	P34M0	P33M0	P32M0	P31M0	P30M0
P3M1	B1H	P37M1	P36M1	P35M1	P34M1	P33M1	P32M1	P31M1	P30M1

配置端口的模式

PnM1.x	PnM0.x	Pn.x 口工作模式
0	0	准双向口
0	1	推挽输出
1	0	高阻输入
1	1	开漏输出

8.2 配置 I/O 口

每个 I/O 的配置都需要使用两个寄存器进行设置。

以 P0 口为例，配置 P0 口需要使用 P0M0 和 P0M1 两个寄存器进行配置，如下图所示：

即 P0M0 的第 0 位和 P0M1 的第 0 位组合起来配置 P0.0 口的模式

即 P0M0 的第 1 位和 P0M1 的第 1 位组合起来配置 P0.1 口的模式

其他所有 I/O 的配置都与此类似。

PnM0 与 PnM1 的组合方式如下表所示

PnM1	PnM0	I/O 口工作模式
0	0	准双向口（传统8051端口模式，弱上拉） 灌电流可达20mA，拉电流为270~150μA（存在制造误差）
0	1	推挽输出（强上拉输出，可达20mA，要加限流电阻）
1	0	高阻输入（电流既不能流入也不能流出）
1	1	开漏输出（Open-Drain），内部上拉电阻断开 开漏模式既可读外部状态也可对外输出（高电平或低电平）。如要正确读外部状态或需要对外输出高电平，需外加上拉电阻，否则读不到外部状态，也对外输出不出高电平。

注：n = 0, 1, 2, 3, 4, 5, 6, 7

8.3 I/O 的结构图

8.3.1 准双向口（弱上拉）

准双向口（弱上拉）输出类型可用作输出和输入功能而不需重新配置端口输出状态。这是因为当端口输出为 1 时驱动能力很弱，允许外部装置将其拉低。当引脚输出为低时，它的驱动能力很强，可吸收相当大的电流。准双向口有 3 个上拉晶体管适应不同的需要。

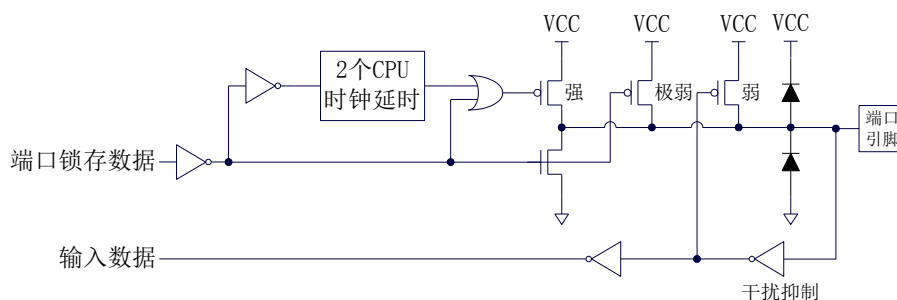
在 3 个上拉晶体管中，有 1 个上拉晶体管称为“弱上拉”，当端口寄存器为 1 且引脚本身为 1 时打开。此上拉提供基本驱动电流使准双向口输出为 1。如果一个引脚输出为 1 而由外部装置下拉到低时，弱上拉关闭而“极弱上拉”维持开状态，为了把这个引脚强拉为低，外部装置必须有足够的灌电流能力使引脚上的电压降到门槛电压以下。对于 5V 单片机，“弱上拉”晶体管的电流约 250uA；对于 3.3V 单片机，“弱上拉”晶体管的电流约 150uA。

第 2 个上拉晶体管，称为“极弱上拉”，当端口锁存为 1 时打开。当引脚悬空时，这个极弱的上拉源产生很弱的上拉电流将引脚上拉为高电平。对于 5V 单片机，“极弱上拉”晶体管的电流约 18uA；对于 3.3V 单片机，“极弱上拉”晶体管的电流约 5uA。

第 3 个上拉晶体管称为“强上拉”。当端口锁存器由 0 到 1 跳变时，这个上拉用来加快准双向口由逻辑 0 到逻辑 1 转换。当发生这种情况时，强上拉打开约 2 个时钟以使引脚能够迅速地上拉到高电平。

准双向口（弱上拉）带有一个施密特触发输入以及一个干扰抑制电路。准双向口（弱上拉）读外部状态前，要先锁存为‘1’，才可读到外部正确的状态。

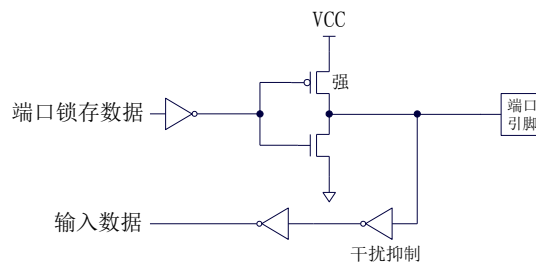
准双向口（弱上拉）输出如下图所示：



8.3.2 推挽输出

强推挽输出配置的下拉结构与开漏输出以及准双向口的下拉结构相同，但当锁存器为 1 时提供持续的强上拉。推挽模式一般用于需要更大驱动电流的情况。

强推挽引脚配置如下图所示：

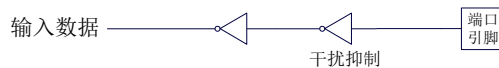


8.3.3 高阻输入

电流既不能流入也不能流出

输入口带有一个施密特触发输入以及一个干扰抑制电路

高阻输入引脚配置如下图所示：



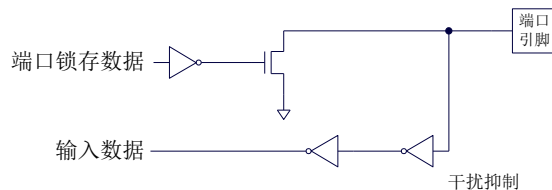
8.3.4 开漏输出

开漏模式既可读外部状态也可对外输出（高电平或低电平）。如要正确读外部状态或需要对外输出高电平，需外加上拉电阻。

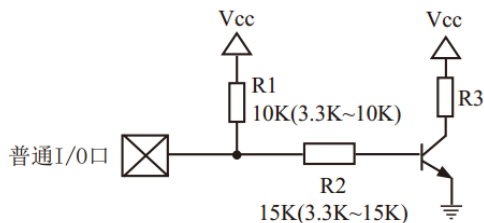
当端口锁存器为 0 时，开漏输出关闭所有上拉晶体管。当作为一个逻辑输出高电平时，这种配置方式必须有外部上拉，一般通过电阻外接到 V_{CC} 。如果外部有上拉电阻，开漏的 I/O 口还可读外部状态，即此时被配置为开漏模式的 I/O 口还可作为输入 I/O 口。这种方式的下拉与准双向口相同。

开漏端口带有一个施密特触发输入以及一个干扰抑制电路。

输出端口配置如下图所示：



8.4 一种典型三极管控制电路



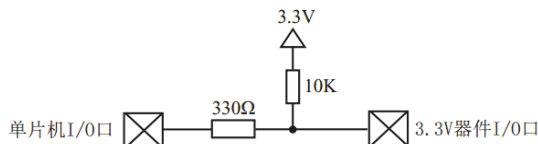
如果上拉控制，建议加上拉电阻 $R1(3.3K \sim 10K)$ ，如果不加上拉电阻 $R1(3.3K \sim 10K)$ ，建议 $R2$ 的值在 $15K$ 以上，或用强推挽输出。

8.5 典型发光二极管控制电路



8.6 混合电压供电系统 3V/5V 器件 I/O 口互连

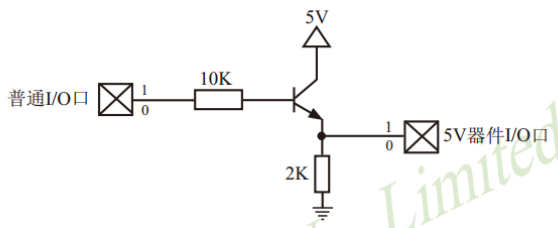
STC 系列宽电压单片机工作在 5V 时，如需要直接连接 3.3V 器件时，为防止 3.3V 器件承受不了 5V，可将相应的单片机 I/O 口先串一个 330Ω 的限流电阻到 3.3V 器件 I/O 口，程序初始化时将单片机的 I/O 口设置成开漏配置，断开内部上拉电阻，相应的 3.3V 器件 I/O 口外部加 10K 上拉电阻到 3.3V 器件的 Vcc，这样高电平是 3.3V，低电平是 0V，输入输出一切正常。



STC 宽电压单片机工作在 3V 时，如需要直接连接 5V 器件时，如果相应的 I/O 口是输入，可在该 I/O 口上串接一个隔离二极管，隔离高压部分。外部信号电压高于单片机工作电压时截止，I/O 口因内部上拉到高电平，所以读 I/O 口状态是高电平；外部信号电压为低时导通，I/O 口被钳位在 0.7V，小于 0.8V 时单片机读 I/O 口状态是低电平。



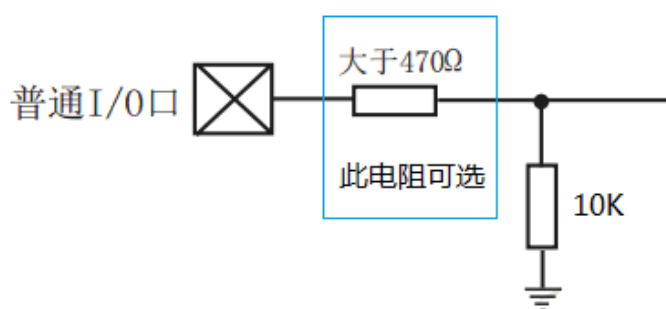
STC 宽电压单片机工作在 3V 时，如需要直接连接 5V 器件时，如果相应的 I/O 口是输出，可用一个 NPN 三极管隔离，电路如下：



8.7 如何让 I/O 口上电复位时为低电平

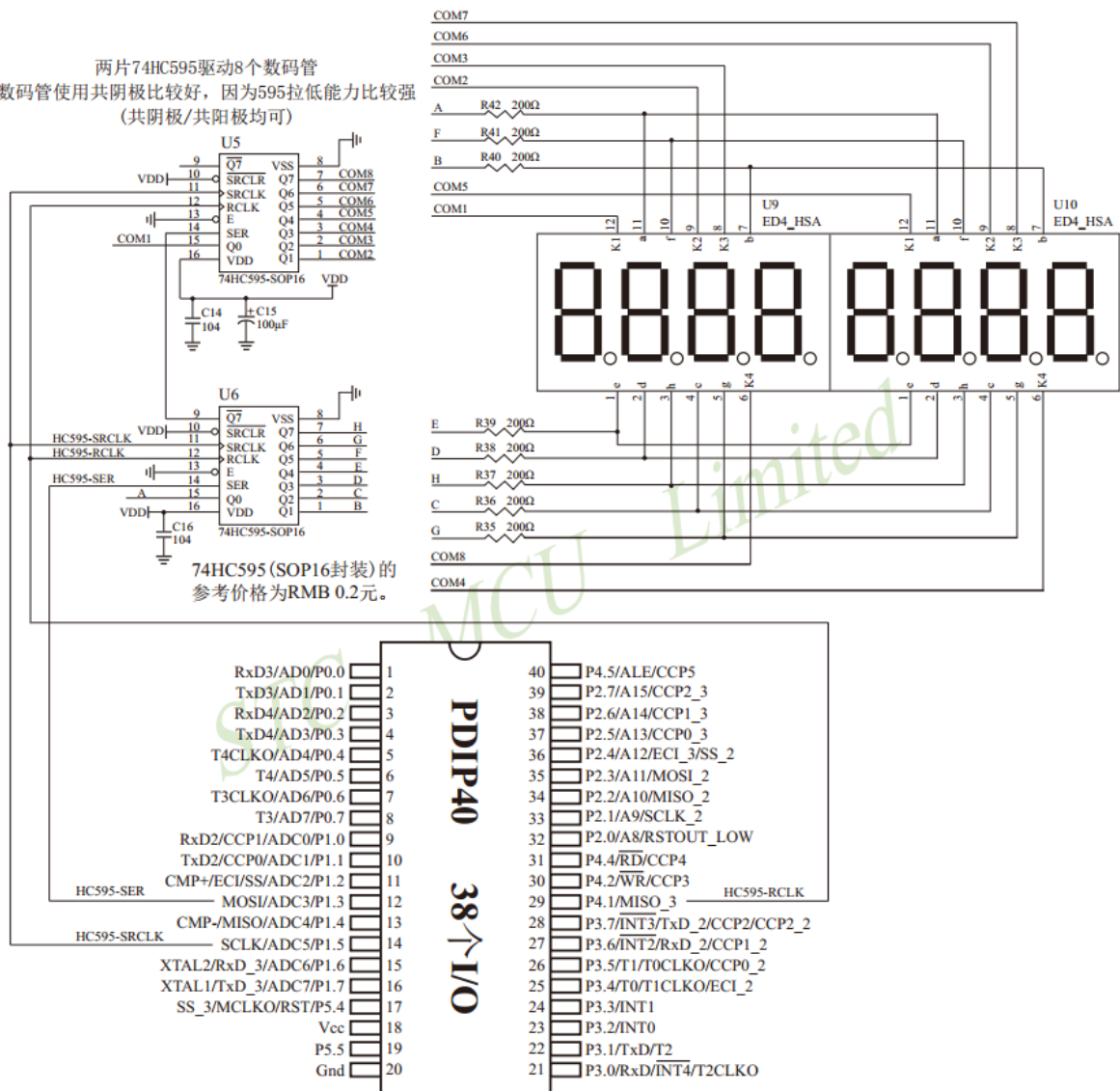
传统 8051 单片机上电复位时普通 I/O 口为弱上拉(准双向口)高电平输出,而很多实际应用要求上电时某些 I/O 口为低电平输出,否则所控制的系统(如马达)就会误动作,新一代单片机由于所有的 I/O 复位后是高阻输入(除 P3.0/P3.1 是传统的弱上拉),加一个下拉电阻就可保证上电时为低电平,后续要改为高电平,只需要将 I/O 的模式改为强推挽输出,对外输出高电平即可。

如下:在 STC 的单片机 I/O 口上加一个下拉电阻(10K 左右),这样上电复位时,除了下载口 P3.0 和 P3.1 为弱上拉(准双向口)外,其他 I/O 口均为高阻输入模式,而外部有下拉电阻,所以该 I/O 口上电复位时外部为低电平。如果要将此 I/O 口驱动为高电平,可将此 I/O 口设置为强推挽输出,而强推挽输出时,I/O 口驱动电流可达 20mA,故肯定可以将该口驱动为高电平输出。

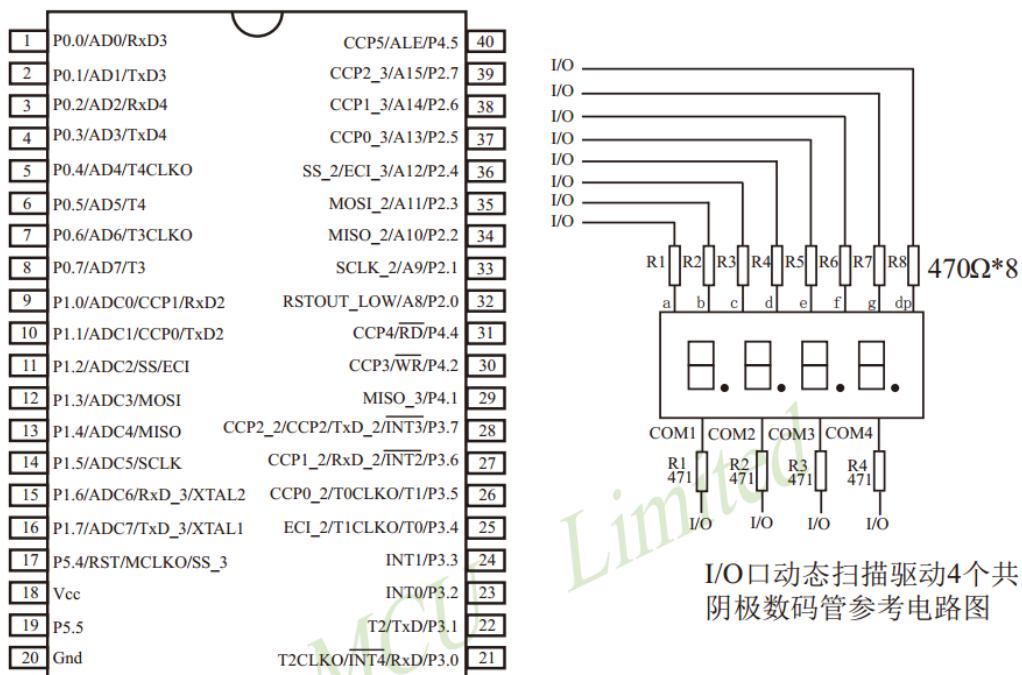


8.8 利用 74HC595 驱动 8 个数码管(串行扩展,3 根线)的线路图

两片74HC595驱动8个数码管
数码管使用共阴极比较好,因为595拉低能力比较强
(共阴极/共阳极均可)



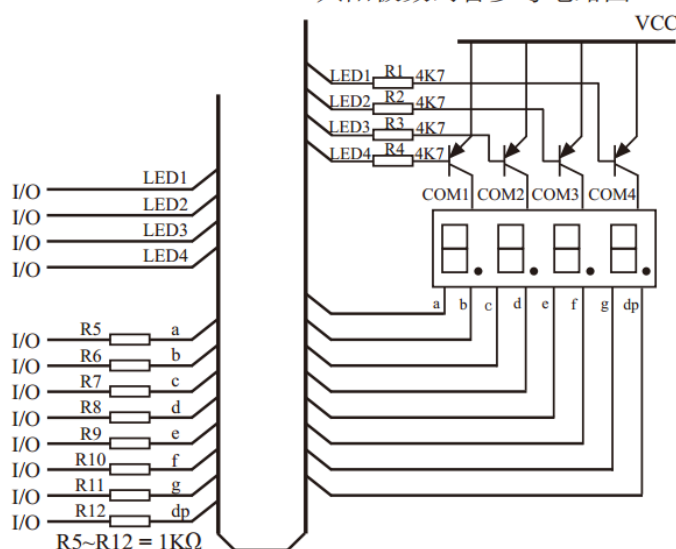
8.9 I/O 口直接驱动 LED 数码管应用线路图



I/O口动态扫描驱动4个共阴极数码管参考电路图

I/O 口动态扫描驱动数码时，可以一次点亮一个数码管中的8段，但为降低功耗，建议可以一次只点亮其中的4段或者2段

I/O 口动态扫描驱动4个共阳极数码管参考电路图



9 指令系统

助记符	指令说明	字节
ADD A,Rn	寄存器内容加到累加器	1
ADD A,direct	直接地址单元的数据加到累加器	2
ADD A,@Ri	间接地址单元的数据加到累加器	1
ADD A,#data	立即数加到累加器	2
ADDC A,Rn	寄存器带进位加到累加器	1
ADDC A,direct	直接地址单元的数据带进位加到累加器	2
ADDC A,@Ri	间接地址单元的数据带进位加到累加器	1
ADDC A,#data	立即数带进位加到累加器	2
SUBB A,Rn	累加器带借位减寄存器内容	1
SUBB A,direct	累加器带借位减直接地址单元的内容	2
SUBB A,@Ri	累加器带借位减间接地址单元的内容	1
SUBB A,#data	累加器带借位减立即数	2
INC A	累加器加1	1
INC Rn	寄存器加1	1
INC direct	直接地址单元加1	2
INC @Ri	间接地址单元加1	1
DEC A	累加器减1	1
DEC Rn	寄存器减1	1
DEC direct	直接地址单元减1	2
DEC @Ri	间接地址单元减1	1
INC DPTR	地址寄存器DPTR加1	1
MUL AB	A乘以B, B存放高字节, A存放低字节	1
DIV AB	A除以B, B存放余数, A存放商	1
DA A	累加器十进制调整	1
ANL A,Rn	累加器与寄存器相与	1
ANL A,direct	累加器与直接地址单元相与	2
ANL A,@Ri	累加器与间接地址单元相与	1
ANL A,#data	累加器与立即数相与	2
ANL direct,A	直接地址单元与累加器相与	2
ANL direct,#data	直接地址单元与立即数相与	3
ORL A,Rn	累加器与寄存器相或	1
ORL A,direct	累加器与直接地址单元相或	2
ORL A,@Ri	累加器与间接地址单元相或	1

ORL	A,#data	累加器与立即数相或	2
ORL	direct,A	直接地址单元与累加器相或	2
ORL	direct,#data	直接地址单元与立即数相或	3
XRL	A,Rn	累加器与寄存器相异或	1
XRL	A,direct	累加器与直接地址单元相异或	2
XRL	A,@Ri	累加器与间接地址单元相异或	1
XRL	A,#data	累加器与立即数相异或	2
XRL	direct,A	直接地址单元与累加器相异或	2
XRL	direct,#data	直接地址单元与立即数相异或	3
CLR	A	累加器清0	1
CPL	A	累加器取反	1
RL	A	累加器循环左移	1
RLC	A	累加器带进位循环左移	1
RR	A	累加器循环右移	1
RRC	A	累加器带进位循环右移	1
SWAP	A	累加器高低半字节交换	1
CLR	C	清零进位位	1
CLR	bit	清0直接地址位	2
SETB	C	置1进位位	1
SETB	bit	置1直接地址位	2
CPL	C	进位位求反	1
CPL	bit	直接地址位求反	2
ANL	C,bit	进位位和直接地址位相与	2
ANL	C,/bit	进位位和直接地址位的反码相与	2
ORL	C,bit	进位位和直接地址位相或	2
ORL	C,/bit	进位位和直接地址位的反码相或	2
MOV	C,bit	直接地址位送入进位位	2
MOV	bit,C	进位位送入直接地址位	2
MOV	A,Rn	寄存器内容送入累加器	1
MOV	A,direct	直接地址单元中的数据送入累加器	2
MOV	A,@Ri	间接地址中的数据送入累加器	1
MOV	A,#data	立即数送入累加器	2
MOV	Rn,A	累加器内容送入寄存器	1
MOV	Rn,direct	直接地址单元中的数据送入寄存器	2
MOV	Rn,#data	立即数送入寄存器	2
MOV	direct,A	累加器内容送入直接地址单元	2
MOV	direct,Rn	寄存器内容送入直接地址单元	2

MOV	direct,direct	直接地址单元中的数据送入另一个直接地址单元	3
MOV	direct,@Ri	间接地址中的数据送入直接地址单元	2
MOV	direct,#data	立即数送入直接地址单元	3
MOV	@Ri,A	累加器内容送间接地址单元	1
MOV	@Ri,direct	直接地址单元数据送入间接地址单元	2
MOV	@Ri,#data	立即数送入间接地址单元	2
MOV	DPTR,#data16	16位立即数送入数据指针	3
MOVC	A,@A+DPTR	以DPTR为基地址变址寻址单元中的数据送入累加器	1
MOVC	A,@A+PC	以PC为基地址变址寻址单元中的数据送入累加器	1
MOVB	A,@Ri	扩展地址(8位地址)的内容送入累加器A中	1
MOVB	A,@DPTR	扩展RAM(16位地址)的内容送入累加器A中	1
MOVB	@Ri,A	将累加器A的内容送入扩展RAM(8位地址)中	1
MOVB	@DPTR,A	将累加器A的内容送入扩展RAM(16位地址)中	1
PUSH	direct	直接地址单元中的数据压入堆栈	2
POP	direct	栈底数据弹出送入直接地址单元	2
XCH	A,Rn	寄存器与累加器交换	1
XCH	A,direct	直接地址单元与累加器交换	2
XCH	A,@Ri	间接地址与累加器交换	1
XCHD	A,@Ri	间接地址的低半字节与累加器交换	1
ACALL	addr11	短调用子程序	2
LCALL	addr16	长调用子程序	3
RET		子程序返回	1
RETI		中断返回	1
AJMP	addr11	短跳转	2
LJMP	addr16	长跳转	3
SJMP	rel	相对跳转	2
JMP	@A+DPTR	相对于DPTR的间接跳转	1
JZ	rel	累加器为零跳转	2
JNZ	rel	累加器非零跳转	2
JC	rel	进位位为1跳转	2
JNC	rel	进位位为0跳转	2
JB	bit,rel	直接地址位为1则跳转	3
JNB	bit,rel	直接地址位为0则跳转	3
JBC	bit,rel	直接地址位为1则跳转, 该位清0	3
CJNE	A,direct,rel	累加器与直接地址单元不相等跳转	3
CJNE	A,#data,rel	累加器与立即数不相等跳转	3
CJNE	Rn,#data,rel	寄存器与立即数不相等跳转	3

CJNE	@Ri,#data,rel	间接地址单元与立即数不相等跳转	3
DJNZ	Rn,rel	寄存器减1后非零跳转	2
DJNZ	direct,rel	直接地址单元减1后非零跳转	3
NOP		空操作	1

[1]: 访问外部扩展 RAM 时, 指令的执行周期与寄存器 BUS_SPEED 中的 SPEED[2:0]位有关

[2]: 对于条件跳转语句的执行时间会依据条件是否满足而不同。当条件不满足时, 不会发生跳转而继续执行下一条指令, 此时条件跳转语句的执行时间为 1 个时钟; 当条件满足时, 则会发生跳转, 此时条件跳转语句的执行时间为 3 个时钟。

[3]: 对于条件跳转语句的执行时间会依据条件是否满足而不同。当条件不满足时, 不会发生跳转而继续执行下一条指令, 此时条件跳转语句的执行时间为 2 个时钟; 当条件满足时, 则会发生跳转, 此时条件跳转语句的执行时间为 3 个时钟。

10 中断系统

中断系统是为使 CPU 具有对外界紧急事件的实时处理能力而设置的。

当中央处理机 CPU 正在处理某件事的时候外界发生了紧急事件请求，要求 CPU 暂停当前的工作，转而去处理这个紧急事件，处理完以后，再回到原来被中断的地方，继续原来的工作，这样的过程称为中断。实现这种功能的部件称为中断系统，请示 CPU 中断的请求源称为中断源。微型机的中断系统一般允许多个中断源，当几个中断源同时向 CPU 请求中断，要求为它服务的时候，这就存在 CPU 优先响应哪一个中断源请求的问题。通常根据中断源的轻重缓急排队，优先处理最紧急事件的中断请求源，即规定每一个中断源有一个优先级别。CPU 总是先响应优先级别最高的中断请求。

当 CPU 正在处理一个中断源请求的时候（执行相应的中断服务程序），发生了另外一个优先级比它还高的中断源请求。如果 CPU 能够暂停对原来中断源的服务程序，转而去处理优先级更高的中断请求源，处理完以后，再回到原低级中断服务程序，这样的过程称为中断嵌套。这样的中断系统称为多级中断系统，没有中断嵌套功能的中断系统称为单级中断系统。

用户可以用关总中断允许位（EA/IE.7）或相应中断的允许位屏蔽相应的中断请求，也可以用打开相应的中断允许位来使 CPU 响应相应的中断申请，每一个中断源可以用软件独立地控制为开中断或关中断状态，部分中断的优先级别均可用软件设置。高优先级的中断请求可以打断低优先级的中断，反之，低优先级的中断请求不可以打断高优先级的中断。当两个相同优先级的中断同时产生时，将由查询次序来决定系统先响应哪个中断。

10.1 中断源

下表中√表示对应的系列有相应的中断源

中断源	STC12C5628AD系列	STC12C5410AD 系列	STC12C2052AD 系列
外部中断 0 中断（INT0）	√	√	√
定时器 0 中断（Timer0）	√	√	√
外部中断 1 中断（INT1）	√	√	√
定时器 1 中断（Timer1）	√	√	√
串口 1 中断（UART1）	√	√	√
模数转换中断（ADC） 串行外设接口中断（SPI）	√	√	√
捕获中断（CCP/PCA/PWM） 低压检测中断（LVD）	√	√	√

10.2 中断列表

中断源	中断向量	次序	优先级设置	优先级	中断请求位	中断允许位
INT0	0003H	0	PX0	0/1	IE0	EX0
Timer0	000BH	1	PT0	0/1	TF0	ET0
INT1	0013H	2	PX1	0/1	IE1	EX1
Timer1	001BH	3	PT1	0/1	TF1	ET1
UART1	0023H	4	PS	0/1	RI TI	ES
ADC	002BH	5	PADC	0/1	ADC_FLAG	EADC
SPI			PSPI	0/1	SPIF	ESPI
PCA	0033H	6	PPCA	0/1	CF	ECF
					CCF0	ECCF0
					CCF1	ECCF1
					CCF2	ECCF2
					CCF3	ECCF3
LVD			PLVD	0/1	LVDF	ELVD

在 C 语言中声明中断服务程序

```

void INT0_Routine(void)    interrupt 0;
void TM0_Routine(void)    interrupt 1;
void INT1_Routine(void)    interrupt 2;
void TM1_Routine(void)    interrupt 3;
void UART1_Routine(void)   interrupt 4;
void ADC_SPI_Routine(void) interrupt 5;
void PCA_LVD_Routine(void) interrupt 6;

```

10.3 中断相关寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
IE	中断允许寄存器	A8H	EA	EPCA_LVD	EADC_SPI	ES	ET1	EX1	ET0	EX0	0000,0000
IP	中断优先级控制寄存器	B8H	-	PPCA_LVD	PADC_SPI	PS	PT1	PX1	PT0	PX0	0000,0000
IPH	中断优先级控制寄存器 2	B7H	-	PPCA_LVDH	PADC_SPIH	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
TCON	定时器控制寄存器	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
SCON	串口 1 控制寄存器	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
AUXR	辅助寄存器	8EH	T0x12	T1x12	UART_M0x6	EADCI	ESPI	ELVDI			0000,0000
PCON	电源控制寄存器	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
ADC_CONTR	ADC 控制寄存器	BCH	ADC_POWER	ADC_START	ADC_FLAG	ADC_EPWMT	ADC_CHS[3:0]				000x,0000
SPSTAT	SPI 状态寄存器	CDH	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
CCON	PCA 控制寄存器	D8H	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,x000
CMOD	PCA 模式寄存器	D9H	CIDL	-	-	-	CPS[2:0]			ECF	0xxx,0000
CCAPM0	PCA 模块 0 模式控制寄存器	DAH	-	ECOM0	CCAPP0	CCAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	PCA 模块 1 模式控制寄存器	DBH	-	ECOM1	CCAPP1	CCAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	PCA 模块 2 模式控制寄存器	DCH	-	ECOM2	CCAPP2	CCAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	PCA 模块 3 模式控制寄存器	DDH	-	ECOM3	CCAPP3	CCAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000

10.3.1 中断使能寄存器（中断允许位）

IE（中断使能寄存器）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IE	A8H	EA	EPCA_LVD	EADC_SPI	ES	ET1	EX1	ET0	EX0

EA：总中断允许控制位。EA 的作用是使中断允许形成多级控制。即各中断源首先受 EA 控制；其次还受各中断源自己的中断允许控制位控制。

0：CPU 屏蔽所有的中断申请

1：CPU 开放中断

EPCA_LVD：PCA/低压检测中断允许位。

0：禁止 PCA/低压检测中断

1：允许 PCA/低压检测中断

EADC_SPI：A/D 转换/SPI 中断允许位。

0：禁止 A/D 转换/SPI 中断

1：允许 A/D 转换/SPI 中断

ES：串行口 1 中断允许位。

0：禁止串行口 1 中断

1：允许串行口 1 中断

ET1：定时/计数器 T1 的溢出中断允许位。

0：禁止 T1 中断

1：允许 T1 中断

EX1：外部中断 1 中断允许位。

0: 禁止 INT1 中断

1: 允许 INT1 中断

ET0: 定时/计数器 T0 的溢出中断允许位。

0: 禁止 T0 中断

1: 允许 T0 中断

EX0: 外部中断 0 中断允许位。

0: 禁止 INT0 中断

1: 允许 INT0 中断

PCA/CCP/PWM 中断控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CMOD	D9H	CIDL	-	-	-	CPS[2:0]			ECF
CCAPM0	DAH	-	ECOM0	CCAPP0	CCAPN0	MAT0	TOG0	PWM0	ECCF0
CCAPM1	DBH	-	ECOM1	CCAPP1	CCAPN1	MAT1	TOG1	PWM1	ECCF1
CCAPM2	DCH	-	ECOM2	CCAPP2	CCAPN2	MAT2	TOG2	PWM2	ECCF2
CCAPM3	DDH	-	ECOM3	CCAPP3	CCAPN3	MAT3	TOG3	PWM3	ECCF3

ECF: PCA 计数器中断允许位。

0: 禁止 PCA 计数器中断

1: 允许 PCA 计数器中断

ECCF0: PCA 模块 0 中断允许位。

0: 禁止 PCA 模块 0 中断

1: 允许 PCA 模块 0 中断

ECCF1: PCA 模块 1 中断允许位。

0: 禁止 PCA 模块 1 中断

1: 允许 PCA 模块 1 中断

ECCF2: PCA 模块 2 中断允许位。

0: 禁止 PCA 模块 2 中断

1: 允许 PCA 模块 2 中断

ECCF3: PCA 模块 3 中断允许位。

0: 禁止 PCA 模块 3 中断

1: 允许 PCA 模块 3 中断

10.3.2 中断请求寄存器（中断标志位）

定时器控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1: 定时器1溢出中断标志。中断服务程序中，硬件自动清零。

TF0: 定时器0溢出中断标志。中断服务程序中，硬件自动清零。

IE1: 外部中断1中断请求标志。中断服务程序中，硬件自动清零。

IE0: 外部中断0中断请求标志。中断服务程序中，硬件自动清零。

串口控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SCON	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

TI: 串口1发送完成中断请求标志。需要软件清零。

RI: 串口1接收完成中断请求标志。需要软件清零。

电源管理寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL

LVDF: 低压检测中断请求标志。需要软件清零。

ADC 控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
ADC_CONTR	BCH	ADC_POWER	ADC_START	ADC_FLAG	ADC_EPWMT	ADC_CHS[3:0]			

ADC_FLAG: ADC转换完成中断请求标志。需要软件清零。

SPI 状态寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SPSTAT	CDH	SPIF	WCOL	-	-	-	-	-	-

SPIF: SPI数据传输完成中断请求标志。需要软件清零。

PCA 控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CCON	D8H	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0

CF: PCA计数器中断请求标志。需要软件清零。

CCF3: PCA模块3中断请求标志。需要软件清零。

CCF2: PCA模块2中断请求标志。需要软件清零。

CCF1: PCA模块1中断请求标志。需要软件清零。

CCF0: PCA模块0中断请求标志。需要软件清零。

10.3.3 中断优先级寄存器

中断优先级控制寄存器

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IP	B8H		PPCA_LVD	PADC_SPI	PS	PT1	PX1	PT0	PX0
IPH	B7H		PPCA_LVDH	PADC_SPIH	PSH	PT1H	PX1H	PT0H	PX0H

PX0H, PX0: 外部中断0中断优先级控制位

- 00: INT0 中断优先级为 0 级（最低级）
- 01: INT0 中断优先级为 1 级（较低级）
- 10: INT0 中断优先级为 2 级（较高级）
- 11: INT0 中断优先级为 3 级（最高级）

PT0H, PT0: 定时器0中断优先级控制位

- 00: 定时器 0 中断优先级为 0 级（最低级）
- 01: 定时器 0 中断优先级为 1 级（较低级）
- 10: 定时器 0 中断优先级为 2 级（较高级）
- 11: 定时器 0 中断优先级为 3 级（最高级）

PX1H, PX1: 外部中断1中断优先级控制位

- 00: INT1 中断优先级为 0 级（最低级）
- 01: INT1 中断优先级为 1 级（较低级）
- 10: INT1 中断优先级为 2 级（较高级）
- 11: INT1 中断优先级为 3 级（最高级）

PT1H, PT1: 定时器1中断优先级控制位

- 00: 定时器 1 中断优先级为 0 级（最低级）
- 01: 定时器 1 中断优先级为 1 级（较低级）
- 10: 定时器 1 中断优先级为 2 级（较高级）
- 11: 定时器 1 中断优先级为 3 级（最高级）

PSH,PS: 串口1中断优先级控制位

- 00: 串口 1 中断优先级为 0 级（最低级）
- 01: 串口 1 中断优先级为 1 级（较低级）
- 10: 串口 1 中断优先级为 2 级（较高级）
- 11: 串口 1 中断优先级为 3 级（最高级）

PADC_SPIH, PADC_SPI: ADC、SPI中断优先级控制位

- 00: ADC、SPI 中断优先级为 0 级（最低级）
- 01: ADC、SPI 中断优先级为 1 级（较低级）
- 10: ADC、SPI 中断优先级为 2 级（较高级）
- 11: ADC、SPI 中断优先级为 3 级（最高级）

PPCA_LVDH, PPCA_LVD: 低压检测、PCA中断优先级控制位

- 00: LVD、PCA 中断优先级为 0 级（最低级）
- 01: LVD、PCA 中断优先级为 1 级（较低级）
- 10: LVD、PCA 中断优先级为 2 级（较高级）
- 11: LVD、PCA 中断优先级为 3 级（最高级）

11 定时器/计数器

单片机内部设置了 2 个 16 位定时器/计数器。2 个 16 位定时器 T0、T1 都具有计数方式和定时方式两种工作方式。对定时器/计数器 T0 和 T1，用它们在特殊功能寄存器 TMOD 中相对应的控制位 C/T 来选择 T0 或 T1 为定时器还是计数器。定时器/计数器的核心部件是一个加法计数器，其本质是对脉冲进行计数。只是计数脉冲来源不同：如果计数脉冲来自系统时钟，则为定时方式，此时定时器/计数器每 12 个时钟或者每 1 个时钟得到一个计数脉冲，计数值加 1；如果计数脉冲来自单片机外部引脚，则为计数方式，每来一个脉冲加 1。

当定时器/计数器 T0、T 工作在定时模式时，特殊功能寄存器 AUXR 中的 T0x12、T1x12 分别决定是系统时钟/12 还是系统时钟/1（不分频）后让 T0、T1 进行计数。当定时器/计数器工作在计数模式时，对外部脉冲计数不分频。

定时器/计数器 0 有 4 种工作模式：模式 0（16 位自动重装载模式），模式 1（16 位不可重装载模式），模式 2（8 位自动重装模式），模式 3（不可屏蔽中断的 16 位自动重装载模式）。定时器/计数器 1 除模式 3 外，其他工作模式与定时器/计数器 0 相同。T1 在模式 3 时无效，停止计数。

11.1 定时器的相关寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
TCON	定时器控制寄存器	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
TMOD	定时器模式寄存器	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000,0000
TL0	定时器 0 低 8 位寄存器	8AH									0000,0000
TL1	定时器 1 低 8 位寄存器	8BH									0000,0000
TH0	定时器 0 高 8 位寄存器	8CH									0000,0000
TH1	定时器 1 高 8 位寄存器	8DH									0000,0000
AUXR	辅助寄存器 1	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2	0000,0001

11.2 定时器 0/1

11.2.1 定时器 0/1 控制寄存器（TCON）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1: T1溢出中断标志。T1被允许计数以后，从初值开始加1计数。当产生溢出时由硬件将TF1位置“1”，并向CPU请求中断，一直保持到CPU响应中断时，才由硬件清“0”（也可由查询软件清“0”）。

TR1: 定时器T1的运行控制位。该位由软件置位和清零。当GATE（TMOD.7）=0，TR1=1时就允许T1开始计数，TR1=0时禁止T1计数。当GATE（TMOD.7）=1，TR1=1且INT1输入高电平时，才允许T1计数。

TF0: T0溢出中断标志。T0被允许计数以后，从初值开始加1计数，当产生溢出时，由硬件置“1”TF0，向CPU请求中断，一直保持CPU响应该中断时，才由硬件清0（也可由查询软件清0）。

TR0: 定时器T0的运行控制位。该位由软件置位和清零。当GATE（TMOD.3）=0，TR0=1时就允许T0开始计数，TR0=0时禁止T0计数。当GATE（TMOD.3）=1，TR0=1且INT0输入高电平时，才允许T0计数，TR0=0时禁止T0计数。

IE1: 外部中断1请求源（INT1/P3.3）标志。IE1=1，外部中断向CPU请求中断，当CPU响应该中断时由硬件清“0”IE1。

IT1: 外部中断源1触发控制位。IT1=0，上升沿或下降沿均可触发外部中断1。IT1=1，外部中断1程控为下降沿触发方式。

IE0: 外部中断0请求源（INT0/P3.2）标志。IE0=1外部中断0向CPU请求中断，当CPU响应外部中断时，由硬件清“0”IE0（边沿触发方式）。

IT0: 外部中断源0触发控制位。IT0=0，上升沿或下降沿均可触发外部中断0。IT0=1，外部中断0程控为下降沿触发方式。

11.2.2 定时器 0/1 模式寄存器（TMOD）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
TMOD	89H	T1_GATE	T1_C/T	T1_M1	T1_M0	T0_GATE	T0_C/T	T0_M1	T0_M0

T1_GATE: 控制定时器1，置1时只有在INT1脚为高及TR1控制位置1时才可打开定时器/计数器1。

T0_GATE: 控制定时器0，置1时只有在INT0脚为高及TR0控制位置1时才可打开定时器/计数器0。

T1_C/T: 控制定时器1用作定时器或计数器，清0则用作定时器（对内部系统时钟进行计数），置1用作计数器（对引脚T1/P3.5外部脉冲进行计数）。

T0_C/T: 控制定时器0用作定时器或计数器，清0则用作定时器（对内部系统时钟进行计数），置1用作计数器（对引脚T0/P3.4外部脉冲进行计数）。

T1_M1/T1_M0: 定时器定时器/计数器1模式选择

T1_M1	T1_M0	定时器/计数器1工作模式
0	0	16位自动重载模式 当[TH1,TL1]中的16位计数值溢出时，系统会自动将内部16位

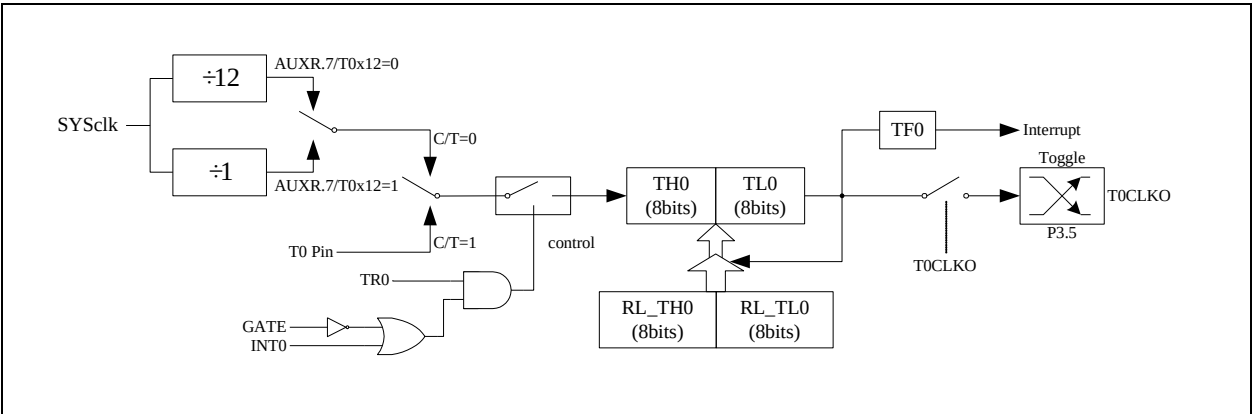
		重载寄存器中的重载值装入[TH1,TL1]中。
0	1	16位不自动重载模式 当[TH1,TL1]中的16位计数值溢出时，定时器1将从0开始计数
1	0	8位自动重载模式 当TL1中的8位计数值溢出时，系统会自动将TH1中的重载值装入TL1中。
1	1	T1停止工作

T0_M1/T0_M0：定时器/计数器0模式选择

T0_M1	T0_M0	定时器/计数器0工作模式
0	0	16位自动重载模式 当[TH0,TL0]中的16位计数值溢出时，系统会自动将内部16位重载寄存器中的重载值装入[TH0,TL0]中。
0	1	16位不自动重载模式 当[TH0,TL0]中的16位计数值溢出时，定时器0将从0开始计数
1	0	8位自动重载模式 当TL0中的8位计数值溢出时，系统会自动将TH0中的重载值装入TL0中。
1	1	不可屏蔽中断的16位自动重载模式 与模式0相同，不可屏蔽中断，中断优先级最高，高于其他所有中断的优先级，并且不可关闭，可用作操作系统的系统节拍定时器，或者系统监控定时器。

11.2.3 定时器 0 模式 0

此模式下定时器/计数器 0 作为可自动重装载的 16 位计数器，如下图所示：



定时器/计数器 0 的模式 0：16 位自动重载模式

当 GATE=0 (TMOD.3) 时，如 TR0=1，则定时器计数。GATE=1 时，允许由外部输入 INTO 控制定时器 0，这样可实现脉宽测量。TR0 为 TCON 寄存器内的控制位，TCON 寄存器各位的具体功能描述见上节 TCON 寄存器的介绍。

当 $C/T=0$ 时，多路开关连接到系统时钟的分频输出，T0 对内部系统时钟计数，T0 工作在定时方式。当 $C/T=1$ 时，多路开关连接到外部脉冲输入 P3.4/T0，即 T0 工作在计数方式。

STC 单片机的定时器 0 有两种计数速率：一种是 12T 模式，每 12 个时钟加 1，与传统 8051 单片机相同；另外一种 1T 模式，每个时钟加 1，速度是传统 8051 单片机的 12 倍。T0 的速率由特殊功能寄存器 AUXR 中的 T0x12 决定，如果 T0x12=0，T0 则工作在 12T 模式；如果 T0x12=1，T0 则工作在 1T 模式。

定时器 0 有两个隐藏的寄存器 RL_TH0 和 RL_TL0。RL_TH0 与 TH0 共有同一个地址，RL_TL0 与 TL0 共有同一个地址。当 TR0=0 即定时器/计数器 0 被禁止工作时，对 TL0 写入的内容会同时写入 RL_TL0，对 TH0 写入的内容也会同时写入 RL_TH0。当 TR0=1 即定时器/计数器 0 被允许工作时，对 TL0 写入内容，实际上不是写入当前寄存器 TL0 中，而是写入隐藏的寄存器 RL_TL0 中，对 TH0 写入内容，实际上也不是写入当前寄存器 TH0 中，而是写入隐藏的寄存器 RL_TH0，这样可以巧妙地实现 16 位重载定时器。当读 TH0 和 TL0 的内容时，所读的内容就是 TH0 和 TL0 的内容，而不是 RL_TH0 和 RL_TL0 的内容。

当定时器 0 工作在模式 0 (TMOD[1:0]/[M1,M0]=00B) 时，[TH0,TL0] 的溢出不仅置位 TF0，而且会自动将 [RL_TH0,RL_TL0] 的内容重新装入 [TH0,TL0]。

当 T0CLKO/INT_CLKO.0=1 时，P3.5/T1 管脚配置为定时器 0 的时钟输出 T0CLKO。输出时钟频率为 T0 溢出率/2。

如果 $C/T=0$ ，定时器/计数器 T0 对内部系统时钟计数，则：

T0 工作在 1T 模式 (AUXR.7/T0x12=1) 时的输出时钟频率 = $(\text{SYSclk}) / (65536 - [\text{RL_TH0}, \text{RL_TL0}]) / 2$

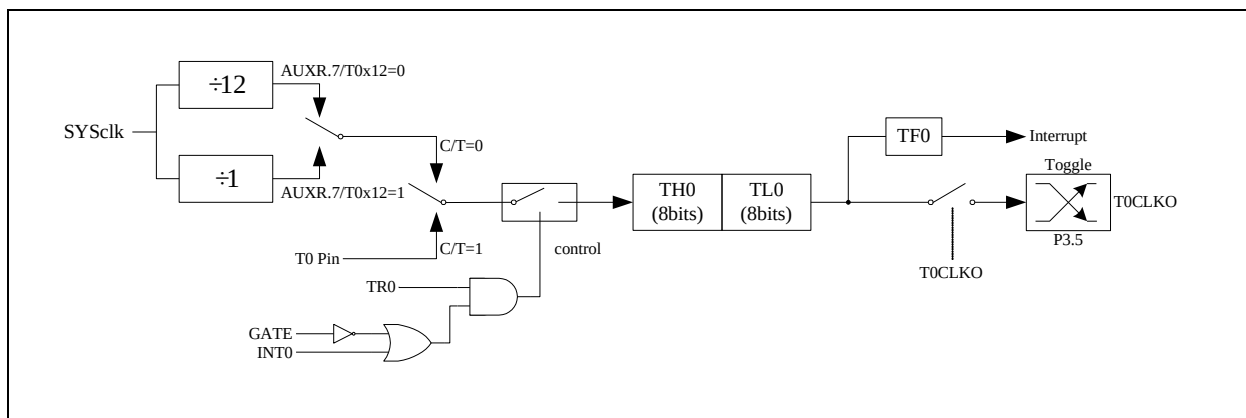
T0 工作在 12T 模式 (AUXR.7/T0x12=0) 时的输出时钟频率 = $(\text{SYSclk}) / 12 / (65536 - [\text{RL_TH0}, \text{RL_TL0}]) / 2$

如果 $C/T=1$ ，定时器/计数器 T0 是对外部脉冲输入(P3.4/T0)计数，则：

输出时钟频率 = $(\text{T0_Pin_CLK}) / (65536 - [\text{RL_TH0}, \text{RL_TL0}]) / 2$

11.2.4 定时器 0 模式 1

此模式下定时器/计数器 0 工作在 16 位不可重载模式，如下图所示



定时器/计数器 0 的模式 1：16 位不可重载模式

此模式下，定时器/计数器 0 配置为 16 位不可重载模式，由 TL0 的 8 位和 TH0 的 8 位所构成。TL0 的 8 位溢出向 TH0 进位，TH0 计数溢出置位 TCON 中的溢出标志位 TF0。

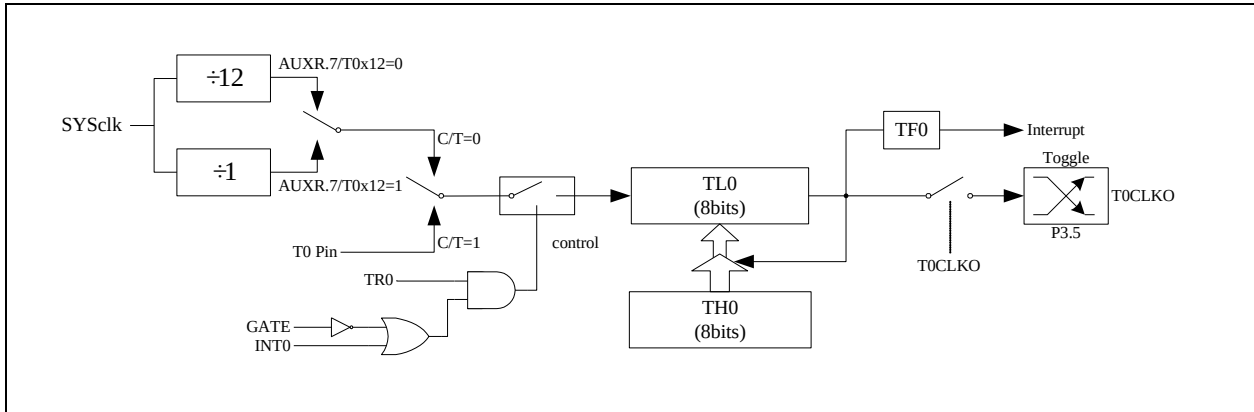
当 GATE=0(TM0D.3)时，如 TR0=1，则定时器计数。GATE=1 时，允许由外部输入 INT0 控制定时器 0，这样可实现脉宽测量。TR0 为 TCON 寄存器内的控制位，TCON 寄存器各位的具体功能描述见上节 TCON 寄存器的介绍。

当 $C/T=0$ 时，多路开关连接到系统时钟的分频输出，T0 对内部系统时钟计数，T0 工作在定时方式。当 $C/T=1$ 时，多路开关连接到外部脉冲输入 P3.4/T0，即 T0 工作在计数方式。

STC 单片机的定时器 0 有两种计数速率：一种是 12T 模式，每 12 个时钟加 1，与传统 8051 单片机相同；另外一种 1T 模式，每个时钟加 1，速度是传统 8051 单片机的 12 倍。T0 的速率由特殊功能寄存器 AUXR 中的 T0x12 决定，如果 $T0x12=0$ ，T0 则工作在 12T 模式；如果 $T0x12=1$ ，T0 则工作在 1T 模式。

11.2.5 定时器 0 模式 2

此模式下定时器/计数器 0 作为可自动重载的 8 位计数器，如下图所示：



定时器/计数器 0 的模式 2：8 位自动重载模式

TL0 的溢出不仅置位 TF0，而且将 TH0 的内容重新装入 TL0，TH0 内容由软件预置，重装时 TH0 内容不变。

当 $T0CLKO/INT_CLKO.0=1$ 时，P3.5/T1 管脚配置为定时器 0 的时钟输出 T0CLKO。输出时钟频率为 T0 溢出率/2。

如果 $C/T=0$ ，定时器/计数器 T0 对内部系统时钟计数，则：

T0 工作在 1T 模式 ($AUXR.7/T0x12=1$) 时的输出时钟频率 = $(SYSclk)/(256-TH0)/2$

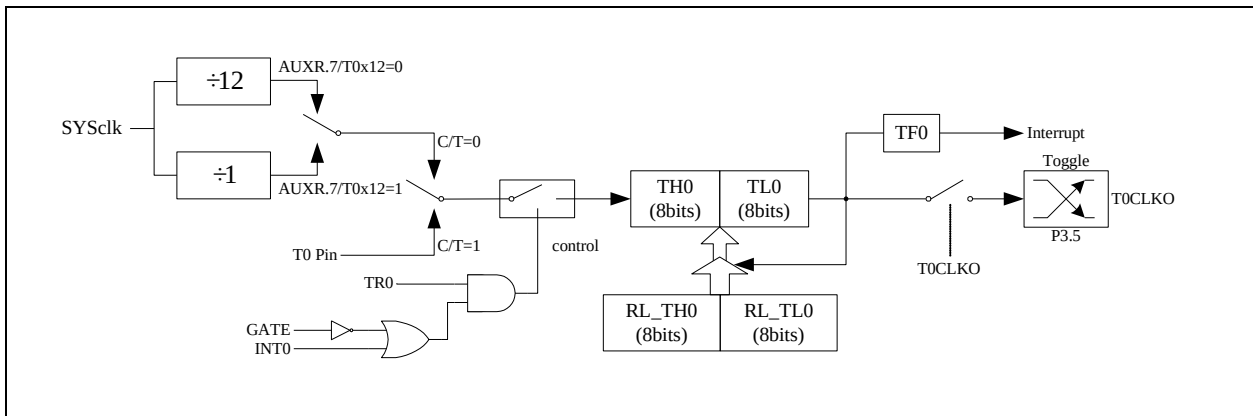
T0 工作在 12T 模式 ($AUXR.7/T0x12=0$) 时的输出时钟频率 = $(SYSclk)/12/(256-TH0)/2$

如果 $C/T=1$ ，定时器/计数器 T0 是对外部脉冲输入(P3.4/T0)计数，则：

输出时钟频率 = $(T0_Pin_CLK) / (256-TH0)/2$

11.2.6 定时器 0 模式 3

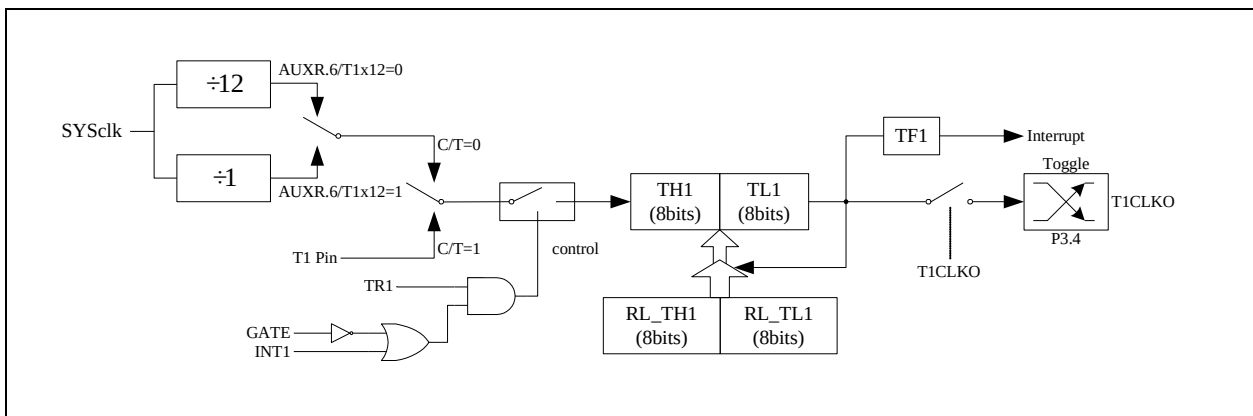
对定时器/计数器 0，其工作模式模式 3 与工作模式 0 是一样的（下图定时器模式 3 的原理图，与工作模式 0 是一样的）。唯一不同的是：当定时器/计数器 0 工作在模式 3 时，只需允许 $ET0/IE.1$ (定时器/计数器 0 中断允许位)，不需要允许 $EA/IE.7$ (总中断使能位) 就能打开定时器/计数器 0 的中断，此模式下的定时器/计数器 0 中断与总中断使能位 EA 无关，一旦工作在模式 3 下的定时器/计数器 0 中断被打开 ($ET0=1$)，那么该中断是不可屏蔽的，该中断的优先级是最高的，即该中断不能被任何中断所打断，而且该中断打开后既不受 $EA/IE.7$ 控制也不再受 $ET0$ 控制，当 $EA=0$ 或 $ET0=0$ 时都不能屏蔽此中断。故将此模式称为不可屏蔽中断的 16 位自动重载模式。



定时器/计数器 0 的模式 3：不可屏蔽中断的 16 位自动重载模式

11.2.7 定时器 1 模式 0

此模式下定时器/计数器 1 作为可自动重载的 16 位计数器，如下图所示：



定时器/计数器 1 的模式 0：16 位自动重载模式

当 $GATE=0$ (TMOD.7) 时，如 $TR1=1$ ，则定时器计数。 $GATE=1$ 时，允许由外部输入 $INT1$ 控制定时器 1，这样可实现脉宽测量。 $TR1$ 为 TCON 寄存器内的控制位，TCON 寄存器各位的具体功能描述见上节 TCON 寄存器的介绍。

当 $C/T=0$ 时，多路开关连接到系统时钟的分频输出，T1 对内部系统时钟计数，T1 工作在定时方式。当 $C/T=1$ 时，多路开关连接到外部脉冲输入 P3.5/T1，即 T1 工作在计数方式。

STC 单片机的定时器 1 有两种计数速率：一种是 12T 模式，每 12 个时钟加 1，与传统 8051 单片机相同；另外一种 1T 模式，每个时钟加 1，速度是传统 8051 单片机的 12 倍。T1 的速率由特殊功能寄存器 AUXR 中的 $T1x12$ 决定，如果 $T1x12=0$ ，T1 则工作在 12T 模式；如果 $T1x12=1$ ，T1 则工作在 1T 模式。

定时器 1 有两个隐藏的寄存器 RL_TH1 和 RL_TL1 。 RL_TH1 与 $TH1$ 共有同一个地址， RL_TL1 与 $TL1$ 共有同一个地址。当 $TR1=0$ 即定时器/计数器 1 被禁止工作时，对 $TL1$ 写入的内容会同时写入 RL_TL1 ，对 $TH1$ 写入的内容也会同时写入 RL_TH1 。当 $TR1=1$ 即定时器/计数器 1 被允许工作时，对 $TL1$ 写入内容，实际上不是写入当前寄存器 $TL1$ 中，而是写入隐藏的寄存器 RL_TL1 中，对 $TH1$ 写入内容，实际上也不是写入当前寄存器 $TH1$ 中，而是写入隐藏的寄存器 RL_TH1 ，这样可以巧妙地实现 16 位重载定时器。当读 $TH1$ 和 $TL1$ 的内容时，所读的内容就是 $TH1$ 和 $TL1$ 的内容，而不是 RL_TH1 和 RL_TL1 的内容。

当定时器 1 工作在模式 1 (TMOD[5:4]/[M1,M0]=00B) 时，[TH1,TL1]的溢出不仅置位 TF1，而且会自动

将[RL_TH1,RL_TL1]的内容重新装入[TH1,TL1]。

当 T1CLKO/INT_CLKO.1=1 时，P3.4/T0 管脚配置为定时器 1 的时钟输出 T1CLKO。输出时钟频率为 T1 溢出率/2。

如果 C/T=0，定时器/计数器 T1 对内部系统时钟计数，则：

T1 工作在 1T 模式（AUXR.6/T1x12=1）时的输出时钟频率 = (SYSclk)/(65536-[RL_TH1, RL_TL1])/2

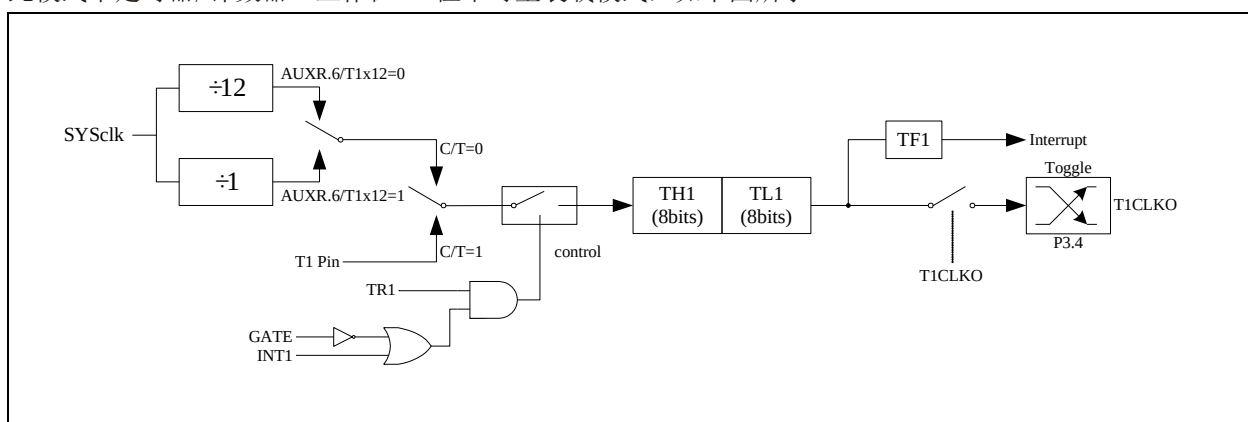
T1 工作在 12T 模式（AUXR.6/T1x12=0）时的输出时钟频率 = (SYSclk)/12/(65536-[RL_TH1, RL_TL1])/2

如果 C/T=1，定时器/计数器 T1 是对外部脉冲输入(P3.5/T1)计数，则：

输出时钟频率 = (T1_Pin_CLK) / (65536-[RL_TH1, RL_TL1])/2

11.2.8 定时器 1 模式 1

此模式下定时器/计数器 1 工作在 16 位不可重装模式，如下图所示



定时器/计数器 1 的模式 1：16 位不可重装模式

此模式下，定时器/计数器 1 配置为 16 位不可重装模式，由 TL1 的 8 位和 TH1 的 8 位所构成。TL1 的 8 位溢出向 TH1 进位，TH1 计数溢出置位 TCON 中的溢出标志位 TF1。

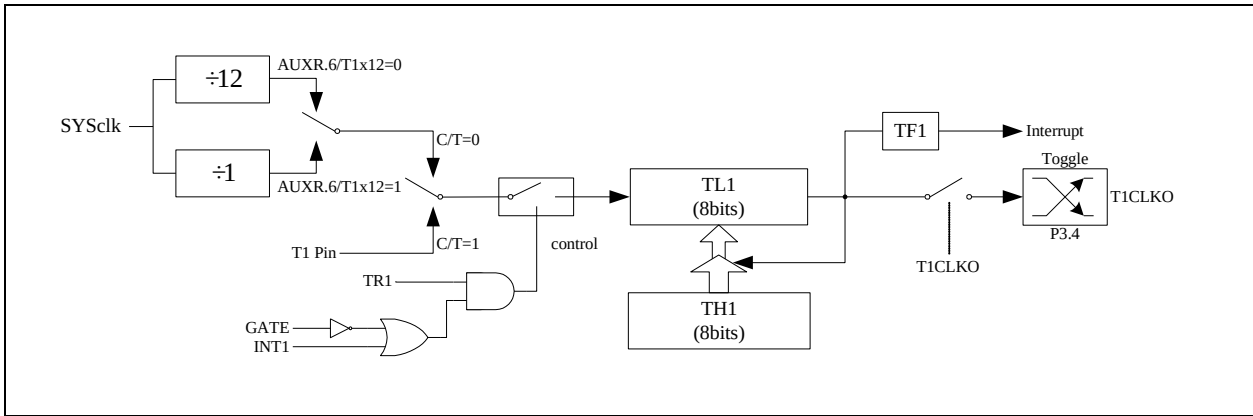
当 GATE=0(TMOD.7)时，如 TR1=1，则定时器计数。GATE=1 时，允许由外部输入 INT1 控制定时器 1，这样可实现脉宽测量。TR1 为 TCON 寄存器内的控制位，TCON 寄存器各位的具体功能描述见上节 TCON 寄存器的介绍。

当 C/T=0 时，多路开关连接到系统时钟的分频输出，T1 对内部系统时钟计数，T1 工作在定时方式。当 C/T=1 时，多路开关连接到外部脉冲输入 P3.5/T1，即 T1 工作在计数方式。

STC 单片机的定时器 1 有两种计数速率：一种是 12T 模式，每 12 个时钟加 1，与传统 8051 单片机相同；另外一种 1T 模式，每个时钟加 1，速度是传统 8051 单片机的 12 倍。T1 的速率由特殊功能寄存器 AUXR 中的 T1x12 决定，如果 T1x12=0，T1 则工作在 12T 模式；如果 T1x12=1，T1 则工作在 1T 模式

11.2.9 定时器 1 模式 2

此模式下定时器/计数器 1 作为可自动重装模式的 8 位计数器，如下图所示：



定时器/计数器 1 的模式 2：8 位自动重载模式

TL1 的溢出不仅置位 TF1，而且将 TH1 的内容重新装入 TL1，TH1 内容由软件预置，重装时 TH1 内容不变。

当 T1CLKO/INT_CLKO.1=1 时，P3.4/T0 管脚配置为定时器 1 的时钟输出 T1CLKO。输出时钟频率为 T1 溢出率/2。

如果 C/T=0，定时器/计数器 T1 对内部系统时钟计数，则：

T1 工作在 1T 模式（AUXR.6/T1x12=1）时的输出时钟频率 = (SYSclk)/(256-TH1)/2

T1 工作在 12T 模式（AUXR.6/T1x12=0）时的输出时钟频率 = (SYSclk)/12/(256-TH1)/2

如果 C/T=1，定时器/计数器 T1 是对外部脉冲输入(P3.5/T1)计数，则：

输出时钟频率 = (T1_Pin_CLK) / (256-TH1)/2

11.2.10 定时器 0 计数寄存器（TL0，TH0）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
TL0	8AH								
TH0	8CH								

当定时器/计数器0工作在16位模式（模式0、模式1、模式3）时，TL0和TH0组合成为一个16位寄存器，

TL0为低字节，TH0为高字节。若为8位模式（模式2）时，TL0和TH0为两个独立的8位寄存器。

11.2.11 定时器 1 计数寄存器（TL1，TH1）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
TL1	8BH								
TH1	8DH								

当定时器/计数器1工作在16位模式（模式0、模式1）时，TL1和TH1组合成为一个16位寄存器，TL1为低

字节，TH1为高字节。若为8位模式（模式2）时，TL1和TH1为两个独立的8位寄存器。

11.2.12 辅助寄存器 1（AUXR）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
AUXR	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

T0x12：定时器0速度控制位

0：12T 模式，即 CPU 时钟 12 分频（FOSC/12）

1: 1T 模式，即 CPU 时钟不分频分频（FOSC/1）

T1x12: 定时器1速度控制位

0: 12T 模式，即 CPU 时钟 12 分频（FOSC/12）

1: 1T 模式，即 CPU 时钟不分频分频（FOSC/1）

11.2.13 中断与时钟输出控制寄存器（INT_CLKO）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
INT_CLKO	8FH						T2CLKO	T1CLKO	T0CLKO

T0CLKO: 定时器0时钟输出控制

0: 关闭时钟输出

1: 使能 P3.5 口的是定时器 0 时钟输出功能

当定时器 0 计数发生溢出时，P3.5 口的电平自动发生翻转。

T1CLKO: 定时器1时钟输出控制

0: 关闭时钟输出

1: 使能 P3.4 口的是定时器 1 时钟输出功能

当定时器 1 计数发生溢出时，P3.4 口的电平自动发生翻转。

12 串口通信

单片机具有 1 个全双工异步串行通信接口。串行口由数据缓冲器、一个移位寄存器、一个串行控制寄存器和一个波特率发生器等组成。每个串行口的数据缓冲器由 2 个互相独立的接收、发送缓冲器构成，可以同时发送和接收数据。

单片机的串口有 4 种工作方式，其中两种方式的波特率是可变的，另两种是固定的，以供不同应用场合选用。用户可用软件设置不同的波特率和选择不同的工作方式。主机可通过查询或中断方式对接收/发送进行程序处理，使用十分灵活。

12.1 串口相关寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
SCON	串口 1 控制寄存器	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
SBUF	串口 1 数据寄存器	99H									0000,0000
PCON	电源控制寄存器	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
AUXR	辅助寄存器 1	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2	0000,0001
SADDR	串口 1 从机地址寄存器	A9H									0000,0000
SADEN	串口 1 从机地址屏蔽寄存器	B9H									0000,0000

12.2 串口 1

12.2.1 串口 1 控制寄存器（SCON）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SCON	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

SM0/FE: 当PCON寄存器中的SMOD0位为1时，该位为帧错误检测标志位。当UART在接收过程中检测到一个无效停止位时，通过UART接收器将该位置1，必须由软件清零。当PCON寄存器中的SMOD0位为0时，该位和SM1一起指定串口1的通信工作模式，如下表所示：

SM0	SM1	串口1工作模式	功能说明
0	0	模式0	同步移位串行方式
0	1	模式1	可变波特率8位数据方式
1	0	模式2	固定波特率9位数据方式
1	1	模式3	可变波特率9位数据方式

SM2: 允许模式 2 或模式 3 多机通信控制位。当串口 1 使用模式 2 或模式 3 时，如果 SM2 位为 1 且 REN 位为 1，则接收机处于地址帧筛选状态。此时可以利用接收到的第 9 位（即 RB8）来筛选地址帧，若 RB8=1，说明该帧是地址帧，地址信息可以进入 SBUF，并使 RI 为 1，进而在中断服务程序中再进行地址号比较；若 RB8=0，说明该帧不是地址帧，应丢掉且保持 RI=0。在模式 2 或模式 3 中，如果 SM2 位为 0 且 REN 位为 1，接收机处于地址帧筛选被禁止状态，不论收到的 RB8 为 0 或 1，均可使接收到的信息进入 SBUF，并使 RI=1，此时 RB8 通常为校验位。模式 1 和模式 0 为非多机通信方式，在这两种方式时，SM2 应设置为 0。

REN: 允许/禁止串口接收控制位

0: 禁止串口接收数据

1: 允许串口接收数据

TB8: 当串口 1 使用模式 2 或模式 3 时，TB8 为要发送的第 9 位数据，按需要由软件置位或清 0。在模式 0 和模式 1 中，该位不用。

RB8: 当串口 1 使用模式 2 或模式 3 时，RB8 为接收到的第 9 位数据，一般用作校验位或者地址帧/数据帧标志位。在模式 0 和模式 1 中，该位不用。

TI: 串口 1 发送中断请求标志位。在模式 0 中，当串口发送数据第 8 位结束时，由硬件自动将 TI 置 1，向主机请求中断，响应中断后 TI 必须用软件清零。在其他模式中，则在停止位开始发送时由硬件自动将 TI 置 1，向 CPU 发请求中断，响应中断后 TI 必须用软件清零。

RI: 串口 1 接收中断请求标志位。在模式 0 中，当串口接收第 8 位数据结束时，由硬件自动将 RI 置 1，向主机请求中断，响应中断后 RI 必须用软件清零。在其他模式中，串行接收到停止位的中间时刻由硬件自动将 RI 置 1，向 CPU 发中断申请，响应中断后 RI 必须由软件清零。

12.2.2 串口 1 数据寄存器（SBUF）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SBUF	99H								

SBUF: 串口 1 数据接收/发送缓冲区。SBUF 实际是 2 个缓冲器，读缓冲器和写缓冲器，两个操作分别对应两个不同的寄存器，1 个是只写寄存器（写缓冲器），1 个是只读寄存器（读缓冲器）。对 SBUF 进行读操作，实际是读取串口接收缓冲区，对 SBUF 进行写操作则是触发串口开始发送数据。

12.2.3 电源管理寄存器（PCON）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL

SMOD：串口 1 波特率控制位

0：串口 1 的各个模式的波特率都不加倍

1：串口 1 模式 1（使用模式 2 的定时器 1 作为波特率发生器时有效）、模式 2、模式 3（使用模式 2 的定时器 1 作为波特率发生器时有效）的波特率加倍

SMOD0：帧错误检测控制位

0：无帧错检测功能

1：使能帧错误检测功能。此时 SCON 的 SM0/FE 为 FE 功能，即为帧错误检测标志位。

12.2.4 辅助寄存器 1（AUXR）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
AUXR	8EH	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

UART_M0x6：串口 1 模式 0 的通讯速度控制

0：串口 1 模式 0 的波特率不加倍，固定为 $F_{osc}/12$

1：串口 1 模式 0 的波特率 6 倍速，即固定为 $F_{osc}/12 \times 6 = F_{osc}/2$

S1ST2：串口 1 波特率发生器选择位

0：选择定时器 1 作为波特率发生器

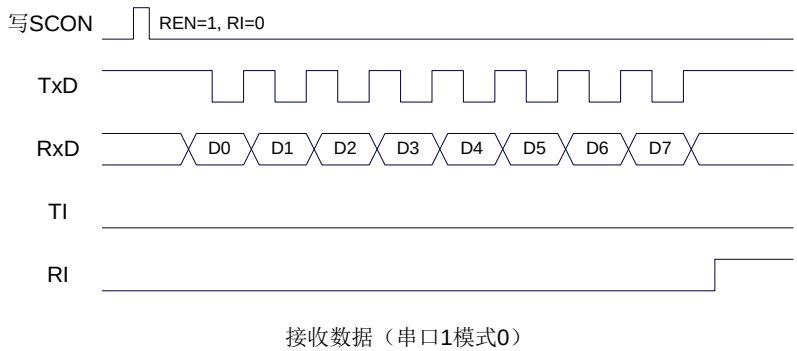
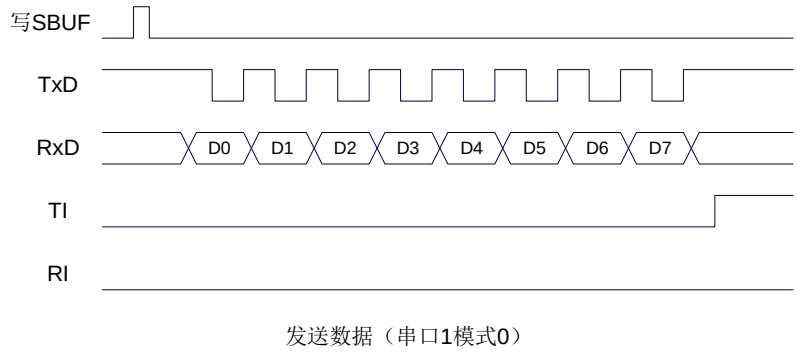
1：选择定时器 2 作为波特率发生器

12.2.5 串口 1 模式 0，模式 0 波特率计算公式

当串口 1 选择工作模式为模式 0 时，串行通信接口工作在同步移位寄存器模式，当串行口模式 0 的通信速度设置位 UART_M0x6 为 0 时，其波特率固定为系统时钟的 12 分频（ $SYSClk/12$ ）；当设置 UART_M0x6 为 1 时，其波特率固定为系统时钟频率的 2 分频（ $SYSClk/2$ ）。RxD 为串行通讯的数据口，TxD 为同步移位脉冲输出脚，发送、接收的是 8 位数据，低位在先。

模式 0 的发送过程：当主机执行将数据写入发送缓冲器 SBUF 指令时启动发送，串行口即将 8 位数据以 $SYSClk/12$ 或 $SYSClk/2$ （由 UART_M0x6 确定是 12 分频还是 2 分频）的波特率从 RxD 管脚输出(从低位到高位)，发送完中断标志 TI 置 1，TxD 管脚输出同步移位脉冲信号。当写信号有效后，相隔一个时钟，发送控制端 SEND 有效(高电平)，允许 RxD 发送数据，同时允许 TxD 输出同步移位脉冲。一帧(8 位)数据发送完毕时，各控制端均恢复原状态，只有 TI 保持高电平，呈中断申请状态。在再次发送数据前，必须用软件将 TI 清 0。

模式 0 的接收过程：首先将接收中断请求标志 RI 清零并置位允许接收控制位 REN 时启动模式 0 接收过程。启动接收过程后，RxD 为串行数据输入端，TxD 为同步脉冲输出端。串行接收的波特率为 $SYSClk/12$ 或 $SYSClk/2$ （由 UART_M0x6 确定是 12 分频还是 2 分频）。当接收完成一帧数据（8 位）后，控制信号复位，中断标志 RI 被置 1，呈中断申请状态。当再次接收时，必须通过软件将 RI 清 0



工作于模式 0 时，必须清 0 多机通信控制位 SM2，使之不影响 TB8 位和 RB8 位。由于波特率固定为 SYSclk/12 或 SYSclk/2，无需定时器提供，直接由单片机的时钟作为同步移位脉冲。

串口 1 模式 0 的波特率计算公式如下表所示（SYSclk 为系统工作频率）：

UART_M0x6	波特率计算公式
0	$\text{波特率} = \frac{\text{SYSclk}}{12}$
1	$\text{波特率} = \frac{\text{SYSclk}}{2}$

12.2.6 串口 1 模式 1，模式 1 波特率计算公式

当软件设置 SCON 的 SM0、SM1 为“01”时，串行口 1 则以模式 1 进行工作。此模式为 8 位 UART 格式，一帧信息为 10 位：1 位起始位，8 位数据位（低位在先）和 1 位停止位。波特率可变，即可根据需要进行设置波特率。TxD 为数据发送口，RxD 为数据接收口，串行口全双工接受/发送。

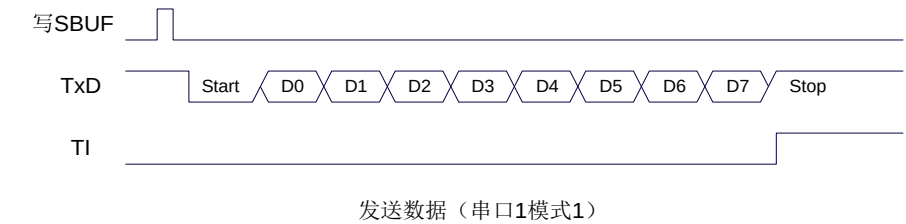
模式 1 的发送过程：串行通信模式发送时，数据由串行发送端 TxD 输出。当主机执行一条写 SBUF 的指令就启动串行通信的发送，写“SBUF”信号还把“1”装入发送移位寄存器的第 9 位，并通知 TX 控制单元开始发送。移位寄存器将数据不断右移送 TxD 端口发送，在数据的左边不断移入“0”作补充。当数据的最高位移到移位寄存器的输出位置，紧跟其后的是第 9 位“1”，在它的左边各位全为“0”，这个状态条件，使 TX 控制单元作最后一次移位输出，然后使允许发送信号“SEND”失效，完成一帧信息的发送，并置位中断请求位 TI，即 TI=1，向主机请求中断处理。

模式 1 的接收过程：当软件置位接收允许标志位 REN，即 REN=1 时，接收器便对 RxD 端口的信号进行检测，当检

测到 RxD 端口发送从“1”→“0”的下降沿跳变时就启动接收器准备接收数据，并立即复位波特率发生器的接收计数器，将 1FFH 装入移位寄存器。接收的数据从接收移位寄存器的右边移入，已装入的 1FFH 向左边移出，当起始位“0”移到移位寄存器的最左边时，使 RX 控制器作最后一次移位，完成一帧的接收。若同时满足以下两个条件：

- RI=0;
- SM2=0 或接收到的停止位为 1。

则接收到的数据有效，实现装载入 SBUF，停止位进入 RB8，RI 标志位被置 1，向主机请求中断，若上述两条件不能同时满足，则接收到的数据作废并丢失，无论条件满足与否，接收器重又检测 RxD 端口上的“1”→“0”的跳变，继续下一帧的接收。接收有效，在响应中断后，RI 标志位必须由软件清 0。通常情况下，串行通信工作于模式 1 时，SM2 设置为“0”。



串口 1 的波特率是可变的，其波特率可由定时器 1 或者定时器 2 产生。当定时器采用 1T 模式时（12 倍速），相应的波特率的速度也会相应提高 12 倍。

串口 1 模式 1 的波特率计算公式如下表所示：（SYSclk 为系统工作频率）

选择定时器	定时器速度	重装值计算公式	波特率
定时器2	1T	定时器2重装值 = $65536 - \frac{\text{SYSclk}}{4 \times \text{波特率}}$	波特率 = $\frac{\text{SYSclk}}{4 \times (65536 - \text{定时器重装数})}$
	12T	定时器2重装值 = $65536 - \frac{\text{SYSclk}}{12 \times 4 \times \text{波特率}}$	波特率 = $\frac{\text{SYSclk}}{12 \times 4 \times (65536 - \text{定时器重装数})}$
定时器1 模式0	1T	定时器1重装值 = $65536 - \frac{\text{SYSclk}}{4 \times \text{波特率}}$	波特率 = $\frac{\text{SYSclk}}{4 \times (65536 - \text{定时器重装数})}$
	12T	定时器1重装值 = $65536 - \frac{\text{SYSclk}}{12 \times 4 \times \text{波特率}}$	波特率 = $\frac{\text{SYSclk}}{12 \times 4 \times (65536 - \text{定时器重装数})}$
定时器1 模式2	1T	定时器1重装值 = $256 - \frac{2^{\text{SMOD}} \times \text{SYSclk}}{32 \times \text{波特率}}$	波特率 = $\frac{2^{\text{SMOD}} \times \text{SYSclk}}{32 \times (256 - \text{定时器重装数})}$
	12T	定时器1重装值 = $256 - \frac{2^{\text{SMOD}} \times \text{SYSclk}}{12 \times 32 \times \text{波特率}}$	波特率 = $\frac{2^{\text{SMOD}} \times \text{SYSclk}}{12 \times 32 \times (256 - \text{定时器重装数})}$

下面为常用频率与常用波特率所对应定时器的重载值

频率 (MHz)	波特率	定时器 2		定时器 1 模式 0		定时器 1 模式 2			
		1T 模式	12T 模式	1T 模式	12T 模式	SMOD=1		SMOD=0	
						1T 模式	12T 模式	1T 模式	12T 模式
11.0592	115200	FFE8H	FFFEH	FFE8H	FFFEH	FAH	-	FDH	-
	57600	FFD0H	FFFCH	FFD0H	FFFCH	F4H	FFH	FAH	-
	38400	FFB8H	FFFAH	FFB8H	FFFAH	EEH	-	F7H	-
	19200	FF70H	FFF4H	FF70H	FFF4H	DCH	FDH	EEH	-
	9600	FEE0H	FFE8H	FEE0H	FFE8H	B8H	FAH	DCH	FDH
18.432	115200	FFD8H	-	FFD8H	-	F6H	-	FBH	-
	57600	FFB0H	-	FFB0H	-	ECH	-	F6H	-
	38400	FF88H	FFF6H	FF88H	FFF6H	E2H	-	F1H	-
	19200	FF10H	FFECH	FF10H	FFECH	C4H	FBH	E2H	-
	9600	FE20H	FFD8H	FE20H	FFD8H	88H	F6H	C4H	FBH
22.1184	115200	FFD0H	FFFCH	FFD0H	FFFCH	F4H	FFH	FAH	-
	57600	FFA0H	FFF8H	FFA0H	FFF8H	E8H	FEH	F4H	FFH
	38400	FF70H	FFF4H	FF70H	FFF4H	DCH	FDH	EEH	-
	19200	FEE0H	FFE8H	FEE0H	FFE8H	B8H	FAH	DCH	FDH
	9600	FDC0H	FFD0H	FDC0H	FFD0H	70H	F4H	B8H	FAH

12.2.7 串口 1 模式 2，模式 2 波特率计算公式

当 SM0、SM1 两位为 10 时，串行口 1 工作在模式 2。串行口 1 工作模式 2 为 9 位数据异步通信 UART 模式，其帧的信息由 11 位组成：1 位起始位，8 位数据位（低位在先），1 位可编程位（第 9 位数据）和 1 位停止位。发送时可编程位（第 9 位数据）由 SCON 中的 TB8 提供，可软件设置为 1 或 0，或者可将 PSW 中的奇/偶校验位 P 值装入 TB8（TB8 既可作为多机通信中的地址数据标志位，又可作为数据的奇偶校验位）。接收时第 9 位数据装入 SCON 的 RB8。TxD 为发送端口，RxD 为接收端口，以全双工模式进行接收/发送。

模式 2 的波特率固定为系统时钟的 64 分频或 32 分频（取决于 PCON 中 SMOD 的值）

串口 1 模式 2 的波特率计算公式如下表所示（SYSclk 为系统工作频率）：

SMOD	波特率计算公式
0	$\text{波特率} = \frac{\text{SYSclk}}{64}$
1	$\text{波特率} = \frac{\text{SYSclk}}{32}$

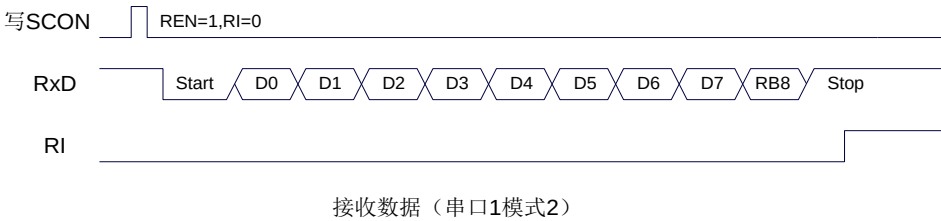
模式 2 和模式 1 相比，除波特率发生源略有不同，发送时由 TB8 提供给移位寄存器第 9 数据位不同外，其余功能结构均基本相同，其接收/发送操作过程及时序也基本相同。

当接收器接收完一帧信息后必须同时满足下列条件：

- RI=0
- SM2=0 或者 SM2=1 且接收到的第 9 数据位 RB8=1。

当上述两条件同时满足时，才将接收到的移位寄存器的数据装入 SBUF 和 RB8 中，RI 标志位被置 1，并向主机请求中断处理。如果上述条件有一个不满足，则刚接收到移位寄存器中的数据无效而丢失，也不置位 RI。无论上述条件满足与否，接收器又重新开始检测 RxD 输入端口的跳变信息，接收下一帧的输入信息。在模式 2 中，接收到的停止位与 SBUF、RB8 和 RI 无关。

通过软件对 SCON 中的 SM2、TB8 的设置以及通信协议的约定，为多机通信提供了方便。



12.2.8 串口 1 模式 3，模式 3 波特率计算公式

当 SM0、SM1 两位为 11 时，串行口 1 工作在模式 3。串行通信模式 3 为 9 位数据异步通信 UART 模式，其帧的信

息由 11 位组成：1 位起始位，8 位数据位（低位在先），1 位可编程位（第 9 位数据）和 1 位停止位。发送时可编程位（第 9 位数据）由 SCON 中的 TB8 提供，可软件设置为 1 或 0，或者可将 PSW 中的奇/偶校验位 P 值装入 TB8（TB8 既可作为多机通信中的地址数据标志位，又可作为数据的奇偶校验位）。接收时第 9 位数据装入 SCON 的 RB8。TxD 为发送端口，RxD 为接收端口，以全双工模式进行接收/发送。

模式 3 和模式 1 相比，除发送时由 TB8 提供给移位寄存器第 9 数据位不同外，其余功能结构均基本相同，其接收‘发送操作过程及时序也基本相同。

当接收器接收完一帧信息后必须同时满足下列条件：

- RI=0
- SM2=0 或者 SM2=1 且接收到的第 9 数据位 RB8=1。

当上述两条件同时满足时，才将接收到的移位寄存器的数据装入 SBUF 和 RB8 中，RI 标志位被置 1，并向主机请求中断处理。如果上述条件有一个不满足，则刚接收到移位寄存器中的数据无效而丢失，也不置位 RI。无论上述条件满足与否，接收器又重新开始检测 RxD 输入端口的跳变信息，接收下一帧的输入信息。在模式 3 中，接收到的停止位与 SBUF、RB8 和 RI 无关。

通过软件对 SCON 中的 SM2、TB8 的设置以及通信协议的约定，为多机通信提供了方便。



串口 1 模式 3 的波特率计算公式与模式 1 是完全相同的。请参考模式 1 的波特率计算公式。

12.2.9 自动地址识别

12.2.10 串口 1 从机地址控制寄存器（SADDR，SADEN）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SADDR	A9H								
SADEN	B9H								

SADDR：从机地址寄存器

SADEN：从机地址屏蔽位寄存器

自动地址识别功能典型应用在多机通讯领域，其主要原理是从机系统通过硬件比较功能来识别来自于主机串口数据流中的地址信息，通过寄存器 SADDR 和 SADEN 设置的本机的从机地址，硬件自动对从机地址进行过滤，当来自于主机的从机地址信息与本机所设置的从机地址相匹配时，硬件产生串口中断；否则硬件自动丢弃串口数据，而不产生中断。当众多处于空闲模式的从机链接在一起时，只有从机地址相匹配的从机才会从空闲模式唤醒，从而可以大大降低从机 MCU 的

功耗，即使从机处于正常工作状态也可避免不停地进入串口中断而降低系统执行效率。

要使用串口的自动地址识别功能，首先需要将参与通讯的 MCU 的串口通讯模式设置为模式 2 或者模式 3（通常都选择波特率可变的模式 3，因为模式 2 的波特率是固定的，不便于调节），并开启从机的 SCON 的 SM2 位。对于串口模式 2 或者模式 3 的 9 位数据位中，第 9 位数据（存放在 RB8 中）为地址/数据的标志位，当第 9 位数据为 1 时，表示前面的 8 位数据（存放在 SBUF 中）为地址信息。当 SM2 被设置为 1 时，从机 MCU 会自动过滤掉非地址数据（第 9 位为 0 的数据），而对 SBUF 中的地址数据（第 9 位为 1 的数据）自动与 SADDR 和 SADEN 所设置的本机地址进行比较，若地址相匹配，则会将 RI 置“1”，并产生中断，否则不予处理本次接收的串口数据。

从机地址的设置是通过 SADDR 和 SADEN 两个寄存器进行设置的。SADDR 为从机地址寄存器，里面存放本机的从机地址。SADEN 为从机地址屏蔽位寄存器，用于设置地址信息中的忽略位，设置方法如下：

例如

SADDR = 11001010

SADEN = 10000001

则匹配地址为 1xxxxxx0

即，只要主机送出的地址数据中的 bit0 为 0 且 bit7 为 1 就可以和本机地址相匹配

再例如

SADDR = 11001010

SADEN = 00001111

则匹配地址为 xxxx1010

即，只要主机送出的地址数据中的低 4 位为 1010 就可以和本机地址相匹配，而高 4 为被忽略，可以为任意值。

主机可以使用广播地址（FFH）同时选中所有的从机来进行通讯。

13 IAP/EEPROM/DATA-FLASH

单片机内部集成了大容量的 EEPROM。利用 ISP/IAP 技术可将内部 Data Flash 当 EEPROM，擦写次数在 10 万次以上。EEPROM 可分为若干个扇区，每个扇区包含 512 字节。

注意：EEPROM 的写操作只能将字节中的 1 写为 0，当需要将字节中的 0 写为 1，则必须执行扇区擦除操作。EEPROM 的读/写操作是以 1 字节为单位进行，而 EEPROM 擦除操作是以 1 扇区（512 字节）为单位进行，在执行擦除操作时，如果目标扇区中有需要保留的数据，则必须预先将这些数据读取到 RAM 中暂存，待擦除完成后再将保存的数据和需要更新的数据一起再写回 EEPROM/DATA-FLASH。

所以在使用时，建议同一次修改的数据放在同一个扇区，不是同一次修改的数据放在不同的扇区，不一定要用满。数据存储器的擦除操作是按扇区进行的（每扇区 512 字节）。

EEPROM 可用于保存一些需要在应用过程中修改并且掉电不丢失的参数数据。在用户程序中，可以对 EEPROM 进行字节读/字节编程/扇区擦除操作。在工作电压偏低时，建议不要进行 EEPROM 操作，以免发送数据丢失的情况。

13.1 EEPROM 相关的寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
IAP_DATA	IAP 数据寄存器	E2H									1111,1111
IAP_ADDRH	IAP 高地址寄存器	E3H									0000,0000
IAP_ADDRL	IAP 低地址寄存器	E4H									0000,0000
IAP_CMD	IAP 命令寄存器	E5H	-	-	-	-	-	-	CMD[1:0]	xxxx,xx00	
IAP_TRIG	IAP 触发寄存器	E6H									0000,0000
IAP_CONTR	IAP 控制寄存器	E7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT[2:0]		0000,x000	

13.1.1 EEPROM 数据寄存器（IAP_DATA）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_DATA	E2H								

在进行 EEPROM 的读操作时，命令执行完成后读出的 EEPROM 数据保存在 IAP_DATA 寄存器中。在进行 EEPROM 的写操作时，在执行写命令前，必须将待写入的数据存放在 IAP_DATA 寄存器中，再发送写命令。擦除 EEPROM 命令与 IAP_DATA 寄存器无关。

13.1.2 EEPROM 地址寄存器（IAP_ADDR）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_ADDRH	E3H								
IAP_ADDRL	E4H								

EEPROM 进行读、写、擦除操作的目标地址寄存器。IAP_ADDRH 保存地址的高字节，IAP_ADDRL 保存地址的低字节

13.1.3 EEPROM 命令寄存器 (IAP_CMD)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_CMD	E5H	-	-	-	-	-	-	CMD[1:0]	

CMD[1:0]: 发送EEPROM操作命令

00: 空操作

01: 读 EEPROM 命令。读取目标地址所在的 1 字节。

10: 写 EEPROM 命令。写目标地址所在的 1 字节。

11: 擦除 EEPROM。擦除目标地址所在的 1 页 (1 扇区/512 字节)。

13.1.4 EEPROM 触发寄存器 (IAP_TRIG)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_TRIG	E6H								

设置完成 EEPROM 读、写、擦除的命令寄存器、地址寄存器、数据寄存器以及控制寄存器后，需向触发寄存器 IAP_TRIG 依次写入 5AH、A5H (顺序不能交换) 两个触发命令来触发相应的读、写、擦除操作。操作完成后，EEPROM 地址寄存器 IAP_ADDRH、IAP_ADDRL 和 EEPROM 命令寄存器 IAP_CMD 的内容不变。如果接下来要对下一个地址的数据进行操作，需手动更新地址寄存器 IAP_ADDRH 和寄存器 IAP_ADDRL 的值。

注意：每次 EEPROM 操作时，都要对 IAP_TRIG 先写入 5AH，再写入 A5H，相应的命令才会生效。写完触发命令后，CPU 会处于 IDLE 等待状态，直到相应的 IAP 操作执行完成后 CPU 才会从 IDLE 状态返回正常状态继续执行 CPU 指令。

13.1.5 EEPROM 控制寄存器 (IAP_CONTR)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
IAP_CONTR	E7H	IAPEN	SWBS	SWRST	CMD_FAIL	-	WT[2:0]		

IAPEN: EEPROM操作使能控制位

- 0: 禁止 EEPROM 操作
- 1: 使能 EEPROM 操作

SWBS: 软件复位选择控制位, (需要与SWRST配合使用)

- 0: 软件复位后从用户代码开始执行程序
- 1: 软件复位后从系统 ISP 监控代码区开始执行程序

SWRST: 软件复位控制位

- 0: 无动作
- 1: 产生软件复位

CMD_FAIL: EEPROM操作失败状态位, 需要软件清零

- 0: EEPROM 操作正确
- 1: EEPROM 操作失败

WT[2:0]: 设置EEPROM操作等待时间

WT[2:0]	字节读取	字节编程	扇区擦除	参考系统工作频率
000	2 个时钟	1760 个时钟	672384 个时钟	>= 30MHz
001	2 个时钟	1320 个时钟	504288 个时钟	>= 24MHz
010	2 个时钟	1100 个时钟	420240 个时钟	>= 20MHz
011	2 个时钟	660 个时钟	252144 个时钟	>= 12MHz
100	2 个时钟	330 个时钟	126072 个时钟	>= 6MHz
101	2 个时钟	165 个时钟	63036 个时钟	>= 3MHz
110	2 个时钟	110 个时钟	42024 个时钟	>= 2MHz
111	2 个时钟	55 个时钟	21012 个时钟	>= 1MHz

14 ADC 模数转换

单片机内部集成了一个 10 位高速 A/D 转换器。ADC 的时钟频率为系统频率 2 分频再经过用户设置的分频系数进行再次分频（ADC 的工作时钟频率范围为 SYSclk/2/1 到 SYSclk/2/16）。

ADC 转换结果的数据格式有两种：左对齐和右对齐。可方便用户程序进行读取和引用。

14.1 ADC 相关的寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
ADC_CONTR	ADC 控制寄存器	C5H	ADC_POWER	SPEED[1:0]		ADC_FLAG	ADC_START	ADC_CHS[2:0]			000x,0000
ADC_RES	ADC 转换结果高位寄存器	C6H									0000,0000
ADC_RES1	ADC 转换结果低位寄存器	BEH									0000,0000

14.1.1 ADC 控制寄存器（ADC_CONTR），PWM 触发 ADC 控制

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
ADC_CONTR	C5H	ADC_POWER	SPEED[1:0]		ADC_START	ADC_FLAG	ADC_CHS[2:0]		

ADC_POWER: ADC 电源控制位

- 0: 关闭 ADC 电源
- 1: 打开 ADC 电源。

ADC_START: ADC 转换启动控制位。写入 1 后开始 ADC 转换，转换完成后硬件自动将此位清零。

- 0: 无影响。即使 ADC 已经开始转换工作，写 0 也不会停止 A/D 转换。
- 1: 开始 ADC 转换，转换完成后硬件自动将此位清零。

ADC_FLAG: ADC 转换结束标志位。当 ADC 完成一次转换后，硬件会自动将此位置 1，并向 CPU 提出中断请求。此标志位必须软件清零。

SPEED[1:0]: ADC 转换速度控制

SPEED[1:0]	ADC 通道
00	540 个时钟周期转换一次
01	360 个时钟周期转换一次
10	180 个时钟周期转换一次
11	90 个时钟周期转换一次

ADC_CHS[2:0]: ADC 模拟通道选择位

ADC_CHS[2:0]	ADC 通道
000	P1.0/ADC0
001	P1.1/ADC1
010	P1.2/ADC2
011	P1.3/ADC3
100	P1.4/ADC4
101	P1.5/ADC5

110	P1.6/ADC6
111	P1.7/ADC7

14.1.2 ADC 转换结果寄存器（ADC_RES, ADC_RES1）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
ADC_RES	C6H								
ADC_RES1	BEH								

当 A/D 转换完成后，10 位的转换结果会自动保存到 ADC_RES 和 ADC_RES1 中。保存结果的数据格式请参考 ADC_CFG 寄存器中的 RESFMT 设置。

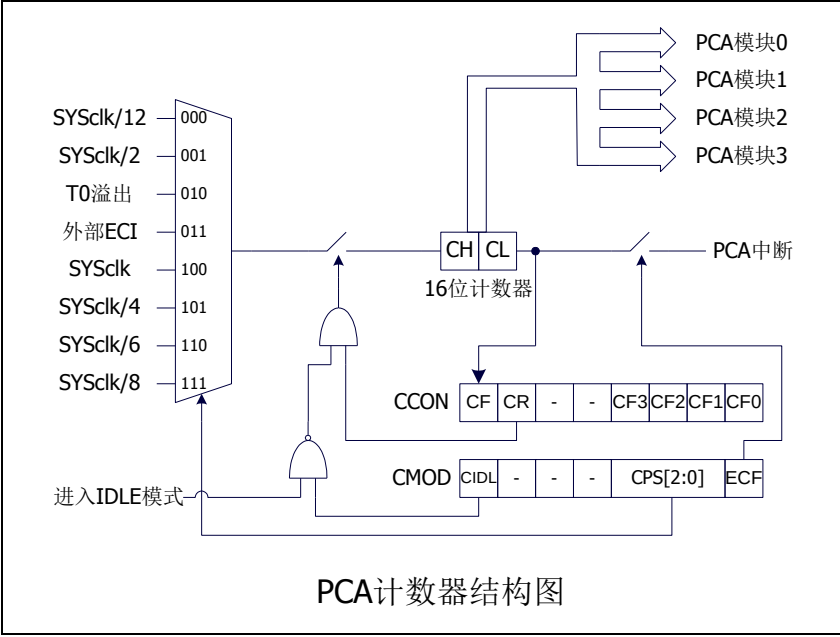
14.2 10 位 ADC 静态特性

符号	描述	最小值	典型值	最大值	单位
RES	分辨率	-	10	-	Bits
E _T	整体误差	-	1.3	3	LSB
E _O	偏移误差	-	0.3	1	LSB
E _G	增益误差	-	0	1	LSB
E _D	微分非线性误差	-	0.7	1.5	LSB
E _I	积分非线性误差	-	1	2	LSB
R _{AIN}	通道等效电阻	-	∞	-	ohm
R _{ESD}	采样保持电容前串接的抗静电电阻	-	700	-	ohm
C _{ADC}	内部采样保持电容	-	16.5	-	pF

15 PCA/CCP/PWM 应用

单片机内部集成了 4 组可编程计数器阵列（PCA/CCP/PWM）模块，可用于软件定时器、外部脉冲捕获、高速脉冲输出和 PWM 脉宽调制输出。

PCA 内部含有一个特殊的 16 位计数器，4 组 PCA 模块均与之相连接。PCA 计数器的结构图如下：



15.1 PCA 相关的寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
CCON	PCA 控制寄存器	D8H	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,x000
CMOD	PCA 模式寄存器	D9H	CIDL	-	-	-	CPS[2:0]			ECF	0xxx,0000
CCAPM0	PCA 模块 0 模式控制寄存器	DAH	-	ECOM0	CCAPP0	CCAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	PCA 模块 1 模式控制寄存器	DBH	-	ECOM1	CCAPP1	CCAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	PCA 模块 2 模式控制寄存器	DCH	-	ECOM2	CCAPP2	CCAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	PCA 模块 3 模式控制寄存器	DDH	-	ECOM3	CCAPP3	CCAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000
CL	PCA 计数器低字节	E9H									0000,0000
CCAP0L	PCA 模块 0 低字节	EAH									0000,0000
CCAP1L	PCA 模块 1 低字节	EBH									0000,0000
CCAP2L	PCA 模块 2 低字节	ECH									0000,0000
CCAP3L	PCA 模块 3 低字节	EDH									0000,0000
PCA_PWM0	PCA0 的 PWM 模式寄存器	F2H							EPC0H	EPC0L	00xx,xx00
PCA_PWM1	PCA1 的 PWM 模式寄存器	F3H							EPC1H	EPC1L	00xx,xx00
PCA_PWM2	PCA2 的 PWM 模式寄存器	F4H							EPC2H	EPC2L	00xx,xx00
PCA_PWM3	PCA3 的 PWM 模式寄存器	F5H							EPC3H	EPC3L	00xx,xx00
CH	PCA 计数器高字节	F9H									0000,0000
CCAP0H	PCA 模块 0 高字节	FAH									0000,0000

CCAP1H	PCA 模块 1 高字节	FBH		0000,0000
CCAP2H	PCA 模块 2 高字节	FCH		0000,0000
CCAP3H	PCA 模块 3 高字节	FDH		0000,0000

15.1.1 PCA 控制寄存器 (CCON)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CCON	D8H	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0

CF: PCA 计数器溢出中断标志。当 PCA 的 16 位计数器计数发生溢出时, 硬件自动将此位置 1, 并向 CPU 提出中断请求。此标志位需要软件清零。

CR: PCA 计数器允许控制位。

0: 停止 PCA 计数

1: 启动 PCA 计数

CCFn (n=0,1,2,3): PCA 模块中断标志。当 PCA 模块发生匹配或者捕获时, 硬件自动将此位置 1, 并向 CPU 提出中断请求。此标志位需要软件清零。

15.1.2 PCA 模式寄存器 (CMOD)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CMOD	D9H	CIDL	-	-	-	CPS[2:0]			ECF

CIDL: 空闲模式下是否停止 PCA 计数。

0: 空闲模式下 PCA 继续计数

1: 空闲模式下 PCA 停止计数

CPS[2:0]: PCA 计数脉冲源选择位

CPS[2:0]	PCA 的输入时钟源
000	系统时钟/12
001	系统时钟/2
010	定时器 0 的溢出脉冲
011	ECI 脚的外部输入时钟
100	系统时钟
101	系统时钟/4
110	系统时钟/6
111	系统时钟/8

ECF: PCA 计数器溢出中断允许位。

0: 禁止 PCA 计数器溢出中断

1: 使能 PCA 计数器溢出中断

15.1.3 PCA 计数器寄存器 (CL, CH)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CL	E9H								
CH	F9H								

由 CL 和 CH 两个字节组合成一个 16 位计数器, CL 为低 8 位计数器, CH 为高 8 位计数器。每个 PCA 时钟 16 位计数器自动加 1。

15.1.4 PCA 模块模式控制寄存器 (CCAPMn)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CCAPM0	DAH	-	ECOM0	CCAPP0	CCAPN0	MAT0	TOG0	PWM0	ECCF0
CCAPM1	DBH	-	ECOM1	CCAPP1	CCAPN1	MAT1	TOG1	PWM1	ECCF1
CCAPM2	DCH	-	ECOM2	CCAPP2	CCAPN2	MAT2	TOG2	PWM2	ECCF2
CCAPM3	DDH	-	ECOM3	CCAPP3	CCAPN3	MAT3	TOG3	PWM3	ECCF3

ECOMn: 允许 PCA 模块 n 的比较功能

CCAPPn: 允许 PCA 模块 n 进行上升沿捕获

CCAPNn: 允许 PCA 模块 n 进行下降沿捕获

MATn: 允许 PCA 模块 n 的匹配功能

TOGn: 允许 PCA 模块 n 的高速脉冲输出功能

PWMn: 允许 PCA 模块 n 的脉宽调制输出功能

ECCFn: 允许 PCA 模块 n 的匹配/捕获中断

15.1.5 PCA 模块模式捕获值/比较值寄存器 (CCAPnL, CCAPnH)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
CCAP0L	EAH								
CCAP1L	EBH								
CCAP2L	ECH								
CCAP3L	EDH								
CCAP0H	FAH								
CCAP1H	FBH								
CCAP2H	FCH								
CCAP3H	FDH								

当 PCA 模块捕获功能使能时, CCAPnL 和 CCAPnH 用于保存发生捕获时的 PCA 的计数值 (CL 和 CH);

当 PCA 模块比较功能使能时, PCA 控制器会将当前 CL 和 CH 中的计数值与保存在 CCAPnL 和 CCAPnH 中的值进行比较, 并给出比较结果; 当 PCA 模块匹配功能使能时, PCA 控制器会将当前 CL 和 CH 中的计数值与保存在 CCAPnL 和 CCAPnH 中的值进行比较, 看是否匹配 (相等), 并给出匹配结果。

15.1.6 PCA 模块 PWM 模式控制寄存器 (PCA_PWMn)

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
PCA_PWM0	F2H				-	-	-	EPC0H	EPC0L
PCA_PWM1	F3H				-	-	-	EPC1H	EPC1L
PCA_PWM2	F4H				-	-	-	EPC2H	EPC2L
PCA_PWM3	F5H				-	-	-	EPC3H	EPC3L

EPCnH: PWM 模式下, 重载值的最高位 (8 位 PWM 的第 9 位)

EPCnL: PWM 模式下, 比较值的最高位 (8 位 PWM 的第 9 位)

15.2 PCA 工作模式

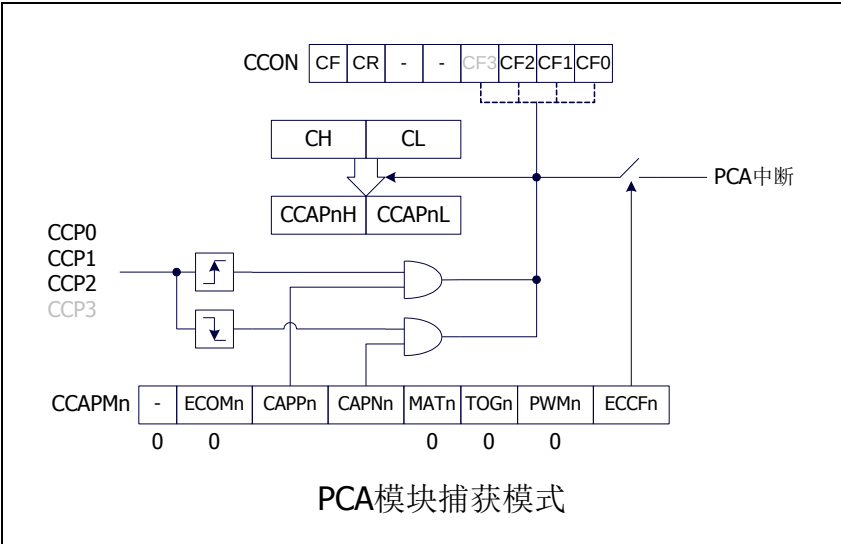
每组模块都可独立设置工作模式。模式设置如下所示：

CCAPMn								模块功能
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	
-	0	0	0	0	0	0	0	无操作
-	1	0	0	0	0	1	0	6/7/8 位 PWM 模式，无中断
-	1	1	0	0	0	1	1	6/7/8 位 PWM 模式，产生上升沿中断
-	1	0	1	0	0	1	1	6/7/8 位 PWM 模式，产生下降沿中断
-	1	1	1	0	0	1	1	6/7/8 位 PWM 模式，产生边沿中断
-	0	1	0	0	0	0	x	16 位上升沿捕获
-	0	0	1	0	0	0	x	16 位下降沿捕获
-	0	1	1	0	0	0	x	16 位边沿捕获
-	1	0	0	1	0	0	x	16 位软件定时器
-	1	0	0	1	1	0	x	16 位高速脉冲输出

15.2.1 捕获模式

要使一个 PCA 模块工作在捕获模式，寄存器 CCAPMn 中的 CAPNn 和 CAPPn 至少有一位必须置 1（也可两位都置 1）。PCA 模块工作于捕获模式时，对模块的外部 CCP0/CCP1/CCP2 管脚的输入跳变进行采样。当采样到有效跳变时，PCA 控制器立即将 PCA 计数器 CH 和 CL 中的计数值装载到模块的捕获寄存器中 CCAPnL 和 CCAPnH，同时将 CCON 寄存器中相应的 CCFn 置 1。若 CCAPMn 中的 ECCFn 位被设置为 1，将产生中断。由于所有 PCA 模块的中断入口地址是共享的，所以在中断服务程序中需要判断是哪一个模块产生了中断，并注意中断标志位需要软件清零。

PCA 模块工作于捕获模式的结构图如下图所示：

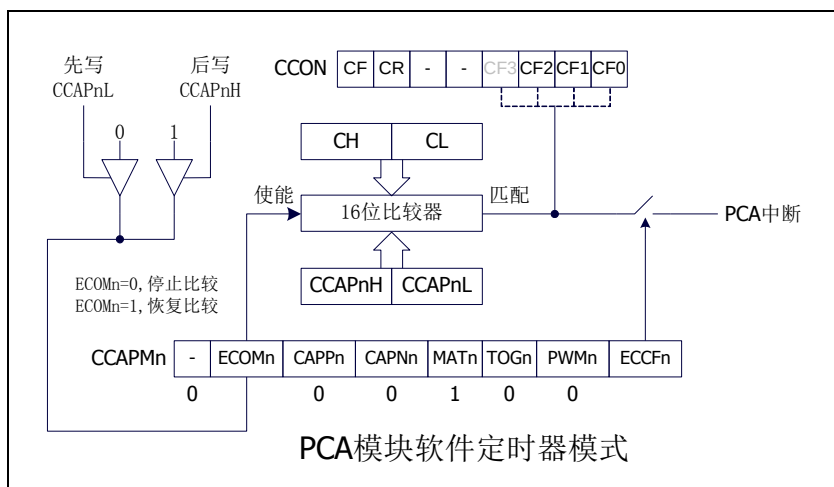


15.2.2 软件定时器模式

通过置位 CCAPMn 寄存器的 ECOM 和 MAT 位，可使 PCA 模块用作软件定时器。PCA 计数器值 CL 和 CH 与模块捕获寄存器的值 CCAPnL 和 CCAPnH 相比较，当两者相等时，CCON 中的 CCFn 会被

置 1，若 CCAPMn 中的 ECCFn 被设置为 1 时将产生中断。CCFn 标志位需要软件清零。

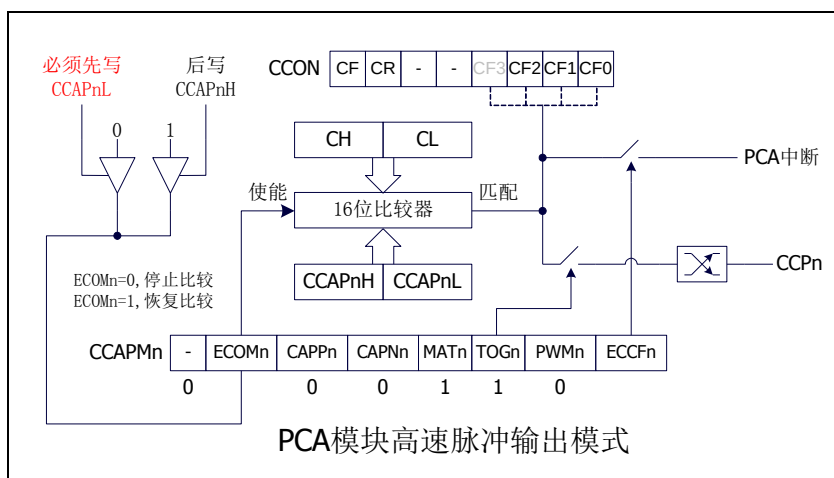
PCA 模块工作于软件定时器模式的结构图如下图所示：



15.2.3 高速脉冲输出模式

当 PCA 计数器的计数值与模块捕获寄存器的值相匹配时,PCA 模块的 CCPn 输出将发生翻转。要激活高速脉冲输出模式，CCAPMn 寄存器的 TOGn、MATn 和 ECOMn 位必须都置 1。

PCA 模块工作于高速脉冲输出模式的结构图如下图所示：



15.2.4 PWM 脉宽调制模式及频率计算公式

脉宽调制是使用程序来控制波形的占空比、周期、相位波形的一种技术，在三相电机驱动、D/A 转换等场合有广泛的应用。PCA 模块可以通过设定各自的 PCA_PWMn 寄存器使其工作于 8 位 PWM 或 7 位 PWM 或 6 位 PWM 或 10 位 PWM 模式。要能使 PCA 模块的 PWM 功能,模块寄存器 CCAPMn 的 PWMn 和 ECOMn 位必须置 1。

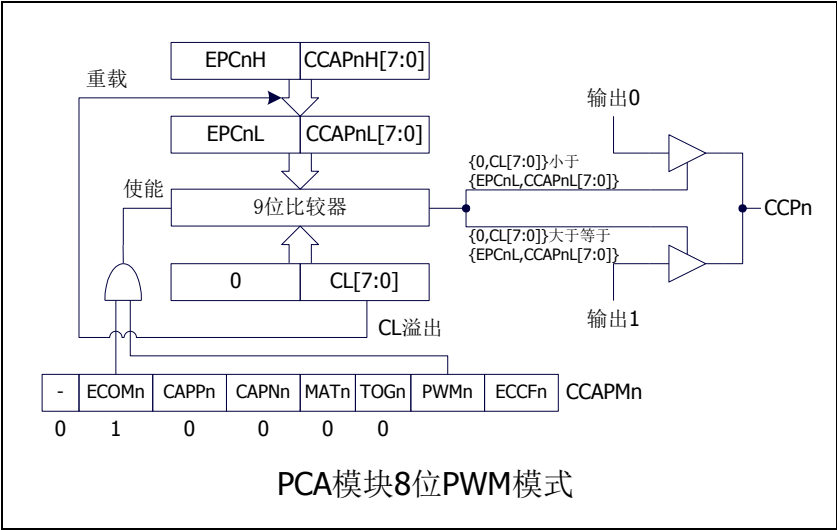
PCA_PWMn 寄存器中的 EBSn[1:0]设置为 00 时,PCA 模块 n 工作于 8 位 PWM 模式,此时将{0,CL[7:0]}与捕获寄存器{EPCnL,CCAPnL[7:0]}进行比较。当 PCA 模块工作于 8 位 PWM 模式时,由于所有模块共用一个 PCA 计数器,所有它们的输出频率相同。各个模块的输出占空比使用寄存器{EPCnL,CCAPnL[7:0]}进行设置。当{0,CL[7:0]}的值小于{EPCnL,CCAPnL[7:0]}时,输出为低电平;当{0,CL[7:0]}的值等于或大于{EPCnL,CCAPnL[7:0]}时,输出为高电平。当 CL[7:0]的值由 FF 变为 00 溢出时,{EPCnH,CCAPnH[7:0]}的内容重新装载到{EPCnL,CCAPnL[7:0]}中。这样就可实现无干扰地更新 PWM。

PCA时钟输入源频率

8位模式的PWM频率= $\frac{\text{PCA时钟输入源频率}}{256}$

当EPCnH=0及CCAPnH=00H时，PWM固定输出高
当EPCnH=1及CCAPnH=FFH时，PWM固定输出低

PCA 模块工作于 8 位 PWM 模式的结构图如下图所示：

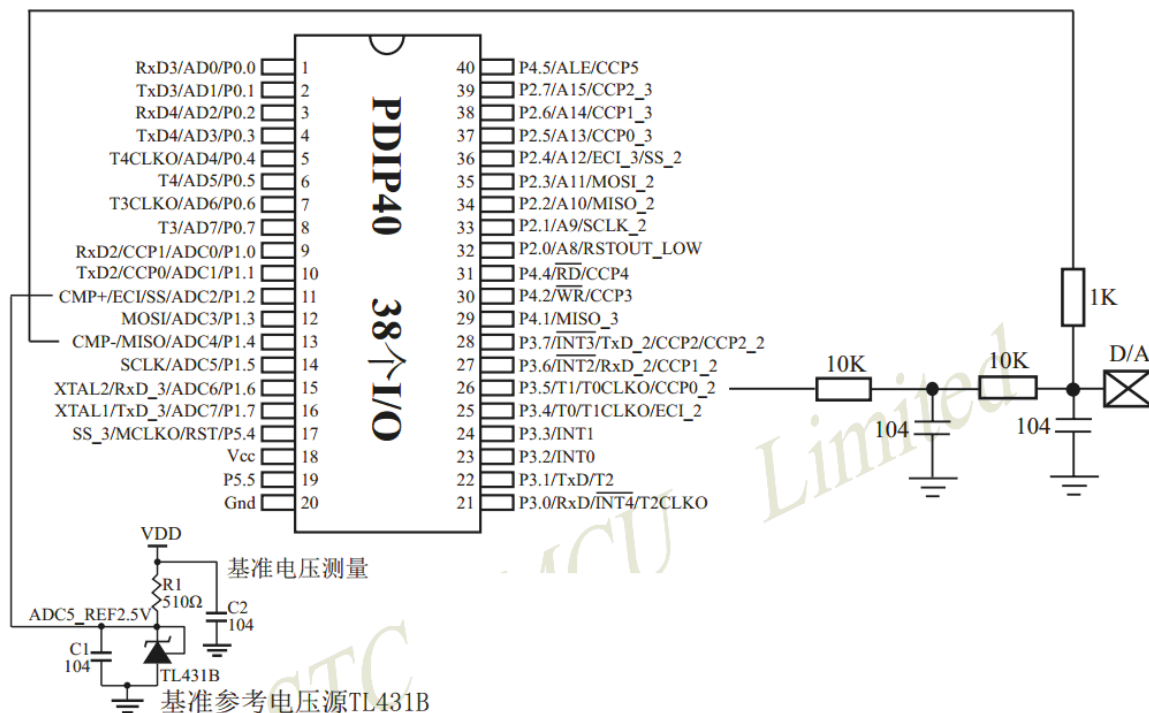


15.2.4.1 如何控制 PWM 固定输出高电平/低电平

当 PCA_PWMn &= 0xC0，CCAPnH = 0x00 时，PWM 固定输出高电平

当 PCA_PWMn |= 0x3F，CCAPnH = 0xFF 时，PWM 固定输出低电平

15.3 利用 CCP/PCA/PWM 模块实现 8~16 位 DAC 的参考线路图



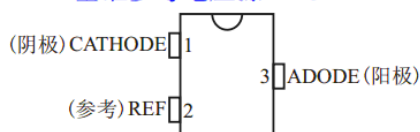
如应用简单,可无需基准参考电压源,直接与 V_{CC} 比较即可。

提示：

利用CCP/PCA模块的高速脉冲输出功能实现9~16位PWM来实现9~16位DAC，或用本身的硬件8位PWM来实现8位DAC，单片机本身也有10位ADC。

- (1) PWM频率越高，输出波形越平滑。
- (2) 如果工作电压为5V，需输出1V电压，则设置高电平为1/5，低电平为4/5，则PWM输出电压就为1V。
- (3) 如果要输出高精度电压，建议用A/D检测输出的电压值，然后根据A/D检测的电压值逐步调整到需要的电压。

基准参考电压源TL431B



SOT23-3封装, RMB ¥ 0.15~0.3

基准参考电压源TL431B的符号



如应用简单,可无需基准参考电压源,直接与 V_{CC} 比较即可。

16 同步串行外设接口 SPI

单片机内部集成了一种高速串行通信接口——SPI 接口。SPI 是一种全双工的高速同步通信总线。SPI 接口提供了两种操作模式：主模式和从模式。

16.1 SPI 相关的寄存器

符号	描述	地址	位地址与符号								复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
SPSTAT	SPI 状态寄存器	84H	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
SPCTL	SPI 控制寄存器	85H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]		0000,0100
SPDAT	SPI 数据寄存器	86H									0000,0000

16.1.1 SPI 状态寄存器（SPSTAT）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SPSTAT	84H	SPIF	WCOL	-	-	-	-	-	-

SPIF: SPI 中断标志位。

当发送/接收完成 1 字节的数据后，硬件自动将此位置 1，并向 CPU 提出中断请求。当 SSIG 位被设置为 0 时，由于 SS 管脚电平的变化而使得设备的主/从模式发生改变时，此标志位也会被硬件自动置 1，以标志设备模式发生变化。

注意：此标志位必须用户通过软件方式向此位写 1 进行清零。

WCOL: SPI 写冲突标志位。

当 SPI 在进行数据传输的过程中写 SPDAT 寄存器时，硬件将此位置 1。

注意：此标志位必须用户通过软件方式向此位写 1 进行清零。

16.1.2 SPI 控制寄存器（SPCTL），SPI 速度控制

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SPCTL	85H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]	

SSIG: SS 引脚功能控制位

0: SS 引脚确定器件是主机还是从机

1: 忽略 SS 引脚功能，使用 MSTR 确定器件是主机还是从机

SPEN: SPI 使能控制位

0: 关闭 SPI 功能

1: 使能 SPI 功能

DORD: SPI 数据位发送/接收的顺序

0: 先发送/接收数据的高位（MSB）

1: 先发送/接收数据的低位（LSB）

MSTR: 器件主/从模式选择位

设置主机模式：

若 SSIG=0，则 SS 管脚必须为高电平且设置 MSTR 为 1

若 SSIG=1，则只需要设置 MSTR 为 1（忽略 SS 管脚的电平）

设置从机模式：

若 SSIG=0，则 SS 管脚必须为低电平（与 MSTR 位无关）

若 SSIG=1，则只需要设置 MSTR 为 0（忽略 SS 管脚的电平）

CPOL: SPI 时钟极性控制

0: SCLK 空闲时为低电平，SCLK 的前时钟沿为上升沿，后时钟沿为下降沿

1: SCLK 空闲时为高电平，SCLK 的前时钟沿为下降沿，后时钟沿为上升沿

CPHA: SPI 时钟相位控制

0: 数据 SS 管脚为低电平驱动第一位数据并在 SCLK 的后时钟沿改变数据，前时钟沿采样数据（必须 SSIG=0）

1: 数据在 SCLK 的前时钟沿驱动，后时钟沿采样

SPR[1:0]: SPI 时钟频率选择

SPR[1:0]	SCLK 频率
00	SYScLk/4
01	SYScLk/8
10	SYScLk/16
11	SYScLk/32

16.1.3 SPI 数据寄存器（SPDAT）

符号	地址	B7	B6	B5	B4	B3	B2	B1	B0
SPDAT	86H								

SPI 发送/接收数据缓冲器。

16.2 SPI 通信方式

SPI 的通信方式通常有 3 种：单主单从（一个主机设备连接一个从机设备）、互为主从（两个设备连接，设备和互为主机和从机）、单主多从（一个主机设备连接多个从机设备）

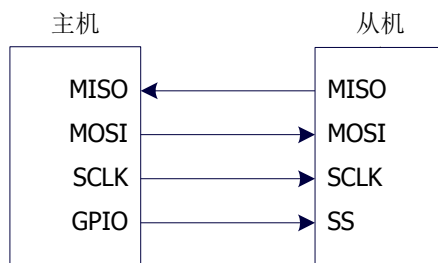
16.2.1 单主单从

两个设备相连，其中一个设备固定作为主机，另外一个固定作为从机。

主机设置：SSIG 设置为 1，MSTR 设置为 1，固定为主机模式。主机可以使用任意端口连接从机的 SS 管脚，拉低从机的 SS 脚即使能从机

从机设置：SSIG 设置为 0，SS 管脚作为从机的片选信号。

单主单从连接配置图如下所示：



单主单从配置

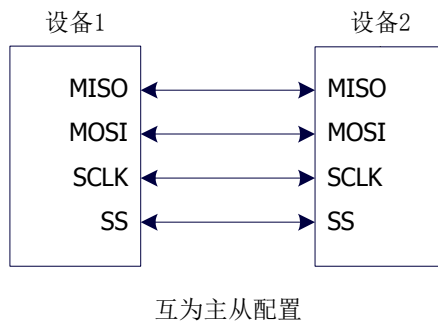
16.2.2 互为主从

两个设备相连，主机和从机不固定。

设置方法 1：两个设备初始化时都设置为 SSIG 设置为 0，MSTR 设置为 1，且将 SS 脚设置为双向口模式输出高电平。此时两个设备都是不忽略 SS 的主机模式。当其中一个设备需要启动传输时，可将自己的 SS 脚设置为输出模式并输出低电平，拉低对方的 SS 脚，这样另一个设备就被强行设置为从机模式了。

设置方法 2：两个设备初始化时都将自己设置成忽略 SS 的从机模式，即将 SSIG 设置为 1，MSTR 设置为 0。当其中一个设备需要启动传输时，先检测 SS 管脚的电平，如果时候高电平，就将自己设置成忽略 SS 的主模式，即可进行数据传输了。

互为主从连接配置图如下所示：



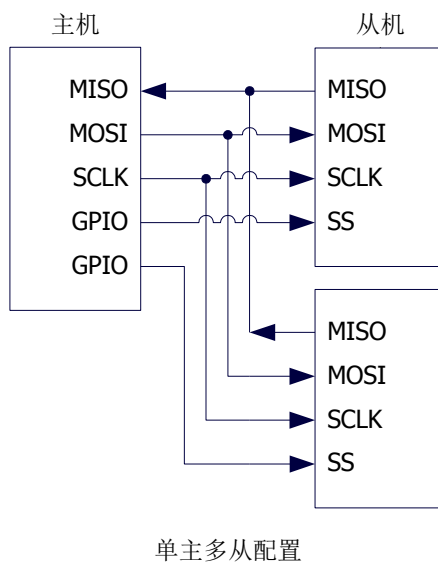
16.2.3 单主多从

多个设备相连，其中一个设备固定作为主机，其他设备固定作为从机。

主机设置：SSIG 设置为 1，MSTR 设置为 1，固定为主机模式。主机可以使用任意端口分别连接各个从机的 SS 管脚，拉低其中一个从机的 SS 脚即使能相应的从机设备

从机设置：SSIG 设置为 0，SS 管脚作为从机的片选信号。

单主多从连接配置图如下所示：



16.3 配置 SPI

控制位			通信端口				说明
SPEN	SSIG	MSTR	SS	MISO	MOSI	SCLK	
0	x	x	x	输入	输入	输入	关闭 SPI 功能, SS/MOSI/MISO/SCLK 均为普通 I/O
1	0	0	0	输出	输入	输入	从机模式, 且被选中
1	0	0	1	高阻	输入	输入	从机模式, 但未被选中
1	0	1→0	0	输出	输入	输入	从机模式, 不忽略 SS 且 MSTR 为 1 的主机模式, 当 SS 管脚被拉低时, MSTR 将被硬件自动清零, 工作模式将被被动设置为从机模式
1	0	1	1	输入	高阻	高阻	主机模式, 空闲状态
					输出	输出	主机模式, 激活状态
1	1	0	x	输出	输入	输入	从机模式
1	1	1	x	输入	输出	输出	主机模式

从机模式的注意事项:

当 CPHA=0 时, SSIG 必须为 0 (即不能忽略 SS 脚)。在每次串行字节开始还发送前 SS 脚必须拉低, 并且在串行字节发送完后须重新设置为高电平。SS 管脚为低电平时不能对 SPDAT 寄存器执行写操作, 否则将导致一个写冲突错误。CPHA=0 且 SSIG=1 时的操作未定义。

当 CPHA=1 时, SSIG 可以置 1 (即可以忽略脚)。如果 SSIG=0, SS 脚可在连续传输之间保持低有效 (即一直固定为低电平)。这种方式适用于固定单主单从的系统。

主机模式的注意事项:

在 SPI 中, 传输总是由主机启动的。如果 SPI 使能 (SPEN=1) 并选择作为主机时, 主机对 SPI 数据寄存器 SPDAT 的写操作将启动 SPI 时钟发生器和数据的传输。在数据写入 SPDAT 之后的半个到一个 SPI 位时间后, 数据将出现在 MOSI 脚。写入主机 SPDAT 寄存器的数据从 MOSI 脚移出发送到从机的 MOSI 脚。同时从机 SPDAT 寄存器的数据从 MISO 脚移出发送到主机的 MISO 脚。

传输完一个字节后, SPI 时钟发生器停止, 传输完成标志 (SPIF) 置位, 如果 SPI 中断使能则会产生一个 SPI 中断。主机和从机 CPU 的两个移位寄存器可以看作是一个 16 位循环移位寄存器。当数据从主机移位传送到从机的同时, 数据也以相反的方向移入。这意味着在一个移位周期中, 主机和从机的数据相互交换。

通过 SS 改变模式

如果 SPEN=1, SSIG=0 且 MSTR=1, SPI 使能为主机模式, 并将 SS 脚可配置为输入模式或准双向口模式。这种情况下, 另外一个主机可将该脚驱动为低电平, 从而将该器件选择为 SPI 从机并向其发送数据。为了避免争夺总线, SPI 系统将该从机的 MSTR 清零, MOSI 和 SCLK 强制变为输入模式, 而 MISO 则变为输出模式, 同时 SPSTAT 的 SPIF 标志位置 1。

用户软件必须一直对 MSTR 位进行检测, 如果该位被一个从机选择动作而被动清零, 而用户想继续将 SPI 作为主机, 则必须重新设置 MSTR 位, 否则将一直处于从机模式。

写冲突

SPI 在发送时为单缓冲，在接收时为双缓冲。这样在前一次发送尚未完成之前，不能将新的数据写入移位寄存器。当发送过程中对数据寄存器 SPDAT 进行写操作时，WCOL 位将被置 1 以指示发生数据写冲突错误。在这种情况下，当前发送的数据继续发送，而新写入的数据将丢失。

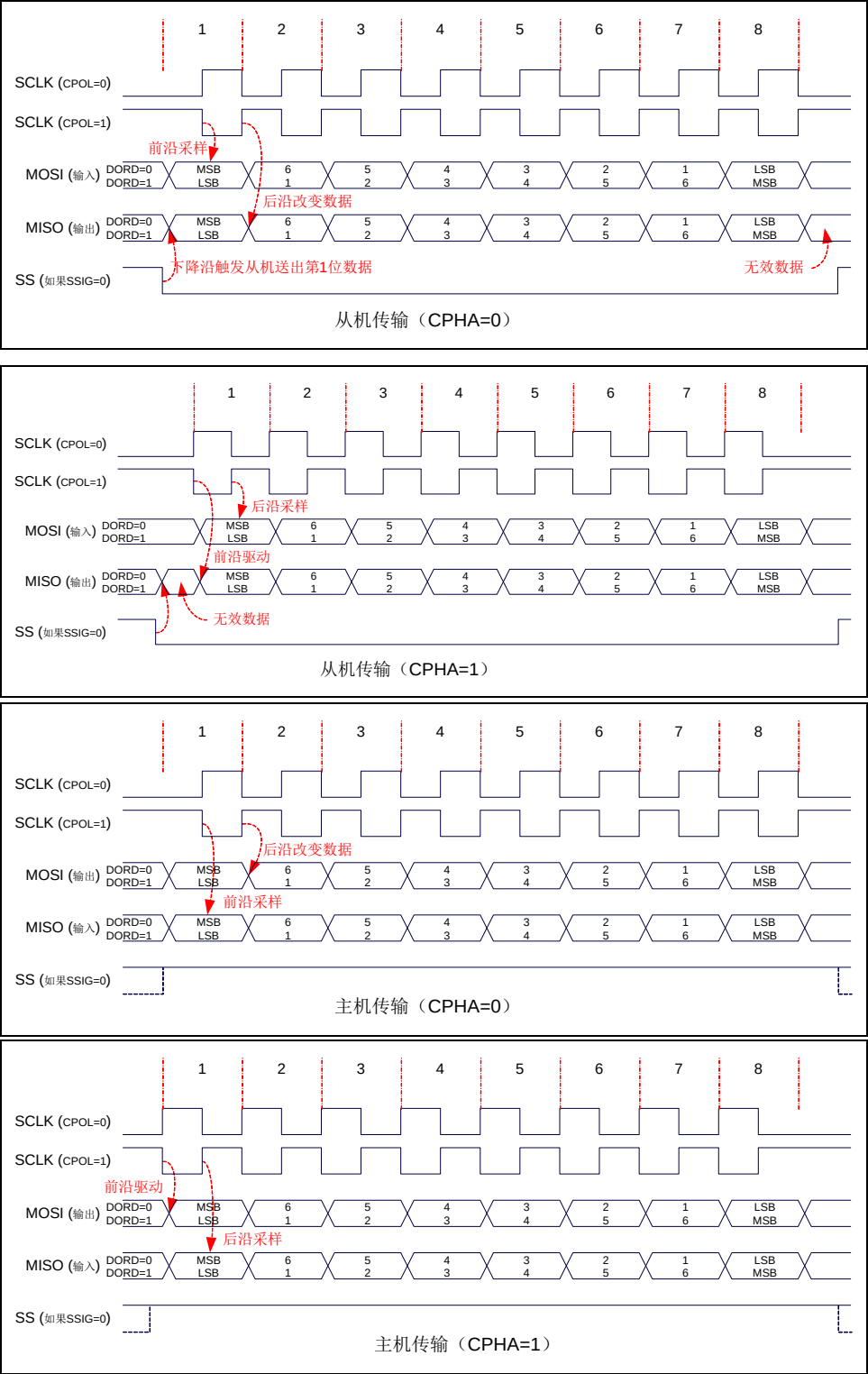
当对主机或从机进行写冲突检测时，主机发生写冲突的情况是很罕见的，因为主机拥有数据传输的完全控制权。但从机有可能发生写冲突，因为当主机启动传输时，从机无法进行控制。

接收数据时，接收到的数据传送到一个并行读数据缓冲区，这样将释放移位寄存器以进行下一个数据的接收。但必须在下个字符完全移入之前从数据寄存器中读出接收到的数据，否则，前一个接收数据将丢失。

WCOL 可通过软件向其写入“1”清零。

16.4 数据模式

SPI 的时钟相位控制位 CPHA 可以让用户设定数据采样和改变时的时钟沿。时钟极性位 CPOL 可以让用户设定时钟极性。下面图例显示了不同时钟相位、极性设置下 SPI 通讯时序。



附录A 电气特性

A.1 绝对最大额定值

参数	最小值	最大值	单位
存储温度	-55	+150	℃
工作温度	-40	+85	℃
工作电压（5V 器件）	3.5	5.5	V
工作电压（3.3V 器件）	2.2	3.6	V
VDD 对地电压	-0.3	+5.5	V
I/O 口对地电压	-0.3	VDD+0.3	V

A.2 直流特性（3.3V）

（VSS=0V，VDD=3.3V，测试温度=25℃）

标号	参数	范围				测试环境及说明
		最小值	典型值	最大值	单位	
I _{PD}	掉电模式电流	-	0.4	-	uA	
I _{WKT}	掉电唤醒定时器	-	1.4	-	uA	
I _{LVD}	低压检测模块功耗	-	10	-	uA	
I _{CMP}	比较器功耗	-	90	-	uA	
I _{IDL}	空闲模式电流（内部 32KHz）	-	0.48	-	mA	相当于传统 8051 的 0.5M
	空闲模式电流（6MHz）	-	0.88	-	mA	相当于传统 8051 的 79M
	空闲模式电流（12MHz）	-	1.00	-	mA	相当于传统 8051 的 158M
	空闲模式电流（24MHz）	-	1.16	-	mA	相当于传统 8051 的 317M
I _{NOR}	正常模式电流（内部 32KHz）	-	0.48	-	mA	相当于传统 8051 的 0.5M
	正常模式电流（6MHz）	-	1.49	-	mA	相当于传统 8051 的 79M
	正常模式电流（12MHz）	-	2.09	-	mA	相当于传统 8051 的 158M
	正常模式电流（24MHz）	-	3.16	-	mA	相当于传统 8051 的 317M
V _{IL1}	输入低电平	-	-	0.99	V	打开施密特触发
		-	-	1.07	V	关闭施密特触发
V _{IH1}	输入高电平（普通 I/O）	1.18	-	-	V	打开施密特触发
		1.09	-	-	V	关闭施密特触发
V _{IH2}	输入高电平（复位脚）	1.18	-	0.99	V	
I _{OL1}	输出低电平的灌电流	-	20	-	mA	端口电压 0.45V
I _{OH1}	输出高电平电流（双向模式）	200	270	-	uA	
I _{OH2}	输出高电平电流（推挽模式）	-	20	-	mA	端口电压 2.4V
I _{IL}	逻辑 0 输入电流	-	-	50	uA	端口电压 0V
I _{TL}	逻辑 1 到 0 的转移电流	100	270	600	uA	端口电压 2.0V
R _{PU}	I/O 口上拉电阻	5.8	5.9	6.0	KΩ	
I/O 速度	I/O 大电流驱动，I/O 快速转换		25		MHz	PxDR=0, PxSR=0
	I/O 小电流驱动，I/O 快速转换		22		MHz	PxDR=1, PxSR=0
	I/O 大电流驱动，I/O 慢速转换		16		MHz	PxDR=0, PxSR=1
	I/O 小电流驱动，I/O 慢速转换		12		MHz	PxDR=1, PxSR=1
比较器	最快速度		10		MHz	关闭所有模拟和数字滤波
	模拟滤波时间		0.1		us	
	数字滤波时间		0		系统	LCDTY=0
			n+2		时钟	LCDTY=n (n=1~63)
I _{PD2}	使能比较器时掉电模式功耗	-	400	-	uA	
I _{PD3}	使能 LVD 时掉电模式功耗	-	470	-	uA	

A.3 直流特性（5.0V）

（VSS=0V，VDD=5.0V，测试温度=25℃）

标号	参数	范围				测试环境及说明
		最小值	典型值	最大值	单位	
I _{PD}	掉电模式电流	-	0.6	-	uA	
I _{WKT}	掉电唤醒定时器	-	3.6	-	uA	
I _{LVD}	低压检测模块功耗	-	30	-	uA	
I _{CMP}	比较器功耗	-	90	-	uA	
I _{IDL}	空闲模式电流（内部 32KHz）	-	0.58	-	mA	相当于传统 8051 的 0.5M
	空闲模式电流（6MHz）	-	0.98	-	mA	相当于传统 8051 的 79M
	空闲模式电流（12MHz）	-	1.10	-	mA	相当于传统 8051 的 158M
	空闲模式电流（24MHz）	-	1.25	-	mA	相当于传统 8051 的 317M
I _{NOR}	正常模式电流（内部 32KHz）	-	0.58	-	mA	相当于传统 8051 的 0.5M
	正常模式电流（6MHz）	-	1.59	-	mA	相当于传统 8051 的 79M
	正常模式电流（12MHz）	-	2.19	-	mA	相当于传统 8051 的 158M
	正常模式电流（24MHz）	-	3.27	-	mA	相当于传统 8051 的 317M
V _{IL1}	输入低电平	-	-	1.32	V	打开施密特触发
		-	-	1.48	V	关闭施密特触发
V _{IH1}	输入高电平（普通 I/O）	1.60	-	-	V	打开施密特触发
		1.54	-	-	V	关闭施密特触发
V _{IH2}	输入高电平（复位脚）	1.60	-	1.32	V	
I _{OL1}	输出低电平的灌电流	-	20	-	mA	端口电压 0.45V
I _{OH1}	输出高电平电流（双向模式）	200	270	-	uA	
I _{OH2}	输出高电平电流（推挽模式）	-	20	-	mA	端口电压 2.4V
I _{IL}	逻辑 0 输入电流	-	-	50	uA	端口电压 0V
I _{TL}	逻辑 1 到 0 的转移电流	100	270	600	uA	端口电压 2.0V
R _{PU}	I/O 口上拉电阻	4.1	4.2	4.4	KΩ	
I/O 速度	I/O 大电流驱动，I/O 快速转换		36		MHz	PxDR=0, PxSR=0
	I/O 小电流驱动，I/O 快速转换		32		MHz	PxDR=1, PxSR=0
	I/O 大电流驱动，I/O 慢速转换		26		MHz	PxDR=0, PxSR=1
	I/O 小电流驱动，I/O 慢速转换		22		MHz	PxDR=1, PxSR=1
比较器	最快速度		10		MHz	关闭所有模拟和数字滤波
	模拟滤波时间		0.1		us	
	数字滤波时间		0		系统	LCDTY=0
			n+2		时钟	LCDTY=n (n=1~63)
I _{PD2}	使能比较器时掉电模式功耗	-	460	-	uA	
I _{PD3}	使能 LVD 时掉电模式功耗	-	520	-	uA	