

MCS-51 单片机主要应用特性（一）

作者:中源单片机 发布:2004-12-18 浏览:287

MCS-51 单片机主要应用特性（一）

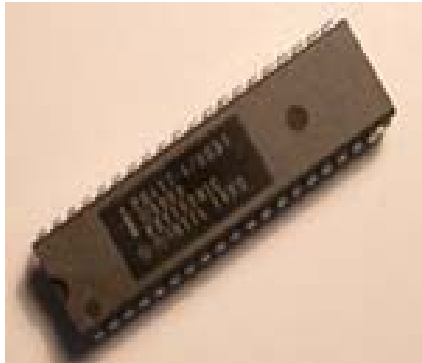
MCS-51 单片机是美国 INTE 公司于 1980 年推出的产品 ,与 MCS-48 单片机相比 ,它的结构更先进 ,功能更强 ,在原来的基础上增加了更多的电路单元和指令,指令数达 111 条 , MCS-51 单片机可以算是相当成功的产品 ,一直到现在 , MCS-51 系列或其兼容的单片机仍是应用的主流产品 ,各高校及专业学校的培训教材仍与 MCS-51 单片机作为代表进行理论基础学习。我们也以这一代表性的机型进行系统的讲解。

MCS-51 系列单片机主要包括 8031、8051 和 8751 等通用产品 ,其主要功能如下 :

8 位 CPU

- 4kbytes 程序存储器(ROM)
- 128bytes 的数据存储器(RAM)
- 32 条 I/O 口线
- 111 条指令 ,大部分为单字节指令
- 21 个专用寄存器
- 2 个可编程定时/计数器
- 5 个中断源 , 2 个优先级
- 一个全双工串行通信口
- 外部数据存储器寻址空间为 64kB

- 外部程序存储器寻址空间为 64kB
- 逻辑操作位寻址功能
- 双列直插 40PinDIP 封装
- 单一+5V 电源供电



MCS-51 以其典型的结构和完善的总线专用寄存器的集中管理，众多的逻辑位操作功能及面向控制的丰富的指令系统，堪称为一代“名机”，为以后的其它单片机的发展奠定了基础。正因为其优越的性能和完善的结构，导致后来的许多厂商多沿用或参考了其体系结构，有许多世界大的电气商丰富和发展了 MCS-51 单片机，象 PHILIPS、Dallas、ATMEL 等著名的半导体公司都推出了兼容 MCS-51 的单片机产品，就连我国的台湾 WINBOND 公司也发展了兼容 C51(人们习惯将 MCS-51 简称 C51,如果没有特别声明，二者同指 MCS-51 系列单片机)的单片机品种。

近年来 C51 获得了飞速的发展，C51 的发源公司 INTEL 由于忙于开发 PC 及高端微处理器而无精力继续发展自己的单片机，而由其它厂商将其发展，最典型的是 PHILIPS 和 ATML 公司，PHILIPS 公司主要是改善其性能，在原来的基础上发展了高速 I/O 口，A/D 转换器，PWM(脉宽调制)、WDT 等增强功能，并在低电压、低功耗、扩展串行总线(I²C)和控制网络总线(CAN)等功能加以完善

MCS-51 单片机主要应用特性（二）

PHILIPS 公司在发展 C51 的低功耗，高速度和增强型功能上作了不少贡献，当初主要由其来发展 C51 单片机，他们的 83Cxx 和 87Cxx 系列省去了并行扩展总线，适合于作为家用电器类控制的经济型单片机。

ATMEL 公司推出的 AT89Cxx 系列兼容 C51 的单片机，完美地将 Flash(非易失闪存技术)EEPROM 与 80C51 内核结合起来，仍采用 C51 的总体结构和指令系统，Flash 的可反擦写程序存储器能有效地降低开发费用，并能使单片机作多次重复使用。

Siemens 公司也沿用 C51 的内核，相继推出了 C500 系列单片机，在保持了与 C51 指令兼容的前提下，其产品的性能得到了进一步的提升，特别是在抗干扰性能，电磁兼容和通信控制总线功能上独树一帜，其产品常用于工作环境恶劣的场合。亦适用于通信和家用电器控制领域。

还有台湾的 WINBOND 公司亦开发了一系列兼容 C51 的单片机，其产品通常具备丰富的功能特性，而且与其质优价廉在市场也占有一定的份额。

下表是 80C51 系列单片机的主要分类及功能特性：

系列	典型芯片	I/O 口	定时/计数器	中断源	串行通信口	片内 RAM	片内 ROM	说明
51 系列	80C31	4x8 位	2x16 位	5	1	128 字节	无	
	80C51	4x8 位	2x16 位	5	1	128 字节	4kB 掩膜 ROM	
	87C51	4x8 位	2x16 位	5	1	128 字节	4kBEPROM	
	89C51	4x8 位	2x16 位	5	1	128 字节	4kBEEPROM	
52 系列	80C32	4x8 位	2x16 位	6	1	256 字节	无	
	80C51	4x8 位	2x16 位	6	1	256 字节	8kB 掩膜 ROM	
	87C52	4x8 位	2x16 位	6	1	256 字节	4kBEPROM	

	89C52	4x8 位	2x16 位	6	1	256 字节	4kBEEPROM	
--	-------	-------	--------	---	---	--------	-----------	--

MCS-51 单片机主要应用特性（三）

8051 是 MCS-51 系列单片机中的代表产品，它内部集成了功能强大的中央处理器，包含了硬件乘除法器、21 个专用控制寄存器、4kB 的程序存储器、128 字节的数据存储器、4 组 8 位的并行口、两个 16 位的可编程定时/计数器、一个全双工的串行口以及布尔处理器。

MCS-51 具有比较大的寻址空间，地址线宽达 16 条，即外部数据存储器和程序存储器的寻址范围达 $2^{16}=64\text{kB}$ ，这作为单片机控制来说已是比较大的，这同时具备对 I/O 口的访问能力。此外，MCS-51 采用模块化结构，可方便地增删一个模块就可引脚和指令兼容的新产品。从而容易使产品形成系列化。

由于 MCS-51 集成了几乎完善的 8 位中央处理单元，处理功能强，中央处理单元中集成了方便灵活的专用寄存器，硬件的加、减、乘、除法和布尔处理机及各种逻辑运算和转移指令，这给应用提供了极大的便利。

MCS-51 的指令系统近乎完善，指令系统中包含了全面的数据传送指令、完善的算术和逻辑运算指令、方便的逻辑操作和控制指令、对于编程来说，是相当灵活和方便的。

MCS-51 单片机的工作频率为 2-12MHz，当振荡频率为 12MHz 时，一个机器周期为 1us,这个速度应该说是比较快的。

8051 中集成了完善的各种中断源，用户可十分方便地控制和使用其功能，使得它的应用范围加大，可以说它可以满足绝大部分的应用场合。

MCS-51 把微型计算机的主要部件都集成在一块芯片上，使得数据传送距离大大缩短，可靠性更高，运行速度更快。由于属于芯片化的微型计算机，各功能部件在芯片中的布局 and 结构达最优化，抗干扰能力加强，工作亦相对稳定。因此，在工业测控系统中，使用单片机是最理想的选择。单片机属于典型的嵌入式系统，所以它是低端控制系统最佳器件。

MCS-51 的开发环境要求较低，软件资源十分丰富，介绍其功能特性书籍和开发软件随处可取，只需配备一台 PC（个人电脑——对电脑的配置基本上无要求），一台仿真编程器即可实现产品开发，早期的开发软件多使用 DOS 版本，随着 Windows 视窗软件的普及，现在几乎都使用 Windows 版本，并且软件种类繁多，琳琅满目，在众多的单片机品种中，C51 的环境资源是最丰富的，这给 C51 用户带来极大的便利。

MCS-51CPU 和存储器

单片机 8051 的 CPU 由运算器和控制器组成。

一、运算器

运算器以完成二进制的算术/逻辑运算部件 ALU 为核心，再加上暂存器 TMP、累加器 ACC、寄存器 B、程序状态标志寄存器 PSW

及布尔处理器。累加器 ACC 是一个八位寄存器,它是 CPU 中最频繁的寄存器。在进行算术、逻辑运算时,累加器 ACC 往往在运算前暂存一个操作数(如被加数),而运算后又保存其结果(如代数和)。寄存器 B 主要用于乘法和除法操作。标志寄存器 PSW 也是一个八位寄存器,用来存放运算结果的一些特征,如有无进位、借位等。其每位的具体含意如下所示。PSW CY AC FO RS1 RS0 OV - P 对用户来讲,最关心的是以下四位。

1 进位标志 CY (PSW 7)。它表示了运算是否有进位(或借位)。如果操作结果在最高位有进位(加法)或者借位(减法),则该位为 1,否则为 0。

2 辅助进位标志 AC。又称半进位标志,它反映了两个八位数运算低四位是否有半进位,即低四位相加(或减)有否进位(或借位),如有则 AC 为 1 状态,否则为 0。

3 溢出标志位 OV。MCS - 51 反映带符号数的运算结果是否有溢出,有溢出时,此位为 1,否则为 0。

4 奇偶标志 P。反映累加器 ACC 内容的奇偶性,如果 ACC 中的运算结果有偶数个 1(如 11001100B,其中有 4 个 1),则 P 为 0,否则, P=1。

PSW 的其它位,将在以后再介绍。由于 PSW 存放程序执行中的状态,故又叫程序状态字。运算器中还有一个按位(bit)进行逻辑

运算的逻辑处理机(又称布尔处理机)。其功能在介绍位指令时再说明。

二、控制器

控制器是 CPU 的神经中枢, 它包括定时控制逻辑电路、指令寄存器、译码器、地址指针 DPTR 及程序计数器 PC、堆栈指针 SP 等。这里程序计数器 PC 是由 16 位寄存器构成的计数器。要单片机执行一个程序, 就必须把该程序按顺序预先装入存储器 ROM 的某个区域。单片机动作时应按顺序一条条取出指令来加以执行。因此, 必须有一个电路能找出指令所在的单元地址, 该电路就是程序计数器 PC。当单片机开始执行程序时, 给 PC 装入第一条指令所在地址, 它每取出一条指令(如为多字节指令, 则每取出一个指令字节), PC 的内容就自动加 1, 以指向下一条指令的地址, 使指令能顺序执行。只有当程序遇到转移指令、子程序调用指令, 或遇到中断时(后面将介绍), PC 才转到所需要的地方去。8051 CPU 确 C 指定的地址, 从 ROM 相应单元中取出指令字节放在指令寄存器中寄存, 然后, 指令寄存器中的指令代码被译码器译成各种形式的控制信号, 这些信号与单片机时钟振荡器产生的时钟脉冲在定时与控制电路中相结合, 形成按一定时间节拍变化的电平和时钟, 即所谓控制信息, 在 CPU 内部协调寄存器之间的数据传输、运算等操作。

三、存储器

存储器是单片机的又一个重要组成部分,图 6 给出了一种存储容量为 256 个单元的存储器结构示意图。其中每个存储单元对应一个地址,256 个单元共有 256 个地址,用两位 16 进制数表示,即存储器的地址(00H~FFH)。存储器中每个存储单元可存放一个八位二进制信息,通常用两位 16 进制数来表示,这就是存储器的内容。存储器的存储单元地址和存储单元的内容是不同的两个概念,不能混淆。

一、程序存储器

程序是控制计算机动作的一系列命令,单片机只认识由“0”和“1”代码构成的机器指令。如前述用助记符编写的命令 MOV A, #20H,换成机器认识的代码 74H、20H:(写成二进制就是 01110100B 和 00100000B)。在单片机处理问题之前必须事先将编好的程序、表格、常数汇编成机器代码后存入单片机的存储器中,该存储器称为程序存储器。程序存储器可以放在片内或片外,亦可片内片外同时设置。由于 PC 程序计数器为 16 位,使得程序存储器可用 16 位二进制地址,因此,内外存储器的地址最大可从 0000H 到 FFFFH。8051 内部有 4k 字节的 ROM,就占用了由 0000H~0FFFH 的最低 4k 个字节,这时片外扩充的程序存储器地址编号应由 1000H 开始,如果将 8051 当做 8031 使用,不想利用片内 4kROM,全用片外存储器,则地址编号仍可由 0000H 开始。不过,这时应使 8051 的第{31}脚(即 EA 脚)保持低电平。

当 EA 为高电平时，用户在 0000H 至 0FFFH 范围内使用内部 ROM，大于 0FFFH 后，单片机 CPU 自动访问外部程序存储器。

二、数据存储器

单片机的数据存储器由读写存储器 RAM 组成。其最大容量可扩展到 64k，用于存储实时输入的数据。8051 内部有 256 个单元的内部数据存储器，其中 00H ~ 7FH 为内部随机存储器 RAM，80H ~ FFH 为专用寄存器区。实际使用时应首先充分利用内部存储器，从使用角度讲，搞清内部数据存储器的结构和地址分配是十分重要的。因为将来在学习指令系统和程序设计时会经常用到它们。8051 内部数据存储器地址由 00H 至 FFH 共有 256 个字节的地址空间，该空间被分为两部分，其中内部数据 RAM 的地址为 00H ~ 7FH (即 0 ~ 127)。而用做特殊功能寄存器的地址为 80H ~ FFH。在此 256 个字节中，还开辟有一个所谓“位地址”区，该区域内不但可按字节寻址，还可按“位 (bit)”寻址。对于那些需要进行位操作的数据，可以存放到这个区域。从 00H 到 1FH 安排了四组工作寄存器，每组占用 8 个 RAM 字节，记为 R0 ~ R7。究竟选用那一组寄存器，由前述标志寄存器中的 RS1 和 RS0 来选用。在这两位上放入不同的二进制数，即可选用不同的寄存器组，如附表 1 所示。

三、特殊功能寄存器

特殊功能寄存器 (SFR) 的地址范围为 80H ~ FFH。在 MCS - 51 中,除程序计数器 PC 和四个工作寄存器区外,其余 21 个特殊功能寄存器都在这 SFR 块中。其中 5 个是双字节寄存器,它们共占用了 26 个字节。各特殊功能寄存器的符号和地址见附表 2。其中带 * 号的可位寻址。特殊功能寄存器反映了 8051 的状态,实际上是 8051 的状态字及控制字寄存器。用于 CPU PSW 便是典型一例。这些特殊功能寄存器大体上分为两类,一类与芯片的引脚有关,另一类作片内功能的控制用。与芯片引脚有关的特殊功能寄存器是 P0 ~ P3,它们实际上是 4 个八位锁存器(每个 I/O 口一个),每个锁存器附加有相应的输出驱动器和输入缓冲器就构成了一个并行口。MCS - 51 共有 P0 ~ P3 四个这样的并行口,可提供 32 根 I/O 线,每根线都是双向的,并且大都有第二功能。其余用于芯片控制的寄存器中,累加器 A、标志寄存器 PSW、数据指针 DPTR 等的功能前已提及,而另一些寄存器的功能在后面有关部分再作进一步介绍

MCS-51 单片机存储器结构

MCS-51 的存储器可分为四类:

程序存储器

一个微处理器能够聪明地执行某种任务,除了它们强大的硬件外,还需要它们运行的软件,其实微处理器并不聪明,它们只是完全按

照人们预先编写的程序而执行之。那么设计人员编写的程序就存放在微处理器的程序存储器中，俗称只读程序存储器（ROM）。程序相当于给微处理器处理问题的一系列命令。其实程序和数据一样，都是由机器码组成的代码串。只是程序代码则存放于程序存储器中。

MCS-51 具有 64kB 程序存储器寻址空间，它是用于存放用户程序、数据和表格等信息。对于内部无 ROM 的 8031 单片机，它的程序存储器必须外接，空间地址为 64kB，此时单片机的端必须接地。强制 CPU 从外部程序存储器读取程序。对于内部有 ROM 的 8051 等单片机，正常运行时，则需接高电平，使 CPU 先从内部的程序存储器中读取程序，当 PC 值超过内部 ROM 的容量时，才会转向外部的程序存储器读取程序。

8051 片内有 4kB 的程序存储单元，其地址为 0000H—0FFFH，单片机启动复位后，程序计数器的内容为 0000H，所以系统将从 0000H 单元开始执行程序。但在程序存储中有些特殊的单元，这在使用中应加以注意：

其中一组特殊是 0000H—0002H 单元，系统复位后，PC 为 0000H，单片机从 0000H 单元开始执行程序，如果程序不是从 0000H 单元开始，则应在这三个单元中存放一条无条件转移指令，让 CPU 直接去执行用户指定的程序。

另一组特殊单元是 0003H—002AH，这 40 个单元各有用途，它们被均匀地分为五段，它们的定义如下：

0003H—000AH 外部中断 0 中断地址区。

000BH—0012H 定时/计数器 0 中断地址区。

0013H—001AH 外部中断 1 中断地址区。

001BH—0022H 定时/计数器 1 中断地址区。

0023H—002AH 串行中断地址区。

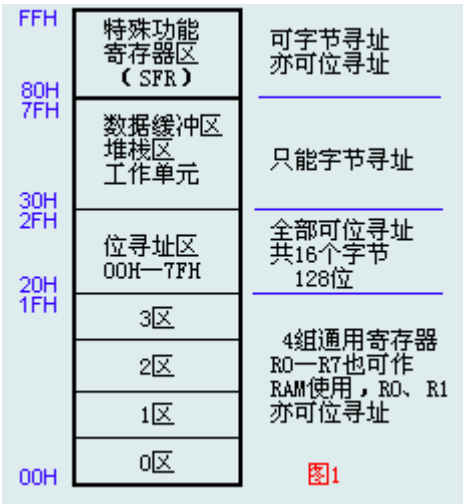
可见以上的 40 个单元是专门用于存放中断处理程序的地址单元，中断响应后，按中断的类型，自动转到各自的中断区去执行程序。因此以上地址单元不能用于存放程序的其他内容，只能存放中断服务程序。但是通常情况下，每段只有 8 个地址单元是不能存下完整的中断服务程序的，因而一般也在中断响应的地址区安放一条无条件转移指令，指向程序存储器的其它真正存放中断服务程序的空间去执行，这样中断响应后，CPU 读到这条转移指令，便转向其他地方去继续执行中断服务程序。

数据存储器

数据存储器也称为随机存取数据存储器。MCS-51 单片机的数据存储器在物理上和逻辑上都分为两个地址空间，一个是内部数据存储区和一个外部数据存储区。MCS-51 内部 RAM 有 128 或 256 个

字节的用户数据存储（不同的型号有分别），它们是用于存放执行的中间结果和过程数据的。MCS-51 的数据存储器均可读写，部分单元还可以位寻址。

8051 内部 RAM 共有 256 个单元，这 256 个单元共分为两部分。其一是地址从 00H—7FH 单元（共 128 个字节）为用户数据 RAM。从 80H—FFH 地址单元（也是 128 个字节）为特殊寄存器（SFR）单元。从图 1 中可清楚地看出它们的结构分布。



在 00H—1FH 共 32 个单元中被均匀地分为四块，每块包含八个 8 位寄存器，均以 R0—R7 来命名，我们常称这些寄存器为通用寄存器。这四块中的寄存器都称为 R0—R7，那么在程序中怎么区分和使用它们呢？聪明的 INTEL 工程师们又安排了一个寄存器——程序状态字寄存器（PSW）来管理它们，CPU 只要定义这个寄存的 PSW 的第 3 和第 4 位（RS0 和 RS1），即可选中这四组通用寄存器。对应的编码关系如图 2 所示。

PSW. 4 (RS1)	PSW. 3 (RS0)	工作寄存器区
0	0	0区00H—07H
0	1	1区08H—1FH
1	0	2区10H—17H
1	1	3区18H—1FH

图2 程序状态字与工作寄存器对应关系

内部 RAM 的 20H—2FH 单元为位寻址区，既可作为一般单元用字节寻址，也可对它们的位进行寻址。位寻址区共有 16 个字节，128 个位，位地址为 00H—7FH。位地址分配如表 1 所示，CPU 能直接寻址这些位，执行例如置“1”、清“0”、求“反”、转移，传送和逻辑等操作。我们常称 MCS-51 具有布尔处理功能，布尔处理的存储空间指的就是这些为寻址区。

表 1 RAM 位寻址区地址表

单元地址	MSB	位地址	LSB
2FH	7FH	7EH	7DH 7CH 7BH 7AH 79H 78H
2EH	77H	76H	75H 74H 73H 72H 71H 70H
2DH	6FH	6EH	6DH 6CH 6BH 6AH 69H 68H
2CH	67H	66H	65H 64H 63H 62H 61H 60H
2BH	5FH	5EH	5DH 5CH 5BH 5AH 59H 58H
2AH	57H	56H	55H 54H 53H 52H 51H 50H
29H	4FH	4EH	4DH 4CH 4BH 4AH 49H 48H
28H	47H	46H	45H 44H 43H 42H 41H 40H

27H 3FH 3EH 3DH 3CH 3BH 3AH 39H 38H
26H 37H 36H 35H 34H 33H 32H 31H 30H
25H 2FH 2EH 2DH 2CH 2BH 2AH 29H 28H
24H 27H 26H 25H 24H 23H 22H 21H 20H
23H 1FH 1EH 1DH 1CH 1BH 1AH 19H 18H
22H 17H 16H 15H 14H 13H 12H 11H 10H
21H 0FH 0EH 0DH 0CH 0BH 0AH 09H 08H
20H 07H 06H 05H 04H 03H 02H 01H 00H

特殊功能寄存器

特殊功能寄存器（SFR）也称为专用寄存器，特殊功能寄存器反映了 MCS-51 单片机的运行状态。很多功能也通过特殊功能寄存器来定义和控制程序的执行。

MCS-51 有 21 个特殊功能寄存器，它们被离散地分布在内部 RAM 的 80H—FFH 地址中，这些寄存器的功能已作了专门的规定，用户不能修改其结构。表 2 是特殊功能寄存器分布一览表，我们对其主要的寄存器作一些简单的介绍。

表 2 特殊功能寄存器

标识符号	地址	寄存器名称
------	----	-------

ACC	0E0H	累加器
-----	------	-----

B	0F0H	B 寄存器
---	------	-------

PSW	0D0H	程序状态字
-----	------	-------

SP 81H 堆栈指针

DPTR 82H、83H 数据指针 (16 位) 含 DPL 和 DPH

IE 0A8H 中断允许控制寄存器

IP 0B8H 中断优先控制寄存器

P0 80H I/O 口 0 寄存器

P1 90H I/O 口 1 寄存器

P2 0A0H I/O 口 2 寄存器

P3 0B0H I/O 口 3 寄存器

PCON 87H 电源控制及波特率选择寄存器

SCON 98H 串行口控制寄存器

SBUF 99H 串行数据缓冲寄存器

TCON 88H 定时控制寄存器

TMOD 89H 定时器方式选择寄存器

TL0 8AH 定时器 0 低 8 位

TH0 8CH 定时器 0 高 8 位

TL1 8BH 定时器 1 低 8 位

TH1 8DH 定时器 1 高 8 位

程序计数器 PC(program Counter)

程序计数器在物理上是独立的，它不属于特殊内部数据存储器块中。PC 是一个 16 位的计数器，用于存放一条要执行的指令地址，寻址范围为 64kB，PC 有自动加 1 功能，即完成了一条指令的执

行后，其内容自动加 1。PC 本身并没有地址，因而不可寻址，用户无法对它进行读写，但是可以通过转移、调用、返回等指令改变其内容，以控制程序按我们的要求去执行。

累加器 ACC(Accumulator)

累加器 A 是一个最常用的专用寄存器，大部分单操作指令的一个操作数取自累加器，很多双操作数指令中的一个操作数也取自累加器。加、减、乘、除法运算的指令，运算结果都存放于累加器 A 或 AB 累加器对中。大部分的数据操作都会通过累加器 A 进行，它形象于一个交通要道，在程序比较复杂的运算中，累加器成了制约软件效率的“瓶颈”，它的功能较多，地位也十分重要。以至于后来发展的单片机，有的集成了多累加器结构，或者使用寄存器阵列来代替累加器，即赋予更多寄存器以累加器的功能，目的是解决累加器的“交通堵塞”问题。提高单片机的软件效率。

寄存器 B

在乘除法指令中，乘法指令中的两个操作数分别取自累加器 A 和寄存器 B，其结果存放于 AB 寄存器对中。除法指令中，被除数取自累加器 A，除数取自寄存器 B，结果商存放于累加器 A，余数存放于寄存器 B 中。

程序状态字(Program Status Word)

程序状态字是一个 8 位寄存器 ,用于存放程序运行的状态信息 ,这个寄存器的一些位可由软件设置 ,有些位则由硬件运行时自动设置的。寄存器的各位定义如下 ,其中 PSW.1 是保留位 ,未使用。下表是它的功能说明 ,并对各个位的定义介绍如下 :

表 3 程序状态字

位序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	F0	RS1	RS0	OV	-	P

PSW.7 (CY) 进位标志位 ,此位有两个功能 :一是存放执行某写
算数运算时 ,存放进位标志 ,可被硬件或软件置位或清零。二是在
位操作中作累加位使用。

PSW.6 (AC) 辅助进位标志位 ,当进行加、减运算时当有低 4 位
向高 4 位进位或借位时 ,AC 置位 ,否则被清零。AC 辅助进位位
也常用于十进制调整。

PSW.5 (F0) 用户标志位 ,供用户设置的标志位。

PSW.4、PSW.3 (RS1 和 RS0) 寄存器组选择位。可参见本章的
图 2 定义。

PSW.2 (OV) 溢出标志。带符号加减运算中，超出了累加器 A 所能表示的符号数有效范围 (-128—+127) 时，即产生溢出，OV=1。表明运算结果错误。如果 OV=0，表明运算结果正确。执行加法指令 ADD 时，当位 6 向位 7 进位，而位 7 不向 C 进位时，OV=1。或者位 6 不向位 7 进位，而位 7 向 C 进位时，同样 OV=1。

除法指令，乘积超过 255 时，OV=1。表面乘积在 AB 寄存器对中。

若 OV=0，则说明乘积没有超过 255，乘积只在累加器 A 中。

除法指令，OV=1，表示除数为 0，运算不被执行。否则 OV=0。

PSW.0 (P) 奇偶校验位。声明累加器 A 的奇偶性，每个指令周期都由硬件来置位或清零，若值为 1 的位数奇数，则 P 置位，否则清零。

数据指针(DPTR)

数据指针为 16 位寄存器，编程时，既可以按 16 位寄存器来使用，也可以按两个 8 位寄存器来使用，即高位字节寄存器 DPH 和低位字节 DPL。

DPTR 主要是用来保存 16 位地址，当对 64kB 外部数据存储器寻址时，可作为间址寄存器使用，此时，使用如下两条指令：

```
MOVX A, @DPTR
```

```
MOVX @DPTR, A
```

在访问程序存储器时，DPTR 可用来作基址寄存器，采用基址+变址寻址方式访问程序存储器，这条指令常用于读取程序存储器内的表格数据。

MOVC A, @A+@DPTR

堆栈指针 SP(Stack Pointer)

堆栈是一种数据结构，它是一个 8 位寄存器，它指示堆栈顶部在内部 RAM 中的位置。系统复位后，SP 的初始值为 07H，使得堆栈实际上是从 08H 开始的。但从 RAM 的结构分布中可知，08H—1FH 隶属 1—3 工作寄存器区，若编程时需要用到这些数据单元，必须对堆栈指针 SP 进行初始化，原则上设在任何一个区域均可，但一般设在 30H—1FH 之间较为适宜。

数据的写入堆栈我们称为入栈（PUSH，有些文献也称作插入运算或压入），从堆栈中取出数据称为出栈（POP，也称为删除运算或弹出），堆栈的最主要特征是“后进先出”规则，也即最先入栈的数据放在堆栈的最底部，而最后入栈的数据放在栈的顶部，因此，最后入栈的数据出栈时则是最先的。这和我们往一个箱里存放书本一样，需将最先放入箱底部的书取出，必须先取走最上层的书籍。这个道理非常相似。

那么堆栈有何用途呢？堆栈的设立是为了中断操作和子程序的调用而用于保存数据的，即常说的断点保护和现场保护。微处理器无论是在转入子程序和中断服务程序的执行，执行完后，还是要回到

主程序中来，在转入子程序和中断服务程序前，必须先将现场的数据进行保存起来，否则返回时，CPU 并不知道原来的程序执行到哪一步，原来的中间结果如何？所以在转入执行其它子程序前，先将需要保存的数据压入堆栈中保存。以备返回时，再复原当时的数据。供主程序继续执行。

转入中断服务程序或子程序时，需要保存的数据可能有若干个，都需要一一地保留。如果微处理器进行多重子程序或中断服务程序嵌套，那么需保存的数据就更多，这要求堆栈还需要有相当的容量。否则会造成堆栈溢出，丢失应备份的数据。轻者使运算和执行结果错误，重则使整个程序紊乱。

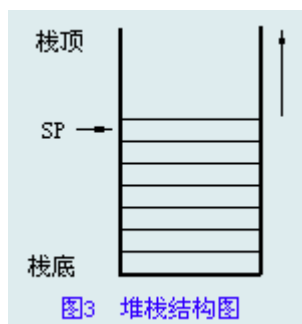
MCS-51 的堆栈是在 RAM 中开辟的，即堆栈要占据一定的 RAM 存储单元。同时 MCS-51 的堆栈可以由用户设置，SP 的初始值不同，堆栈的位置则不一定，不同的设计人员，使用的堆栈区则不同，不同的应用要求，堆栈要求的容量也有所不同。堆栈的操作只有两种，即进栈和出栈，但不管是向堆栈写入数据还是从堆栈中读出数据，都是对栈顶单元进行的，SP 就是即时指示出栈顶的位置（即地址）。在子程序调用和中断服务程序响应的开始和结束期间，CPU 都是根据 SP 指示的地址与相应的 RAM 存储单元交换数据。

堆栈的操作有两种方法：其一是自动方式，即在中断服务程序响应或子程序调用时，返回地址自动进栈。当需要返回执行主程序时，返回的地址自动交给 PC，以保证程序从断点处继续执行，这种方

式是不需要编程人员干预的。第二种方式是人工指令方式，使用专用的堆栈操作指令进行进出栈操作，也只有两条指令：进栈为 PUSH 指令，在中断服务程序或子程序调用时作为现场保护。出栈操作 POP 指令，用于子程序完成时，为主程序恢复现场。

I/O 口专用寄存器(P0、P1、P2、P3)

I/O 口寄存器 P0、P1、P2 和 P3 分别是 MCS-51 单片机的四组 I/O 口锁存器。MCS-51 单片机并没有专门的 I/O 口操作指令，而是把 I/O 口也当作一般的寄存器来使用，数据传送都统一使用 MOV 指令来进行，这样的好处在于，四组 I/O 口还可以当作寄存器直接寻址方式参与其他操作。



定时/计数器(TL0、TH0、TL1 和 TH1)

MCS-51 单片机中有两个 16 位的定时/计数器 T0 和 T1，它们由四个 8 位寄存器组成的，两个 16 位定时/计数器却是完全独立的。我们可以单独对这四个寄存器进行寻址，但不能把 T0 和 T1 当作 16 位寄存来使用。

定时/计数器方式选择寄存器(TMOD)

TMOD 寄存器是一个专用寄存器，用于控制两个定时计数器的工作方式，TMOD 可以用字节传送指令设置其内容，但不能位寻址，各位的定义如下，更详细的内容，我们将在《MCS-51 定时器和中断系统》章节中叙述。

表 4 定时/计数器工作方式控制寄存器 TMOD

位序	D7	D6	D5	D4	D3	D2	D1	D0
位标志	GATE	M1	M0	GATE	M1	M0		
定时/计数器	1							0

串行数据缓冲器(SBUF)

串行数据缓冲器 SBUF 用来存放需发送和接收的数据，它由两个独立的寄存器组成，一个是发送缓冲器，另一个是接收缓冲器，要发送和接收的操作其实都是对串行数据缓冲器进行。

其他控制寄存器(TMOD)

除了以上我们简述的几个专用寄存外，还有 IP、IE、TCON、SCON 和 PCON 等几个寄存器，这几个控制寄存器主要用于中断和定时的，我们将在《MCS-51 定时器和中断系统》中详细说明。

MCS-51 单片机的指令时序

时序是用定时单位来描述的，MCS-51 的时序单位有四个，它们分别是节拍、状态、机器周期和指令周期，接下来我们分别加以说明。

·节拍与状态：

我们把振荡脉冲的周期定义为节拍（为方便描述，用 P 表示），振荡脉冲经过二分频后即得到整个单片机工作系统的时钟信号，把时钟信号的周期定义为状态（用 S 表示），这样一个状态就有两个节拍，前半周期相应的节拍我们定义为 1(P1)，后半周期对应的节拍定义为 2(P2)。

·机器周期：

MCS-51 有固定的机器周期，规定一个机器周期有 6 个状态，分别表示为 S1-S6，而一个状态包含两个节拍，那么一个机器周期就有 12 个节拍，我们可以记着 S1P1、S1P2.....S6P1、S6P2，一个机器周期共包含 12 个振荡脉冲，即机器周期就是振荡脉冲的 12 分频，显然，如果使用 6MHz 的时钟频率，一个机器周期就是 2us，而如使用 12MHz 的时钟频率，一个机器周期就是 1us。

·指令周期：

执行一条指令所需要的时间称为指令周期，MCS-51 的指令有单字节、双字节和三字节的，所以它们的指令周期不尽相同，也就是说它们所需的机器周期不相同，可能包括一到四个不等的机器周期（这些内容，我们将在下面的章节中加以说明）。

·MCS-51 的指令时序：

MCS-51 指令系统中,按它们的长度可分为单字节指令、双字节指令和三字节指令。执行这些指令需要的时间是不同的,也就是它们所需的机器周期是不同的,有下面几种形式：

·单字节指令单机器周期

·单字节指令双机器周期

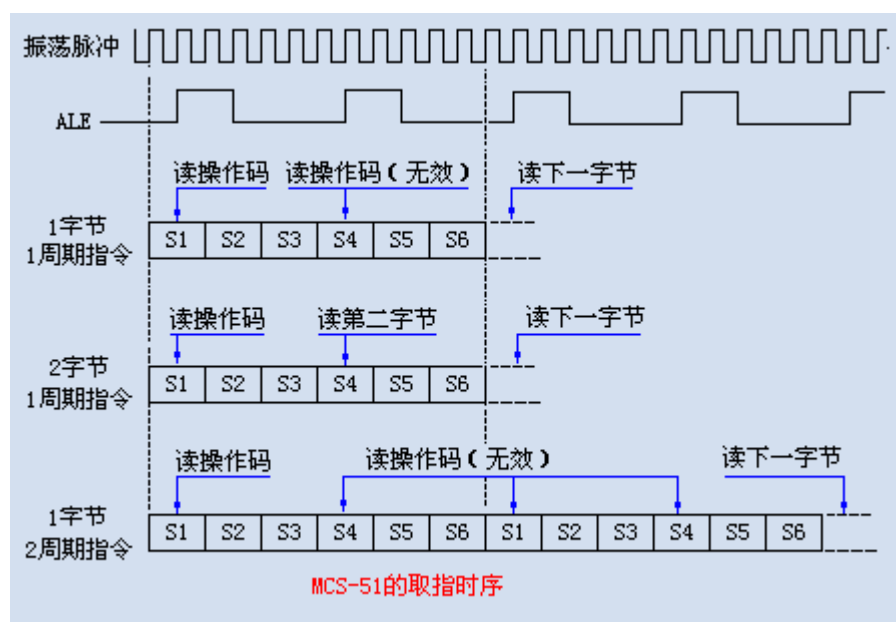
·双字节指令单机器周期

·双字节指令双机器周期

·三字节指令双机器周期

·单字节指令四机器周期(如单字节的乘除法指令)

下图是 MCS-51 系列单片机的指令时序图：



上图是单周期和双周期取指及执行时序，图中的 ALE 脉冲是为了

锁存地址的选通信号，显然，每出现一次该信号单片机即进行一次读指令操作。从时序图中可看出，该信号是时钟频率 6 分频后得到，在一个机器周期中，ALE 信号两次有效，第一次在 S1P2 和 S2P1 期间，第二次在 S4P2 和 S5P1 期间。

接下来我们分别对几个典型的指令时序加以说明。

·单字节单周期指令：

单字节单周期指令只进行一次读指令操作，当第二个 ALE 信号有效时，PC 并不加 1，那么读出的还是原指令，属于一次无效的读操作。

·双字节单周期指令：

这类指令两次的 ALE 信号都是有效的，只是第一个 ALE 信号有效时读的是操作码，第二个 ALE 信号有效时读的是操作数。

·单字节双周期指令：

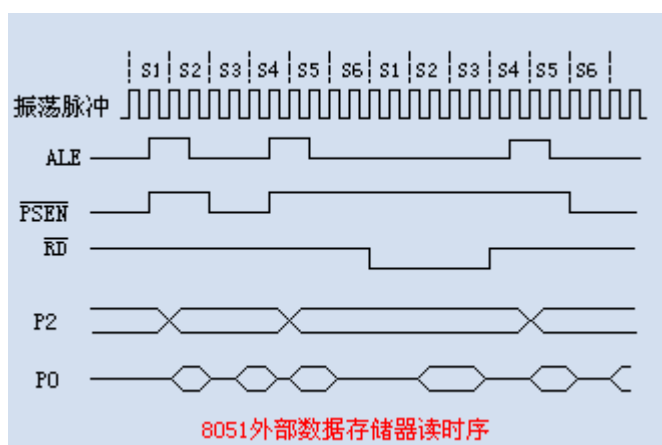
两个机器周期需进行四次读指令操作，但只有一次读操作是有效的，后三次的读操作均为无效操作。

单字节双周期指令有一种特殊的情况，象 MOVX 这类指令，执行这类指令时，先在 ROM 中读取指令，然后对外部数据存储器进行读或写操作，头一个机器周期的第一次读指令的操作码为有效，而第二次读指令操作则为无效的。在第二个指令周期时，则访问外部

数据存储器，这时，ALE 信号对其操作无影响，即不会再有读指令操作动作。

上页的时序图中，我们只描述了指令的读取状态，而没有画出指令执行时序，因为每条指令都包含了具体的操作数，而操作数类型种类繁多，这里不便列出，有兴趣的读者可参阅有关书籍。

·外部程序存储器(ROM)读时序



右图 8051 外部程序存储器读时序图，从图中可看出，P0 口提供低 8 位地址，P2 口提供高 8 位地址，S2 结束前，P0 口上的低 8 位地址是有效的，之后出现在 P0 口上的就不再是低 8 位的地址信号，而是指令数据信号，当然地址信号与指令数据信号之间有一段缓冲的过度时间，这就要求，在 S2 其间必须把低 8 位的地址信号锁存起来，这时是用 ALE 选通脉冲去控制锁存器把低 8 位地址予以锁存，而 P2 口只输出地址信号，而没有指令数据信号，整个机器周期地址信号都是有效的，因而无需锁存这一地址信号。

从外部程序存储器读取指令，必须有两个信号进行控制，除了上述的 ALE 信号，还有一个 PSEN（外部 ROM 读选通脉冲），上图显

然可看出，PSEN 从 S3P1 开始有效，直到将地址信号送出和外部程序存储器的数据读入 CPU 后方才失效。而又从 S4P2 开始执行第二个读指令操作。

·外部数据存储器(RAM)读时序

右图 8051 外部数据存储器读写时序图，从 ROM 中读取的需执行的指令，而 CPU 对外部数据存储的访问是对 RAM 进行数据的读或写操作，属于指令的执行周期，值得一提的是，读或写是两个不同的机器周期，但他们的时序却是相似的，我们只对 RAM 的读时序进行分析。

上一个机器周期是取指阶段，是从 ROM 中读取指令数据，接着的下个周期才开始读取外部数据存储器 RAM 中的内容。

在 S4 结束后，先把需读取 RAM 中的地址放到总线上，包括 P0 口上的低 8 位地址 A0-A7 和 P2 口上的高 8 位地址 A8-A15。当 RD 选通脉冲有效时，将 RAM 的数据通过 P0 数据总线读进 CPU。第二个机器周期的 ALE 信号仍然出现，进行一次外部 ROM 的读操作，但是这一次的读操作属于无效操作。

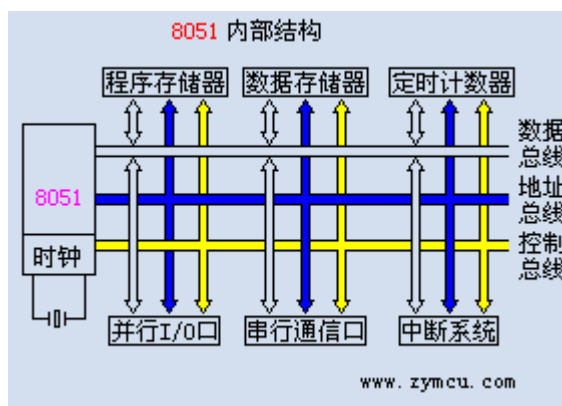
对外部 RAM 进行写操作时，CPU 输出的则是 WR(写选通信号)，将数据通过 P0 数据总线写入外部存储中。

MCS-51 单片机内部结构

8051 是 MCS-51 系列单片机的典型产品，我们以这一代表性的机

型进行系统的讲解。

8051 单片机包含中央处理器、程序存储器(ROM)、数据存储器(RAM)、定时/计数器、并行接口、串行接口和中断系统等几大单元及数据总线、地址总线和控制总线等三大总线，现在我们分别加以说明：



·中央处理器：

中央处理器(CPU)是整个单片机的核心部件，是 8 位数据宽度的处理器，能处理 8 位二进制数据或代码，CPU 负责控制、指挥和调度整个单元系统协调的工作，完成运算和控制输入输出功能等操作。

·数据存储器(RAM)：

8051 内部有 128 个 8 位用户数据存储单元和 128 个专用寄存器单元，它们是统一编址的，专用寄存器只能用于存放控制指令数据，用户只能访问，而不能用于存放用户数据，所以，用户能使用的 RAM 只有 128 个，可存放读写的数据，运算的中间结果或用户定义的字型表。

·程序存储器(ROM)：

8051 共有 4096 个 8 位掩膜 ROM，用于存放用户程序，原始数据或表格。

·定时/计数器(TIMER)：

8051 有两个 16 位的可编程定时/计数器，以实现定时或计数产生中断用于控制程序转向。

·并行输入输出(I/O)口：

8051 共有 4 组 8 位 I/O 口(P0、 P1、 P2 或 P3)，用于对外部数据的传输。

·全双工串行口：

8051 内置一个全双工串行通信口，用于与其它设备间的串行数据传送，该串行口既可以用作异步通信收发器，也可以当同步移位器使用。

·中断系统：

8051 具备较完善的中断功能，有两个外中断、两个定时/计数器中断和一个串行中断，可满足不同的控制要求，并具有 2 级的优先级别选择。

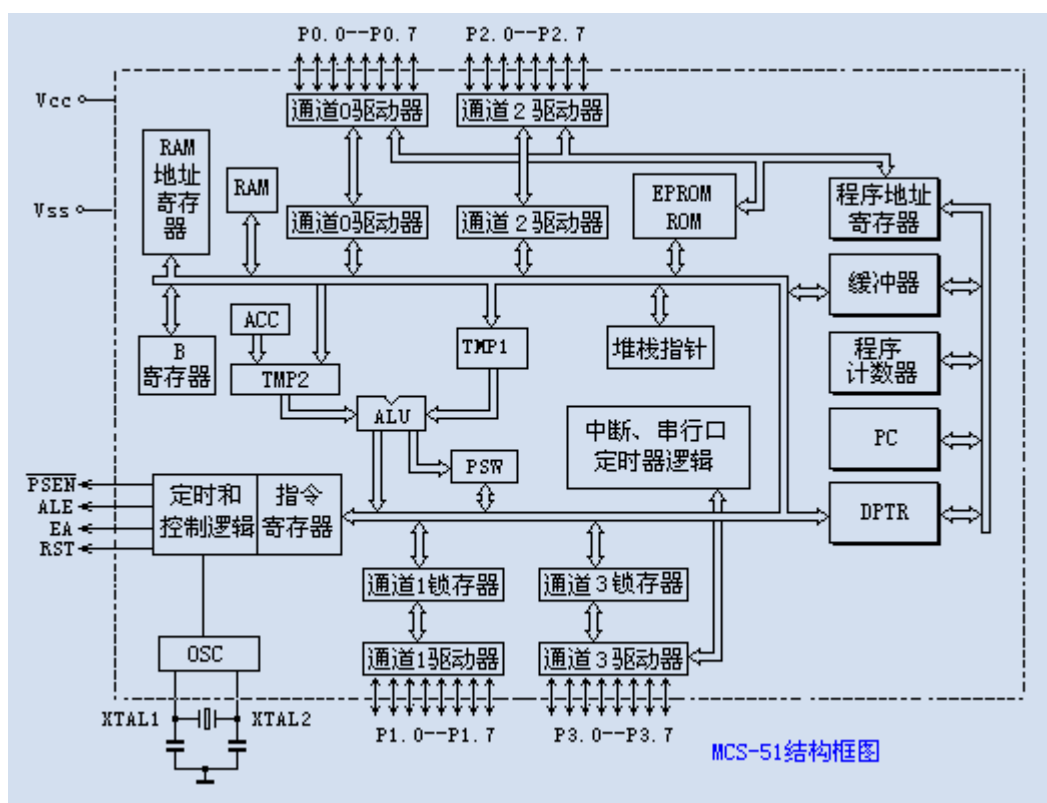
·时钟电路：

8051 内置最高频率达 12MHz 的时钟电路，用于产生整个单片机

运行的脉冲时序，但 8051 单片机需外置振荡电容。

单片机的结构有两种类型，一种是程序存储器和数据存储器分开
形式，即哈佛(Harvard)结构，另一种是采用通用计算机广泛使用的
程序存储器与数据存储器合二为一的结构，即普林斯顿
(Princeton)结构。INTEL 的 MCS-51 系列单片机采用的是哈佛结
构的形式，而后续产品 16 位的 MCS-96 系列单片机则采用普林斯
顿结构。

下图是 MCS-51 系列单片机的内部结构示意图。



■ MCS-51 的引脚说明：

MCS-51 系列单片机中的 8031、8051 及 8751 均采用 40Pin 封

装的双列直接 DIP 结构，右图是它们的引脚配置，40 个引脚中，正电源和地线两根，外置石英振荡器的时钟线两根，4 组 8 位共 32 个 I/O 口，中断口线与 P3 口线复用。现在我们对这些引脚的功能加以说明：

P1.0	1	40	V _{CC}
P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RD/P3.0	10	31	EA/V _{PP}
WD/P3.1	11	30	ALE/PROG
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
GND	20	21	P2.0/A8

PDIP

·Pin20:接地脚。

·Pin40:正电源脚，正常工作或对片内 EPROM 烧写程序时，接+5V 电源。

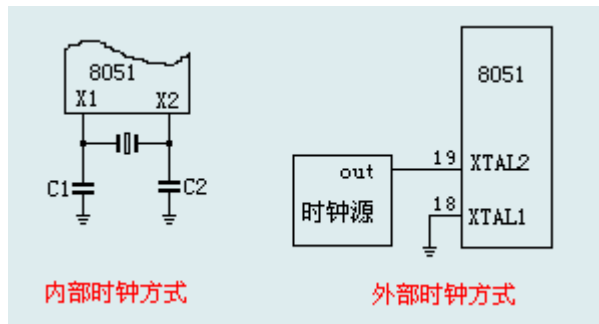
·Pin19:时钟 XTAL1 脚，片内振荡电路的输入端。

·Pin18:时钟 XTAL2 脚，片内振荡电路的输出端。

8051 的时钟有两种方式，一种是片内时钟振荡方式，但需在 18 和 19 脚外接石英晶体(2-12MHz)和振荡电容，振荡电容的值一般取 10p-30p。另外一种外部时钟方式，即将 XTAL1 接地，外部时钟信号从 XTAL2 脚输入。

·输入输出(I/O)引脚：

Pin39-Pin32 为 P0.0-P0.7 输入输出脚 , Pin1-Pin1 为 P1.0-P1.7 输入输出脚 , Pin21-Pin28 为 P2.0-P2.7 输入输出脚 , Pin10-Pin17 为 P3.0-P3.7 输入输出脚 , 这些输入输出脚的功能说明将在以下内容阐述。



·Pin9:RESET/V_{pd} 复位信号复用脚 , 当 8051 通电 , 时钟电路开始工作 , 在 RESET 引脚上出现 24 个时钟周期以上的高电平 , 系统即初始复位。初始化后 , 程序计数器 PC 指向 0000H , P0-P3 输出口全部为高电平 , 堆栈指针写入 07H , 其它专用寄存器被清 “0” 。 RESET 由高电平下降为低电平后 , 系统即从 0000H 地址开始执行程序。然而 , 初始复位不改变 RAM (包括工作寄存器 R0-R7) 的状态 , 8051 的初始态如下表 :

特殊功能寄存器 初始态 特殊功能寄存器 初始态

ACC 00H B 00H

PSW 00H SP 07H

DPH 00H TH0 00H

DPL 00H TL0 00H

IP xxx00000B TH1 00H

IE 0xx00000B TL1 00H

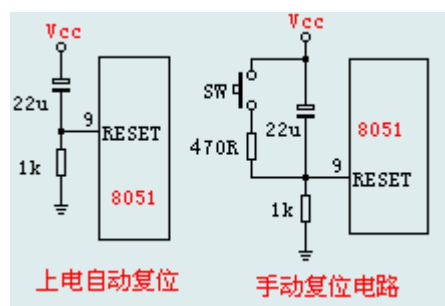
TMOD 00H TCON 00H

SCON xxxxxxxxB SBUF 00H

P0-P3 1111111B PCON 0xxxxxxxB

8051 的复位方式可以是自动复位，也可以是手动复位，见下图。

此外，RESET/Vpd 还是一复用脚，Vcc 掉电期间，此脚可接上备用电源，以保证单片机内部 RAM 的数据不丢失。



·Pin30:ALE/当访问外部程序器时，ALE(地址锁存)的输出用于锁存地址的低位字节。而访问内部程序存储器时，ALE 端将有一个 1/6 时钟频率的正脉冲信号，这个信号可以用于识别单片机是否工作，也可以当作一个时钟向外输出。更有一个特点，当访问外部程序存储器，ALE 会跳过一个脉冲。

如果单片机是 EPROM，在编程其间，将用于输入编程脉冲。

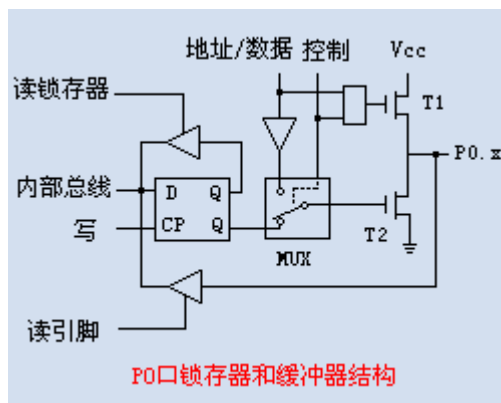
·Pin29:当访问外部程序存储器时，此脚输出负脉冲选通信号，PC 的 16 位地址数据将出现在 P0 和 P2 口上，外部程序存储器则把指令数据放到 P0 口上，由 CPU 读入并执行。

·Pin31:EA/Vpp 程序存储器的内外部选通线，8051 和 8751 单片机，内置有 4kB 的程序存储器，当 EA 为高电平并且程序地址小于 4kB 时，读取内部程序存储器指令数据，而超过 4kB 地址则读取外部指令数据。如 EA 为低电平，则不管地址大小，一律读取外部程序存储器指令。显然，对内部无程序存储器的 8031,EA 端必须接地。

在编程时，EA/Vpp 脚还需加上 21V 的编程电压。

MCS-51 单片机输入输出口

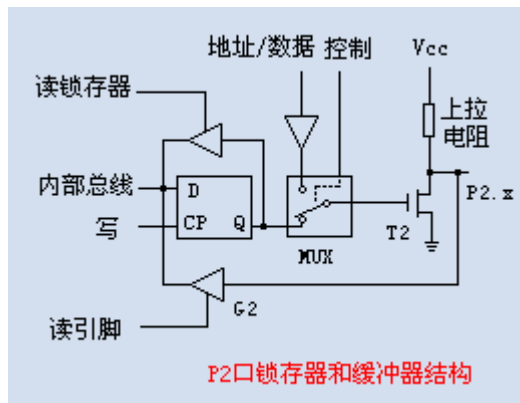
8051 有 4 组 8 位 I/O 口：P0、P1、P2 和 P3 口，P1、P2 和 P3 为准双向口，P0 口则为双向三态输入输出口，下面我们分别介绍这几个口线：



·P0 口和 P2 口：

右图为 P0 口和 P2 口其中一位的电路图，由图可见，电路中包含一个数据输出锁存器和两个三态数据输入缓冲器，另外还有一个数

据输出的驱动和控制电路。这两组口线用来作为 CPU 与外部数据存储器、外部程序存储器和 I/O 扩展口，而不能象 P1、P3 直接用作输出口。它们一起可以作为外部地址总线，P0 口身兼两职，既可作为地址总线，也可作为数据总线。参考图 2。



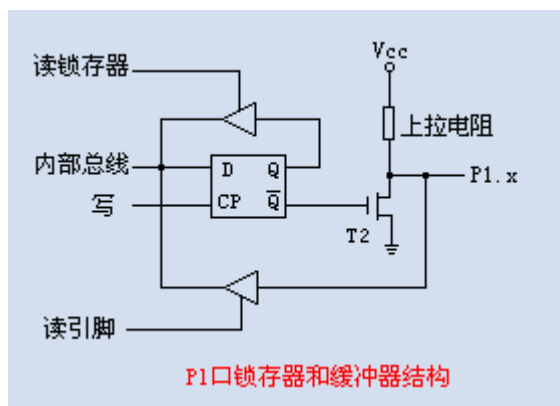
P2 口作为外部数据存储器或程序存储器的地址总线的高 8 位输出口 AB8-AB15，P0 口由 ALE 选通作为地址总线的低 8 位输出口 AB0-AB7。外部的程序存储器由 PSEN 信号选通，数据存储器则由 WR 和 RD 读写信号选通，因为 $2^{16}=64k$ ，所以 8051 最大可外接 64kB 的程序存储器和数据存储器。

·P1 口：

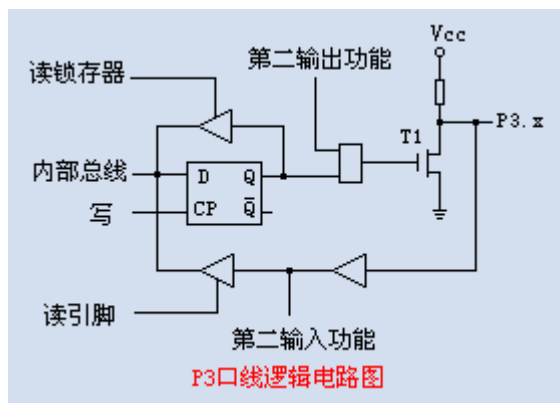
右图为 P1 口其中一位的电路图，P1 口为 8 位准双向口，每一位均可单独定义为输入或输出口，当作为输入口时，1 写入锁存器， $Q(\text{非})=0$ ，T2 截止，内上拉电阻将电位拉至"1"，此时该口输出为 1，当 0 写入锁存器， $Q(\text{非})=1$ ，T2 导通，输出则为 0。

作为输入口时，锁存器置 1， $Q(\text{非})=0$ ，T2 截止，此时该位既可以把外部电路拉成低电平，也可由内部上拉电阻拉成高电平，正因为

这个原因，所以 P1 口常称为准双向口。需要说明的是，作为输入口使用时，有两种情况，其一是：首先是读锁存器的内容，进行处理后再写到锁存器中，这种操作即读—修改—写操作，象 JBC(逻辑判断)、CPL(取反)、INC(递增)、DEC(递减)、ANL(与逻辑)和 ORL(逻辑或)指令均属于这类操作。其二是：读 P1 口线状态时，打开三态门 G2,将外部状态读入 CPU。



·P3 口：



P3 口的电路如上图所示，P3 口为准双向口，为适应引脚的第二功能的需要，增加了第二功能控制逻辑，在真正的应用电路中，第二功能显得更为重要。由于第二功能信号有输入输出两种情况，我们分别加以说明。

P3 口的输入输出及 P3 口锁存器、中断、定时/计数器、串行口和特殊功能寄存器有关，P3 口的第一功能和 P1 口一样可作为输入输出端口，同样具有字节操作和位操作两种方式，在位操作模式下，每一位均可定义为输入或输出。

我们着重讨论 P3 口的第二功能，P3 口的第二功能各管脚定义如下：

- P3.0 串行输入口(RXD)
- P3.1 串行输出口(TXD)
- P3.2 外中断 0(INT0)
- P3.3 外中断 1(INT1)
- P3.4 定时/计数器 0 的外部输入口(T0)
- P3.5 定时/计数器 1 的外部输入口(T1)
- P3.6 外部数据存储器写选通(WR)
- P3.7 外部数据存储器读选通(RD)

对于第二功能为输出引脚，当作 I/O 口使用时，第二功能信号线应保持高电平，与非门开通，以维持从锁存器到输出口数据输出通路畅通无阻。而当作第二功能口线使用时，该位的锁存器置高电平，使与非门对第二功能信号的输出是畅通的，从而实现第二功能信号的输出。对于第二功能为输入的信号引脚，在口线上的输入通路增设了一个缓冲器，输入的第二功能信号即从这个缓冲器的输出端取得。而作为 I/O 口线输入端时，取自三态缓冲器的输出端。这样，

不管是作为输入口使用还是第二功能信号输入,输出电路中的锁存器输出和第二功能输出信号线均应置“1”。

MCS-51 的串行通信口

MCS-51 单片机内部有一个全双工的串行通信口,即串行接收和发送缓冲器(SBUF),这两个在物理上独立的接收发送器,既可以接收数据也可以发送数据。但接收缓冲器只能读出不能写入,而发送缓冲器则只能写入不能读出,它们的地址为 99H。这个通信口既可以用于网络通信,亦可实现串行异步通信,还可以构成同步移位寄存器使用。如果在串行口的输入输出引脚上加上电平转换器,就可方便地构成标准的 RS-232 接口。下面我们分别介绍。

[1]. 基本概念

数据通信的传输方式

常用于数据通信的传输方式有单工、半双工、全双工和多工方式。

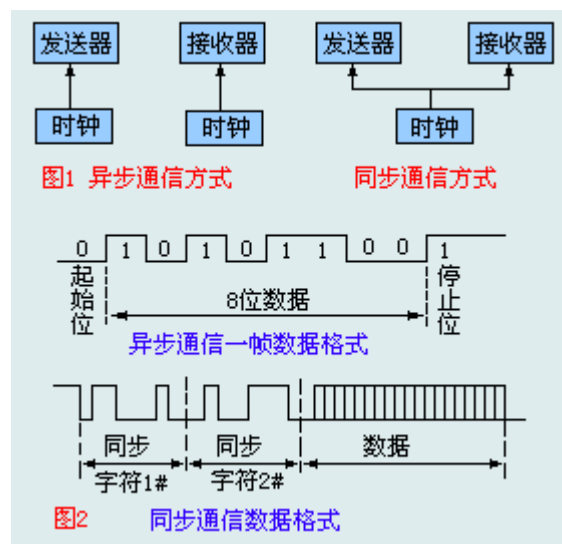
单工方式:数据仅按一个固定方向传送。因而这种传输方式的用途有限,常用于串行口的打印数据传输与简单系统间的数据采集。

半双工方式:数据可实现双向传送,但不能同时进行,实际的应用采用某种协议实现收/发开关转换。

全双工方式:允许双方同时进行数据双向传送,但一般全双工传输方式的线路和设备较复杂。

多工方式：以上三种传输方式都是用同一线路传输一种频率信号，为了充分地利用线路资源，可通过使用多路复用器或多路集线器，采用频分、时分或码分复用技术，即可实现在同一线路上资源共享功能，我们称之为多工传输方式。

串行数据通信两种形式



异步通信

在这种通信方式中，接收器和发送器有各自的时钟，它们的工作是非同步的，异步通信用一帧来表示一个字符，其内容如下：一个起始位，仅接着是若干个数据位，图 2 是传输 45H 的数据格式。

同步通信

同步通信格式中，发送器和接收器由同一个时钟源控制，为了克服在异步通信中，每传输一帧字符都必须加上起始位和停止位，占用了传输时间，在要求传送数据量较大的场合，速度就慢得多。同步

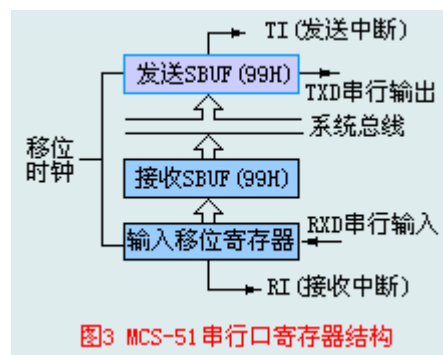
传输方式去掉了这些起始位和停止位,只在传输数据块时先送出一个同步头(字符)标志即可。

同步传输方式比异步传输方式速度快,这是它的优势。但同步传输方式也有其缺点,即它必须要用一个时钟来协调收发器的工作,所以它的设备也较复杂。

串行数据通信的传输速率

串行数据传输速率有两个概念,即每秒转送的位数 bps (Bit per second) 和每秒符号数—波特率 (Band rate),在具有调制解调器的通信中,波特率与调制速率有关。

[2]. MCS-51 的串行口和控制寄存器



串行口控制寄存器

MCS-51 单片机串行口寄存器结构如图 3 所示。SBUF 为串行口的收发缓冲器,它是一个可寻址的专用寄存器,其中包含了接收器和发送器寄存器,可以实现全双工通信。但这两个寄存器具有同一地

址 (99H)。MCS-51 的串行数据传输很简单，只要向发送缓冲器写入数据即可发送数据。而从接收缓冲器读出数据即可接收数据。

此外，从图中可看出，接收缓冲器前还加上一级输入移位寄存器，MCS-51 这种结构目的在于接收数据时避免发生数据帧重叠现象，以免出错，部分文献称这种结构为双缓冲器结构。而发送数据时就不需要这样设置，因为发送时，CPU 是主动的，不可能出现这种现象。

串行通信控制寄存器

在上一节我们已经分析了 SCON 控制寄存器，它是一个可寻址的专用寄存器，用于串行数据的通信控制，单元地址是 98H，其结构格式如下：

表 1 SCON 寄存器结构

SCON D7 D6 D5 D4 D3 D2 D1 D0

SM0 SM1 SM2 REN TB8 RB8 TI RI

位地址 9FH 9EH 8DH 9CH 9BH 9AH 99H 98H

下面我们对各控制位功能介绍如下：

(1).SM0、SM1：串行口工作方式控制位。

SM0，SM1 工作方式

00 方式 0

01 方式 1

10 方式 2

11 方式 3

(2).SM2：多机通信控制位。

多机通信是工作于方式 2 和方式 3，SM2 位主要用于方式 2 和方式 3。接收状态，当串行口工作于方式 2 或 3，以及 SM2=1 时，只有当接收到第 9 位数据（RB8）为 1 时，才把接收到的前 8 位数据送入 SBUF，且置位 RI 发出中断申请，否则会将接受到的数据放弃。当 SM2=0 时，就不管第 9 位数据是 0 还是 1，都难得数据送入 SBUF，并发出中断申请。

工作于方式 0 时，SM2 必须为 0。

(3).REN：允许接收位。

REN 用于控制数据接收的允许和禁止，REN=1 时，允许接收，REN=0 时，禁止接收。

(4).TB8：发送接收数据位 8。

在方式 2 和方式 3 中，TB8 是要发送的——即第 9 位数据位。在多机通信中同样亦要传输这一位，并且它代表传输的地址还是数据，TB8=0 为数据，TB8=1 时为地址。

(5).RB8：接收数据位 8。

在方式 2 和方式 3 中，RB8 存放接收到的第 9 位数据，用以识别接收到的数据特征。

(6).TI：发送中断标志位。

可寻址标志位。方式 0 时，发送完第 8 位数据后，由硬件置位，其它方式下，在发送或停止位之前由硬件置位，因此，TI=1 表示帧发送结束，TI 可由软件清“0”。

(7).RI：接收中断标志位。

可寻址标志位。接收完第 8 位数据后，该位由硬件置位，在其他工作模式下，该位由硬件置位，RI=1 表示帧接收完成。

电源管理寄存器 PCON

PCON 主要是为 CHMOS 型单片机的电源控制而设置的专用寄存器，单元地址是 87H，其结构格式如下：

表 2 PCON 电源管理寄存器结构

PCON D7 D6 D5 D4 D3 D2 D1 D0

位符号 SMOD - - - GF1 GF0 PD IDL

在 CHMOS 型单片机中，除 SMOD 位外，其他位均为虚设的，SMOD 是串行口波特率倍增位，当 SMOD=1 时，串行口波特率加倍。系统复位默认为 SMOD=0。

中断允许寄存器 IE

中断允许寄存器在前一节中已阐述,这里重述一下对串行口有影响的位 ES。ES 为串行中断允许控制位,ES=1 允许串行中断,ES=0,禁止串行中断。

表 3 IE 中断允许控制寄存器结构

位符号 EA - - ES ET1 EX1 ET0 EX0

位地址 AFH AEH ADH ACH ABH AAH A9H A8H

MCS-51 的中断系统

对初学者来说,中断这个概念比较抽象,其实单片机的处理系统与人的思维有着许多异曲同工之妙,我们举个很贴切的比方,在日常生活和工作中有很多类似的情况。假如你正在上班,例如是编译资料,这时候电话铃响了,你在书本上做个记号(以记下你现在正编译到某某页),然后与对方通电话,而此时恰好有客人到访,你先停下通电话,与客人说几句话,叫客人稍候,然后回头继续通完电话,再与客人谈话。谈话完毕,送走客人,继续你的资料编译工作。

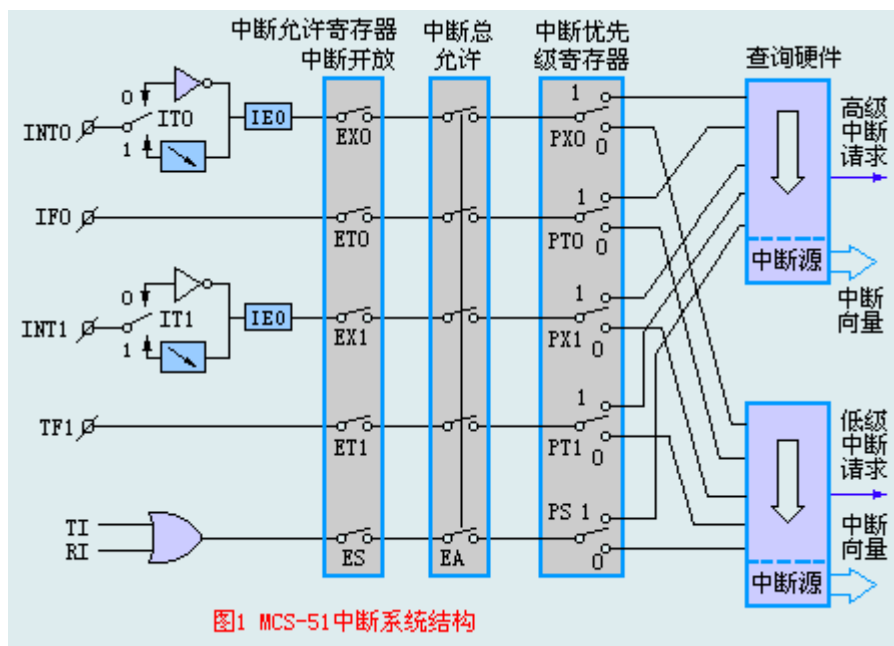
这就是日常生活和工作中的中断现象,类似的情况还有很多,从编译资料到接电话是第一次中断,通电话的过程中引有客人到访,这是第二次中断,即在中断的过程中又出现第二次中断,这就是我们

常说的中断嵌套。处理完第二个中断任务后,回头处理第一个中断,第一个中断完成后,再继续你原先的主要工作。

为什么会出现这样的中断呢?道理很简单,人非三头六臂,人只有一个脑袋,在一种特定的时间内,可能会面对着两、三甚至更多的任务。但一个人又不可能在同一时间去完成多样任务,因此你只能采分析任务的轻重缓急,采用中断的方法穿插去完成它们。那么这种情况对于单片机中的中央处理器也是如此,单片机中 CPU 只有一个,但在同一时间内可能会面临着处理很多任务的情况,如运行主程序、数据的输入和输出,定时/和计数时间已到要处理、可能还有一些外部的更重要的中断请求(如超温超压)要先处理。此时也得象人的思维一样停下某一样(或几样)工作先去完成一些紧急任务的中断方法。

这样的一样处理方法上升到计算机理论,就是一个资源面对多项任务的处理方式,由于资源有限,面对多项任务同时要处理时,就会出现资源竞争的现象。中断技术就是为了解决资源竞争的一个可行的方法,采用中断技术可使多项任务共享一个资源,所以有些文献也称中断技术是一种资源共享技术。

[1].MCS-51 的中断结构



计算机的中断系统能够加强 CPU 对多任务事件的处理能力。从而使它的应用范围进一步扩大。在 MCS-48 结构的基础上，MCS-51 在增强了 I/O 的种类、功能和数量的同时，也增强了中断能力。MCS-51 提供了 5 个中断源，两个中断优先级控制，可实现两个中断服务嵌套。当 CPU 支持中断屏蔽指令后，可将一部分或所有的中断关断，只有打开相应的中断控制位后，方可接收相应的中断请求。程序设置中断的允许或屏蔽，也可设置中断的优先级。

[2].中断处理流程

CPU 响应中断请求后，就立即转入执行中断服务程序。不同的中断源、不同的中断要求可能有不同的中断处理方法，但它们的处理流程一般都如下所述。

现场保护和现场恢复

中断是在执行其它任务的过程中转去执行临时的任务,为了在执行完中断服务程序后,回头执行原先的程序时,知道程序原来在何处打断的,各有关寄存器的内容如何,就必须在转入执行中断服务程序前,将这些内容和状态进行备份——即保护现场。就象文章开头举的例子,在看书时,电话铃响需传去接电话时,必须在书本上做个记号,以便在接完电话后回来看书时,知道从哪些内容继续往下看。计算机的中断处理方法也如此,中断开始前需将个有关寄存器的内容压入堆栈进行保存,以便在恢复原来程序时使用。

中断服务程序完成后,继续执行原先的程序,就需把保存的现场内容从堆栈中弹出,恢复寄存器 and 存储单元的原有内容,这就是现场恢复。

如果在执行中断服务时不是按上述方法进行现场保护和恢复现场,就会是程序运行紊乱,程序跑飞,自然使单片机不能正常工作。

中断打开和中断关闭

在中断处理进行过程中,可能又有新的中断请求到来,这里规定,现场保护和现场恢复的操作是不允许打扰的,否则保护和恢复的过程就可能使数据出错,为此在进行现场保护和现场恢复的过程中,必须关闭总中断,屏蔽其它所有的中断,待这个操作完成后再打开总中断,以便实现中断嵌套。

中断服务程序

既然有中断产生，就必然有其具体的需执行的任务，中断服务程序就是执行中断处理的具体内容，一般以子程序的形式出现，所有的中断都要转去执行中断服务程序，进行中断服务。

中断返回

执行完中断服务程序后，必然要返回，中断返回就是被程序运行从中断服务程序转回到原工作程序上来。在 MCS-51 单片机中，中断返回是通过一条专门的指令实现的，自然这条指令是中断服务程序的最后一条指令。

[3].MCS-51 的中断源

8051 有 5 个中断源，它们是两个外中断 INT0 (P3.2) 和 INT1 (P3.3)、两个片内定时/计数器溢出中断 TF0 和 TF1，一个是片内串行口中断 TI 或 RI，这几个中断源由 TCON 和 SCON 两个特殊功能寄存器进行控制。

在前一节，我们已对 TCON 的控制位进行了说明，现在继续对它的中断控制有关的位进行谈论。TCON 寄存器的结构如下：

表 1 TCON 寄存器结构

TCON D7 D6 D5 D4 D3 D2 D1 D0

TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0

位地址 8FH 8EH 8DH 8CH 8BH 8AH 89H 88H

- IE1:外部边沿触发中断 1 请求标志，其功能和操作类似于 TF0。
- IT1:外部中断 1 类型控制位，通过软件设置或清楚，用于控制外中断的触发信号类型。IT1=1，边沿触发。IT=0 是电平触发。
- IE0:外部边沿触发中断 0 请求标志，其功能和操作类似于 IE1。
- IT0:外部中断 0 类型控制位，通过软件设置或清楚，用于控制外中断的触发信号类型。其功能和操作类似于 IE1。

SCON 是串行口控制寄存器，字节地址为 98H，SCON 的低二位是串行口的发送和接收中断标志，其格式如下：

表 2 SCON 寄存器结构

SCON D7 D6 D5 D4 D3 D2 D1 D0

- - - - - TI RI

位地址 99H 98H

- TI:MCS-51 串行口的发送中断标志，在串行口以方式 0 发送时，每当发送完 8 位数据，由硬件置位。如果以方式 1、方式 2 或方式 3 发送时，在发送停止位的开始时 TI 被置 1，TI=1 表示串行发送器正向 CPU 发出中断请求，向串行口的数据缓冲器 SBUF 写入一

个数据后就立即启动发送器继续发送。但是 CPU 响应中断请求后，转向执行中断服务程序时，并不清零 TI，TI 必须由用户的中断服务程序清“0”，即中断服务程序必须有“CLR TI”或“ANL SCON, #0FDH”等指令来清零 TI。

· RI:串行口接收中断标志.若串行口接收器允许接收,并以方式 0 工作,每当接收到 8 位数据时,RI 被置 1,若以方式 1、2、3 方式工作,当接收到半个停止位时, TI 被置 1,当串行口一方式 2 或 3 方式工作,且当 SM2=1 时,仅当接收到第 9 位数据 RB8 为 1 后,同时还要在接收到半个停止位时, RI 被置 1。RI 为 1 表示串行口接收器正向 CPU 申请中断。同样 RI 标志栩栩如生由用户的软件清“0”。

[4].中断的控制

对于中断控制，在上一节中我们已经对 TCON 和 SCON 进行了分析，其实它们两个寄存器也是中断的控制寄存器，负责对中断的部分功能进行控制。我们这里谈论的是另外两个控制寄存器 IE 和 IP。

MCS-51 的对中断的开放和屏蔽是由中断允许寄存器 IE 控制来实现的，IE 的结构格式如下。

表 3 IE 寄存器结构

IE D7 D6 D5 D4 D3 D2 D1 D0

EA - - ES ET1 EX1 ET0 EX0

位地址 AFH ACH ABH AAH A9H A8H

下面我们对 IE 寄存器的各控制位进行介绍：

- EA:中断总控制位，EA=1，CPU 开放中断。EA=0，CPU 禁止所有中断。

- ES:串行口中断控制位，ES=1 允许串行口中断，ES=0，屏蔽串行口中断。

- ET1:定时/计数器 T1 中断控制位。ET1=1，允许 T1 中断，ET1=0，禁止 T1 中断。

- EX1:外中断 1 中断控制位，EX1=1，允许外中断 1 中断，EX1=0，禁止外中断 1 中断。

- ET0:定时/计数器 T0 中断控制位。ET1=1，允许 T0 中断，ET1=0，禁止 T0 中断。

- EX0:外中断 0 中断控制位，EX1=1，允许外中断 0 中断，EX1=0，禁止外中断 0 中断。

MCS-51 有两个中断优先级，即高优先级和低优先级，每个中断源都可设置为高或低中断优先级。如果有一低优先级的中断正在执行，那么高优先级的中断出现中断请求时，CPU 则会响应这个高

有限级的中断，也即高优先级的中断可以打断低优先级的中断。而若 CPU 正在处理一个高优先级的中断，此时，就算是有低优先级的中断发出中断请求，CPU 也不会理会这个中断，而是继续执行正在执行的中断服务程序，一直到程序结束，执行最后一条返回指令，返回主程序然后再执行一条指令后才会响应新的中断请求。

为了实现上述功能，MCS-51 的中断系统有两个不可寻址的优先级状态触发器，一个指出 CPU 是否在执行高优先级中断服务程序，另一个指出 CPU 是否正在执行低优先级的中断服务程序，这两个中断触发器的 1 状态分别屏蔽所有中断申请和同一级别的其他中断申请，此外，MCS-51 还有一个申请优先级寄存 IP，IP 的格式如下，字节地址是 B8H。

表 4 IP 中断控制寄存器结构

IP	D7	D6	D5	D4	D3	D2	D1	D0
- - -	PS	PT1	Px1	PT0	Px0			
位地址	BCH	BBH	BAH	B9H	B8H			

- PS:串行口中断口优先级控制位，PS=1，串行口中断声明为高优先级中断，PS=0，串行口定义为低优先级中断。
- PT1:定时器 1 优先级控制位。PT1=1，声明定时器 1 为高优先级中断，PT1=0 定义定时器 1 为低优先级中断。

· PX1:外中断 1 优先级控制位。PT1=1，声明外中断 1 为高优先级中断，PX1=0 定义外中断 1 为低优先级中断。

· PT0:定时器 0 优先级控制位。PT1=1，声明定时器 0 为高优先级中断，PT1=0 定义定时器 0 为低优先级中断。

· PX0:外中断 0 优先级控制位。PT1=1，声明外中断 0 为高优先级中断，PX1=0 定义外中断 0 为低优先级中断。

[5].中断的响应

MCS-51CPU 在每一个机器周期顺序检查每一个中断源，在机器周期的 S6 按优先级处理所有被激活的中断请求，此时，如果 CPU 没有正在处理更高或相同优先级的中断，或者现在的机器周期不是所执行指令的最后一个机器周期，或者 CPU 不是正在执行 RETI 指令或访问 IE 和 IP 的指令（因为按 MCS-51 中断系统的特性规定，在执行完这些指令之后，还要在继续执行一条指令，才会响应中断），CPU 在下一个机器周期响应激活了的最高级中断请求。

中断响应的主要内容就是由硬件自动生成一条长调用 LCALL addr16 指令，这里的 addr16 就是程序存储器中相应的中断区入口地址，这些中断源的服务程序入口地址如下：

表 5 个中断源的服务程序入口地址

中断源 入口地址

外中断 0 0003H

定时/计数器 0 000BH

外中断 1 0013H

定时/计数器 0 001BH

串行口中断 0023H

生成 LCALL 指令后,CPU 紧跟着便执行之.首先将 PC(程序计数器)的内容压入堆栈保护断点,然后把中断入口地址赋予 PC , CPU 便按新的 PC 地址 (即中断服务程序入口地址) 执行程序。

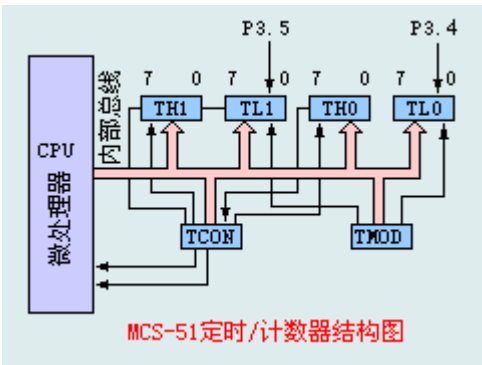
值得一提的是,各中断区只有 8 个单元,一般情况下 (除非中断程序非常简单) , 都不可能安装下一个完整的中断服务程序。因此,通常是在这些入口地址区放置一条无条件转移指令,使程序按转移的实际地址去执行真正的中断服务程序。

MCS-51 定时/计数器

MCS-51 的单片机内有两个 16 位可编程的定时/计数器,它们具有四种工作方式,其控制字和状态均在相应的特殊功能寄存器中,通过对控制寄存器的编程,就可方便地选择适当的工作方式。下面我们对它们的特性进行阐述。

定时/计数器的工作方式

MCS-51 单片机内部的定时/计数器的结构如图 1 所示 ,定时器 T0 特性功能寄存器 TL0 (低 8 位) 和 TH0 (高 8 位) 构成 , 定时器 T1 由特性功能寄存器 TL1 (低 8 位) 和 TH1 (高 8 位) 构成。特殊功能寄存器 TMOD 控制定时寄存器的工作方式 , TCON 则用于控制定时器 T0 和 T1 的启动和停止计数 , 同时管理定时器 T0 和 T1 的溢出标志等。程序开始时需对 TL0、TH0、TL1 和 TH1 进行初始化编程 , 以定义它们的工作方式和控制 T0 和 T1 的计数。



TMOD 和 TCON 这两个特殊功能寄存器的格式参见下表：

[1]. 定时/计数器的方式控制字 TMOD , 字节地址为 89H , 其格式如表 1：

表 1 TMOD 寄存器结构

D7 D6 D5 D4 D3 D2 D1 D0

GATA M1 M0 GATA M1 M0

←T1 方式字段→ ←T0 方式字段→

[2]. 定时器控制寄存器 TCON，字节地址为 88H，位地址为 88H—8FH，其格式如表 2：

表 2 TCON 结构

D7 D6 D5 D4 D3 D2 D1 D0

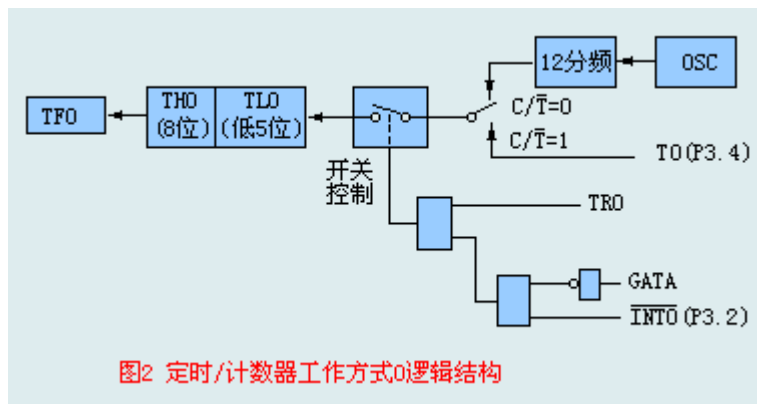
TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0

TMOD 和 TCON 各位的意义和用途我们将在下面的章节中予以介绍，需要注意的是，TCON 的 D0—D3 位与中断有关，我们会在中断的内容中加以说明，MCS-51 的定时/计数器共有四种工作方式，我们逐个进行讨论。

工作方式 0

定时/计数器 0 的工作方式 0 电路逻辑结构见图 2（定时/计数器 1 与其完全一致），工作方式 0 是 13 位计数结构的工作方式，其计数器由 TH 的全部 8 位和 TL 的低 5 位构成，TL 的高 3 位没有使用。当=0 时，多路开关接通振荡脉冲的 12 分频输出，13 位计数器以次进行计数。这就是定时工作方式。当=1 时，多路开关接通计数

引脚 (To)，外部计数脉冲由银南脚 To 输入。当计数脉冲发生负跳变时，计数器加 1，这就是我们常称的计数工作方式。



不管是哪种工作方式，当 TL 的低 5 位溢出时，都会向 TH 进位，而全部 13 位计数器溢出时，则会向计数器溢出标志位 TFO 进位。

我们讨论门控位 GATA 的功能，GATA 位的状态决定定时器运行控制取决于 TR0 的一个条件还是 TR0 和 INTO 引脚这两个条件。

当 GATA=1 时，由于 GATA 信号封锁了与门，使引脚 INTO 信号无效。而这时候如果 TR0=1，则接通模拟开关，使计数器进行加法计数，即定时/计数工作。而 TR0=0，则断开模拟开关，停止计数，定时/计数不能工作。

当 GATA=0 时，与门的输出端由 TR0 和 INTO 电平的状态确定，此时如果 TR0=1，INT0=1 与门输出为 1，允许定时/计数器计数，在这种情况下，运行控制由 TR0 和 INTO 两个条件共同控制，TR0 是确定定时/计数器的运行控制位，由软件置位或清“0”。

如上所述，TF0 是定时/计数器的溢出状态标志，溢出时由硬件置位，TF0 溢出中断被 CPU 响应时，转入中断时硬件清“0”，TF0 也可由程序查询和清“0”。

在工作方式下，计数器的计数值范围是：

1—8192 (213)

当为定时工作方式时，定时时间的计算公式为：

(213—计数初值) ×晶振周期×12

或 (213—计数初值) ×机器周期

其时间单位与晶振周期或机器周期相同。

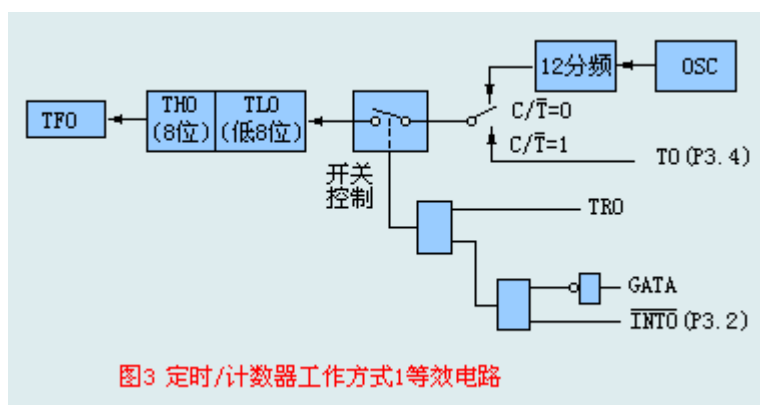
如果单片机的晶振选为 6.000MHz，则最小定时时间为：

$[213 - (213 - 1)] \times 1/6 \times 10^{-6} \times 12 = 2 \times 10^{-6}(\text{s}) = 2(\mu\text{s})$

$(213 - 0) \times 1/6 \times 10^{-6} \times 12 = 16384 \times 10^{-6}(\text{s}) = 16384(\mu\text{s})$ 。

工作方式 1

当 M1，M0=01 时，定时/计数器处于工作方式 1，此时，定时/计数器的等效电路如图 3 所示，仍以定时器 0 为例，定时器 1 与之完全相同。



可以看出，方式0和方式1的区别仅在于计数器的位数不同，方式0为13位，而方式1则为16位，由TH0作为高8位，TL0为低8位，有关控制状态字（GATA、TF0、TR0）和方式0相同。

在工作方式1下，计数器的计数值范围是：

1—65536 (216)

当为定时工作方式1时，定时时间的计算公式为：

$(216 - \text{计数初值}) \times \text{晶振周期} \times 12$

或 $(216 - \text{计数初值}) \times \text{机器周期}$

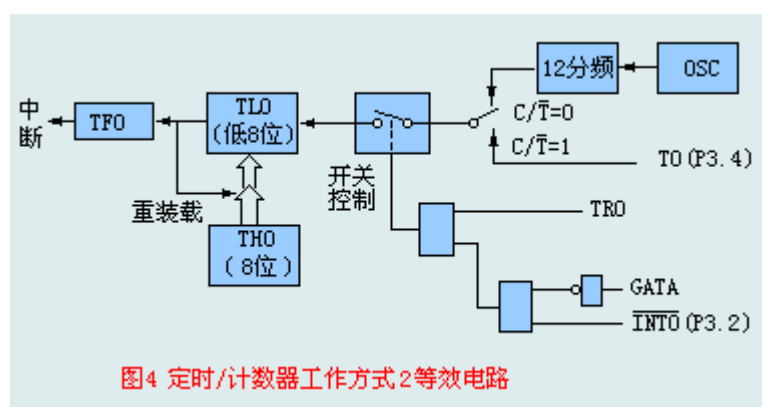
其时间单位与晶振周期或机器周期相同。

如果单片机的晶振选为6.000MHz，则最小定时时间为：

$[213 - (216 - 1)] \times 1/6 \times 10^{-6} \times 12 = 2 \times 10^{-6} (\text{s}) = 2 (\mu\text{s})$

$$(2^{16}-0) \times 1/6 \times 10^{-6} \times 12 = 131072 \times 10^{-6} (\text{s}) = 131072 (\mu\text{s}).$$

当 M1M0=10 时,定时/计数器处于工作方式 2.此时定时器的等效电阻如图 4 所示.我们还是以定时/计数器 0 为例,定时/计数器 1 与之完全一致。

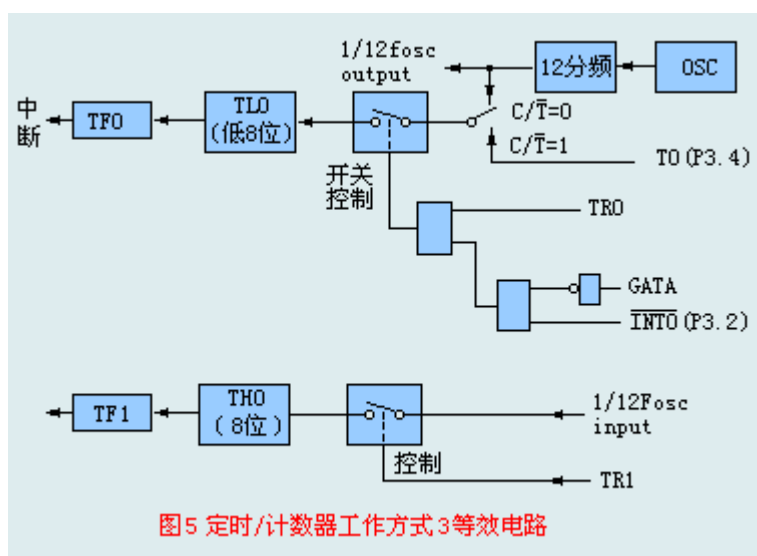


工作方式 0 和工作方式 1 的最大特点就是计数溢出后，计数器为全 0，因而循环定时或循环计数应用时就存在反复设置初值的问题，这给程序设计带来许多不便，同时也会影响计时精度，工作方式 2 就针对这个问题而设置，它具有自动重装载功能，即自动加载计数初值，所以也有的文献称之为自动重加载工作方式。在这种工作方式中，16 位计数器分为两部分，即以 TLO 为计数器，以 TH0 作为预置寄存器，初始化时把计数初值分别加载至 TLO 和 TH0 中，当计数溢出时，不再象方式 0 和方式 1 那样需要“人工干预”，由软件重新赋值，而是由预置寄存器 TH 以硬件方法自动给计数器 TLO 重新加载。

程序初始化时，给 TL0 和 TH0 同时赋以初值，当 TL0 计数溢出时，置位 TF0 的同时把预置寄存器 TH0 中的初值加载给 TL0，TL0 重新计数。如此反复，这样省去了程序不断需给计数器赋值的麻烦，而且计数准确度也提高了。但这种方式也有其不利的一面，就是这样一来的计数结构只有 8 位，计数值有限，最大只能到 255。所以这种工作方式很适合于那些重复计数的应用场合。例如我们可以通过这样的计数方式产生中断，从而产生一个固定频率的脉冲。也可以当作串行数据通信的波特率发送器使用。

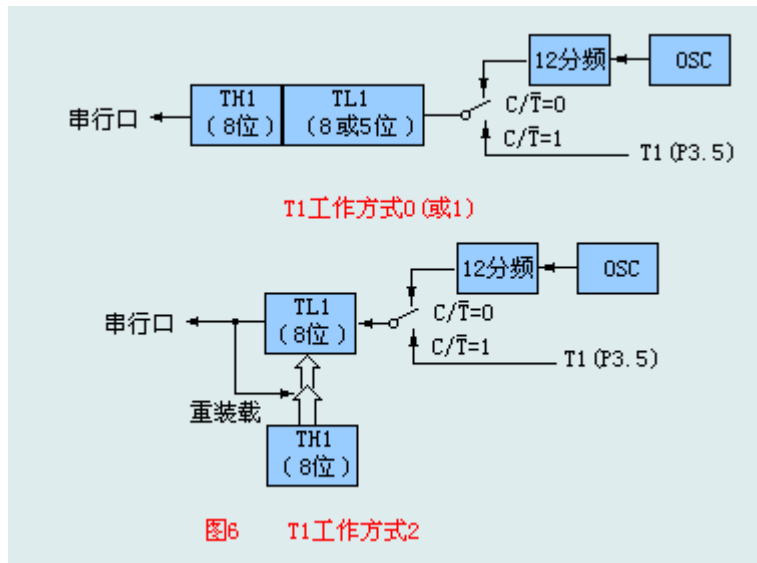
工作方式 3

当 M1M0=11 时，定时/计数器处于工作方式 3，此时，定时/计数器的等效电路如图 3 所示，仍以定时器 0 为例，值得注意的是，在工作方式 3 模式下，定时/计数器 1 的工作方式与之不同，下面我们分别讨论。



在工作方式 3 模式下，定时/计数器 0 被拆成两个独立的 8 位计数器 TL0 和 TH0。其中 TL0 既可以作计数器使用，也可以作为定时器使用，定时/计数器 0 的各控制位和引脚信号全归它使用。其功能和操作与方式 0 或方式 1 完全相同。TH0 就没有那么多“资源”可利用了，只能作为简单的定时器使用，而且由于定时/计数器 0 的控制位已被 TL0 占用，因此只能借用定时/计数器 1 的控制位 TR1 和 TF1，也就是以计数溢出去置位 TF1，TR1 则负责控制 TH0 定时的启动和停止。等效电路参见图 6。由于 TL0 既能作定时器也能作计数器使用，而 TH0 只能作定时器使用而不能作计数器使用，因此在方式 3 模式下，定时/计数器 0 可以构成二个定时器或者一个定时器和一个计数器。

如果定时/计数器 0 工作于工作方式 3，那么定时/计数器 1 的工作方式就不可避免受到一定的限制，因为自己的一些控制位已被定时/计数器借用，只能工作在方式 0、方式 1 或方式 2 下，等效电路参见图 6



在这种情况下，定时/计数器 1 通常作为串行口的波特率发生器使用，以确定串行通信的速率，因为已没有 TF1 被定时/计数器 0 借用了，只能把计数溢出直接送给串行口。当作波特率发生器使用时，只需设置好工作方式，即可自动运行。如要停止它的工作，需送入一个把它设置为方式 3 的方式控制字即可，这是因为定时/计数器本身就不能工作在方式 3，如硬把它设置为方式 3，自然会停止工作。