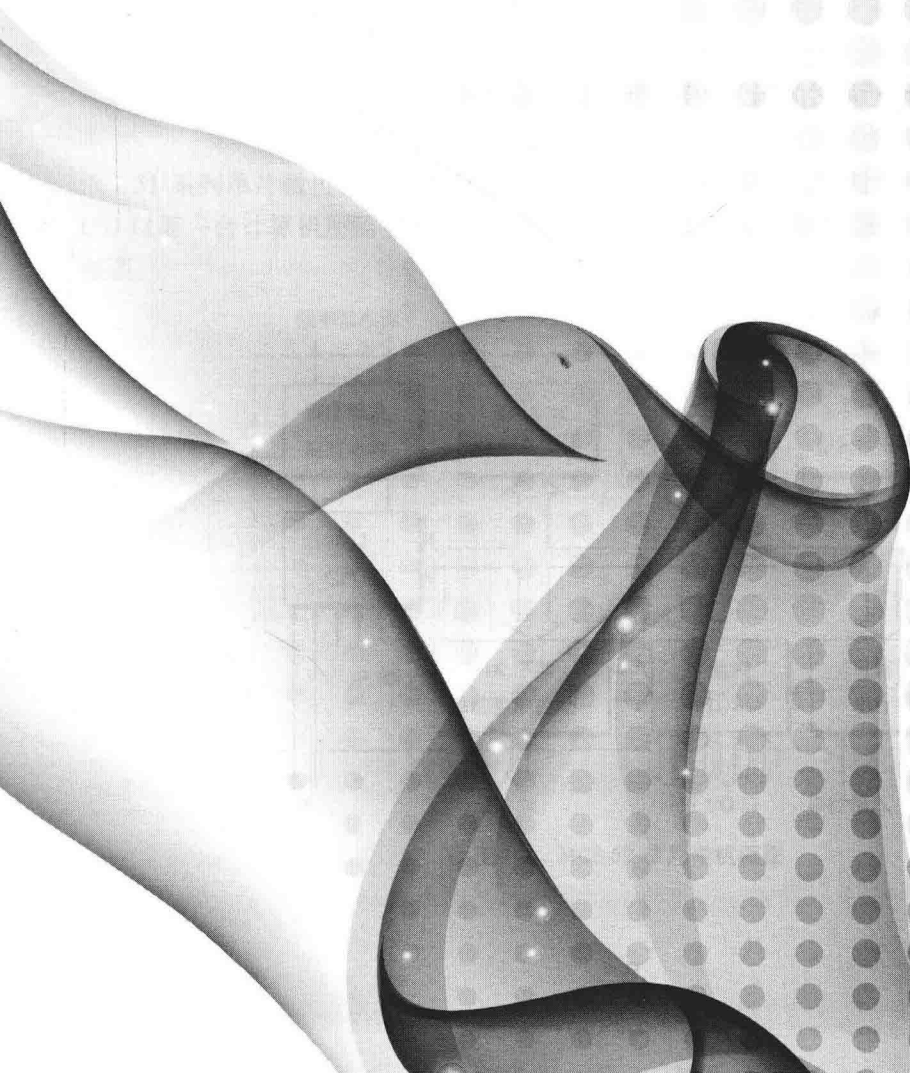


实验篇



第1章 51 系列单片微型计算机简介

单片微型计算机(简称单片机)技术在现代社会的各个领域正越来越广泛地得到应用。目前 51 系列单片机仍然被广泛应用。这是因为 Intel 公司实施了对 51 系列单片微机的技术开放政策,与多家半导体公司签订了技术协议,允许这些公司在 51 内核的基础上开发与之兼容的新型产品。这一策略使 51 系列兼容单片微机的产品种类和数量得到了迅速的发展。众多半导体厂商在 51 系列单片微机的基础上,结合了最新的技术成果,推出了各具特色的 51 兼容单片机。这给 51 单片机这一早期开发的产品赋予了新的生命力,并形成了众星捧月、不断更新、长久不衰的发展格局,在 8 位单片机的发展中成为一道独特的风景线。51 系列以及由其派生出的 ATME189CX 系列、PHILIPS 80C51 系列、NXP 80C51 系列、C8051F 系列、华邦系列、Dallas 系列、STC 系列等以其优越的性能、成熟的技术及高可靠性和高性能价格比,迅速占领了工业测控和智能仪器仪表应用的主要市场,成为国内单片机应用领域的主流。

1.1 51 系列单片微型计算机的硬件组成

51 系列单片机在一块芯片中集成了 CPU、ROM、RAM、定时器/计数器和多种功能的 I/O 口等一台计算机所需要的基本功能部件。如图 1-1 所示为 MCS-52 单片微机的基本结构框图。

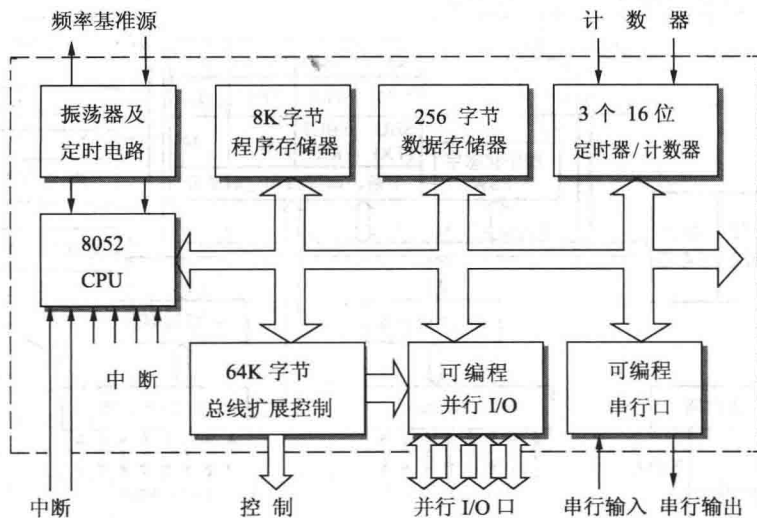


图 1-1 MCS-52 单片微机框图

MCS-52 单片微机内部主要包含了下列几个部件：

- 一个 8 位 CPU.
- 一个片内振荡器及定时电路.
- 8K 字节程序存储器.
- 256 字节数据存储器.
- 3 个 16 位定时器/计数器.
- 一个可编程全双工串行口.
- 4 个 8 位可编程并行 I/O 端口.
- 64K 字节外部数据存储器器和 64K 字节程序存储器扩展控制电路.
- 6 个中断源.
- 两个优先级嵌套中断结构.

以上各部分通过总线相连接.

51 系列单片微机的处理器及内部结构如图 1-2 所示.

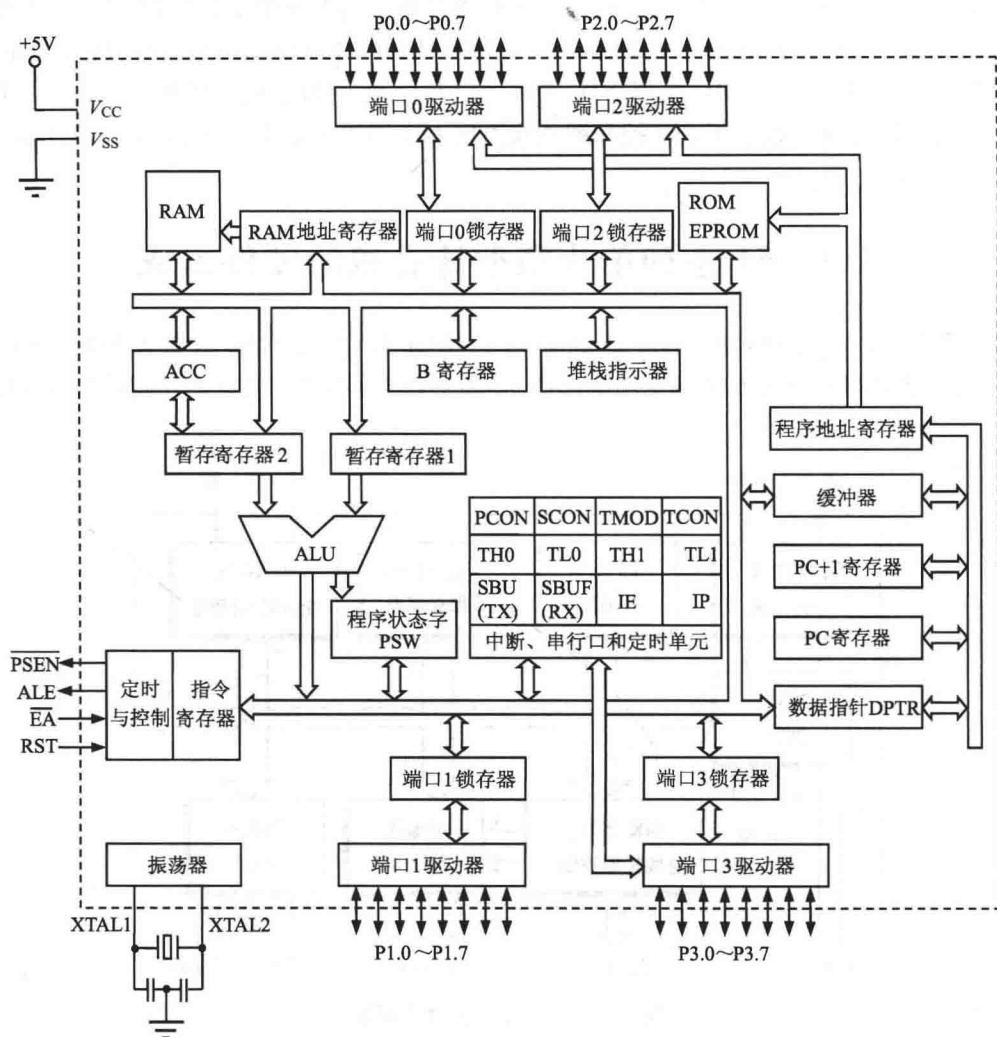


图 1-2 51 单片微机处理器及内部结构框图

和一般微处理器相比,它除了增加了接口部分外,基本结构是相似的,只是有的部件名称不同.如程序状态字 PSW(Program Status Word)就相当于一般微处理器中的标志寄存器 FR(Flag Register).但也有明显不同的地方,如数据指针 DPTR(Data Pointer)是专门为指示存储器地址而设置的寄存器. CPU 是单片微机的核心部件,它是由运算器、控制器和若干个专用寄存器等部件组成的.

1.2 51 系列单片微型计算机的引脚功能

51 系列单片微机中各种型号芯片的引脚是相互兼容的,40 引脚双列直插封装引脚排列如图 1-3(a)所示,图(b)为功能框图.

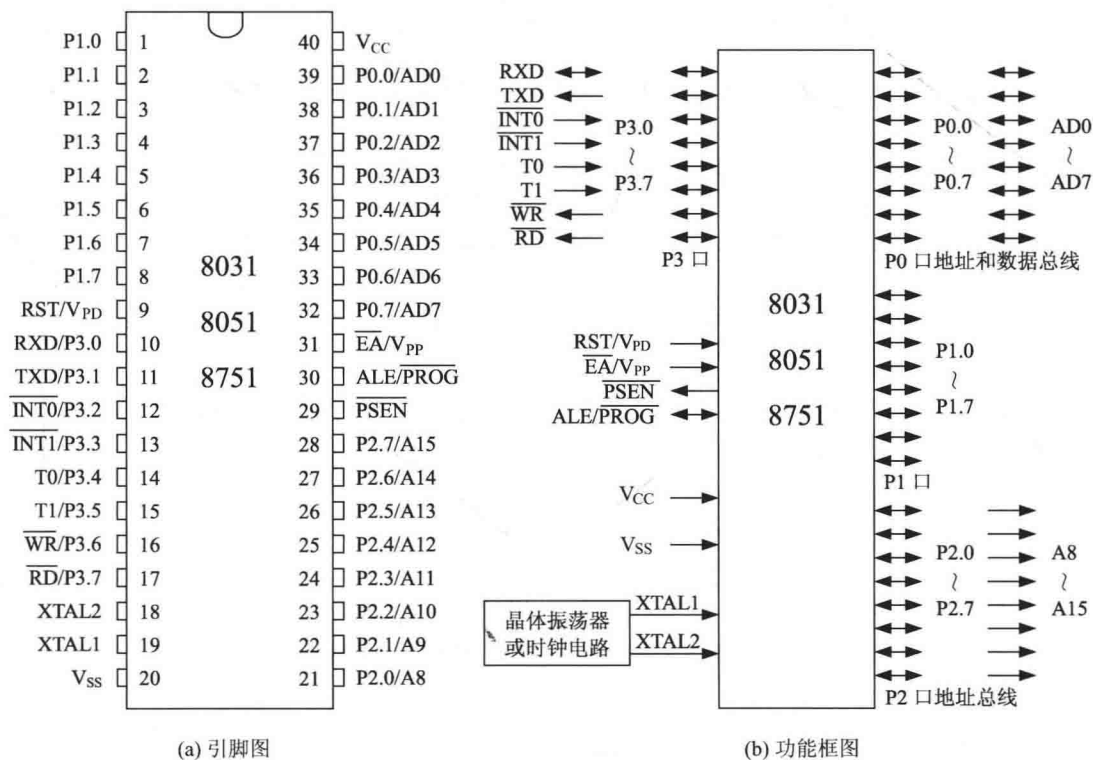


图 1-3 MCS-51 单片微机引脚图

一、主电源引脚 V_{CC} 和 V_{SS}

- V_{CC} (40): 正常操作时接 +5V 电源.
- V_{SS} (20): 接地.

二、外接晶体引脚 XTAL1 和 XTAL2

- XTAL1 (19): 接外部晶体和微调电容的一个引脚.
- XTAL2 (18): 接外部晶体和微调电容的另一个引脚.

三、控制或其他电源复用引脚RST/V_{PD}、ALE/PROG、PSEN和EA/V_{PP}

- RST/V_{PD}(9): 当振荡器工作时,在此引脚上出现两个机器周期的高电平将使单片微机复位.
- ALE/PROG(30): 当访问外部存储器时,CPU 输出地址锁存允许 ALE(Address Latch Enable)信号,用于锁存低 8 位地址信息. 对于片内有程序存储器的单片微机,在编程期间,此引脚用于输入编程脉冲信号(PROG).
- PSEN(29): 该端输出外部程序存储器读选通信号.
- EA/V_{PP}(31): 访问外部程序存储器控制端. 当EA端保持高电平时,单片微机复位后,访问内部程序存储器,当EA保持低电平时,则只访问外部程序存储器,而不管内部是否有程序存储器.

四、输入/输出引脚P0.0 ~ P0.7、P1.0 ~ P1.7、P2.0 ~ P2.7、P3.0 ~ P3.7

- P0.0 ~ P0.7(39 ~32): P0 是一个 8 位漏极开路型双向 I/O 口. 在访问外部存储器时可作为地址(低 8 位)/数据分时复用总线使用. 作为地址/数据分时复用总线时,在访问存储器期间能激活内部的上拉电阻,此时 P0 成为一个双向口.
- P1.0 ~ P1.7(1 ~8): P1 是一个内部带上拉电阻的 8 位准双向 I/O 口.
- P2.0 ~ P2.7(28 ~21): P2 是一个内部带上拉电阻的 8 位准双向 I/O 口. 在访问外部存储器时,它送出高 8 位地址.
- P3.0 ~ P3.7(10 ~17): P3 是一个内部带上拉电阻的 8 位准双向 I/O 口,P3 每个引脚分别具有第二功能,如表 1-1 所示.

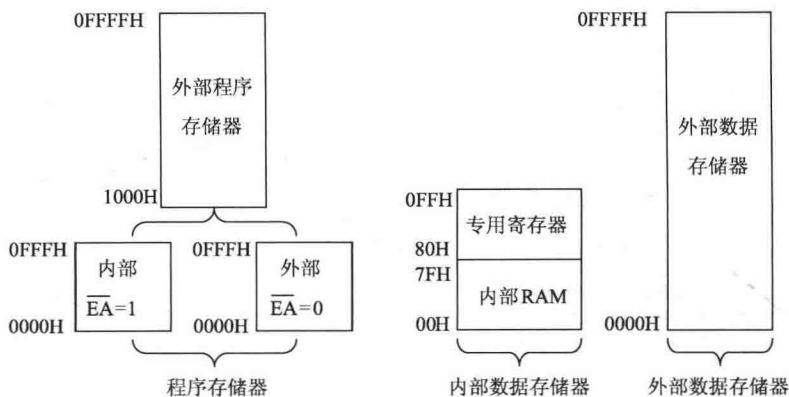
表 1-1 P3 各口线的第二功能

口 线	第 二 功 能
P3.0	RXD(串行口输入)
P3.1	TXD(串行口输出)
P3.2	$\overline{\text{INT0}}$ (外部中断 0 外部输入)
P3.3	$\overline{\text{INT1}}$ (外部中断 1 外部输入)
P3.4	T0(定时器/计数器 0 外部输入)
P3.5	T1(定时器/计数器 1 外部输入)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写选通)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读选通)

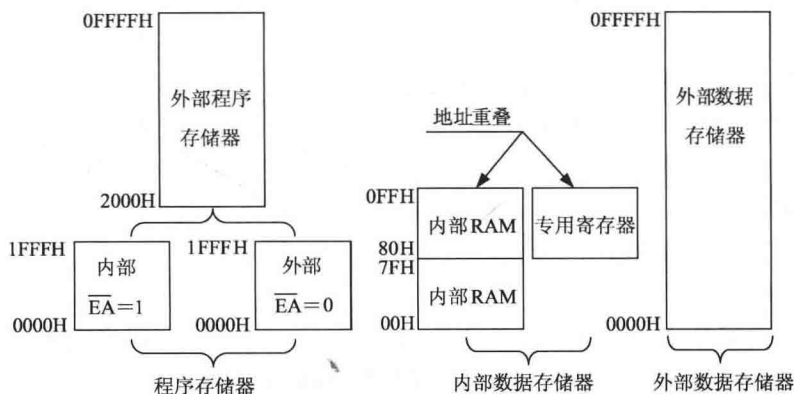
1.3 51 系列单片微型计算机的存储器与寄存器结构

51 系列单片微机的存储器结构和配置与常见的微型计算机的配置方式不同,它把程序存储器和数据存储器分开,即程序存储器和数据存储器独立编址,各有自己的寻址系统. 从

物理地址空间分析,有4个存储器空间:片内程序存储器、片外程序存储器、片内数据存储器 and 片外数据存储器.从逻辑地址空间分析,有3个存储器空间:片内外统一的64KB 程序存储器地址空间、256B(对51子系列)或384B(对52子系列)的内部数据存储器地址空间(其中有128B的专用寄存器地址空间)及64KB 的外部数据存储器地址空间.51系列存储器的配置如图1-4所示.



(a) 51子系列



(b) 52子系列

图1-4 MCS-51 单片微机存储器的配置

一、程序存储器

51系列中,64KB 程序存储器的地址空间是统一的.对于有内部程序存储器的单片机,应把 $\overline{EA}$ 引脚接高电平,使程序从内部程序存储器开始执行,当PC 值超过内部程序存储器的容量时,会自动转向外部程序存储器地址空间.无内部程序存储器的芯片, $\overline{EA}$ 引脚应始终接低电平,迫使系统复位后从外部程序存储器中取指令.

64KB 程序存储器中有7个单元具有特殊功能.

对于0000H 单元,MCS-51 系统复位后程序计数器PC 的内容为0000H,故系统是从0000H 单元开始取指,执行程序.

除了0000H 单元外,0003H、000BH、0013H、001BH、0023H 和002BH 这6个单元分别对



应于 6 种中断服务子程序的入口地址,如表 1-2 所示. 通常在这些入口地址处都安放一条绝对跳转指令,而真正的中断服务子程序则是从转移地址开始安放的,这 6 个单元又称为中断矢量单元地址,因此 0003H ~ 002BH 单元应被保留专用于中断服务处理.

表 1-2 各中断源所对应的中断入口地址

中 断 源	入口地址
外部中断 0	0003H
定时器/计数器 0 溢出中断	000BH
外部中断 1	0013H
定时器/计数器 1 溢出中断	001BH
串行口	0023H
* 定时器/计数器 2 溢出或 T2EX(P1.1)端负跳变时	002BH

注: 标有“*”的为 52 子系列所特有.

二、内部数据存储器

51 系列的数据存储器无论从物理上还是逻辑上都分为两个地址空间,一个是内部数据存储器;另一个为外部数据存储器.

内部数据存储器在物理上又可以分为 3 个不同的块: 00H ~ 07FH(0 ~ 127) 单元组成的低 128 字节的 RAM 块, 80H ~ 0FFH(128 ~ 255) 单元组成的高 128 字节的 RAM 块(仅为 52 子系列所有)、80H ~ 0FFH(128 ~ 255) 单元组成的高 128 字节的专用寄存器块(SFR). 应注意的是,高 128 字节的 SFR 块中仅有 26 个字节是有定义的,若访问的是这一块中没有定义的单元,则将得到一个随机数.

1. 内部 RAM.

MCS-51 的内部 RAM 结构如图 1-5 所示,其中 00H ~ 1FH(0 ~ 31) 单元共 32 个字节是 4 个通用工作寄存器区,每个区含有 8 个工作寄存器,标识符为 R0 ~ R7.

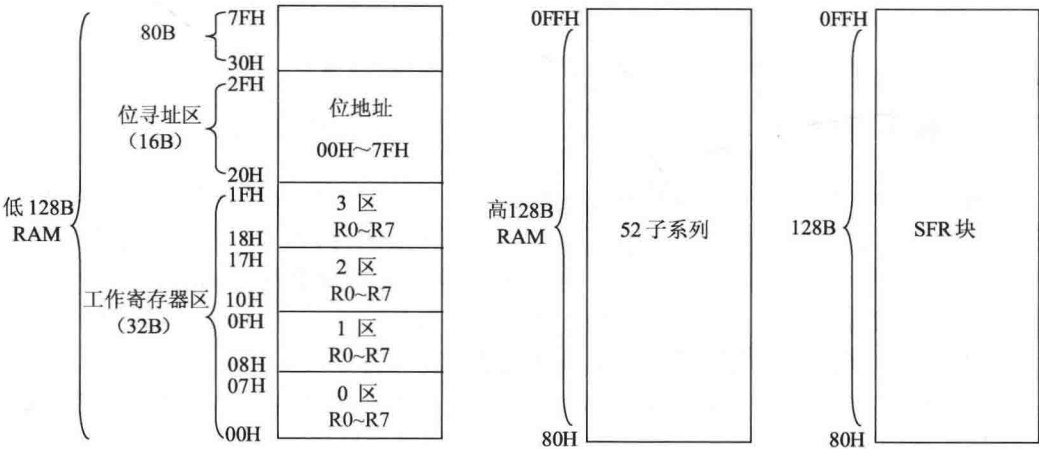


图 1-5 MCS-51 内部 RAM 结构



内部 RAM 的 20H~2FH 为位寻址区,位地址如表 1-3 所示.

表 1-3 内部 RAM 位寻址区位地址

位 地 址								字节地址
D7	D6	D5	D4	D3	D2	D1	D0	
7F	7E	7D	7C	7B	7A	79	78	2FH
77	76	75	74	73	72	71	70	2EH
6F	6E	6D	6C	6B	6A	69	68	2DH
67	66	65	64	63	62	61	60	2CH
5F	5E	5D	5C	5B	5A	59	58	2BH
57	56	55	54	53	52	51	50	2AH
4F	4E	4D	4C	4B	4A	49	48	29H
47	46	45	44	43	42	41	40	28H
3F	3E	3D	3C	3B	3A	39	38	27H
37	36	35	34	33	32	31	30	26H
2F	2E	2D	2C	2B	2A	29	28	25H
27	26	25	24	23	22	21	20	24H
1F	1E	1D	1C	1B	1A	19	18	23H
17	16	15	14	13	12	11	10	22H
0F	0E	0D	0C	0B	0A	09	08	21H
07	06	05	04	03	02	01	00	20H

2. 专用寄存器.

MCS-51 内部锁存器、定时器/计数器、串行口、数据缓冲器及各种控制寄存器和状态寄存器都是以专用寄存器的形式出现的,它们分散地分布在高 128 字节的内部数据存储器(80H~0FFH)内.表 1-4 列出了这些专用寄存器的助记标识符、名称及地址,表 1-5 介绍了专用寄存器的详细地址.

表 1-4 专用寄存器(除 PC 外)

标 识 符	名 称	地 址
* ACC	累加器	0E0H
* B	B 寄存器	0F0H
* PSW	程序状态字	0D0H
SP	堆栈指针	81H
DPTR	堆栈指针(包括 DPH 和 DPL)	83H 和 82H
* P0	口 0	80H

续表

标识符	名称	地址
* P1	口 1	90H
* P2	口 2	0A0H
* P3	口 3	0B0H
* IP	中断优先级控制	0B8H
* IE	允许中断控制	0A8H
TMOD	定时器/计数器方式控制	89H
* TCON	控制寄存器	88H
+ * T2CON	定时器/计数器 2 控制	0C8H
TH0	定时器/计数器 0(高位字节)	8CH
TL0	定时器/计数器 0(低位字节)	8AH
TH1	定时器/计数器 1(高位字节)	8DH
TL1	定时器/计数器 1(低位字节)	8BH
+ TH2	定时器/计数器 2(高位字节)	0CDH
+ TL2	定时器/计数器 2(低位字节)	0CCH
+ RLDH	定时器/计数器 2 自动再装载(高位字节)	0CBH
+ RLDL	定时器/计数器 2 自动再装载(低位字节)	0CAH
* SCON	串行控制	98H
SBUF	串行数据缓冲器	99H
PCON	电源控制	87H

注：带“*”号寄存器可按字节和按位寻址,带“+”号的寄存器是与定时器/计数器 2 有关的寄存器，仅在 52 子系列中存在。

表 1-5 专用寄存器地址表

位 地 址								字节地址	标识符
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80	P0
87	86	85	84	83	82	81	80		
—								81	SP
								82	DPL
								83	DPH
								87	PCON
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88	TCON
8F	8E	8D	8C	8B	8A	89	88		



续表

位 地 址								字节地址	标 识 符
—								89	TMOD
								8A	TL0
								8B	TL1
								8C	TH0
								8D	TH1
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90	P1
97	96	95	94	93	92	91	90		
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98	SCON
9F	9E	9D	9C	9B	9A	99	98		
—								99	SBUF
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0	P2
A7	A6	A5	A4	A3	A2	A1	A0		
EA	—	ET2	ES	ET1	EX1	ET0	EX0	A8	IE
AF	—	AD	AC	AB	AA	A9	A8		
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0	P3
B7	B6	B5	B4	B3	B2	B1	B0		
—	—	PT2	PS	PT1	PX1	PT0	PX0	B8	IP
—	—	BD	BC	BB	BA	B9	B8		
CY	AC	F0	RS1	RS0	OV		P	D0	PSW
D7	D6	D5	D4	D3	D2	D1	D0		
ACC·7	ACC·6	ACC·5	ACC·4	ACC·3	ACC·2	ACC·1	ACC·0	E0	ACC
E7	E6	E5	E4	E3	E2	E1	E0		
—	—	—	—	—	—	—	—	F0	B
F7	F6	F5	F4	F3	F2	F1	F0		

下面将介绍部分专用寄存器。

(1) 程序计数器 PC。

程序计数器 PC 用于安放下一条将要执行的指令的地址(程序存储器地址),它是一个 16 位专用寄存器,可以满足程序存储器 64KB 字节的寻址要求。PC 在物理上是独立的,它不属于内部数据存储器 SFR 块。

(2) 累加器 ACC。

累加器是一个最常用的专用寄存器,大部分单操作数指令的操作数取自累加器。

(3) B 寄存器。

在乘除指令中,用到了 B 寄存器。

(4) 程序状态字 PSW.

程序状态字是一个 8 位寄存器,即

CY	AC	FO	RS1	RS0	OV	—	P
----	----	----	-----	-----	----	---	---

,它反映了程序状态信息,其各位的功能说明如下.

- CY(PSW.7): 进位标志.
- AC(PSW.6): 辅助进位标志.
- FO(PSW.5): 用户标志.
- RS1、RS0(PSW.4、PSW.3): 寄存器区选择控制位,可以用软件来置位或清除以确定工作寄存器区.

RS1、RS0 与寄存器区的对应关系如下:

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

- OV(PSW.2): 溢出标志.
- PSW.1: 保留位,未用.
- P(PSW.0): 奇偶标志,每个指令周期都由硬件置位或清除,以表示累加器 A 中为 1 的位数的奇偶.若为 1 的位数为奇数,则 P=1,否则 P=0.

(5) 堆栈指针 SP.

堆栈指针 SP 是一个 8 位专用寄存器,它指示出堆栈顶部在内部 RAM 中的位置.系统复位后,SP 初始化为 07H,使得系统的堆栈从 08H 单元开始,因为 08H 是工作寄存器 1 区中的 R0,所以一般应使用指令对其重新定义.

(6) 串行数据缓冲器 SBUF.

串行数据缓冲器 SBUF 用于存放待发送或已接收的数据,它实际上由两个独立的寄存器组成,一个是发送缓冲器,一个是接收缓冲器.当把发送的数据送 SBUF 时,进的是发送缓冲器,当要从 SBUF 取数据时,则取自接收缓冲器.

❁ 三、外部数据存储器

MCS-51 的外部数据存储器寻址空间为 64KB,这对很多应用场合已足够使用.对外部数据存储器均采用间接寻址方式,运用 MOVX 指令. R0、R1 和 DPTR 都可以作为间址寄存器用.

1.4 51 系列单片微型计算机的指令系统

51 系列的指令系统是一个具有 255 种操作代码的集合.42 种指令功能助记符、7 种寻址方式,一共构造出 111 种指令.111 种指令中,单字节指令 49 种,双字节指令 46 种,三字节指令仅 16 种.

按指令的功能,可以把 111 种指令分成下面 5 类:



- 数据传送类(28 条).
- 算术运算类(24 条).
- 逻辑运算类(25 条).
- 控制转移类(17 条).
- 位操作类(17 条).

一、数据传送类指令

数据的传送是一种最基本、最主要的操作. 所谓“传送”, 是把源地址单元的内容传送到目的地址单元中去, 而源地址单元的内容不变; 或源、目的地址单元的内容互换.

数据传送类指令用到的助记符有 MOV、MOVX、MOVC、XCH、XCHD、PUSH、POP 7 种. 此外, MCS-51 单片机的指令系统还有一条 16 位的传送指令, 专用于设定数据指针 DPTR.

1. 内部存储器间的传送指令.

(1) 以累加器为目的的操作数的指令.

指令格式及其功能如下:

```
MOV  A, Rn           ; (A) ← (Rn)
MOV  A, direct       ; (A) ← (direct)
MOV  A, @ Ri         ; (A) ← ((Ri))
MOV  A, #data        ; (A) ← data
```

(2) 以寄存器 Rn 为目的的操作数的指令.

指令格式及其功能如下:

```
MOV  Rn, A           ; (Rn) ← (A)
MOV  Rn, direct      ; (Rn) ← (direct)
MOV  Rn, #data       ; (Rn) ← data
```

(3) 以直接地址 direct 为目的的操作数的指令.

指令及其格式功能如下:

```
MOV  direct, A       ; (direct) ← (A)
MOV  direct, Rn      ; (direct) ← (Rn)
MOV  direct1, direct2 ; (direct1) ← (direct2)
MOV  direct, @ Ri    ; (direct) ← ((Ri))
MOV  direct, #data   ; (direct) ← data
```

(4) 以间接地址为目的的操作数的指令.

指令及其格式功能如下:

```
MOV  @ Ri, A         ; ((Ri)) ← (A)
MOV  @ Ri, direct    ; ((Ri)) ← (direct)
MOV  @ Ri, #data     ; ((Ri)) ← data
```

2. 16 位数据传送指令.

指令格式及其功能如下:

```
MOV  DPTR, #data16   ; (DPH) ← dataH, (DPL) ← dataL
```



3. 堆栈操作指令.

(1) 入栈指令.

指令及其格式功能如下:

PUSH direct ; $(SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (direct)$

(2) 出栈指令.

指令及其格式功能如下:

POP direct ; $(direct) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$

4. 交换指令.

(1) 字节交换指令.

指令格式及其功能如下:

XCH A, Rn ; $(A) \leftrightarrow (Rn)$

XCH A, direct ; $(A) \leftrightarrow (direct)$

XCH A, @ Ri ; $(A) \leftrightarrow ((Ri))$

(2) 半字节交换指令.

指令格式及其功能如下:

XCHD A, @ Ri ; $(A)_{0-3} \leftrightarrow ((Ri))_{0-3}$

5. 程序存储器读数据指令.

(1) 远程查表指令.

指令格式及其功能如下:

MOVC A, @ A + DPTR ; $(A) \leftarrow ((A) + (DPTR))$

(2) 近程查表指令.

指令格式及其功能如下:

MOVC A, @ A + PC ; $(PC) \leftarrow (PC) + 1, (A) \leftarrow ((A) + (PC))$

6. 累加器 A 与外部数据存储器传送数据指令.

(1) 外部数据存储器内容送累加器(即读外部数据存储器).

指令格式及其功能如下:

MOVX A, @ Ri ; $(A) \leftarrow ((Ri))$, 指令执行时 P2 同时送出高 8 位地址

MOVX A, @ DPTR ; $(A) \leftarrow ((DPTR))$

在执行这两条指令时, P3.7 引脚上输出有效的 $\overline{RD}$ 信号, 用作外部数据存储器的读选通信号.

(2) 累加器内容送外部数据存储器(即写外部数据存储器).

指令格式及其功能如下:

MOVX @ Ri, A ; $((Ri)) \leftarrow (A)$, 指令执行时 P2 同时送出高 8 位地址

MOVX @ DPTR, A ; $((DPTR)) \leftarrow (A)$

在执行这两条指令时, P3.6 引脚上输出有效的 $\overline{WR}$ 信号, 用作外部数据存储器的写选通信号.

* 二、算术运算类指令

1. 加法指令(ADD、ADDC、DA).



(1) 不带进位加法指令.

指令格式及其功能如下:

ADD A, Rn	; $(A) \leftarrow (A) + (Rn)$
ADD A, direct	; $(A) \leftarrow (A) + (\text{direct})$
ADD A, @ Ri	; $(A) \leftarrow (A) + ((Ri))$
ADD A, #data	; $(A) \leftarrow (A) + \text{data}$

(2) 带进位加法指令.

指令格式及其功能如下:

ADDC A, Rn	; $(A) \leftarrow (A) + (Rn) + CY$
ADDC A, direct	; $(A) \leftarrow (A) + (\text{direct}) + CY$
ADDC A, @ Ri	; $(A) \leftarrow (A) + ((Ri)) + CY$
ADDC A, #data	; $(A) \leftarrow (A) + \text{data} + CY$

(3) 进制调整指令.

指令格式及其功能如下:

DA A	; 调整累加器 A 的内容为 BCD 码
------	----------------------

2. 减法指令 (SUBB).

指令格式及其功能如下:

SUBB A, Rn	; $(A) \leftarrow (A) - (Rn) - CY$
SUBB A, direct	; $(A) \leftarrow (A) - (\text{direct}) - CY$
SUBB A, @ Ri	; $(A) \leftarrow (A) - ((Ri)) - CY$
SUBB A, #data	; $(A) \leftarrow (A) - \text{data} - CY$

3. 递增/减指令 (INC、DEC).

(1) 增 1 指令.

指令格式及其功能如下:

INC A	; $(A) \leftarrow (A) + 1$
INC Rn	; $(Rn) \leftarrow (Rn) + 1$
INC direct	; $(\text{direct}) \leftarrow (\text{direct}) + 1$
INC @ Ri	; $((Ri)) \leftarrow ((Ri)) + 1$
INC DPTR	; $(DPTR) \leftarrow (DPTR) + 1$

(2) 减 1 指令.

指令格式及其功能如下:

DEC A	; $(A) \leftarrow (A) - 1$
DEC Rn	; $(Rn) \leftarrow (Rn) - 1$
DEC direct	; $(\text{direct}) \leftarrow (\text{direct}) - 1$
DEC @ Ri	; $((Ri)) \leftarrow ((Ri)) - 1$

4. 乘法指令 (MUL).

指令格式及其功能如下:

MUL AB	; $(B)(A) \leftarrow (A) \times (B)$
--------	--------------------------------------

5. 除法指令(DIV).

指令格式及其功能如下:

DIV AB ; (A) $\leftarrow$ (A)/(B)的商,(B) $\leftarrow$ (A)/(B)的余数

三、逻辑运算类指令

1. 累加器专用指令(CLR、CPL、RL、RR、RLC、RRC、SWAP).

(1) 累加器清零.

指令格式及其功能如下:

CLR A ; (A) $\leftarrow$ 0

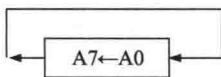
(2) 累加器内容按位取反.

指令格式及其功能如下:

CPL A ; (A) $\leftarrow$ $\overline{(A)}$

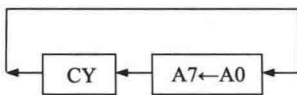
(3) 累加器内容循环左移.

指令格式及其功能如下:

RL A ; 

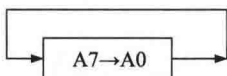
(4) 累加器带进位循环左移.

指令格式及其功能如下:

RLC A ; 

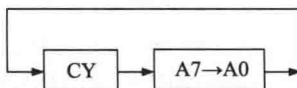
(5) 累加器内容循环右移.

指令格式及其功能如下:

RR A ; 

(6) 累加器带进位循环右移.

指令格式及其功能如下:

RRC A ; 

(7) 累加器半字节交换.

指令格式及其功能如下:

SWAP A ; (A)_{0~3} $\leftrightarrow$ (A)_{4~7}

2. 与(ANL)、或(ORL)、异或(XRL).

(1) 逻辑与.

指令格式及其功能如下:

ANL A, Rn ; (A) $\leftarrow$ (A) $\wedge$ (Rn)

ANL A, direct ; (A) $\leftarrow$ (A) $\wedge$ (direct)

ANL A, @ Ri ; (A) $\leftarrow$ (A) $\wedge$ ((Ri))

ANL A, #data ; (A) $\leftarrow$ (A) $\wedge$ data



ANL direct,A ; $(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$

ANL direct,#data ; $(\text{direct}) \leftarrow (\text{direct}) \wedge \text{data}$

(2) 逻辑或.

指令格式及其功能如下:

ORL A,Rn ; $(A) \leftarrow (A) \vee (Rn)$

ORL A,direct ; $(A) \leftarrow (A) \vee (\text{direct})$

ORL A,@ Ri ; $(A) \leftarrow (A) \vee ((Ri))$

ORL A,#data ; $(A) \leftarrow (A) \vee \text{data}$

ORL direct,A ; $(\text{direct}) \leftarrow (\text{direct}) \vee (A)$

ORL direct,#data ; $(\text{direct}) \leftarrow (\text{direct}) \vee \text{data}$

(3) 逻辑异或.

指令格式及其功能如下:

XRL A,Rn ; $(A) \leftarrow (A) \oplus (Rn)$

XRL A,direct ; $(A) \leftarrow (A) \oplus (\text{direct})$

XRL A,@ Ri ; $(A) \leftarrow (A) \oplus ((Ri))$

XRL A,#data ; $(A) \leftarrow (A) \oplus \text{data}$

XRL direct,A ; $(\text{direct}) \leftarrow (\text{direct}) \oplus (A)$

XRL direct,#data ; $(\text{direct}) \leftarrow (\text{direct}) \oplus \text{data}$

四、控制转移指令

1. 无条件转移(LJMP、AJMP、SJMP、JMP).

(1) 2KB 内跳转.

指令格式及其功能如下:

AJMP addr11 ; $(PC) + 2 \rightarrow (PC), \text{addr11} \rightarrow (PC)_{10 \sim 0}, (PC)_{15 \sim 11}$ 不变

(2) 长跳转.

指令格式及其功能如下:

LJMP addr16 ; $\text{addr16} \rightarrow (PC)$

(3) 相对转移(短跳转).

指令格式及其功能如下:

SJMP rel ; $(PC) + 2 \rightarrow (PC), (PC) + \text{rel} \rightarrow (PC)$

(4) 跳转(散转).

指令格式及其功能如下:

JMP @A + DPTR ; $(A) + (DPTR) \rightarrow (PC)$

2. 调用子程序及返回(LCALL、ACALL、RET、RETI).

(1) 2KB 内调用.

指令格式及其功能如下:

ACALL addr11 ; $(PC) + 2 \rightarrow (PC)$

; $(SP) + 1 \rightarrow (SP)$