



微机原理和接口技术

第四讲 指令系统与汇编程序1

提 纲

1. 指令系统概述

2. 寻址方式

3. 数据传送类指令

4. 算术运算类指令

5. 逻辑运算类指令

6. 控制转移类指令

7. 位操作指令

8. 查表指令的应用

9. 堆栈操作指令的应用

10. 十进制调整指令的应用

11. 逻辑指令与字节状态操作

12. 转移指令的应用

提 纲

1. 指令系统概述



指令系统概述

微控制器具有的指令集合即为该微控制器的指令系统（或指令集），指令系统中的各条指令对应不同的机器代码，是由微控制器内核设计人员确定的。

1. 指令分类

8051微控制器采用CISC结构的指令集，共有111条指令。可根据指令的长度（即指令机器码的字节数）、指令的执行速度（即指令的机器周期数）和指令的功能进行分类。



指令系统概述

1. 指令分类

(1) 按指令的长度（字节数）分类：

- **单字节指令49条**：1字节机器码，指令的操作数隐含在操作码中。

如INC Rn和MOV A, Rn 的机器码分别为00001xxx和11001xxx，其中xxx可表示的000-111，分别代表R0~R7，实际上各有8条指令。

- **双字节指令46条**：机器码有2个字节：第1字节为操作码，第2字节为操作数。

如MOV A, #data; 机器码：74H, data;

- **三字节指令16条**：机器码有3个字节：第1字节为操作码，第2、3字节为操作数。

如MOV DPTR, #data16; 机器码：90H, #data的高8位, #data的低8位



指令系统概述

1. 指令分类

(2) 根据指令的执行时间(速度)分:

- 单机器周期指令: 64条
- 双机器周期指令: 45条
- 四机器周期指令: 2条

(3) 根据指令功能划分:

- 数据传送类指令: 29条
- 算术运算类指令: 24条
- 逻辑运算类指令: 24条
- 控制转移类指令: 17条
- 位操作类指令: 17条

指令系统概述

2. 指令格式

指令的表示方式称为指令格式，任一条指令均由操作码和操作数两部分组成。操作码用来规定指令所要完成的操作；操作数是指令操作的对象。

指令的典型格式：

标号： 助记符 目的操作数, 源操作数 ;注释

如： **LOOP: ADD A, Rn ;加法**

- **标号**：是该指令的符号地址，根据需要设置。标号的第一个字符必须是字母，其余可以是符号或数字。
- **助记符**：表述指令的功能，规定执行某种操作；助记符用英文名称或缩写表示。
- **操作数**：是指令操作的对象，可以是具体数据、数据保存的地址、寄存器或标号等。对于有两个操作数的指令，左边的为目的操作数，右边的为源操作数。
- **注释**：是对该指令在程序中作用的说明，帮助阅读、理解和使用源程序。

指令系统概述

• 3. 指令代码

指令代码是用二进制数表示的编码，即机器代码。指令的第1字节为操作码，表示指令的功能，第2、3字节则为操作数。对于单字节指令，其操作数隐含在操作码中。（指令长度与指令时间没有相关性）

汇编指令	指令代码		指令长度	指令时间
	操作码	操作数		
MOV A,#40H	74	40	双字节	单周期
MOV A,40H	E5	40	双字节	单周期
RET	22	隐含	单字节	双周期
INC A	04	隐含，实际为A的内容	单字节	单周期
DIV AB	84	隐含，实际为A、B的内容	单字节	四周期



指令系统概述

- 4. 符号约定
- 在汇编指令系统中，常用一些符号来表示指令中的寄存器、存储单元或立即数等。8051微控制器指令系统中通常采用的符号和意义如下所示：
- **Rn (n=0~7)** : 当前选中的工作寄存器组的R0~R7，它的片内RAM地址由PSW中的RS1，RS0确定。

Ri (i=0, 1) : 当前选中的工作寄存器组中可作为地址指针的两个工作寄存器R0，R1。

#data: 8位立即数，即指令中给出的8位常数。

#data16: 16位立即数，即指令中给出的16位常数。

direct: 8位片内RAM单元（包括SFR）的直接地址。

addr16: 16位目的地址。用于LCALL和LJMP指令中，目的地址在64k字节的ROM空间。



指令系统概述

• 4. 符号约定

addr11: 11位目的地址。用于ACALL和AJMP指令中，目的地址必须与下一条指令的第一字节在同一个2k字节的ROM空间之内。（已很少使用）

rel: 8位带符号的偏移字节。用于SJMP和所有的条件转移指令中。偏移量以下一条指令第一字节地址为基址，地址偏移量范围在 $-128 \sim +127$ 。（即偏移量的补码形式表示）

bit: 片内RAM和SFR的直接寻址的位地址。

@: 间接寻址方式，表示间址寄存器的符号。

/: 位操作指令中。表示对该位先求反再参与操作，不影响该位原始值。



指令系统概述

- 4. 符号约定

(x): 表示x中内容。如 (30H) , 表示内部RAM 30H中的内容。

((x)): 表示以x中的内容为地址寻址。如 ((30H)) , 表示内部RAM 30H单元中的值为内部RAM的地址, 该地址单元中的内容。

←: 指令操作流程, 将箭头右边的内容送到箭头左边的单元中。

如MOV A, R1 ; A ← R1

提 纲

2. 寻址方式



寻址方式

寻址方式: 就是寻找指令中参与运算的操作数或操作数所在地址的方式。

寻址种类: 8051微控制器指令系统中共有 7种 寻址方式:

- 立即寻址
- 直接寻址
- 寄存器寻址
- 间址寻址: 寄存器间接寻址
- 变址寻址: 基址寄存器加变址寄存器寻址
- 相对寻址
- 位寻址



寻址方式

1. 立即寻址

操作数以立即数的形式在指令中直接给出，立即数表示为#data，#data16。数字前加“#”符号，即表示为立即数。

例： **MOV A,#dataH** **;(A) ←data**
 MOV DPTR,#data16 **;(DPTR)←data16**

例： MOV A,#30H 74H, 30H ; 操作码，立即数
 MOV DPTR,#8000H 90H, 80H, 00H ; 操作码，立即数

 XRL A, #0FH ANL P1, #0FH
 ADD A, #07H MOV R7, #20H
 MOV @R1, #55H

立即寻址的寻址空间为**ROM**（即操作数存放在**ROM**中）



寻址方式

2. 直接寻址

指令中给出的操作数是实际操作数的存储地址，即存放实际操作数的内部RAM的单元地址或SFR的地址，该单元的内容为指令的操作数。

例： **MOV A, direct** ;(A) ←(direct)

机器码： E5H, direct ; direct为操作数所在的内存地址

例： **MOV 30H, 50H** ; 操作码 直接地址，直接地址

 ; 85 30 50

MOV 30H, #50H ; 操作码 直接地址，立即数

 ; 75 30 50

(该类指令长度为2或3字节)



寻址方式

2. 直接寻址

例: **MOV 30H, P0** ; 操作码 直接地址, 直接地址
等同于

MOV 30H, 80H ; 操作码 直接地址, 直接地址
; 85 30 80

直接寻址的寻址空间:

内部RAM的低128个字节单元 (00H-7FH)

特殊功能寄存器区 (直接寻址是访问SFR的唯一方式)

寻址方式

3. 寄存器寻址

指令中给出操作数所在的寄存器，即寄存器的内容是操作数。

例： MOV A, R1 ;(A) \leftarrow (R1)
 INC R0

(该类指令大多数为单字节指令，操作数隐含在操作码中)

寄存器寻址的寻址空间：

R0~R7、A。



寻址方式

4. 寄存器间接寻址

- 操作数的地址存放在寄存器中，即寄存器的内容是操作数所在的内存地址。用于间接寻址的寄存器有 $R_i(i=0 \text{ 或 } 1)$ 、 $DPTR$ ，要在寄存器前加@。
- 例： $MOV \quad A, @R1 \quad \quad \quad ; (A) \leftarrow ((R1))$
- $MOVX \quad A, @DPTR \quad \quad \quad ; (A) \leftarrow ((DPTR))$
- 寄存器间接寻址的寻址空间：
- 内部RAM： $00H - FFH$ ； $(@R0, @R1)$
- 外部RAM： $0000H - FFFFH$ ； $(@R0, @R1, @DPTR)$

寻址方式

- 直接寻址和寄存器间接寻址举例：
 - 对于SFR，只能用直接寻址方式，直接地址可用单元地址或寄存器名，
其汇编结果是一样的，机器码均为：E5 80

MOV A, P0 ; (A) ← (P0) (以SFR寄存器表示)
MOV A, 80H ; (A) ← (P0) (以单元地址表示)
 - 对于内部RAM的80H-FFH（地址与SFR重叠），只能用寄存器间接寻址。
如：将内部RAM 80H单元的内容取到A：

MOV R0, #80H ; 用R0作为间址寄存器，指向80H
MOV A, @R0 ; 将R0间址的内存单元的内容读到A
 - 对于内部RAM的00H-7FH，既可以用直接寻址，也可以用寄存器间接寻址。



寻址方式

5. 变址寻址（基址寄存器加变址寄存器间接寻址）

以DPTR或PC作基址寄存器，A作变址寄存器（存放8位无符号数），两者相加形成的16位程序存储器地址，作为操作数所在地址。

8051微控制器共有3条变址寻址的指令（均为单字节指令）

- `MOVC A, @A+DPTR` ; $(A) \leftarrow ((A)+(DPTR))$ 93H
- `MOVC A, @A+PC` ; $(A) \leftarrow ((A)+(PC))$ 83H
- `JMP @A+DPTR` ; $(PC) \leftarrow (A)+(DPTR)$ 73H

前两条是程序存储器读指令（也称查表指令），后一条是无条件散转指令。

变址寻址的寻址空间：

程序存储器



寻址方式

6. 相对寻址

用于程序控制，利用指令修改PC指针内容实现转移。指令中给出的操作数为程序转移的偏移量（用rel表示）。以PC的当前值（执行这条指令时的PC）加上偏移量，所得结果即为转移的目的地址。

目的地址= 转移指令下一条指令首址+rel

= 转移指令所在地址+转移指令字节数+rel 等价于 当前PC值+rel

例： SJMP rel ; JNZ rel ; DJNZ Rn, rel 等

rel 是一个带符号数的8位二进制补码数，其范围为 (-128) - (+127)

相对寻址的寻址空间：

程序存储器。以转移指令的下一条指令所在地址为基点，向低地址（PC值减小）方向最大可转移128字节，向高地址（PC值增大）方向最大可转移127个字节。



寻址方式

7. 位寻址

8051微控制器具有位寻址和操作功能的布尔处理器，可以对位寻址空间的各位直接进行操作。

位寻址的寻址空间：

8051 MCU中的位地址空间（通用内部RAM20H~2FH对应的00H~7FH的128bit，SFR中的可位寻址的83bit，共211bit。）

SFR中的可寻址位既有位地址又有符号名称，如PSW的位7为Cy，其位地址为D7H；P0的位0为P0.0，位地址为80H。建议在指令中使用位符号名称，以增加程序的可读性。

例：MOV C, P1.0 ; P1口中最低位P1.0口线的状态输入到Cy。



寻址方式

7. 位寻址

8051微控制器具有位寻址和操作功能的布尔处理器，可以对位寻址空间的各位直接进行操作。

位寻址的寻址空间：

8051 MCU中的位地址空间（通用内部RAM20H~2FH对应的00H~7FH的128bit，SFR中的可位寻址的83bit，共211bit。）

SFR中的可寻址位既有位地址又有符号名称，如PSW的位7为Cy，其位地址为D7H；P0的位0为P0.0，位地址为80H。建议在指令中使用位符号名称，以增加程序的可读性。

例：MOV C, P1.0 ; P1口中最低位P1.0口线的状态输入到Cy。



寻址方式

7. 位寻址

位寻址（对于位）与直接寻址（对于字节）的地址形式完全相同，通过操作码来区分。

- 例：MOV C, 20H ; $C \leftarrow (20H)$,位传送
-
- MOV A, 20H ; $(A) \leftarrow (20H)$, 字节传送
-
- CLR 10H ; $(10H) \leftarrow 0$, 位清0, 有专门指令 (CLR)
- MOV 10H, #00H ; $(10H) \leftarrow 00H$, 字节清0, 只能用赋值方法



寻址方式

7. 位寻址

为提高程序的可读性，对于SFR和SFR中的位地址通常用符号表示。

如：C、P、OV、P1.0-P1.7、EA等等；累加器A的最高位表示为ACC.7等。

```
MOV    P0, #0FFH    ; 等价于  MOV    80H, #0FFH
MOV    IE, #81H
MOV    C, ACC.7      ; C ← ACC.7
MOV    P1.0, C       ; P1.0 ← C
SETB   EA
JNB    P, rel
```

寻址方式



寻址方式

8. 寻址方式和寻址空间

指令的寻址方式通常是指源操作数的寻址方式。8051指令系统源操作数的7种寻址方式可使用的变量与可寻址的空间有所不同，概括如下。

寻址方式	使用的变量	寻址空间
直接寻址	direct	内部RAM低128字节、SFR
寄存器寻址	R0~R7、A	R0~R7、A
寄存器间接寻址	@R0~R1, SP (PUSH、POP)	内部RAM的256字节
	@R0~R1、@DPTR	外部RAM
立即寻址	#data、#data16	程序存储器
变址寻址	基址寄存器: DPTR、PC; 变址寄存器: A;	程序存储器
相对寻址	PC+偏移量	程序存储器
位寻址	bit、C	位寻址空间

Thank you!

