

《单片机原理及接口技术》-内部 RAM 数据传送指令

一、内部 RAM 数据传送指令

操作码助记符: MOV

格式: MOV 目的操作数, 源操作数

功能: 传送字节变量 (目的操作数) \leftarrow (源操作数)

1. 以累加器 A 为目的操作数的数据传送指令

MOV A, #data ; (A) \leftarrow data 例如: MOV A, #66H ; (A) \leftarrow 66H

MOV A, Rn ; (A) \leftarrow (Rn) 例如: MOV A, R5 ; (A) \leftarrow (R5)

MOV A, @Ri ; (A) \leftarrow ((Ri)) 例如: MOV A, @R0 ; (A) \leftarrow ((R0))

MOV A, direct ; (A) \leftarrow (direct) 例如: MOV A, 70H ; (A) \leftarrow (70H)

2. 以寄存器 Rn 为目的操作数的数据传送指令

MOV Rn, #data ; (Rn) \leftarrow data 例如: MOV R2, #68H ; (R2) \leftarrow 68H

MOV Rn, direct ; (Rn) \leftarrow (direct) 例如: MOV R3, 50H ; (R3) \leftarrow (50H)

MOV Rn, A ; (Rn) ← (A) 例如: MOV R4, A ; (R4) ← (A)

3. 以 Ri 间址单元为目的的操作数的数据传送指令

MOV @Ri, #data ; ((Ri)) ← data 例如: MOV @R1, #68H
; 若 (R1) = 30H, 则 (30H) = 68H

MOV @Ri, direct ; ((Ri)) ← (direct) 例如: MOV @R0, 70H
; 若 (R0) = 40H, 则 (40H) = (70H)

MOV @Ri, A ; ((Ri)) ← (A) 例如: MOV @R1, A
; 若 (R1) = 50H, 则 (50H) = (A)

4. 以直接地址 direct 为目的的操作数的数据传送指令

MOV direct, #data ; (direct) ← data 例如: MOV 30H, #68H ; (30H) ← 68H

MOV direct2, direct1 ; (direct2) ← (direct1) 例如: MOV 30H, 40H ; (30H) ← (40H)

MOV direct, Rn ; (direct) ← (Rn) 例如: MOV 30H, R3 ; (30H) ← (R3)

MOV direct, @Ri ; (direct) ← ((Ri)) 例如: MOV 30H, @R0
; 若 (R0) = 40H 则 (30H) ← (40H)

MOV direct, A ; (direct) ← (A) 例如: MOV 30H, A ; (30H) ← (A)

5. 16 位数据传送指令

MOV DPTR, #data16 ; (DPTR) ← #data16

例如:

MOV DPTR, #1234H ; (DPTR) = 1234H, 其中 (DPH) = 12H, (DPL) = 34H

综上所述, MCS—51 系列单片机的内部 RAM 数据传送类指令的路径如图 3—7 所示。

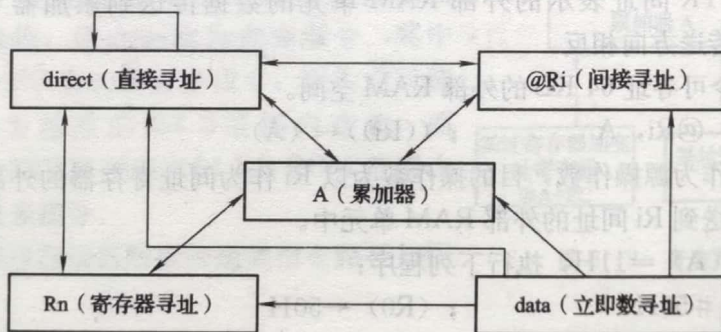


图 3—7 内部 RAM 数据传送类指令的路径

例 3—1 分析下列程序段的功能。

(1) MOV 50H, 20H

(2) MOV R1, #50H
MOV @R1, 20H

(3) MOV A, 20H
MOV 50H, A

解: 以上三段程序的功能都是把存放在内部 RAM 20H 单元的数据送到 50H 单元中。

例 3—2 分析下列程序段的功能。

(1) MOV A, 30H

(2) MOV R1, #30H
MOV A, @R1

解：以上两段程序的功能都是把存放在内部 RAM 30H 单元的内容送到累加器 A 中。

例 3—3 已知内部 RAM 30H 单元内容为 50H，请分析以下两段程序的功能。

(1) MOV R1, 30H

(2) MOV A, 30H

MOV R1, A

解：(1) MOV R1, 30H；功能为 $(R1) \leftarrow (30H)$ ，即 $(R1) = 50H$

(2) MOV A, 30H；功能为 $(A) \leftarrow (30H)$

MOV R1, A； $(R1) \leftarrow (A)$ ，执行后 $(R1) = 50H$

可见，这两段程序完成了相同的功能。