

Homework 7 — April 17

Lecturer: Ke Zhang

Completed by: 吉骏雄

[hw7]: 7.12, 7.13, 7.15, 7.16

7.12 画出“SUB R1”指令对操作数的寻址及减法过程的流程图。设被减数和结果存于 ACC 中，表示间接寻址，R1 寄存器的内容为 2074H。

解

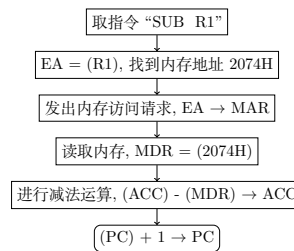


图 7.1. 题 7.12 流程图

7.13 画出执行“ADD *-5”指令（* 为相对寻址特征）的信息流程图。设另一个操作数和结果存于 ACC 中，并假设 (PC) = 4000H。

解

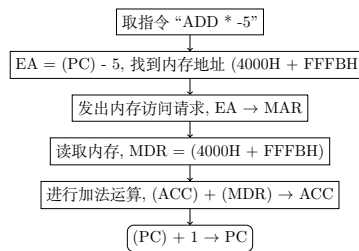


图 7.2. 题 7.13 流程图

7.15 一相对寻址的转移指令占 3 个字节，第一字节是操作码，第二、三字节为相对位移量，而且数据在存储器中采用以高字节地址为字地址的存放方式（大端序）。假设 PC 当前值是 4000H。试问当结果为 0，执行“JZ * +35”和“JZ * -17”指令时，该指令的第二、第三字节的机器代码各为多少？

解

1. “JZ * +35”: $35 = 0x23$. 取出 3 字节的指令后, $(PC)+3 \rightarrow PC$, 现值为 4003H. 寻址之后的偏移应该是 $0x23 - 3 = 0x20$, 因此指令的二/三字节存储的应该是 00 20

2. “JZ * -17”: 17 = 0xFFEF. 取出 3 字节的指令后, (PC)+3 \rightarrow PC, 现值为 4003H. 寻址之后的偏移应该是 0xFFEF - 3 = 0xFFEC, 因此指令的二/三字节存储的应该是 *FF EC*

7.16 某机主存容量为 4Mx16 位, 且存储字长等于指令字长, 若该机指令系统可完成 108 种操作, 操作码位数固定, 且具有直接、间接、变址、基址、相对、立即等六种寻址方式, 试回答以下问题。

1. 画出一地址指令格式并指出各字段的作用。
2. 该指令直接寻址的最大范围。
3. 一次间接寻址和多次间接寻址的寻址范围。
4. 立即数的范围 (十进制表示)。
5. 相对寻址的位移量 (十进制表示)。
6. 上述六种寻址方式的指令中哪一种执行时间最短, 哪一种最长, 为什么? 哪一种便于程序浮动, 哪一种最适合处理数组问题?
7. 如何修改指令格式, 使指令的寻址范围可扩大到 4M?
8. 为使一条转移指令能转移到主存的任一位置, 可采取什么措施? 简要说明之。

解

1. 一地址指令格式如图 7.3. 各字段作用: OP 是指令的操作码字段, 为提供 108 种操作, 需要有 7 位以实

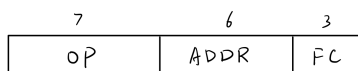


图 7.3. 题 7.16

现 $2^7 = 128$ 种组合. ADDR 是寻址地址, 用于不同的寻址. 使用补码. FC 是寻址类型标志段, 为提供 6 种类型, 需要有 3 位以实现 $2^3 = 8$ 种组合.

2. 有 6 位地址, 最大范围是 $2^6 = 64$, 单位为字 (16 位).
3. 一次间接寻址的寻址范围来自内存中一个字存储的地址, 最大范围是 $2^{16} = 64K$ 字. 多次间接寻址, 可每次移动 $-32K \sim +32K - 1$ 字, n 次即可转移 $-32nK \sim +32nK - n$ 字, 范围是 $64nK - n + 1$ 字.
4. 因为有 6 位, 故为 $-32 \sim 31$.
5. $-32 \sim 31$.
6. 立即寻址时间最短, 因为立即数可以直接访存, 不需要复杂寻址.
间接寻址时间最长, 因为需要进行 (至少) 一次访存来寻址, 访存花时间长.

随程序浮动, 使用相对寻址最合适, 因为这个操作数的结果会随着程序执行到的位置浮动, 是跟 PC 有关的.

处理数组, 使用变址寻址最合适, 因为可以设定一个可变的位置来寻址, 如果一直访问一个数组的话就不需要改变这个寄存器.

7. $4M = 2^{22}$, 因此需要 22 位的地址, 而 $22 = 16 + 6$, 需要将指令改成双字长的 (即添加 16 位地址), 如图 7.4.

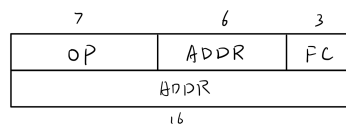


图 7.4. 题 7.16

8. 按照如上所述的方式, 使用直接寻址即可实现.