第7章 指令系统 习题与解答

> 课件来源：cod 版 Slide 371–412。共 18 题：课件原题 9 题（100% 收录课件中的例题/示例）+ 考点补充题 9 题。计算题均给出全部步骤；课件存疑处已标注。

一、课件原题

题1（课件 p.373–374）

指令字长 16 位，每个地址码 4 位，采用扩展操作码技术。课件方案中三地址、二地址、一地址、零地址指令各最多多少条？写出各级操作码的二进制范围；并说明：三地址操作码每减少一种，可多构成多少种二地址指令？二地址操作码每减少一种呢？

解答：

1. 三地址：操作码 4 位，范围 0000 ~ 1110，共 **15 条**；1111 留作扩展前缀。
2. 二地址：操作码 8 位，范围 1111 0000 ~ 1111 1110，共 **15 条**；1111 1111 留作前缀。
3. 一地址：操作码 12 位，范围 1111 1111 0000 ~ 1111 1111 1110，共 **15 条**；1111 1111 1111 留作前缀。
4. 零地址：操作码 16 位，范围 1111 1111 1111 0000 ~ 1111 1111 1111 1111，共 **16 条**（最后一级不再留前缀）。
5. 三地址操作码每减少一种，让出的 4 位前缀可扩展出 2^4 = **16 种**二地址指令；同理二地址每减少一种可多构成 2^4 = **16 种**一地址指令（课件 p.374）。

要点：短操作码不能是长操作码的前缀，故每级必须保留全 1 编码作扩展窗口。

题2（课件 p.375–376）

设指令字长 32 位、操作码固定 8 位。分别求四地址、三地址、二地址、一地址指令的地址码位数、寻址范围和访存次数；并说明 PC 代替 A4、A3 用 A1 或 A2 代替、结果存于 ACC、ACC 代替 A1（或 A2）各得到什么格式。

解答：

地址数	地址码位数计算	每段地址位数	寻址范围	访存次数（含取指）
四地址	(32−8)/4 = 6	6	2^6 = 64	4（取指1+取数2+存结果1）
三地址	(32−8)/3 = 8	8	2^8 = 256	4（同上）
二地址	(32−8)/2 = 12	12	2^{12} = 4 K	4；结果存 ACC 时 3（取指1+取数2）
一地址	32−8 = 24	24	2^{24} = 16 M	2（取指1+取数1，结果写 ACC）
零地址	无地址码	—	—	操作数隐含（堆栈/ACC）

演化：四地址中 PC 代替 A4（下条指令地址自动给出）→ 三地址；三地址中 A3 用 A1 或 A2 代替（结果覆盖源操作数）→ 二地址；二地址中 ACC 代替 A1（或 A2）且结果存 ACC → 一地址；操作数全隐含 → 零地址（课件 p.375–376）。

题3（课件 p.382）

地址 305 处为指令 SKP DZ（D=0 则跳，D 为完成触发器），306、307 为后续两条指令。问 D=0 和 D=1 时，下一条执行的指令地址各是多少?

解答:

- SKP 的功能是"跳过一条指令"（课件 p.382）。
- D=0：条件满足，跳过 306，下一条执行 **307**。
- D=1：条件不满足，不跳，顺序执行 **306**。

题4（课件 p.386）

读课件指令寻址示例图（0: LDA 1000；1: ADD 1001；2: DEC 1200；3: JMP 7；4: LDA 2000；5: SUB 2001；6: INC；7: STA 2500；8: LDA 1100），写出实际执行流中每次取指后 PC 的内容及所用的指令寻址方式。

解答:

执行顺序	指令地址	指令	取指后 PC	获得下条指令的寻址方式
1	0	LDA 1000	1	顺序 (PC)+1→PC
2	1	ADD 1001	2	顺序
3	2	DEC 1200	3	顺序
4	3	JMP 7	4，执行后 PC=7	跳跃（由转移指令指出）
5	7	STA 2500	8	顺序
6	8	LDA 1100	9	顺序

即执行流 0→1→2→3→7→8→9；地址 4、5、6 的指令被跳过（课件 p.386）。

题5（课件 p.391）

课件"间接寻址编程举例"中，主程序在 80 与 201 两处调用同一子程序，返回地址分别为 81 与 202。说明如何利用间接寻址实现子程序返回，为什么不必修改子程序中的指令。

解答：

1. 约定一个主存单元 A 存放返回地址，子程序最后一条指令为 **JMP @ A**（@ 为间址特征，EA=(A)）。
2. 调用点 80 处调用前把返回地址 81 写入 A，即 (A)=81；调用点 201 处调用前把 202 写入 A，即 (A)=202（课件 p.391）。
3. 子程序执行 JMP @ A 时，EA=(A)，取出的是当前调用点写入的返回地址，故分别跳回 81 或 202。
4. 返回地址作为**数据**存放在单元 A 中，子程序代码本身不变，因此同一子程序可被多处调用而无需修改指令——这正是间接寻址"便于编制程序"的体现（课件 p.390–391）。

题6（课件 p.397）

设数据块首地址为 D，求 N 个数的平均值。分别用直接寻址和变址寻址编写程序，统计指令条数并比较。

解答：

（本题助记符沿用课件模型机，如 LDA/STA/LDX/INX/CPX/BNE，非 8086 语法。）

直接寻址（课件 p.397）：

```
LDA  D          ; 取第 1 个数
ADD  D+1        ; 加第 2 个数
ADD  D+2        ; 加第 3 个数
; ……
ADD  D+(N-1)    ; 加第 N 个数
DIV  #N         ; 除以 N（立即数）
STA  ANS        ; 存平均值
```

条数 = 1 + (N−1) + 1 + 1 = **N+2 条**，随 N 线性增长。

变址寻址（X 为变址寄存器，D 为形式地址，课件 p.397）：

```
LDA  #0         ; ACC 清 0（累加器）
LDX  #0         ; 变址寄存器 X 清 0
M:   ADD  X, D  ; EA=(X)+D，累加第 (X)+1 个数
INX             ; (X)+1→X
CPX  #N         ; (X) 与 #N 比较
BNE  M          ; 结果不为零则转 M，循环 N 次
DIV  #N         ; 求平均
STA  ANS        ; 存结果
```

共 **8 条**，与 N 无关。结论：变址寻址把"变化的地址"交给寄存器，形式地址 D 不变，特别适合数组/循环处理（课件 p.396–397）。

题7（课件 p.399）

课件相对寻址举例：M: ADD X, D；M+1: INX；M+2: CPX #N；M+3: BNE M（相对寻址特征 *）。课件给出位移量 −3、EA = (M+3) − 3 = M。解释该位移量的含义，并说明为什么程序浮动时位移量不变。

解答：

1. 课件口径：把 PC 取为 BNE 指令自身地址 M+3，位移量 A = 目标 M − (M+3) = **−3**（补码表示），故 EA = (M+3) + (−3) = M，跳回循环首 ADD X, D（课件 p.399）。
2. 程序浮动时整个程序平移，M 随之变化，但 BNE 与 ADD 之间相隔 3 条指令的**相对距离不变**，故位移量 −3 不变，无需修改指令——相对寻址支持程序浮动（课件 p.399）。
3. 课件原文如此，存疑：p.399 以"当前指令地址 M+3"作 PC；若按 p.400"取指后 PC 自增"的约定，BNE 取指后 PC = M+4，位移量应为 M − (M+4) = **−4**。两页口径不一致；考试时以题目给定的 PC 口径为准（通常用取指后 PC，见题8）。

题8（课件 p.400）

按字节寻址。当前指令地址 PC = 2000H，指令 JMP *+8 为二字节指令，转移目的地址为 2008H。求该指令第二字节（位移量）的十六进制值。

解答：

1. 取出二字节指令后 PC 自增 2：PC = 2000H + 2 = **2002H**（课件 p.400）。
2. 相对寻址 EA = (PC) + A，要求 EA = 2008H，故 A = 2008H − 2002H = **06H**。
3. 即指令第二字节为 06H。注意不是 08H：位移量相对的是**取指后**的 PC，不是指令首地址（课件 p.400）。

题9（课件 p.402）

课件堆栈寻址举例：栈底 200H，栈向低地址生长。PUSH A 前 ACC=15、SP=200H、(200H)=X；POP A 前 ACC=Y。写出 PUSH A 后与 POP A 后 ACC、SP 及相关主存单元的内容，并说明进栈/出栈的操作顺序。

解答：

状态	ACC	SP	主存
PUSH A 前	15	200H	(200H)=X
PUSH A 后	15	1FFH	(1FFH)=15，(200H)=X
POP A 前	Y	1FFH	(1FFH)=15
POP A 后	15	200H	(1FFH)=15（保留），(200H)=X

操作顺序（课件 p.401–402）：

- 进栈：先 (SP)−1→SP（200H−1=1FFH），再把 (ACC) 写入新栈顶 (1FFH)=15；
- 出栈：先把栈顶内容 ((SP))=(1FFH)=15 读入 ACC，再 (SP)+1→SP（1FFH+1=200H）；
- 出栈后原栈顶单元数据仍保留，只是 SP 不再指向它（课件 p.402 图中 POP 后 1FFH 仍为 15）。

二、考点补充题

题10【补充•考点：寻址方式与访存次数】（基础•选择）

一次间接寻址在**执行阶段**的访存次数为（ ）。 A. 1 B. 2 C. 3 D. 4

解答：选 **B**。

- 执行阶段第 1 次访存：按形式地址 A 取有效地址 EA=(A)；第 2 次访存：按 EA 取操作数。共 2 次（课件 p.390）。
- 排除 A：1 次是直接寻址、寄存器间接寻址的执行阶段访存数；排除 C、D：一次间址只需取 EA 与取操作数两步，3 次及以上对应多次间址或含写回/取指的口径，与"一次间址执行阶段"不符。

题11【补充•考点：十种寻址方式的 EA 公式】（基础•填空）

填写下表的有效地址表达式与执行阶段访存次数（软堆栈按每次进/出栈计）。

寻址方式	EA	执行阶段访存
立即	①	②
直接	③	1
隐含	隐含于操作码（不设空）	视指令（如 ADD A 为 1）
间接（一次）	④	⑤
寄存器	⑥	0
寄存器间接	⑦	⑧
基址	⑨	1
变址	⑩	1
相对	⑪	1
堆栈	⑫	1

解答：① 无 EA，操作数即形式地址 A；② 0；③ A；④ (A)；⑤ 2；⑥ Ri；⑦ (Ri)；⑧ 1；⑨ (BR)+A；⑩ (IX)+A；⑪ (PC)+A（PC 为取指后内容）；⑫ (SP)。（课件 p.387–403）

题12【补充•考点：基址寻址与变址寻址辨析】（基础•简答）

基址寻址与变址寻址的 EA 公式形式相同，简述两者的区别（寄存器内容谁设定、执行中谁可变、典型用途）。

解答（评分要点）：

1. 公式：基址 EA=(BR)+A；变址 EA=(IX)+A（课件 p.394、p.396）。
2. 设定者：基址寄存器内容由**操作系统或管理程序**确定；变址寄存器内容由**用户**给定（课件 p.394–396）。
3. 执行中变化：基址 **BR 不变、形式地址 A 可变**；变址 **IX 可变、形式地址 A 不变**（课件 p.394、p.396）。
4. 用途：基址面向**多道程序/程序重定位**（扩大寻址范围）；变址面向**数组、循环**处理（课件 p.394、p.396–397）。
5. 寄存器来源：两者都可用专用寄存器或通用寄存器（课件 p.395–396）。

题13【补充•考点：扩展操作码设计】（综合•计算）

指令字长 16 位，每段地址码 4 位。要求三地址指令 12 条、二地址指令 62 条、一地址指令 30 条，问零地址指令最多可有多少条？给出一组可行的操作码分配。

解答：

1. 4 位操作码共 $2^4=16$ 个编码；三地址用 12 个（0000～1011），剩 16−12=4 个前缀（1100、1101、1110、1111）。
2. 二地址空间 = $4\times2^4 = 64$；用 62 个（1100 0000～1111 1101），剩 64−62=2 个前缀（1111 1110、1111 1111）。
3. 一地址空间 = $2\times2^4 = 32$；用 30 个（1111 1110 0000～1111 1111 1101），剩 32−30=2 个前缀（1111 1111 1110、1111 1111 1111）。
4. 零地址空间 = 2×2^4 = **32 条**（1111 1111 1110 0000～1111 1111 1111 1111），最后一级不留前缀。

答：零地址指令最多 32 条。校验：12 + 62/16 + 30/256 + 32/4096 = 12/16 + 62/256 + 30/4096 + 32/65536 = 0.75+0.2421875+0.00732421875+0.00048828125 = 1，编码空间恰好用完，方案可行。

题14【补充•考点：有效地址综合计算】（综合•计算）

按字编址、每指令 1 字。当前指令地址 100，取指后 PC=101，形式地址 A=200。已知 (200)=300、(300)=400、(1200)=600、(250)=700、(301)=800；(R1)=300、(BR)=1000、(IX)=50。求立即、直接、间接、寄存器、寄存器间接、基址、变址、相对八种方式下的有效地址（或操作数来源）与操作数。

解答：

寻址方式	EA 计算	EA	操作数
立即	操作数=A，无 EA	—	200
直接	EA=A	200	(200)=300
间接	EA=(A)=(200)	300	(300)=400
寄存器	EA=R1	R1	(R1)=300
寄存器间接	EA=(R1)	300	(300)=400
基址	EA=(BR)+A=1000+200	1200	(1200)=600
变址	EA=(IX)+A=50+200	250	(250)=700
相对	EA=(PC)+A=101+200	301	(301)=800

要点：① 相对寻址必须用取指后 PC=101，用 100 会错得 EA=300；② 间接与寄存器间接 EA 相同但地址来源不同（主存 vs 寄存器）；③ 括号一律表示"取内容"。

题15【补充•考点：堆栈寻址与 SP 修改】（基础•计算）

栈向低地址生长、SP 指向栈顶。初值 SP=2000H。① 按字节编址、存储字长 32 位：连续 3 次进栈、再 2 次出栈后 SP 为多少？栈顶地址？② 若改为按字编址，同样操作后 SP 为多少？

解答：

① 按字节编址、字长 32 位：每次进栈 (SP)−4→SP，出栈 (SP)+4→SP（课件 p.403）。

- 进栈 1：2000H−4=1FFCH；进栈 2：1FFCH−4=1FF8H；进栈 3：1FF8H−4=1FF4H；
- 出栈 1：1FF4H+4=1FF8H；出栈 2：1FF8H+4=1FFCH；
- 最终 **SP=1FFCH**，栈顶地址即 1FFCH。

② 按字编址：每次 ±1（课件 p.403）。进栈：1FFFH、1FFEH、1FFDH；出栈：1FFEH、1FFFH；最终 **SP=1FFFH**。

结论：SP 修改量 = 一个存储字的字节数（按字编址时为 1 个地址单位），与编址方式直接相关。

题16【补充•考点：RISC 与 CISC 比较】（基础•简答）

简述 RISC 的主要特征（至少 5 条），并指出 RISC 相对 CISC 的一个主要缺点。

解答（评分要点，课件 p.410–412）：

1. 只选用使用频度高的简单指令，复杂功能由简单指令组合；
2. 指令长度固定、格式种类少、寻址方式少；
3. 只有 LOAD/STORE 指令访存；

4. 采用流水技术，一个时钟周期完成一条指令；
5. 组合逻辑控制器（CISC 为微程序控制器）；
6. CPU 中多个通用寄存器（CISC 多专用寄存器）；
7. 采用优化编译程序。
8. 优点补充：充分利用 VLSI 面积、提高速度、便于设计降低成本提高可靠性、利于编译优化。
9. 缺点：**不易实现指令系统兼容**（课件 p.412）。

题17【补充•考点：8086 指令格式与指令字长】（基础•选择）

下列 8086 指令中，指令字长为 4 字节的是（ ）。 A. ADD AX, BX B. ADD AX, 3048H C. ADD AX, [3048H] D. INC AX

解答：选 **C**（课件 p.408）。

- A：寄存器–寄存器二地址，2 字节，排除；
- B：寄存器–立即数二地址，操作码 1+立即数 2=3 字节，排除；
- C：寄存器–存储器二地址，操作码 1+寻址字节 1+位移 2=4 字节，正确；
- D：一地址，1 字节，排除。
- 辨析：B 与 C 只差方括号——无括号是立即数，有括号才是存储器直接寻址。

题18【补充•考点：哈夫曼操作码（补充常识）】（拔高•计算）

某机 7 条指令的使用频度依次为 0.40、0.20、0.15、0.10、0.08、0.05、0.02。① 构造哈夫曼操作码，给出各指令码长与一种编码；② 求平均码长；③ 与等长操作码比较，并说明为何实际机器不全部采用哈夫曼操作码。

解答：

① 哈夫曼合并过程（每次合并两个最小频度）：

1. 0.02+0.05=0.07；
2. 0.07+0.08=0.15；
3. 0.10+0.15=0.25；
4. 0.15(节点)+0.20=0.35；
5. 0.25+0.35=0.60；
6. 0.40+0.60=1.00。

一种编码（左 0 右 1）：I1(0.40)=0；I4(0.10)=100；I3(0.15)=101；I2(0.20)=111；I5(0.08)=1101；I7(0.02)=11000；I6(0.05)=11001。码长依次为 **1、3、3、3、4、5、5** 位（任一编码均非其他编码的前缀）。

② 平均码长 $L = \Sigma\, p_i \times l = 0.40\times1 + 0.20\times3 + 0.15\times3 + 0.10\times3 + 0.08\times4 + 0.05\times5 + 0.02\times5 = 0.40+0.60+0.45+0.30+0.32+0.25+0.10 =$ **2.42 位/指令**。

③ 等长操作码需 $\lceil \log_2 7 \rceil$ = **3 位**；哈夫曼平均节省 3−2.42 = 0.58 位/指令。但哈夫曼操作码长度不规则，译码硬件复杂、不利于流水与规整格式，故实际机器多用**等长扩展操作码**折中（课件 p.373–374 只讲等长扩展；哈夫曼为

补充常识）。