第7章 指令系统

> 课件来源：cod 版课件 Slide 371–412（extracted/cod/chapters/ch07.md），单一版本、以其为准。本章是组成原理期末的核心章：扩展操作码设计、有效地址（EA）计算、基址/变址/相对寻址辨析、RISC 与 CISC 对比几乎必考，计算题与简答题分值高。

0. 本章考点清单

考点	常见题型	重要度
有效地址计算与寻址方式识别（10 种数据寻址）	计算、选择、填空	★★★
扩展操作码设计与计算（等长扩展；哈夫曼为补充常识）	计算、综合设计	★★★
RISC 与 CISC 的特征与比较	简答、选择	★★★
地址码个数与访存次数、寻址范围（四→零地址）	计算、填空	★★
基址 / 变址 / 相对寻址的区别与用途	简答、选择	★★
堆栈寻址与 SP 修改量（编址方式相关）	计算、填空	★★
指令字长与指令格式举例（PDP-8、PDP-11、IBM 360、8086）	选择、填空	★★
指令寻址（顺序 (PC)+1、跳跃）	填空、选择	★
操作数类型、大小端存放与边界对准	选择、填空	★
操作类型（传送/算术逻辑/移位/转移/调用返回/陷阱/I/O）	选择、简答	★
指令格式设计考虑因素（兼容性等）	简答	★

1. 机器指令的一般格式（操作码 + 地址码）

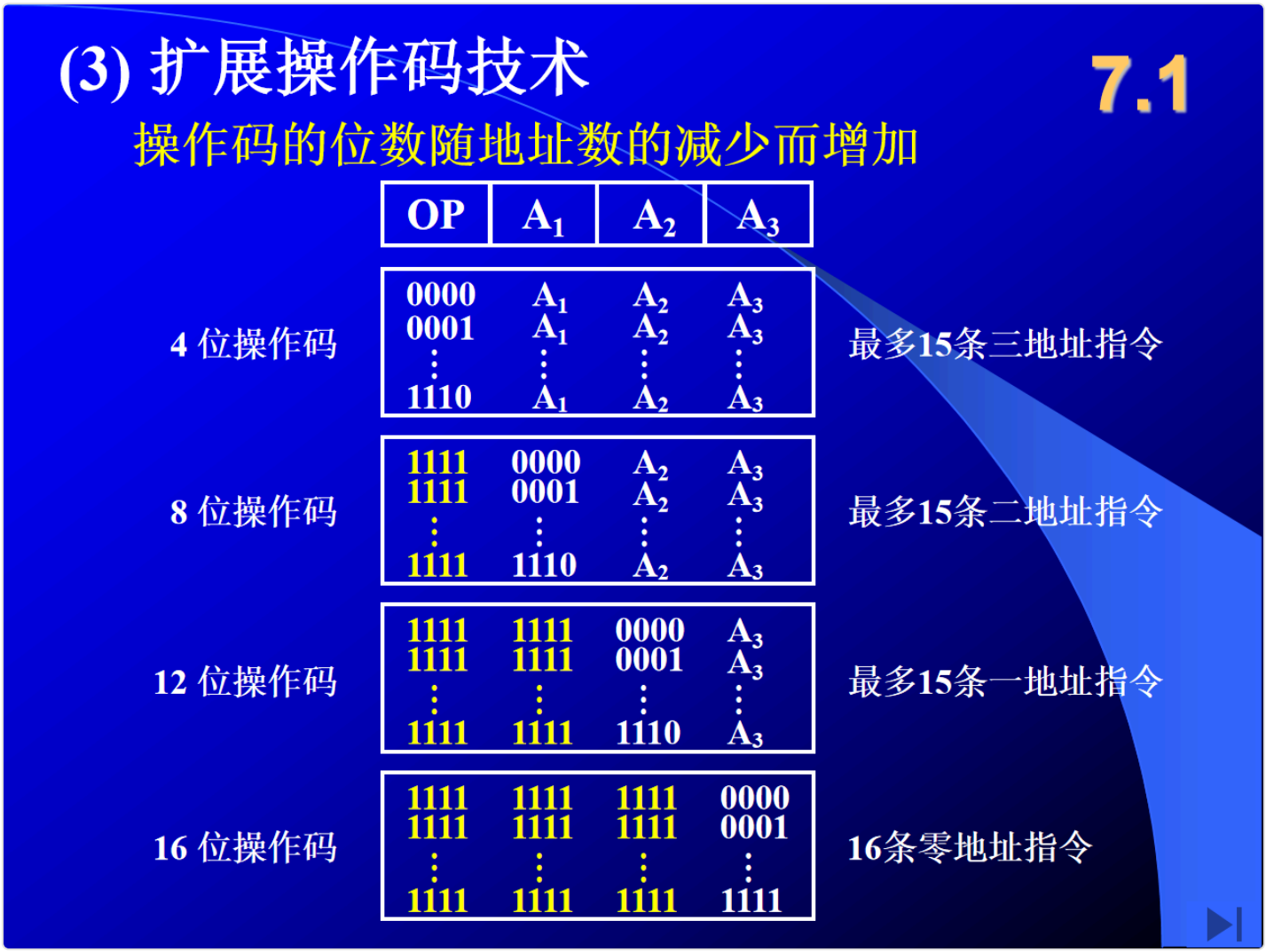
1.1 指令格式与操作码

一条机器指令 = 操作码字段 + 地址码字段。操作码反映机器做什么操作（课件 p.372）。

- **长度固定**：操作码位数固定，译码简单，用于 RISC（课件 p.372）。
- **长度可变**：用于指令字长较长的情况，如 IBM 370，操作码 8 位，且操作码分散在指令字的不同字段中（课件 p.372）。
- **扩展操作码技术**：操作码的位数随地址数的减少而增加（课件 p.373），见 1.2。

1.2 扩展操作码技术（★★★ 计算题高发）

设指令字长 16 位、每个地址码 4 位，课件给出的经典方案（课件 p.373）：



指令类型	操作码位数	操作码范围（二进制）	条数
三地址	4	0000 ~ 1110	最多 15
二地址	8	1111 0000 ~ 1111 1110	最多 15
一地址	12	1111 1111 0000 ~ 1111 1111 1110	最多 15
零地址	16	1111 1111 1111 0000 ~ 1111 1111 1111 1111	16

要点：

1. 每一级把全 1 编码（1111、1111 1111、1111 1111 1111）留作**扩展前缀**，不分配给本级指令，否则长操作码会与短操作码冲突（课件 p.373 图中黄色前缀）。
2. 三地址操作码每减少一种，可多构成 $2^4 = 16$ 种二地址指令；二地址操作码每减少一种，可多构成 $2^4 = 16$ 种一地址指令（课件 p.374）。即：**上一级每让出 1 个编码，下一级多出 2^(地址码位数) 个编码**。

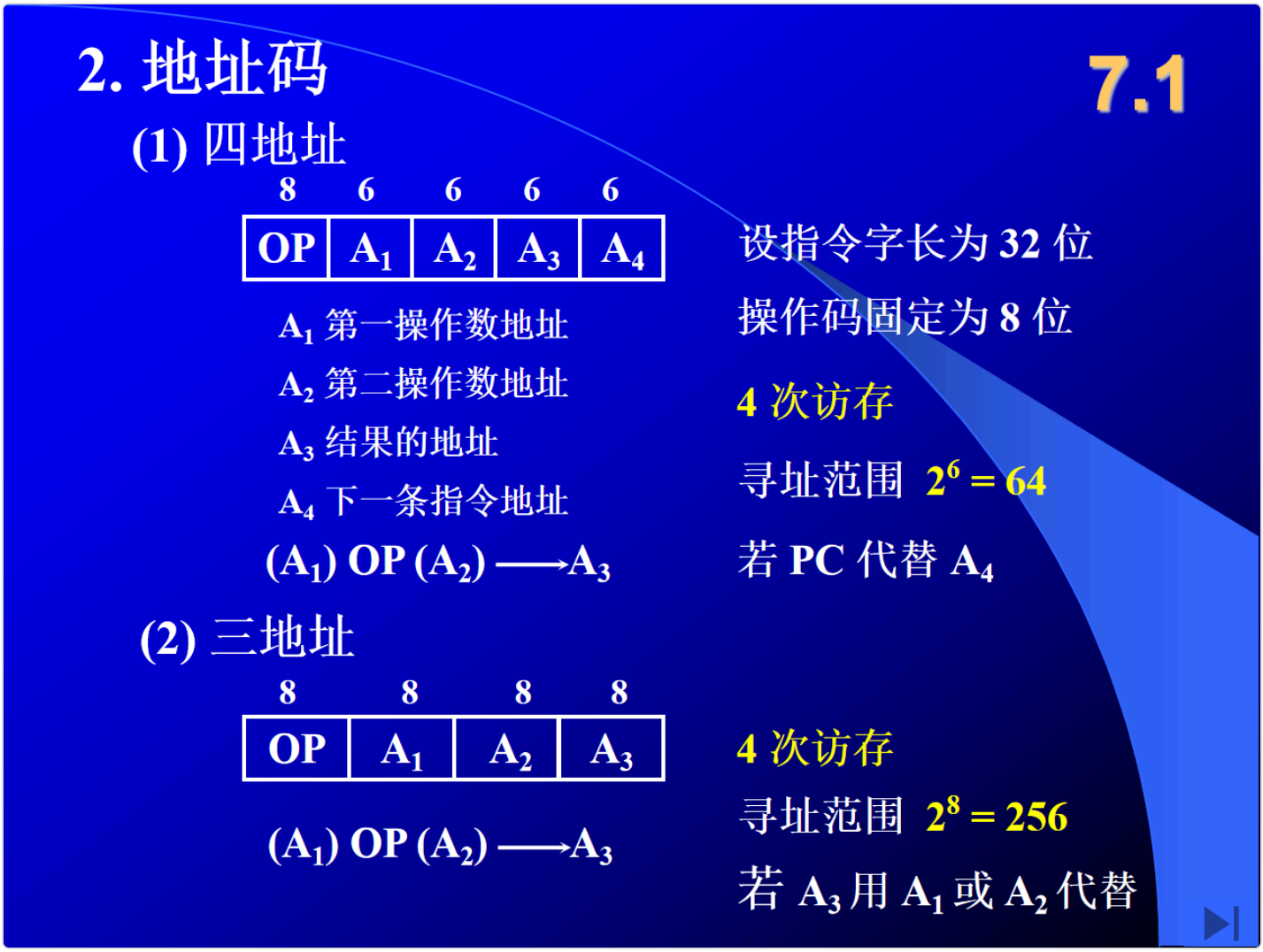
3. 设计题通用递推：设地址码每段 k 位，本级可用前缀数 = 上一级剩余编码数；本级指令数上限 = 前缀数 × 2^k（最后一级零地址不再留前缀时可全用）。
4. 扩展操作码是**等长扩展**（每级操作码长度整齐地增加 k 位），用频度高的指令给短操作码、频度低的给长操作码，缩短平均操作码长度。

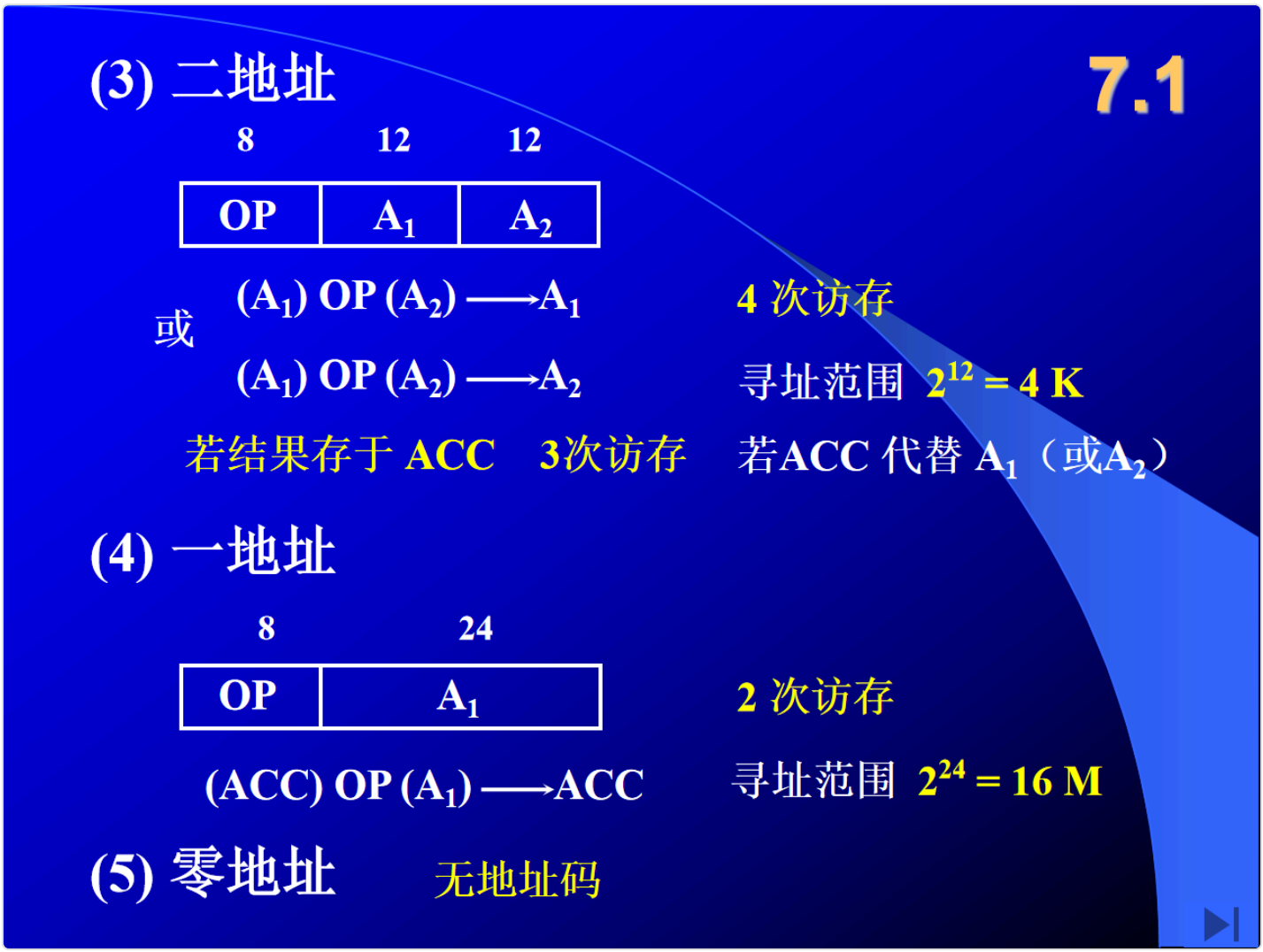
（补充常识）哈夫曼操作码：课件未展开。按指令使用频度构造哈夫曼树，频度高的指令操作码短、频度低的长，平均码长 $L = \Sigma\, p_i \times l_i$ 最小（信息最优）；缺点是操作码长度不规则、译码硬件复杂，故实际机器多用等长扩展作折中。计算方法见习题册补充题。

易错警示：① 留前缀时"全 1"只是课件方案，本质是"短码不能是长码的前缀"；② 计算零地址条数时最后一级不必再留前缀（课件 16 条全用）；③ 2^4 中的 4 是**一段地址码的位数**，不是操作码位数。

1.3 地址码：四地址 → 零地址（★★ 访存次数/寻址范围）

设指令字长 32 位、操作码固定 8 位（课件 p.375–376）：





地址数	格式（位数）	操作语义	寻址范围	访存次数（含取指 1 次）
四地址	OP 8 + A1 A2 A3 A4 各 6	(A1) OP (A2) → A3，A4 为下条指令地址	$2^6 = 64$	4
三地址	OP 8 + A1 A2 A3 各 8	(A1) OP (A2) → A3	$2^8 = 256$	4
二地址	OP 8 + A1 A2 各 12	(A1) OP (A2) → A1（或 A2）	$2^{12} = 4$ K	4；若结果存于 ACC 则 3
一地址	OP 8 + A1 24	(ACC) OP (A1) → ACC	$2^{24} = 16$ M	2
零地址	无地址码	操作数隐含（如堆栈栈顶、ACC）（补充常识）	—	视隐含位置而定

演化链条（课件 p.375–376）：

- 四地址中 **PC 代替 A4** → 三地址（下条指令地址由 PC 自动给出）；
- 三地址中 **A3 用 A1 或 A2 代替**（结果覆盖某个源操作数）→ 二地址；
- 二地址中 **ACC 代替 A1（或 A2）**、结果存 ACC → 一地址；
- 操作数全部隐含（堆栈/ACC）→ 零地址。

访存次数拆解（含取指）：四/三/二地址 = 取指 1 + 取两个操作数 2 + 写结果 1 = 4；二地址结果存 ACC = 取指 1 + 取数 2 = 3；一地址 = 取指 1 + 取数 1 = 2（结果写 ACC 不访存）。

易错警示：① 课件 p.375–376 的访存次数**包含取指**；而寻址方式一节（p.387 起）说的"执行阶段访存"**不含取指**，两处口径不同，审题先看问法；② 寻址范围由**地址码位数**决定（2^地址位数），与操作码无关；③ 零地址不是"没有操作数"，而是操作数隐含。

1.4 指令字长

指令字长决定于：操作码的长度、操作数地址的长度、操作数地址的个数（课件 p.377）。

- **指令字长固定**：指令字长 = 存储字长（课件 p.377）。
- **指令字长可变**：按字节的倍数变化（课件 p.377），如 8086 为 1～6 字节（p.408）。

易错警示：指令字长、存储字长、机器字长三者概念不同，课件仅在寻址讨论中约定三者相等（p.387），不要当成普遍结论。

1.5 小结：用硬件资源代替地址码字段

当用一些硬件资源（PC、ACC、寄存器）代替指令字中的地址码字段后（课件 p.378）：

- 可**扩大指令的寻址范围**（如 PC/基址寄存器位数多于地址字段）；
- 可**缩短指令字长**；
- 可**减少访存次数**。

当指令的地址字段为**寄存器**时：三地址 OP R1, R2, R3；二地址 OP R1, R2；一地址 OP R1。寄存器编号位数少 → 缩短指令字长；操作数在寄存器中 → **指令执行阶段不访存**（课件 p.378）。

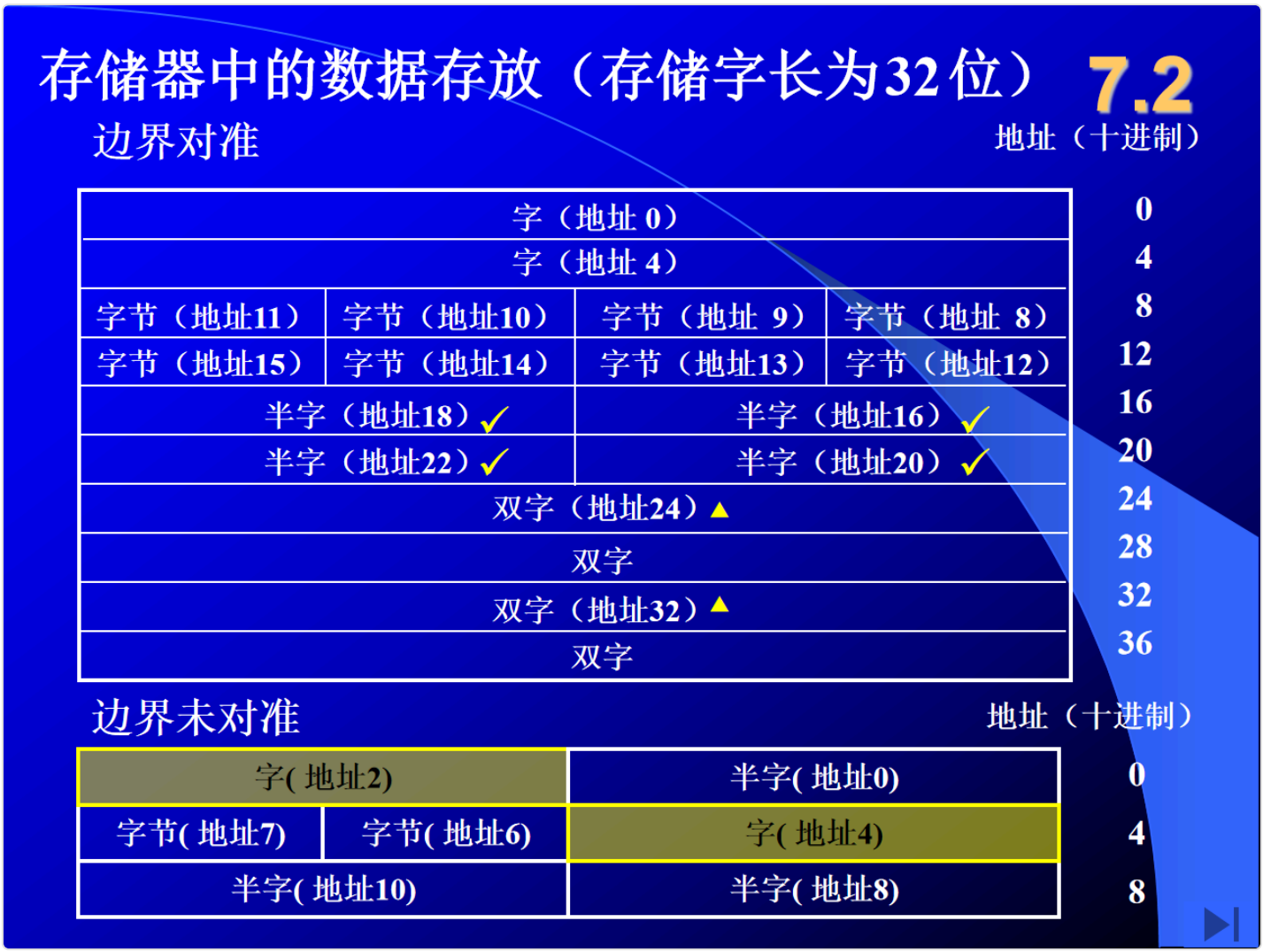
2. 操作数类型和操作类型

2.1 操作数类型（课件 p.379）

类型	内容
地址	操作数地址（无符号整数看待）
数字	无符号整数；定点数、浮点数、十进制数
字符	ASCII
逻辑数	参与逻辑运算的位串

2.2 数据在存储器中的存放方式与边界对准

- **字地址为低字节地址**：低字节放低地址（小端，补充常识命名，x86 属此类）；**字地址为高字节地址**：高字节放低地址（大端）（课件 p.379）。
- **边界对准**（存储字长 32 位、按字节编址，课件 p.380）：



数据宽度	对准要求（地址为其倍数）	课件对准示例地址
字节 1 B	任意	8, 9, 10, 11, 12, 13, 14, 15
半字 2 B	2	16, 18, 20, 22
字 4 B	4	0, 4
双字 8 B	8	24, 32

边界未对准的后果：如从地址 2 开始存放 4 B 的字（字节 2–5），跨越两个存储字（0–3 与 4–7），读出一个字需**两次访存**；对准存放则一次访存即可（课件 p.380 下图示意）。

易错警示：对准的"倍数"针对**数据自身宽度**（半字 2、字 4、双字 8），不是存储字长一律 4；课件 p.380 未对准示意图中"字(地址2)"与"字(地址4)"的字节区间有重叠，图为示意，按对准原理理解即可（课件原文如此，存疑：示意图标注区间重叠）。

2.3 操作类型（课件 p.381–384）

1. **数据传送**：寄存器↔寄存器、寄存器↔存储器、存储器↔存储器；置"1"、清"0"。课件举例助记符：MOVE、STORE、LOAD、PUSH、POP（课件 p.381）。
2. **算术逻辑操作**：加、减、乘、除、增 1、减 1、求补、浮点运算、十进制运算；与、或、非、异或、位操作、位测试、位清除、位求反。8086 例：ADD SUB MUL DIV INC DEC CMP NEG；十进制调整 AAA AAS AAM AAD；AND OR NOT XOR TEST（课件 p.381）。
3. **移位操作**：算术移位、逻辑移位、循环移位（带进位和不带进位）（课件 p.382）。
4. **转移**： - 无条件转移 JMP； - 条件转移：结果为零转 JZ（Z=1）、结果溢出转 JO（O=1）、结果有进位转 JC（C=1）； - 跳过一条指令 SKP：如地址 305 处 SKP DZ，D=0 则跳过 306 直接执行 307（D 为完成触发器）（课件 p.382）。
5. **调用和返回**：CALL SUB1 / CALL SUB2 / RETURN，支持嵌套调用。课件 p.383 示例：主程序 2100 处 CALL SUB1（返回地址 2101）；SUB1 中 2500 处 CALL SUB2（返回地址 2501）；SUB2 的 RETURN（2700）先回 2501，SUB1 的 RETURN 再回 2101。（补充常识：返回地址靠堆栈保存才能嵌套。）
6. **陷阱（Trap）与陷阱指令**：意外事故的中断由 CPU 自动产生并执行（**隐指令**），一般不提供给用户直接使用；另设置供用户使用的陷阱指令完成**系统调用**，如 8086 的 INT TYPE（软中断）（课件 p.384）。
7. **输入输出**：入 = 端口地址 → CPU 寄存器；出 = CPU 寄存器 → 端口地址。8086 例：IN AL, n；IN AX, n；IN AL, DX；IN AX, DX；OUT n, AL；OUT n, AX；OUT DX, AL；OUT DX, AX（课件 p.384）。

易错警示：① 隐指令（事故中断自动执行）≠ 用户陷阱指令（INT TYPE，主动系统调用）；② 条件转移依据的是标志位（Z/O/C），不是操作数本身；③ SKP 是"跳过下一条"，不是转移到任意地址。

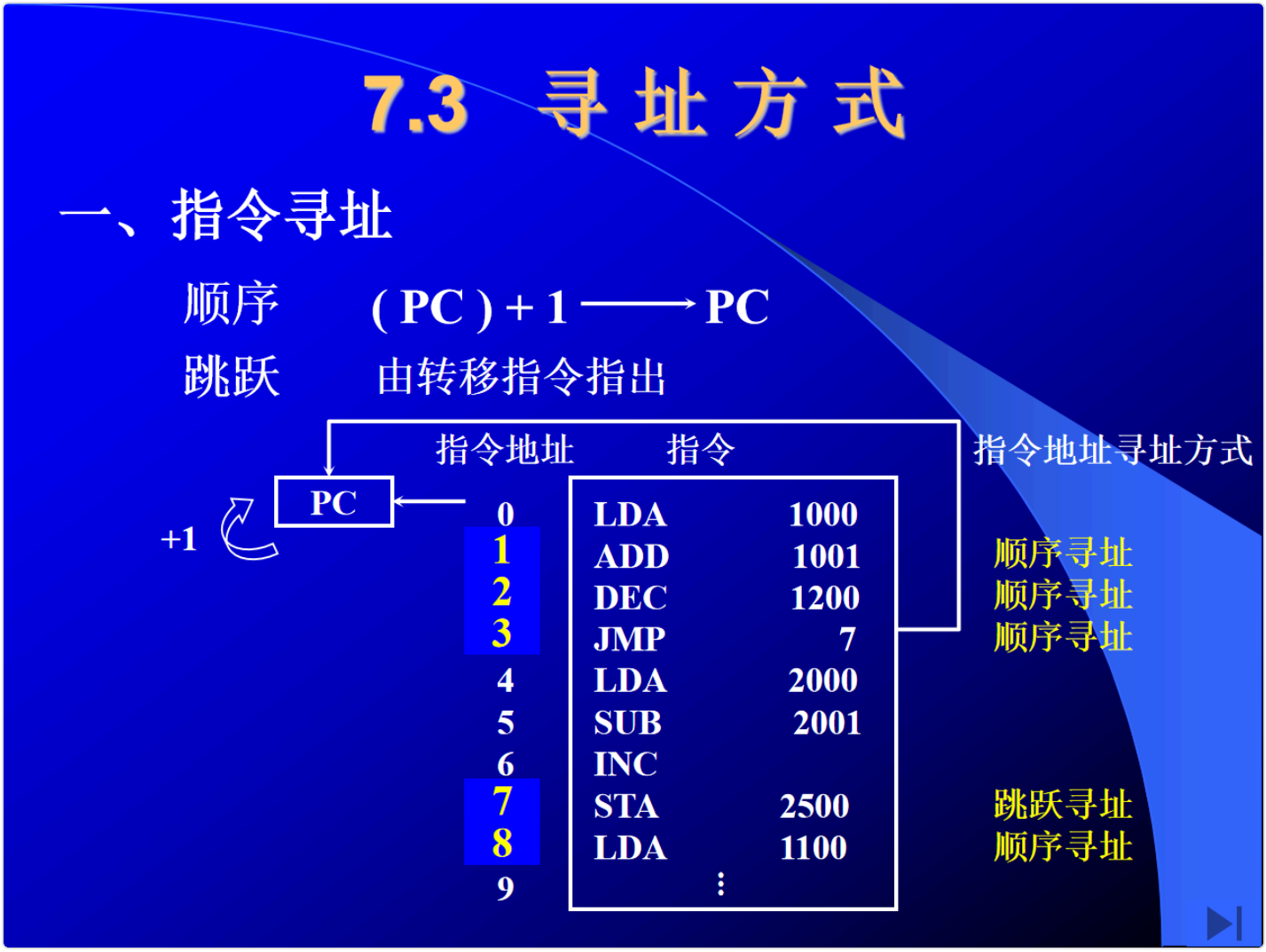
3. 寻址方式（★★★ 全章核心）

3.1 寻址方式概述与指令寻址

寻址方式 = 确定**本条指令的操作数地址** + **下一条欲执行指令的指令地址**；分为指令寻址与数据寻址（课件 p.385）。

指令寻址（课件 p.386）：

- 顺序：(PC) + 1 → PC（按字编址、每指令 1 字时；按字节编址时加指令字节数，见 p.400 例）；
- 跳跃：由转移指令指出。



课件示例执行流：0 LDA 1000 → 1 ADD 1001 → 2 DEC 1200 → 3 JMP 7 → 7 STA 2500 → 8 LDA 1100 → 9 …；其中 0→1→2→3 为顺序寻址，3→7 为跳跃寻址，7→8→9 又为顺序寻址（课件 p.386）。

3.2 形式地址与有效地址

- **形式地址 A**：指令字中的地址；
- **有效地址 EA**：操作数的真实地址；
- 约定：指令字长 = 存储字长 = 机器字长（课件 p.387）。

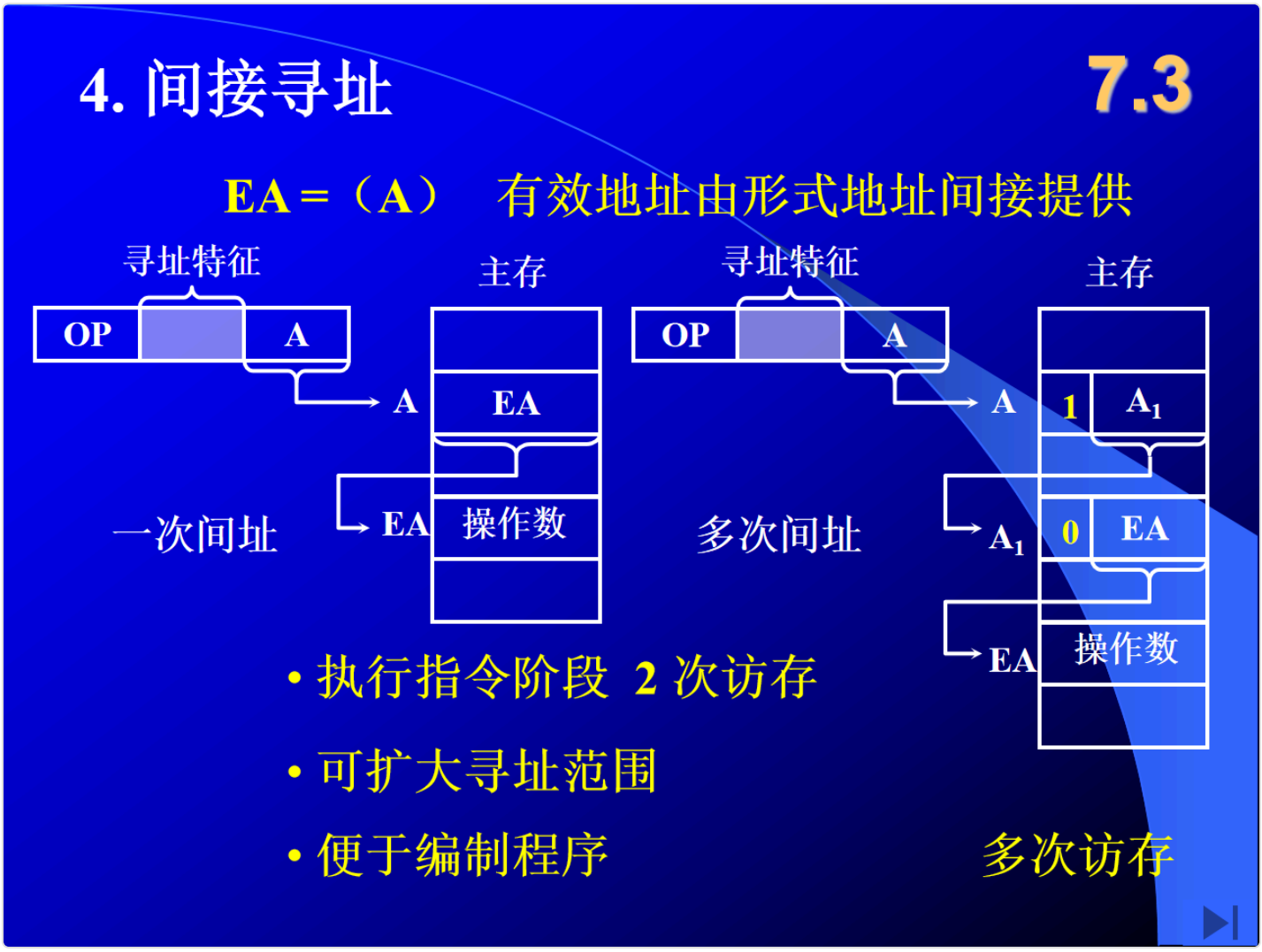
3.3 十种数据寻址方式

1) 立即寻址（课件 p.387）：形式地址 A 就是操作数本身（立即数，可正可负、补码表示），指令格式 OP # A。**执行阶段不访存**；A 的位数限制了立即数的范围。

2) 直接寻址（课件 p.388）：EA = A。执行阶段访问一次存储器；A 的位数决定操作数寻址范围；操作数地址不易修改（必须修改 A）。

3) 隐含寻址（课件 p.389）：操作数地址隐含在操作码中。如 ADD A 的另一个操作数隐含在 ACC 中；8086 MUL 的被乘数隐含在 AX（16 位）或 AL（8 位）；MOVS 源地址隐含在 SI、目的地址隐含在 DI。少一个地址字段 → 缩短指令字长。

4) 间接寻址（课件 p.390–391）：EA = (A)，有效地址由形式地址间接提供。

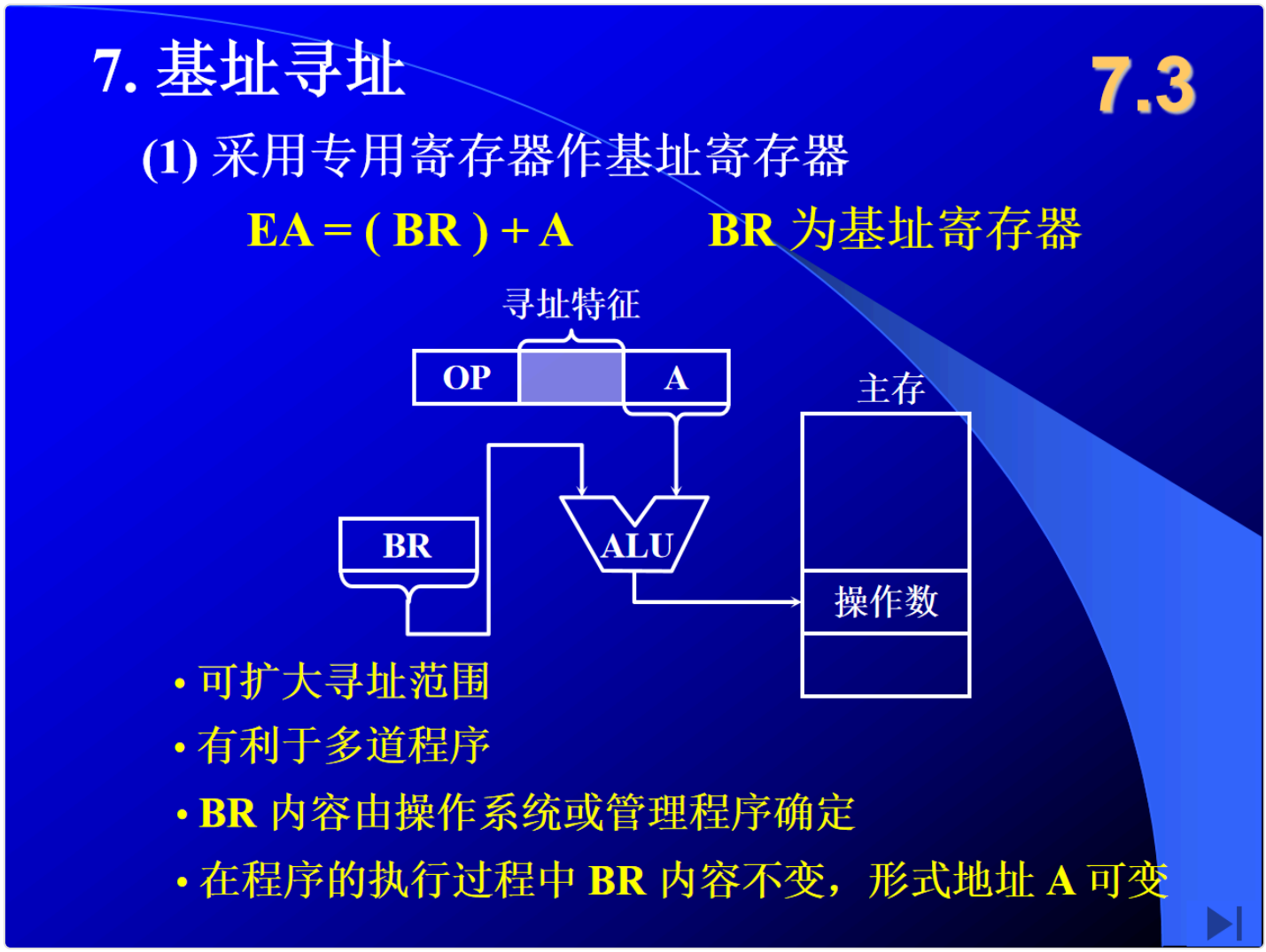


- 一次间址：执行阶段 **2 次访存**（取 EA 一次 + 取操作数一次）；
- 多次间址：地址字最高位作间址标志（1 继续间址、0 给出 EA），多次访存；
- 优点：可扩大寻址范围（存储字长 > 地址码位数）、便于编制程序。
- 编程举例（课件 p.391）：子程序返回用 JMP @ A（@ 为间址特征）。主程序在 80、201 两处调用，返回地址 81、202 分别写入单元 A，即 (A)=81 或 (A)=202；子程序末尾 JMP @ A 即返回各自调用点，**无需修改指令本身**。

5) 寄存器寻址（课件 p.392）：EA = Ri，有效地址即寄存器编号，操作数在寄存器中。执行阶段不访存、只访问寄存器，速度快；寄存器个数有限、编号位数少 → 缩短指令字长。

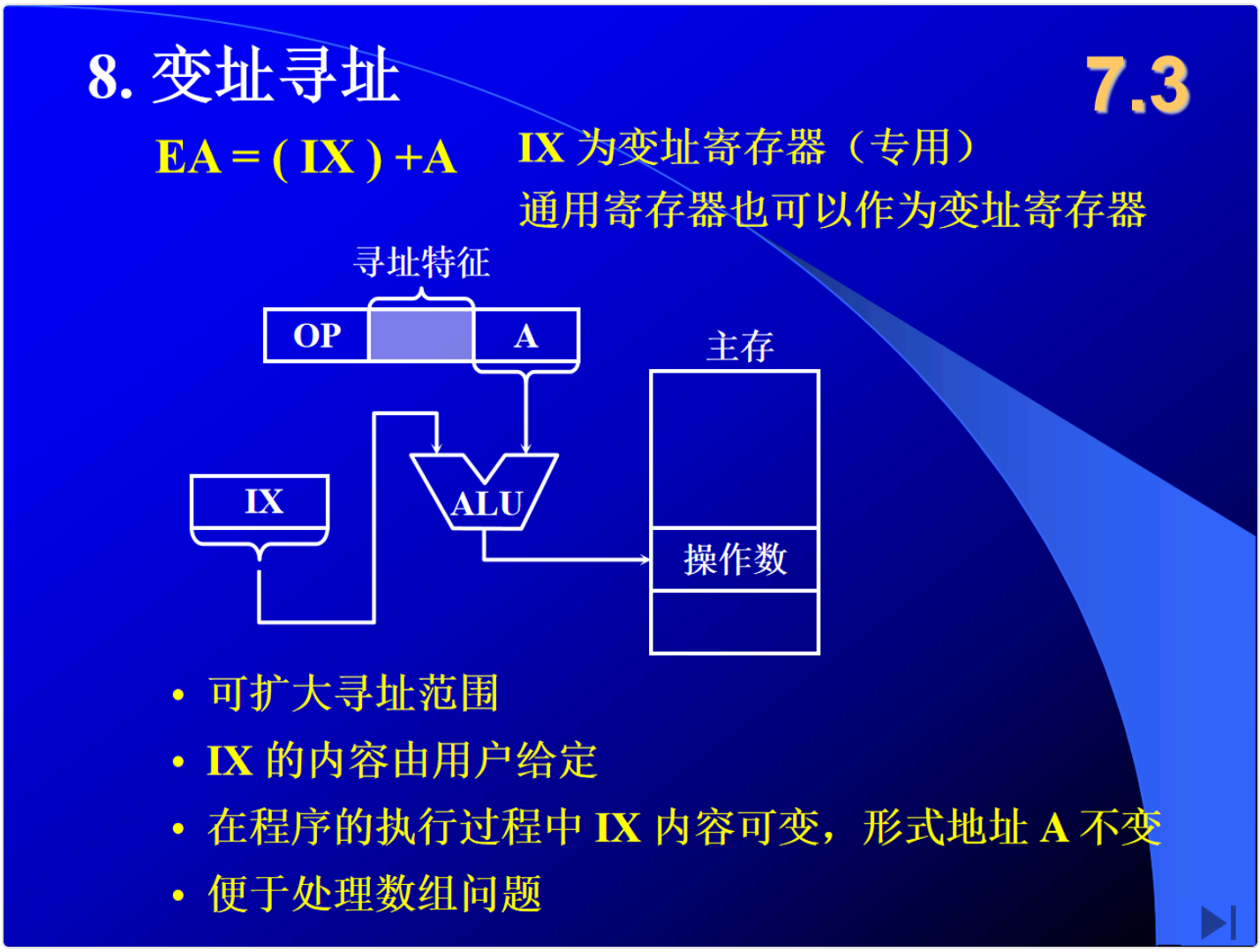
6) 寄存器间接寻址（课件 p.393）：EA = (Ri)。有效地址在寄存器中、操作数在存储器中，执行阶段访存 1 次；便于编制循环程序（改寄存器即改地址）。

7) 基址寻址（课件 p.394–395）：EA = (BR) + A。



- 可采用专用基址寄存器 BR，也可由用户指定某个通用寄存器作基址寄存器（p.395）；
- 基址寄存器内容由**操作系统或管理程序**确定；
- 程序执行过程中 **BR 内容不变、形式地址 A 可变**；
- 可扩大寻址范围、有利于多道程序（程序重定位）。

8) 变址寻址（课件 p.396–397）：EA = (IX) + A。



- IX 为变址寄存器（专用），通用寄存器也可作变址寄存器；
- IX 内容由**用户**给定；程序执行过程中 **IX 内容可变、形式地址 A 不变**；
- 可扩大寻址范围、便于处理数组问题。
- 例（课件 p.397）：数据块首地址 D，求 N 个数平均值。直接寻址需 LDA D、ADD D+1 … ADD D+(N-1)、DIV #N、STA ANS 共 **N+2 条**；变址寻址只需 8 条：LDA #0；LDX #0；M: ADD X, D；INX；CPX #N；BNE M；DIV #N；STA ANS（X 为变址寄存器，D 为形式地址）。

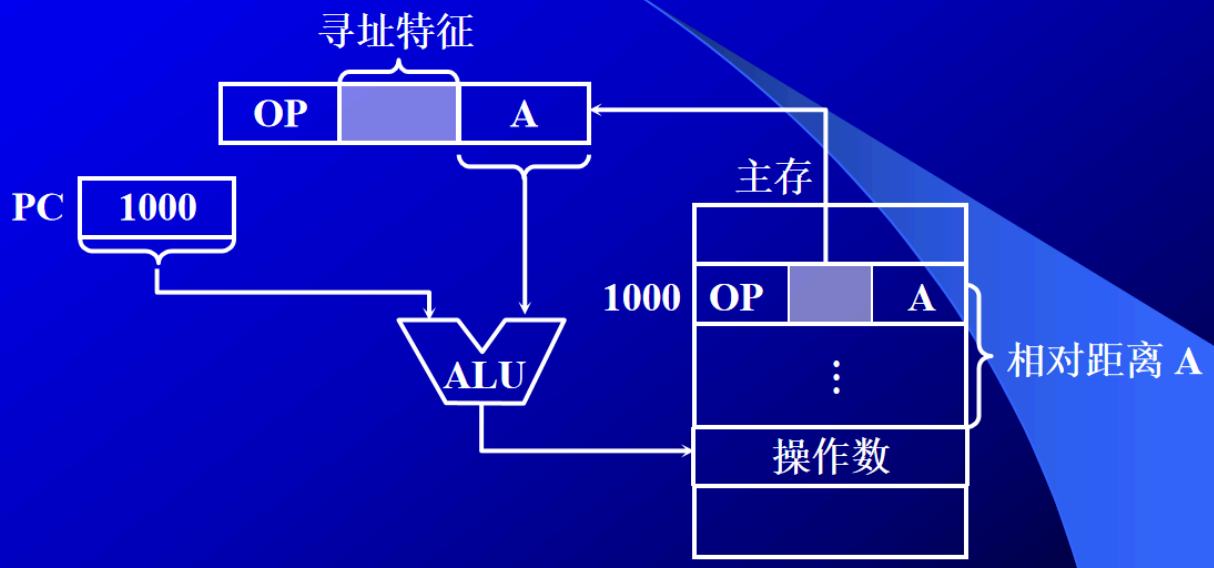
9) 相对寻址（课件 p.398–400）：EA = (PC) + A。

9. 相对寻址

7.3

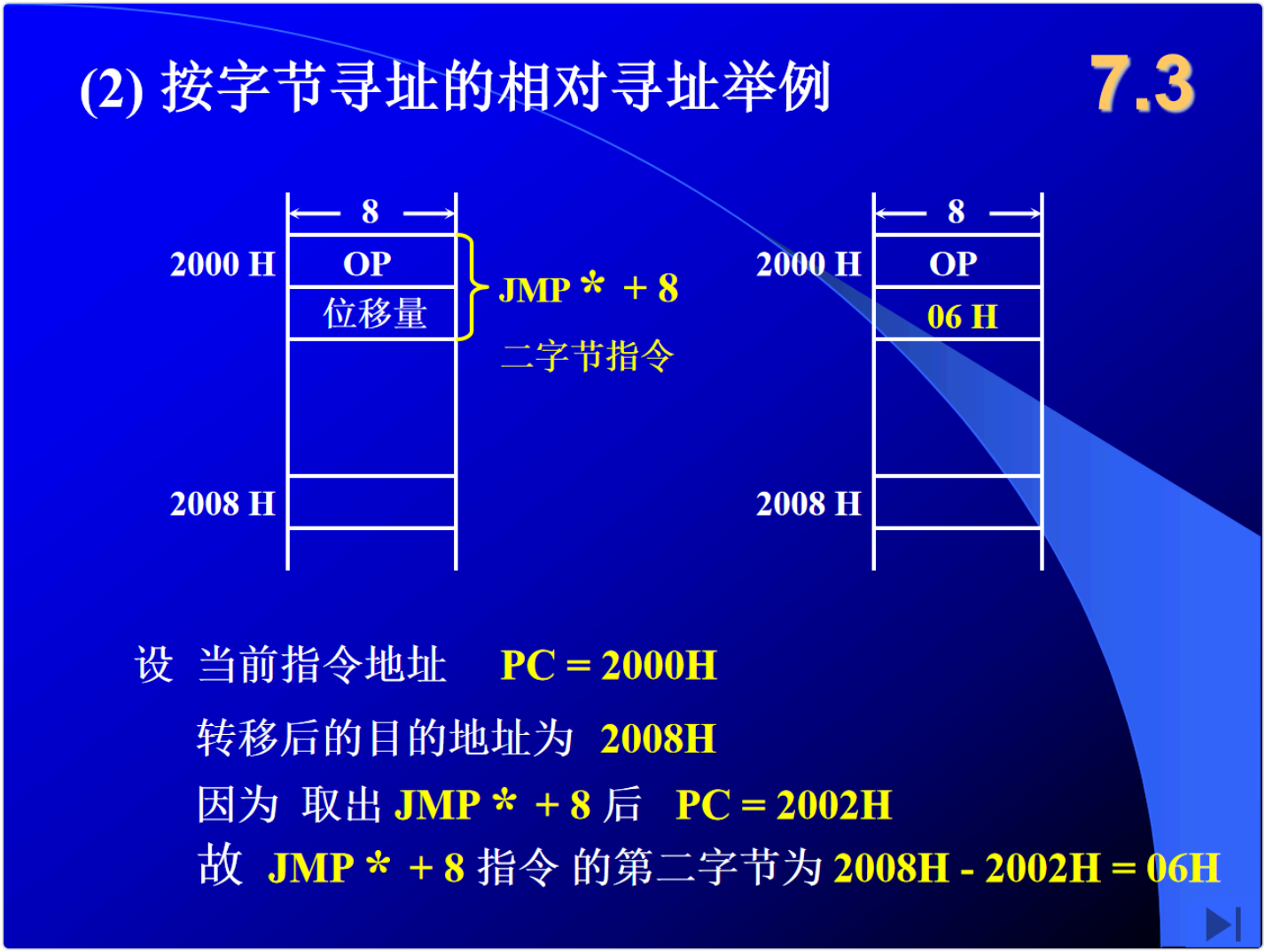
EA = (PC) + A

A 是相对于当前指令的位移量（可正可负，补码）



- **A 的位数决定操作数的寻址范围**
- **程序浮动**
- **广泛用于转移指令**

- A 是相对于当前指令的位移量（可正可负、补码）；A 的位数决定操作数寻址范围；
- 便于**程序浮动**；广泛用于转移指令。
- 例 1（课件 p.399）：循环体 M: ADD X, D；M+1 INX；M+2 CPX #N；M+3 BNE M。课件取位移量 −3，EA = (M+3) − 3 = M；程序整体浮动时 M 变化，但 BNE 与 ADD 的相对位移不变。（课件原文如此，存疑：此处把 PC 当作 BNE 自身地址 M+3；若按 p.400"取指后 PC 自增"的约定，取指后 PC = M+4，位移量应为 −4。两页约定不一致，做题以题目给定的 PC 口径为准。）
- 例 2（课件 p.400，按字节寻址）：



当前指令地址 PC = 2000H，JMP *+8 为二字节指令，取出后 PC = 2002H；转移目的地址 2008H，故指令第二字节（位移量）= 2008H − 2002H = **06H**。**位移量 = 目标地址 − 取指后的 PC**，这是考试标准口径。

10) 堆栈寻址（课件 p.401–403）：

10. 堆栈寻址

7.3

(1) 堆栈的特点

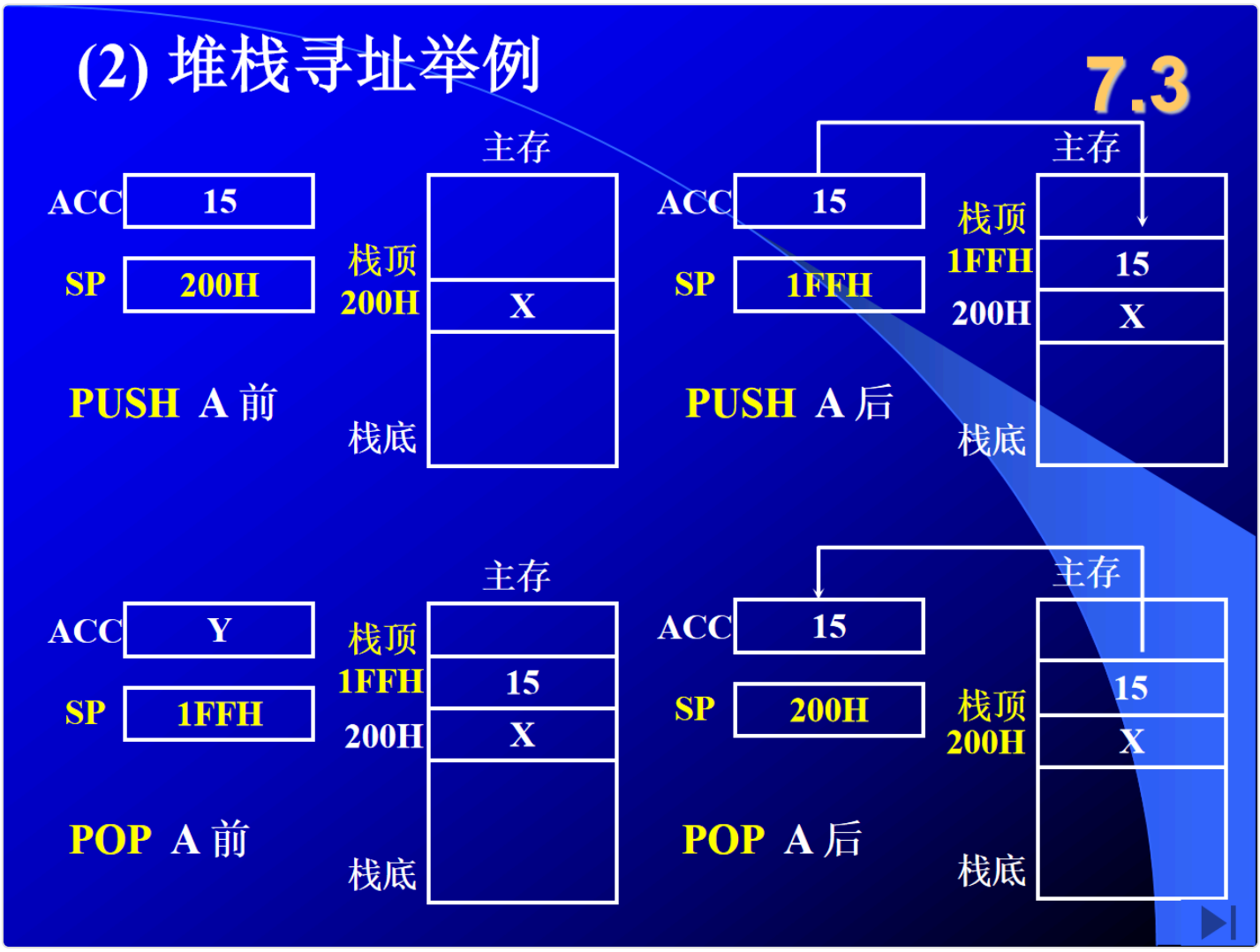
堆栈 { 硬堆栈 多个寄存器
软堆栈 指定的存储空间

先进后出（一个入出口） 栈顶地址 由 SP 指出

进栈 （SP）– 1 → SP 出栈 （SP）+ 1 → SP



- 硬堆栈 = 多个寄存器；软堆栈 = 指定的存储空间；**先进后出**（一个入出口）；栈顶地址由 **SP** 指出；
- 课件示例栈向低地址生长：进栈 (SP) − 1 → SP，出栈 (SP) + 1 → SP（栈底 2000H，进栈后 SP = 1FFFH）；
- 举例（课件 p.402）：



PUSH A 前：ACC=15、SP=200H、(200H)=X；PUSH A 后：(1FFH)=15、SP=1FFH。POP A 前：ACC=Y、SP=1FFH、(1FFH)=15；POP A 后：ACC=15、SP=200H（1FFH 中数据仍保留，只是不再属栈顶）。即**进栈先改 SP 再写入，出栈先读出再改 SP**（满递减栈，补充常识命名）。 - **SP 的修改与主存编址方法有关**（课件 p.403）：

编址方式	进栈	出栈
按字编址	(SP) − 1 → SP	(SP) + 1 → SP
按字节编址、存储字长 16 位	(SP) − 2 → SP	(SP) + 2 → SP
按字节编址、存储字长 32 位	(SP) − 4 → SP	(SP) + 4 → SP

3.4 十种寻址方式速比表

寻址方式	有效地址 EA	执行阶段访存次数	主要特点/用途
立即	无（操作数 = A）	0	快；立即数范围受 A 位数限制
直接	A	1	简单；地址不易修改
隐含	隐含于操作码	视指令（如 ADD A 为 1）	缩短指令字长
间接（一次）	(A)	2	扩大寻址范围、便于编程
寄存器	Ri	0	快、指令短
寄存器间接	(Ri)	1	便于循环
基址	(BR) + A	1	OS 设基址、执行中不变；多道程序/重定位
变址	(IX) + A	1	用户设变址、执行中可变；数组
相对	(PC) + A	1	位移量补码；程序浮动、转移指令
堆栈	(SP)	1（软堆栈每次进/出栈）	先进后出；保存返回地址/现场

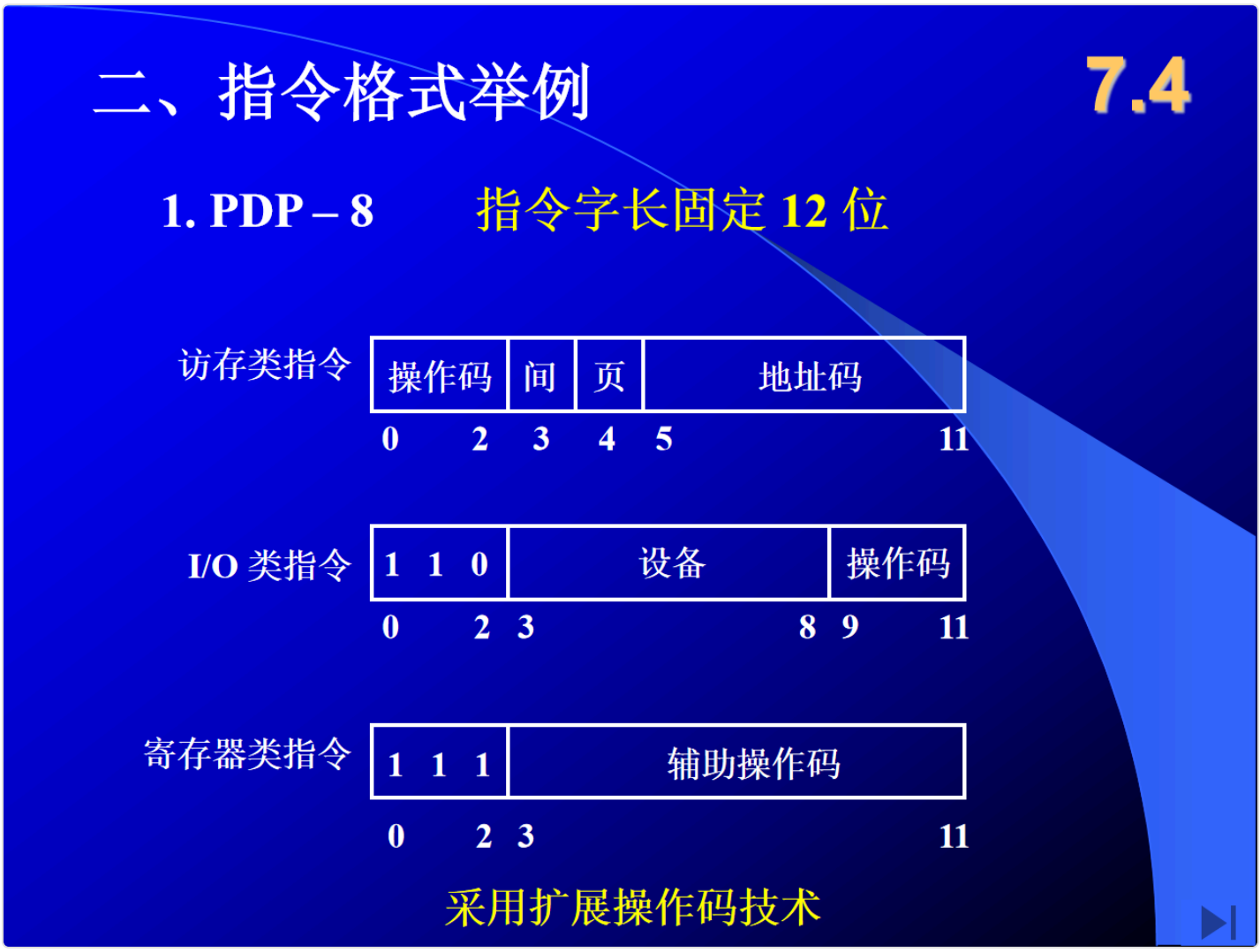
易错警示：① 基址与变址公式形式相同，区别在"谁的内容可变、谁设定"：基址 BR 由 OS 定、执行中不变、A 可变；变址 IX 由用户定、执行中可变、A 不变；② 间接 EA=(A) 与寄存器间接 EA=(Ri) 只差"地址的存放处"，访存次数差 1；③ 相对寻址的 PC 一律用**取指后**的值（p.400 口径），p.399 课件口径存疑；④ 立即寻址执行阶段不访存，但取指仍访存 1 次。

4. 指令格式举例

4.1 设计指令格式时应考虑的因素（课件 p.404）

1. **指令系统的兼容性**（向上兼容：低档机程序能在高档机上运行，补充常识释义）；
2. 其他因素：操作类型（指令个数及操作难易程度）、数据类型（哪些数据类型可参与操作）、指令格式（指令字长是否固定）、寻址方式（指令寻址、操作数寻址）、寄存器个数（直接影响指令执行时间）、地址码位数/地址个数/寻址方式类型、操作码位数/是否采用扩展操作码技术。

4.2 PDP-8：指令字长固定 12 位（课件 p.405）



类别	位 0–2	位 3	位 4	位 5–11
访存类	操作码（3 位）	间址位	页位	地址码（7 位）
I/O 类	1 1 0	设备（位 3–8）	同左	操作码（位 9–11）
寄存器类	1 1 1	辅助操作码（位 3–11）	同左	同左

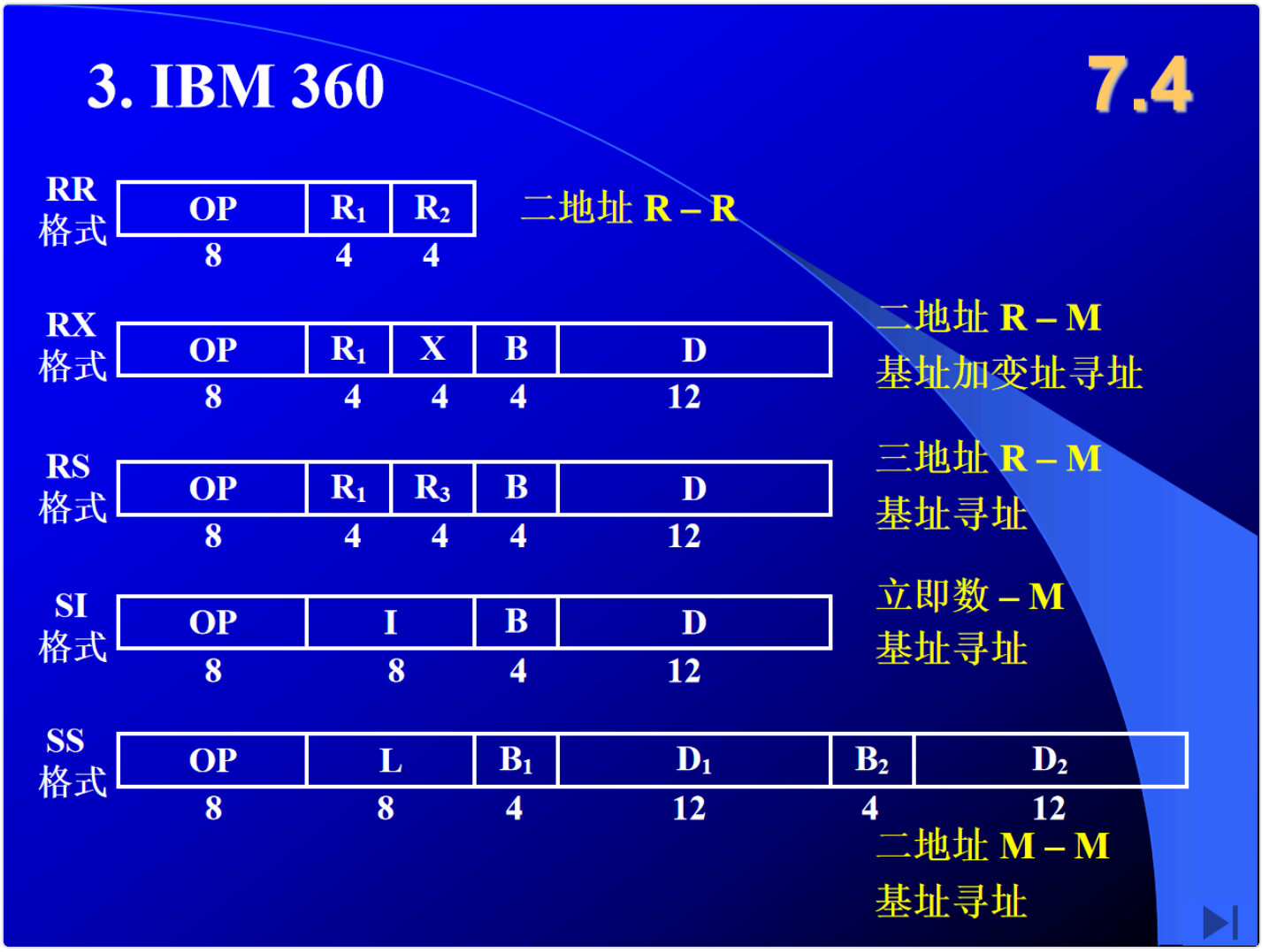
采用扩展操作码技术：用操作码 110、111 作前缀扩展出 I/O 类与寄存器类（课件 p.405）。

4.3 PDP-11：指令字长 16 / 32 / 48 位三种（课件 p.406）

格式	字长	字段划分
零地址	16	OP-CODE 16
一地址	16	OP 10 + 目的地址 6
二地址 R–R	16	OP 4 + 源地址 6 + 目的地址 6
二地址 R–M	32	OP 10 + 目的地址 6 + 存储器地址 16
二地址 M–M	48	OP 10 + 目的地址 6 + 存储器地址1 16 + 存储器地址2 16

扩展操作码技术：操作码 4 位与 10 位两级（课件 p.406）。

4.4 IBM 360：五种格式（课件 p.407）



格式	字段（位数）	字长	地址数与寻址
RR	OP 8 + R1 4 + R2 4	16	二地址 R–R
RX	OP 8 + R1 4 + X 4 + B 4 + D 12	32	二地址 R–M，基址加变址寻址
RS	OP 8 + R1 4 + R3 4 + B 4 + D 12	32	三地址 R–M，基址寻址
SI	OP 8 + I 8 + B 4 + D 12	32	立即数–M，基址寻址
SS	OP 8 + L 8 + B1 4 + D1 12 + B2 4 + D2 12	48	二地址 M–M，基址寻址

4.5 Intel 8086：指令字长 1～6 字节（课件 p.408）

地址数	例子	字节数
零地址	NOP	1
一地址	INC AX	1
一地址	CALL 段内调用 / 段间调用	3 / 5
二地址 R–R	ADD AX, BX	2
二地址 R–立即数	ADD AX, 3048H	3
二地址 R–M	ADD AX, [3048H]	4
M–立即数	MOV WORD PTR [0204], 0138H	6

易错警示：① PDP-11 的 R–R 用 4 位 OP、含存储器地址时用 10 位 OP，是扩展操作码的又一实例；② 8086 中 ADD AX, 3048H 是立即数（3 字节），ADD AX, [3048H] 才是直接寻址（4 字节），方括号决定寻址方式；③ IBM 360 的 RX 才是"基址+变址"，RS/SI/SS 只有基址。

5. RISC 技术

5.1 RISC 的产生：80—20 规律（课件 p.409）

典型程序中 **80% 的语句仅仅使用处理机中 20% 的指令**；执行频度高的简单指令因复杂指令的存在执行速度无法提高。于是提出：能否用 20% 的简单指令组合不常用的 80% 的指令功能？→ RISC（Reduced Instruction Set Computer）；对照 CISC（Complex Instruction Set Computer）。

5.2 RISC 的主要特征（课件 p.410）

1. 选用使用频度较高的一些简单指令，复杂指令的功能由简单指令来组合；
2. 指令长度固定、指令格式种类少、寻址方式少；
3. 只有 LOAD / STORE 指令访存；

4. 采用流水技术，一个时钟周期内完成一条指令；
5. 采用组合逻辑实现控制器；
6. CPU 中有多个通用寄存器；
7. 采用优化的编译程序。

5.3 CISC 的主要特征（课件 p.411）

1. 系统指令复杂庞大，各种指令使用频度相差大；
2. 指令长度不固定、指令格式种类多、寻址方式多；
3. 访存指令不受限制；
4. 大多数指令需要多个时钟周期执行完毕；
5. 采用微程序控制器；
6. CPU 中设有专用寄存器；
7. 难以用优化编译生成高效的目的代码。

5.4 RISC 和 CISC 的比较（课件 p.412）

比较项	RISC	CISC
指令系统	简单、频度集中	复杂庞大、频度相差大
指令字长/格式/寻址	固定、少、少	不固定、多、多
访存	仅 LOAD/STORE	不受限制
执行周期	流水、单周期一条	多周期
控制器	组合逻辑	微程序
寄存器	多个通用寄存器	专用寄存器
编译优化	有利于优化	难以优化
VLSI 面积利用	更能充分利用	较差
设计成本/可靠性	便于设计、成本低、可靠性高	相反
兼容性	**不易实现指令系统兼容**（缺点）	易兼容

易错警示：① "只有 LOAD/STORE 访存"≠"不访存"，运算指令的操作数必须先在寄存器中；② RISC 的缺点是**兼容性差**，选择题常把"有利于兼容"当作 RISC 优点来设错；③ 组合逻辑控制器 ↔ RISC、微程序控制器 ↔ CISC 是固定搭配。

6. 易错点与高频陷阱汇总

1. 扩展操作码必须给长码留前缀（课件用全 1），短码不能是长码的前缀；最后一级（零地址）可不留前缀。

2. "每减少一种三地址操作码多 16 种二地址"中的 16 = 2^4，指数是**一段地址码位数**。
3. p.375–376 访存次数含取指；p.387 起"执行阶段访存"不含取指，两种口径别混用。
4. 一地址指令结果存 ACC，执行阶段只访存 1 次（加取指共 2 次）；二地址结果存 ACC 共 3 次。
5. 零地址指令仍有操作数（隐含于堆栈/ACC），不是"无操作数指令"。
6. 立即寻址的操作数在指令中，执行阶段不访存；但 A 的位数限制立即数范围。
7. 间接寻址 EA=(A) 访存 2 次（执行阶段）；寄存器间接 EA=(Ri) 访存 1 次；寄存器寻址 EA=Ri 访存 0 次。
8. 基址：BR 由 OS 设定、执行中不变、A 可变，面向多道程序/重定位；变址：IX 由用户设定、执行中可变、A 不变，面向数组/循环。两者公式同形、语义相反。
9. 相对寻址位移量 = 目标地址 − **取指后** PC（p.400：2008H−2002H=06H）；p.399 课件用当前指令地址算 −3，口径存疑，考试按题目约定。
10. 堆栈进栈先改 SP 再写、出栈先读再改 SP（课件满递减示例）；SP 修改量取决于编址方式：按字 ±1，按字节且字长 16/32 位 ±2/±4。
11. 边界对准看数据自身宽度：半字地址 2 的倍数、字 4 的倍数、双字 8 的倍数；未对准一个字要两次访存。
12. 指令字长 = 存储字长只是"固定字长"情形与寻址讨论的约定，可变字长按字节倍数变化（8086 1～6 字节）。
13. RISC 仅 LOAD/STORE 访存、组合逻辑控制器、通用寄存器多、单周期流水；其缺点是兼容性差，不是优点。
14. 8086 中 ADD AX, 3048H（立即数，3 字节）与 ADD AX, [3048H]（直接寻址，4 字节）只差方括号。
15. 哈夫曼操作码平均码长最短但译码复杂，课件只讲等长扩展；哈夫曼计算属补充常识。

7. 典型例题精讲

例 1（扩展操作码设计） 指令字长 16 位，每段地址码 4 位。已定义三地址指令 12 条、二地址指令 62 条、一地址指令 30 条，问零地址指令最多多少条?

解：① 4 位操作码共 2^4 = 16 个编码，三地址用 12 个，剩 16 − 12 = 4 个作前缀；② 二地址空间 = 4 × 2^4 = 64，用 62 个，剩 64 − 62 = 2 个前缀；③ 一地址空间 = 2 × 2^4 = 32，用 30 个，剩 32 − 30 = 2 个前缀；④ 零地址空间 = 2 × 2^4 = **32 条**（最后一级不留前缀）。答：最多 32 条。

例 2（有效地址综合计算） 按字编址，当前指令地址 100，取指后 PC = 101，形式地址 A = 200；(200)=300，(300)=400，(1200)=600，(250)=700，(301)=800；(R1)=300，(BR)=1000，(IX)=50。求各寻址方式的 EA 与操作数。

解：立即：操作数 = 200，无 EA；直接：EA = 200，操作数 = (200) = 300；间接：EA = (200) = 300，操作数 = (300) = 400；寄存器：EA = R1，操作数 = (R1) = 300；寄存器间接：EA = (R1) = 300，操作数 = 400；基址：EA = 1000 + 200 = 1200，操作数 = 600；变址：EA = 50 + 200 = 250，操作数 = 700；相对：EA = 101 + 200 = 301，操作数 = 800。要点：先写 EA 公式再代入，"括号"表示取内容。

例 3（相对寻址位移量） 按字节编址，JMP *+8 存于 2000H，二字节指令，目标 2008H，求位移量字节。

解：取指后 PC = 2000H + 2 = 2002H；位移量 = 2008H − 2002H = 06H。若误用当前指令地址 2000H 会得 08H，错。推广：位移量 = 目标 − (本指令地址 + 本指令字节数)；程序浮动时该差值不变，故相对寻址支持程序浮

动（课件 p.399–400）。

附：本章速查表

EA 公式：立即 操作数=A；直接 EA=A；间接 EA=(A)；寄存器 EA=Ri；寄存器间接 EA=(Ri)；基址 EA=(BR)+A；变址 EA=(IX)+A；相对 EA=(PC)+A；堆栈 EA=(SP)。

执行阶段访存：立即 0、寄存器 0、直接 1、寄存器间接 1、基址 1、变址 1、相对 1、一次间接 2、软堆栈进/出栈各 1。

扩展操作码（16 位指令、4 位地址段）：15 条三地址（0000–1110）→ 15 条二地址（1111 0000–1111 1110）→ 15 条一地址（1111 1111 0000–1111 1111 1110）→ 16 条零地址；每让出 1 个上级编码 = 下级多 $2^4 = 16$ 个。

32 位指令、OP 8 位：四地址 6 位/64/4 次；三地址 8 位/256/4 次；二地址 12 位/4K/4 次（结果入 ACC 3 次）；一地址 24 位/16M/2 次。

SP 修改：按字 ±1；按字节字长 16 位 ±2、32 位 ±4；课件栈向低地址生长（进栈减、出栈加）。

格式位宽：PDP-8 固定 12 位（OP3+间1+页1+地址7）；PDP-11 16/32/48 位；IBM 360 RR16、RX32、RS32、SI32、SS48；8086 1～6 字节。

RISC 七特征口诀：简指令、定长少格式少寻址、仅 LOAD/STORE 访存、流水单周期、组合逻辑、多通用寄存器、优化编译；缺点：兼容性差。