第4章 指令系统

> 课件来源：《第四章 指令系统.ppt》（173 页，extracted/ch4，本章唯一版本）。本章是全书篇幅最大、考试分值最重的一章：寻址方式计算、指令正误判断、执行结果分析、程序阅读与编写几乎全部出自本章，必须逐条指令过关。

0. 本章考点清单

考点	常见题型	重要度
寻址方式辨析、有效地址 EA 与物理地址 PA 计算	填空、计算、综合	★★★
指令正误判断（MOV/ADD/XCHG/INC 等操作数规则）	选择、判断	★★★
指令执行后寄存器/存储单元/标志位结果	填空、计算	★★★
移位与循环移位计算（SHL/SAL/SHR/SAR/ROL/ROR/RCL/RCR）	计算	★★★
程序阅读与编写（求最大数、求和、BCD→ASCII、串处理）	程序、综合	★★★
串操作指令与重复前缀（MOVS/CMPS/SCAS/LODS/STOS + REP/REPE/REPNE）	填空、程序	★★★
MOV、堆栈 PUSH/POP、XCHG、XLAT、LEA/LDS/LES 的使用规则	选择、判断	★★
ADD/ADC/SUB/SBB/CMP/INC/DEC/NEG 与标志位	计算	★★
MUL/IMUL/DIV/IDIV/CBW/CWD 的隐含操作数与结果	计算	★★
BCD 码调整指令 AAA/DAA/AAS/DAS/AAM/AAD	计算	★★
逻辑指令的位操作技巧（屏蔽、置位、取反、拼字）	填空、程序	★★
JMP/CALL/RET、条件转移、LOOP 循环控制	填空、程序	★★
处理器控制、IN/OUT、中断指令	选择、填空	★
80286/80386/80486 增强与新增指令	选择	★

1. 指令系统概述与操作数类型（课件 p.2–5）

概念精讲 - 指令：计算机用以控制各个部件协调动作的命令。指令系统：计算机能够执行的全部命令的集合，取决于硬件设计，**因机而异，没有通用性**（课件 p.2）。 - 指令中应包含三类信息：执行的运算、运算结果的去向、运算数据的来源（课件 p.2）。 - 指令格式：`操作码 [操作数],[操作数]`，即「操作码 目的操作数，源操作数」。源操作数是加工之前的数据，目的操作数是加工之后形成的数据（课件 p.3）。 - 操作数的表征方法：表征数据本身→立即数；表征数据存放的地址→寄存器或存储器（课件 p.4）。

三种操作数类型（课件 p.5）

类型	含义	特点
立即数	操作数包含在指令中，随指令一起取出	只能作源操作数
寄存器操作数	操作数在 CPU 寄存器中，指令给出寄存器编码	不访问存储器，速度快
存储器操作数	操作数在内存数据区，指令给出地址的计算方法	需按寻址方式求 EA/PA

易错警示：指令系统没有通用性（判断题常考）；「目的操作数」既可以是存放结果的寄存器/存储单元，也参与运算（如 ADD）。

2. 寻址方式（课件 p.6–27）

寻址方式 = 寻找指令操作数所在地址的方式，以确定数据的来源和去处（课件 p.6）。8086/8088 共 10 种。

通用公式：物理地址 PA = 段基址×16 + 有效地址 EA（偏移地址）。EA 由寻址方式决定，段基址由默认段寄存器或段超越前缀决定。

2.1 立即寻址（课件 p.7–8）

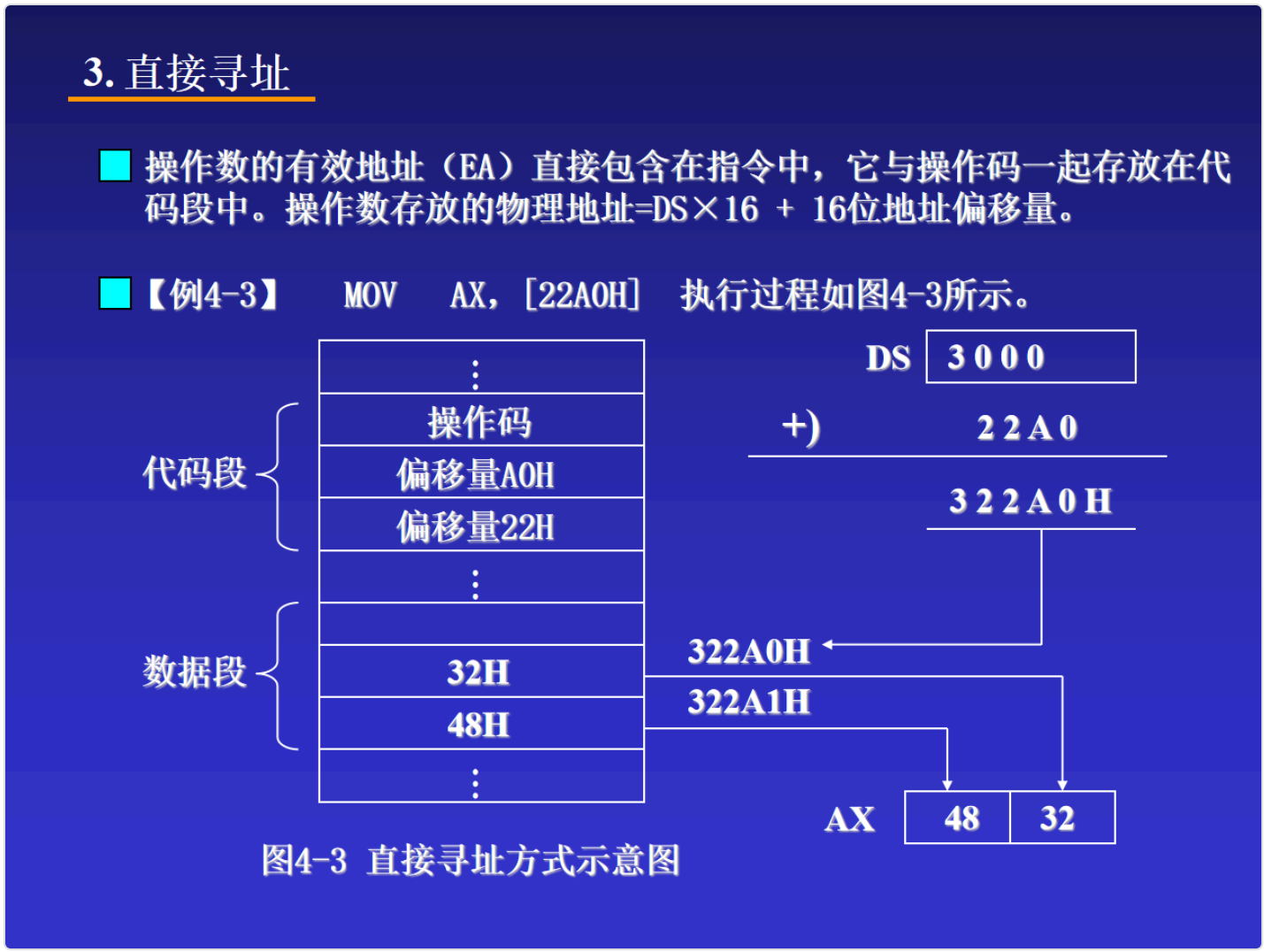
操作数（8/16 位立即数）直接包含在指令中，放在操作码之后，**低字节在低地址**。主要用于给寄存器/存储器赋初值。 - 例：`MOV AX，080AH` → AH=08H，AL=0AH（课件 p.7 例4-1）。 - 注意：① 只能用于**源操作数**；② **不执行总线周期**，执行速度快（课件 p.8）。

2.2 寄存器寻址（课件 p.9–10）

操作数在寄存器中。16 位：AX、BX、CX、DX、SI、DI、SP、BP；8 位：AH、AL、BH、BL、CH、CL、DH、DL。 - 例：`MOV AX，BX`（课件 p.9 例4-2）。 - 注意：① CPU 内部执行，**不执行总线周期**；② 源、目的均可，且可同时使用（课件 p.10）。

2.3 直接寻址（课件 p.11–12）

EA 直接包含在指令中（与操作码一起存放在代码段），默认段为 DS。PA = DS×16 + 16 位偏移量。



- 例4-3：`MOV AX，[22A0H]`，DS=3000H → PA=30000H+22A0H=322A0H；(322A0H)=32H、(322A1H)=48H → **AX=4832H**（低地址字节送 AL）（课件 p.11）。
- 段超越：操作数在其他段时加前缀，如 `MOV AH，ES:[2000H]` → PA=ES×16+2000H（课件 p.12）。

2.4 寄存器间接寻址（课件 p.13–15）

EA 在间址寄存器中。**只有 SI、DI、BX、BP 可作间址寄存器**（`MOV AX，[CX]` 错误）。 - 用 **SI/DI/BX**：默认段 **DS**，PA = DS×16 + (SI/DI/BX)。

4.寄存器间接寻址

寄存器间接寻址情况一

若以SI、DI、BX间接寻址，操作数通常在现行数据段中，即DS中的值左移四位加上SI、DI或BX中的有效地址即为操作数的物理地址。

【例4-5】　MOV　AX，[SI]　执行过程如图4-4所示。

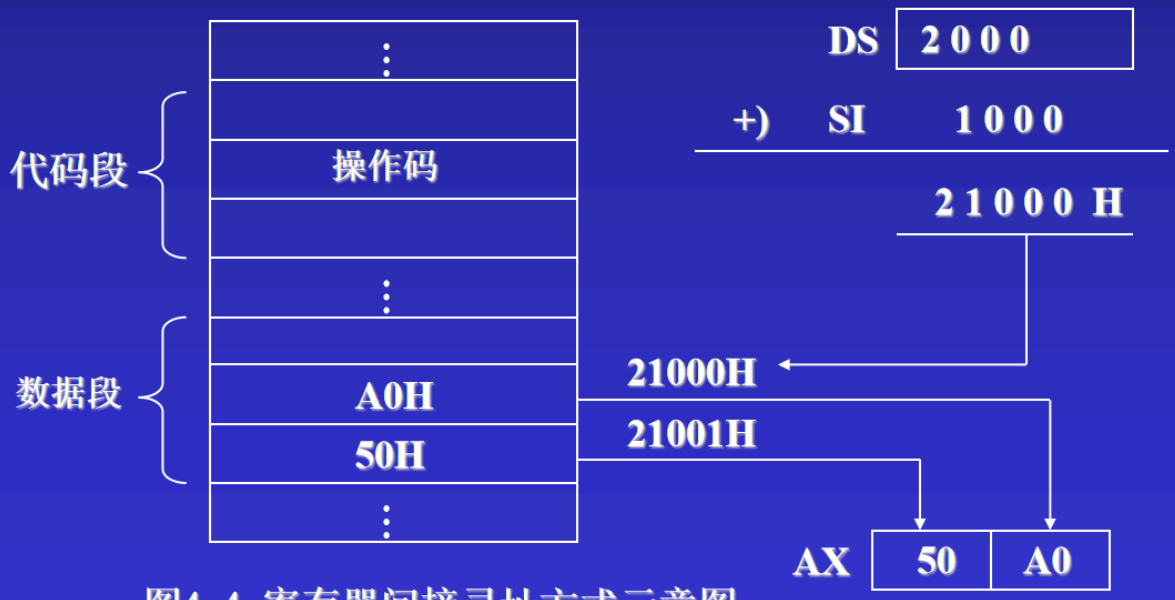
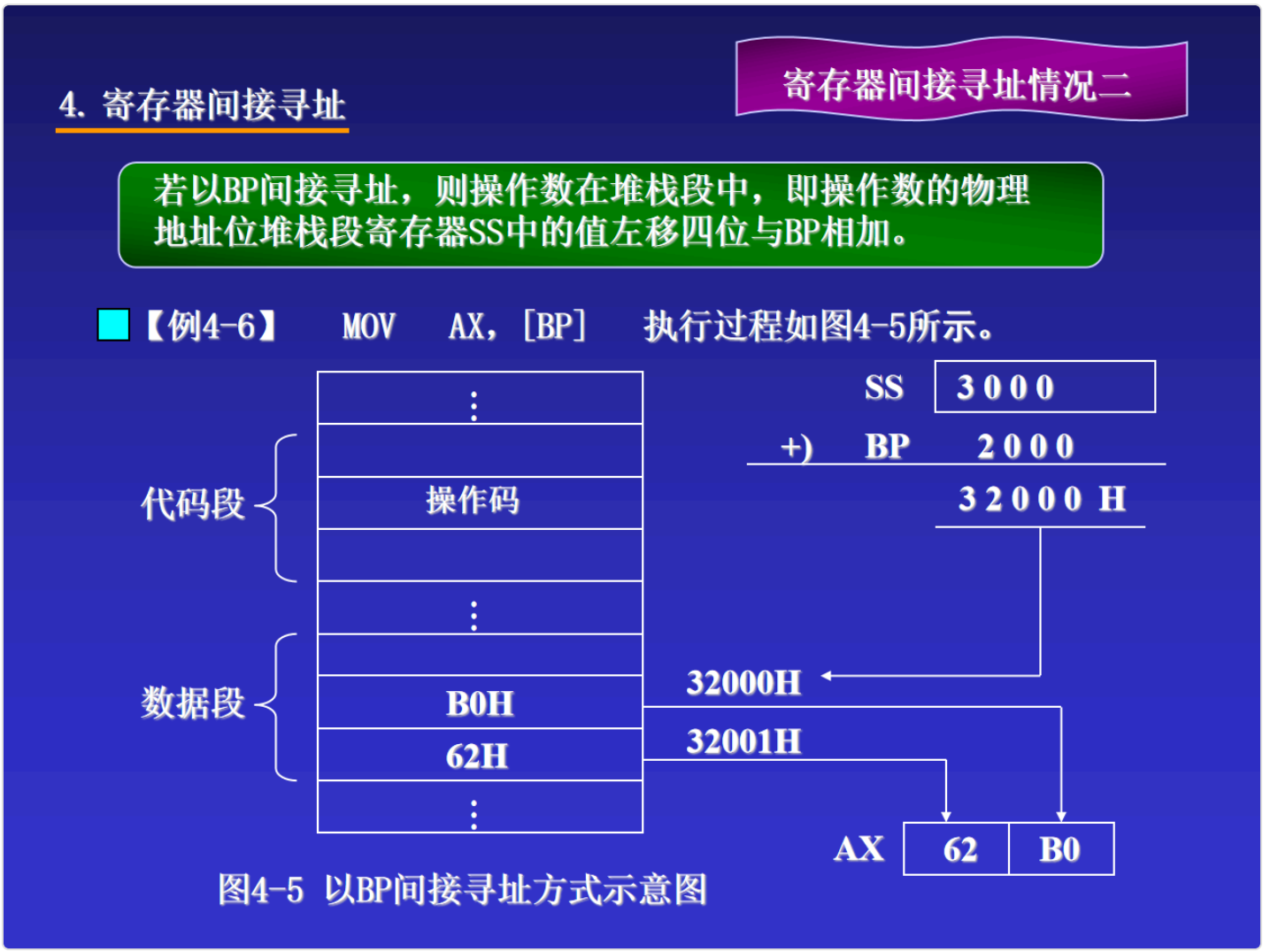


图4-4 寄存器间接寻址方式示意图

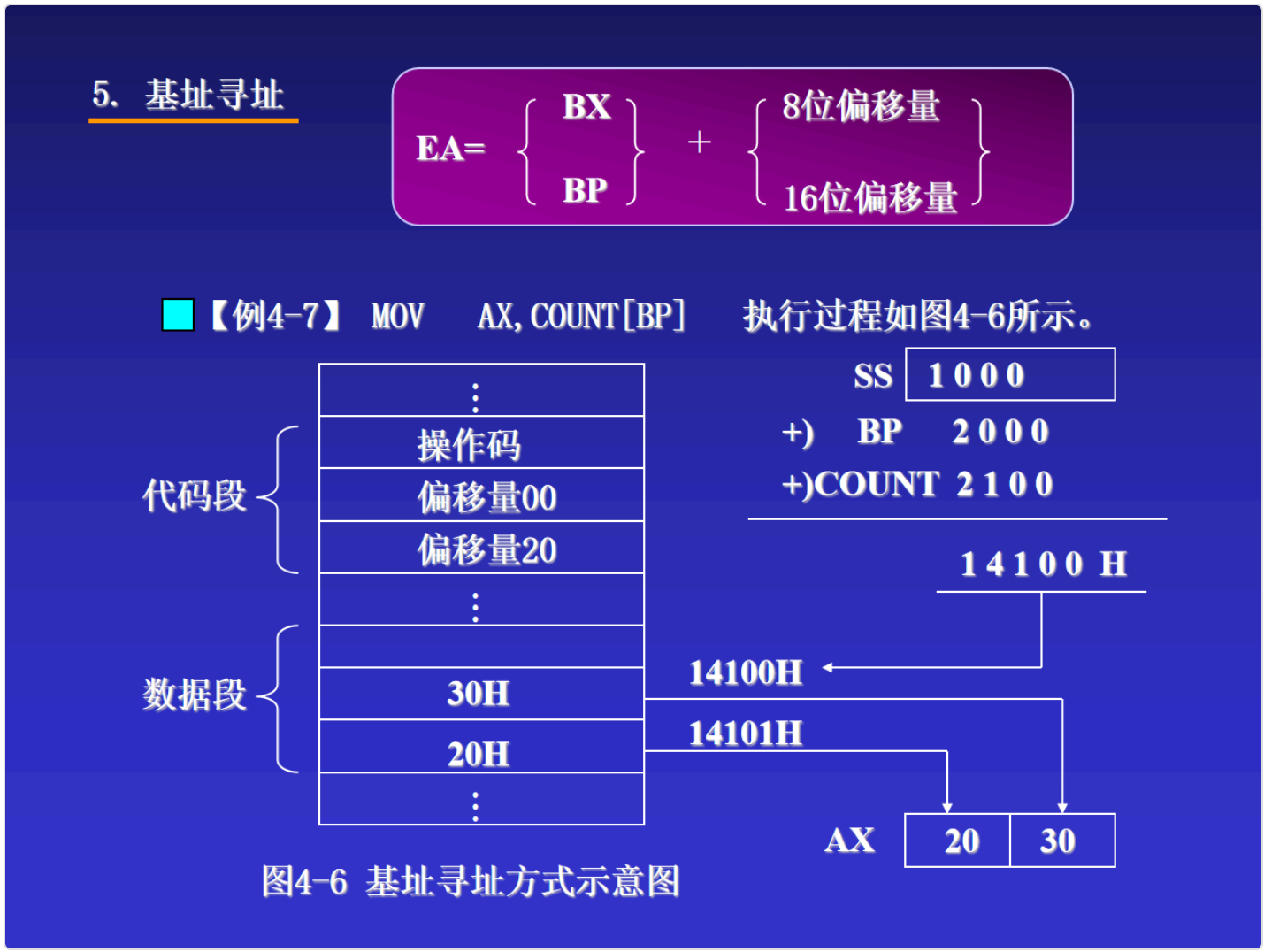
- 用 **BP**：默认段 **SS**，PA = SS×16 + BP（课件 p.14）。



- 例4-5：`MOV AX，[SI]`，DS=2000H、SI=1000H → PA=21000H，AX=50A0H（课件 p.13）。
- 例4-6：`MOV AX，[BP]`，SS=3000H、BP=2000H → PA=32000H，AX=62B0H（课件 p.14）。
- 例：DS=2100H、DI=2000H，`MOV AX，[DI]` → PA=21000H+2000H=23000H，取 23000H、23001H 两单元送 AX（课件 p.15）。
- 不在默认段时同样用段超越前缀（课件 p.15）。

2.5 基址寻址（课件 p.16–17）

EA = (BX 或 BP) + 8 位/16 位偏移量（位移量）。**BX 默认 DS，BP 默认 SS**。



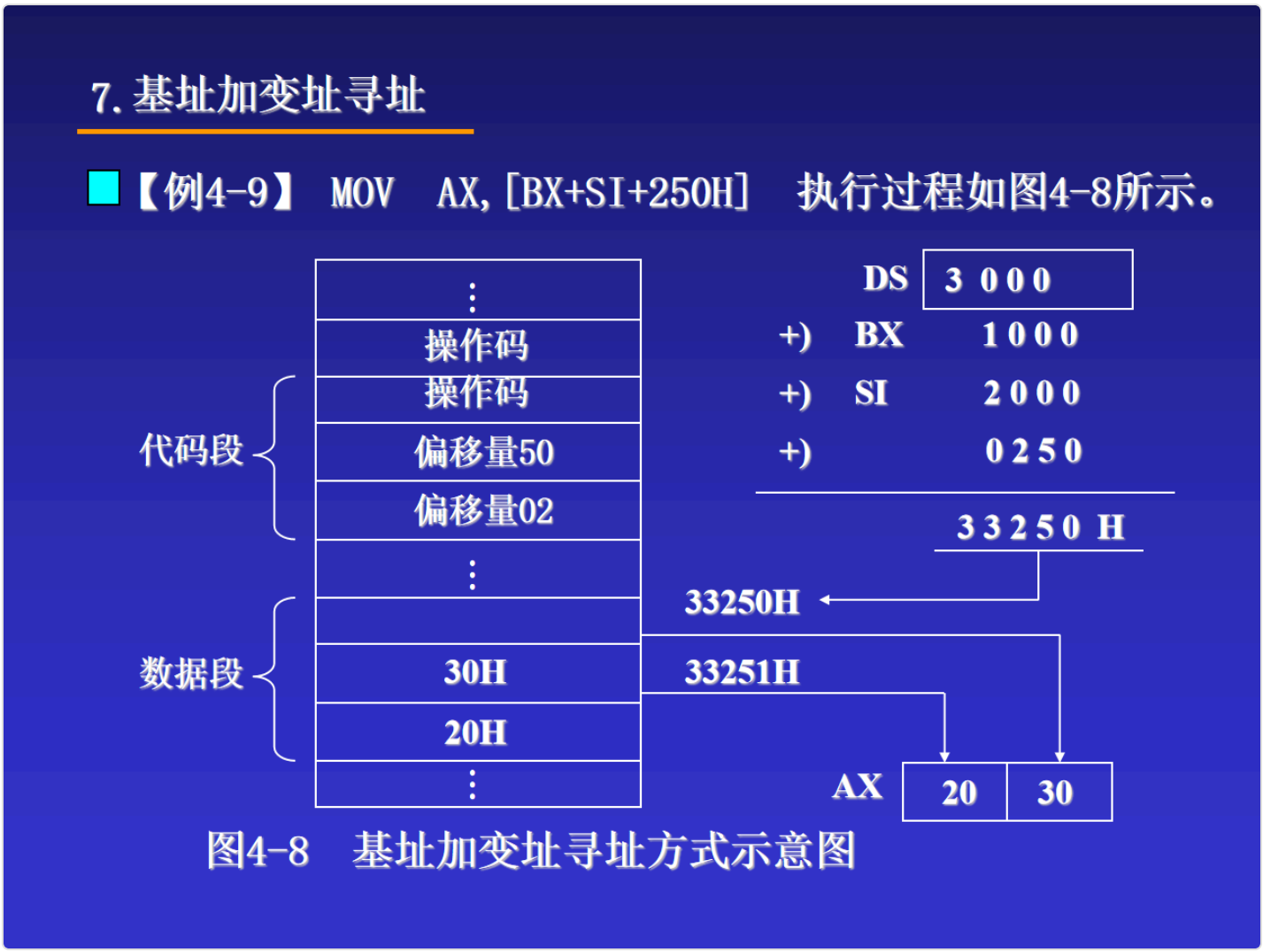
- 例4-7：`MOV AX，COUNT[BP]`，图中 SS=1000H、BP=2000H、COUNT=2100H → PA=10000H+2000H+2100H=14100H，AX=2030H（课件 p.16）。
- 例：SS=2000H、BP=1000H、COUNT=2000H → PA=20000H+1000H+2000H=23000H（课件 p.17）。
- 等价写法：`MOV BL，2[BX]` = `MOV BL，[BX+2]`，即 (DS:[BX+2])→BL（课件 p.17）。

2.6 变址寻址（课件 p.18–20）

EA = (SI 或 DI) + 8 位/16 位偏移量，**默认段 DS**（课件 p.18）。 - 例4-8：`MOV AX，COUNT[SI]`，图中 DS=3000H、SI=2000H、COUNT=3000H → PA=35000H，AX=2030H（课件 p.19）。 - 例：DS=3000H、SI=1000H、COUNT=2000H → PA=30000H+1000H+2000H=33000H（课件 p.20）。

2.7 （相对）基址加变址寻址（课件 p.21–22）

EA = (BX 或 BP) + (SI 或 DI) + 8 位/16 位偏移量。**用 BX 默认 DS，用 BP 默认 SS**（课件 p.21）。基址与变址只能各取一个（BX+BP、SI+DI 组合非法，补充常识）。



- 例4-9：`MOV AX，[BX+SI+250H]`，DS=3000H、BX=1000H、SI=2000H → PA=30000H+1000H+2000H+0250H=33250H，AX=2030H（课件 p.22）。

2.8 数据串寻址（课件 p.23）

用于串操作指令：**SI 指向源串（默认 DS），DI 指向目标串（默认 ES，且 ES 不可超越）**；执行后 SI、DI 自动 ±1（字节）或 ±2（字），方向由 DF 决定。 - 例4-10：`MOVSB` 执行后：[DI]←[SI]，SI←SI±1，DI←DI±1（课件 p.23）。

2.9 I/O 端口寻址（课件 p.24）

方式	端口地址来源	寻址范围
直接端口寻址	指令中直接给出 8 位端口地址	256 个端口（00H–FFH）
间接端口寻址	DX 给出 16 位端口地址	64K 个端口

- 例4-11：`IN AL，20H`（读 20H 端口→AL）；`OUT DX，AL`（AL→DX 所指端口）（课件 p.24）。

2.10 隐含寻址（课件 p.25）

指令中不明显给出操作数，操作数隐含约定。例4-12：`AAA` 隐含操作数为 AH 和 AL（对 AL 做十进制加法调整，结果放 AH、AL）（课件 p.25）。

2.11 寻址方式总表与综合计算

寻址方式	EA 构成	默认段	例
立即	无（数在指令中）	—	MOV AX，080AH
寄存器	无（数在寄存器）	—	MOV AX，BX
直接	16 位偏移量	DS	MOV AX，[22A0H]
寄存器间接	BX/BP/SI/DI	BX/SI/DI→DS；BP→SS	MOV AX，[SI]
基址（相对基址）	BX/BP + 位移	BX→DS；BP→SS	MOV AX，COUNT[BP]
变址（相对变址）	SI/DI + 位移	DS	MOV AX，COUNT[SI]
基址加变址（相对）	BX/BP + SI/DI + 位移	BX→DS；BP→SS	MOV AX，[BX+SI+250H]
串	SI（源）、DI（目标）	DS（源）、ES（目标）	MOVSB
I/O 端口	8 位立即数或 DX	—（I/O 空间）	IN AL，20H
隐含	约定寄存器	—	AAA

综合例（课件 p.26–27）：BX=0158H，DI=10A5H，位移量=1B57H，DS=2100H：

方式	EA	PA=21000H+EA
① 直接	1B57H	22B57H
② 寄存器间接(BX)	0158H	21158H
③ 相对基址(BX)	0158H+1B57H=1CAFH	22CAFH
④ 变址(DI)	10A5H	220A5H
⑤ 相对变址(DI)	10A5H+1B57H=2BFCH	23BFCH
⑥ 基址加变址(BX+DI)	0158H+10A5H=11FDH	221FDH
⑦ 相对基址加变址	0158H+10A5H+1B57H=2D54H	23D54H

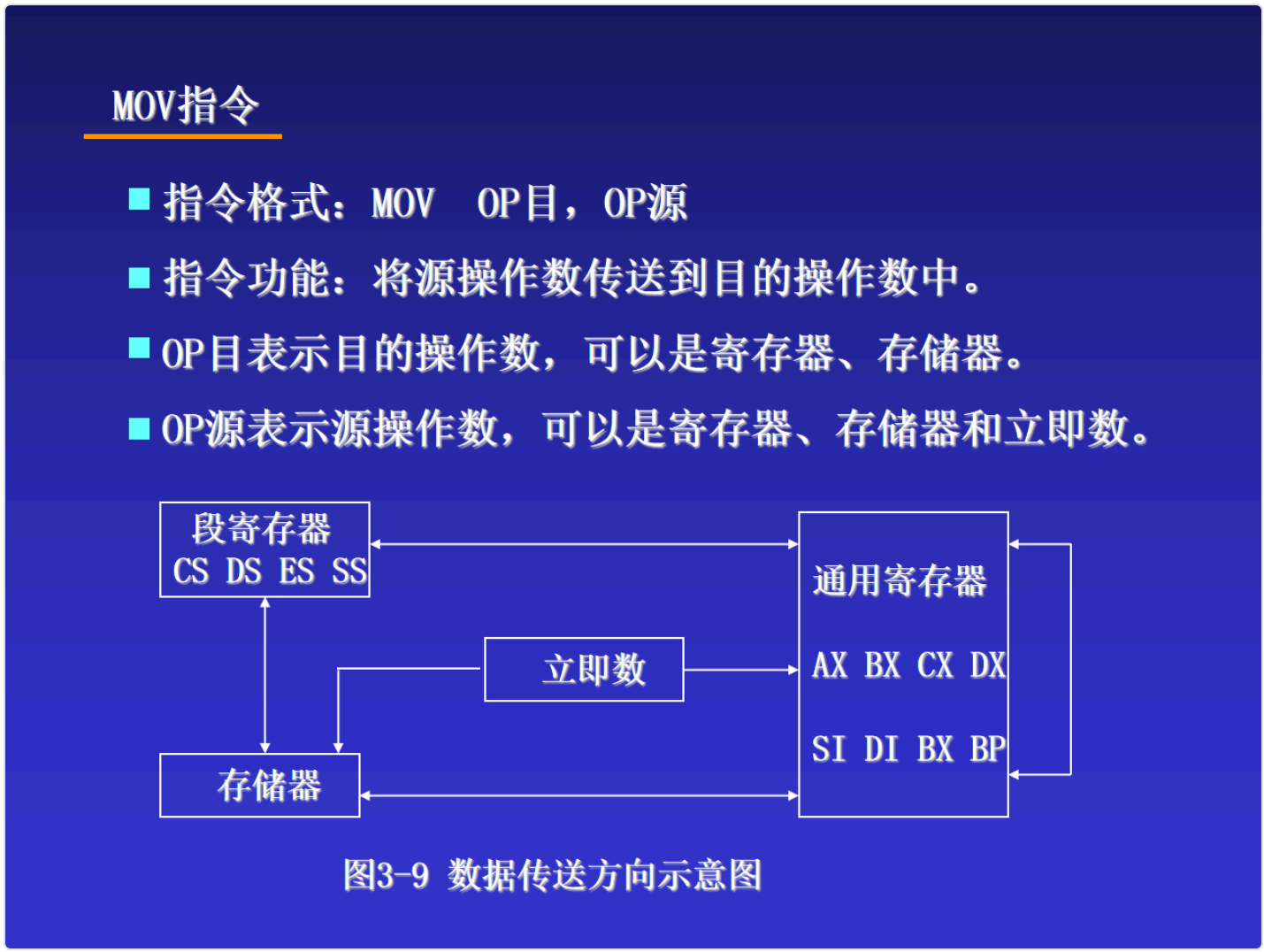
易错警示：① 间址/基址/变址寄存器只能用 BX、BP、SI、DI，CX、DX、SP 不行；② 见 BP 想 SS，见 BX/SI/DI 想 DS；③ EA 是 16 位偏移地址，PA 是 20 位物理地址，段基址要×16（左移 4 位）；④ 字操作取两个连续字节，**低地址字节→低 8 位寄存器**；⑤ 串操作目标段 ES 不能用前缀超越。

3. 数据传送指令（课件 p.29–54）

8086 指令系统共九大类、133 种基本指令，通过寻址方式与字节/字形式组合可构成上千条指令（课件 p.28）。数据传送类分三组（课件 p.29）：通用传送（MOV、PUSH、POP、XCHG、XLAT）、目标地址传送（LEA、LDS、LES）、标志传送（LAHF、SAHF、PUSHF、POPF）。

3.1 MOV（课件 p.31–34）

格式：`MOV OP目，OP源`；功能：源→目的。OP目：寄存器、存储器；OP源：寄存器、存储器、立即数。

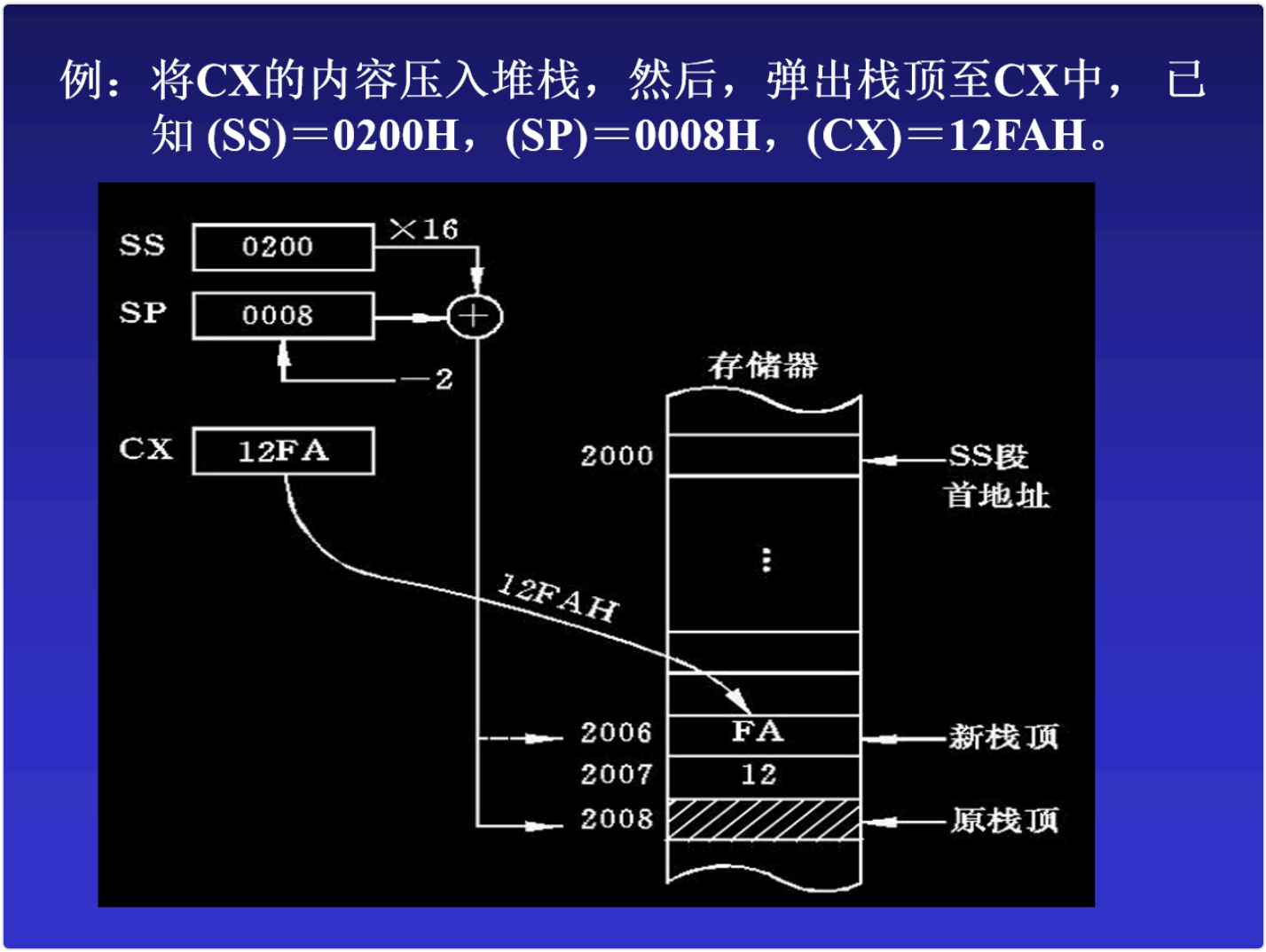


六条铁律（课件 p.33）：1. 源、目的类型必须一致（字节对字节、字对字）；2. 立即数不能作目的操作数，**也不能直接送段寄存器**；3. 段寄存器之间、存储器单元之间不能直接传送；4. **CS、IP 不能作目的操作数**（`MOV CS，AX` ×）；5. 两操作数可采用不同寻址方式；6. 一般传送指令**不影响标志位**。

用法示例（课件 p.32）：`MOV AL，BL`；`MOV SP，2AC0H`；`MOV [DI]，AX`；`MOV SI，ES:[BP]`；`MOV WORD PTR [SI]，6070H`（存储器目的操作数类型不明时用 WORD PTR/BYTE PTR 说明）。 正误判断例见习题册题16（课件 p.34）。

3.2 堆栈操作 PUSH/POP（课件 p.35–43）

堆栈是按「先进后出」原则工作的存储区；SS 给段地址，**SP 始终指向当前栈顶**（最新入栈数据所在单元）。PUSH 向低地址扩展，POP 向高地址恢复（课件 p.35）。 - `PUSH OP`（16 位寄存器/存储器/段寄存器）：① SP←SP-1，[SP]←OP 高字节；② SP←SP-1，[SP]←OP 低字节（课件 p.36）。 - `POP OP`（16 位）：① OP 低字节←[SP]，SP←SP+1；② OP 高字节←[SP]，SP←SP+1（课件 p.38）。



注意（课件 p.43）：① 堆栈操作**总是按字**进行，PUSH 使 SP-2、POP 使 SP+2；② 操作数可以是寄存器、存储器、段寄存器，**不能是立即数**（8086）；③ `PUSH CS` √ 但 `POP CS` ×；④ 主要用于子程序/中断的**现场保护与恢复**。 例：SS=0200H、SP=0008H、CX=12FAH，PUSH CX 后 SP=0006H、(02006H)=FAH、(02007H)=12H；再 POP CX 后 SP=0008H、CX=12FAH（课件 p.40–41）。 例：AX=1122H、BX=3344H、SP=1010H，执行 PUSH AX / PUSH BX / POP AX / POP BX 后 **AX=3344H、BX=1122H、SP=1010H**（两数交换）（课件 p.42）。

3.3 XCHG（课件 p.44–45）

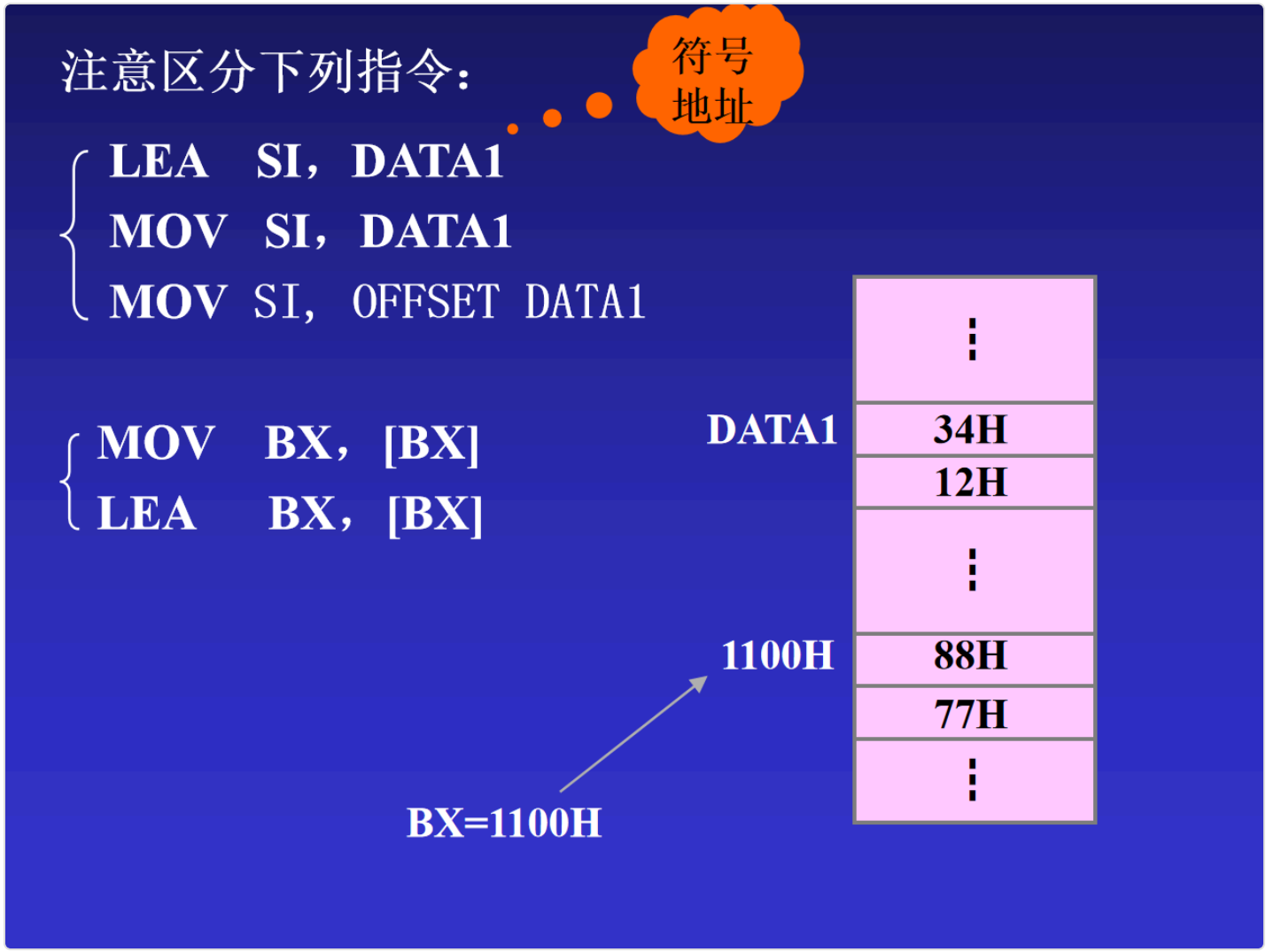
格式：`XCHG OP1，OP2`；功能：两操作数内容互换。操作数为通用寄存器或存储器。 **限制**：段寄存器、IP 不能作操作数；两存储单元之间不能直接交换；立即数不行（课件 p.44–45）。 例：`XCHG AL，BL`、`XCHG BX，CX` √；`XCHG AX，1122H`、`XCHG DS，AX`、`XCHG [SI]，[DI]` ×（课件 p.45）。

3.4 XLAT 换码（课件 p.46–48）

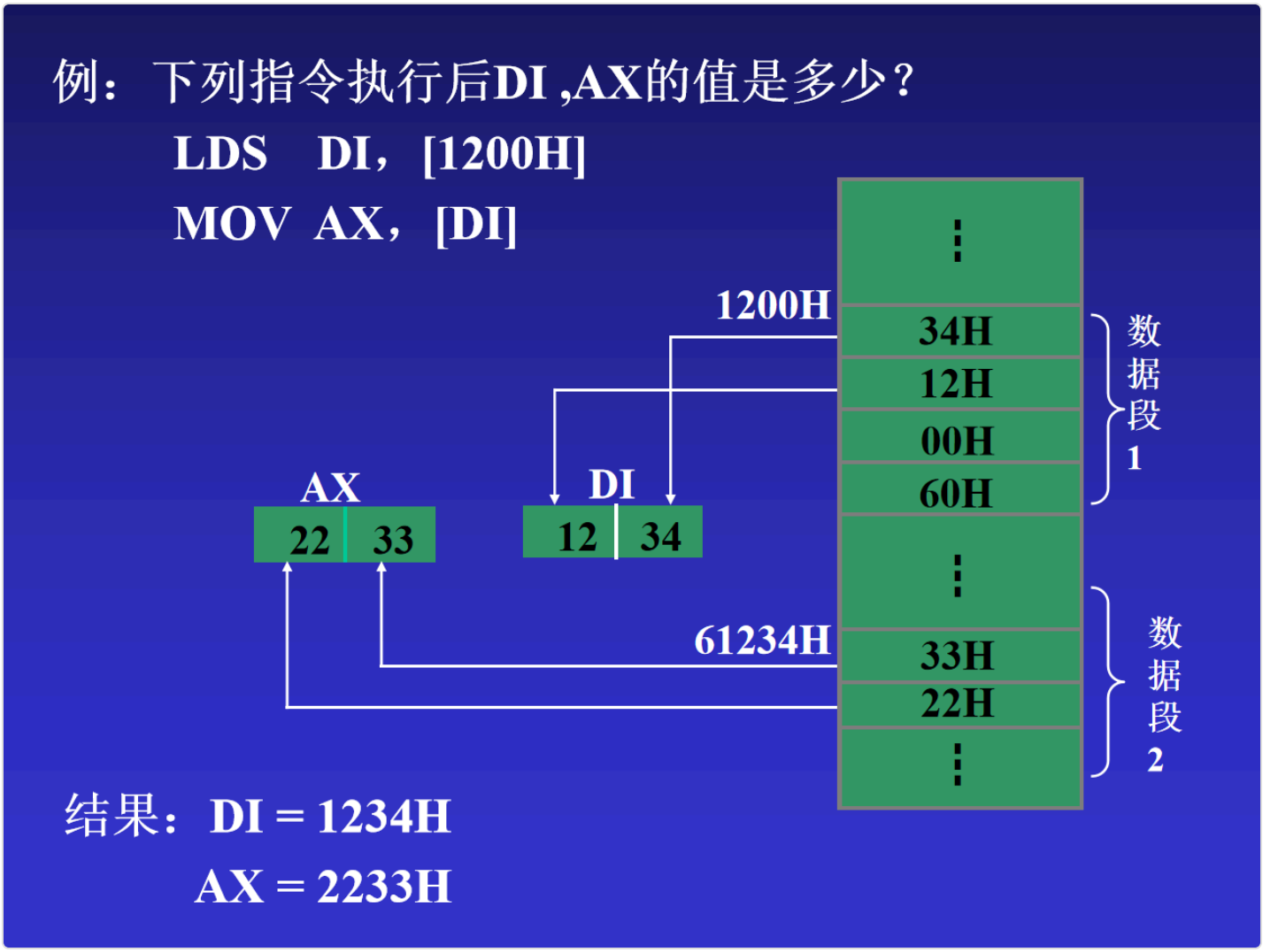
格式：`XLAT`；功能：**AL←(DS:[BX+AL])**，即 BX（表首偏移）+AL（表内序号）为 EA 查表，结果送 AL。 使用前提：数据段中预置字节表，**表长≤256 字节**；表首偏移装 BX，序号装 AL（课件 p.46）。只能用 BX、AL，**不影响标志位**（课件 p.47）。 例：表首 2000H、DS=4000H，查序号 9：`MOV BX，2000H` / `MOV AL，09H` / `XLAT` → EA=2009H，PA=42009H，**AL=39H**（'9' 的 ASCII）（课件 p.48；课件该页助记符误写为 XALT）。

3.5 目标地址传送 LEA/LDS/LES（课件 p.49–53）

- `LEA OP目，OP源`：把源操作数的**有效地址 EA**（不是内容）送 16 位通用寄存器。OP源必须是存储器操作数。不影响标志位。等效于 `MOV 寄存器，OFFSET 变量`（课件 p.49）。 例：`LEA BX，[BP+SI]` → BX=BP+SI；`LEA DI，TABLE`；`LEA AX，[5678H]` → AX=5678H（课件 p.49）。



区分（课件 p.50，图中 [DATA1]=34H、[DATA1+1]=12H，[1100H]=88H、[1101H]=77H）：`LEA SI，DATA1` 与 `MOV SI，OFFSET DATA1` → SI=DATA1 的偏移地址；`MOV SI，DATA1` → SI=1234H（取内容）；`MOV BX，[BX]` → BX=7788H（取内容）；`LEA BX，[BX]` → BX=1100H（取 EA）。 - `LDS OP目，OP源`：从源指定的 4 个连续单元取双字指针，**低地址字→OP目（16 位通用寄存器），高地址字→DS**（课件 p.51）。例：`LDS DI，[2130H]`：2130H/2131H→DI，2132H/2133H→DS。 - `LES OP目，OP源`：同 LDS，但高地址字→**ES**（课件 p.53）。



例（课件 p.52）：`LDS DI，[1200H]` → DI=1234H、DS=6000H；`MOV AX，[DI]` → PA=60000H+1234H=61234H，(61234H)=33H、(61235H)=22H → **AX=2233H**。

3.6 标志传送（课件 p.54）

指令	功能
LAHF	标志寄存器低 8 位→AH
SAHF	AH→标志寄存器低 8 位
PUSHF	16 位标志寄存器入栈，SP-2
POPF	栈顶字→标志寄存器，SP+2

易错警示：MOV/LEA/LDS/LES/XCHG/XLAT/PUSH/POP 均不影响标志位（POPF、SAHF 除外，它们本身就是写标志）；`MOV SI，DATA1` 取内容、`LEA SI，DATA1` 取地址，一字之差结果完全不同；XLAT 的表只能在 DS 段、长度≤256。

4. 算术运算指令（课件 p.55–102）

可对二进制或 BCD 码做加、减、乘、除；操作数为 8/16 位无符号或带符号数（课件 p.55）。

4.1 加法：ADD / ADC / INC（课件 p.55–62）

指令	格式	功能	标志
ADD	ADD OP目，OP源	OP目←OP目+OP源	影响 CF、OF、SF、ZF、AF、PF
ADC	ADC OP目，OP源	OP目←OP目+OP源+CF	同上
INC	INC OP	OP←OP+1	影响 AF、OF、PF、SF、ZF，**不影响 CF**

- 操作数要求与 MOV 相同，但**段寄存器不能作操作数**；两操作数类型相同、不能同为存储器（课件 p.55–56）。
- ADC 用于多字节运算：低位产生的进位加到高位（课件 p.58）。
- INC 常用于修改偏移地址和计数；操作数为 8/16 位寄存器或存储器，不能为立即数；存储器操作数须用 BYTE PTR/WORD PTR 指明类型（课件 p.61）。
- 例（双字相加，课件 p.58）：0123FAB5H+0ABC212AH：`MOV DX，0123H` / `MOV AX，0FAB5H` / `ADD AX，212AH`（AX=1BDFH，CF=1）/ `ADC DX，0ABCH`（DX=0BE0H，CF=0）→ 结果 0BE01BDFH。
- 例（4 字节相加程序，课件 p.59–60）：见习题册题25。

4.2 减法：SUB / SBB / DEC / NEG / CMP（课件 p.63–73）

指令	格式	功能	标志
SUB	SUB OP目，OP源	OP目←OP目-OP源	影响全部状态标志
SBB	SBB OP目，OP源	OP目←OP目-OP源-CF	同上，用于多精度减法
DEC	DEC OP	OP←OP-1	影响 AF、OF、PF、SF、ZF，**不影响 CF**
NEG	NEG OP	OP←0-OP（求补）	CF：操作数≠0 则 CF=1，=0 则 CF=0；对 80H/8000H 取补操作数不变但 OF=1
CMP	CMP OP目，OP源	OP目-OP源，**不回送结果**，只置标志	同 SUB

- NEG 例：`MOV AX，1` / `NEG AX` → AX=0FFFFH（课件 p.66）。
- CMP 后的判断规则（课件 p.68–69，以 CMP A，B 为例）：

比较目的	依据标志	条件与转移
相等	ZF	ZF=1→A=B（JZ）；ZF=0→A≠B（JNZ）
无符号大小	CF	CF=1→A<B（JC）；CF=0→A≥B（JNC）
带符号、同号	SF	SF=1→A<B（JS）；SF=0→A≥B（JNS）
带符号、异号	SF⊕OF	SF⊕OF=1→A<B；SF⊕OF=0→A≥B

统一记法：带符号比较用 SF⊕OF（同号时 OF=0，退化为 SF）。 - 例（课件 p.71）：A=-36、B=+127，A-B=[A]补+[-B]补=11011100+10000001=（1）01011101，进位自然丢失；SF=0、OF=1（负+负得正，溢出），SF⊕OF=1 → **A<B**（真值 -163 超出 8 位范围）。 - 程序例：50 字节比较（课件 p.70）、20 个数求最大（课件 p.72–73），见习题册题27、题29。

4.3 乘法：MUL / IMUL（课件 p.74–78）

指令	操作数	功能
MUL OP	8/16 位寄存器或存储器（无符号）	字节：AX←AL×OP；字：(DX,AX)←AX×OP
IMUL OP	同上（带符号）	同 MUL，按带符号数运算

3. 乘法指令

▲（1）无符号数乘法指令

指令格式：MUL　OP
指令功能：　AX←AL×OP　8位数乘法；　(DX、AX)←AX×OP　16位数乘法。完成两个无符号数的8位/16位二进制数的乘法运算。

●指令中给出的操作数是乘数，它可以是8位/16位通用寄存器及存储器；另一操作数被乘数隐含在AL/AX中，乘积存放在AH：AL或DX：AX中，如图3-13所示。

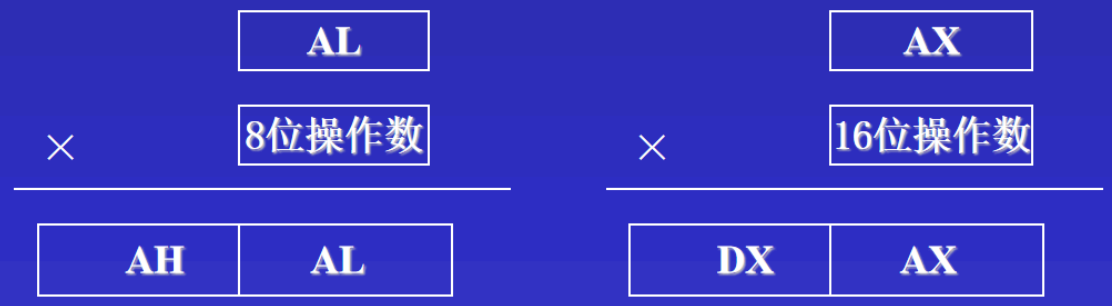
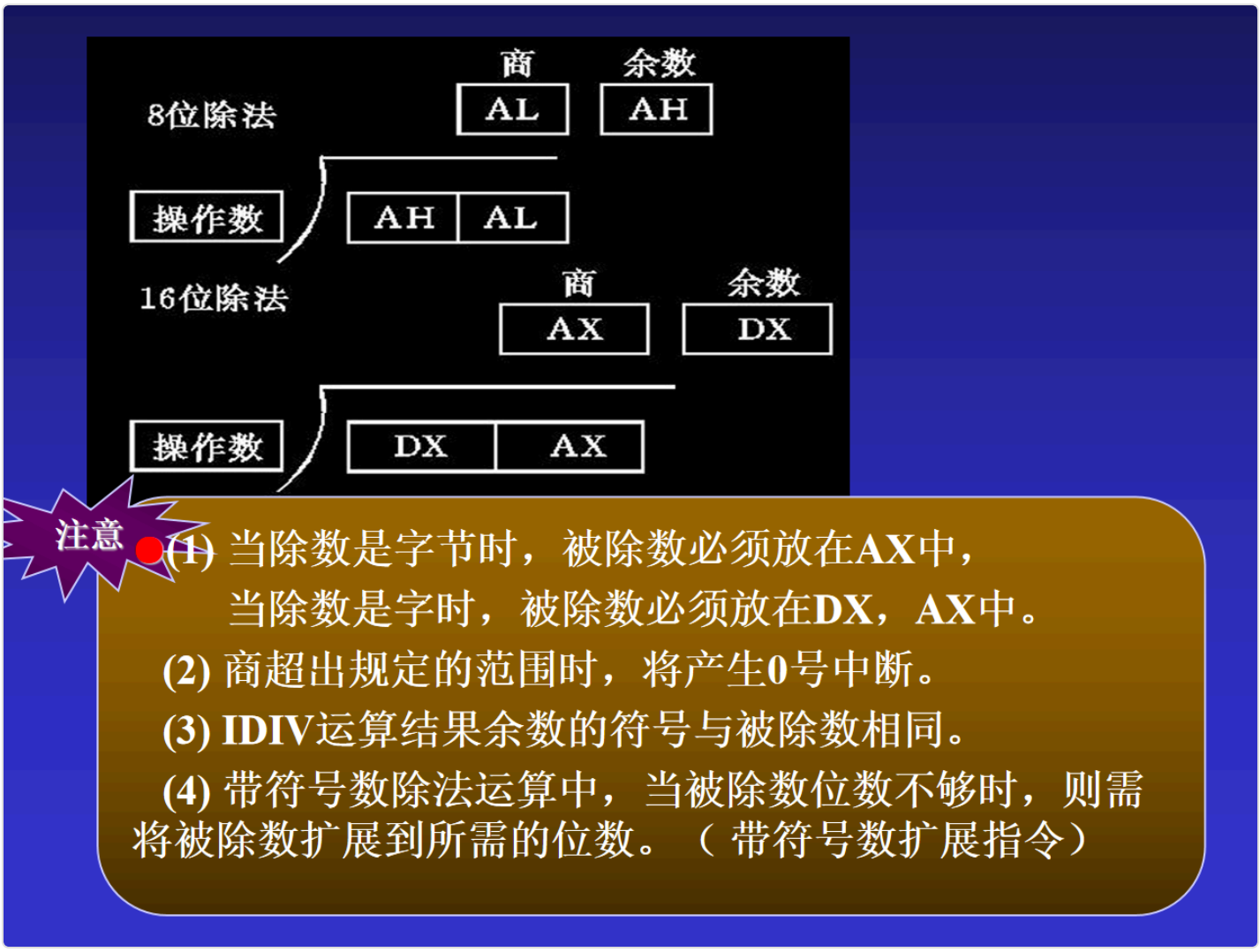


图3-13　乘法运算示意图

约定（课件 p.76）：① 被乘数隐含在 AL/AX，乘积在 AH:AL 或 DX:AX；② 源操作数**不允许立即数**；③ 只影响 **CF、OF**，其余标志无定义；④ MUL：字节积 AH=0（或字积 DX=0）则 CF=OF=0，否则 CF=OF=1；IMUL：高一半是低一半的**符号扩展**则 CF=OF=0，否则 CF=OF=1（课件 p.75–76）。 - 例（课件 p.77）：AL=0FEH（无符号 254 / 带符号 -2），BH=0AH（10）：`MUL BH` → 254×10=2540=09ECH，AX=09ECH，AH≠0 → CF=OF=1；`IMUL BH` → (-2)×10=-20=FFECH，AX=FFECH，AH=FFH 是 AL=ECH 的符号扩展 → CF=OF=0。 - 例（课件 p.78）：M1、M2 为 16 位无符号数，积存 R 开始的双字：`MOV AX，M1` / `MUL M2` / `MOV R，AX` / `MOV R+2，DX`。

4.4 除法：DIV / IDIV 与符号扩展 CBW / CWD（课件 p.79–85）

指令	功能
DIV OP	无符号：字节 AX/OP→AL 商、AH 余数；字 (DX,AX)/OP→AX 商、DX 余数
IDIV OP	带符号，同上；**余数符号与被除数相同**
CBW	AL 符号扩展到 AH：AL<80H→AH=00H；AL≥80H→AH=FFH
CWD	AX 符号扩展到 DX：AX<8000H→DX=0000H；否则 DX=FFFFH



注意（课件 p.80–82）：① 被除数必须是除数的双倍字长（字节除数→AX，字除数→DX:AX）；② **商超出范围产生 0 号中断**（字节商范围 -128~+127，字商 -32768~+32767）；③ 带符号除法位数不够先 CBW/CWD 扩展；④ CBW/CWD **不影响标志位**，常放在 IDIV 之前。 - 例（课件 p.83）：`MOV AL，44H` / `CBW` → AX=0044H；`MOV AX，0AFDEH` / `CWD` → DX=FFFFH、AX=0AFDEH；`MOV AL，86H` / `CBW` → AX=FF86H。 - 例（课件 p.85）：(-125)×10÷300：`MOV AL，83H` / `MOV BL，10` / `IMUL BL`（AX=-1250）/ `CWD` / `MOV BX，300` / `IDIV BX` → **AX=FFFCH（-4），DX=FFCEH（-50）**。 - 程序例（课件 p.84）：DATA1 起依次存被除数、除数、商、余数（各 16 位）：`LEA BX，DATA1` / `MOV AX，[BX]` / `CWD` / `IDIV WORD PTR [BX+2]` / `MOV [BX+4]，AX` / `MOV [BX+6]，DX`。

4.5 十进制（BCD）调整指令（课件 p.86–102）

BCD 码：非组合（非压缩）= 一字节一位十进制；组合（压缩）= 一字节两位十进制（课件 p.86）。四位二进制 16 个编码中 1010~1111 为无效码，运算进入/跳过无效码区必须调整：**结果>9 或 D3(D7)向高位有进位/借位 → 加 6（减 6）调整**（课件 p.87）。

指令	配合指令	调整对象	操作要点	影响标志
AAA	ADD/ADC 之后	非组合加法（AL）	低 4 位>9 或 AF=1：AL+6、AH+1、AF=CF=1、清 AL 高 4 位；否则清高 4 位及 AF、CF	只 AF、CF
DAA	ADD/ADC 之后	组合加法（AL）	低 4 位>9 或 AF=1：AL+6、AF=1；高 4 位>9 或 CF=1：AL+60H、CF=1	AF、CF、SF、ZF、PF（OF 无影响）
AAS	SUB/SBB 之后	非组合减法（AL）	低 4 位>9 或 AF=1：AL-6、AH-1、AF=CF=1、清高 4 位	只 AF、CF
DAS	SUB/SBB 之后	组合减法（AL）	低 4 位>9 或 AF=1：AL-6、AF=1；高 4 位>9 或 CF=1：AL-60H、CF=1	同 DAA
AAM	MUL 之后	非组合乘法（AX）	AH←AL/10 商，AL←AL/10 余数	SF、ZF、PF
AAD	DIV 之前	非组合除法（AX）	AL←AH×10+AL，AH←0	SF、ZF、PF

（标志影响总结见课件 p.100；AAA/AAS 对其余标志无意义，AAM/AAD 对 AF、CF、OF 无影响。） **使用纪律**：加法/减法调整必须**紧跟**在 ADD/ADC 或 SUB/SBB 之后，且只调整 AL，故多字节十进制运算必须**从低字节逐字节运算+调整**（课件 p.90、94）。 - 例（课件 p.89）：07+08：ADD 后 AL=0FH → AAA → AL=05H、AH=01H、CF=AF=1。 - 例（课件 p.91）：68H+59H=C1H（AF=1）→ DAA → AL=27H、CF=1（即 68+59=127）。 - 例（课件 p.93）：AX=0103H，SUB AL,04H → AL=FFH → AAS → AL=09H、AH=00H（即 3-4 借位得 9）。 - 例（课件 p.95）：83H-38H=4BH → DAS → AL=45H。 - 例（课件 p.97）：07×09=3FH → AAM → AH=06H、AL=03H（63）。（课件该页误写 `MUL AL,BL`，MUL 为单操作数指令，应为 `MUL BL`。） - 例（课件 p.99）：AX=0703H、BL=02H：AAD → AL=49H(73)；DIV BL → AL=24H(36)、AH=01H；AAM → AH=03H、AL=06H。 - 程序例：AX/BX 中组合 BCD 相加（课件 p.101）、1000H 起两组合 BCD 数相加（课件 p.102），见习题册题35。

易错警示：① INC/DEC **不影响 CF**（多精度循环里用 INC 修改指针不会破坏进位链，这是考点也是陷阱）；② NEG 的 CF 规则特殊（非 0 即 1）；③ MUL/IMUL 源操作数不能是立即数、只影响 CF/OF；④ 除法商溢出→0 号中断；⑤ DAA/DAS 与 AAA/AAS 的调整对象（组合/非组合）和标志影响范围不同；⑥ AAD 在 DIV **之前**、AAM 在 MUL **之后**，顺序不能反。

5. 逻辑运算指令（课件 p.103–108）

OP源：8/16 位通用寄存器、存储器或立即数；OP目：通用寄存器或存储器。**除 NOT 外，其余指令均使 OF=CF=0**（课件 p.103）。

指令	功能	典型用途	标志
AND OP目，OP源	按位与，回送	屏蔽/截取位（对应 0 清 0，对应 1 不变）	CF=OF=0，SF/ZF/PF 按结果，AF 无意义
OR OP目，OP源	按位或，回送	置 1、拼字（对应 1 置 1，对应 0 不变）	同上
XOR OP目，OP源	按位异或，回送	按位取反（对应 1 求反）、清 0	同上
NOT OP	逐位取反，回送	求反（含符号位）	**不影响标志**
TEST OP目，OP源	按位与，**不回送**，只置标志	检测某些位是 1 还是 0	同 AND

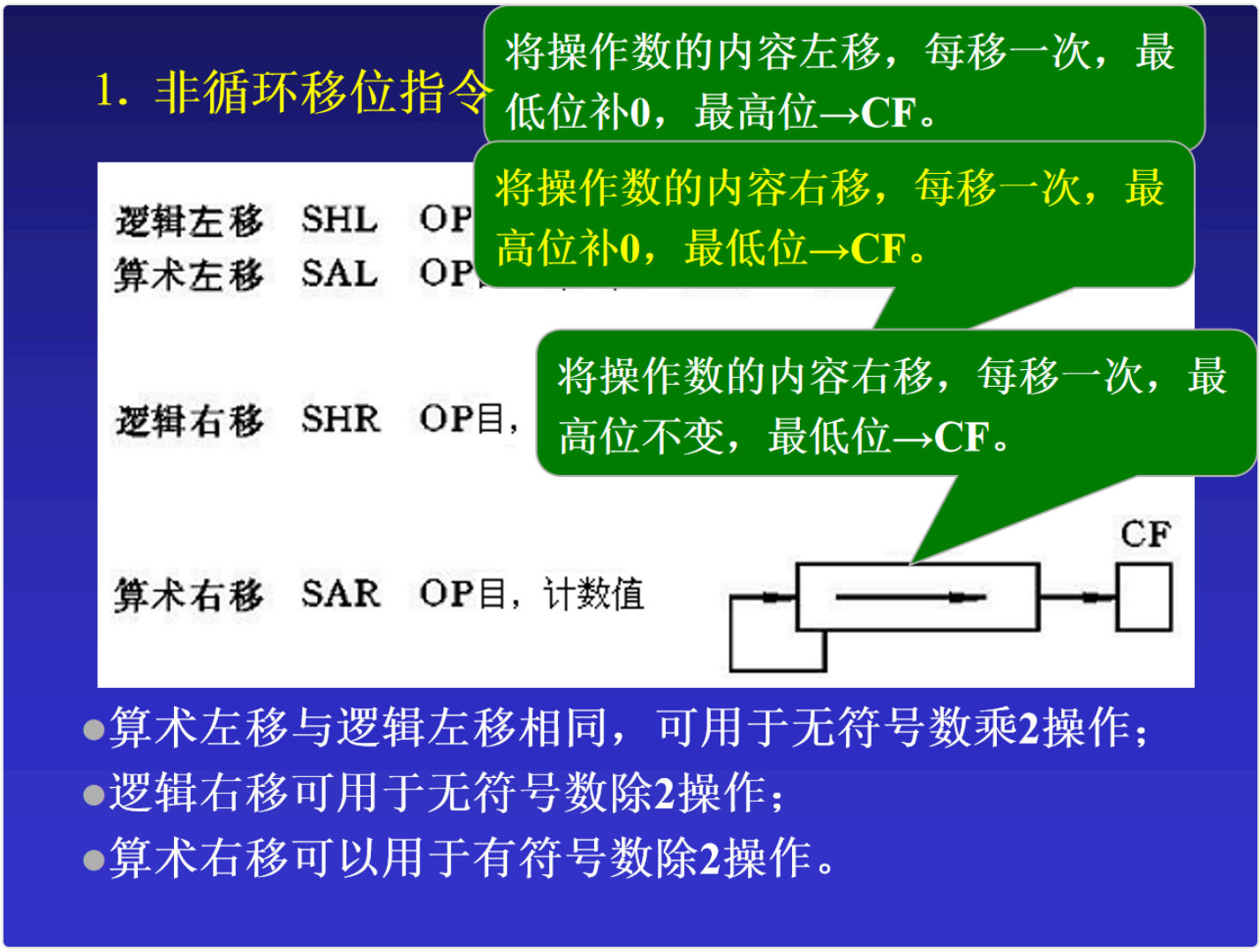
- 例：AL=46H，`AND AL，0FH` → AL=06H（屏蔽高 4 位）（课件 p.103）。
- 例：AX=8888H，`OR AX，00FFH` → AX=88FFH（低 8 位置 1）；非组合 BCD→ASCII：`OR AL，30H`（课件 p.104）。
- 例：AX=3333H，`XOR AX，00FFH` → AX=33CCH（AL 求反、AH 不变）；`XOR AL，0FFH` 为 AL 按位求反（课件 p.105）。
- **AX 清 0 的四条指令**：`XOR AX，AX`、`MOV AX，0`、`SUB AX，AX`、`AND AX，0`（课件 p.105 以 XOR AL,AL 等举例）。
- 例：AX=1，`NOT AX` → AX=0FFFEH（课件 p.106）。
- 例：测 AX 的 D15：`TEST AX，8000H`，D15=1→ZF=0，否则 ZF=1（课件 p.107）。
- 拼字综合例（课件 p.108）：AX=AAAAH、BX=BBBBH、CX=CCCCH，取 BX 高 4 位、CX 低 4 位、AX 中间 8 位取反拼成新字：`AND BH，0F0H`（BH=B0H）/ `AND CL，0FH`（CL=0CH）/ `AND AX，0FF0H`（AX=0AA0H）/ `XOR AX，0FF0H`（AX=0550H）/ `OR AH，BH` / `OR AL，CL` → **AX=B55CH**。

易错警示：NOT 不影响标志、且对符号位也求反（不是求补）；TEST 与 AND 的区别是回送与否；逻辑指令清 CF/OF 的特性常用于「清进位」。

6. 移位指令（课件 p.109–120）

分类：非循环（逻辑/算术）与循环（带/不带进位）（课件 p.109）。**计数规则：移 1 位由指令直接给出；移 2 位及以上必须先把次数装入 CL**（8086）（课件 p.109）。操作数可为除立即数外的任何寻址方式（课件 p.111）。

6.1 非循环移位（课件 p.110–112）



指令	移动方向	空位补法	移出位	用途
SHL / SAL	左	最低位补 0	最高位→CF	无符号数×2
SHR	右	最高位补 0	最低位→CF	无符号数÷2
SAR	右	最高位不变（符号位保持）	最低位→CF	带符号数÷2

- 例：BX=FFFCH（-4），`SAR BX，1` → BX=FFFEH（-2）；BL=0CH（12），`SHL BL，1` → BL=18H（24）（课件 p.111）。
- 例（×10 程序，课件 p.112）：AX×10=AX×2+AX×8：`MOV BX，AX` / `SAL BX，1` / `MOV CL，3` / `SAL AX，CL` / `ADD AX，BX`。
- 例（课件 p.116）：DS=7B00H、SI=1B79H、(7CB79H)=07H，`MOV CL，3` / `SAL [SI]，CL`：PA=7B000H+1B79H=7CB79H；07H→0EH→1CH→38H，每次移出 0 → **(7CB79H)=38H，CF=0**。

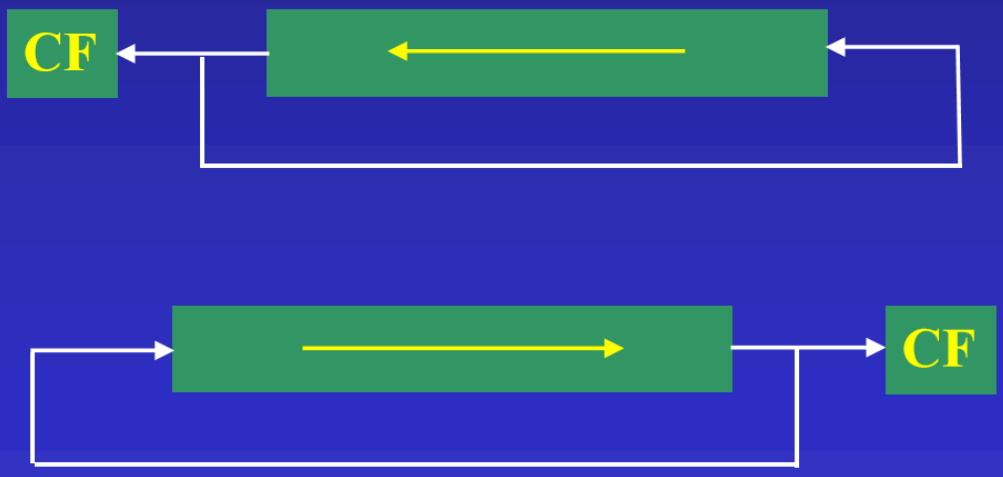
6.2 循环移位（课件 p.113–115）

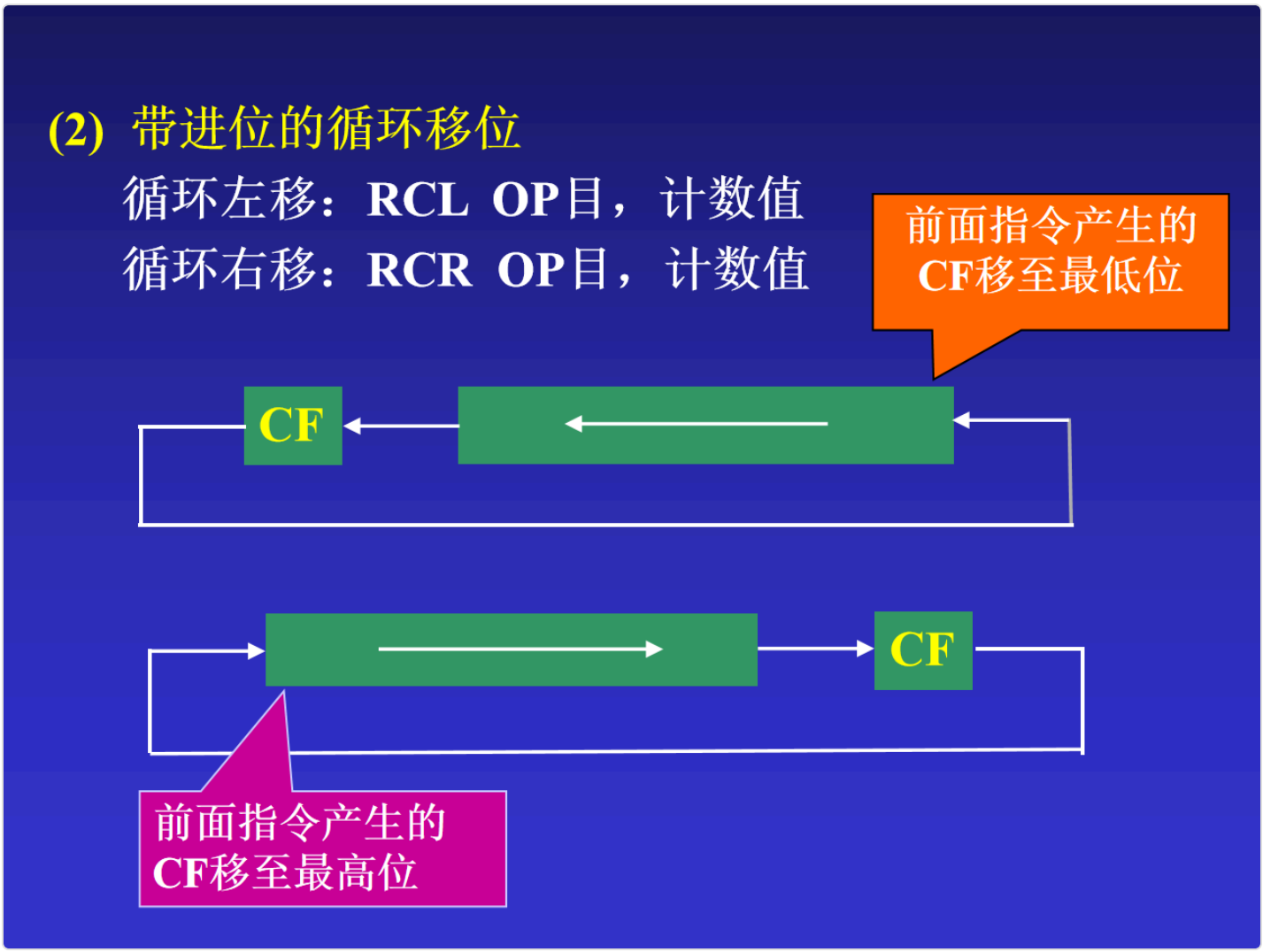
2. 循环移位指令

（1）不带进位的循环移位

循环左移：**ROL OP目，计数值**

循环右移：**ROR OP目，计数值**

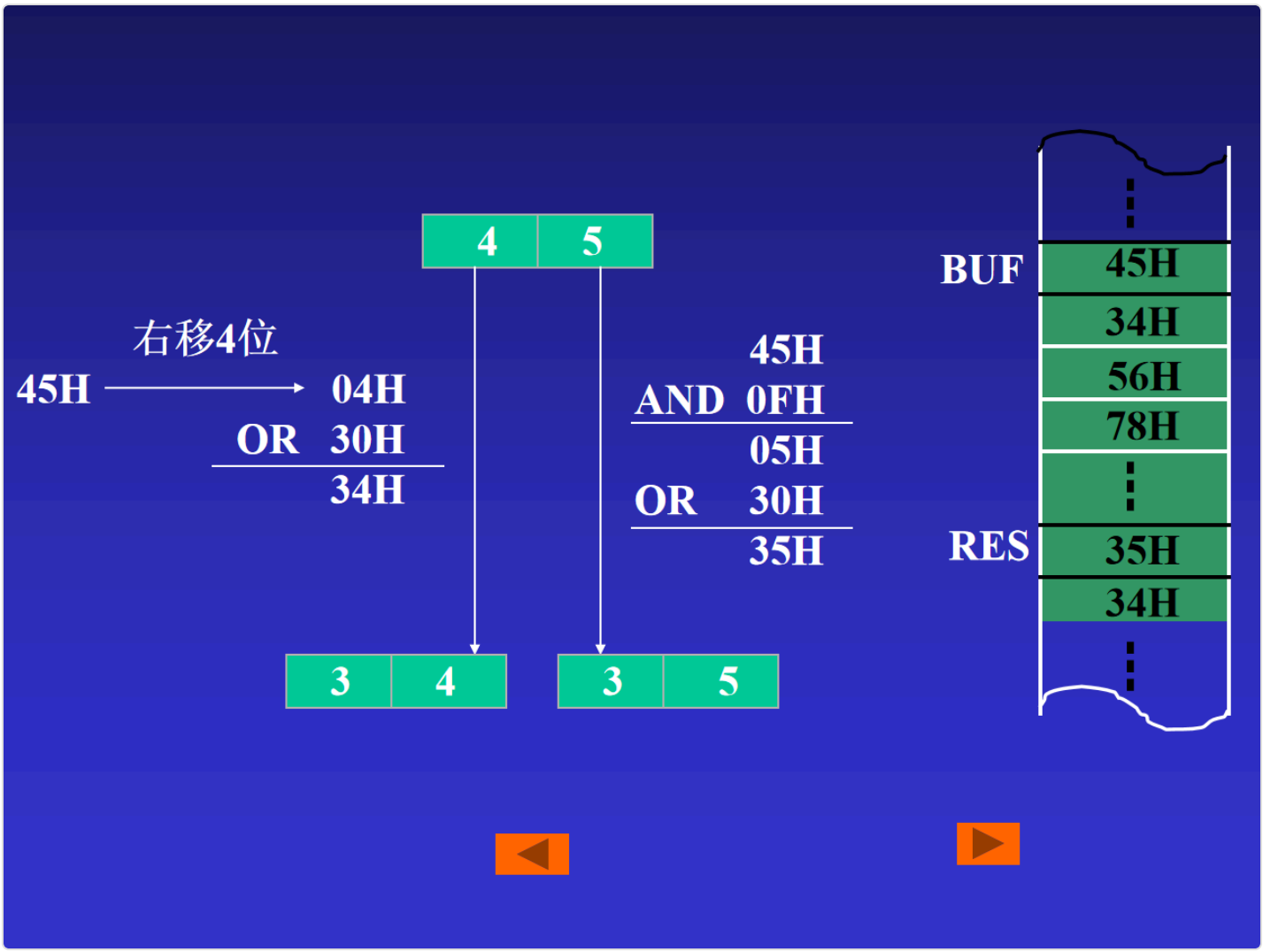




指令	环的构成	CF 的作用
ROL OP目，计数	操作数自身成环	每次移出的最高位→CF（CF 不参与环）
ROR OP目，计数	操作数自身成环	每次移出的最低位→CF
RCL OP目，计数	CF+操作数成环（9/17 位）	原 CF→最低位，移出最高位→CF
RCR OP目，计数	CF+操作数成环	原 CF→最高位，移出最低位→CF

6.3 移位应用：BCD→ASCII（课件 p.117–120）

组合 BCD 一字节拆两位：低 4 位 `AND 0FH` 后 `OR 30H`；高 4 位 `SHR 4 位` （CL=4）后 `OR 30H`。



- 程序例 1（课件 p.117–118）：BUF 起 8 个组合 BCD → RES 起 16 个 ASCII，高位 BCD 的 ASCII 放高地址（完整代码见习题册题41）。
- 程序例 2（课件 p.119–120）：1000H 起 4 个组合 BCD → 3000H 起 8 个 ASCII；用 `PUSH CX` / `POP CX` 保护串计数、临时借 CL=4 做 SHR（功能分析见习题册题42）。

易错警示：① 8086 移位次数>1 必须用 CL，写 `SHL AX，3` 非法（80286 起才允许 1~31 立即数）；② SAR 保持符号位、SHR 补 0，负数右移两者结果不同；③ 循环移位中 ROL/ROR 的 CF 只是「记录移出位」，RCL/RCR 的 CF 才参与环；④ 左移 n 位=×2^n、右移 n 位=÷2^n（SAR 对负数向负无穷取整，补充常识）。

7. 字符串操作指令（课件 p.121–134）

串 = 存储区中的数据序列；串操作对串中每个字/字节做相同操作（课件 p.121）。**三条共同约定**（课件 p.121）：1. SI 寻址源串（默认 DS），DI 寻址目标串（默认 ES）； 2. 每次操作后自动修改 SI/DI：DF=0 加 1/加 2，DF=1 减 1/减 2（CLD 清 DF、STD 置 DF）； 3. 串长放 CX。

指令	功能	指针修改	常用前缀
MOVS / MOVSB / MOVSW	[ES:DI]←[DS:SI]（字节/字）	SI、DI	REP
CMPS / CMPSB / CMPSW	[DS:SI]-[ES:DI]，不回送，置标志	SI、DI	REPE/REPNE
SCAS / SCASB / SCASW	AL/AX - [ES:DI]，不回送，置标志	DI	REPE/REPNE
LODS / LODSB / LODSW	AL/AX←[DS:SI]	SI	一般不用
STOS / STOSB / STOSW	[ES:DI]←AL/AX	DI	REP

（课件 p.122–126；MOVS 用符号操作数时必须说明类型；SCAS 前关键字送 AL/AX、串首偏移送 DI、串长送 CX；LODS 一般不用重复前缀，STOS 常用 REP。）

重复前缀（课件 p.127–128）： | 前缀 | 重复条件 | 退出条件 | |---|---|---| | REP | CX≠0 | CX=0 | | REPE / REPZ | CX≠0 且 ZF=1 | CX=0 或 ZF=0 | | REPNE / REPNZ | CX≠0 且 ZF=0 | CX=0 或 ZF=1 |

每次重复 CX←CX-1；前缀不能单独使用，必须紧跟基本串指令；重复期间 SI/DI 修改但 **IP 仍指向带前缀的串指令**；**重复前缀本身不影响标志位**（ZF 由串指令产生）（课件 p.128）。

程序例（详见习题册题43–47）： - 100 字节传送三种写法：一般 MOV 循环 / MOVSB+DEC CX+JNZ / `CLD` + `REP MOVSB`（课件 p.129–130）。 - 两串比较：`REPZ CMPSB` 后 JZ 判全同（课件 p.131）。 - 查 '$'：`REPNZ SCASB`，找到后 `DEC DI` 回退到匹配位置（课件 p.132）。 - 20 个 0AAAAH 送 3000H 起：`CLD` / `LEA DI，[3000H]` / `MOV CX，14H` / `MOV AX，0AAAAH` / `REP STOSW`（课件 p.133；课件未设 ES，实际须使 ES 指向目标段）。 - 正负数分离：LODSB 取数 + `TEST AL，80H` 判符号 + STOSB 分存（负数区用 XCHG BX,DI 切换指针）（课件 p.134）。

易错警示：① 串指令的段约定是硬性的：源 DS:SI、目标 ES:DI，ES 不可超越；② DF 决定增减方向，用 STD 时指针须预置到串**末**元素；③ REP 配 MOVS/STOS，REPE/REPNE 配 CMPS/SCAS；④ SCAS 是「AL/AX 减串元素」，CMPS 是「源串减目标串」，方向别记反；⑤ 重复结束后 CX=0 与 ZF 状态要分开判断（CX=0 退出时 ZF 保留最后一次比较结果）。

8. 控制转移指令（课件 p.135–154）

转移通过修改 IP（段内）或 CS:IP（段间）实现（课件 p.137）。寻址方式（课件 p.135–136）： - 段内直接：给出以 IP 为基准的 8/16 位带符号补码偏移量； - 段间直接：直接给出目标段地址与段内偏移，分别取代 CS、IP； - 段内间接：目标偏移地址在寄存器或存储字单元中，取代 IP； - 段间间接：存储器连续 4 字节，低字→IP、高字→CS。

8.1 无条件转移 JMP（课件 p.138–146）

类型	格式	操作	范围
段内直接短	JMP SHORT OP	IP←IP+8 位偏移	-128~+127
段内直接近	JMP NEAR PTR OP	IP←IP+16 位偏移	-32768~+32767
段内间接	JMP OP / JMP WORD PTR OP	IP←EA（寄存器或字单元）	段内任意
段间直接	JMP FAR PTR OP	IP←OP 偏移，CS←OP 段址	全空间
段间间接	JMP DWORD PTR OP	低字→IP，高字→CS	全空间

（课件 p.140、144；偏移量相对「下一条指令的 IP」计算，正向转移为正、反向为负，课件 p.140–141。） 例：`JMP BX`（BX 内容为 EA）；`JMP WORD PTR [SI]`；`JMP FAR Label`；`JMP 2000H:1000H`；`JMP DWORD PTR [DI]`（课件 p.142、145–146）。

8.2 调用与返回 CALL/RET（课件 p.147–148）

- `CALL OP`：把**返回地址**（段内调用压 IP，段间调用压 CS 后再压 IP）压栈，SP 相应减 2/减 4，目标地址装入 IP（或 CS:IP）。（课件 p.147 操作过程文字中「当前 SP 压栈」应为「当前 IP 压栈」，课件原文如此，存疑。）
- `RET`：子程序最后一条指令，从栈顶弹出断点送 IP（段间再弹 CS），SP 相应加 2/加 4。

8.3 条件转移（课件 p.149–150）

类别	指令	条件
无符号比较	JA/JNBE	高于（CF=0 且 ZF=0）
	JAE/JNB	高于等于（CF=0）
	JB/JNAE	低于（CF=1）
	JBE/JNA	低于等于（CF=1 或 ZF=1）
带符号比较	JG/JNLE	大于（ZF=0 且 SF⊕OF=0）
	JGE/JNL	大于等于（SF⊕OF=0）
	JL/JNGE	小于（SF⊕OF=1）
	JLE/JNG	小于等于（ZF=1 或 SF⊕OF=1）
进位	JC / JNC	CF=1 / CF=0
零	JZ/JE / JNZ/JNE	ZF=1 / ZF=0
溢出	JO / JNO	OF=1 / OF=0
奇偶	JP/JPE / JNP/JPO	PF=1 / PF=0
符号	JS / JNS	SF=1 / SF=0

（条件转移均为短转移，范围 -128~+127，补充常识。） 例：100 个 8 位无符号数求最大存 MAX：`MOV BX,OFFSET DATA1` / `MOV AL,[BX]` / `MOV CX,63`（应为 63H=99）/ 循环 `INC BX` / `CMP AL,[BX]` / `JAE NEXT` / `MOV AL,[BX]` / `NEXT: DEC CX` / `JNZ P1` / `MOV MAX,AL`（课件 p.151）。

8.4 循环控制（课件 p.152–154）

指令	功能
LOOP 标号	CX←CX-1，CX≠0 则循环
LOOPZ/LOOPE 标号	CX←CX-1，CX≠0 且 ZF=1 则循环
LOOPNZ/LOOPNE 标号	CX←CX-1，CX≠0 且 ZF=0 循环
JCXZ 标号	CX=0 则转移（**不修改 CX**）

例4-59（课件 p.154）：10 字节数组中找 '$'，把 '$' 前所有数据求和送 AX：先用 `LOOPNZ` 扫描定位，`SUB SI,OFFSET BUF` 得 '$' 前字节数作计数，再用 `LOOP` 累加（`ADD AL,[SI]` / `ADC AH,0`）。完整分析见习题册题 49。

易错警示：① LOOP 先减 CX 再判断，CX 初值=循环次数；LOOP 类**不影响标志位**；② JCXZ 不减 CX，常用于循环前防 0 次；③ 段间间接转移的双字是「低字 IP、高字 CS」，与 LDS 的「低字寄存器、高字段址」同序；④ 无符号比较用 CF（JA/JB 族），带符号用 SF⊕OF（JG/JL 族），混用是高频错误。

9. 处理器控制指令（课件 p.155–158）

标志操作（只影响相关标志，课件 p.155）：

指令	操作	指令	操作
CLC	CF←0	CLD	DF←0
STC	CF←1	STD	DF←1
CMC	CF←CF 取反	CLI	IF←0（关中断）
STI	IF←1（开中断）		

CPU 控制（均不影响标志位，课件 p.156–158）：

指令	功能
HLT	处理器暂停（停机等待中断/复位）
WAIT	等待，直到 TEST 引脚信号有效才继续
ESC EXTOPCD，OP源	交权：向协处理器提供操作码和操作数
NOP	空操作，耗 3 个时钟周期
LOCK（前缀）	封锁总线，禁止其他设备访问总线，可加在任意指令前

10. 输入输出指令（课件 p.159）

指令	格式	功能
IN	IN 累加器，端口	端口字节/字→AL/AX
OUT	OUT 端口，累加器	AL/AX→端口

端口两种给法：8 位直接端口地址（00H–FFH，256 个）；DX 间接（16 位，64K 个）（课件 p.24、159）。累加器只能是 AL（字节）或 AX（字）。

11. 中断指令（课件 p.160–161）

指令	功能	标志
INTO	OF=1 时产生 4 号中断；OF=0 不起作用	检测 OF
INT n	软中断，n=00H–0FFH，转向 n 号中断服务程序	影响 IF、TF（清 0）
IRET	中断返回：弹 IP、CS、标志，回到断点	影响所有标志

12. 80X86 增强型指令（课件 p.162–172）

12.1 80286（课件 p.162–165）

- 增强：`PUSH 立即数` 合法；`IMUL 寄存器，立即数`、`IMUL 寄存器，存储器，立即数`（16 位带符号，积超 16 位高位丢失并 CF=OF=1）；移位计数值可为 **1~31 的立即数**（SHL/SAL/SAR/SHR/ROL/ROR/RCL/RCR 同）。
- 新增：`PUSHA/POPA`（压入顺序 AX、CX、DX、BX、SP、BP、SI、DI，弹出相反）；串 I/O `INS/OUTS`（INSB/OUTSB、INSW/OUTSW，端口在 DX，可加 REP）；`BOUND 16位寄存器，双字存储器`（检验边界）；`ENTER 16位立即数，8位立即数` / `LEAVE`（登录/撤销栈空间）。

12.2 80386（课件 p.166–170）

- 增强：`IMUL 寄存器，寄存器/存储器`（积存目标寄存器）；`CWDE`（AX→EAX 符号扩展）、`CDQ`（EAX→EDX:EAX）；串指令加后缀 D 成双字操作（MOVSD 等）；`PUSHFD/POPFD`（32 位标志）、`PUSHAD/POPAD`（32 位通用寄存器组；课件 p.167 将 POPAD 误写为 POPFD）。
- 新增：`MOVSX`（带符号扩展传送）、`MOVZX`（零扩展传送）；双精度右移 `SHRD 寄存器/存储器，寄存器，CL/8位立即数`；位操作 `BT/BTC/BTR/BTS`（指定位送 CF，然后分别：不变/取反/清 0/置 1。课件 p.169 把 BT 的功能误写成与 BTC 相同的「取反」，课件原文如此，存疑：BT 只测试不修改）；位扫描 `BSF/BSR`（从低位/高位扫描第一个 1，位号送目标寄存器，补充常识）；条件设置 `SET条件 寄存器/存储器`（按标志测试结果置 1/置 0，如 SETZ、SETNZ）。

12.3 80486（课件 p.171–172）

- `BSWAP 32位寄存器`：双字内字节高低交换（字节序反转）。
- `CMPXCHG 寄存器/存储器，寄存器`：比较并交换（补充常识：与累加器比较，相等则源→目的，否则目的→累加器）。
- `XADD 寄存器/存储器，寄存器`：交换并相加，和→目的，原目的→源。
- Cache/TLB 管理：`INVD`（作废 Cache）、`WBINVD`（写回并作废）、`INVLPG`（作废 TLB 项）。

易错警示：8086 与 80286+ 的合法指令范围不同（PUSH 立即数、移位立即数>1、三操作数 IMUL 都是 286 起才有），判断题若未说明机型按 8086 判。

13. 易错点与高频陷阱汇总

1. 立即数只能作源操作数，且不能直送段寄存器；`MOV DS，1234H` 错，须 `MOV AX，1234H` / `MOV DS，AX`。
2. CS、IP 不能作 MOV 目的；`POP CS` 错而 `PUSH CS` 对。
3. 存储器↔存储器、段寄存器↔段寄存器不能直接传送/交换，必须经寄存器中转。
4. 间址/基址/变址寄存器只有 BX、BP、SI、DI；`MOV AX，[CX]`、`[DX]`、`[SP]` 全错。
5. 默认段：BP→SS，BX/SI/DI→DS；串目标 ES:DI 不可段超越；忘记 BP 配 SS 是 PA 计算头号错误。
6. 字数据低地址字节送低 8 位（AL/DL…），看图做题先确认字节顺序。
7. 类型不明要加 PTR：`INC [DI]` 错、`INC BYTE PTR [DI]` 对；`ADD [BX]，28` 错（类型不明）。
8. INC/DEC 不影响 CF；NEG 操作数非 0 则 CF=1；对 80H/8000H 取补值不变但 OF=1。
9. MUL/IMUL 单操作数、源不能立即数、只影响 CF/OF；DIV/IDIV 商溢出→0 号中断；IDIV 余数符号随被除数。
10. CBW/CWD 不影响标志；AAD 在 DIV 前、AAM 在 MUL 后；DAA/DAS/AAA/AAS 必须紧跟加减指令且只调整 AL。
11. NOT 不影响标志；AND/OR/XOR/TEST 使 CF=OF=0；TEST 不回送结果。
12. 8086 移位次数>1 必须 CL；SAR 保符号、SHR 补 0；ROL/ROR 的 CF 不入环、RCL/RCR 的 CF 入环。
13. 堆栈按字操作、向低地址生长；PUSH 先高字节后低字节（高字节在高地址）；SP 始终指栈顶。
14. XLAT 只能用 BX+AL、表≤256 字节、在 DS 段、不影响标志。

15. LEA 取地址、MOV 取内容；`MOV SI，OFFSET X` 与 `LEA SI，X` 等价。
16. 串指令：源 DS:SI、目标 ES:DI；DF=0 增、DF=1 减；REP 配 MOVS/STOS，REPE/REPNE 配 CMPS/SCAS；前缀不影响标志。
17. 无符号比较看 CF（JA/JB），带符号看 SF⊕OF（JG/JL）；CMP 不改变操作数。
18. LOOP 先减后判、不影响标志；JCXZ 不减 CX；条件转移与 LOOP 均为短转移（补充常识）。
19. 段间间接转移/LDS 的双字顺序：低字→IP/寄存器，高字→CS/DS。
20. 80286 起才允许 PUSH 立即数、移位立即数 1~31、三操作数 IMUL；按 8086 判题时这些均错。

14. 典型例题精讲

例1（课件 p.26–27，寻址方式综合）：BX=0158H、DI=10A5H、位移量=1B57H、DS=2100H，求 7 种寻址方式的 EA 与 PA。 解：先背公式 EA=基址+变址+位移（按方式取舍），PA=DS×16+EA=21000H+EA。 ① 直接：EA=1B57H，PA=22B57H；② 间接(BX)：EA=0158H，PA=21158H；③ 相对基址：EA=0158H+1B57H=1CAFH，PA=22CAFH；④ 变址(DI)：EA=10A5H，PA=220A5H；⑤ 相对变址：EA=10A5H+1B57H=2BFCH，PA=23BFCH；⑥ 基址加变址：EA=0158H+10A5H=11FDH，PA=221FDH；⑦ 相对基址加变址：EA=11FDH+1B57H=2D54H，PA=23D54H。 教学要点：十六进制加法逐位算（8+7=F、5+5=A、1+B=C…），EA 超过 FFFFH 时按 16 位回卷（本题未超）。

例2（课件 p.42，堆栈交换）：AX=1122H、BX=3344H、SP=1010H，执行 PUSH AX / PUSH BX / POP AX / POP BX。 解：PUSH AX：SP=100EH，(100EH)=22H、(100FH)=11H；PUSH BX：SP=100CH，(100CH)=44H、(100DH)=33H；POP AX：AX←(100CH..100DH)=3344H，SP=100EH；POP BX：BX←(100EH..100FH)=1122H，SP=1010H。结果 AX=3344H、BX=1122H、SP 复原。 教学要点：后进先出使两次 POP 恰好交换两寄存器；SP 先减后存、先弹后加。

例3（课件 p.91，DAA 调整）：AL=68H、BL=59H（组合 BCD），ADD AL,BL 后 DAA。 解：68H+59H=C1H；低半字节 8+9=11H>0FH → AF=1。DAA 第一步：低 4 位=1 但 AF=1 → AL=C1H+06H=C7H，AF=1；第二步：高 4 位=CH>9 → AL=C7H+60H=127H → AL=27H，CF=1。结果 AL=27H、CF=1，即 68+59=127 的 BCD 表示（进位 1 代表百位）。 教学要点：DAA 的两步判断各自独立（低看 AF/低 4 位，高看 CF/高 4 位），加 6 与加 60H 可能同时发生。

例4（课件 p.116，存储器移位）：DS=7B00H、SI=1B79H、(7CB79H)=07H，MOV CL,3 / SAL [SI],CL。 解：PA=7B00H×16+1B79H=7B000H+1B79H=7CB79H，与给定单元吻合。07H=00000111，左移 3 次：00001110（CF=0）→00011100（CF=0）→00111000（CF=0）。结果 (7CB79H)=38H，CF=0（最后一次移出位）。 教学要点：移位作用于存储单元时先求 PA；CF 只保留**最后一次**移出位；SAL 与 SHL 完全相同。

附：本章速查表

EA/PA 公式：PA = 段基址×16 + EA；EA 组合 = 基址(BX/BP) + 变址(SI/DI) + 位移(8/16 位)，按寻址方式取舍；默认段 BP→SS、其余→DS、串目标→ES。

标志影响速查：

指令组	CF	OF	SF/ZF/PF	AF
MOV/LEA/LDS/LES/XCHG/XLAT/PUSH/POP/地址传送	不变	不变	不变	不变
ADD/ADC/SUB/SBB/CMP/NEG	影响	影响	影响	影响
INC/DEC	**不变**	影响	影响	影响
MUL/IMUL	影响	影响	无定义	无定义
DIV/IDIV	无定义	无定义	无定义	无定义
AND/OR/XOR/TEST	=0	=0	按结果	无意义
NOT	不变	不变	不变	不变
移位/循环移位	最后移出位	按规则	按结果	无定义
AAA/AAS	影响	无意义	无意义	影响
DAA/DAS	影响	无影响	影响	影响
AAM/AAD	无影响	无影响	影响	无影响
CLC/STC/CMC、CLD/STD、CLI/STI	只改各自标志			
串重复前缀、LOOP 类	不变	不变	不变	不变

条件转移速查：无符号 JA/JAE/JB/JBE 看 CF、ZF；带符号 JG/JGE/JL/JLE 看 SF⊕OF、ZF；单标志 JC/JNC、JZ/JNZ、JO/JNO、JS/JNS、JP/JNP。

BCD 调整速查：加后 AAA(非组合)/DAA(组合)；减后 AAS/DAS；乘后 AAM；除前 AAD；均只针对 AL（AAM/AAD 用 AX）。

串指令速查：MOVS 搬、CMPS 比、SCAS 搜、LODS 取、STOS 存；SI 源 DS、DI 目标 ES；DF 定方向、CX 定长度；REP 搬/存、REPE/REPNE 比/搜。