第1章 计算机系统概论

> 课件来源：唐朔飞《计算机组成原理》配套课件第1章（Slide 3–35，整页原图见 `../extracted/cod/slides/slide_NNN.png`），以课件提取文本 `extracted/cod/chapters/ch01.md` 为主、整页原图核对为辅。本章是全书的"地图"，概念题（选择/填空/简答）与 MAR/MDR 容量计算、MIPS/CPI 计算是期末必考点，难度低但覆盖面广，属于"背了就能拿分"的章节。

0. 本章考点清单

考点	常见题型	重要度
冯·诺依曼计算机六大特点、存储程序思想	选择/填空/简答	★★★
主要技术指标：机器字长、主频、MIPS、CPI、FLOPS、存储容量（MAR/MDR 换算）	选择/填空/计算	★★★
计算机系统层次结构 M0–M4、软硬件分界、翻译与解释	选择/填空/简答	★★★
运算器寄存器 ACC/MQ/X 在加减乘除中的分工	填空/选择/简答	★★★
主机完成一条指令的过程（取数/存数微操作序列）	简答/排序/综合	★★★
计算机体系结构 vs 计算机组成	选择/简答	★★
硬件框图：五大部件、以运算器/存储器为中心、现代框图（CPU/主机/I/O）	选择/填空	★★
存储器基本组成：存储体/存储单元/存储字/存储字长、MAR/MDR	填空/选择	★★
控制器基本组成：PC/IR/CU 功能、(PC)+1→PC	填空/选择	★★
软硬件概念、系统软件 vs 应用软件	选择	★★
指令格式（操作码+地址码）、ax^2+bx+c 程序清单	填空/综合	★★
计算机的工作步骤、编程举例（两种算法对比）	简答/选择	★
本书结构（四篇对应关系）	选择	★

1. 计算机的软硬件概念（Slide 4–5）

概念精讲：计算机系统 = 硬件 + 软件，二者缺一不可。

- **硬件**：计算机的实体，如主机、外设等（课件 p.4）。
- **软件**：由具有各类特殊功能的信息（程序）组成（课件 p.4）。

软件分类（课件 p.5）：

类别	定义	包含
系统软件	用来管理整个计算机系统	语言处理程序、操作系统、服务性程序、数据库管理系统、网络软件
应用软件	按任务需要编制成的各种程序	各类用户应用程序

要点： - 系统软件是"管机器的"，应用软件是"干活的"；操作系统是最核心的系统软件。 - 语言处理程序（编译/汇编/解释程序）属于系统软件，不要误归为应用软件。

易错警示：数据库管理系统、网络软件在课件中列于系统软件一侧（课件 p.5），不要凭直觉把它们当成应用软件。

2. 计算机的解题过程与系统层次结构（Slide 6–8）

解题过程（课件 p.6）：

```
高级语言程序 —翻译—> 目标程序 —运行—> 结果
```

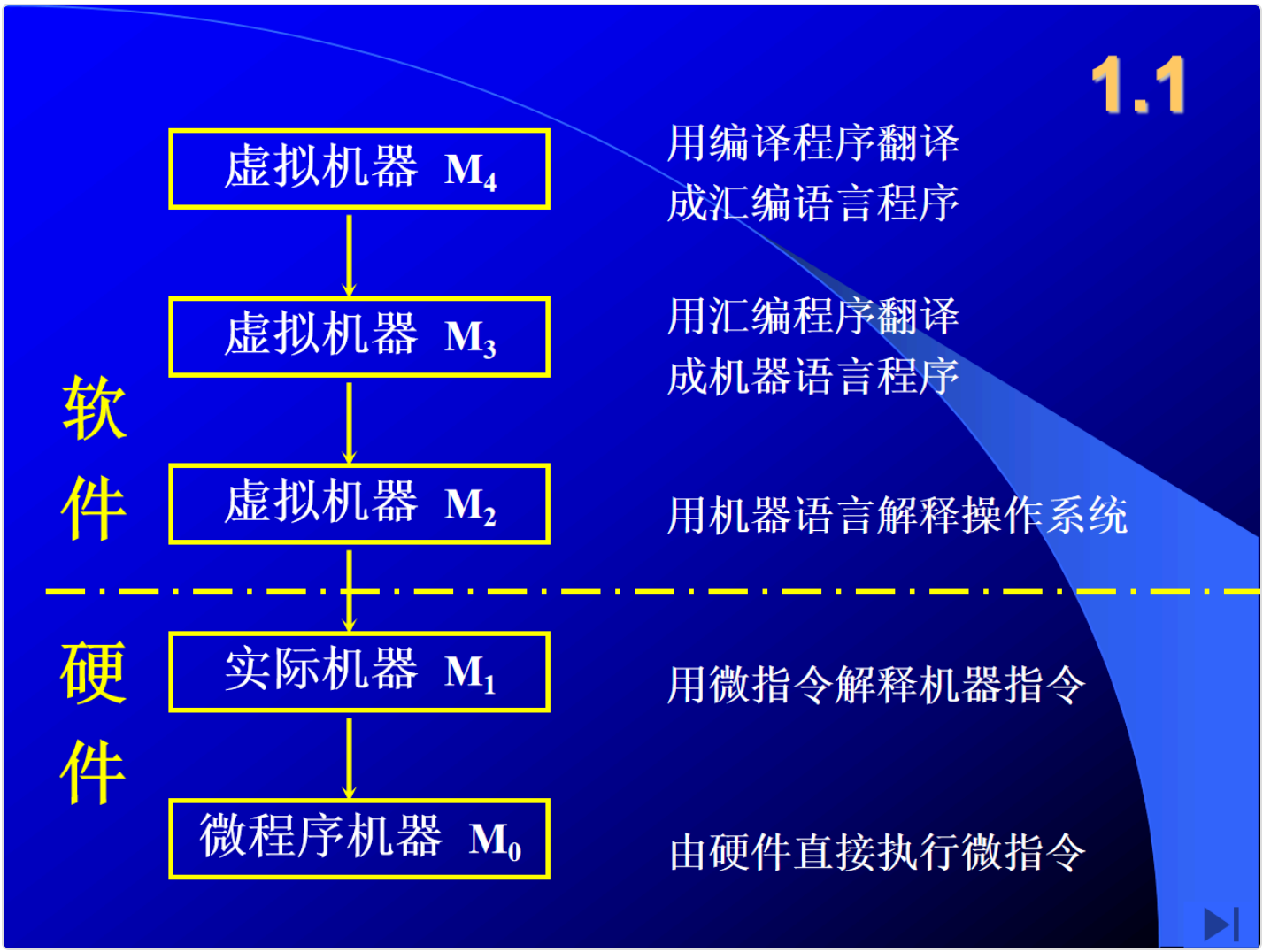
层次结构（课件 p.7、p.8）：自顶向下共 5 层，M4–M2 为虚拟机器，M1 为实际机器，M0 为微程序机器。

层次	名称	语言/系统	属性
M4	虚拟机器	高级语言	软件
M3	虚拟机器	汇编语言	软件
M2	虚拟机器	操作系统	软件
M1	实际机器	机器语言	硬件
M0	微程序机器	微指令系统	硬件

软硬件分界：在 M2 与 M1 之间（课件 p.8 的点划线）。M2 及以上为软件，M1 及以下为硬件。

层间翻译/解释关系（课件 p.8）：

相邻层	实现方式
M4 → M3	用编译程序翻译成汇编语言程序
M3 → M2	用汇编程序翻译成机器语言程序
M2 → M1	用机器语言解释操作系统
M1 → M0	用微指令解释机器指令
M0	由硬件直接执行微指令



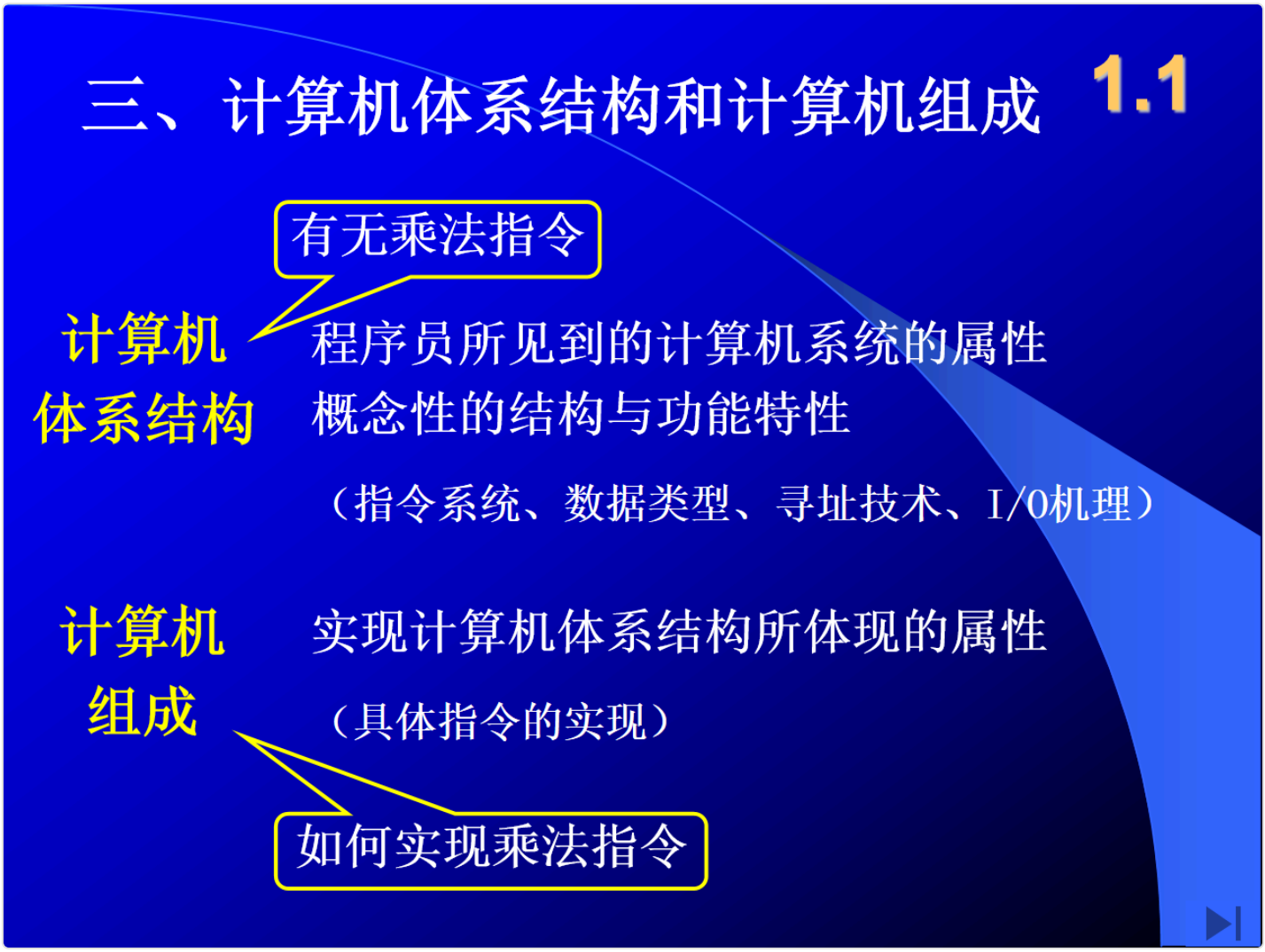
要点：- 上层是下层的"抽象/扩展"，下层是上层的"实现/支撑"；虚拟机器靠软件（翻译或解释）实现，实际机器与微程序机器靠硬件实现。- "翻译"= 先把整段程序换成下层语言再执行（编译、汇编）；"解释"= 逐条边译边执行（操作系统、微指令）。

易错警示：课件 p.7 把高级语言标为 M3、汇编标为 M2、操作系统未编号，与 p.8 的 M4/M3/M2 不一致。**课件原文如此，存疑：p.7 编号整体下移了一位；应以 p.8（与教材通行）为准：M4 高级语言、M3 汇编语言、M2 操作系统、M1 实际机器、M0 微程序机器。** 考试按 M4–M0 五层作答。

3. 计算机体系结构和计算机组成（Slide 9）

概念精讲（课件 p.9）：

概念	定义	关注内容	典型例子
计算机体系结构	程序员所见到的计算机系统的属性，概念性的结构与功能特性	指令系统、数据类型、寻址技术、I/O 机理	**有无**乘法指令
计算机组成	实现计算机体系结构所体现的属性	具体指令的实现	**如何实现**乘法指令



要点： - 体系结构是"做什么/有没有"（逻辑属性），组成是"怎么做"（物理实现）。 - 同一体系结构可以有多种组成实现（如乘法可用硬件乘法器或加法移位微程序实现）。

易错警示：判断归属时抓关键词——"有无/是否提供某指令、数据类型、寻址方式"属体系结构；"用什么器件/微操作/时序实现"属组成。

4. 冯•诺依曼计算机的特点（Slide 10）

六大特点（课件 p.10）：

1. 计算机由**五大部件**组成（运算器、控制器、存储器、输入设备、输出设备）。
2. 指令和数据以**同等地位**存于存储器，可**按地址寻访**。
3. 指令和数据用**二进制**表示。
4. 指令由**操作码和地址码**组成。
5. **存储程序**（程序事先存入存储器，启动后自动逐条取出执行）。
6. 以**运算器**为中心。

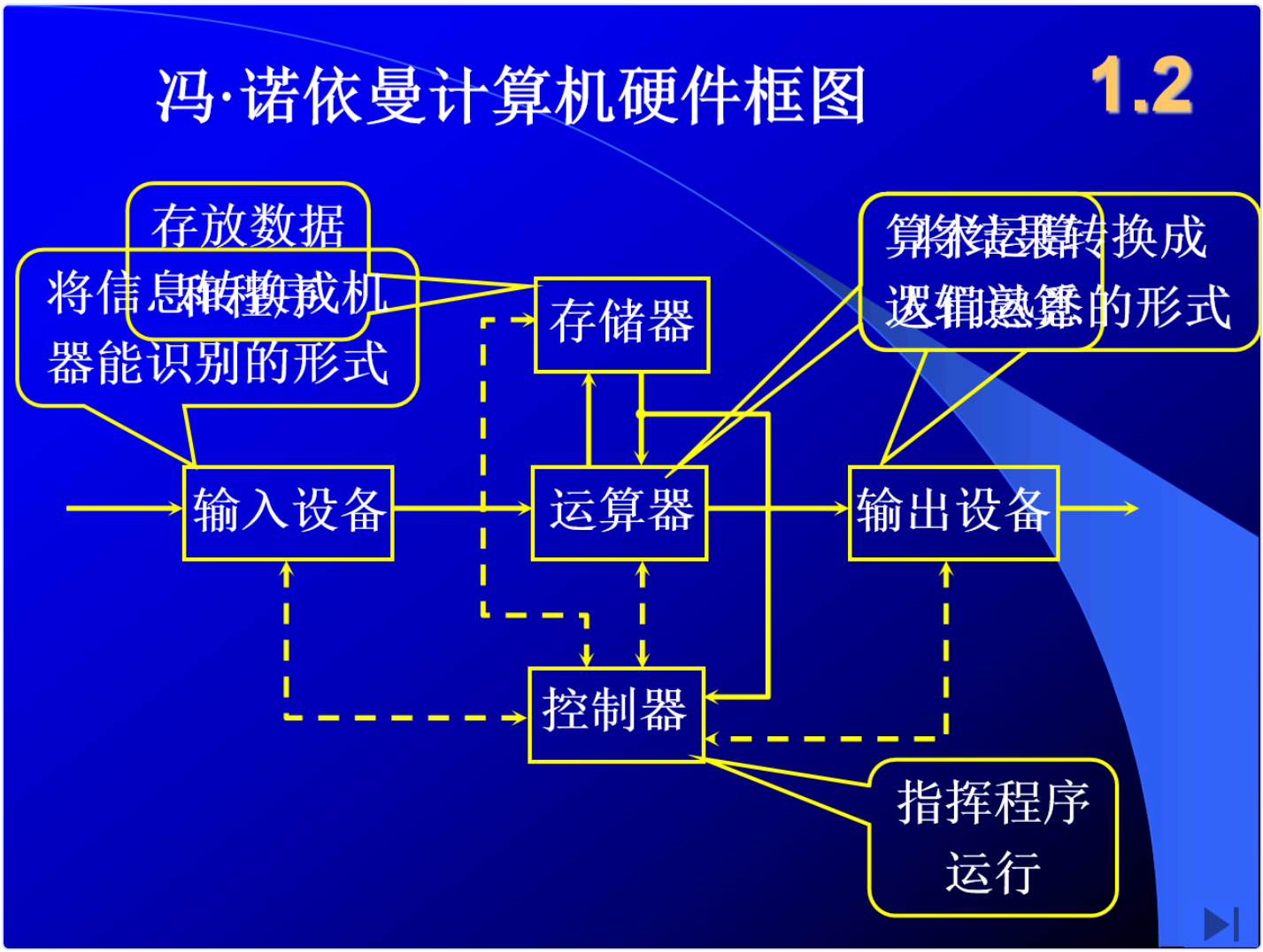
要点： - "存储程序"是冯·诺依曼思想的核心：程序与数据一样预先存放，机器自动连续执行，无需人工干预每一步。 - 经典冯·诺依曼机以**运算器**为中心；现代计算机改为以**存储器**为中心（见 §5）。

易错警示：课件 p.10 中"5. 存储程序"重复出现一次，为排版重复，特点仍为 6 条。不要把"以存储器为中心"当成冯·诺依曼原始特点（那是现代改进）。

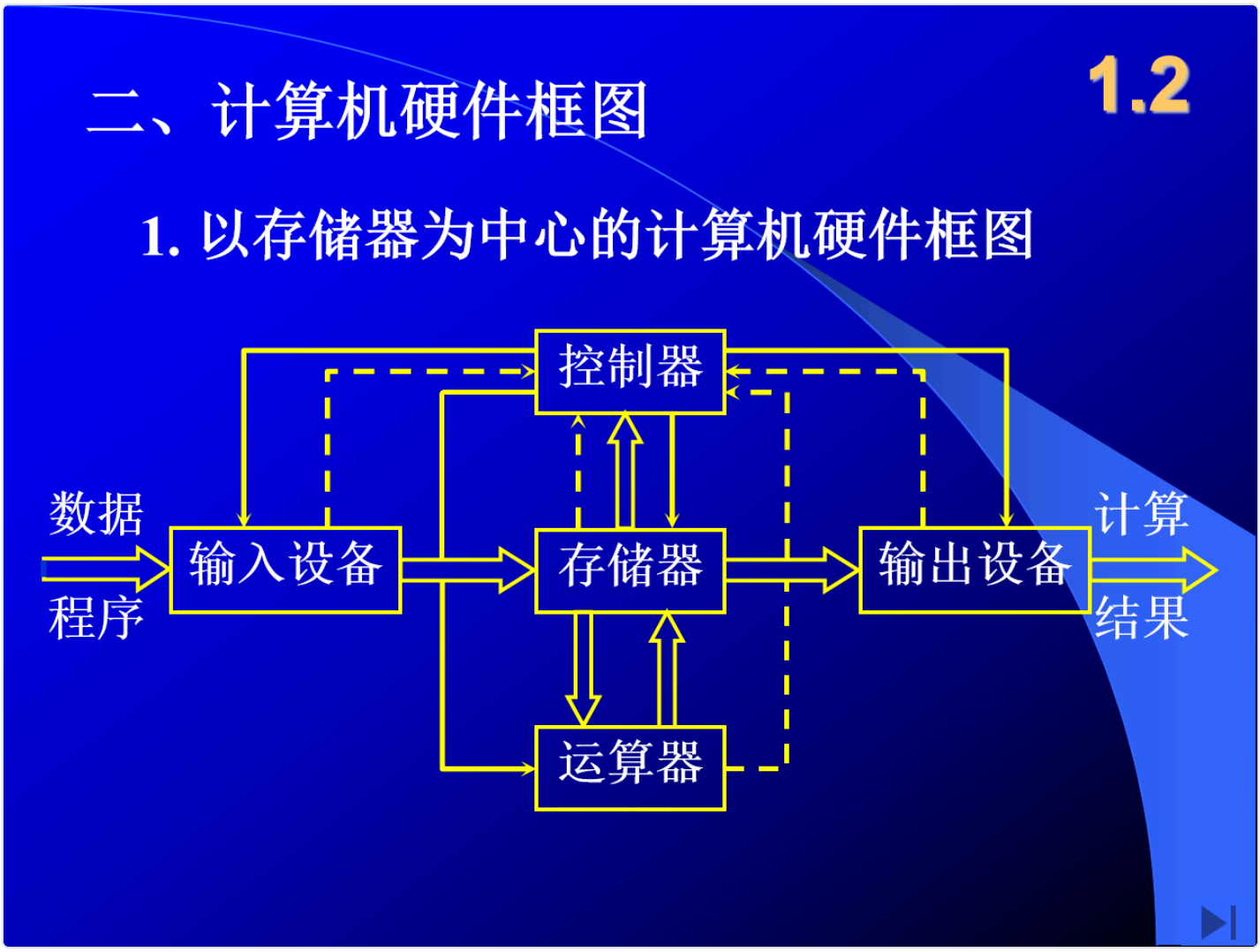
5. 计算机硬件框图（Slide 11–14）

冯•诺依曼硬件框图（以运算器为中心）（课件 p.11、p.12）：五大部件及功能——

部件	功能
存储器	存放数据和程序
输入设备	将信息转换成机器能识别的形式
运算器	算术运算、逻辑运算
控制器	指挥程序运行
输出设备	将结果转换成人们熟悉的形式

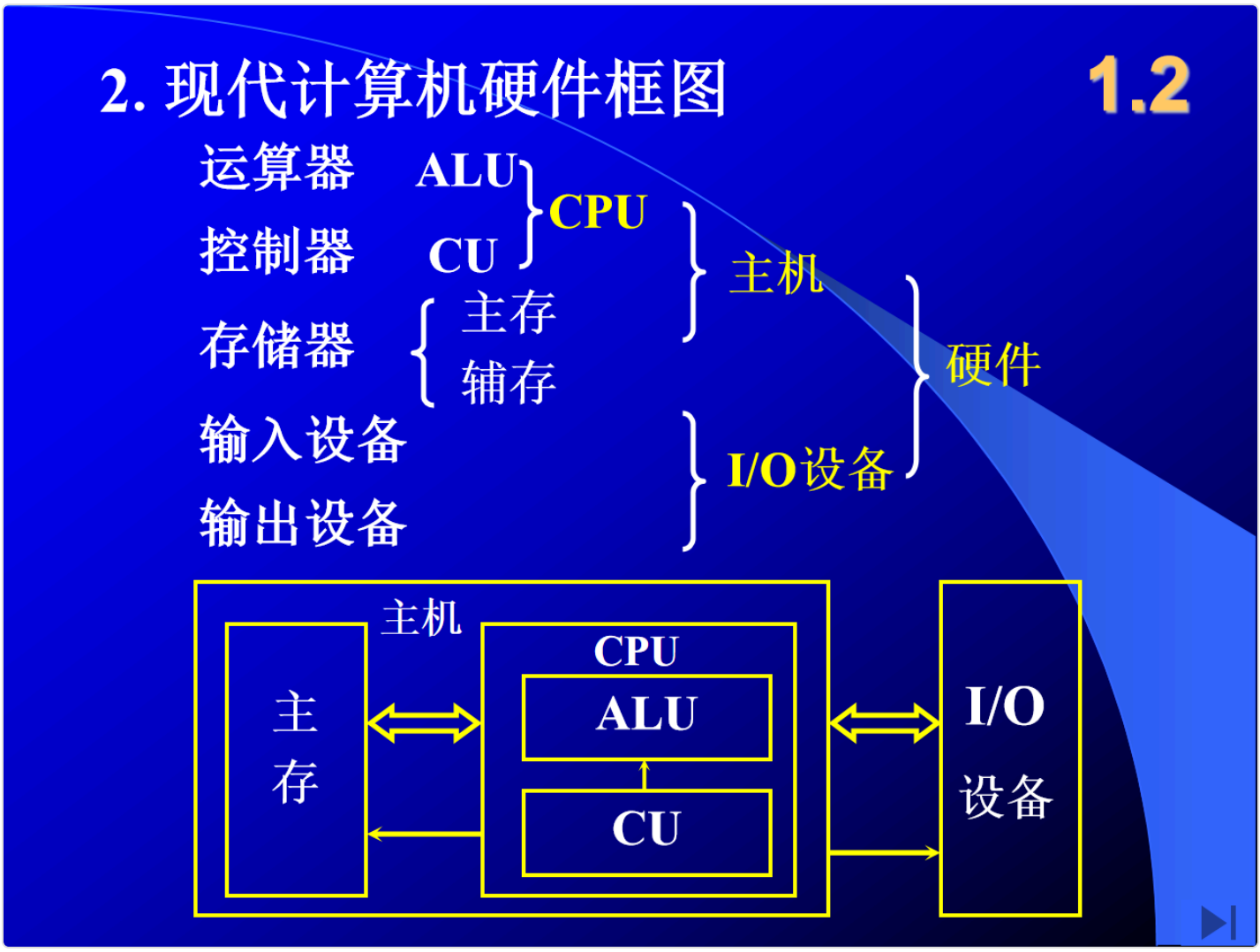


以存储器为中心的硬件框图（课件 p.13）：输入设备、输出设备、运算器都直接与存储器交换数据，控制器指挥各部件；数据/程序经输入设备入存储器，结果经存储器到输出设备。



现代计算机硬件框图（课件 p.14）：

```
运算器 ALU ┐
           ├ CPU ┐
控制器 CU  ┘     │
                 ├ 主机 ┐
存储器 主存 ─────┘      │
                        ├ 硬件
输入设备 ┐              │
         ├ I/O 设备 ────┘
输出设备 ┘
（存储器另含辅存；主机框内仅主存 + CPU）
```



要点：- CPU = 运算器(ALU) + 控制器(CU)；主机 = CPU + 主存；硬件 = 主机 + I/O 设备（课件 p.14）。- 辅存列于"存储器"下，但**不属于主机**（课件 p.14 主机框内只有主存与 CPU）；辅存通常视作外设（补充常识）。- 现代计算机以**存储器**为中心，克服了以运算器为中心时 I/O 与运算争用运算器的瓶颈。

易错警示：主机 ≠ CPU。主机 = CPU + 主存；CPU 只含 ALU + CU。辅存（硬盘等）不算主机。

6. 计算机的工作步骤与编程举例（Slide 15–18）

上机前的准备（课件 p.15）：建立数学模型 → 确定计算方法 → 编制解题程序。

- 建立数学模型例：$\sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \ldots$
- 确定计算方法例（迭代求根）：$\sqrt{x} = \frac{1}{2}\cdot(y_n + x/y_n)$（n = 0, 1, 2, …），即 $y_{n+1} = \frac{1}{2}\cdot(y_n + x/y_n)$ 迭代逼近 $\sqrt{x}$（课件 p.15）。
- **程序 = 运算的全部步骤；指令 = 每一个步骤**（课件 p.15）。

编程举例：计算 $ax^2 + bx + c$（课件 p.16），两种算法：

算法	步骤	步数
直接法	取x→乘x→乘a→存ax²→取b→乘x→加ax²→加c	8
秦九韶/霍纳法 (ax+b)x+c	取x→乘a→加b→乘x→加c	5

霍纳法把 ax²+bx+c 改写为 (ax+b)x+c，减少乘法与访存次数，效率更高。

指令格式举例（课件 p.17）：指令 = 操作码 + 地址码。

指令	操作码	地址码例	功能
取数	000001	α（0000001000）	[α] → ACC
存数	000010	β	[ACC] → β
加	000011	γ	[ACC]+[γ] → ACC
乘	000100	δ	[ACC]×[δ] → ACC
打印	000101	σ	[σ] → 打印机
停机	000110	（无）	停机

计算 ax²+bx+c 程序清单（课件 p.18，霍纳法）：

主存地址	操作码	地址码	注释
0	000001	0000001000	取数 x 至 ACC
1	000100	0000001001	乘 a 得 ax，存于 ACC
2	000011	0000001010	加 b 得 ax+b，存于 ACC
3	000100	0000001000	乘 x 得 (ax+b)x，存于 ACC
4	000011	0000001011	加 c 得 ax²+bx+c，存于 ACC
5	000010	0000001100	将结果存于主存单元 12
6	000101	0000001100	打印
7	000110	（空）	停机
8	x		原始数据 x
9	a		原始数据 a
10	b		原始数据 b
11	c		原始数据 c
12	（空）		存放结果

计算 ax^2+bx+c 程序清单 1.2

指令和数据存于主存单元的地址	指令		注释
	操作码	地址码	
0	000001	0000001000	取数x至ACC
1	000100	0000001001	乘a得ax,存于ACC中
2	000011	0000001010	加b得$ax+b$,存于ACC中
3	000100	0000001000	乘x得（$ax+b$)x,存于ACC中
4	000011	0000001011	加c得ax^2+bx+c,存于ACC
5	000010	0000001100	将ax^2+bx+c,存于主存单元
6	000101	0000001100	打印
7	000110		停机
8	x		原始数据x
9	a		原始数据a
10	b		原始数据b
11	c		原始数据c
12			存放结果

要点：地址码 10 位二进制对应十进制主存地址（0000001000=8、…、0000001100=12）；指令与数据混存于同一主存，体现"指令和数据同等地位、按地址寻访"。

易错警示：停机指令只有操作码、没有地址码；打印指令的地址码指向"存放结果的单元"（12），而非结果本身。

7. 存储器的基本组成（Slide 19–20）

概念精讲（课件 p.19）：

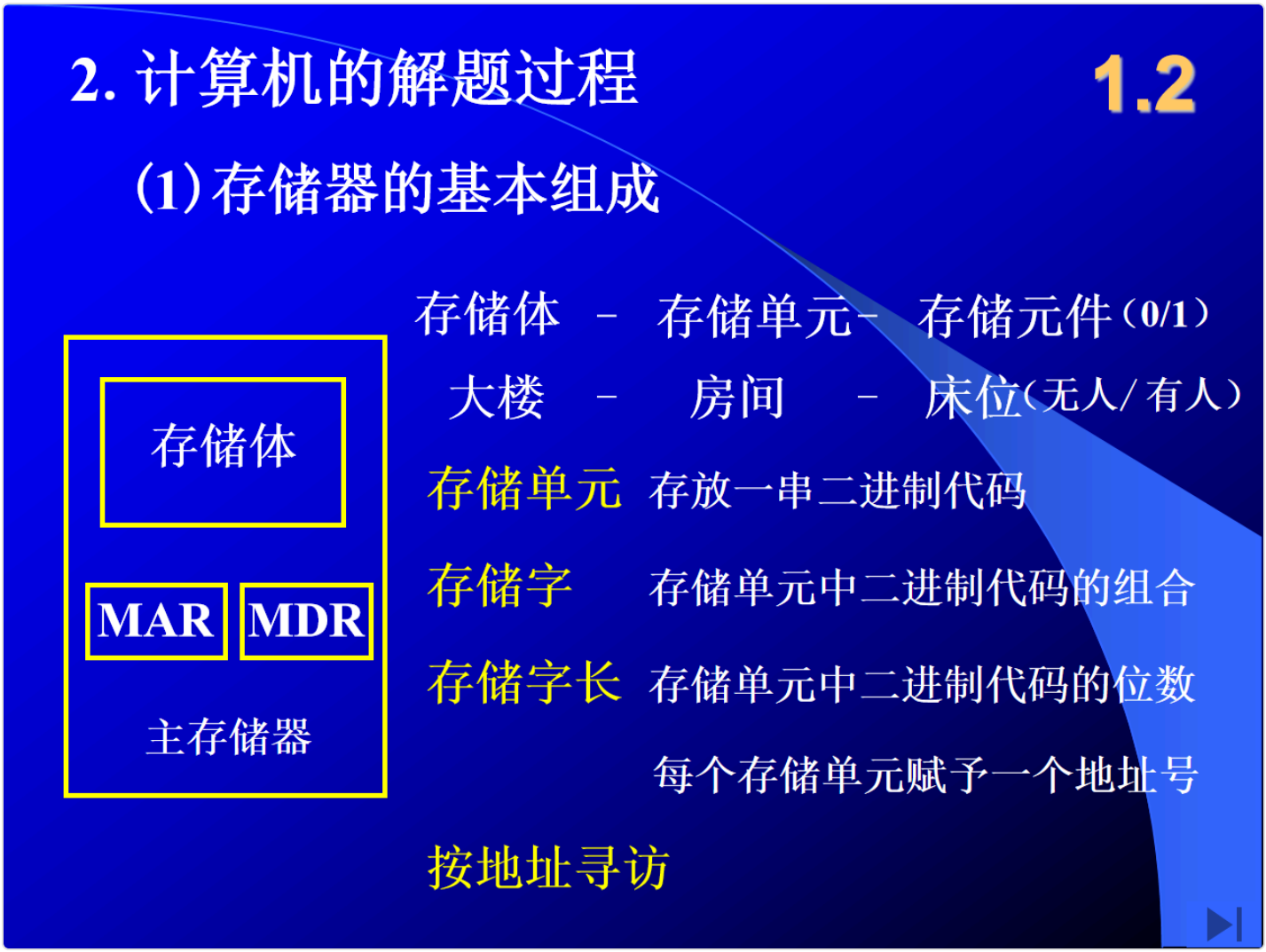
术语	定义	类比
存储体	由若干存储单元组成	大楼
存储单元	存放一串二进制代码	房间
存储元件	存 0/1 的最小单位	床位（无人/有人）
存储字	存储单元中二进制代码的组合	—
存储字长	存储单元中二进制代码的位数	—

每个存储单元赋予一个地址号，**按地址寻访**（课件 p.19）。

MAR 与 MDR（课件 p.20）：

寄存器	全称	反映
MAR	存储器地址寄存器	存储单元的个数（MAR 位数 → 2^MAR 个单元）
MDR	存储器数据寄存器	存储字长（MDR 位数 = 字长位数）

例：设 MAR = 4 位、MDR = 8 位 → 存储单元个数 = 2^4 = 16，存储字长 = 8（课件 p.20）。



要点：MAR 位数决定"有多少个房间"（地址空间），MDR 位数决定"每个房间住几位"（字长）。主存容量 = 2^MAR × MDR（位）。

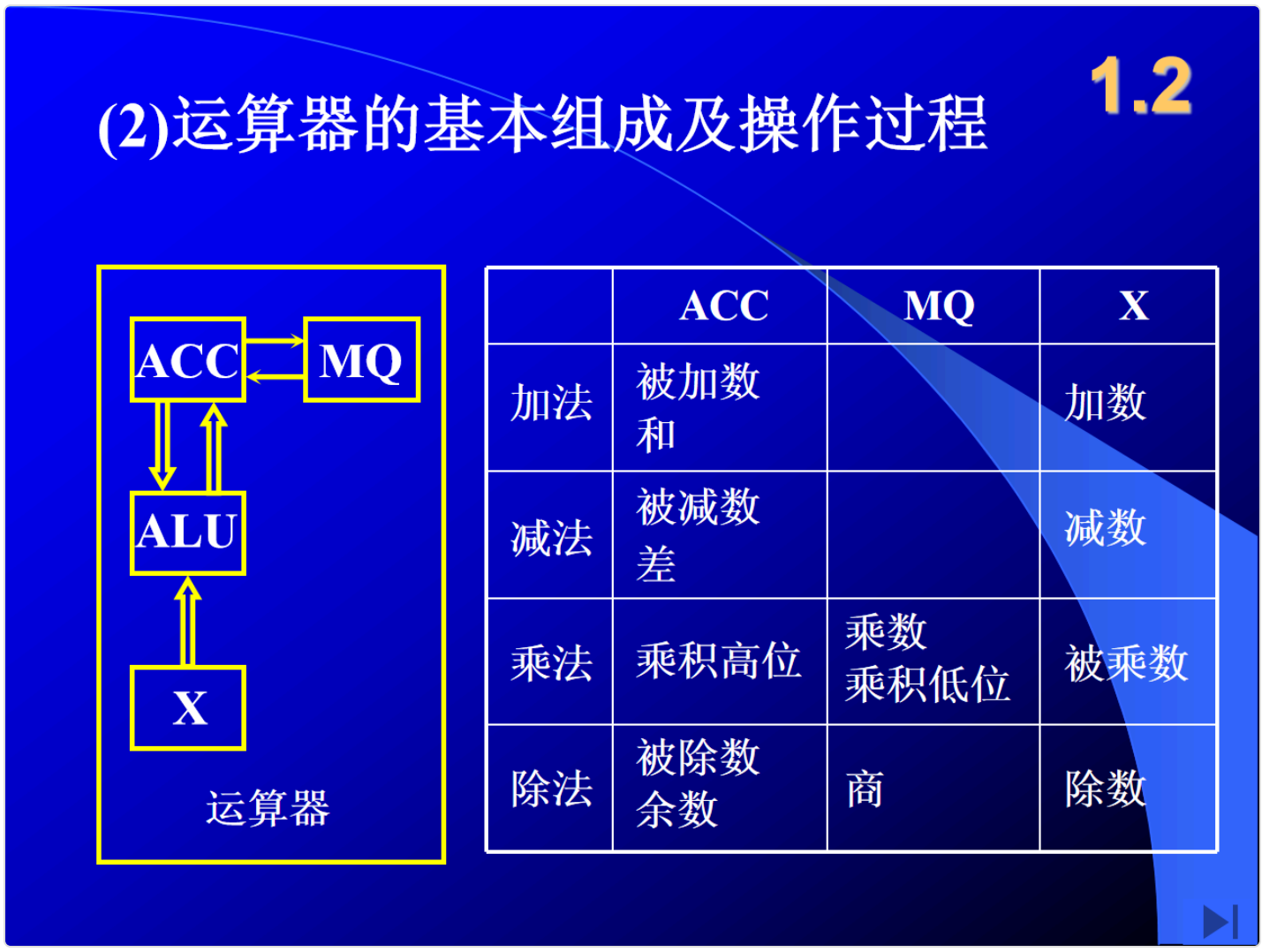
易错警示：MAR 反映的是单元**个数**（2 的幂），不是个数本身；MAR=4 位对应 16 个单元而非 4 个。

8. 运算器的基本组成及操作过程（Slide 21–25）

组成（课件 p.21）：ACC（累加器）、MQ（乘商寄存器）、X（操作数寄存器）、ALU（算术逻辑单元）。

各寄存器在四则运算中的分工（课件 p.21）：

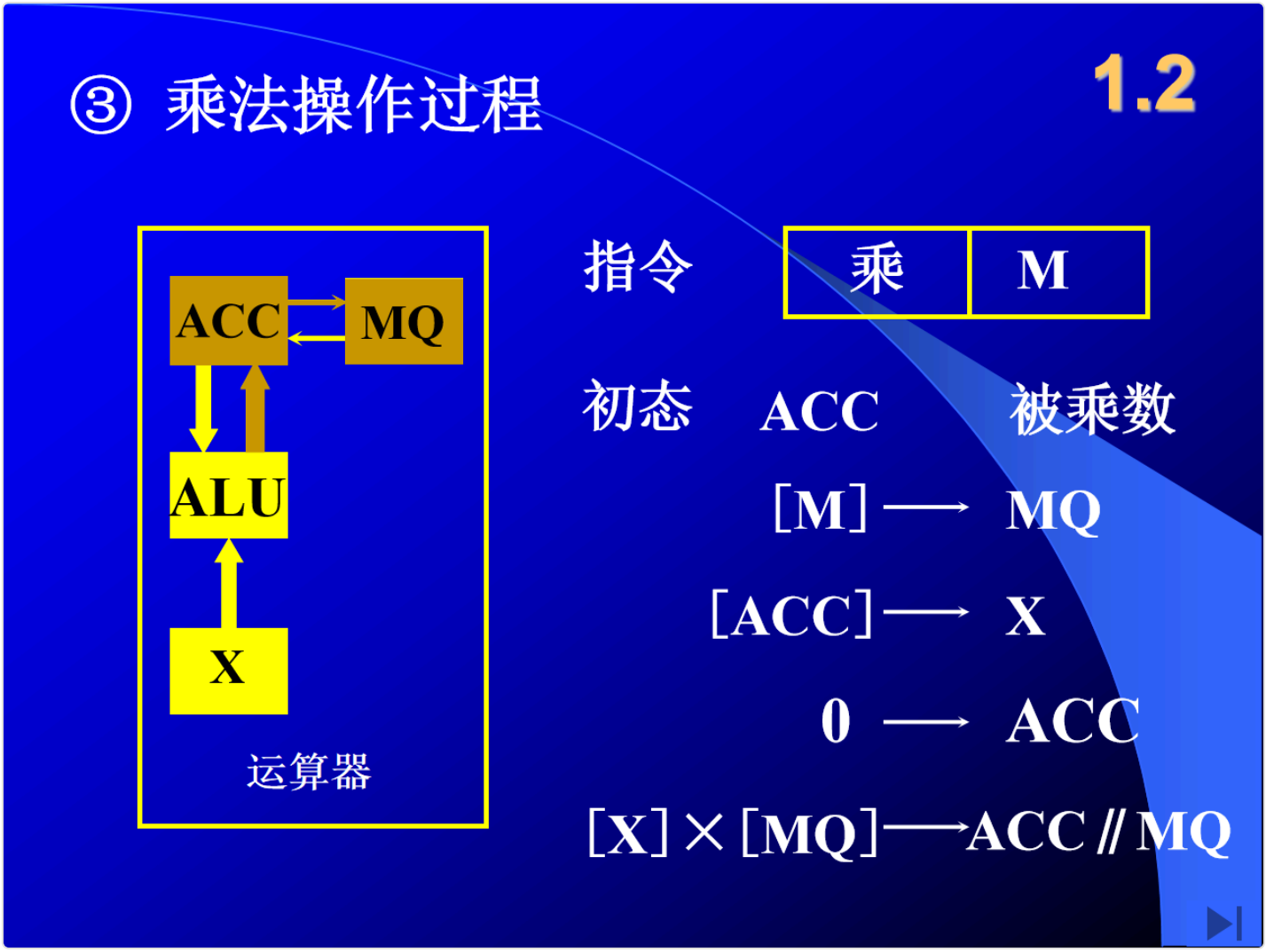
运算	ACC	MQ	X
加法	被加数 → 和	—	加数
减法	被减数 → 差	—	减数
乘法	乘积高位	乘数 → 乘积低位	被乘数
除法	被除数 → 余数	商	除数



操作过程（初态均先将被操作数送 ACC）：

- **加法**（课件 p.22）：初态 ACC=被加数；[M]→X；[ACC]+[X]→ACC（经 ALU）。
- **减法**（课件 p.23）：初态 ACC=被减数；[M]→X；[ACC]−[X]→ACC。

- **乘法**（课件 p.24）：初态 ACC=被乘数；[M]→MQ；[ACC]→X；0→ACC；[X]×[MQ]→ACC‖MQ（乘积高位在 ACC、低位在 MQ）。
- **除法**（课件 p.25）：初态 ACC=被除数；[M]→X；[ACC]÷[X]→MQ（商），余数在 ACC 中。



要点：加减法只用 ACC 与 X；乘法用 ACC、MQ、X 三个寄存器（结果双字长 ACC‖MQ）；除法商在 MQ、余数在 ACC。

易错警示：乘法开始前要把 ACC 清 0（0→ACC）并把被乘数转存到 X，否则被乘数会被覆盖；除法"商在 MQ、余数在 ACC"不要记反。

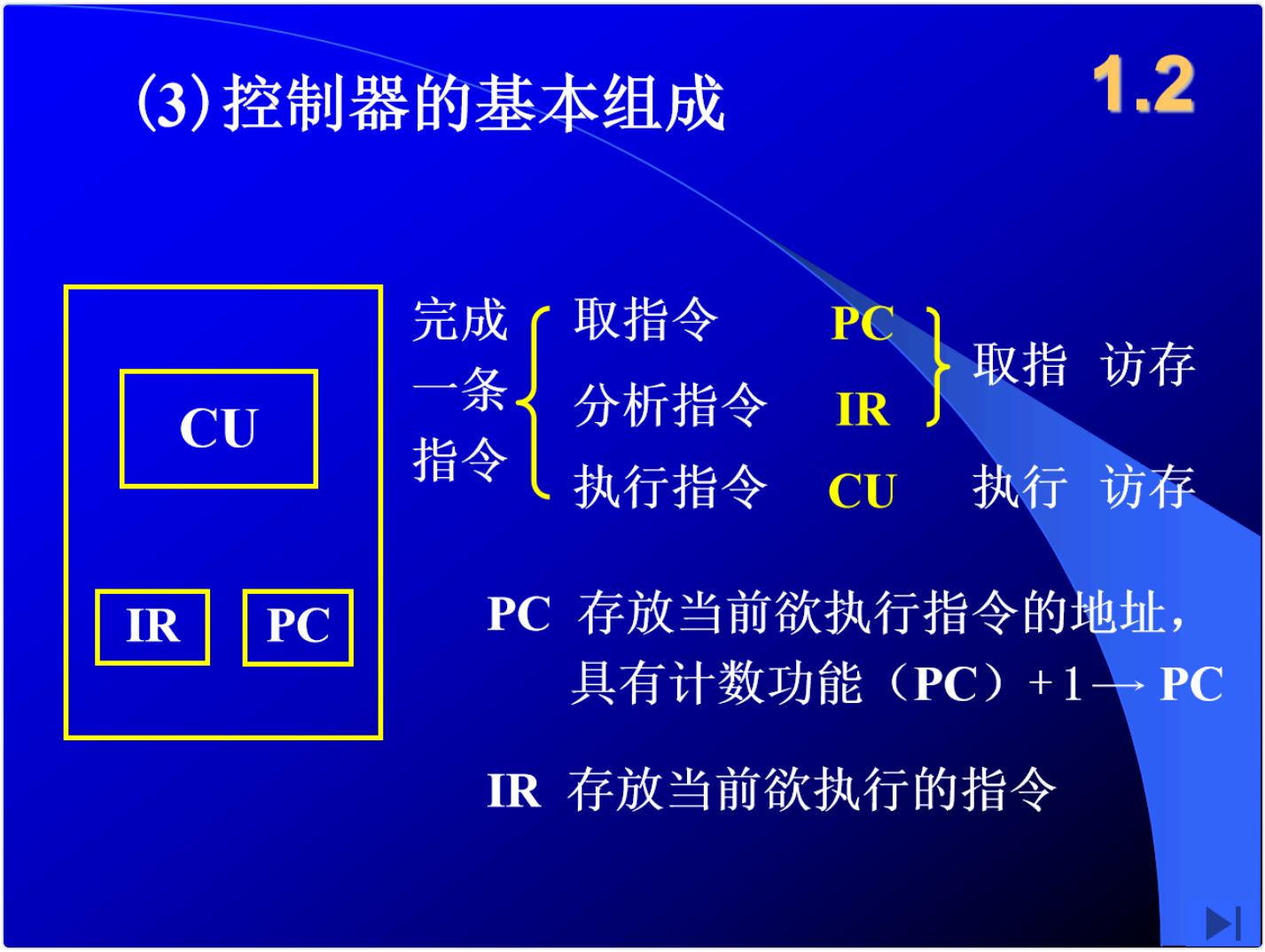
9. 控制器的基本组成（Slide 26）

组成（课件 p.26）：CU（控制单元）、IR（指令寄存器）、PC（程序计数器）。

完成一条指令的三阶段（课件 p.26）：

阶段	主要寄存器	是否访存
取指令	PC	访存
分析指令	IR	—
执行指令	CU	访存

- **PC**：存放当前欲执行指令的地址，具有计数功能 (PC)+1→PC（课件 p.26）。
- **IR**：存放当前欲执行的指令（课件 p.26）。
- **CU**：分析指令并发出各种控制命令，指挥全机协调工作。



要点：PC 指向"下一条要取的指令"，取指后自动 +1；IR 保存"正在执行的这条指令"。取指阶段靠 PC 访存，执行阶段靠 CU 访存。

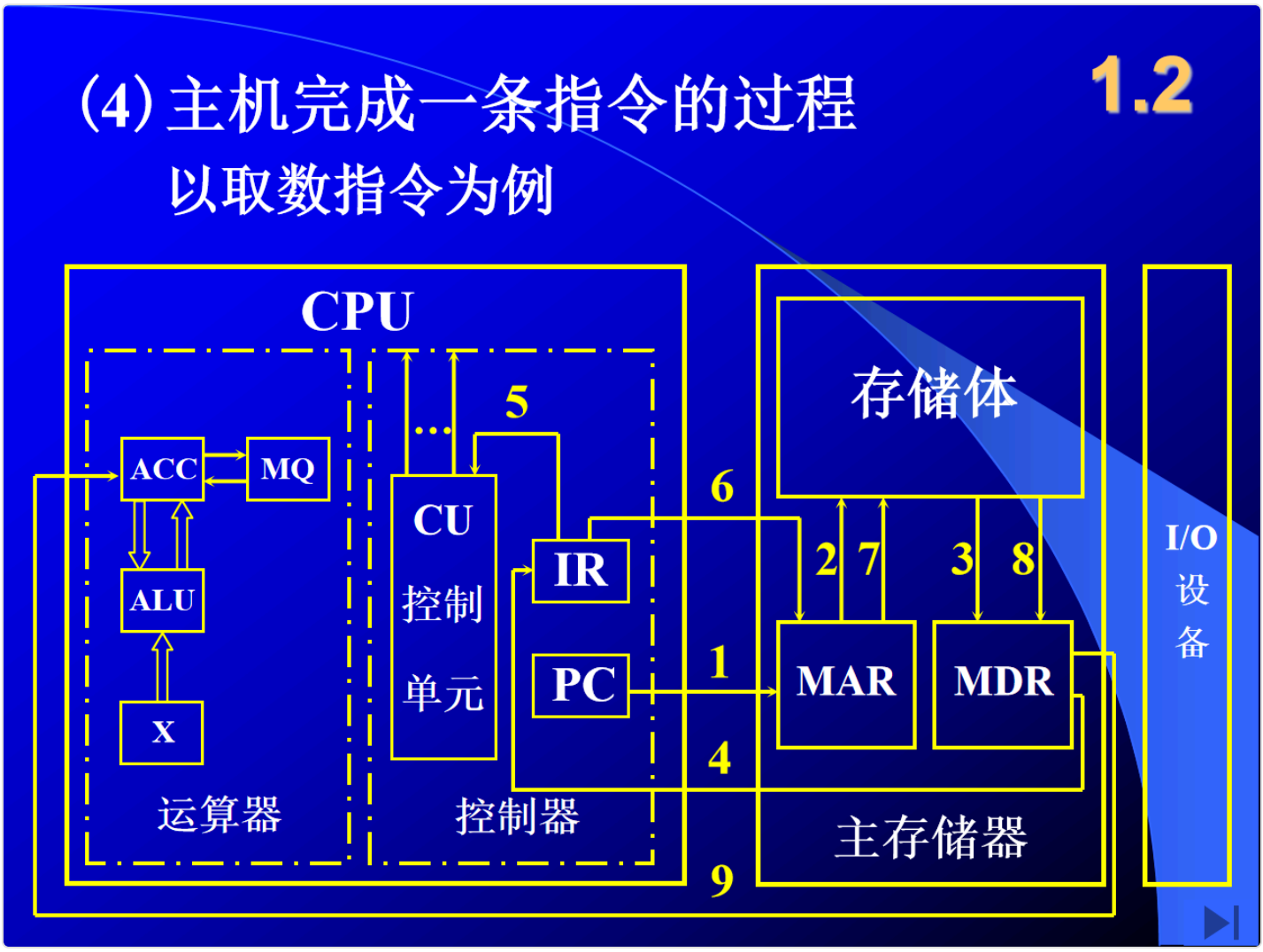
易错警示：PC 存的是**地址**不是指令；IR 存的是**指令**不是地址。(PC)+1 发生在取指阶段，不是执行阶段。

10. 主机完成一条指令的过程（Slide 27–28）

以取数指令为例（课件 p.27，配合 p.29 微操作）：

1. PC→MAR（送指令地址）
2. MAR→M（送地址、发读命令）
3. M→MDR（读出指令）
4. MDR→IR（指令送指令寄存器）
5. OP(IR)→CU（分析操作码）
6. Ad(IR)→MAR（送操作数地址）
7. MAR→M（送地址、发读命令）
8. M→MDR（读出操作数）
9. MDR→ACC（操作数送累加器）

取指阶段同时完成 (PC)+1→PC。



以存数指令为例（课件 p.28）：取指阶段 1–5 与取数相同；执行阶段：

6. Ad(IR)→MAR（送目的地址）
7. MAR→M（送地址、发写命令）
8. ACC→MDR（结果送数据寄存器）
9. MDR→M（写入存储体）

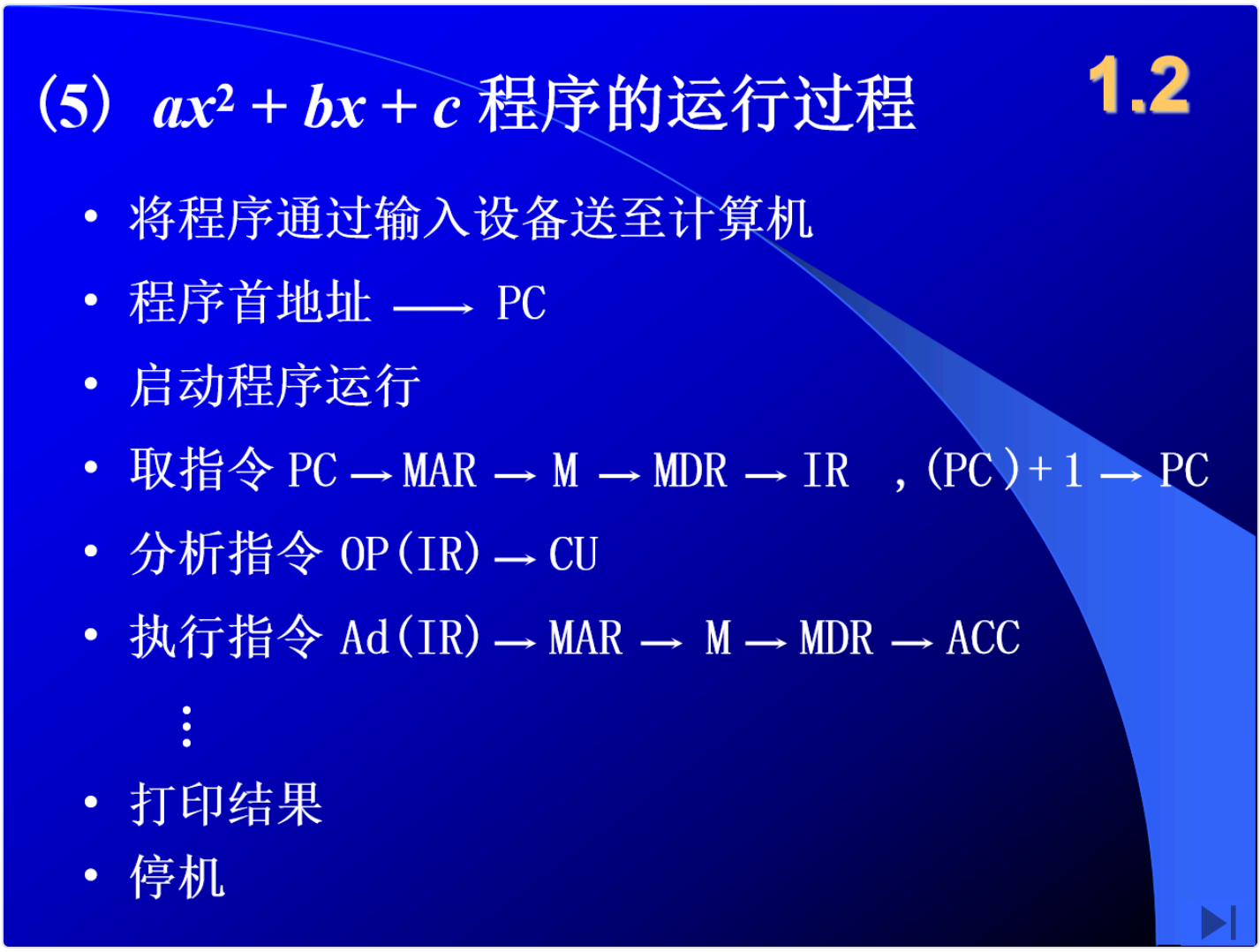
要点：取数与存数的取指阶段完全相同；区别在执行阶段的数据流向——取数是 M→MDR→ACC（读），存数是 ACC→MDR→M（写）。

易错警示：任何指令都要先"取指"（访存一次）再"执行"（可能再访存）；不要漏掉取指阶段的 MDR→IR 与 (PC)+1。

11. ax^2+bx+c 程序的运行过程（Slide 29）

运行流程（课件 p.29）：

1. 将程序通过输入设备送至计算机；
2. 程序首地址→PC；
3. 启动程序运行；
4. 取指令：PC→MAR→M→MDR→IR，(PC)+1→PC；
5. 分析指令：OP(IR)→CU；
6. 执行指令：Ad(IR)→MAR→M→MDR→ACC；
7. …（逐条循环 4–6）；
8. 打印结果；
9. 停机。



要点：程序运行的本质是"取指—分析—执行"的循环；PC 自动递增保证顺序执行，直到停机指令。

易错警示：启动前必须把**程序首地址**送 PC，否则机器不知道从哪条指令开始取。

12. 计算机硬件的主要技术指标（Slide 30–31）

1. 机器字长（课件 p.30）：CPU 一次能处理数据的位数，与 CPU 中的**寄存器位数**有关。

2. 运算速度（课件 p.30）：

指标	含义
主频	CPU 时钟频率（补充常识：时钟周期 T = 1/主频）
吉普森法	$T_m = \Sigma_{i=1}^{n} f \cdot t_i$（考虑指令混合比；补充常识：$f_i$ 为第 i 类指令使用频度/比例，t_i 为其执行时间）
MIPS	每秒执行百万条指令
CPI	执行一条指令所需时钟周期数
FLOPS	每秒浮点运算次数

补充常识（课件未列，衔接用）：MIPS = 主频 / (CPI × 10^6)；**吞吐量** = 单位时间内系统处理的请求/任务数或数据量（如 事务/秒、MB/s），反映系统整体处理能力。

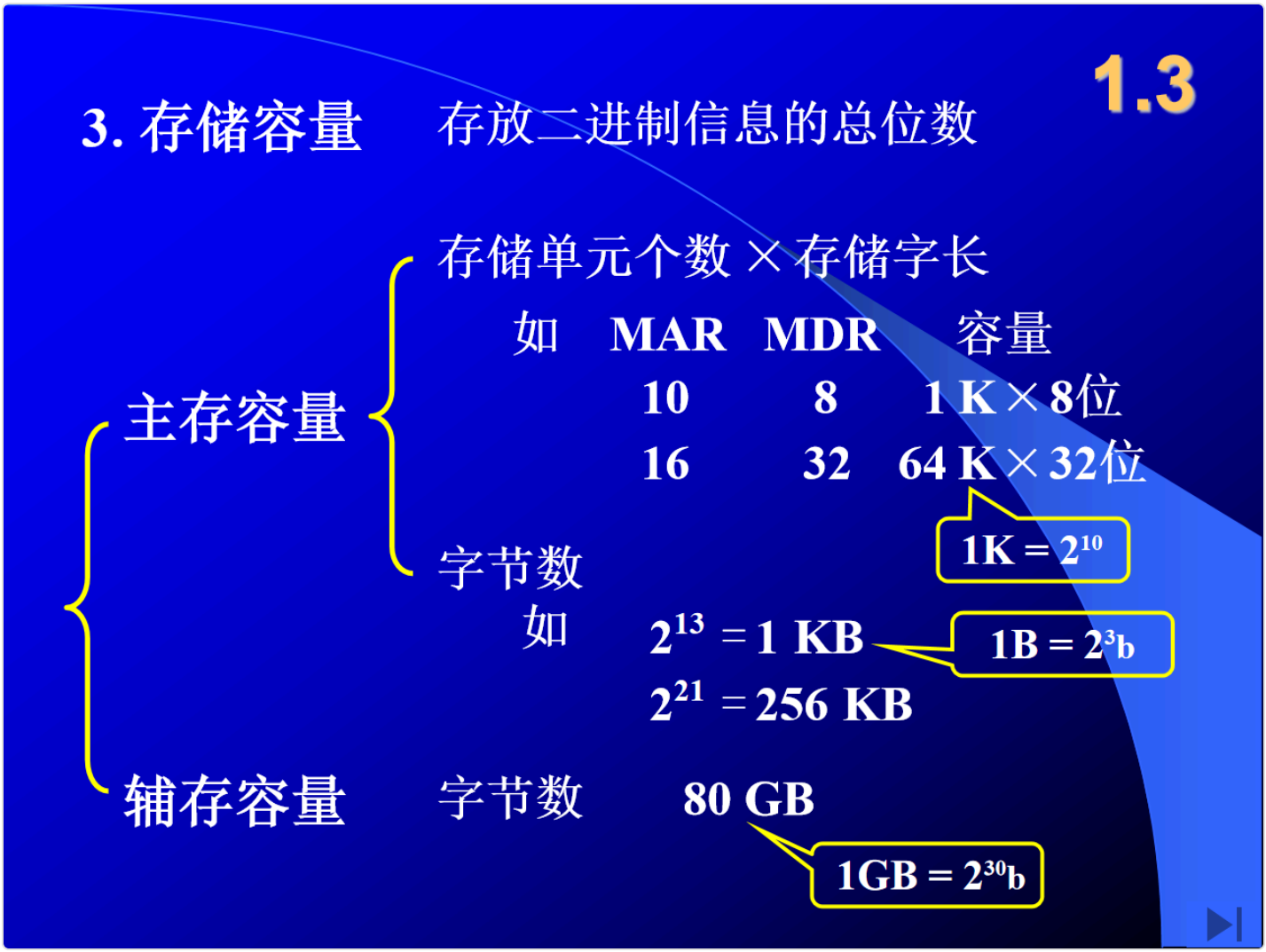


3. 存储容量（课件 p.31）：存放二进制信息的总位数。

- 主存容量 = 存储单元个数 × 存储字长 = 2^MAR位数 × MDR位数（位）；字节数 = 位数 / 8。
- 辅存容量：用字节数表示，如 80 GB。

MAR	MDR	容量
10	8	1K × 8 位
16	32	64K × 32 位

单位换算（课件 p.31）：1K = 2^{10}；1B = 2^3b；2^{13}b = 1KB；2^{21}b = 256KB。



要点：- 由 MAR/MDR 求容量：个数 = 2^MAR，字长 = MDR，容量(位) = 2^MAR × MDR，容量(字节) = 2^MAR × MDR / 8。- 2^{13}b = $2^{13}/2^3$ B = 2^{10} B = 1KB；2^{21}b = $2^{21}/2^3$ B = 2^{18} B = 256KB（课件 p.31 的 2^{13}、2^{21} 指**位数**）。

易错警示：**课件原文如此，存疑：p.31 标注"1GB = 2^{30}b"。按 1B = 2^3b，应为 1GB = 2^{30}B = 2^{33}b；课件此处 b 疑为 B 之误。** 考试若遇 1GB 换算，按 1GB = 2^{30}B = 2^{33}b 计算，并注意与课件 2^{13}b=1KB 的"位→字节"口径保持一致。

13. 本书结构（Slide 32–35）

篇	内容	对应部件
第1篇 概论	计算机系统 overview（本章）	计算机整体
第2篇 计算机系统的硬件结构	存储器、系统总线、I/O、CPU	系统级互连
第3篇 CPU	ALU、CU、寄存器、内部互连	中央处理器
第4篇 CU	控制存储器、排队逻辑、控制、寄存器和解码器	控制单元

（课件 p.32–35。本篇为后续章节的导航：第2篇讲"部件怎么连"，第3篇讲"CPU 内部"，第4篇讲"CU 内部"。）

14. 易错点与高频陷阱汇总

1. 主机 ≠ CPU：主机 = CPU + 主存；CPU = ALU + CU；辅存不属于主机（课件 p.14）。
2. 冯·诺依曼原始特点是"以运算器为中心"，"以存储器为中心"是现代改进（课件 p.10、p.13）。
3. 层次结构编号以 p.8 为准：M4 高级、M3 汇编、M2 操作系统、M1 实际、M0 微程序；p.7 编号下移一位属课件笔误。
4. 软硬件分界在 M1 与 M2 之间，不是 M0 与 M1 之间（课件 p.8）。
5. "有无乘法指令"=体系结构；"如何实现乘法指令"=组成（课件 p.9）。
6. MAR 位数 → 单元个数 = 2^MAR（MAR=4 → 16 个），MDR 位数 = 存储字长（课件 p.20）。
7. PC 存地址、IR 存指令；(PC)+1→PC 在取指阶段（课件 p.26）。
8. 乘法：商/积低位在 MQ、积高位在 ACC；除法：商在 MQ、余数在 ACC；乘法前需 0→ACC（课件 p.21、p.24、p.25）。
9. 取数与存数取指阶段相同，执行阶段数据流向相反（读 M→MDR→ACC vs 写 ACC→MDR→M）（课件 p.27、p.28）。
10. 容量单位口径：课件 2^{13}=1KB、2^{21}=256KB 指位数；1GB=2^{30}b 为课件笔误，应为 2^{30}B=2^{33}b（课件 p.31）。
11. 停机指令无地址码；打印指令地址码指向结果单元（课件 p.17、p.18）。
12. 系统软件含语言处理程序、操作系统、服务性程序、数据库管理系统、网络软件（课件 p.5）。

15. 典型例题精讲

例1（层次结构，选择）：在计算机系统层次结构中，操作系统虚拟机器是（ ）。 A. M1 B. M2 C. M3 D. M4 **解析**：自顶向下 M4 高级语言、M3 汇编语言、M2 操作系统、M1 实际机器、M0 微程序机器（课件 p.8）。操作系统层 = M2。逐项排除：M1 是机器语言实际机器、M3 是汇编、M4 是高级语言。**答案 B**。

例2（MAR/MDR 容量，计算）：某机 MAR = 16 位、MDR = 32 位，求主存容量（位与字节）。 **解析**： - 存储单元个数 = 2^MAR = 2^{16} = 64K； - 存储字长 = MDR = 32 位； - 容量(位) = 64K × 32 位 = 2^{16} × 32 = 2^{21}

位； - 容量(字节) = 2^{21} / 2^3 = 2^{18} B = 256KB。 与课件 p.31 表中"16 / 32 → 64K×32位"及"2^{21}=256KB"一致。**答案：64K×32 位 = 256KB**。

例3（取数指令微操作，排序/简答）：写出取数指令执行阶段的微操作序列，并说明 (PC)+1 发生在哪个阶段。 **解析**：取指阶段：PC→MAR；MAR→M(读)；M→MDR；MDR→IR；同时 (PC)+1→PC。分析：OP(IR)→CU。执行阶段：Ad(IR)→MAR；MAR→M(读)；M→MDR；MDR→ACC（课件 p.27、p.29）。(PC)+1 发生在**取指阶段**。**答案**：执行阶段为 Ad(IR)→MAR→M→MDR→ACC；(PC)+1 在取指阶段。

附：本章速查表

层次结构：M4 高级语言 / M3 汇编语言 / M2 操作系统 / M1 实际机器(机器语言) / M0 微程序机器(微指令)；软硬件分界 M1|M2。

冯•诺依曼六特点：五大部件；指令数据同等地位按地址寻访；二进制；操作码+地址码；存储程序；以运算器为中心。

组成关系：CPU=ALU+CU；主机=CPU+主存；硬件=主机+I/O设备；存储器=主存+辅存。

寄存器分工：加法 ACC(被加数/和)+X(加数)；减法 ACC(被减数/差)+X(减数)；乘法 ACC(积高)+MQ(乘数/积低)+X(被乘数)；除法 ACC(被除数/余数)+MQ(商)+X(除数)。

控制器：PC 存地址、(PC)+1→PC；IR 存指令；CU 发控制命令；取指(PC,访存)→分析(IR)→执行(CU,访存)。

取数微操作：PC→MAR→M→MDR→IR；OP(IR)→CU；Ad(IR)→MAR→M→MDR→ACC。

公式： - 主存容量(位) = 2^MAR × MDR；容量(字节) = 2^MAR × MDR / 8 - 时钟周期 = 1/主频；MIPS = 主频 / (CPI × 10^6) - 吉普森法 $T_m = \Sigma_{i=1}^{n} f_i \cdot t_i$ - 1K = 2^{10}；1B = 2^3b；1KB = 2^{10}B = 2^{13}b；1MB = 2^{20}B；1GB = 2^{30}B = 2^{33}b

opcode 表（p.18）：000001 取数 / 000010 存数 / 000011 加 / 000100 乘 / 000101 打印 / 000110 停机。

技术指标：机器字长=CPU 一次处理位数(与寄存器位数有关)；MIPS=每秒百万条指令；CPI=每条指令时钟周期数；FLOPS=每秒浮点运算次数；吞吐量=单位时间处理任务/数据量（补充常识）。