

知识点回顾与重要考点

I/O设备的基本概念与分类

什么是I/O设备

将数据 Input/Output (输入/输出) 计算机的外部设备

按使用特性分类

人机交互类外部设备

存储设备

网络通信设备

按传输速率分类

低速设备

中速设备

高速设备

按信息交换的单位分类

块设备 (传输快, 可寻址)

字符设备 (传输慢, 不可寻址, 常采用中断驱动方式)

本节内容

I/O控制器

南北mooc147

I/O设备的电子部件（I/O控制器）

CPU无法直接控制I/O设备的机械部件，因此I/O设备还要有一个电子部件作为CPU和I/O设备机械部件之间的“中介”，用于实现CPU对设备的控制。

这个电子部件就是**I/O控制器**，又称**设备控制器**。CPU可控制I/O控制器，又由I/O控制器来控制设备的机械部件。

I/O控制器的功能

接受和识别CPU发出的命令

向CPU报告设备的状态

数据交换

地址识别

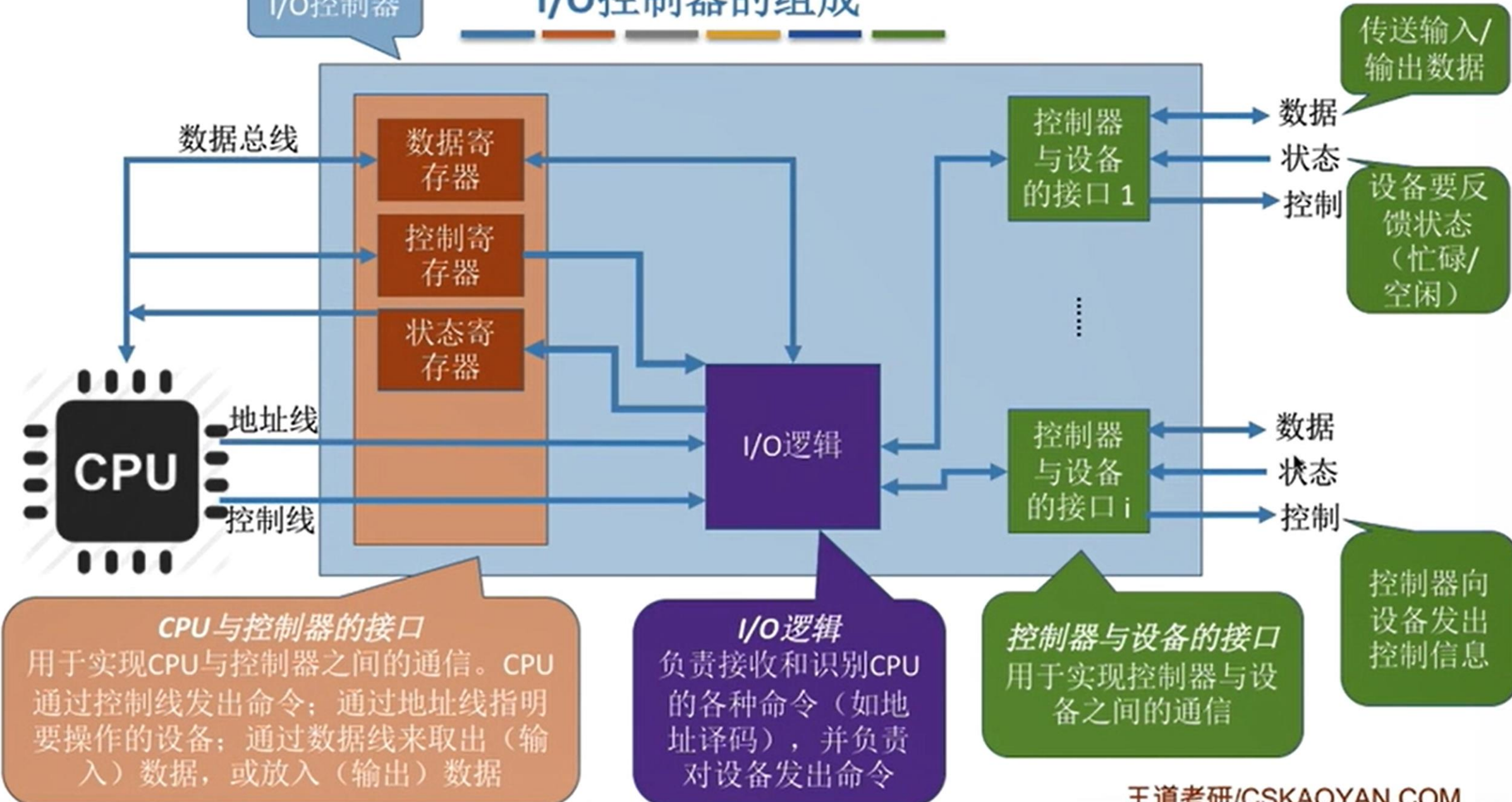
如CPU发来的 read/write 命令，I/O 控制器中会有相应的**控制寄存器**来存放命令和参数

I/O控制器中会有相应的**状态寄存器**，用于记录I/O设备的当前状态。如：1表示空闲，0表示忙碌

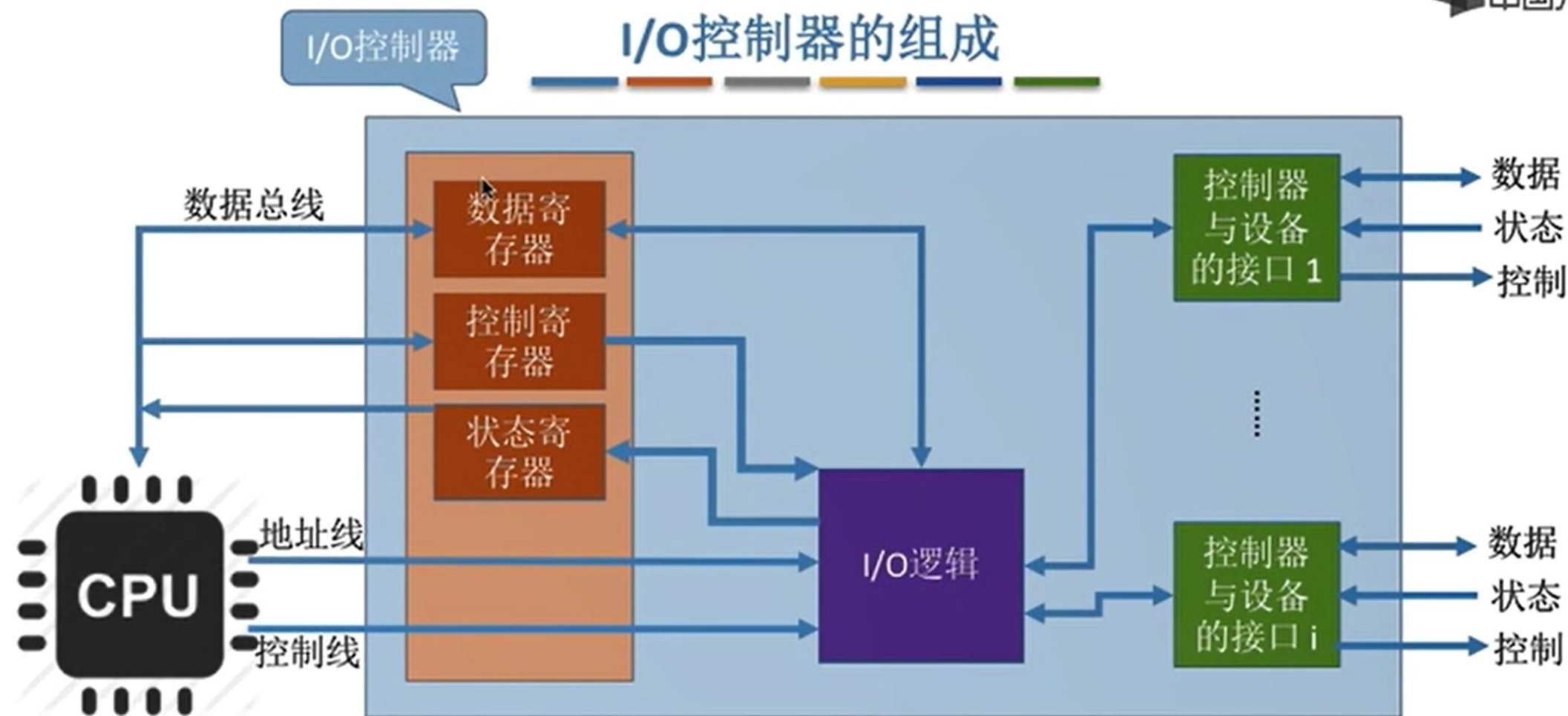
I/O控制器中会设置相应的**数据寄存器**。输出时，数据寄存器用于暂存CPU发来的数据，之后再由控制器传送设备。输入时，数据寄存器用于暂存设备发来的数据，之后CPU从数据寄存器中取走数据。

类似于内存的地址，为了区分设备控制器中的各个寄存器，也需要给各个寄存器设置一个特定的“地址”。I/O控制器通过CPU提供的“地址”来判断CPU要读/写的是哪个寄存器

I/O控制器的组成



I/O控制器的组成



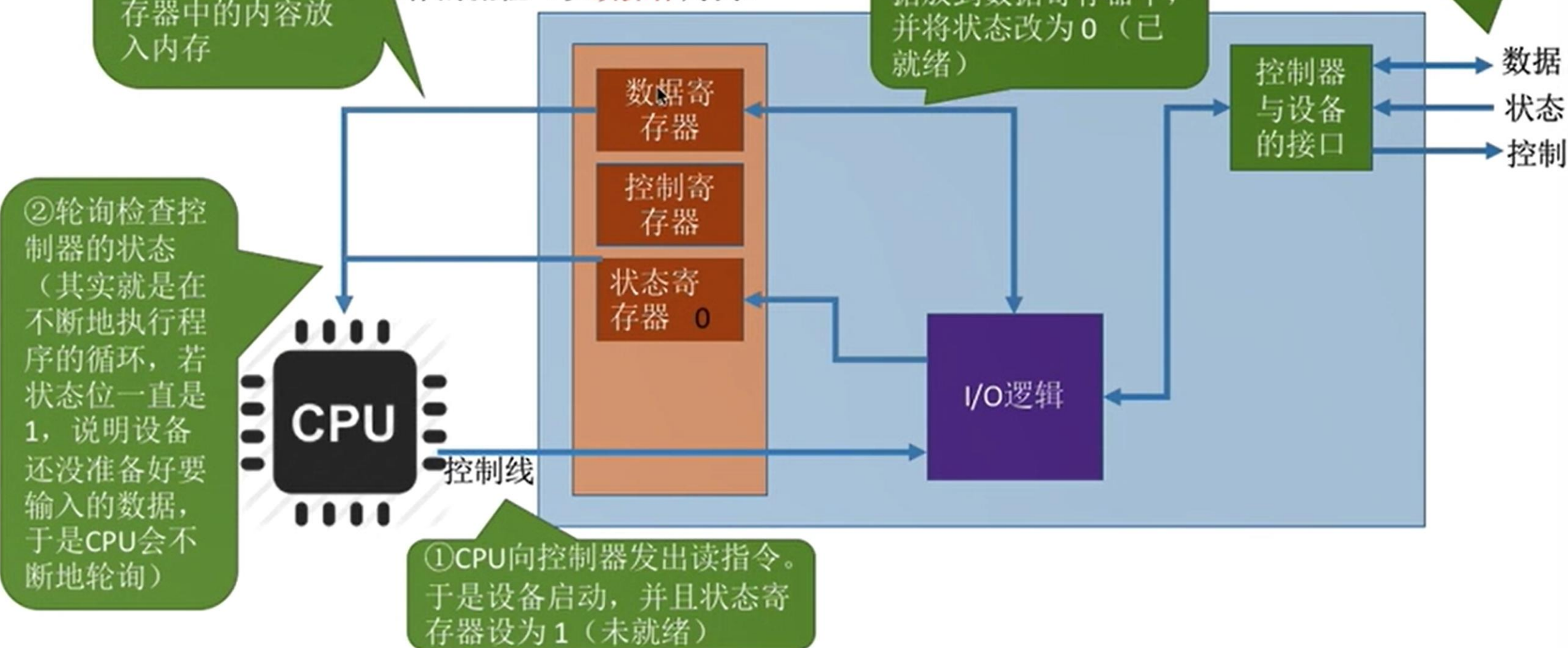
值得注意的小细节：①一个I/O控制器可能会对应多个设备；
②数据寄存器、控制寄存器、状态寄存器可能有多个（如：每个控制/状态寄存器对应一个具体的设备），且这些寄存器都要有相应的地址，才能方便CPU操作。有的计算机会让这些寄存器占用内存地址的一部分，称为**内存映像I/O**；另一些计算机则采用I/O专用地址，即**寄存器独立编址**。

本节内容

I/O控制方式

程序直接控制方式

操作的流程（以读操作为例）



程序直接控制方式

1. 完成一次读/写操作的流程（见右图，Key word: 轮询）

2. CPU干预的频率

很频繁，I/O操作开始之前、完成之后需要CPU介入，并且在等待I/O完成的过程中CPU需要不断地轮询检查。

3. 数据传送的单位

每次读/写一个字

指的是CPU的寄存器

4. 数据的流向

读操作（数据输入）：I/O设备→CPU→内存

写操作（数据输出）：内存→CPU→I/O设备

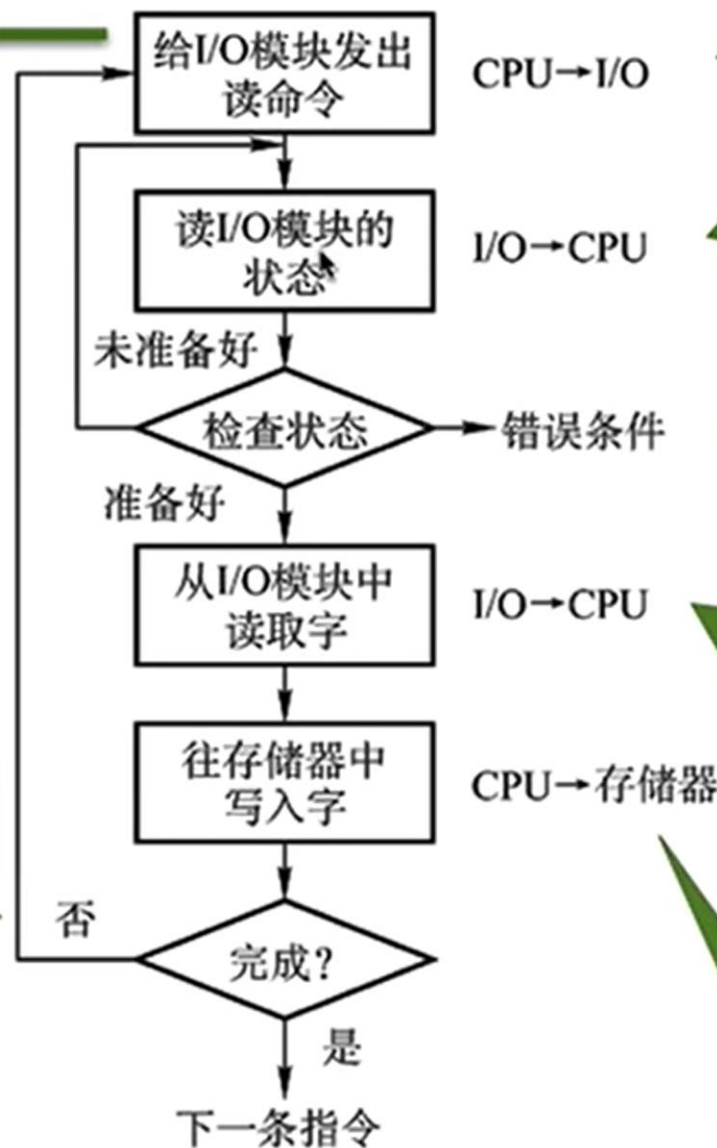
每个字的读/写都需要CPU的帮助

读入下一个字

5. 主要缺点和主要优点

优点：实现简单。在读/写指令之后，加上实现循环检查的一系列指令即可（因此才称为“程序直接控制方式”）

缺点：CPU和I/O设备只能串行工作，CPU需要一直轮询检查，长期处于“忙等”状态，CPU利用率低。



CPU向控制器发出命令

将I/O状态信息读入CPU寄存器

设备可能出现错误

将数据寄存器中的内容读入CPU寄存器

将CPU寄存器中的内容写到主存中

(a) 程序直接控制方式

中断驱动方式

1. 完成一次读/写操作的流程（见右图，Key word: 中断）

2. CPU干预的频率

每次I/O操作开始之前、完成之后需要CPU介入。

等待I/O完成的过程中CPU可以切换到别的进程执行。

3. 数据传送的单位

每次读/写一个字

4. 数据的流向

读操作（数据输入）：I/O设备→CPU→内存

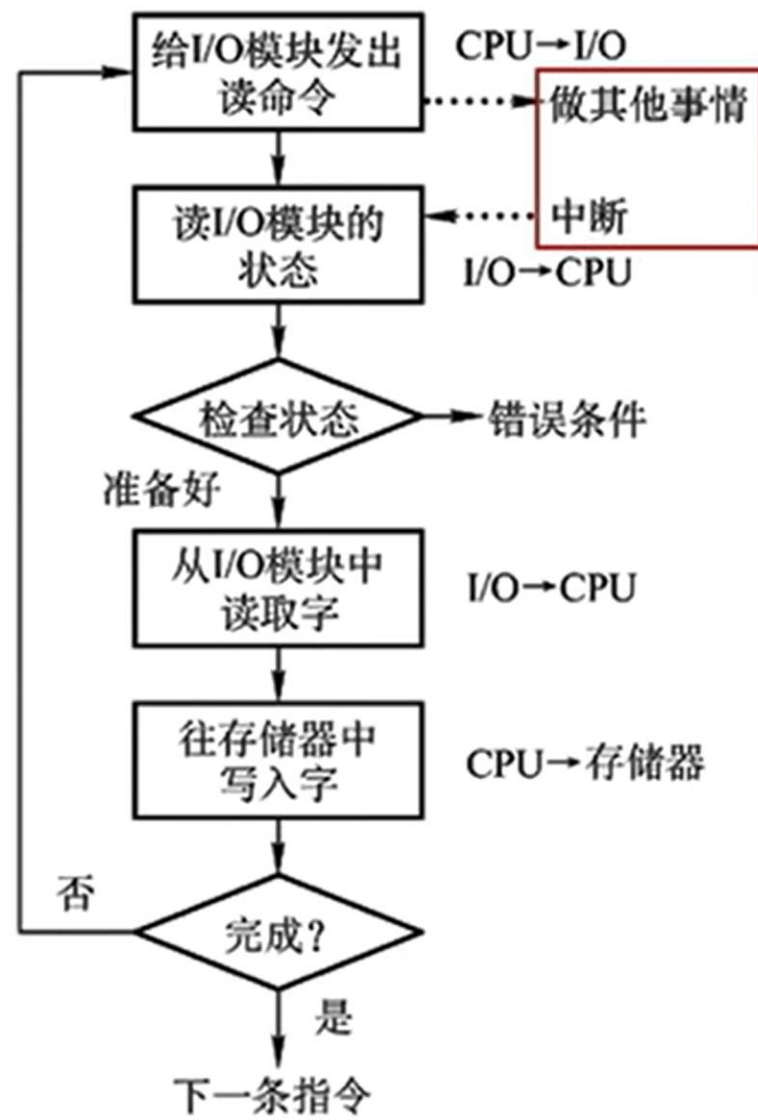
写操作（数据输出）：内存→CPU→I/O设备

5. 主要缺点和主要优点

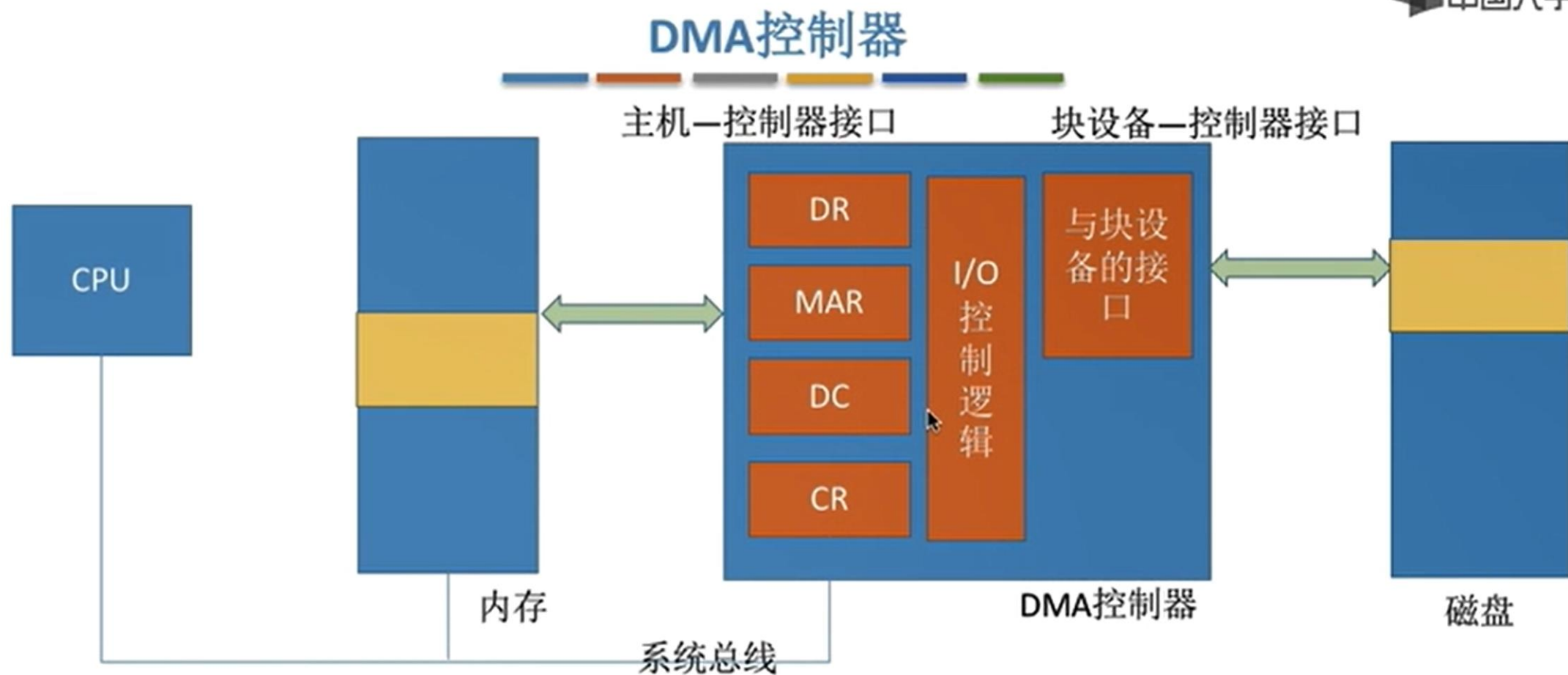
优点：与“程序直接控制方式”相比，在“中断驱动方式”中，I/O控制器会通过中断信号主动报告I/O已完成，CPU不再需要不停地轮询。

CPU和I/O设备可并行工作，CPU利用率得到明显提升。

缺点：每个字在I/O设备与内存之间的传输，都需要经过CPU。而频繁的中断处理会消耗较多的CPU时间。



(b) 中断驱动方式



南北
mooc14

DR (Data Register, 数据寄存器): 暂存从设备到内存, 或从内存到设备的数据。

MAR (Memory Address Register, 内存地址寄存器): 在输入时, MAR 表示数据应放到内存中的什么位置; 输出时 MAR 表示要输出的数据放在内存中的什么位置。

DC (Data Counter, 数据计数器): 表示剩余要读/写的字节数。

CR (Command Register, 命令/状态寄存器): 用于存放CPU发来的I/O命令, 或设备的状态信息。

DMA方式

1. 完成一次读/写操作的流程（见右图）

2. CPU干预的频率

仅在传送一个或多个数据块的开始和结束时，才需要CPU干预。

3. 数据传送的单位

每次读/写一个或多个块（注意：每次读写的只能是连续的多个块，且这些块读入内存后在内存中也必须是连续的）

4. 数据的流向（不再需要经过CPU）

读操作（数据输入）：I/O设备→内存

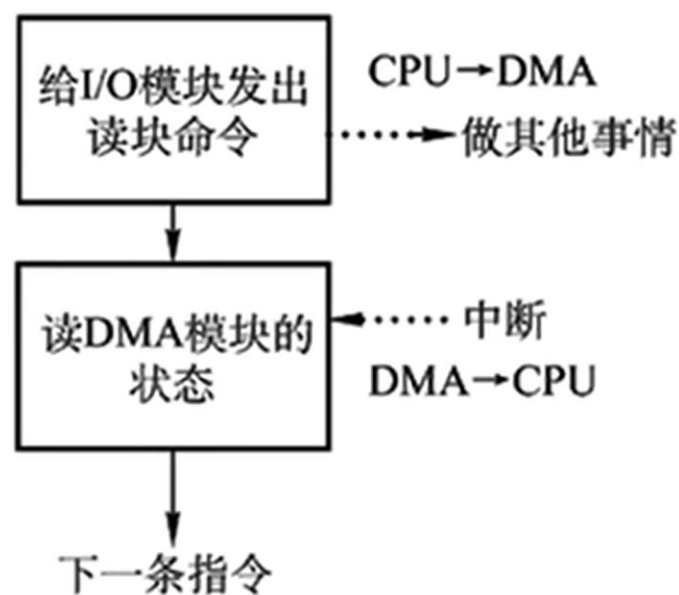
写操作（数据输出）：内存→I/O设备

5. 主要缺点和主要优点

优点：数据传输以“块”为单位，CPU介入频率进一步降低。数据的传输不再需要先经过CPU再写入内存，数据传输效率进一步增加。CPU和I/O设备的并行性得到提升。

缺点：CPU每发出一条I/O指令，只能读/写一个或多个连续的数据块。

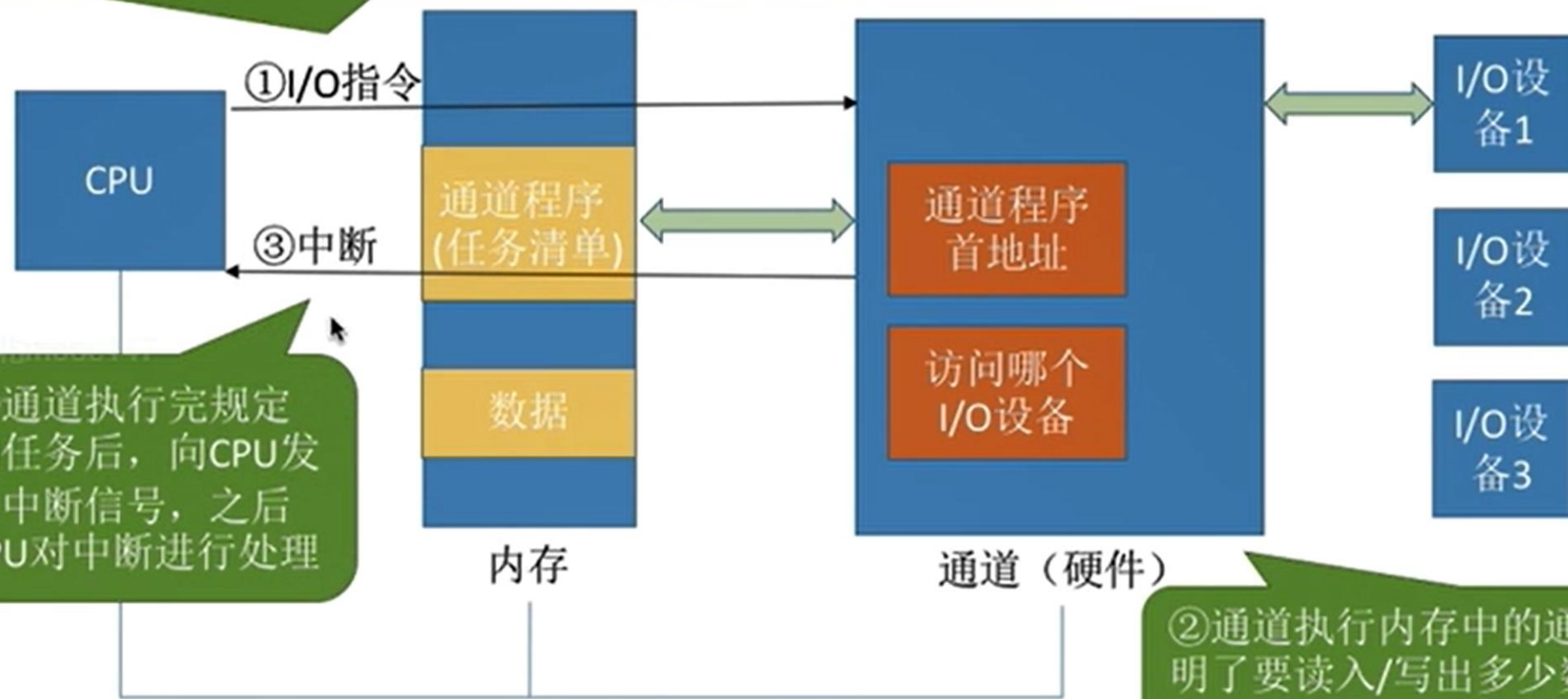
如果要读/写多个离散存储的数据块，或者要将数据分别写到不同的内存区域时，CPU要分别发出多条I/O指令，进行多次中断处理才能完成。



通道控制方式

通道：一种**硬件**，可以理解为是“**弱鸡版的CPU**”。通道可以识别并执行一系列**通道指令**

①CPU向通道发出I/O指令。指明通道程序在内存中的位置，并指明要操作的是哪个I/O设备。之后CPU就切换到其他进程执行了



③通道执行完规定的任务后，向CPU发出中断信号，之后CPU对中断进行处理

②通道执行内存中的通道程序（其中指明了要读入/写出多少数据，读/写的数据应放在内存的什么位置等信息）

通道控制方式

通道：一种**硬件**，可以理解为是“**弱鸡版的CPU**”。通道可以识别并执行一系列**通道指令**

与CPU相比，通道可以执行的指令很单一，并且通道程序是放在主机内存中的，也就是说通道与CPU共享内存

1. 完成一次读/写操作的流程（见右图）

2. CPU干预的频率

极低，通道会根据CPU的指示执行相应的通道程序，只有完成一组数据块的读/写后才需要发出中断信号，请求CPU干预。

3. 数据传送的单位

每次读/写一组**数据块**

4. 数据的流向（在**通道的控制下进行**）

读操作（数据输入）：I/O设备→内存

写操作（数据输出）：内存→I/O设备

5. 主要缺点和主要优点

缺点：实现复杂，需要专门的通道硬件支持

优点：**CPU、通道、I/O设备可并行工作**，资源利用率很高。

