



二、简单微程序控制器的设计

第二讲

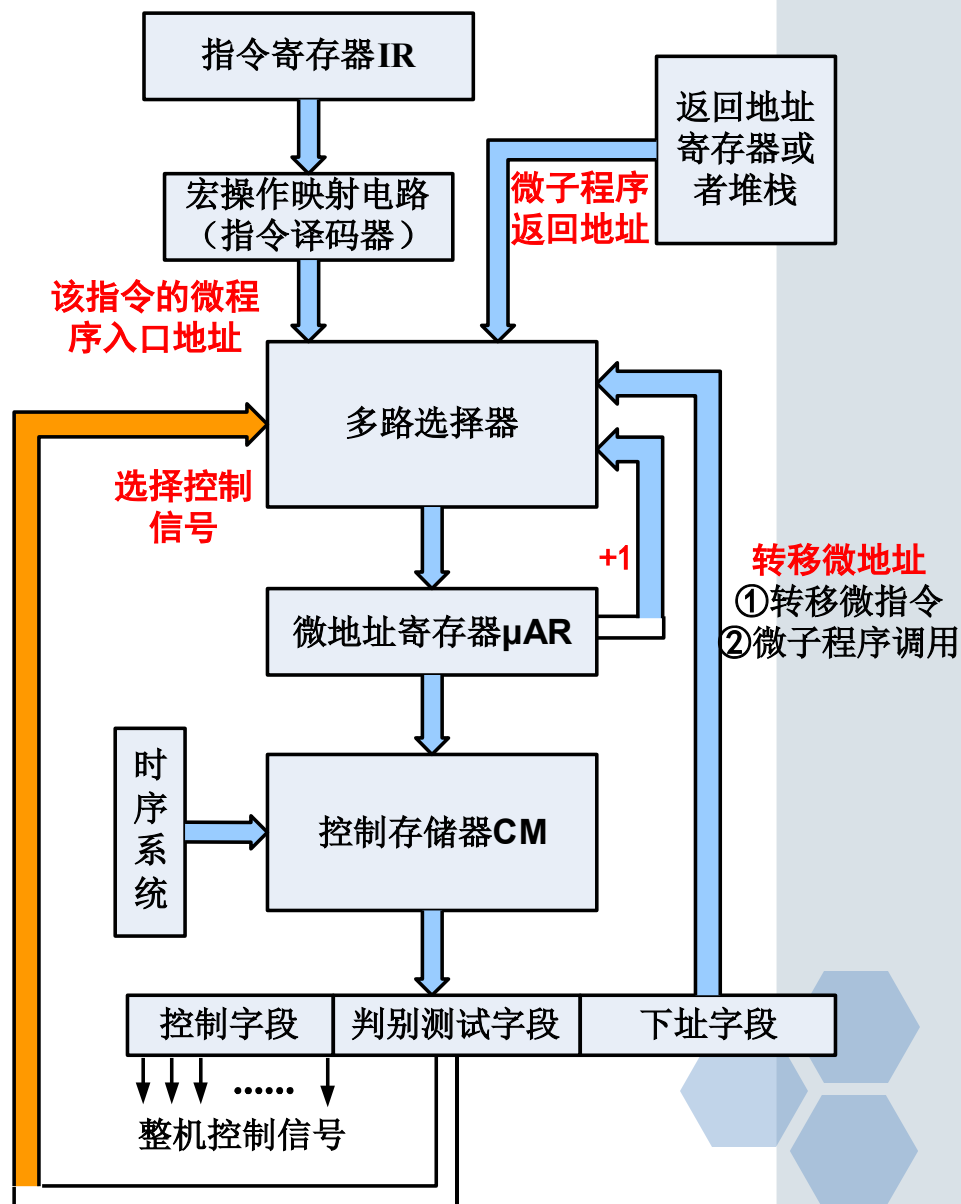
简单微程序控制器的设计（一）



二、简单微程序控制器的设计

❖ 问题：如何确定下一条微指令的地址来源？

❖ 增加判别测试字段，用于指明后继微地址的来源。





二、简单微程序控制器的设计

下址的来源：

- 1 . 根据机器指令操作码产生该指令对应的**微程序入口地址（指令译码）**；
- 2 . **顺序地址**：就是控制存储器中的下一个地址；
- 3 . 在微程序发生转移时的**无条件转移地址**；
- 4 . 当调用微子程序时发生**转移的子程序地址**；
- 5 . **子程序返回地址**（返回寄存器）或硬件堆栈里；
- 6 . 根据运算的条件（或外部条件）形成的**条件转移地址等**





二、简单微程序控制器的设计

描述如下：

开机后首先使微地址寄存器置为取指令的第一条微指令地址，从控制存储器中取出第一条微指令，完成 $PC \rightarrow AR$ 、 $PC+1$ 操作，然后根据微指令的下址字段取出第二条微指令，完成 $RAM \rightarrow IR$ 从内存中读出指令送指令寄存器 IR ，并发译码信号使指令译码器工作，形成该指令的执行指令阶段的微程序入口地址，从控存中取出该指令执行时的第一条微指令送到微指令寄存器，发出控制信号（微命令）实现微操作；





二、简单微程序控制器的设计

描述如下：

然后该指令执行时的其余微指令地址是当前微地址加 1 或由当前微指令下址字段确定，依次从控存中取出其余微指令，实现该指令所需的所有微操作，即完成了该指令的执行。每一条指令的最后一条微指令执行完后均会回到取指令的第一条微指令执行，以取下一条指令，如此重复，直至用户要运行的程序指令执行完为止。



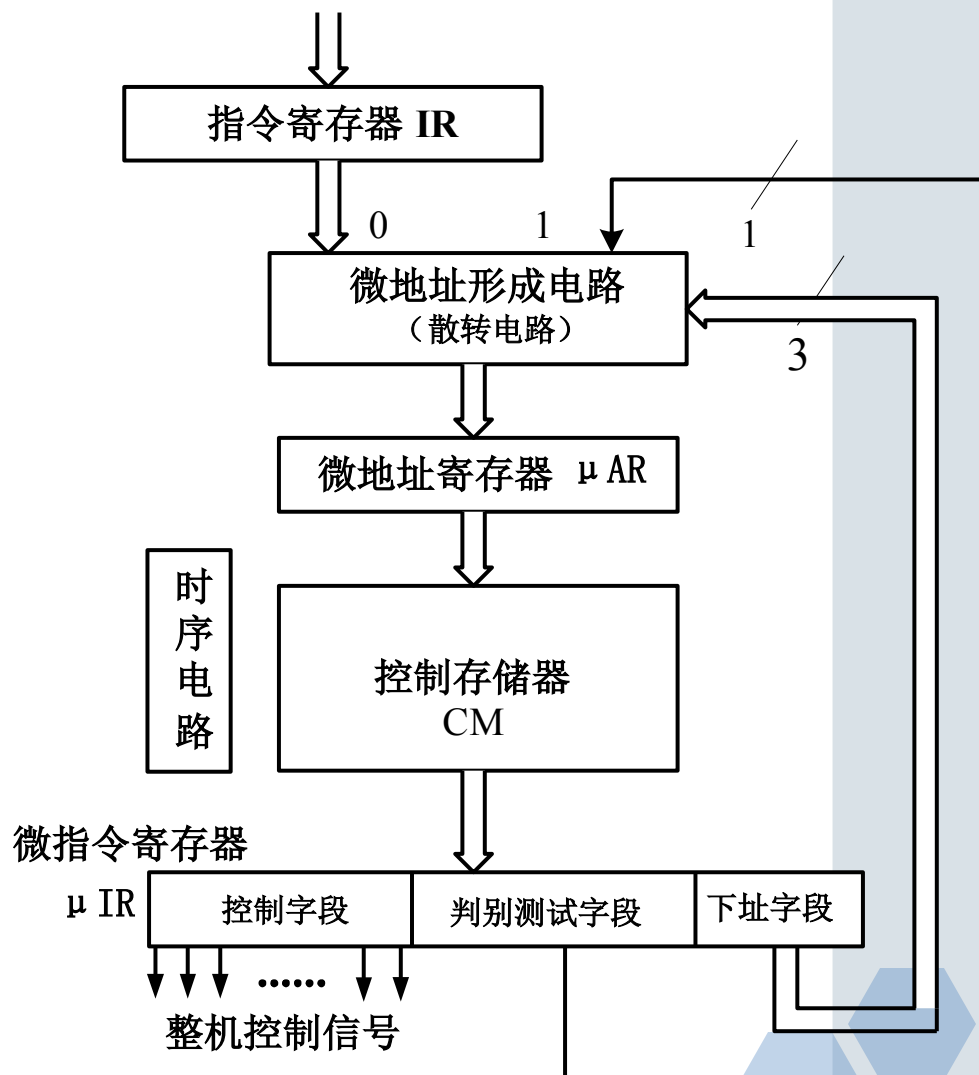


二、简单微程序控制器的设计

❖ 微程序控制器的设计 主要完成两个任务：

- 产生正确的微命令；
- 产生正确的微指令序列（即上述 CPU 状态转换序列）。

❖ 怎样采用微程序控制的方法来设计 CPU 呢？



简单微程序控制器的组成框图



二、简单微程序控制器的设计

❖ 确定指令系统

助记符	格式			操作码 OP	指令机器码	指令功能
ADD R ₀ , 06H	OP	xx	DR	0101	01010000	(R ₀) + 06H→R ₀
	立即数				00000110	
JMP 04H	OP	xxxx		1000	10000000	04H→P C
	转移地址				00000100	

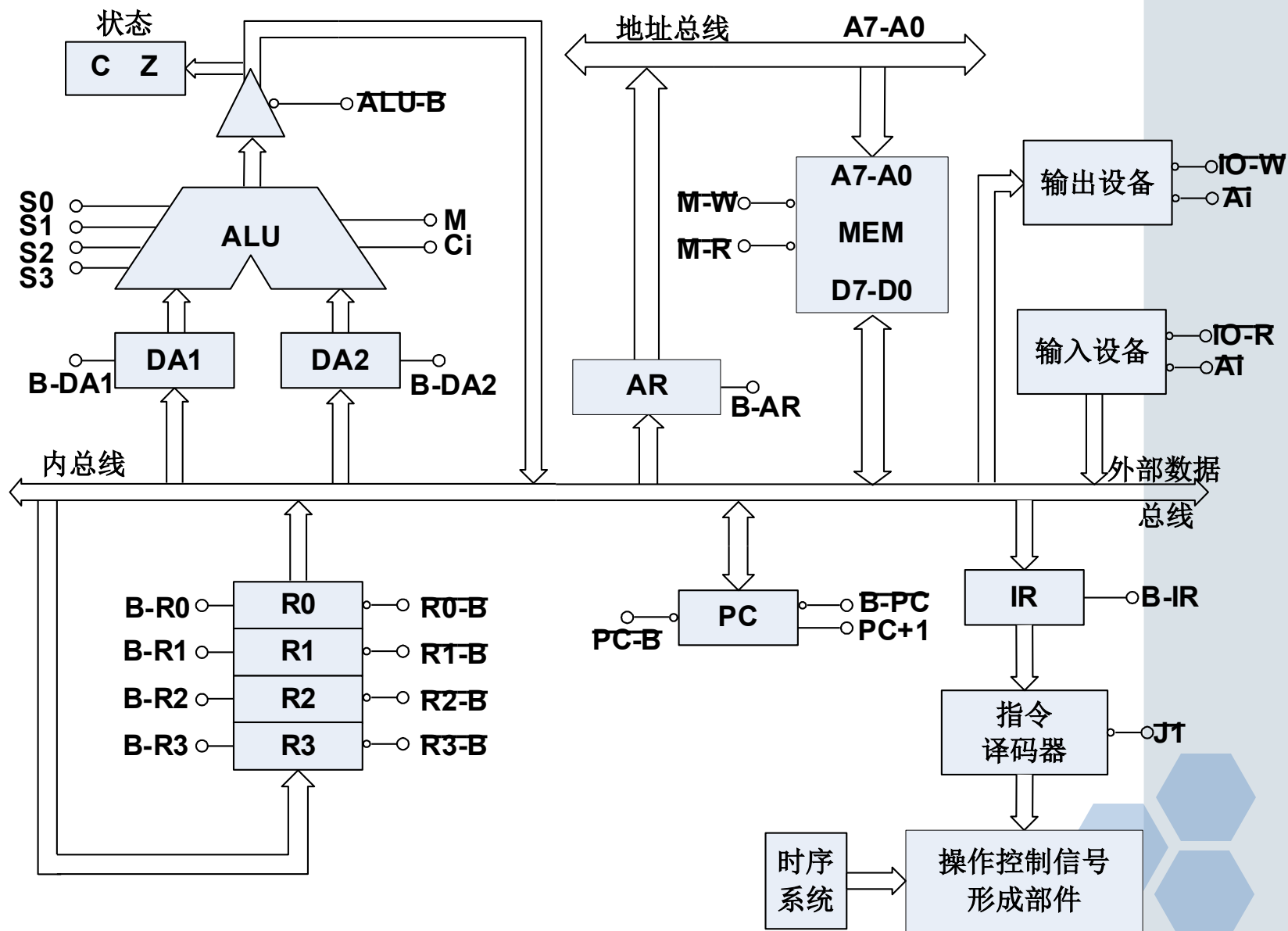




二、简单微程序控制器的设计

2

确定CPU的内部结构





模型机要点：

1. 模型机以总线结构构建；
2. 总线宽度为 8 位二进制；
3. 数据和指令都通过总线传输；
4. 总线以分时复用的形式传输数据和指令，
每一时刻只能有一个 8 位的二进制数据或指令
进行传输
(即：每一时刻只能有一个部件输出数据)；
5. 每个部件有相应的控制信号（微命令）控制
其
数据输入或输出；
6. 微命令由微指令输出控制。
7. 设指令译码控制信号为 J1#（低电平有效）
8. 模型机的运算器采用 74LS181（课本第二