FSE598 前沿计算技术

模块 1　计算思维
单元 1　计算机系统设计
第 4 讲　数据路径与控制单元

本讲座的英文版内容基于教材：
The English version of the lectures are partly based on the book:
Patterson and Hennessy, Computer Organization and Design: The Hardware Software Interface

本讲大纲

学习

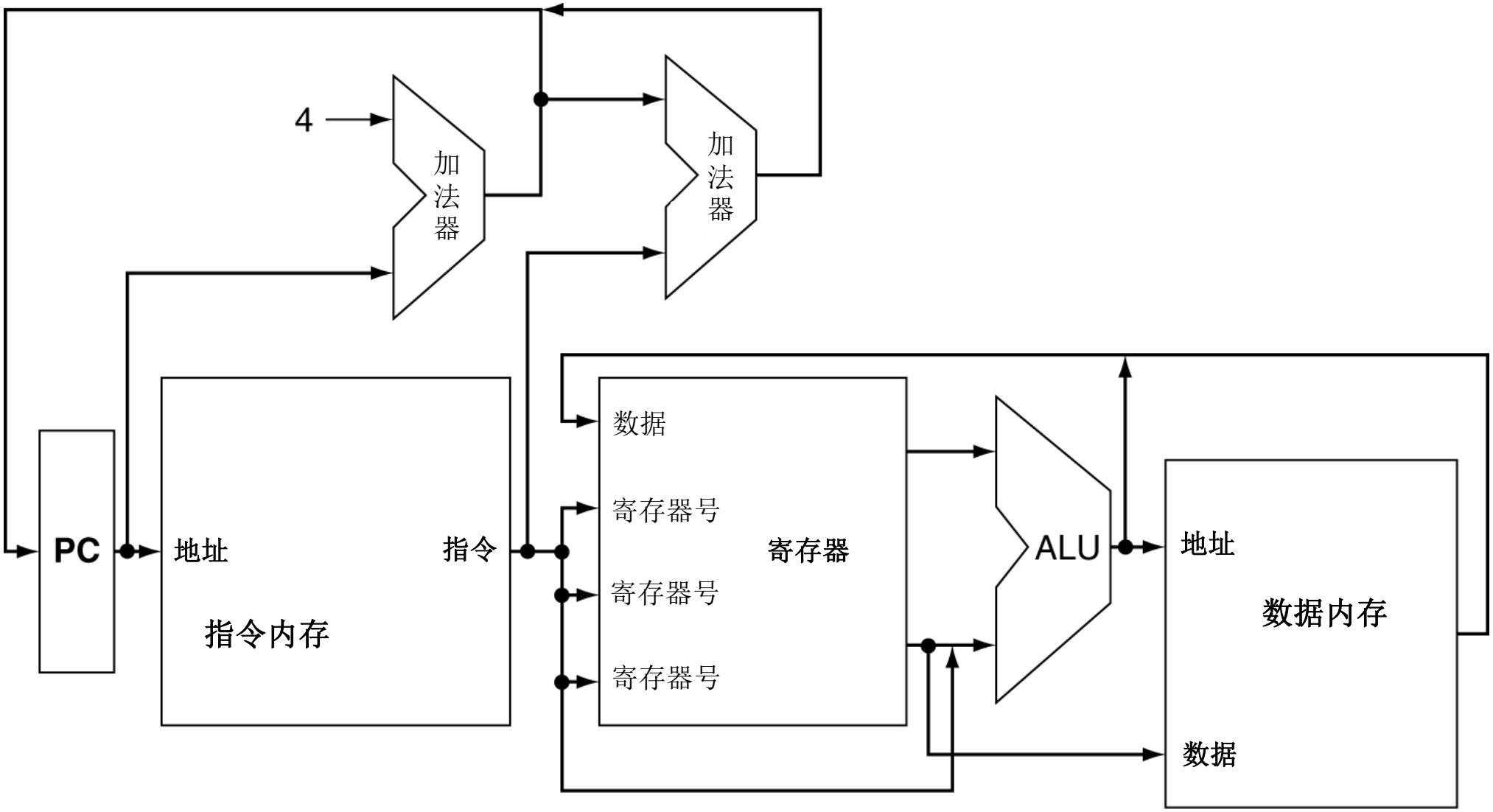
- 指令执行：五个步骤
- 处理器概念模型
- 数据路径与内存
- 不同指令的执行
- 控制信号

指令执行：五个步骤

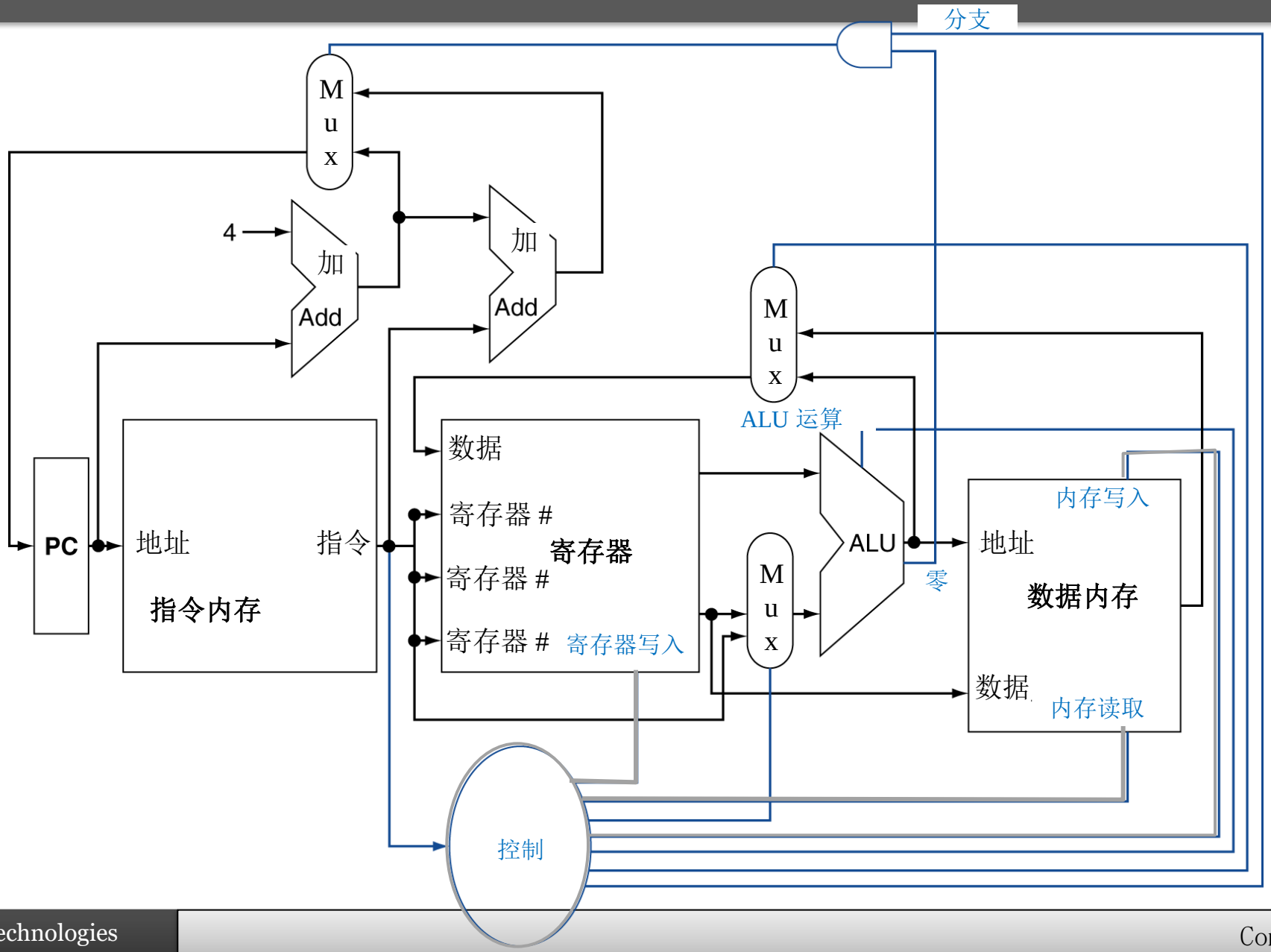
1. PC →指令内存，获取指令，

   和 PC ← PC + 4
2. 寄存器编号 → 寄存器文件，读取寄存器
3. 取决于指令类型
   - 使用 ALU 执行
     - 算术结果
     - 加载/存储的内存地址
     - 分支目标地址
4. 访问加载/存储的数据内存
5. 将结果写回寄存器文件

CPU 概念模型：数据路径与内存



包含多路转接器（Mux) 和控制单元



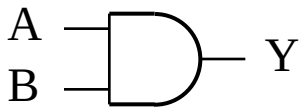
逻辑设计基础：回顾

- 二进制编码的信息
  - 低压 = 0，高压 = 1
  - 每个位连一条线
  - 在多线总线上编码的多位数据

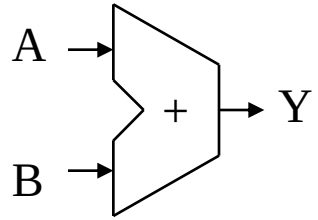
- 组合元件和电路
  - 对数据进行操作
  - 输出是输入的函数
- 状态元件和时序电路
  - 将信息存储在状态（内存）中
  - 输出是输入和状态的函数

组合元件示例

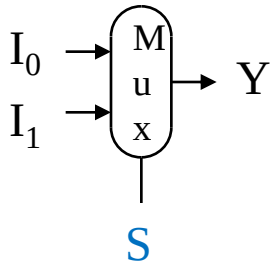
- 与门 (and-gate)
  - Y = A & B



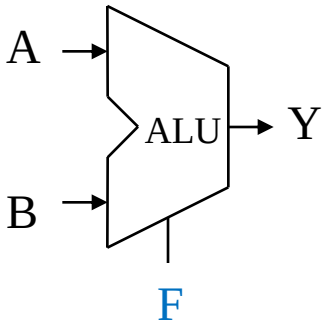
- 加法器
  - Y = A + B



- 多路转接器
  - Y = (S==0) ? I_0 : I_1



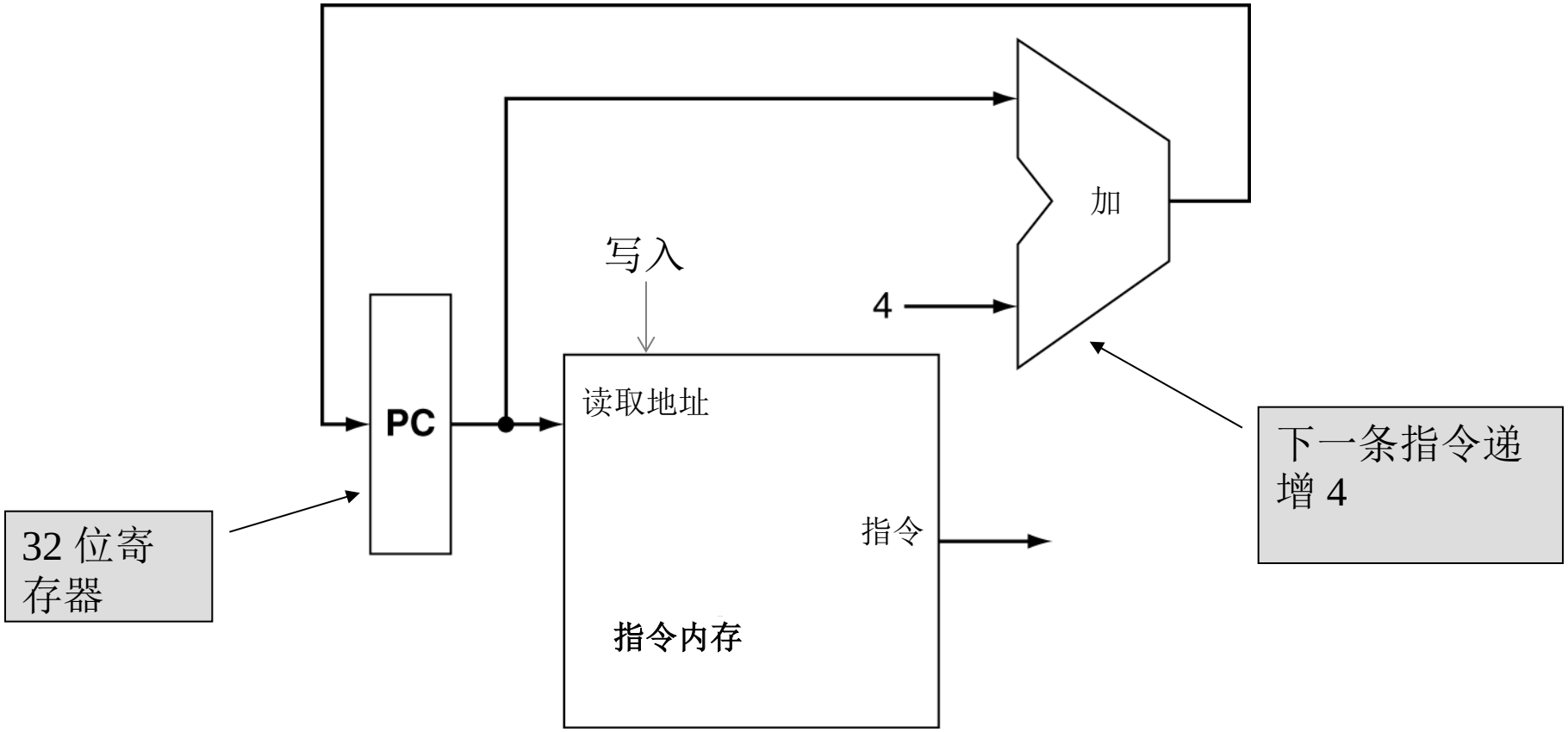
- 算数/逻辑单元
  - Y = fun(A, B, F)



构建数据路径

- 数据路径定义：
  - CPU 中处理数据和地址的元件
    - 寄存器
    - ALU、加法器、多路转接器、
    - 内存的地址计算逻辑（不包括内存和内存中的寻址逻辑）
    - ……
- 我们将逐步构建数据路径
  - 从总体设计逐步完善

指令提取：指令执行的第 1 步

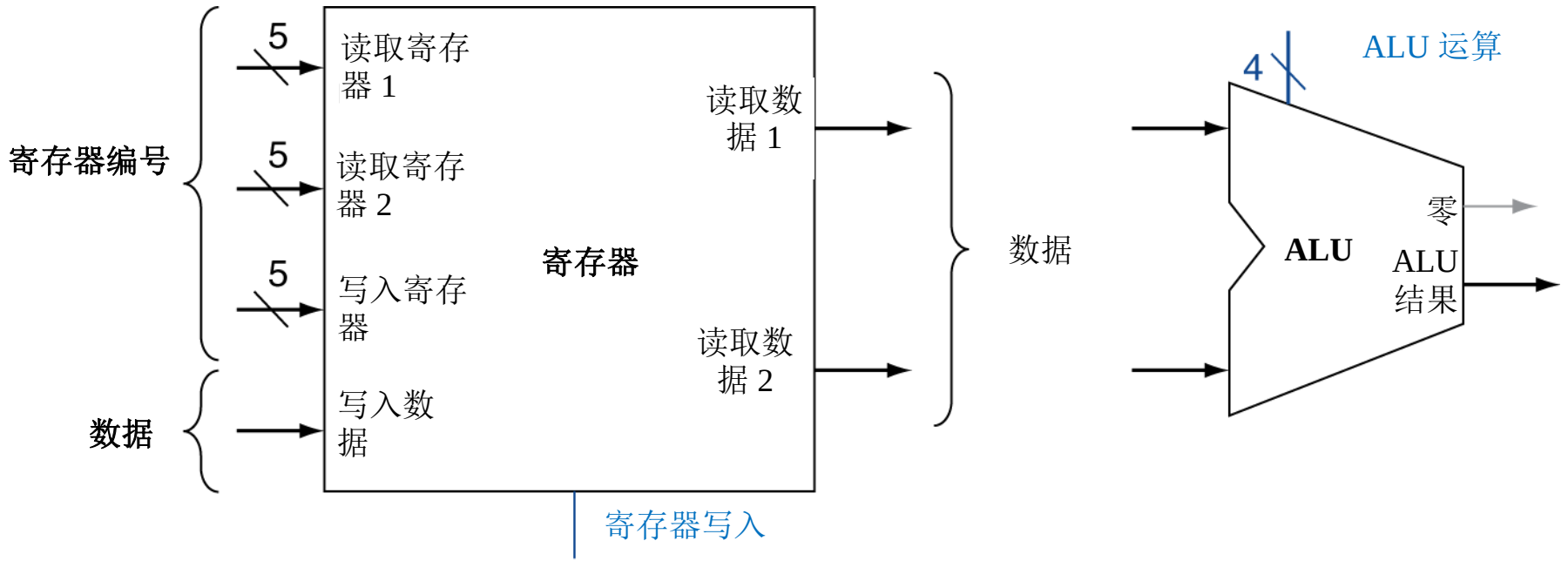



```
add $10, $11, $12    # $10 ← $11 + $12
```

R 格式指令

- 读取两个寄存器运算数
- 执行算数/逻辑运算
- 写入寄存器结果

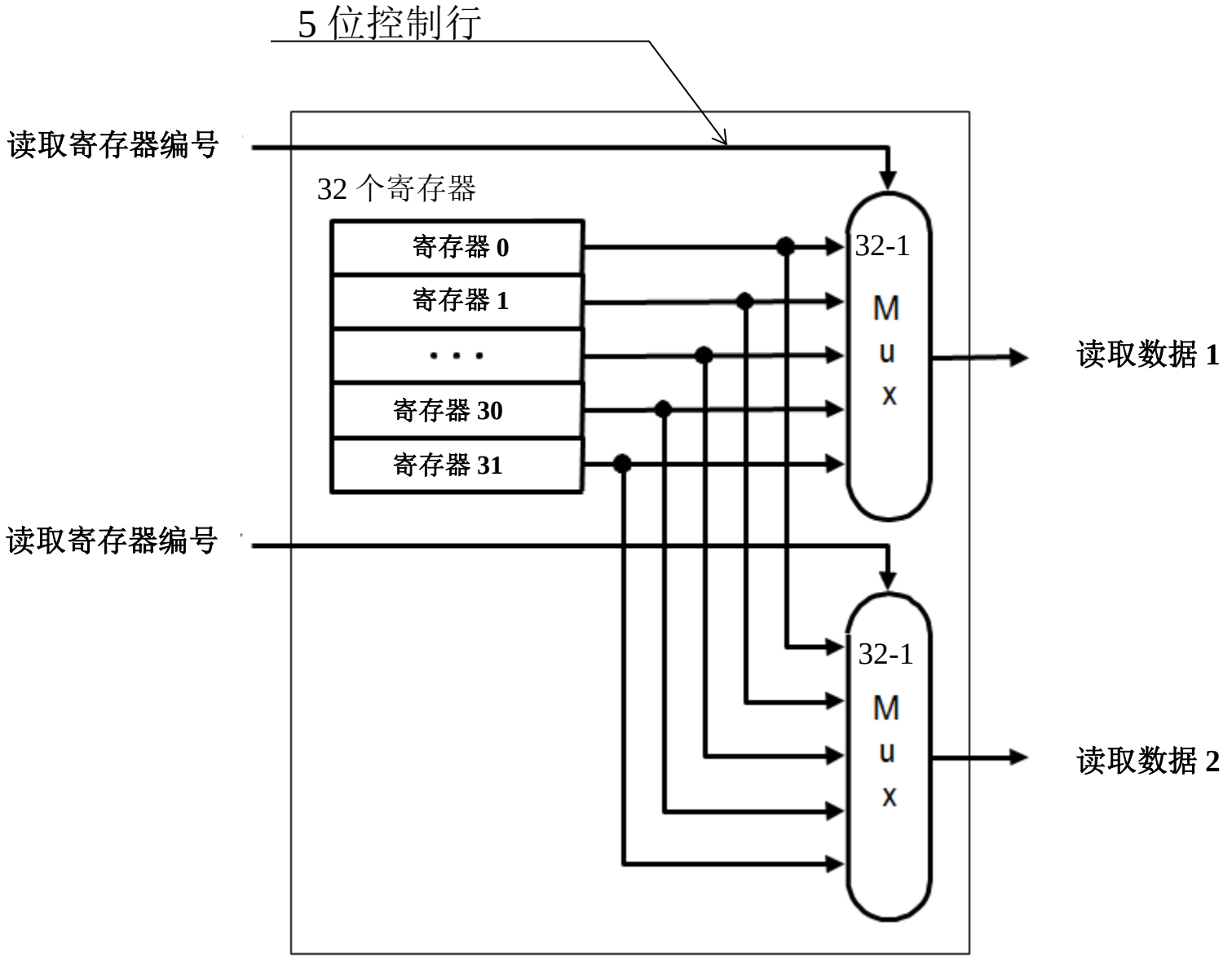
```
add $10, $11, $12 # $10 ← $11 + $12
```



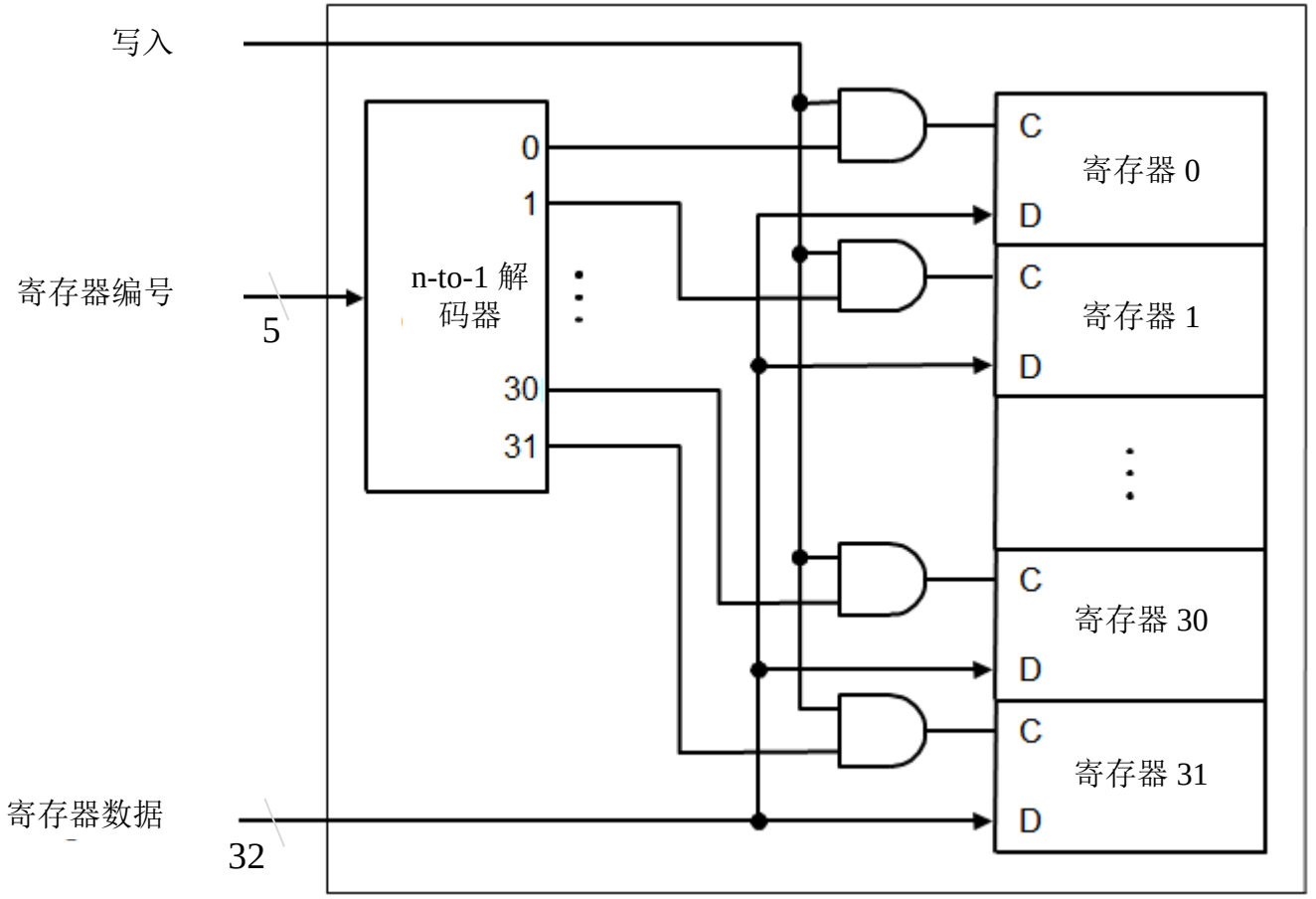
a. 寄存器

b. ALU

读取寄存器文件及其选择逻辑



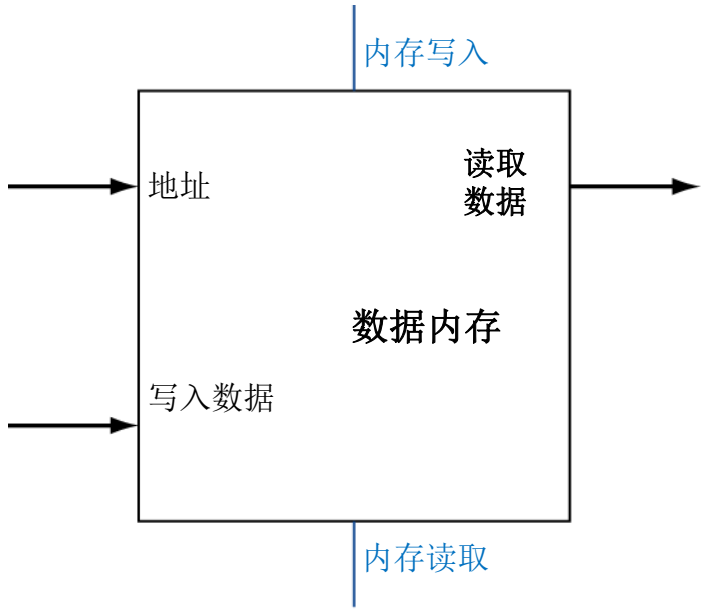
写入寄存器文件及其解码逻辑



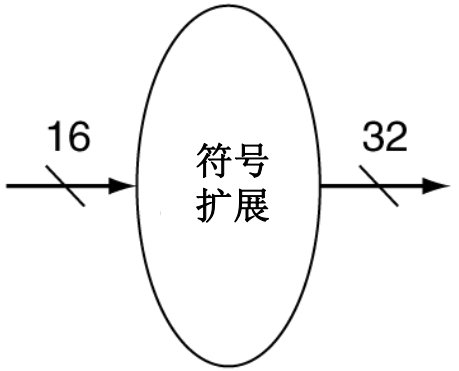
注意：读取和写入内存位置类似于读取和写入寄存器。实际上，寄存器文件只是一个有 32 个位置的小内存。

加载/存储指令

- 读取寄存器运算数
- 使用 16 位偏移量计算地址
  - 使用 ALU，但对偏移进行符号扩展
- 加载：读取内存和更新寄存器
- 存储：将寄存器值写入内存



a. 数据内存单元



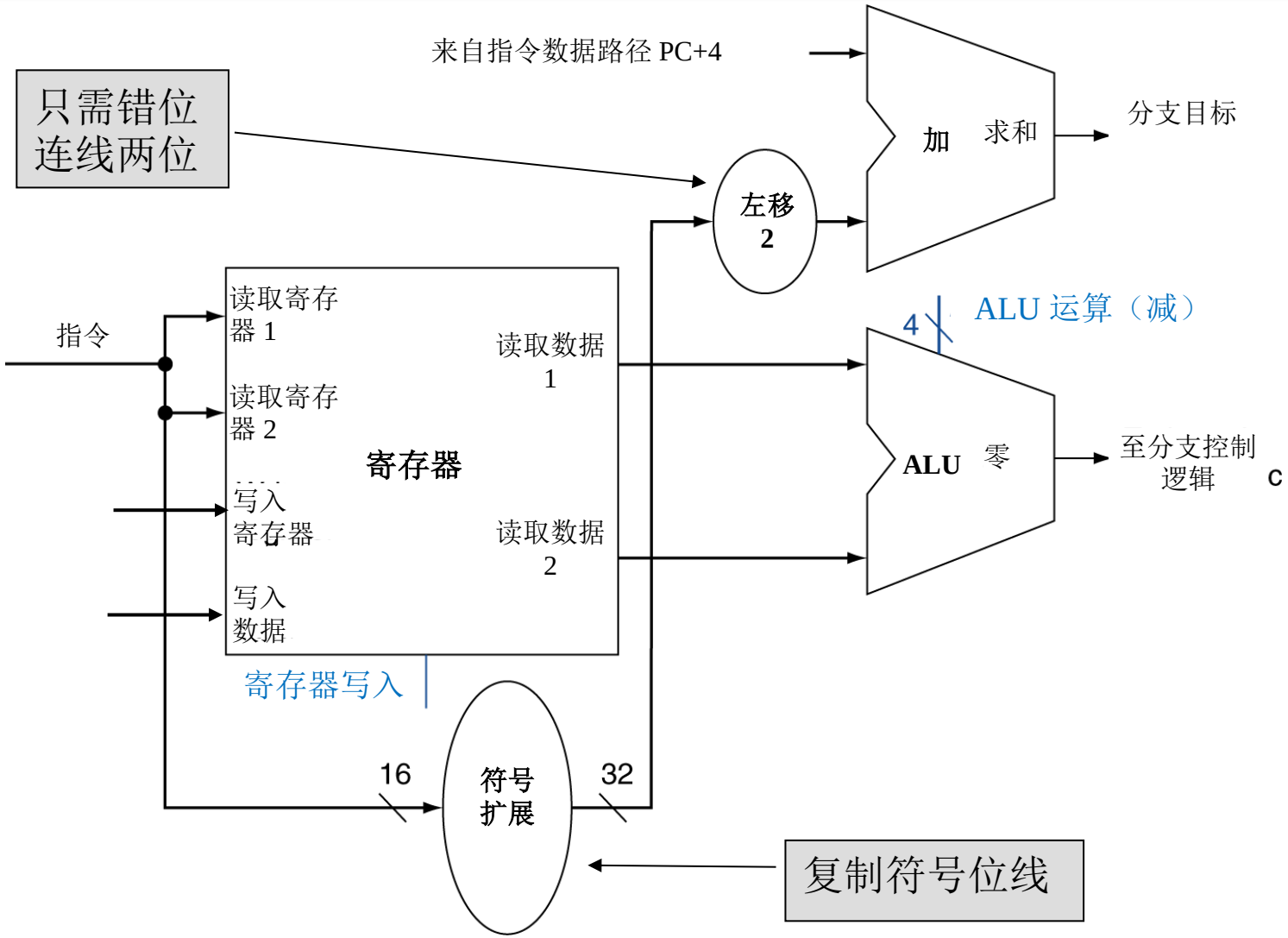
b. 符号扩展单元

分支指令

- 读取寄存器运算数
- 比较运算数
  - 使用 ALU：减去并生成零输出
- 计算目标地址
  - 符号扩展位移
  - 左移 2 个位置（词位移）
  - 加至 PC + 4
    - 已经通过指令提取计算

```
beq $14, $15, L # if ($14 == $15) go to L
```

分支指令

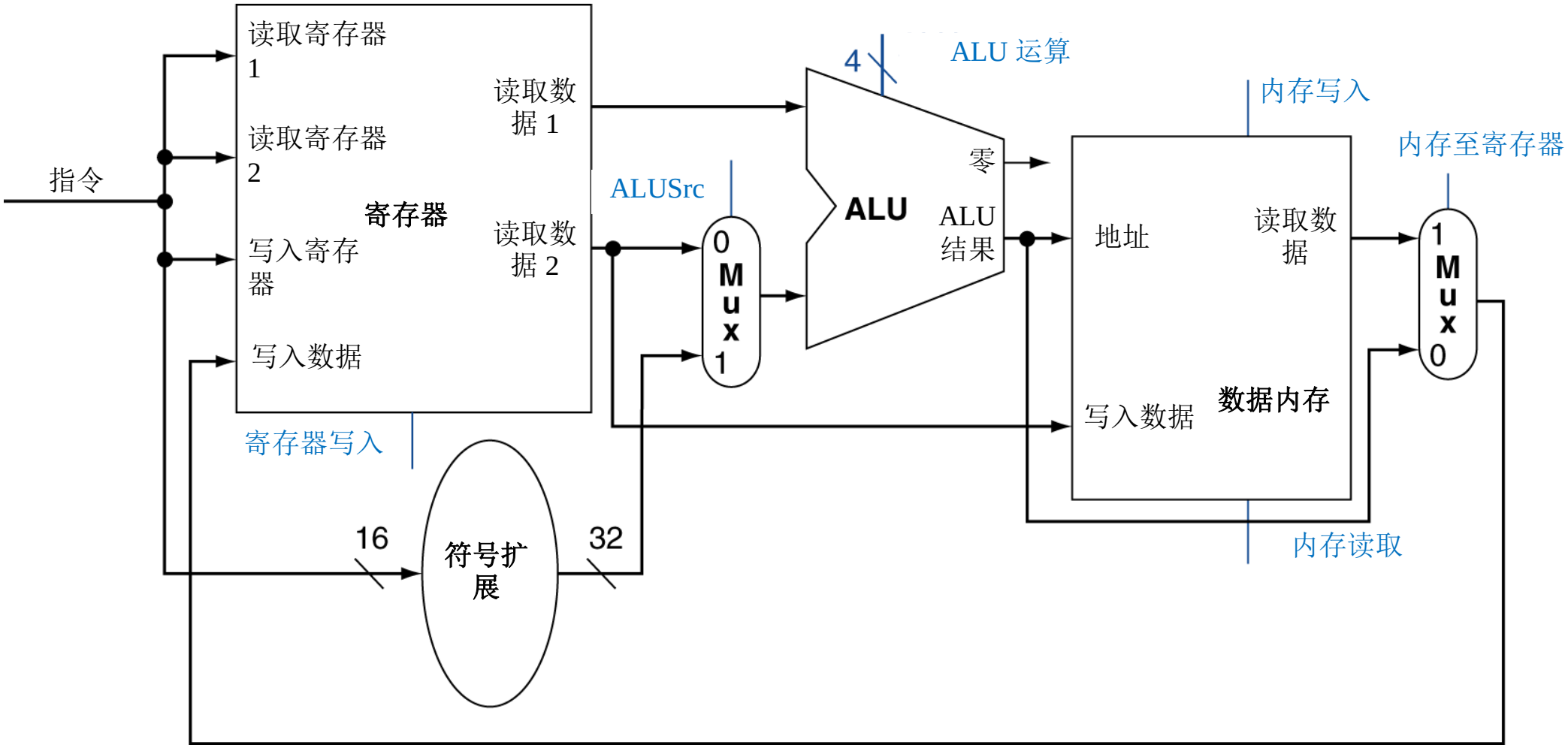



```
beq $14, $15, L # if ($14 == $15) go to L
```

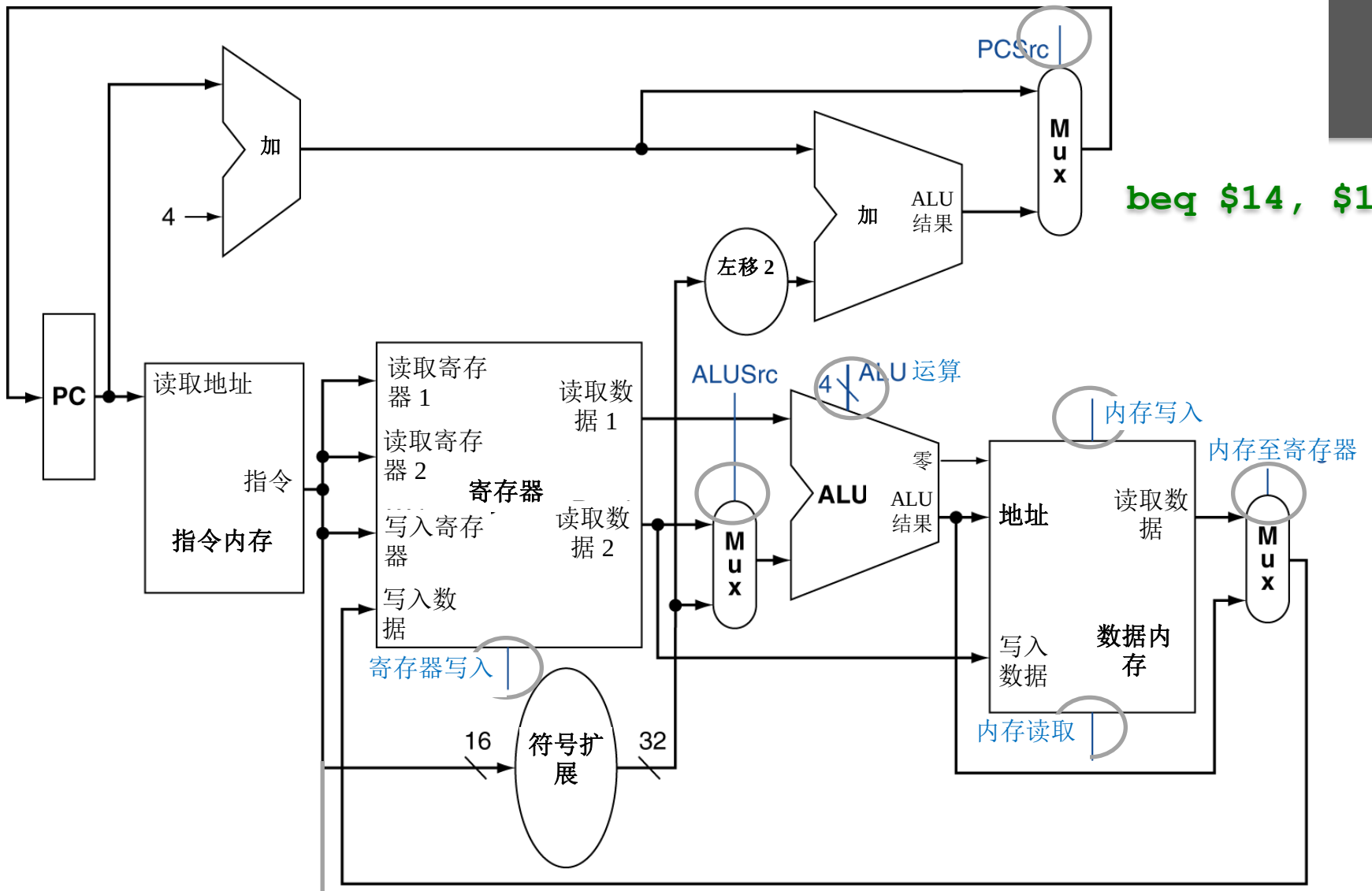
组成元件

- 我们设计数据路径以在**单个时钟周期**内执行指令
  - 每个数据路径元素一次只能执行一个函数
  - 因此，我们需要单独的指令和数据内存
  - 多个 ALU
- 使用多路转接器，可以将不同数据源导于不同的指令

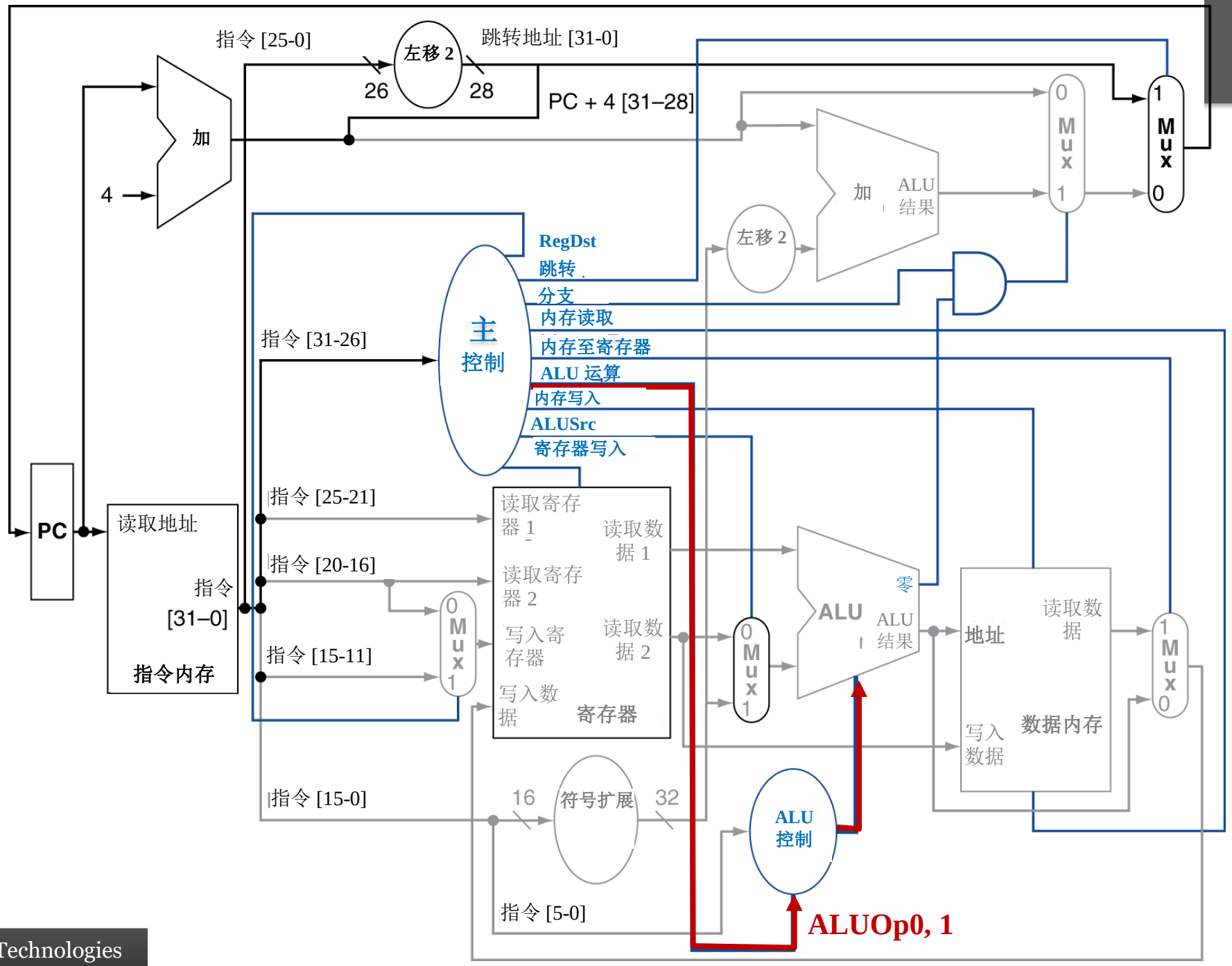
算数/加载/存储类型数据路径



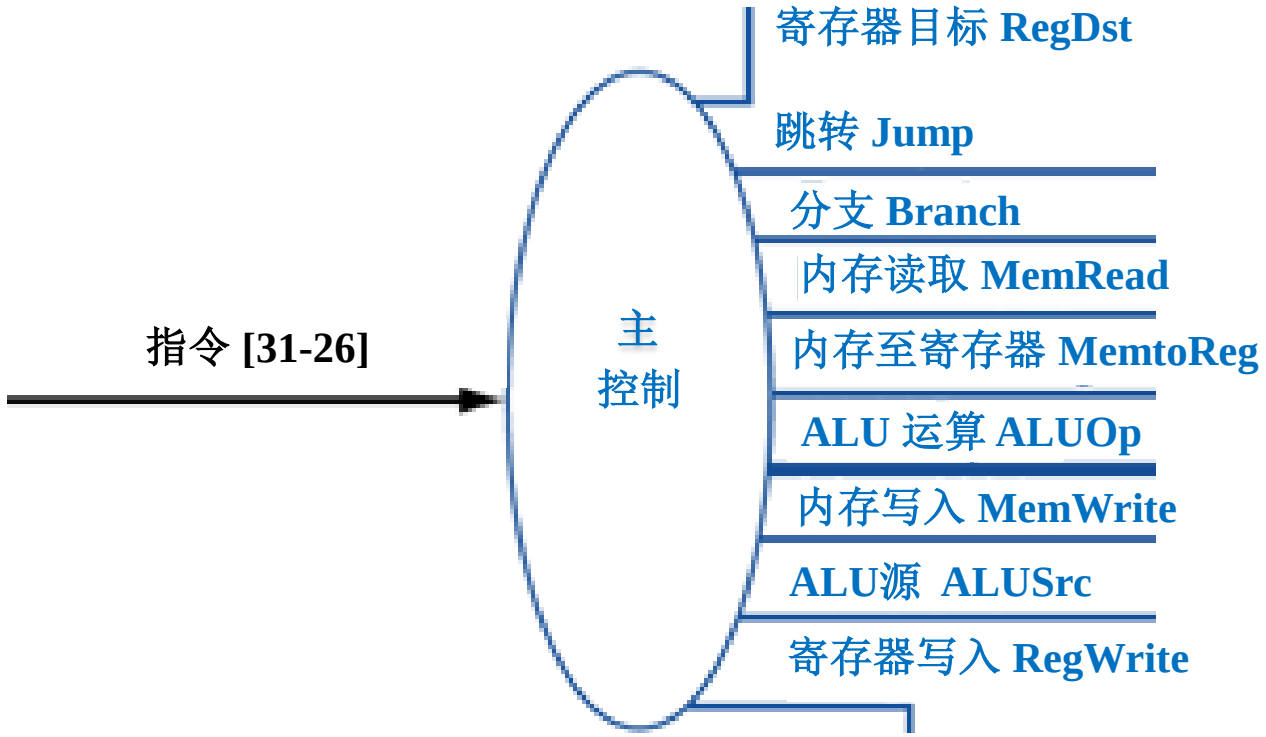

```
add $10, $11, $12    # $10 ← $11 + $12
```


PCSrc
加
4
M
u
x
加
ALU
结果
beq $14, $15, L
左移 2
PC
读取地址
指令
指令内存
读取寄存
器 1
读取寄存
器 2
寄存器
写入寄存
器
写入数
据
读取数
据 1
读取数
据 2
ALUSrc
4
ALU 运算
内存写入
内存至寄存器
M
u
x
ALU
零
ALU
结果
地址
读取数
据
写入
数据
数据内
存
寄存器写入
16
符号扩
展
32
内存读取
add $10, $11, $12
指令
[31-26]
主
控制
独立信号
— 不需要解码器
2 个组合信号：ALUOp0, 1
— 需要解码

完整的控制单元的数据路径



我们如何设计主控制单元？



使用模块化设计的64位输入的或门的真值表

Additional Slides (Not To be Assessed)
额外幻灯片（不会评估）

RISC-V 计算机架构仿真器 (学习VIPLE后可以练习)

https://venus.sod.asu.edu/VIPLE/RISCV/