

本节主题



数据通路的建立

18年原题出处！
高度重视！！
必须掌握！！

北京大学·慕课
计算机组成
制作人：陆俊林



建立数据通路的方法



④ 基本原则

- 根据指令需求，连接组件，建立数据通路

④ 指令的需求

- 所有指令的共同需求
- 不同指令的不同需求

建立数据通路的方法



④ 基本原则

- 根据指令需求，连接组件，建立数据通路

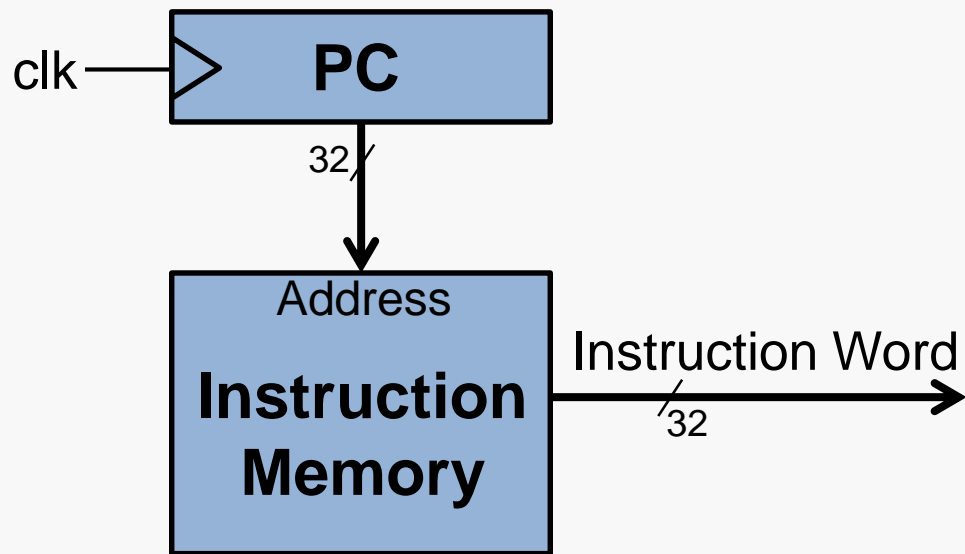
④ 指令的需求

- **所有指令的共同需求**
- 不同指令的不同需求

所有指令的共同需求

取指令

- 程序计数器（PC）的内容是指令的地址
- 用PC的内容作为地址，访问指令存储器获得指令编码



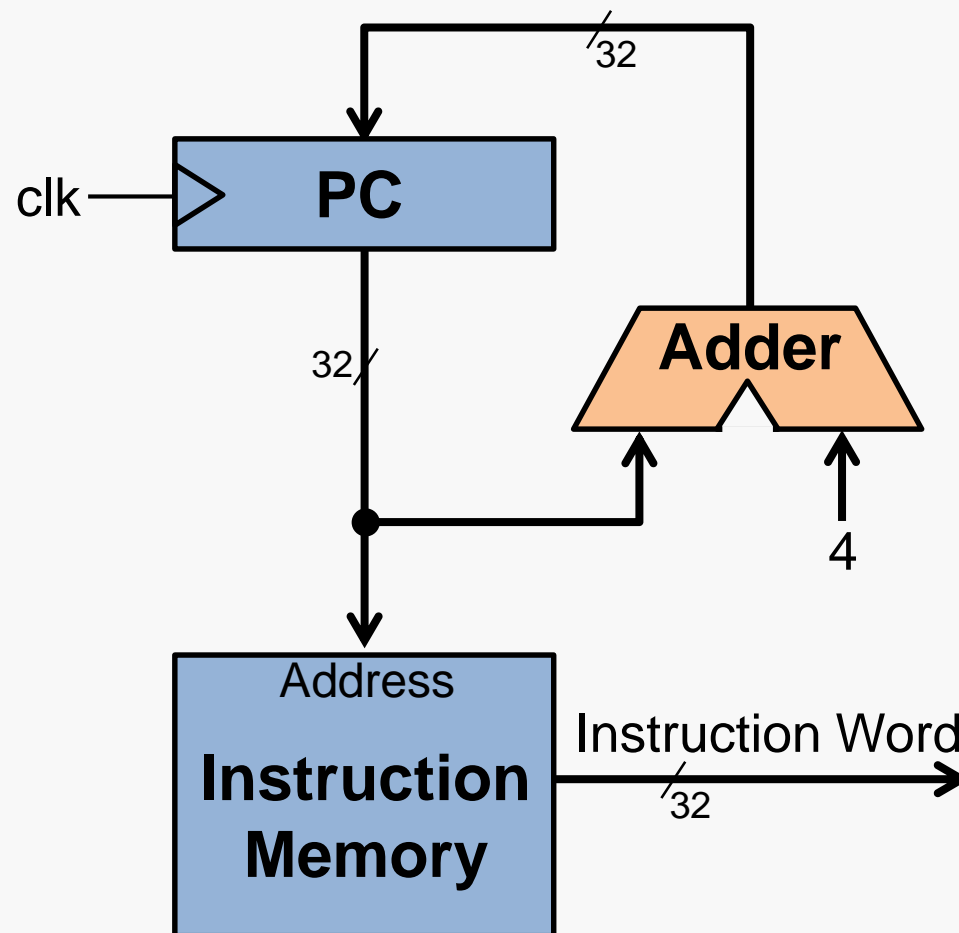
所有指令的共同需求

取指令

- 程序计数器（PC）的内容是指令的地址
- 用PC的内容作为地址，访问指令存储器获得指令编码

更新程序计数器（PC）

- 顺序执行时
 - $PC \leftarrow PC + 4$
- 发生分支时
 - $PC \leftarrow$ 分支目标的地址



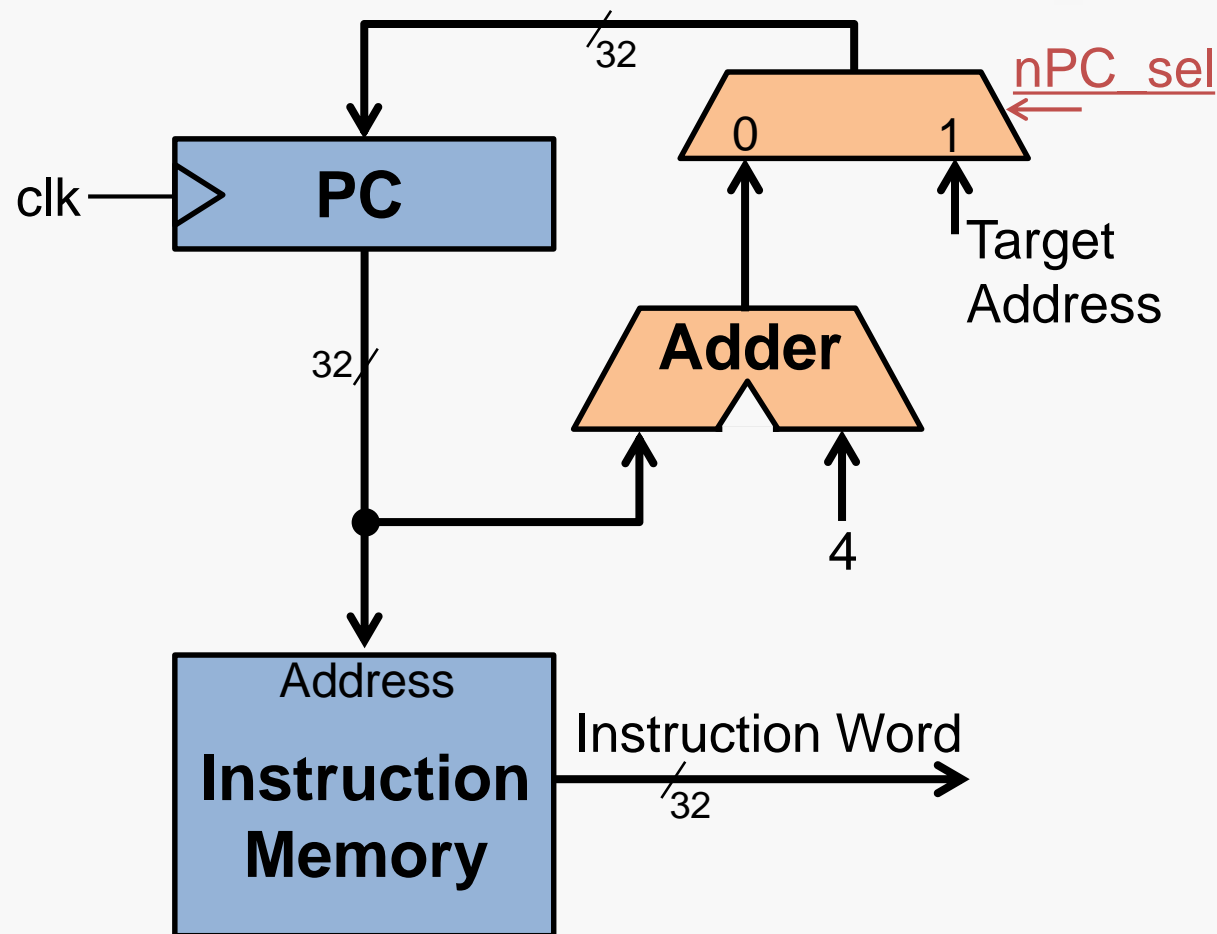
所有指令的共同需求

取指令

- 程序计数器（PC）的内容是指令的地址
- 用PC的内容作为地址，访问指令存储器获得指令编码

更新程序计数器（PC）

- 顺序执行时
 - $PC \leftarrow PC + 4$
- 发生分支时
 - $PC \leftarrow \text{分支目标的地址}$



所有指令的共同需求

这几个组件先记住，后面画图会用到

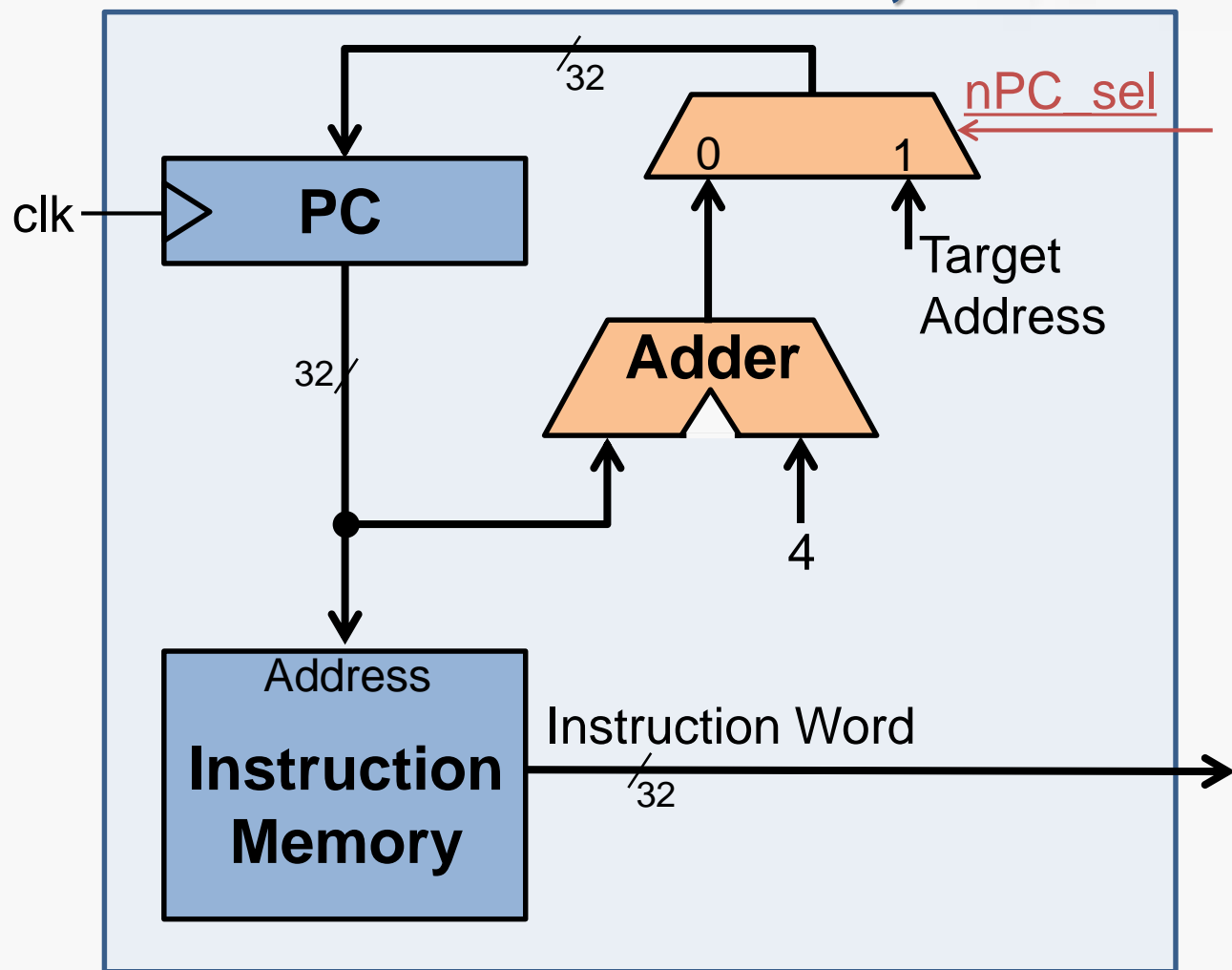
取指令

- 程序计数器（PC）的内容是指令的地址
- 用PC的内容作为地址，访问指令存储器获得指令编码

更新程序计数器（PC）

- 顺序执行时
 - $PC \leftarrow PC + 4$
- 发生分支时
 - $PC \leftarrow$ 分支目标的地址

Instruction Fetch Unit, IFU



建立数据通路的方法



④ 基本原则

- 根据指令需求，连接组件，建立数据通路

④ 指令的需求

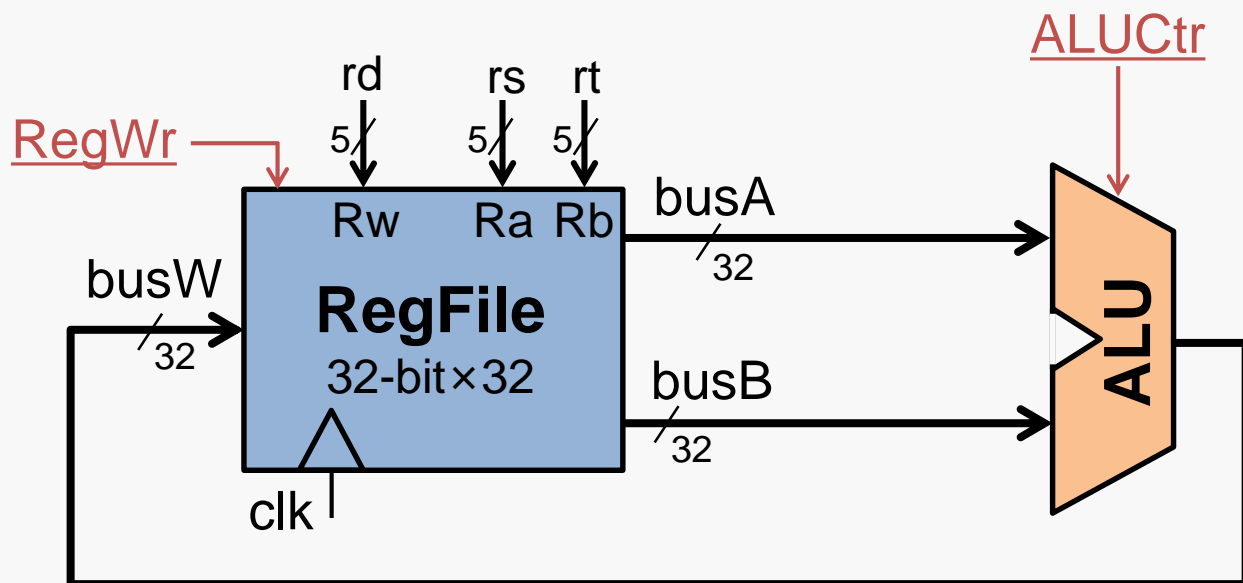
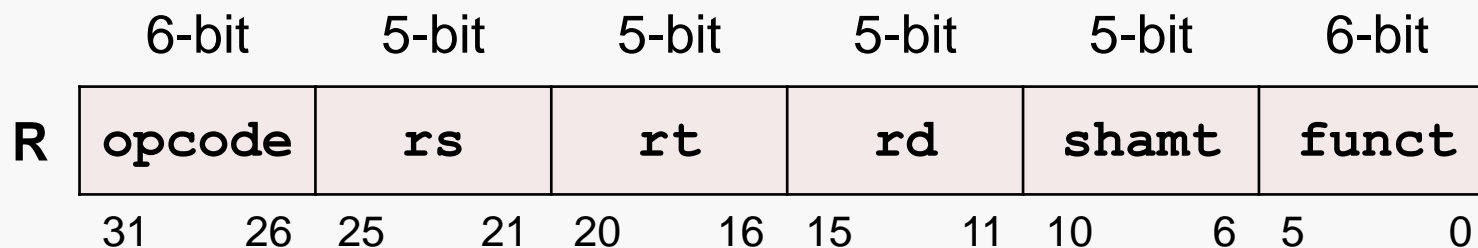
- 所有指令的共同需求
- **不同指令的不同需求**

加法和减法指令的需求

🎯 $R[rd] = R[rs] \text{ op } R[rt]$

◦ `addu rd, rs, rt`

`subu rd, rs, rt`

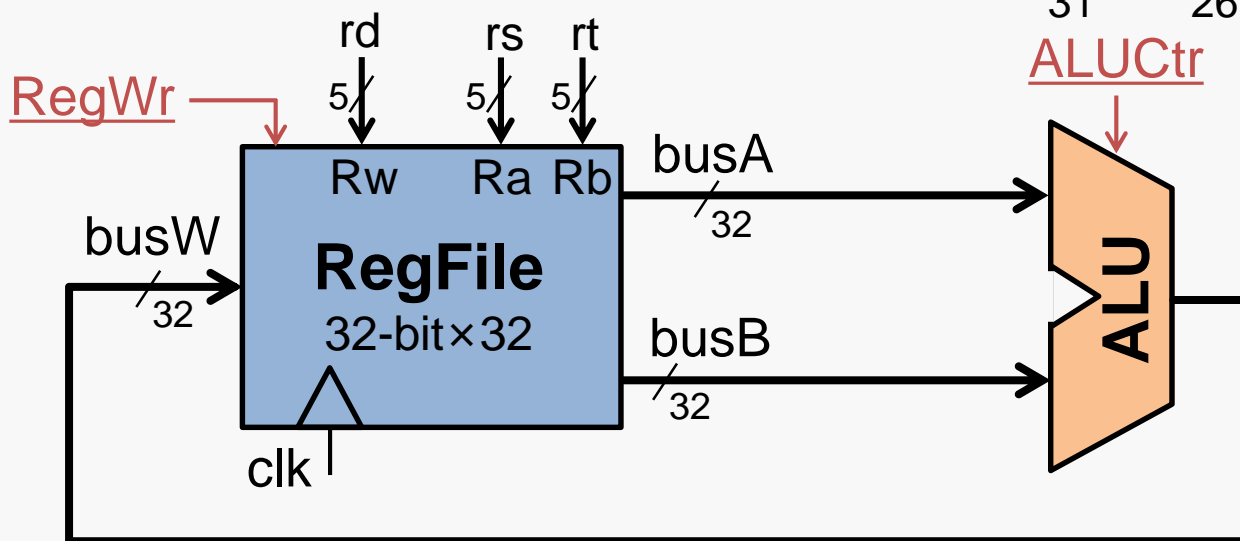
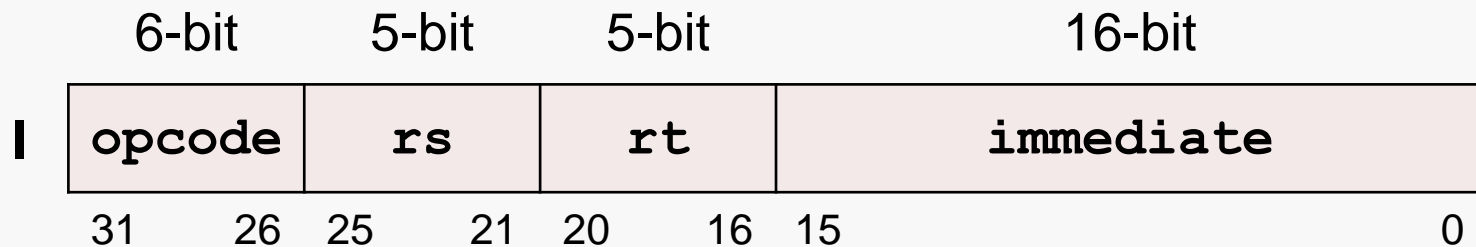


注：ALUCtr和RegWr是由指令译码生成的控制信号

逻辑运算指令的需求

🔍 $R[rt] = R[rs] \text{ op ZeroExt}[imm16]$

◦ `ori rt,rs,imm16`



问题1：目的寄存器是rt而非rd

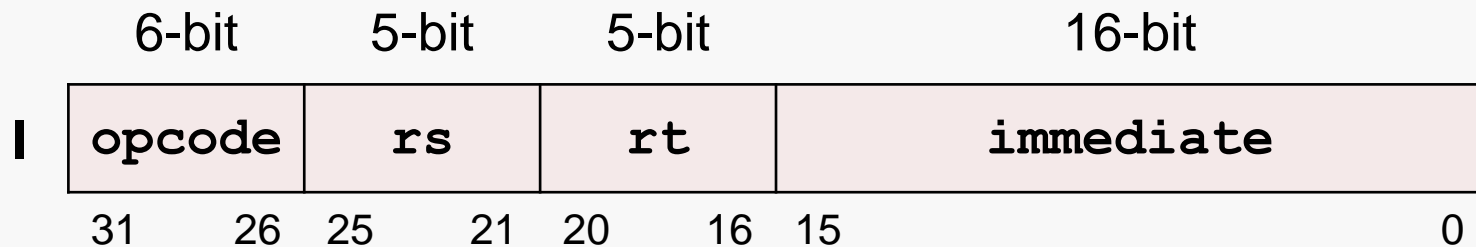
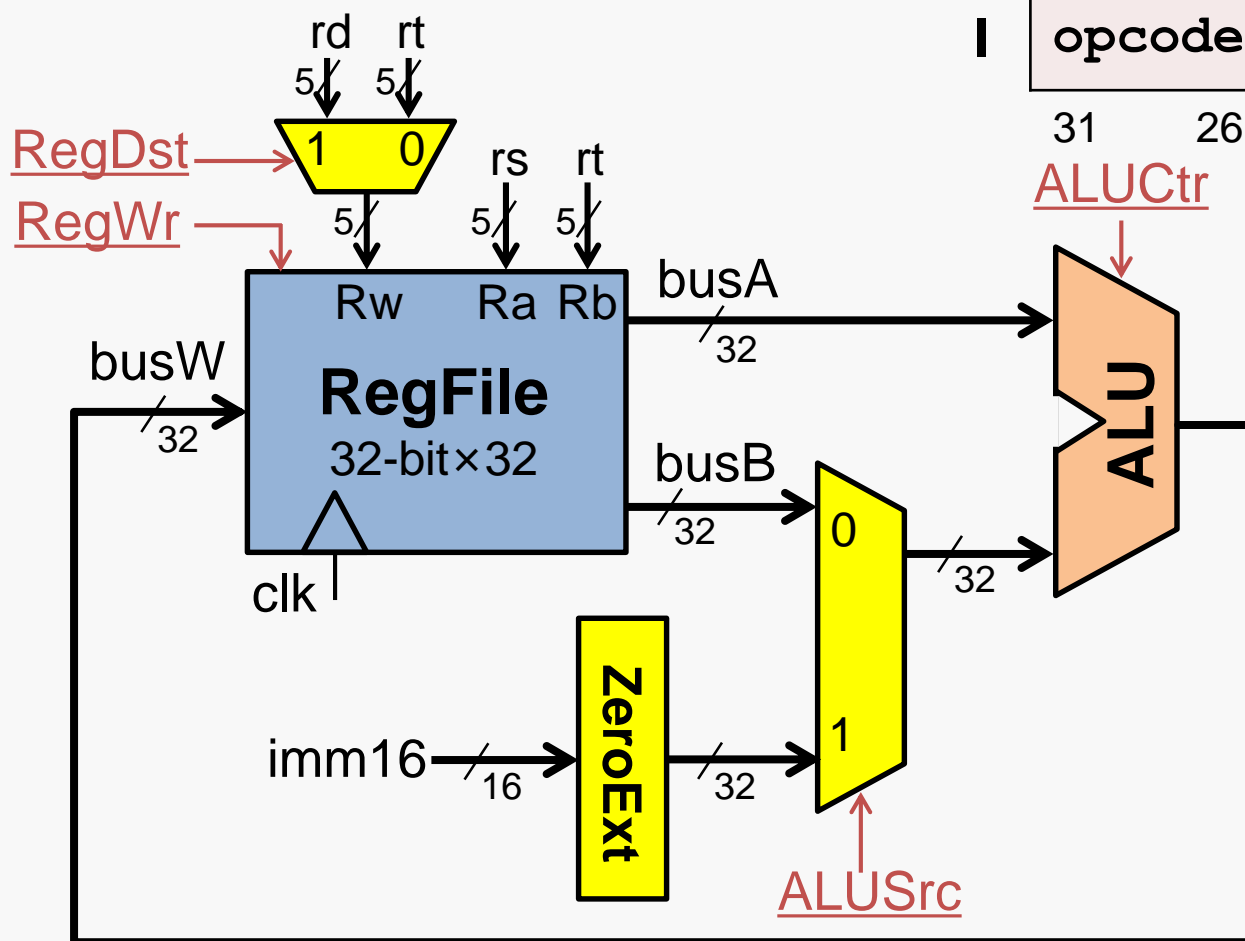
问题2：立即数是ALU的输入

问题3：立即数只有16位

逻辑运算指令的需求

🎯 $R[rt] = R[rs] \text{ op ZeroExt}[imm16]$

◦ `ori rt,rs,imm16`



问题1：目的寄存器是rt而非rd

问题2：立即数是ALU的输入

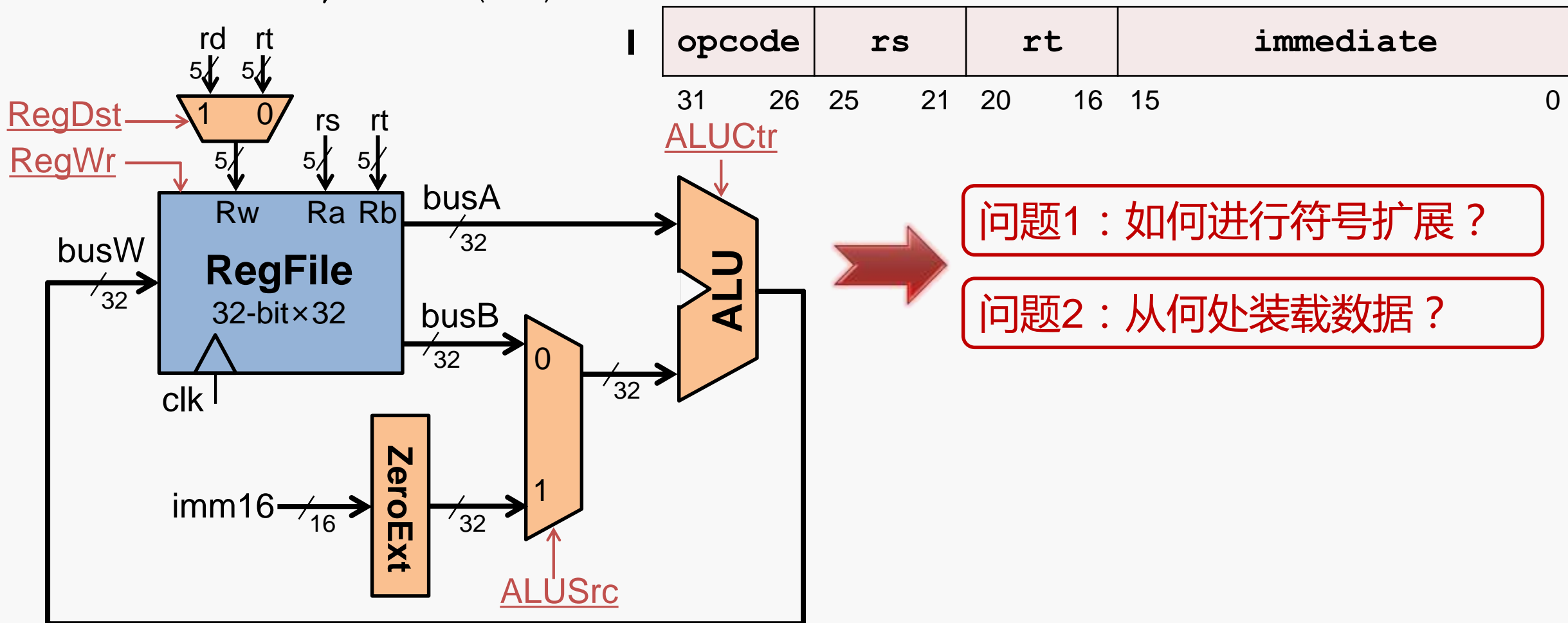
问题3：立即数只有16位

解决方案：增加两个多选器，增加一个零扩展部件

访存指令的需求 (Load)

$$R[\text{rt}] = \text{Mem}[R[\text{rs}] + \text{SignExt}[\text{imm16}]]$$

◦ lw rt, imm16(rs)

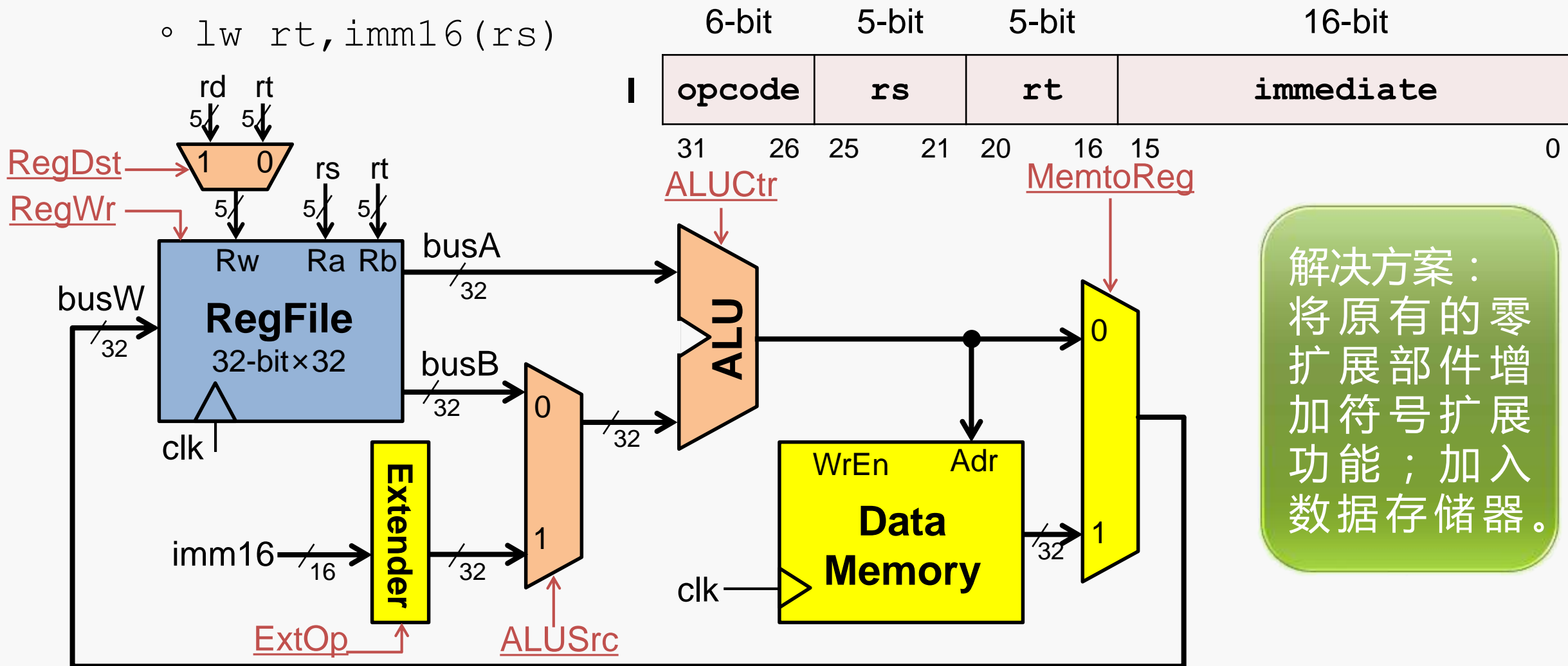




访存指令的需求 (Load)

$$R[rt] = \text{Mem}[R[rs] + \text{SignExt}[\text{imm16}]]$$

◦ lw rt, imm16(rs)

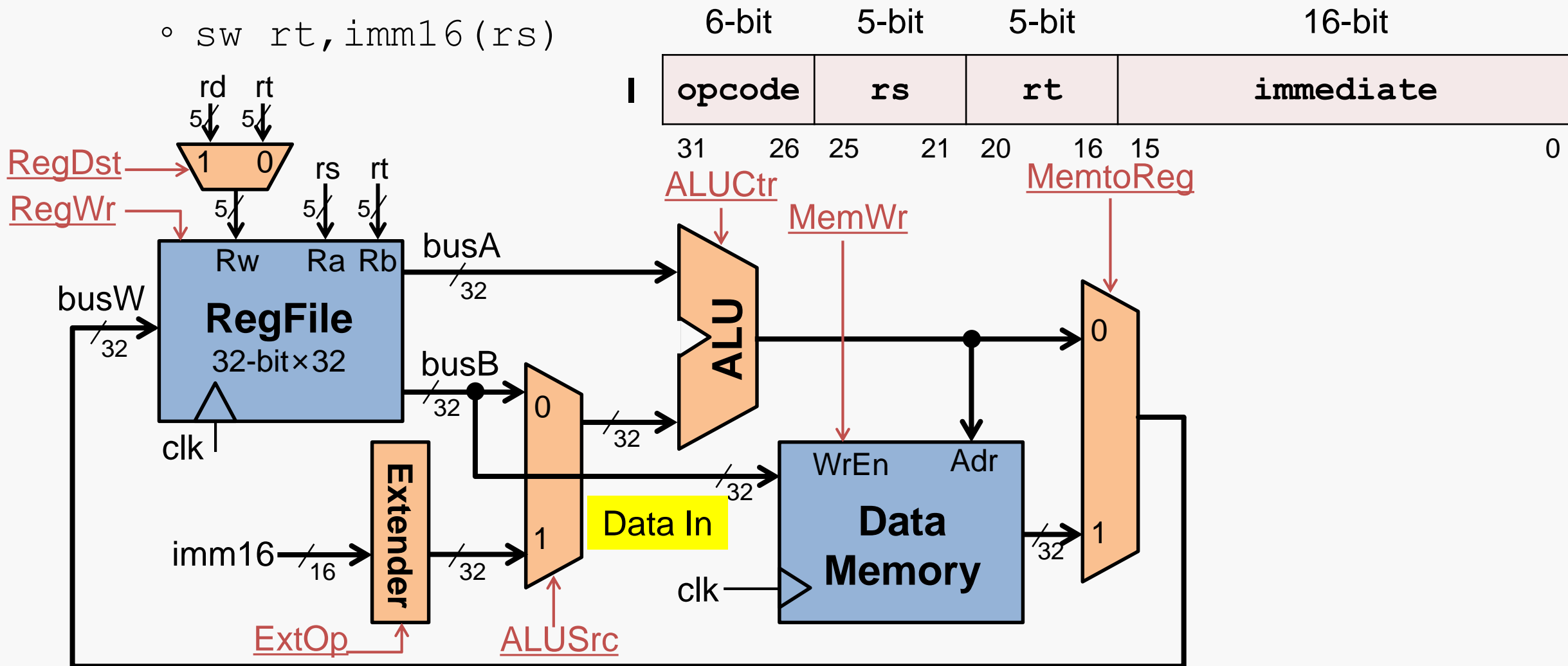


解决方案：
将原有的零扩展部件增加符号扩展功能；加入数据存储器。

访存指令的需求 (Store)

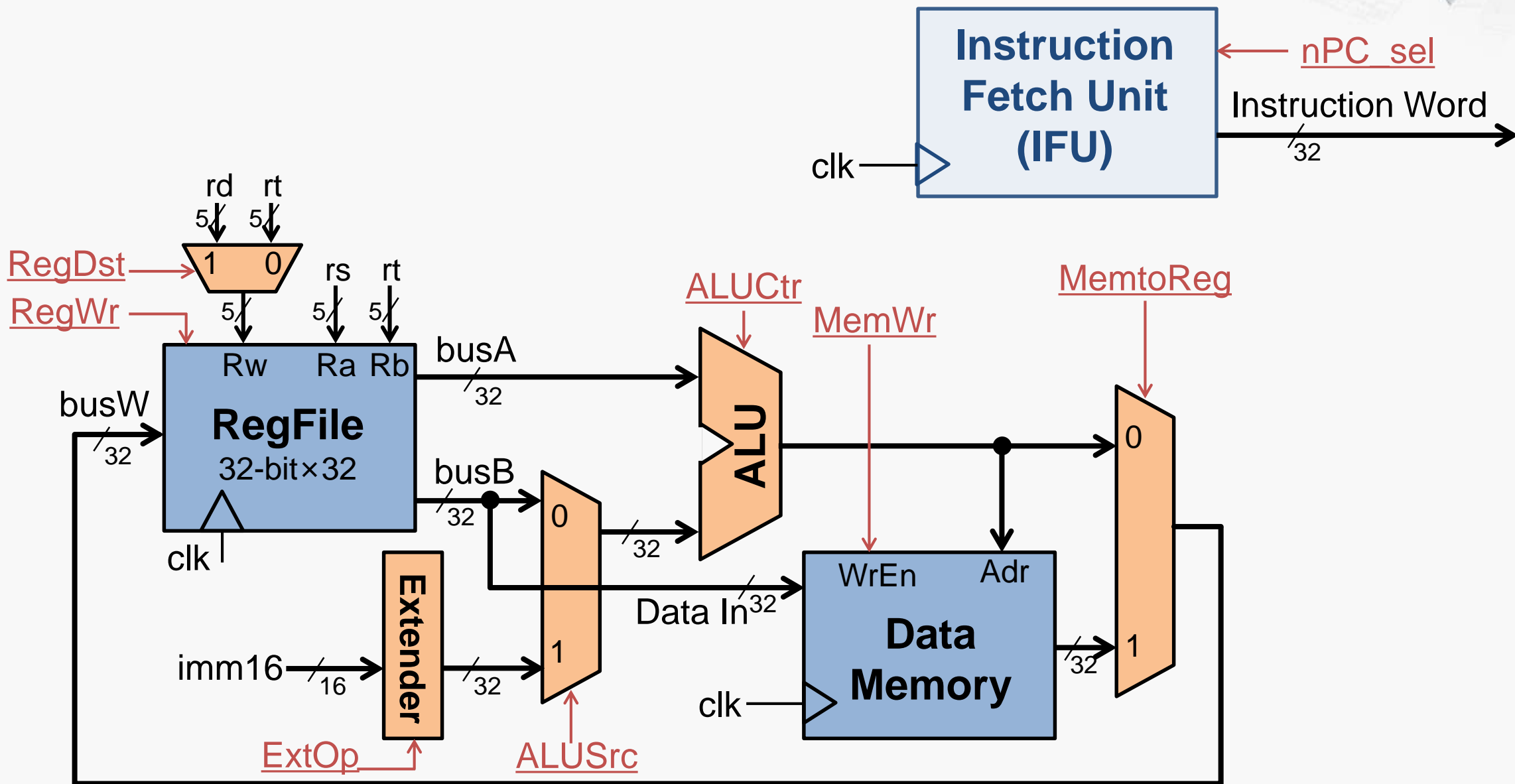
Mem[R[rs] + **SignExt**[imm16]] = R[**rt**]

◦ sw rt, imm16(rs)



数据通路初步完成

18年考的就是画此图，十几分，看了ppt轻松拿到必会！！



处理器的设计步骤



- ① 分析指令系统，得出对数据通路的需求 ✓
- ② 为数据通路选择合适的组件 ✓
- ③ 连接组件建立数据通路 ✓
- ④ 分析每条指令的实现，以确定控制信号
- ⑤ 集成控制信号，形成完整的控制逻辑

本节小结



数据通路的建立

北京大学·慕课
计算机组成
制作人：陆俊林

