

本节主题



处理器设计的主要步骤

北京大学·慕课
计算机组成
制作人：陆俊林



处理器的设计步骤

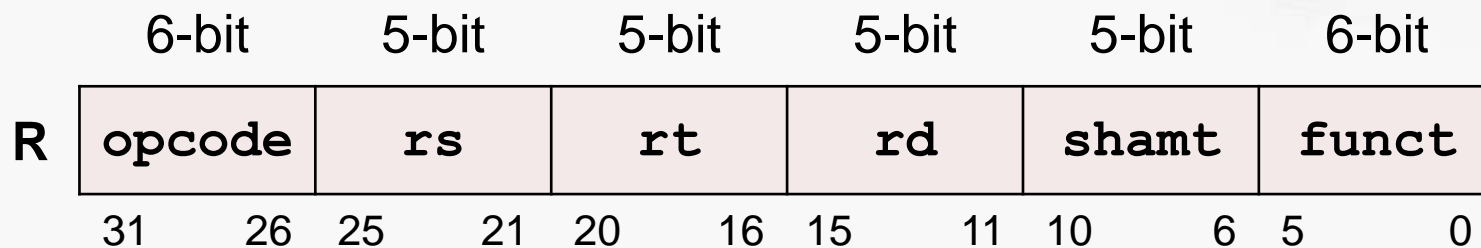


- ① 分析指令系统，得出对数据通路的需求
- ② 为数据通路选择合适的组件
- ③ 连接组件建立数据通路
- ④ 分析每条指令的实现，以确定控制信号
- ⑤ 集成控制信号，形成完整的控制逻辑

MIPS指令系统的简化版本

无符号加法和减法

- `addu rd,rs,rt`
- `subu rd,rs,rt`

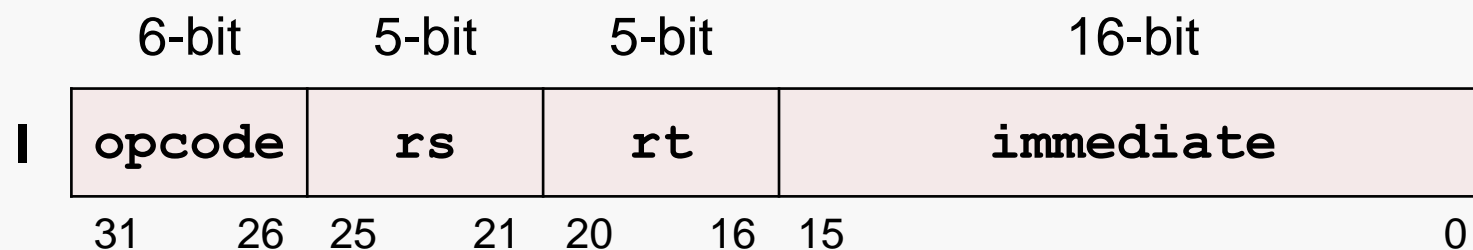


立即数的逻辑或

- `ori rt,rs,imm16`

装载和存储一个字 (32位)

- `lw rt,imm16(rs)`
- `sw rt,imm16(rs)`



条件分支

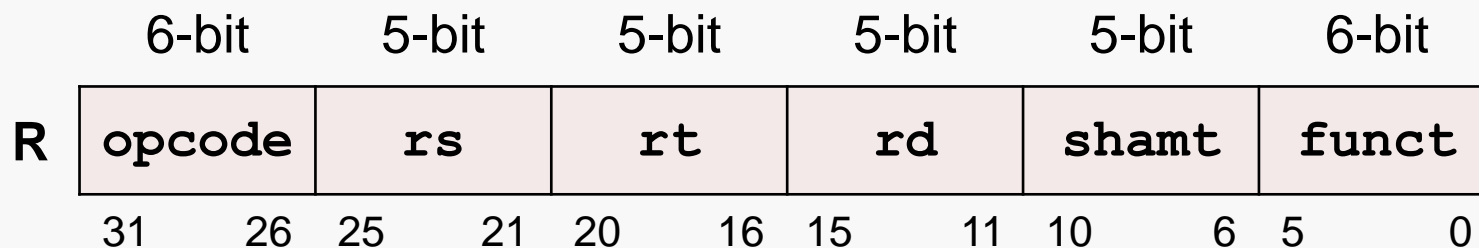
- `beq rs,rt,imm16`

指令的含义

需求：存放指令的存储器，可读，地址和数据均为32位

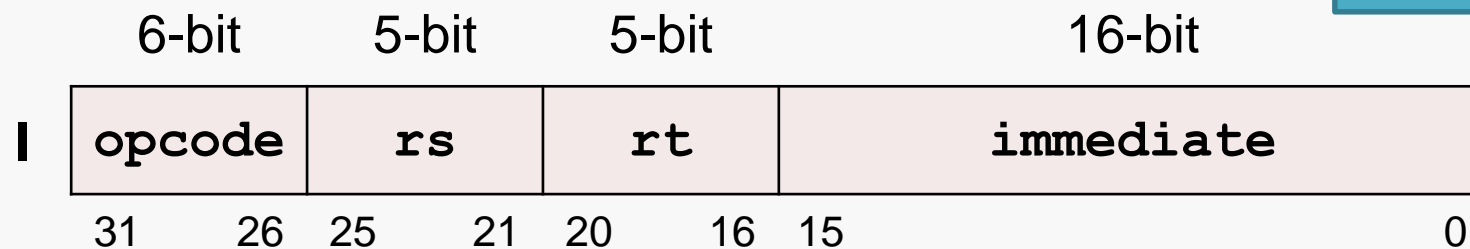
指令位域的分解

◦ R型指令： $\{op, rs, rt, rd, shamt, funct\} \leftarrow MEM[PC]$



◦ I型指令： $\{op, rs, rt, Imm16\} \leftarrow MEM[PC]$

需求：存放指令地址的32位寄存器



指令的含义

指令的操作

◦ ADDU $R[rd] \leftarrow R[rs] + R[rt]; PC \leftarrow PC+4$

◦ SUBU $R[rd] \leftarrow R[rs] - R[rt]; PC \leftarrow PC+4$

需求：改写一个寄存器的内容（rd或rt）

需求：一组存放数据的32位通用寄存器

需求：同时读取两个寄存器的内容（rs和rt）

◦ ORI $R[rt] \leftarrow R[rs] \mid \text{zero_ext}(\text{Imm16}); PC \leftarrow PC+4$

需求：运算的操作数可以是寄存器或者扩展后的立即数

需求：提供加、减、逻辑或三种功能的运算器

需求：将16位立即数扩展到32位（零扩展）

指令的含义

指令的操作

- LOAD $R[rt] \leftarrow \text{MEM}[R[rs] + \text{sign_ext}(\text{Imm16})]; PC \leftarrow PC + 4$
- STORE $\text{MEM}[R[rs] + \text{sign_ext}(\text{Imm16})] \leftarrow R[rt]; PC \leftarrow PC + 4$

需求：存放数据的存储器，可读写，地址和数据均为32位

需求：将16位立即数扩展到32位（符号扩展）

- BEQ $\text{if } (R[rs] == R[rt])$
 then $PC \leftarrow PC + 4 + (\text{sign_ext}(\text{Imm16}) || 00)$
 else $PC \leftarrow PC + 4$

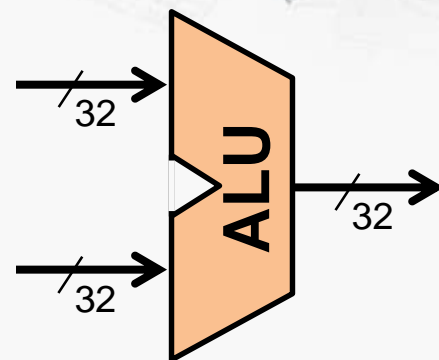
需求：比较两个数，判断是否相等

需求：PC寄存器支持两种自增方式，加4 或 加一个立即数

指令系统的需求

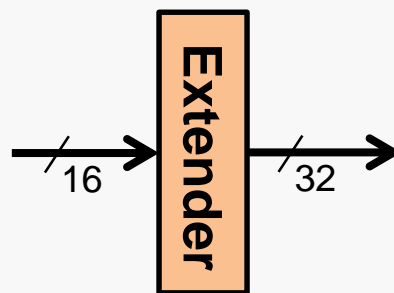
❶ 算术逻辑单元 (ALU)

- 运算类型：加、减、或、比较相等
- 操作数：2个32位的数，来自寄存器 或 扩展后的立即数



❷ 立即数扩展部件

- 将一个16立即数扩展为32位数
- 扩展方式：零扩展、符号扩展



❸ 程序计数器 (PC)

- 一个32位的寄存器
- 支持两种加法：加4 或 加一个立即数



指令系统的需求



寄存器堆

- 每个寄存器为32位宽，共32个
- 支持读操作：rs 和 rt
- 支持写操作：rt 或 rd
- 注：这称为“两读一写”的寄存器堆

存储器

- 一个只读的指令存储器，地址和数据均为32位
- 一个可读写的数据存储器，地址和数据均为32位
- 注：这两个存储器实际对应了CPU中的指令和数据高速缓存（Cache）

存储组件：寄存器堆

内部构成

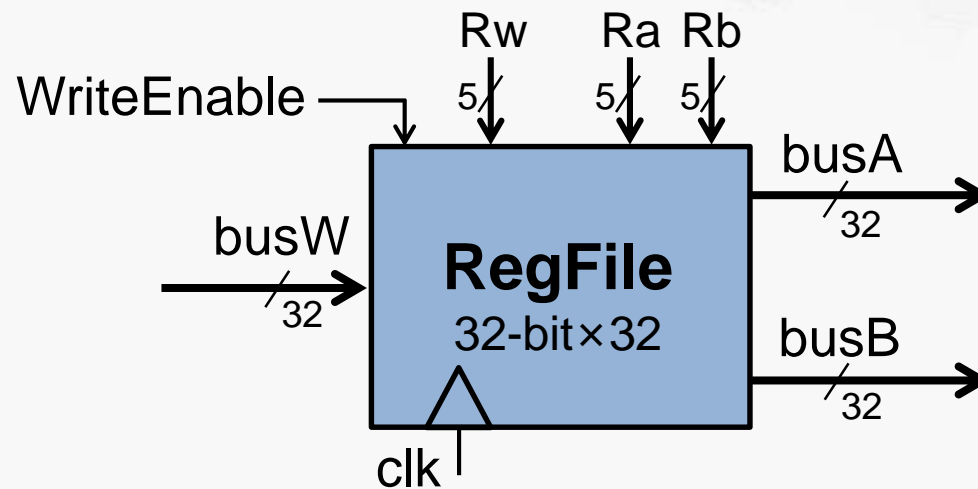
- 32个32位的寄存器

数据接口信号

- busA , busB : 两组32位的数据输出
- busW : 一组32位的数据输入

读写控制

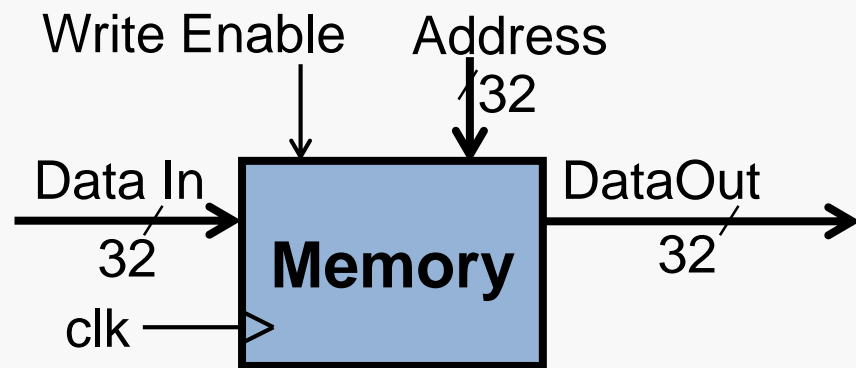
- Ra(5位) : 选中对应编号的寄存器，将其内容放到busA
- Rb(5位) : 选中对应编号的寄存器，将其内容放到busB
- Rw(5位) : 选中对应编号的寄存器，在时钟信号（clk）的上升沿，如果写使能信号有效（WriteEnable==1），将busW的内容存入该寄存器
- 注：寄存器堆的读操作不受时钟控制



存储组件：存储器

数据接口信号

- Data In : 32位的数据输入信号
- Data Out : 32位的数据输出信号



读写控制

- Address : 32位的地址信号。该信号指定一个存储单元，将其内容送到数据输出信号
- Write Enable : 写使能信号。在时钟信号 (clk) 的上升沿，如果写使能信号有效 (为1)，将数据输入信号的内容存入地址信号指定存储单元
- 注：存储器的读操作不受时钟控制

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



处理器设计的主要步骤

北京大学·慕课
计算机组成
制作人：陆俊林

