本节主题

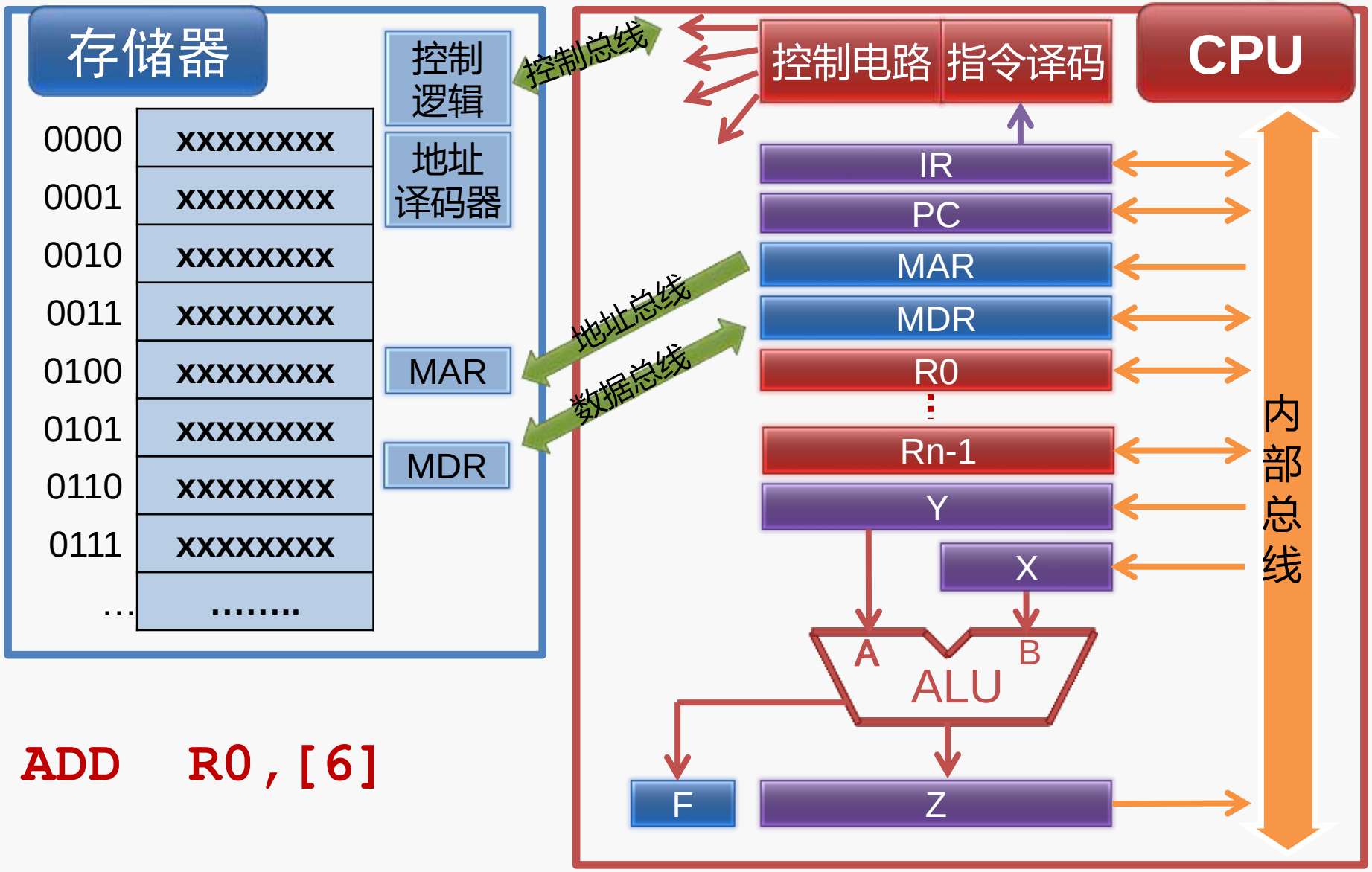
算术运算和逻辑运算

北京大学 · 慕课

计算机组成

制作人：陆俊林

计算机结构的简化模型（模型机）



ADD R0,[6]

加法指令的编码示例（1）

- **add $8,$9,$10** #$8=$9+$10
  - 查指令编码表得到：

  opcode = 0 ，funct = 20_{hex}

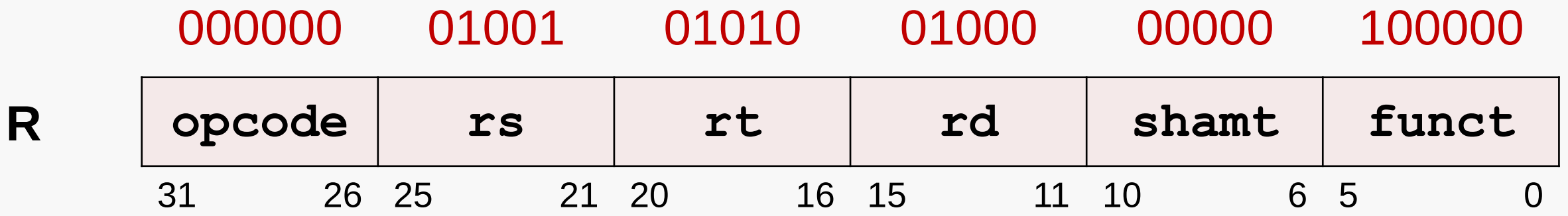
  shamt = 0（非移位指令）
  - 根据指令操作数得到：

  rd = 8（目的操作数），rs = 9（第一个源操作数）

  rt = 10（第二个源操作数）

C语言程序

```
int f,g,h;
...
//f→$8,g→$9,h→$10
f = g + h;
```



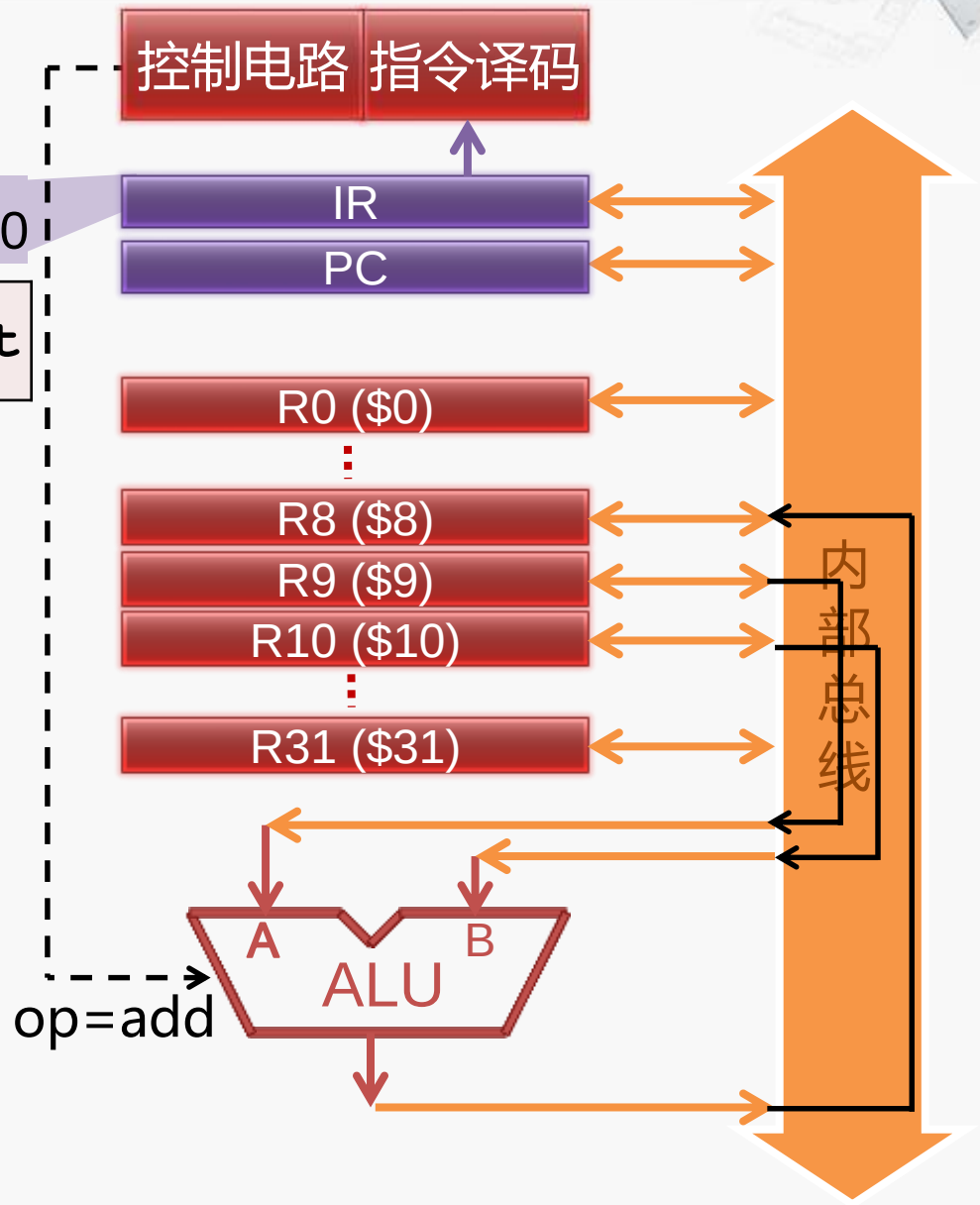
加法运算示例

000000 01001 01010 01000 00000 100000

R

opcode	rs	rt	rd	shamt	funct

```
add $8,$9,$10
```



算术运算指令（MIPS Core Instruction Set）

- R型
  - `add rd,rs,rt` `# R[rd]=R[rs]+R[rt]` (1)
  - `addu rd,rs,rt` `# R[rd]=R[rs]+R[rt]`
  - `sub rd,rs,rt` `# R[rd]=R[rs]-R[rt]` (1)
  - `subu rd,rs,rt` `# R[rd]=R[rs]-R[rt]`

(1) May cause overflow exception

加法指令的编码示例（2）

- **addi $21,$22,-50** # $21=$22+(-50)
  - 查指令编码表得到：

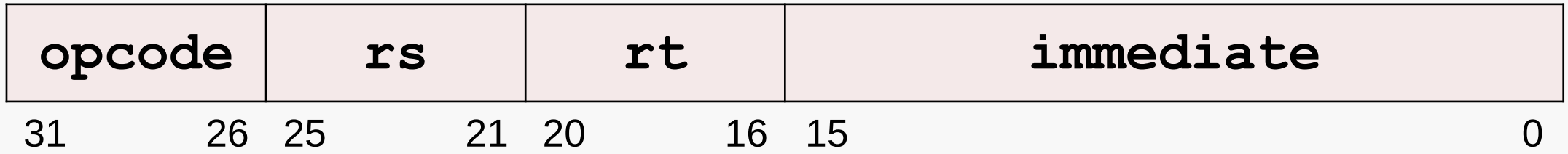
  opcode = 8
  - 分析指令得到：

  rs = 22（源操作数寄存器编号）

  rt = 21（目的操作数寄存器编号）

  immediate = -50（立即数）

001000 10110 10101 1111 1111 1100 1110

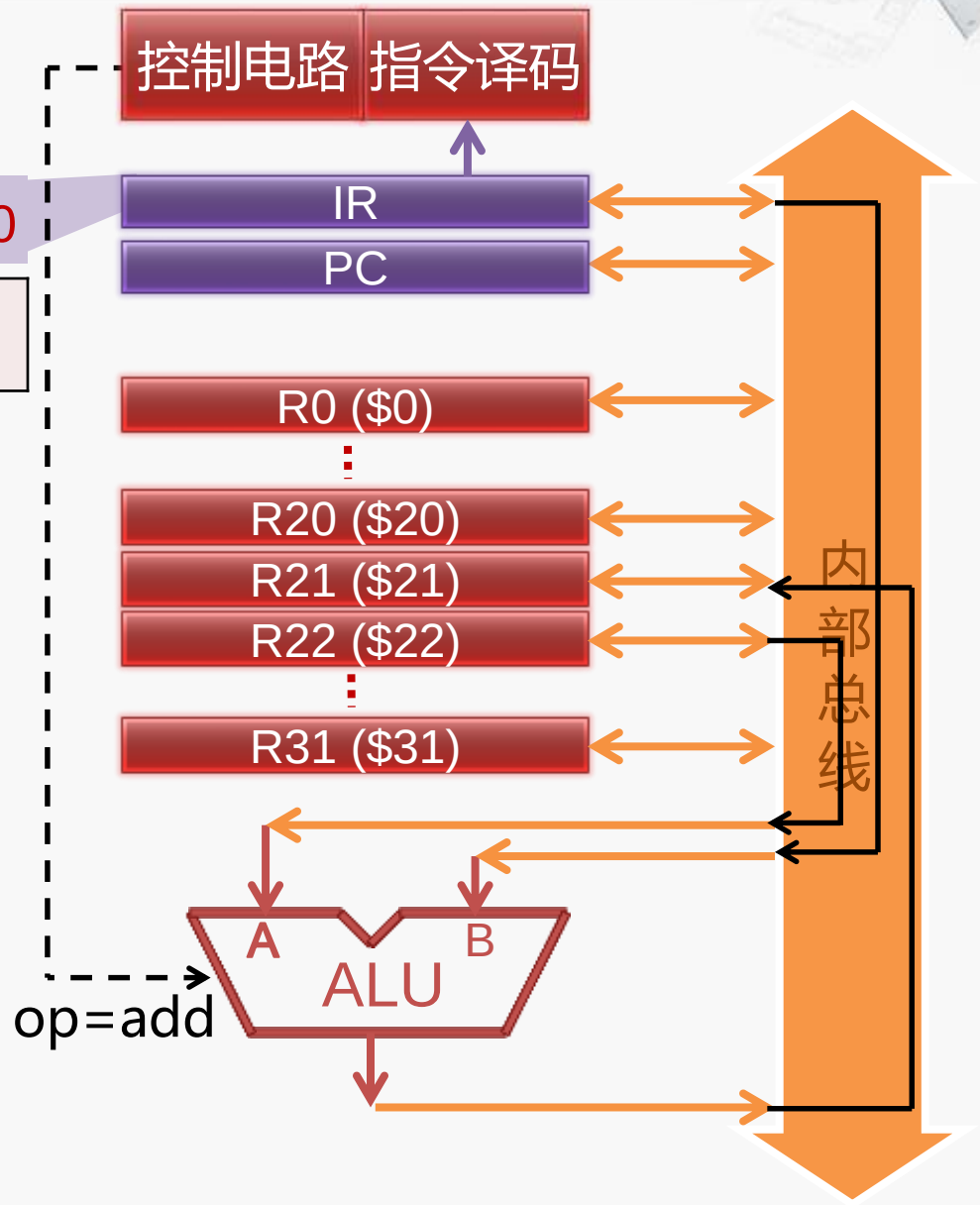


加法运算示例

001000 10110 10101 1111 1111 1100 1110

opcode	rs	rt	immediate

addi $21,$22,-50



算术运算指令（MIPS Core Instruction Set）

- R型
  - `add rd,rs,rt` `# R[rd]=R[rs]+R[rt]` (1)
  - `addu rd,rs,rt` `# R[rd]=R[rs]+R[rt]`
  - `sub rd,rs,rt` `# R[rd]=R[rs]-R[rt]` (1)
  - `subu rd,rs,rt` `# R[rd]=R[rs]-R[rt]`
- I型
  - `addi rt,rs,imm` `# R[rt]=R[rs]+SignExtImm(1,2)`
  - `addiu rt,rs,imm` `# R[rt]=R[rs]+SignExtImm(2)`

(1) May cause overflow exception

(2) SignExtImm={ 16{imm[15]}, imm }

逻辑运算指令（MIPS Core Instruction Set）

R型

- `and rd,rs,rt` `# R[rd]=R[rs]&R[rt]`
- `or rd,rs,rt` `# R[rd]=R[rs]|R[rt]`
- `nor rd,rs,rt` `# R[rd]=~(R[rs]|R[rt])`

I型

- `andi rt,rs,imm` `# R[rt]=R[rs]&ZeroExtImm(3)`
- `ori rt,rs,imm` `# R[rt]=R[rs]|ZeroExtImm(3)`

(3) ZeroExtImm={ 16{1'b0}, imm }

逻辑“与”指令的编码示例

- **and $8,$9,$10** #$8=$9&$10
  - 查指令编码表得到：

  opcode = 0 ，funct = 24_{hex}

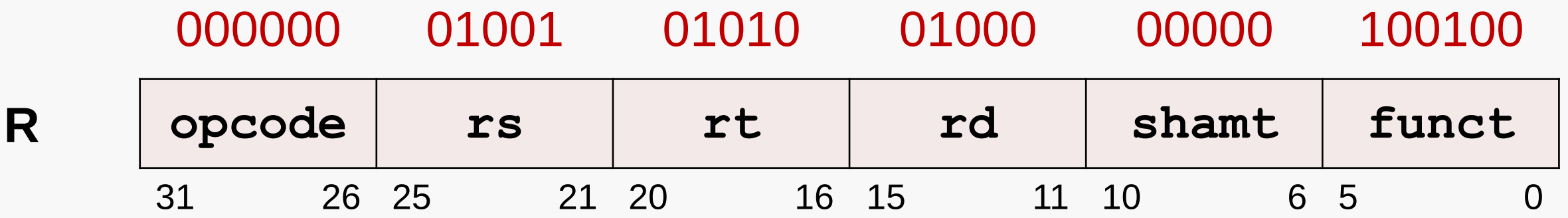
  shamt = 0（非移位指令）
  - 根据指令操作数得到：

  rd = 8（目的操作数），rs = 9（第一个源操作数）

  rt = 10（第二个源操作数）

C语言程序

```
int f,g,h;
...
//f→$8,g→$9,h→$10
f = g & h;
```



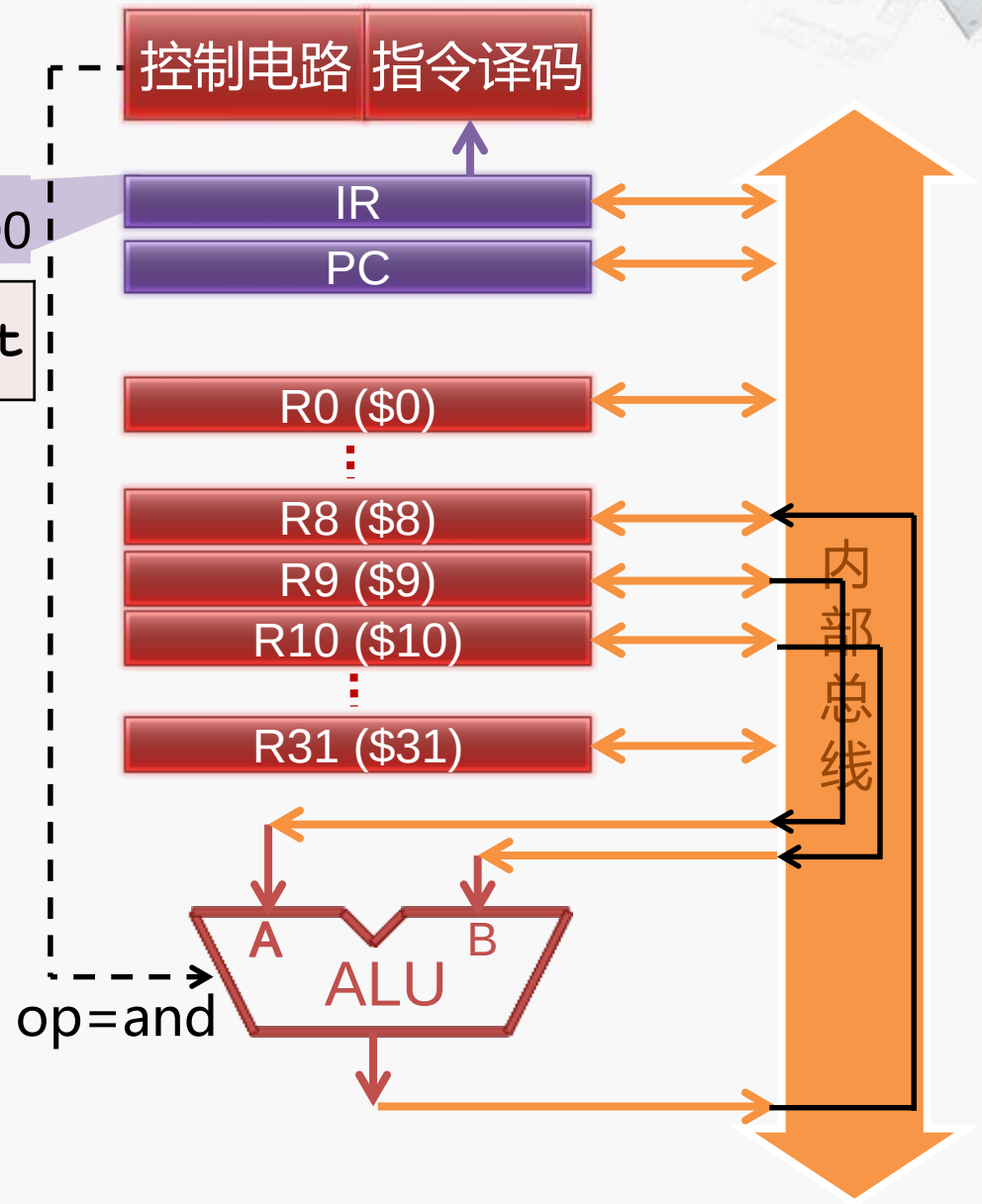
逻辑“与”运算示例

000000 01001 01010 01000 00000 100100

R

opcode	rs	rt	rd	shamt	funct

```
and $8,$9,$10
```



算术逻辑运算的需求

- 算术运算
  - 两个32-bit数的加法，结果为一个32-bit数
  - 两个32-bit数的减法，结果为一个32-bit数
  - 检查加减法的结果是否溢出

- 逻辑运算
  - 两个32-bit数的“与”操作，结果为一个32-bit数
  - 两个32-bit数的“或”操作，结果为一个32-bit数
  - 两个32-bit数的“或非”操作，结果为一个32-bit数

本节小结

算术运算
和逻辑运算

北京大学 · 慕课
计算机组成
制作人：陆俊林