

# 处理器的设计步骤





# 处理器设计

## □ 能实现几条简单指令的处理器

算术逻辑运算指令: `add, sub, ori`

访问存储器: `lw, sw`

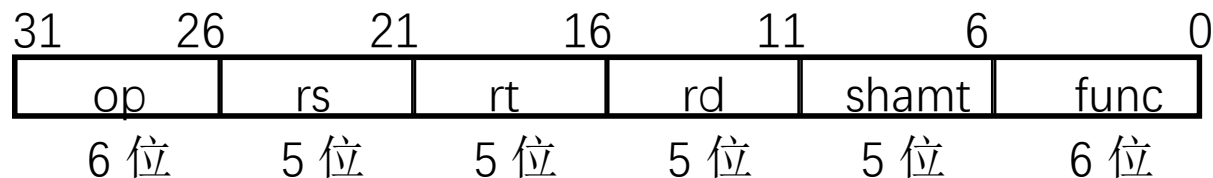
控制流指令: `beq, j`



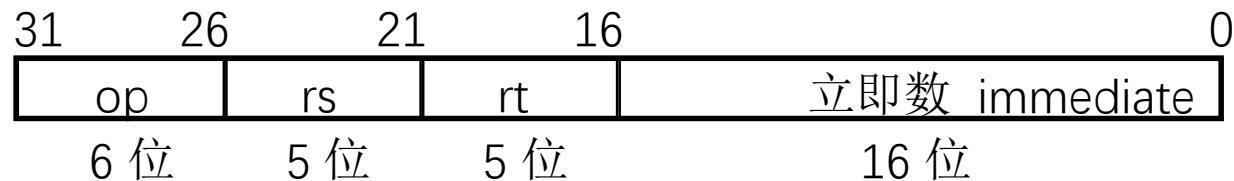
# MIPS指令的子集

- 加法与减法
  - `add rd, rs, rt`
  - `sub rd, rs, rt`
- OR 立即数:
  - `ori rt, rs, imm16`
- 读内存 和 写内存
  - `lw rt, rs, imm16`
  - `sw rt, rs, imm16`
- 条件转移:
  - `beq rs, rt, imm16`
- 无条件转移:
  - `j target`

R 型:



I 型:



J 型:





## 设计处理器的五个步骤

1. 分析指令系统，得出对数据通路的需求
2. 选择数据通路上合适的组件
3. 连接组件构成数据通路
4. 分析每一条指令的实现，以确定控制信号
5. 集成控制信号，完成控制逻辑



# 分析各条指令的数据通路

## 指令的执行过程

取指令

指令译码

指令执行

ADD       $R[rd] \leftarrow R[rs] + R[rt]; \quad PC \leftarrow PC + 4$

ORi       $R[rt] \leftarrow R[rs] \mid \text{zero\_ext}(\text{Imm16}); \quad PC \leftarrow PC + 4$

LOAD     $R[rt] \leftarrow \text{MEM}[R[rs] + \text{sign\_ext}(\text{Imm16})]; \quad PC \leftarrow PC + 4$

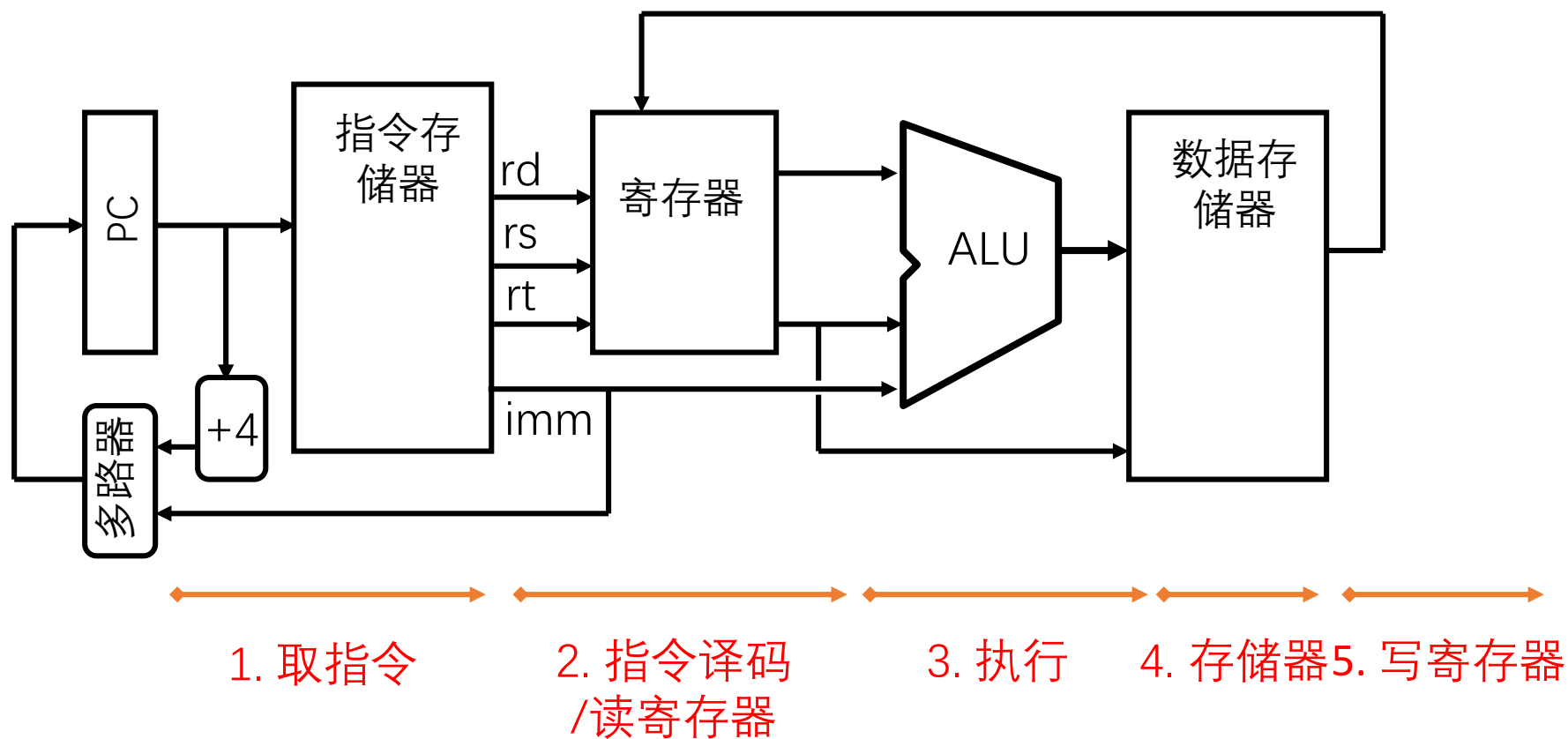
STORE    $\text{MEM}[R[rs] + \text{sign\_ext}(\text{Imm16})] \leftarrow R[rt]; \quad PC \leftarrow PC + 4$

BEQ      if (  $R[rs] == R[rt]$  ) then  $PC \leftarrow PC + 4 + \text{sign\_ext}(\text{Imm16}) \parallel 00$   
                                         else  $PC \leftarrow PC + 4$

JUMP       $PC \leftarrow (PC + 4[31-28], I25-0) \parallel 00$



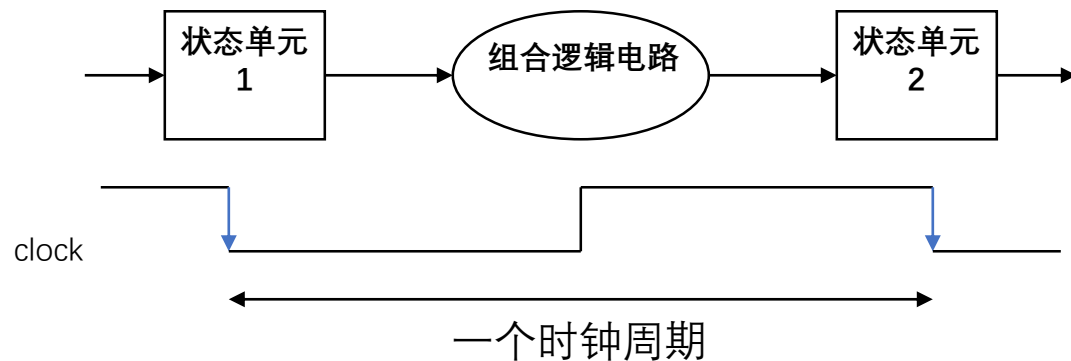
## 数据通路的大致需求





# 单周期处理器

- 单周期处理器：一个时钟周期完成一条指令：



边沿触发

状态转换发生在时钟边沿

状态单元：

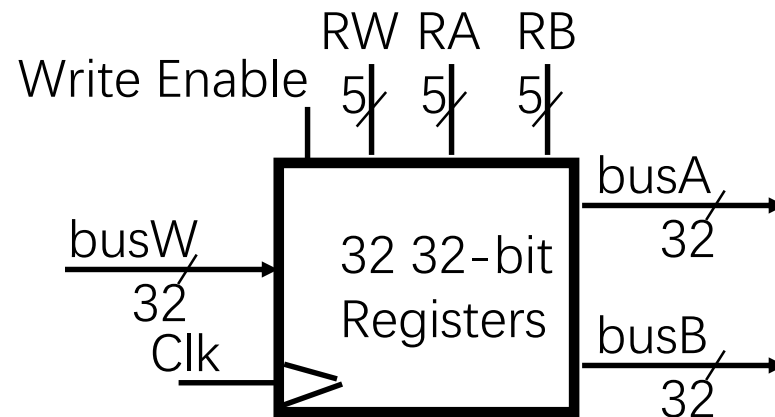
例如：寄存器、存储器

- 读状态单元的内容 -> 通过组合逻辑电路实现指令的功能 -> 将结果写入一个或多个状态单元
- 状态单元每一周期更新一次，是否更新，需要一个显式的控制信号
- 寄存器、存储器：在时钟边沿来到、写允许信号有效时才更新状态

# 选择数据通路上合适的组件

## 状态单元：寄存器文件

- 时钟输入(CLK)
  - 改变寄存器状态需要时钟边沿触发
- 32个寄存器
  - 两个32位输出： busA 、 busB
  - 一个32位输入： busW
- 寄存器选通
  - RA（5位）： 选通RA指定的寄存器
  - RB（5位）： 选通RB 指定的寄存器
    - 读操作，看做一个组合电路模块的实现
    - RA 、 RB 有效 => busA 、 busB 有效
  - RW（5位）： 选通Rw指定的寄存器
    - 写操作： CLK边沿触发
    - 当Write Enable 为1时，将busW 端口上的数据写入Rw指定的寄存器

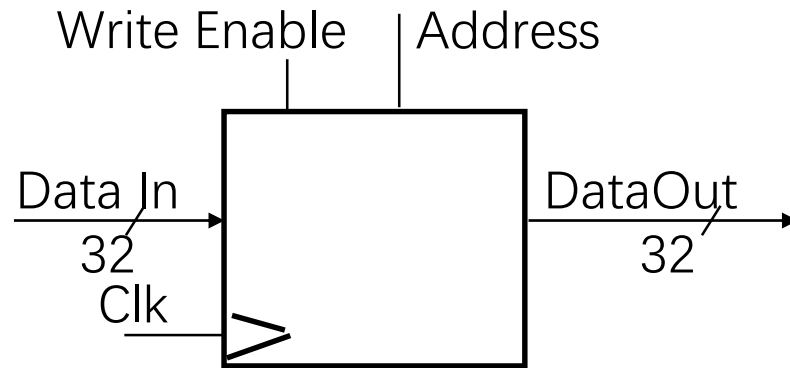




# 选择数据通路上合适的组件

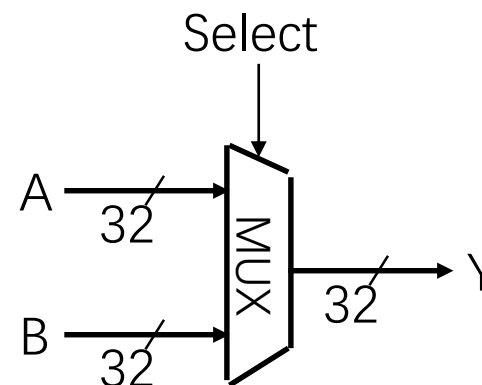
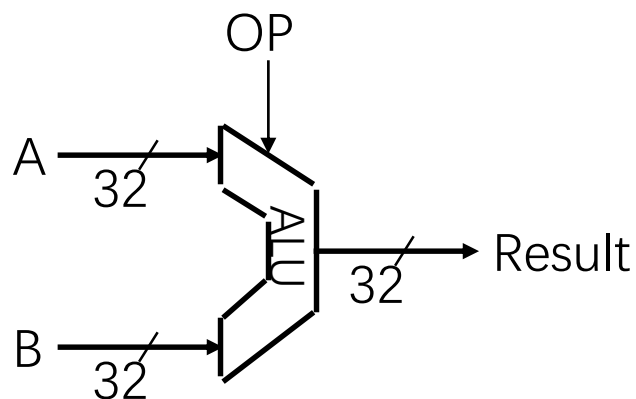
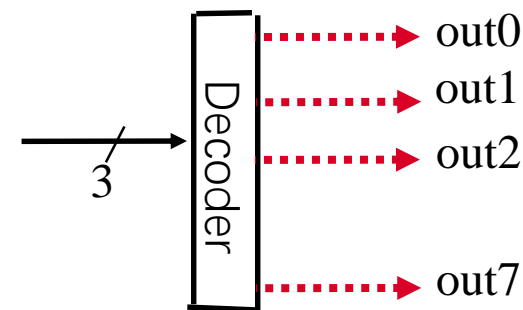
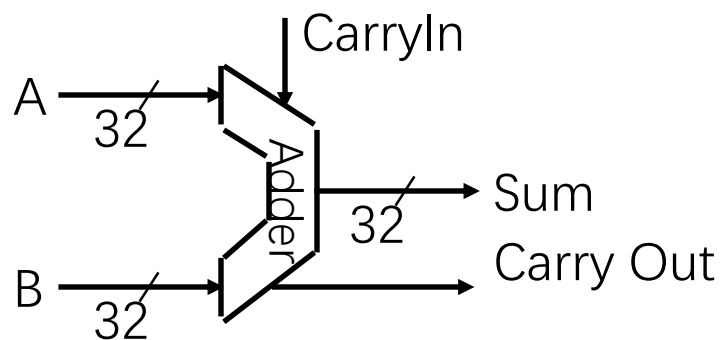
## 状态单元：存储器

- 时钟输入 (CLK)
  - 改变存储器状态需要时钟边沿触发
- 存储器总线
  - 32位数据输入总线: Data In
  - 32位数据输出总线: Data Out
- 读写操作
  - 读操作，看做一个组合电路模块的实现
    - 一定时间内完成从“地址信号有效” (Address) => “数据输出” Data Out
  - 写操作：时序电路
    - CLK边沿触发
    - Write Enable = 1: 将Data In的输入写入Address选中的那个字



# 选择数据通路上合适的组件

- 组合逻辑单元





## 小结

设计处理器的前两个步骤：

1. 分析指令系统， 得出对数据通路的需求
2. 选择数据通路上合适的组件