第一讲 中央处理器基础

翟象平

电邮: blueicezhaixp@nuaa.edu.cn

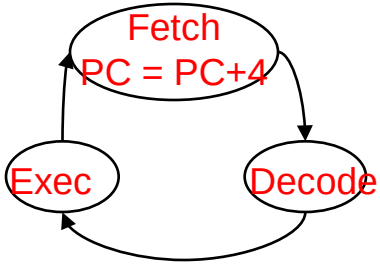
个人主页: http://cyber.nuaa.edu.cn

回顾：MIPS（RISC）设计原则

- 简洁有助于归整
  - 定长指令字
  - 指令格式少
  - opcode总是前6位
- 越小越快
  - 有限指令集
  - 有限的寄存器
  - 有限的寻址方式
- 加速大概率事件
  - 面向寄存器的算术运算指令
  - 允许指令包含立即数
- 优秀的设计需要最优的权衡
  - 3种指令格式

中央处理器：数据通路+控制

- 简洁的MIPS实现
  - 内存访问指令：lw，sw
  - 算术逻辑指令：add, sub, and, or, slt
  - 控制流指令： beq, j
- 常用实现方式
  - PC提供指令地址从内存读取指令，并更新PC
  - 解码指令
  - 执行指令
- 所有指令（j 除外）访问寄存器后使用ALU

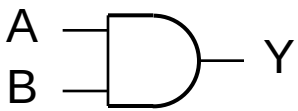


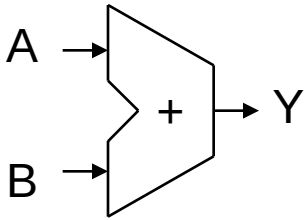
逻辑设计基础

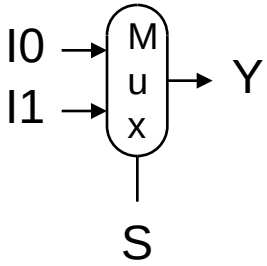
- 信息采用二进制编码
  - 低电平为 0，高电平为 1
  - 每 比特 一条 电线
  - 多比特信息在多电线上编码（总线）
- 操作元件-组合电路
  - 操作数据
  - 输出可看作为基于输入的函数 Output=F(Input)
- 状态元件-时序电路
  - 存储信息

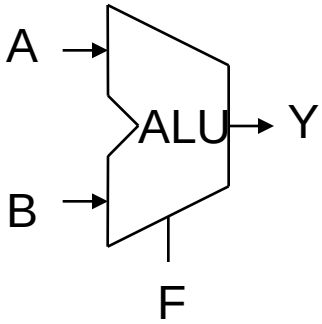
操作元件

- 与门
  - Y = A & B
- 加法
  - Y = A + B
- 多路选择
  - Y = S ? I1 : I0
- 算术逻辑单元
  - Y = F(A,B)



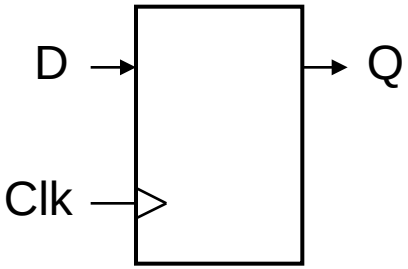


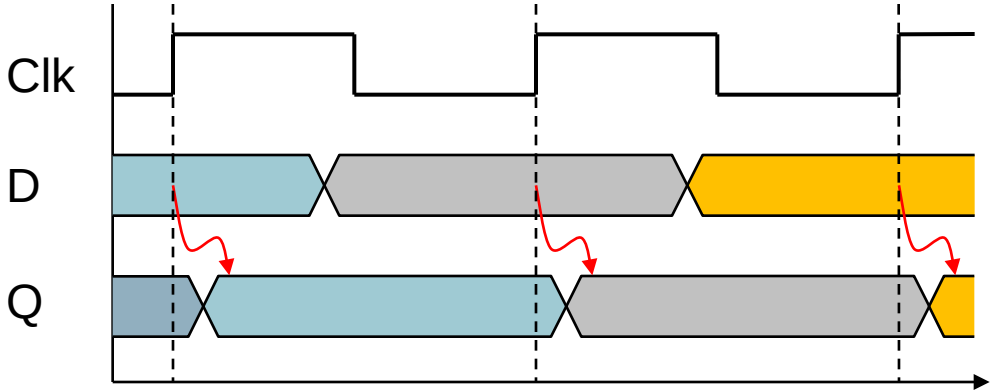




状态元件

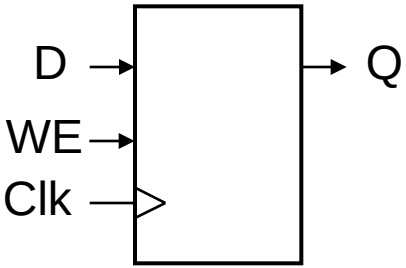
- 寄存器：在电路里存储数据
  - 用时钟信号控制什么时候对存储的数据进行更新
  - 边沿触发：当时钟信号改变的时候更新
    - 上跳沿：0→1
    - 下跳沿：1→0
  - D触发器是实现寄存器的一种方式

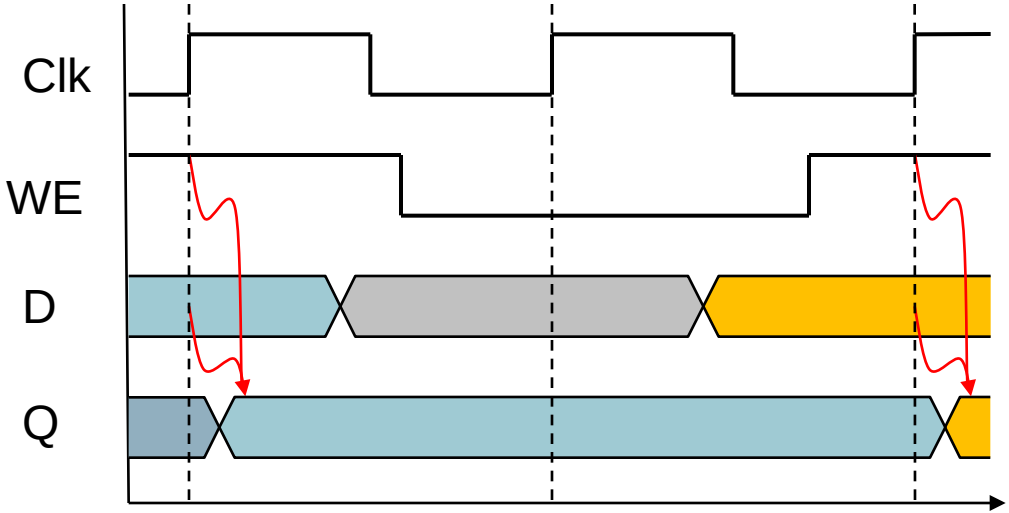




状态元件

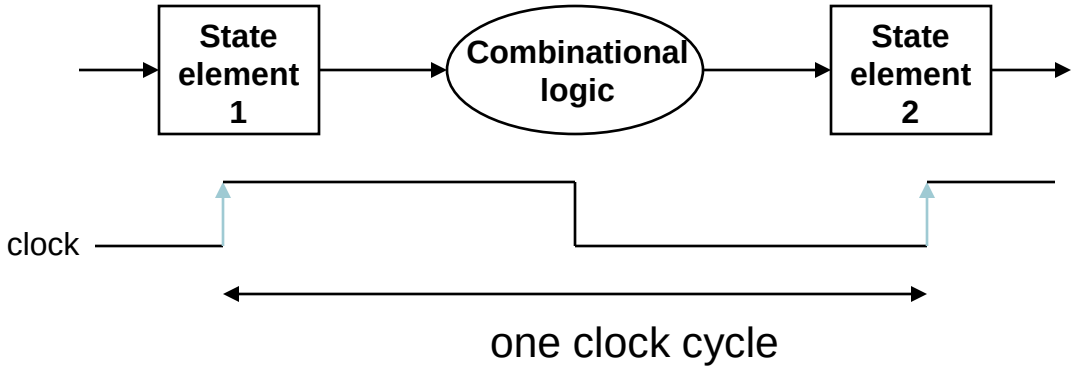
- 带写控制的寄存器
  - 只有在写使能信号有效时才在时钟边沿更新数据





时序控制

- 时序控制决定状态元件中的数据什么时候有效、以及什么时候稳定
  - 边沿触发：所有状态在时钟边沿改变
- 经典时序
  - 状态元件读数据→组合电路传递值→写结果至状态元件
- 写使能信号
  - 写使能+时钟边沿



Q & A

THANKS