

# 指令流水线的基本原理



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



# 流水线处理器



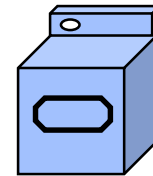
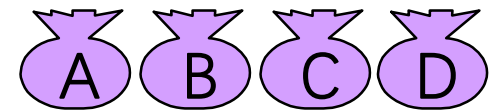
- 流水线的原理和特点
- 指令执行的五个阶段
- MIPS五阶段流水线的原理

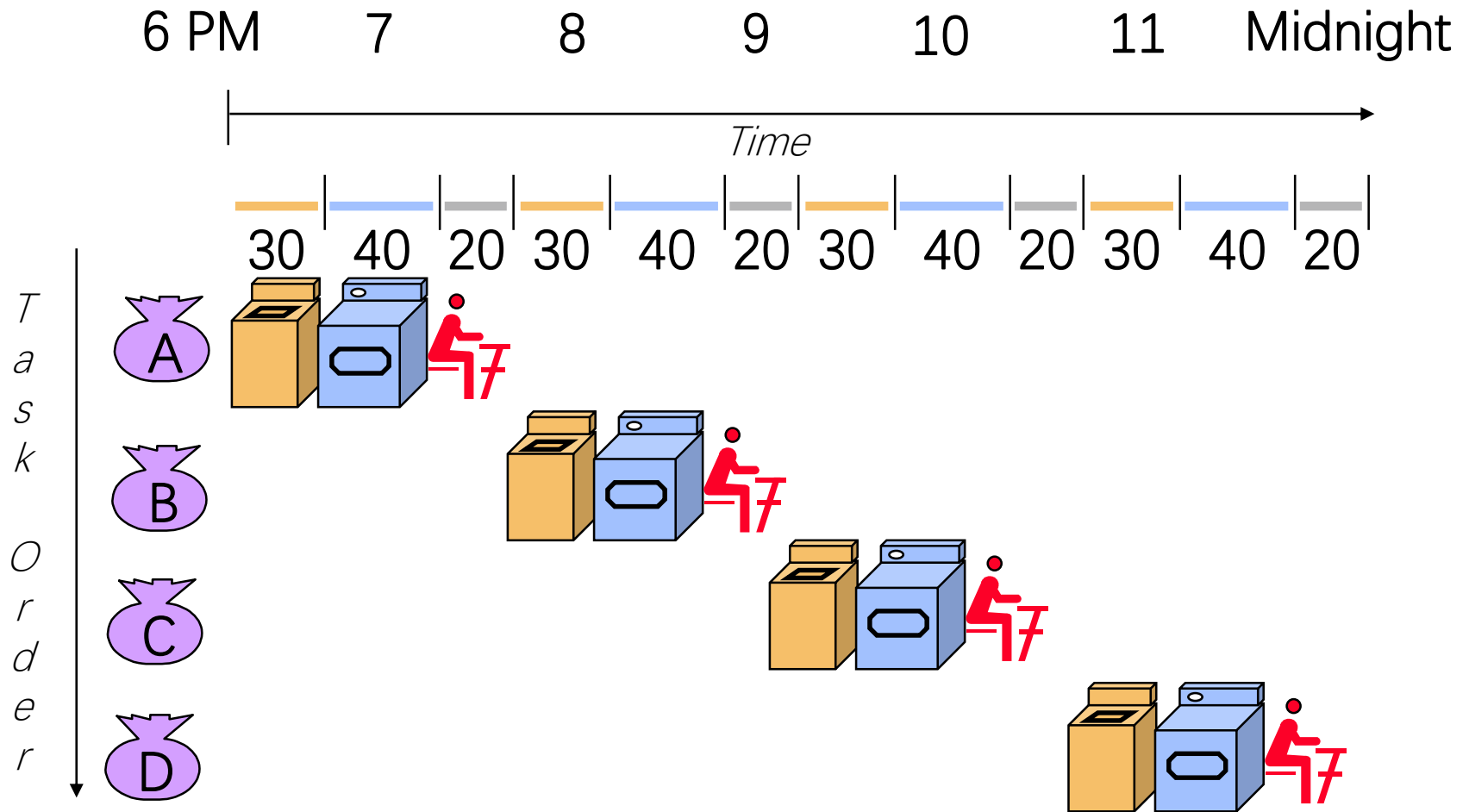
# 解决方案：流水线



## 举例：洗衣房

- Washer takes 30 minutes
- Dryer takes 40 minutes
- Folder takes 20 minutes

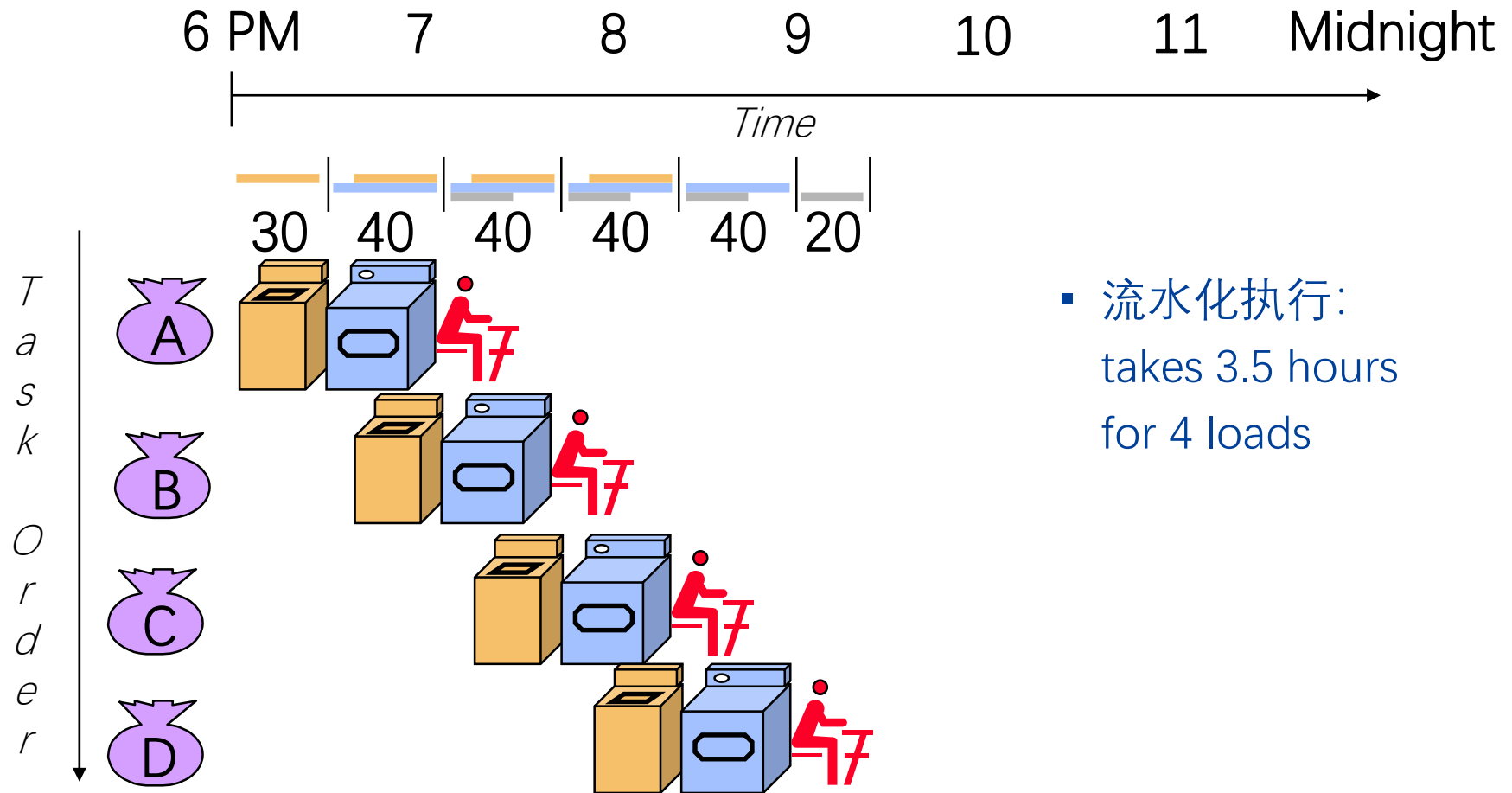




- 顺序执行, 4个任务共花费6小时



# 流水化的洗衣房: 尽快开始工作



# 本节内容



- 流水线的原理和特点
- 指令执行的五个阶段
  - 以LOAD指令为例
- MIPS五阶段流水线的原理

# 回顾：Load指令的五个阶段     lw rt, rs, imm16

Step1	Step2	Step 3	Step 4	Step5
Ifetch	Reg/Dec	Exec	Mem	Wr

- **Ifetch**: 取指令，PC+4

指令存储器、加法器

- Reg/Dec : 读寄存器、指令译码

寄存器读、译码器

- Exec: 计算存储器地址

立即数扩展、运算单元 (ALU)

- Mem : 读数据存储器

数据存储器

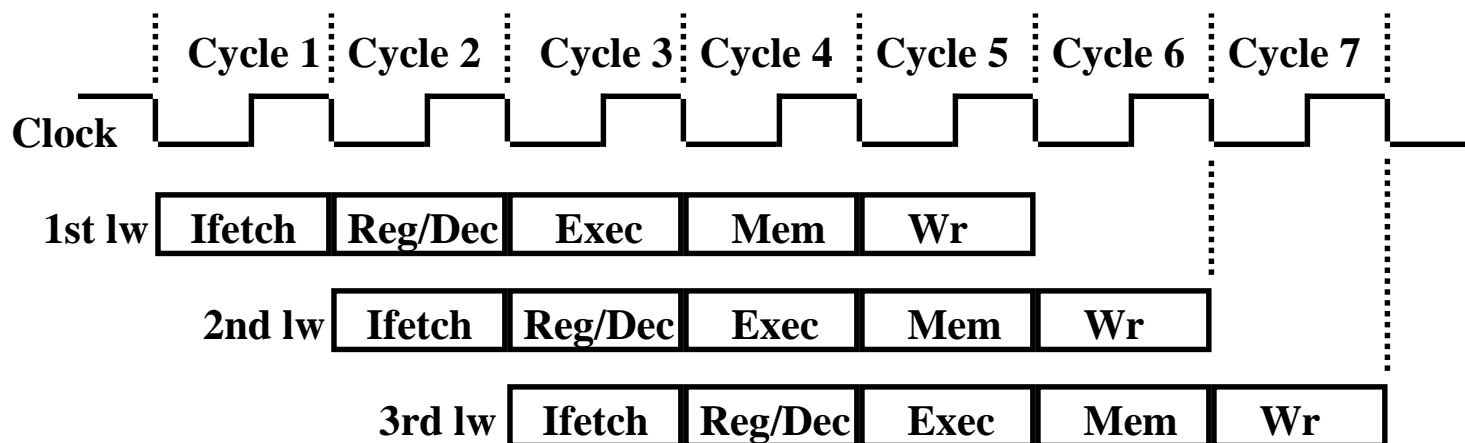
- Wr: 写寄存器

寄存器写

易于流水化:

每一个阶段使用不同的资源（寄存器的读和写可并行执行，后面再讨论）

# 将Load指令的执行流水化



## ■ 在数据通路上有五个功能单元:

- 指令存储器: Instruction Memory : Ifetch (取指令) 阶段
- 寄存器文件的读端口 (bus A and busB) : Reg/Dec (读寄存器/译码) 阶段
- 算术逻辑运算单元 ALU: Exec (执行) 阶段
- 数据存储器 Data Memory : Mem (访存) 阶段
- 寄存器文件的写端口(bus W) : Wr (写结果) 阶段



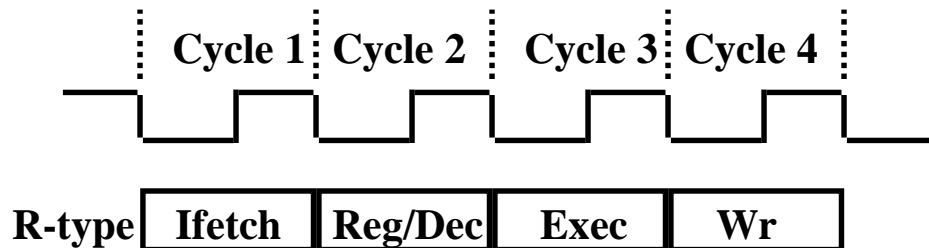


# MIPS 流水化

- 容易流水化

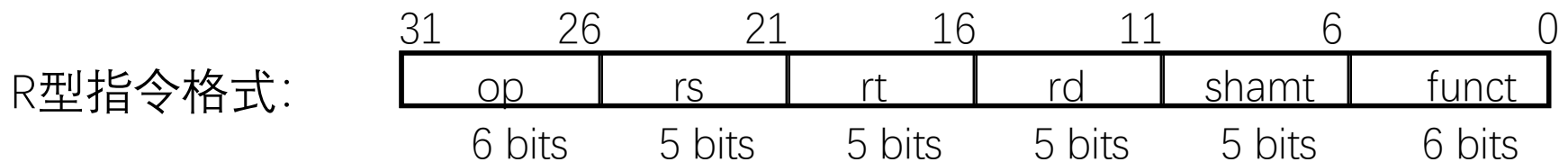
- 1. 所有指令等长 (32 bits)
  - 可以在第一阶段内完成取指， 第二阶段完成译码
- 2. 指令格式少， （三种）
  - 都可以在第二段读寄存器
- 3. 只有loads and stores指令能访问存储器
  - 都可以在第三阶段计算存储地址
- 4. 每条指令最多写一个结果
  - 在最后阶段完成， 例如在MEM阶段 或 WB阶段写结果

# R型指令：只需要4阶段



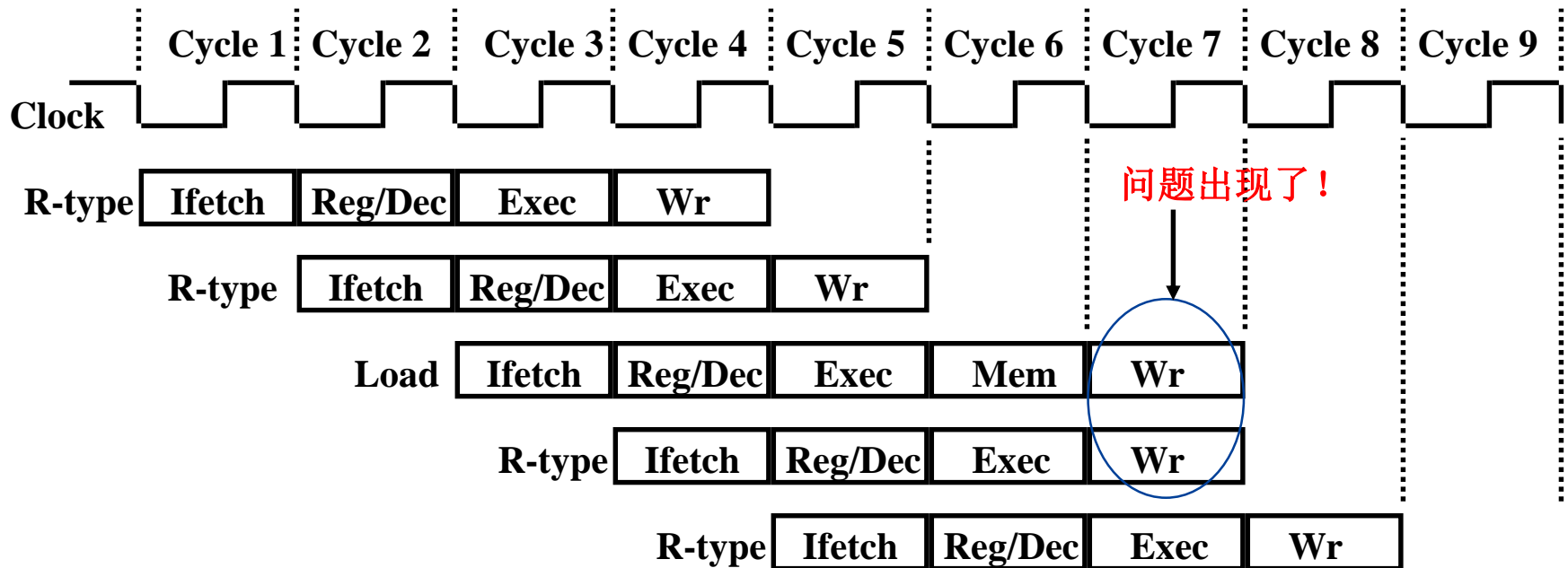
ADD and subtract  
add rd, rs, rt  
sub rd, rs, rt

- Ifetch: 从指令存储器中取指令
- Reg/Dec: 取寄存器内容 、指令译码
- Exec: 使用ALU对从两个寄存器中取出的操作数进行运算
- Wr: 将 ALU 结果写入寄存器文件





# R型指令和 Load指令流水执行



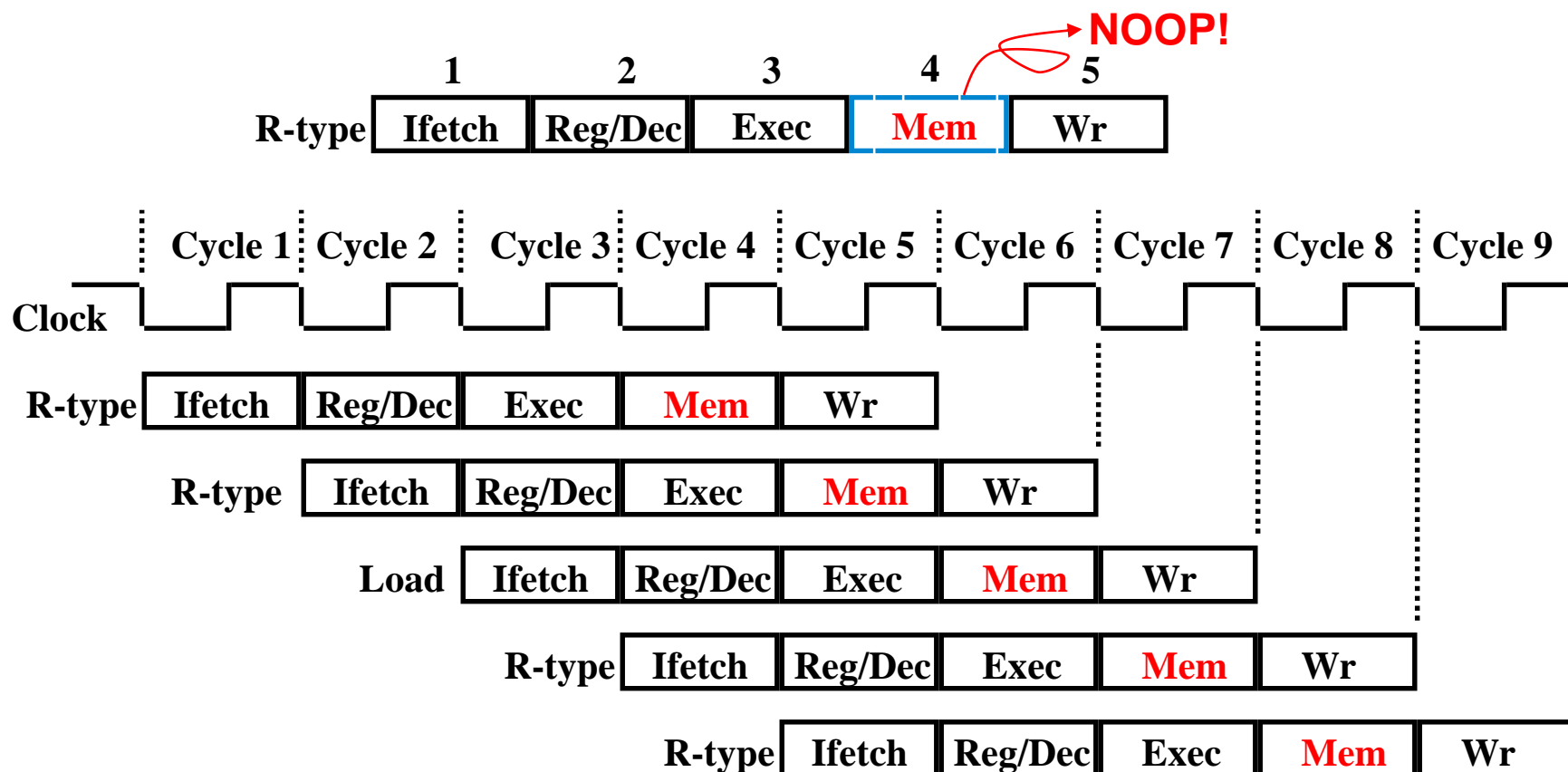
## 出现了资源冲突:

两条指令要在同一个时间段使用寄存器写端口  
只有一个寄存器写端口

# 解决方案：每条指令都有相同的流水段数

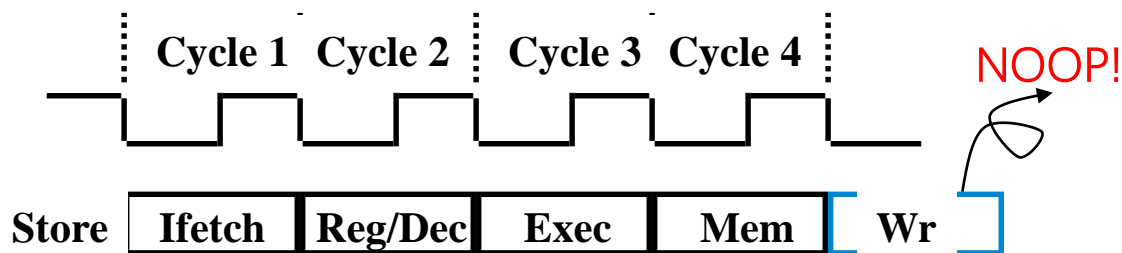
- R型指令的 Write 阶段推后一周期:

- 写寄存器推迟到第五段
- MEM 段是一个空 (NOOP) 操作.





# Store指令



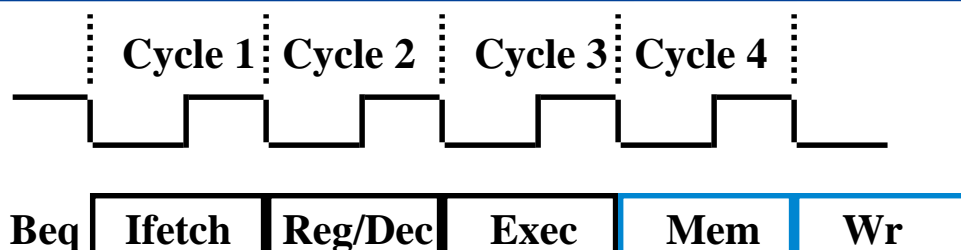
STORE指令:  
sw rt, rs, imm16

- Ifetch: 取指令
  - Fetch the instruction from the Instruction Memory
- Reg/Dec: 取寄存器内容、指令译码
- Exec: 计算存储器地址
- Mem: 将数据写入数据存储器

加一个空阶段，让所有指令的执行段数保持一致



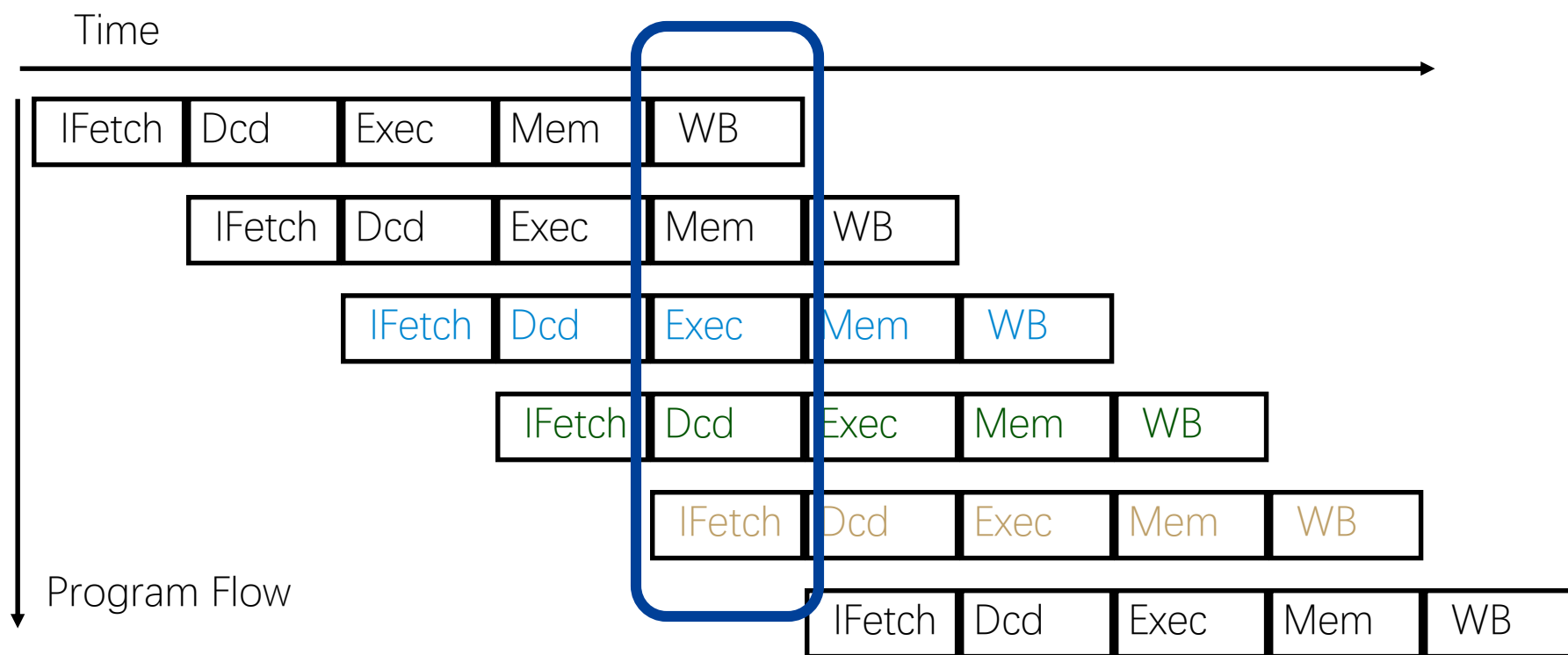
# Beq指令



BRANCH  
beq rs, rt, imm16

- Ifetch:取指令
- Reg/Dec:取寄存器内容 、指令译码
- Exec:
  - 比较两个操作数,
  - 计算转移目标地址
- Mem:
  - 将 计算好的目标地址送给 PC 输入端
- Wr: 空操作

# 理想的指令流水线



理想情况下，每一个周期

完成一条老的指令；

开始一条新的指令。达到CPI=1；

CPI: 完成一条指令所花的周期数

# 小结



- 流水线能提高单位时间的任务吞吐量
- MIPS 指令集很合适流水化执行
- 五阶段指令流水线, 理想情况CPI=1



谢谢！

