

计算机设计与实践

流水线CPU设计-1



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验概况

实验指导书网址: <https://hitsz-cslab.gitee.io/organ/>

实验内容/作业		分值
数据通路表、控制信号取值表		4
汇编实验		10
单周期CPU 设计与实现	课堂检查	6
	调试	20
	下板	10
流水线CPU 设计与实现	理想流水	10
	停顿法	10
	前推法	10
	下板	10
	外设	10

选做: 分支预测 (+10分)

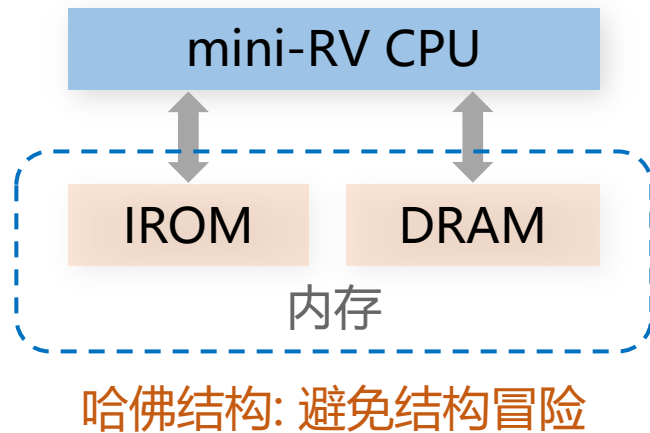
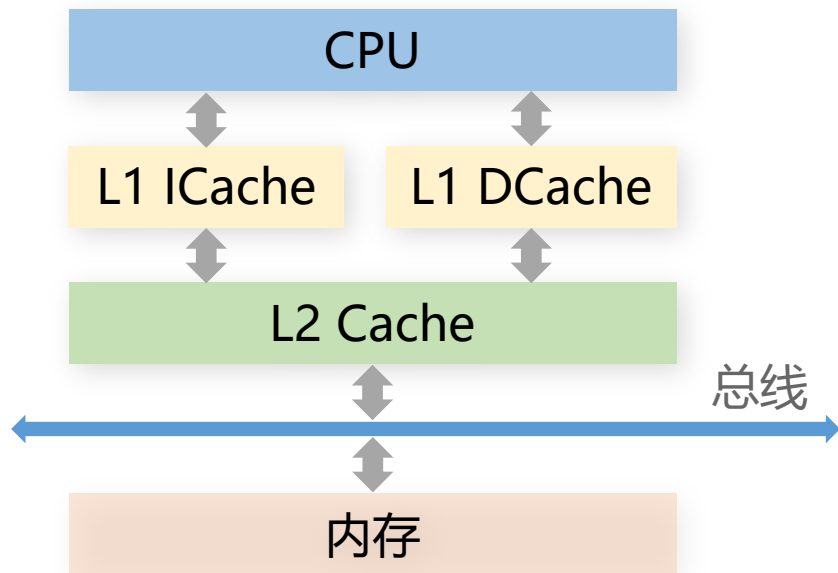
中断异常处理 (+5分)

添加指令 (+5分)



回顾

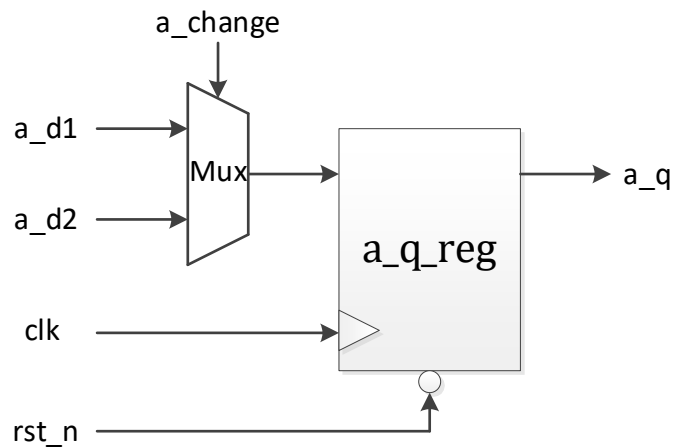
◆ CPU、Cache与内存的关系



回顾

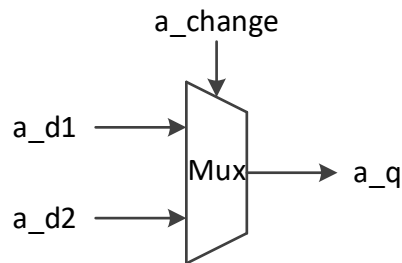
◆ 时序逻辑

```
reg [3:0] a_q;  
  
always @ (posedge clk or negedge rst_n) begin  
    if (~rst_n)      a_q <= 4'h0;  
    else if (a_change) a_q <= a_d1;  
    else              a_q <= a_d2;  
end
```



◆ 组合逻辑

```
reg [3:0] a_q;  
  
always @ (*) begin  
    if (a_change) a_q = a_d1;  
    else          a_q = a_d2;  
end
```



实验目的

- ◆ 加深对流水线CPU**结构**和**工作原理**的理解
- ◆ 熟练掌握数字电路的仿真调试方法



实验内容

- ◆ 根据CPU的功能，划分流水线
- ◆ 根据流水线的划分添加流水寄存器，实现理想流水线
- ◆ 设计testbench，对理想流水线CPU验证



流水线划分

◆ 五级流水线

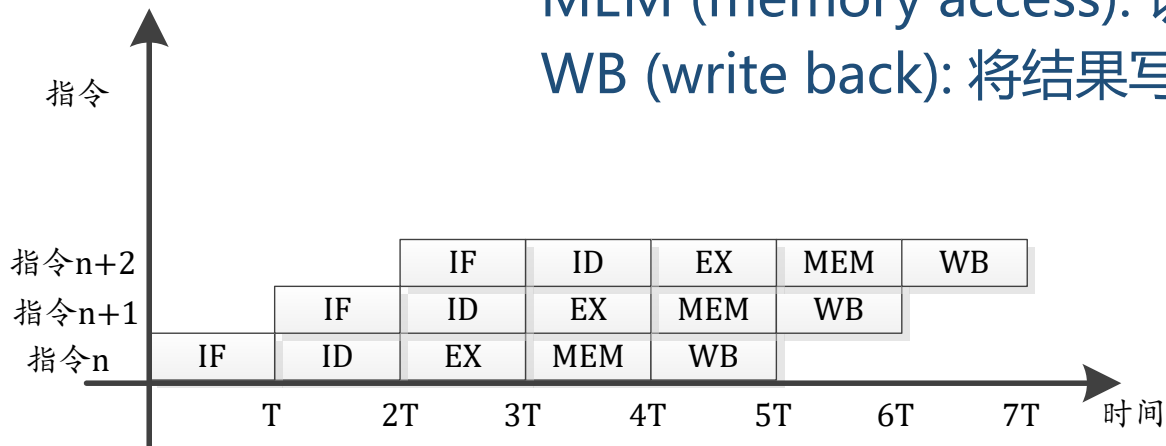
IF (instruction fetch): 从存储器中取出指令

ID (instruction decode): 读寄存器并译码指令

EX (execute): 执行操作或计算地址

MEM (memory access): 访问数据存储器中的操作数

WB (write back): 将结果写入寄存器



流水线划分

◆ 六级流水线

IF1 (instruction fetch1): 从存储器中取出指令

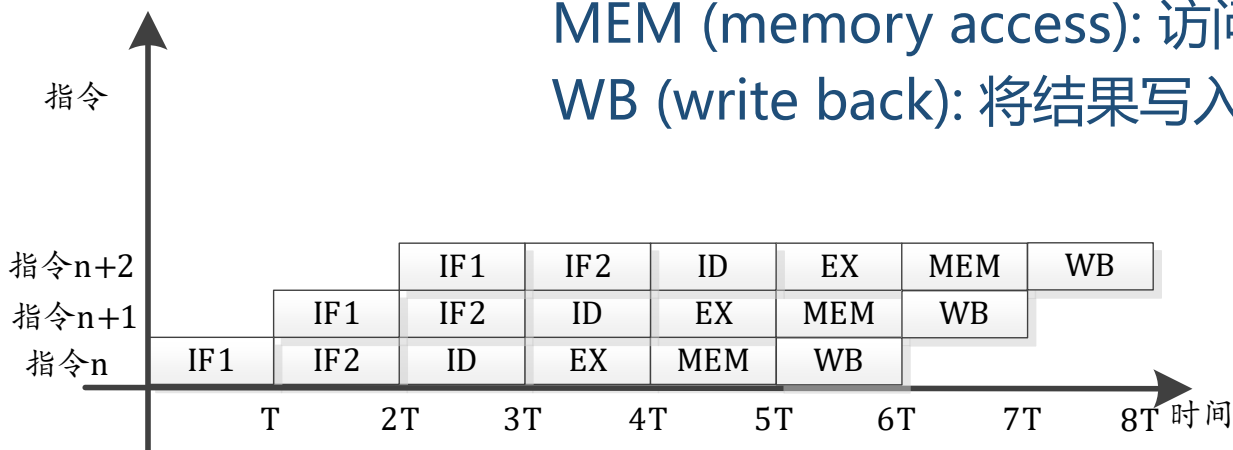
IF2 (instruction fetch2): 进行指令分支预测

ID (instruction decode): 读寄存器并译码指令

EX (execute): 执行操作或计算地址

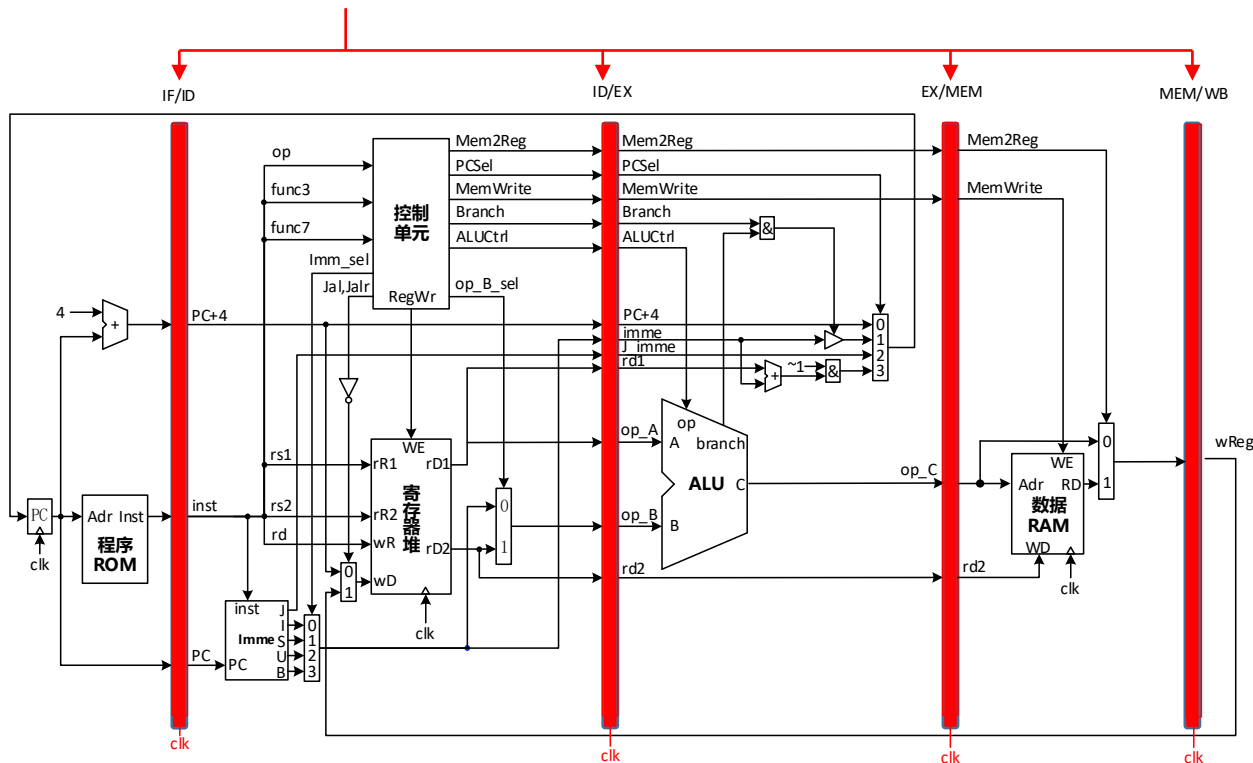
MEM (memory access): 访问数据存储器中的操作数

WB (write back): 将结果写入寄存器



流水寄存器

- 添加流水寄存器



- 1) 保存该流水级输出数据信息，交给下一级处理，并共享给其他指令
- 2) 切割组合逻辑，提升系统频率

流水寄存器

- 确认流水寄存器内容 (IF/ID)

示例

IF/ID



PC: J型指令、B型指令

PC+4: J型指令

inst: 所有指令

RTL实现

```
16 always @ (posedge clk or negedge rst_n) begin
17     if (~rst_n) id_pc <= 32'h0;
18     else        id_pc <= if_pc;
19 end
20
21 always @ (posedge clk or negedge rst_n) begin
22     if (~rst_n) id_inst <= 32'h0;
23     else        id_inst <= if_inst;
24 end
```

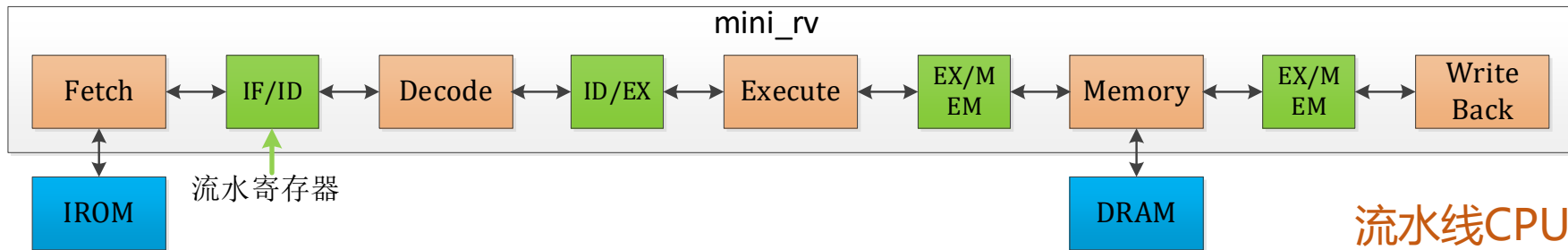
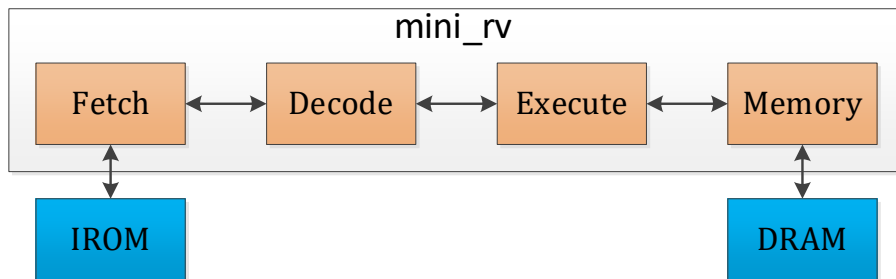


流水寄存器

- 添加流水寄存器的框图

示例

单周期CPU



流水线CPU



理想流水线仿真

- ◆ 编写无冲突指令序列
- ◆ 加载到IROM
- ◆ 仿真





Let's roll !



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

