

计算机设计与实践

流水线CPU设计-2



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

实验目的

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



实验内容

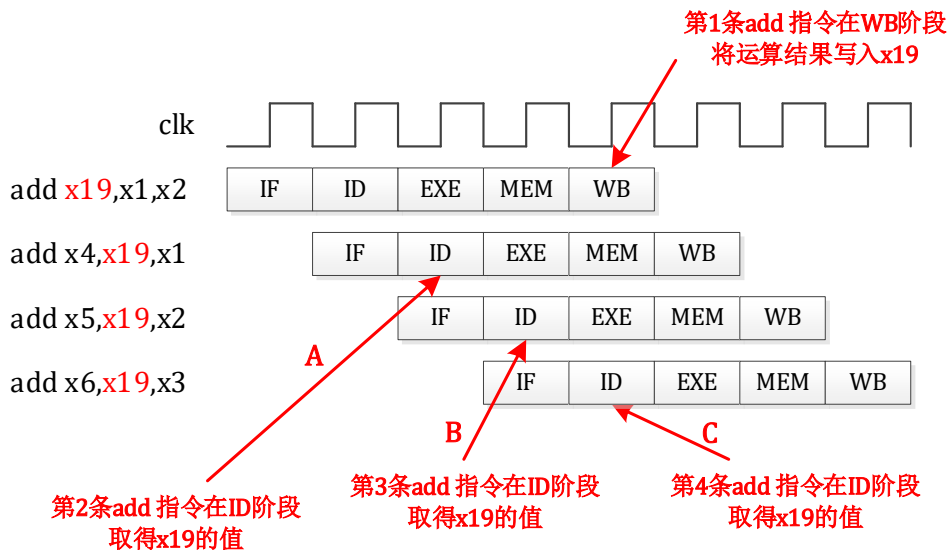
- ◆ 检测流水线CPU的数据冒险和控制冒险
- ◆ 实现流水线暂停
- ◆ 设计testbench, 对增加停顿的流水线CPU验证



流水线冒险检测

◆ 数据冒险

一条指令依赖于前面一条尚在流水线中的指令；因**无法提供指令执行所需数据**而导致指令不能在预期的时钟周期内执行的情况



流水线冒险检测

◆ 数据冒险检测

情形A:

ID/EX .RS1 = EX/MEM.RD = x19

ID/EX .RS2 = EX/MEM.RD = x19

情形B:

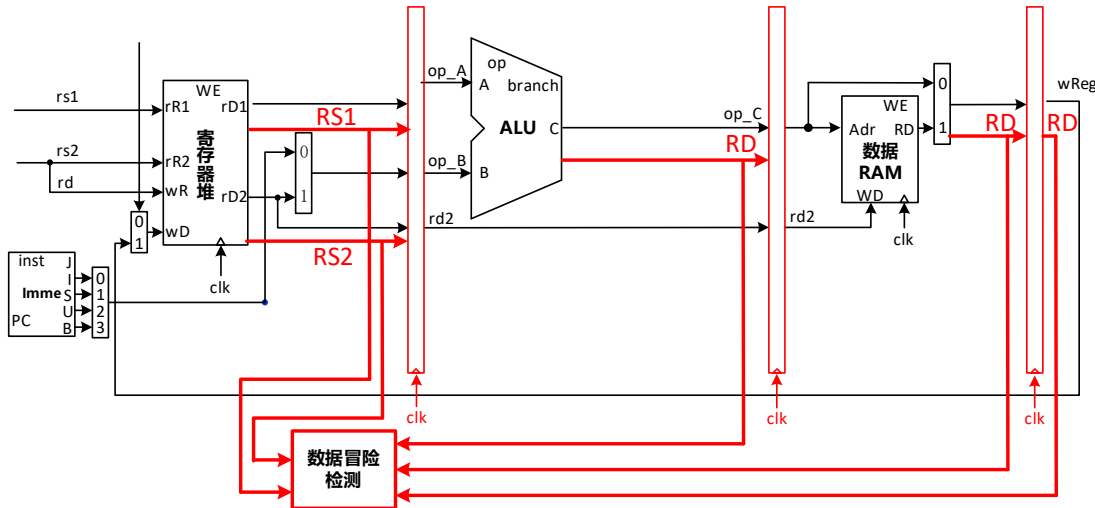
ID/EX .RS1 = MEM/WB.RD = x19

ID/EX .RS2 = MEM/WB.RD = x19

情形C:

ID/EX .RS1 = WB.RD = x19

ID/EX .RS2 = WB.RD = x19



RTL实现

当前ID阶段，寄存器是否有被读取



```
43 wire rs1_id_wb_con = (raddr1 == waddr) & we & re1;  
44 wire rs2_id_wb_con = (raddr2 == waddr) & we & re2;  
45
```



流水线暂停

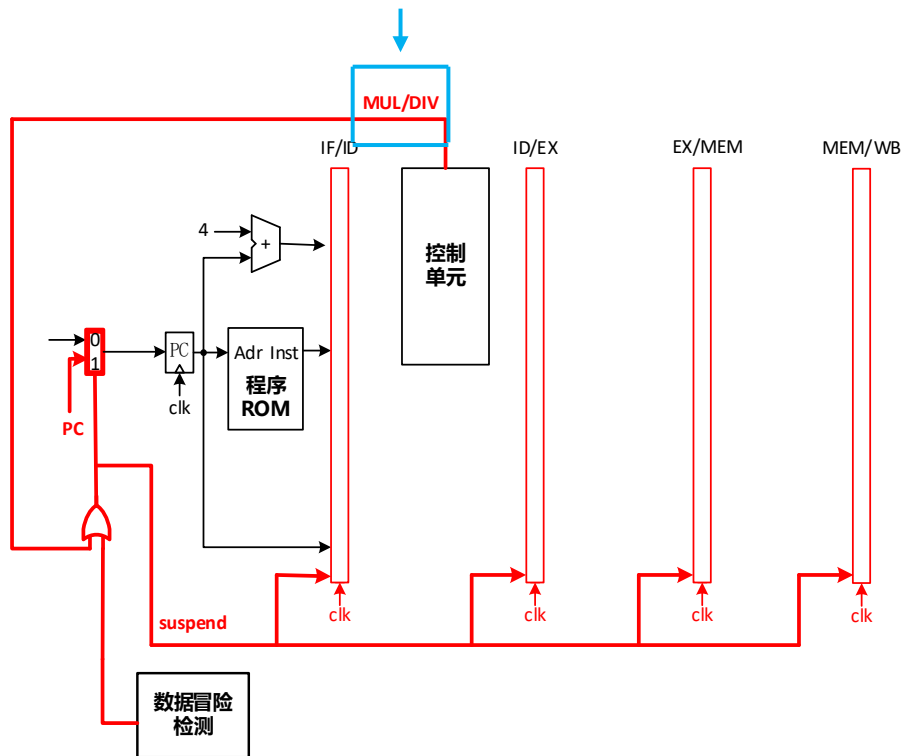
多周期乘除法也可以用暂停法解决

◆ 流水线暂停

实现方法

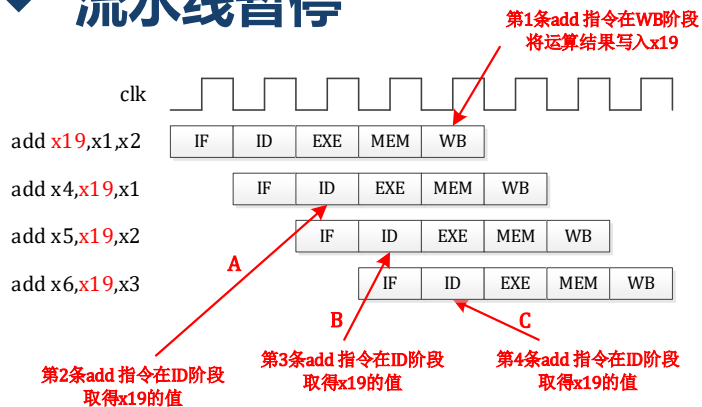
- 1) 保持取值地址PC的值不变;
- 2) 保持流水线各个阶段的寄存器 (IF/ID、ID/EX、EX/MEM、MEM/WB模块的输出) 不变;

提示：控制冒险也可用暂停方法来解决

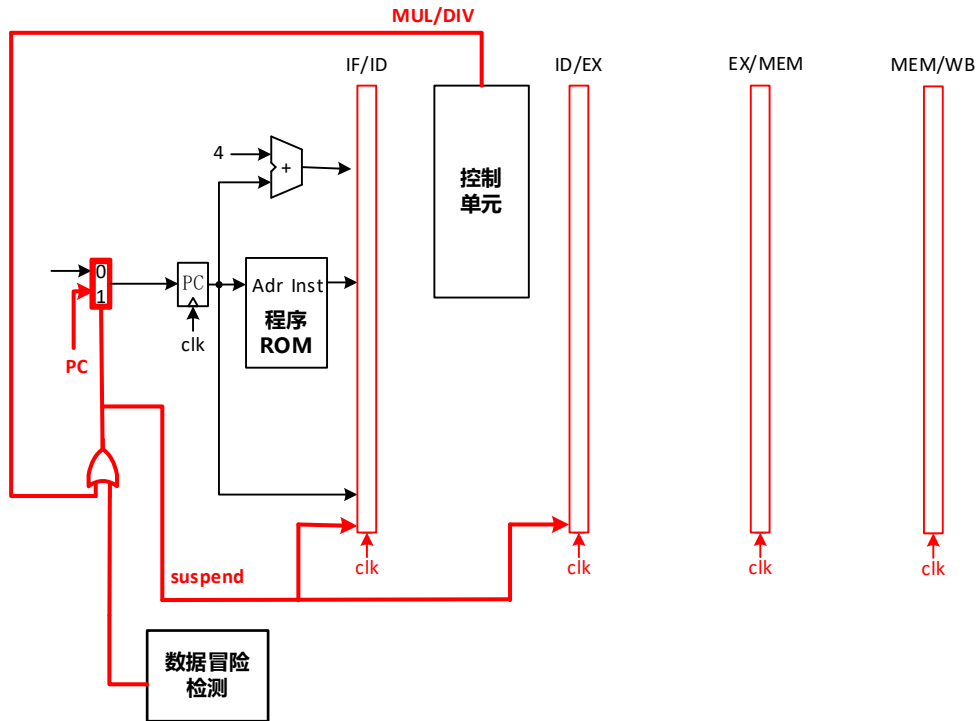


流水线暂停

◆ 流水线暂停



情形	A	B	C
暂停周期数	3	2	1

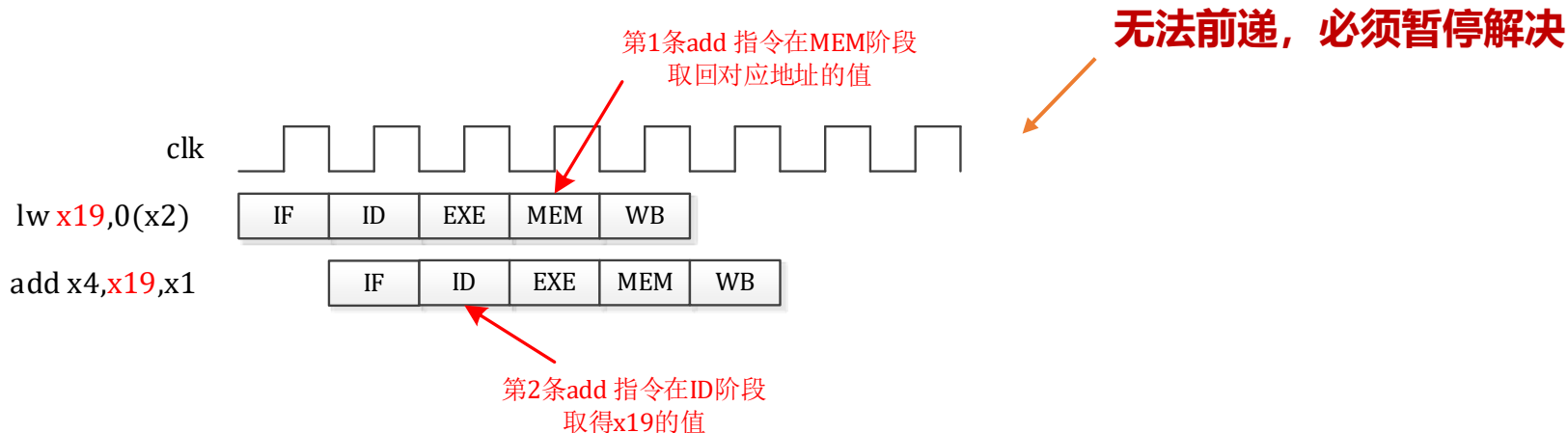


保持PC， IF/ID、ID/EX模块的输出不变

流水线暂停

◆ 流水线暂停

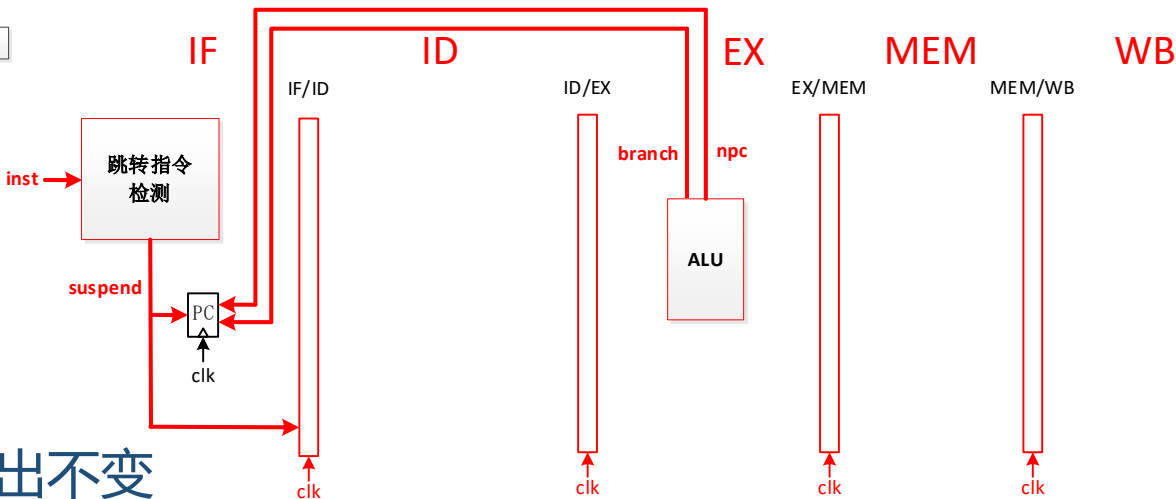
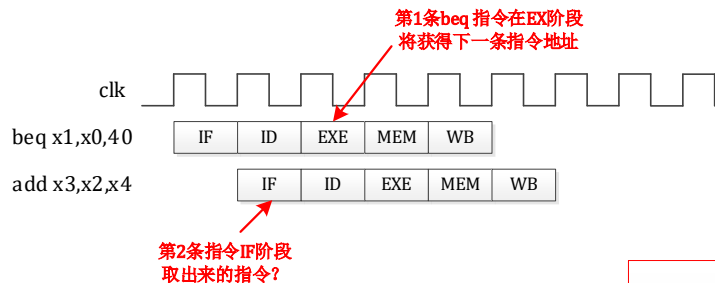
特殊情况 - 访存(读)后读



流水线暂停

解决控制冒险

注意：若不在IF阶段判断跳转指令，
需清空已错误取出的指令



实现方法

暂停2个周期

保持PC， IF/ID模块的输出不变

流水线暂停

- 流水线暂停

要点：暂停信号pipeline_stop的来源以及拉高的周期数

RTL实现示例 (IF/ID)

```
always @ (posedge clk or negedge rst_n) begin
    if (~rst_n)
        id_pc <= 32'h0;
    else if (pipeline_stop) id_pc <= id_pc;
    else
        id_pc <= if_pc;
end

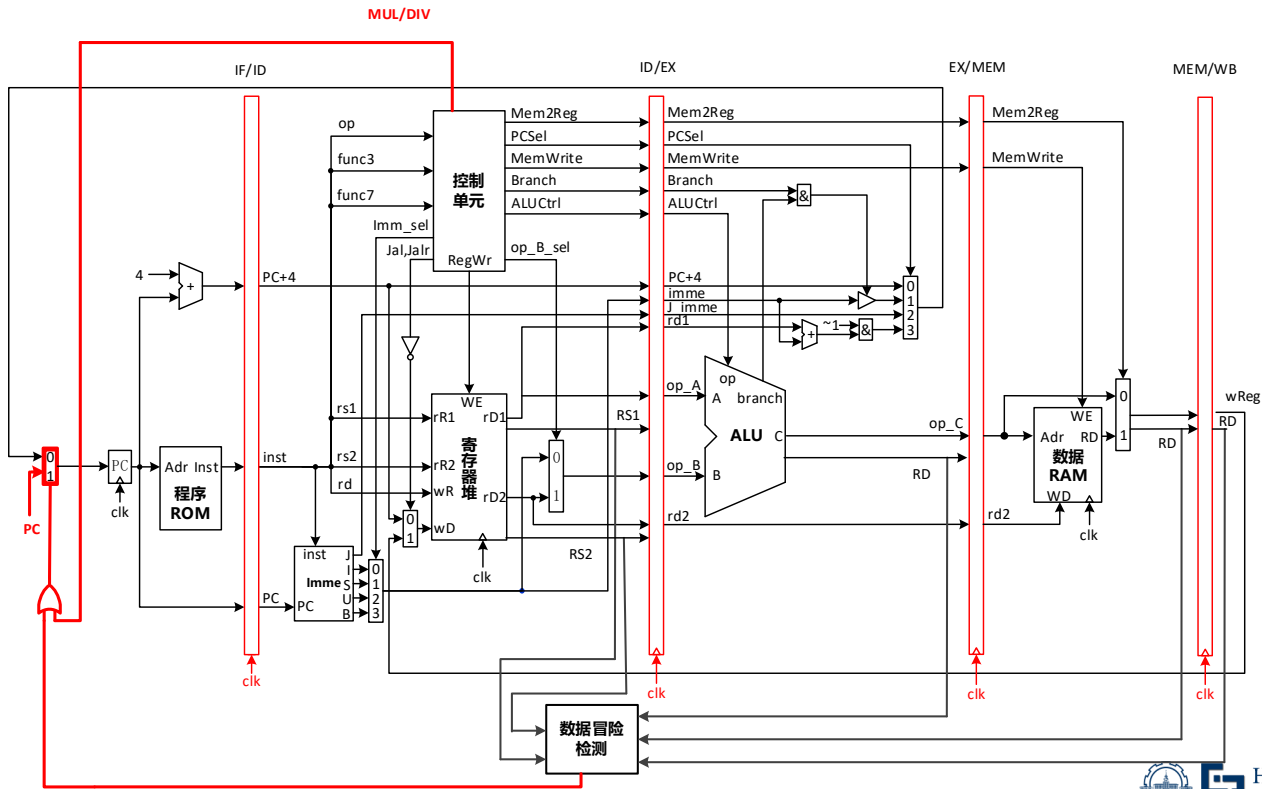
always @ (posedge clk or negedge rst_n) begin
    if (~rst_n)
        id_inst <= 32'h0;
    else if (pipeline_stop) id_inst <= id_inst;
    else
        id_inst <= if_inst;
end
```



流水线数据通路

• 增加暂停机制的数据通路

示例



流水线暂停仿真

- ◆ 编写含冲突指令序列
- ◆ 加载到IROM
- ◆ 仿真

注：亦可用trace仿真





Let's roll !



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