

# 流水线实验总结

---

## 基本流水线：

---

1. 加入 J 类指令、bne、l 类指令。将 ALUcontrol 与控制单元合并。
2. J 类指令应设计为 3 个周期，4 个周期无法执行。
3. beq 指令后应插入 3 个 nop，jump 后插入 2 个 nop。（将跳转结果写回 PC 时，应保证 IF/ID, ID/EX, EX/MEM 内均为空指令）

## 处理 Hazard：

---

### 0. 结构相关

将存储器分为数据存储器 and 指令存储器（哈佛结构）。

### 1. 非load-use数据相关（转发 forwarding）

加入模块：Forwarding\_unit

1. ALU\_out 到 ALU 的转发（EX/MEM 转发至 ID/EX）
2. MEMtoReg 到 ALU 的转发（MEM/WB 转发至 ID/EX）
3. 将寄存器段改为前半周期写（上升沿触发），后半周期读（下降沿触发），使寄存器堆能在一个时钟周期内同时读写。注意应将该段除寄存器堆外的所有寄存器都改为下降沿触发（包括控制单元中的寄存器）（巨坑）

### 2. load-use数据相关（阻塞 interlock）

加入模块：Hazard\_detection\_unit

1. 软件实现方案：加 nop 解决。（已舍弃）
2. 硬件实现方案：将控制单元、IR、PC 暂停一个周期。（因为要使用 ID/EX 段的控制信号 MemRead，需保持同步，故需要将 IR 与 PC 同样改为下降沿触发）（√）

### 3. 控制相关（静态分支预测）

加入模块：Branch\_flush\_unit

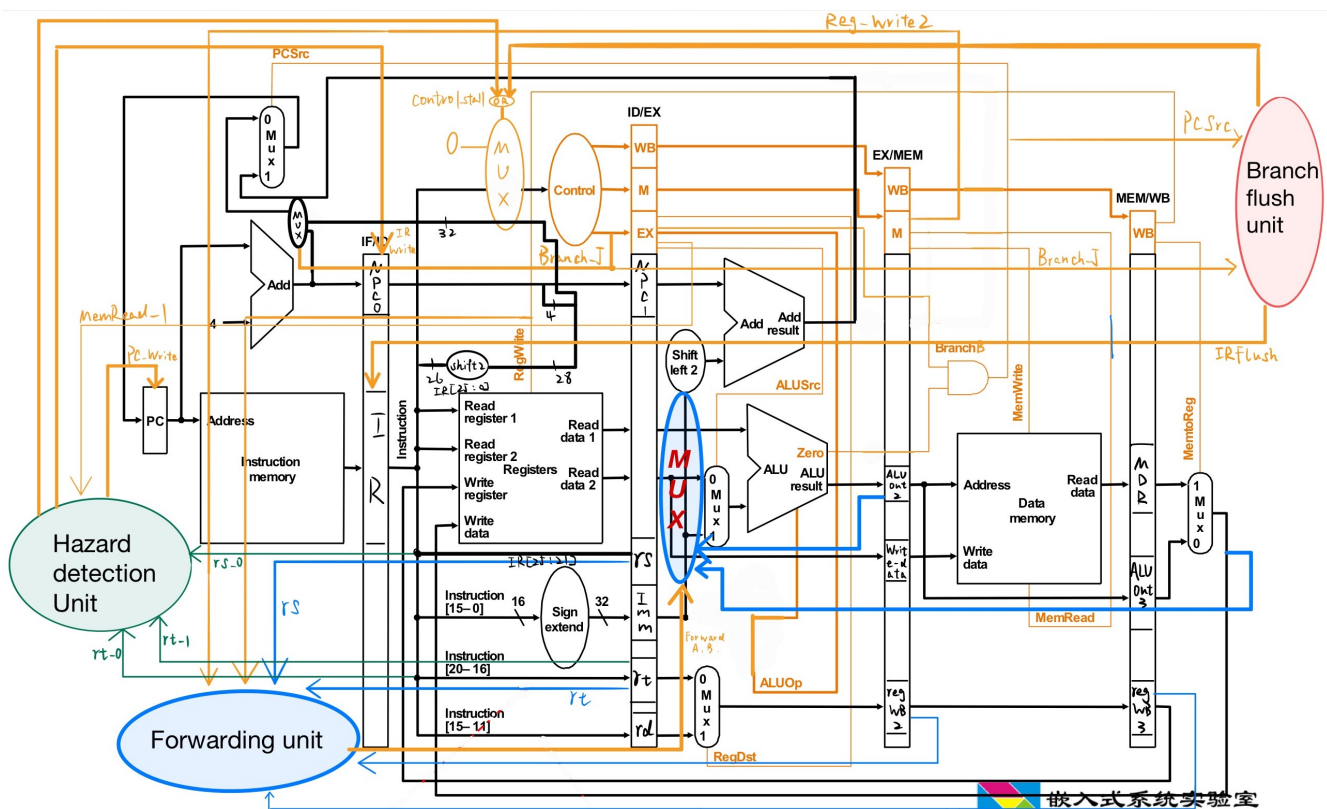
1. 修改数据通路，将 jump, beq, bne 均改为 2 个周期就执行完。× 按照 PPT 方案此处有大坑！修改数据通路应考虑转发。要利用之前完成的转发数据通路，beq 最快也要 3 个周期，故设计 3 个周期的方案。

此时 ID/EX 段的控制信号 PCSrc 依赖于转发，为保持同步，需要将 EX/MEM, MEM/WB 同样改为下降沿触发。

至此，整个系统除了写寄存器堆是上升沿敏感以外，其余时序均修改为了下降沿敏感！于是得出一开始的修改方案有问题，其实只需要将写寄存器堆置为下降沿敏感，其余不需要改动。兜兜转转一圈终于弄明白。

2. 分支预测：假设分支跳转都不成功。若分支跳转成功，则flush掉 IF/ID 的寄存器，beq 和 bne 指令还需要 flush 掉 ID/EX 段的控制信号。jump 指令需要延迟一个周期，beq 若跳转成功需要延迟两个周期。

## 最终电路图：



至此，完成了流水线 CPU 的完全转发、阻塞和静态分支预测。可以不加气泡的执行任意一个不超出指令范围、不产生异常的 MIPS 汇编程序，性能优秀。后续若需要继续提升性能，可加入动态分支预测。若需要处理异常，需加入中断。

