

学号：202200130048	姓名：陈静雯	班级：6
实验题目：实验五：数据相关		
实验学时：2	实验日期：2025. 5. 23	
实验目的： 通过本实验，加深对数据相关的理解，掌握如何使用定向技术来减少数据相关带来的暂停。		
硬件环境： Windows		
软件环境： Otvdm		
实验程序：data_d.s		
代码解读：		
1. 初始化阶段		
• lhi r2, 0x0 + addui r2, r2, 0x134 将寄存器 R2 初始化为地址 0x00000134，这是数组 A 的起始地址。		
• lhi r3, 0x0 + addui r3, r3, 0x15c 将寄存器 R3 初始化为地址 0x0000015c，这是数组 B 的起始地址。		
2. 循环阶段（loop 标签）		
• lw r1, 0x0[r2] 从 R2 指向的地址（即数组 A 的当前元素）加载值到 R1。		
• add r1, r1, r3 将 R1（原数组 A 的值）与 R3（数组 B 的基地址）相加，结果存回 R1。 功能：将数组 A 的元素修改为指向数组 B 的地址（A[i] = &B[A[i]]）。		
• sw 0x0[r2], r1 将新的地址值（&B[A[i]]）写回数组 A 的当前位置。		
• lw r5, 0x0[r1] 从 R1 指向的地址（即数组 B 的某个元素）加载值到 R5。		
• addi r5, r5, 0xa 将 R5 的值加 10。		
• addi r2, r2, 0x4 移动 R2 到数组 A 的下一个元素（地址递增 4 字节）。		
• sub r4, r3, r2 计算 R3（数组 B 的基地址）与 R2（当前 A 的地址）的差值，结果存入 R4。 功能：当 R2 移动到数组 B 的起始地址时，R4 变为 0，循环终止。		
• bnez r4, loop 若 R4 不为 0，继续循环。		
3. 终止阶段		
• trap 0x0		

程序终止指令，结束执行。

4. 数据段（地址 0x0000015c 开始）

- A 和 B 的存储区域：

从 0x0000015c 开始的内存被标记为非法指令，实际是数据区域，用于存储数组 A 和 B 的值。

- A 的初始值为 0, 4, 8, ..., 36（每个元素占 4 字节）。
- B 的初始值为 9, 8, 7, ..., 0（每个元素占 4 字节）。

关键逻辑

1. 数组 A 的转换：

原数组 A 的元素是数值（如 0, 4, 8），通过循环将其转换为指向数组 B 的地址（ $A[i] = \&B[i]$ ）。

2. 循环终止条件：

当 R2（A 的指针）移动到 R3（B 的基地址）时，R4 为 0，循环结束。

实验内容：

(1) 在不采用定向技术的情况下（通过 Configuration 菜单中的 Enable Forwarding 选项设置），用 WinDLX 模拟器运行程序 data_d.s。

(2) 记录数据相关引起的暂停时钟周期数以及程序执行的总时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。

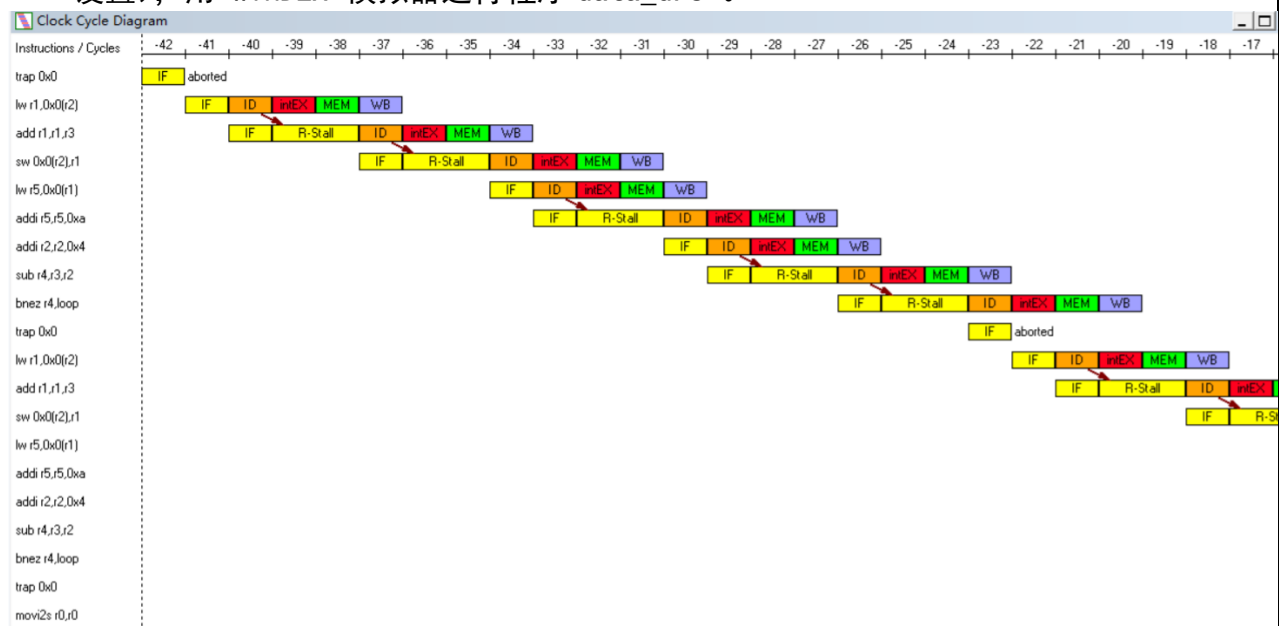
(3) 在采用定向技术的情况下，用 WinDLX 模拟器再次运行程序 data_d.s。

(4) 记录数据相关引起的暂停时钟周期数以及程序执行的总时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。

(5) 根据上面记录的数据，计算采用定向技术后性能提高的倍数。

实验步骤和结果：

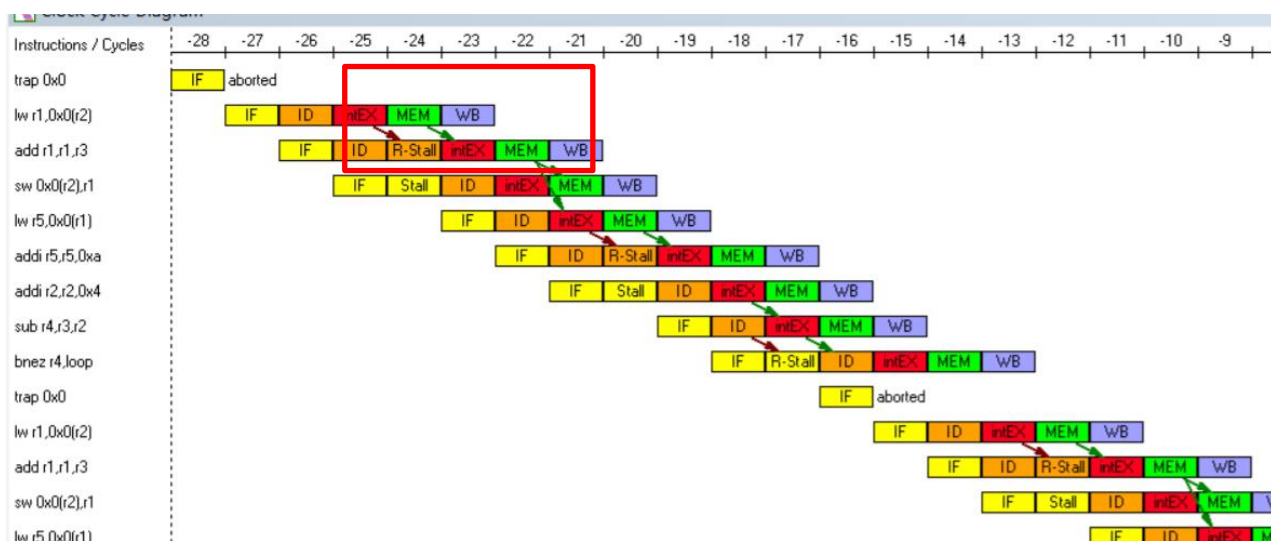
(1) 在不采用定向技术的情况下（通过 Configuration 菜单中的 Enable Forwarding 选项设置），用 WinDLX 模拟器运行程序 data_d.s。



(2) 记录数据相关引起的暂停时钟周期数以及程序执行的总时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。

暂停周期数 10，总周期数 42，百分比 23.8%

(3) 在采用定向技术的情况下，用 WinDLX 模拟器再次运行程序 data_d.s。



(4) 记录数据相关引起的暂停时钟周期数以及程序执行的总时钟周期数，计算暂停时钟周期数占总执行周期数的百分比。

add 指令需要 lw 指令从内存加载到 r1 的值，但 lw 的结果在 MEM 阶段才能获取，而 add 的 EX 阶段需要该值。

定向实现：将 lw 在 MEM 阶段读取的内存值直接转发给 add 的 EX 阶段，无需等待 lw 完成 WB 阶段。

所以没有数据相关引起的暂停，暂停周期数为 0，总周期数是 28，百分比 0。

(5) 根据上面记录的数据，计算采用定向技术后性能提高的倍数。

未启用总周期数/启用总周期数=42/28=1.5

结论分析与体会：

1. 定向技术的作用：

- 减少因数据冒险导致的流水线停顿，总周期数减少。
- 暂停周期数占比下降，性能提升 1.5 倍。

2. 实际应用意义：

- 定向技术通过直接转发中间结果，避免了等待写回阶段的延迟，显著提升了流水线效率。
- 对于频繁存在数据相关的代码（如循环结构），性能提升效果更明显。