

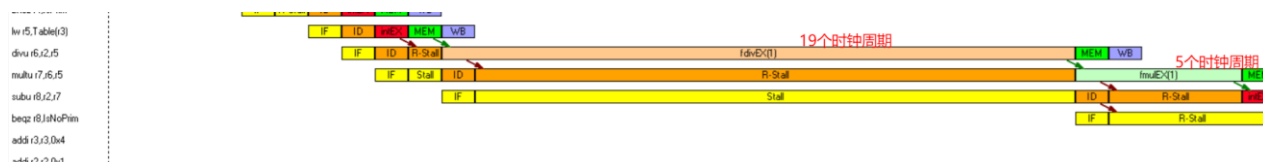
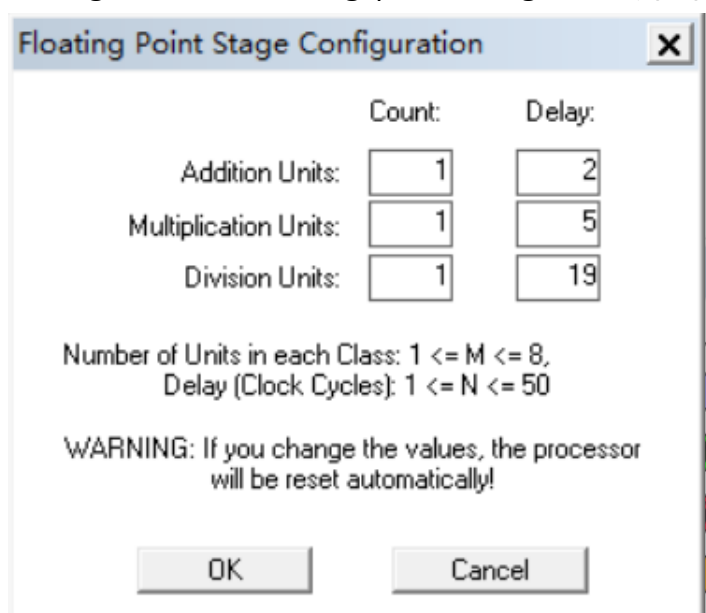
# 山东大学 计算机科学与技术 学院

## 计算机体系结构 课程实验报告

学号：202200130048	姓名：陈静雯	班级：6
实验题目：实验三 用 WinDLX 模拟器完成求素数程序		
实验学时：2	实验日期：2025. 5. 9	
<b>实验目的：</b> 通过本实验，熟悉 WinDLX 模拟器的操作和使用，了解 DLX 指令集结构及其特点。		
<b>硬件环境：</b> Windows		
<b>软件环境：</b> Otvdm		
<b>实验程序理解：</b> 生成指定数量（Count）的素数，并存储到 Table 中。 <pre> .data ;定义全局变量 Count，表示要生成的素数个数 .global Count Count: .word 10          ; 初始值为 10  ;定义全局数组 Table，存储素数 .global Table Table: .space Count*4    ; 分配 Count*4 字节（10 个 32 位整数）空间  .text .global main main: ;初始化部分 addi r1, r0, 0           ; r1 = 0，作为 Table 的索引（字节偏移） addi r2, r0, 2           ; r2 = 2，从 2 开始检查素数  NextValue: ;检查 r2 是否为素数 addi r3, r0, 0           ; r3 = 0，内循环索引，遍历已存储的素数  Loop: ;判断是否遍历完所有已存素数 seq  r4, r1, r3          ; 比较 r1 和 r3，若相等（索引到头）则 r4=1 bnez r4, IsPrim          ; 若 r4≠0（已遍历完），跳转到 IsPrim  ;检查当前数是否能被已存素数整除 </pre>		



2. 在执行程序过程中，注意体验单步执行除法和乘法指令的节拍数，并和主菜单 configuration/floating point stages 中的各指令执行拍数进行比较。



实际单步执行时观察到的乘法和除法指令的节拍数与配置中设置的数值是一致的

## 结论分析与体会：

### 1. 指令节拍数的验证：

- 通过单步执行观察，浮点乘法（FMUL）和除法（FDIV）指令的实际执行节拍数与 Configuration/Floating Point Stages 中的配置完全一致。
- 这表明 WinDLX 模拟器严格按照流水线配置执行指令，验证了流水线阶段延迟参数的准确性。

### 2. 流水线操作的关键性：

- 实验中观察到，长延迟指令（如 FDIV）会显著增加流水线的停顿时间，导致后续指令的发射延迟（如 RAW 数据冲突）。这体现了流水线设计中指令调度和冲突解决的重要性。

### 3. 对 WinDLX 流水线的深入理解：

- 通过单步跟踪指令执行过程，直观体会到流水线各阶段（IF、ID、EX、MEM、WB）的分工与协作。例如，乘/除法指令在 EX 阶段的长时间占用，会导致后续指令的 IF 和 ID 阶段被阻塞，充分验证了流水线性能瓶颈的来源。