

## Lab5.1 流水线 CPU 在线虚拟仿真实验 课堂练习

### 实验报告

姓名：潘文峥

学号：520030910232

#### 1、实验目的

- 1、理解流水线 CPU 指令执行过程。
- 2、理解流水线冒险处理的概念。
- 3、理解不同流水线硬件结构对冒险处理方法的区别。

#### 2、课堂练习

##### 2.1 练习一

输入代码段（参考课本习题 4.22）：

```
sd x29,1024(x16)
ld x29,1024(x16)
sub x17,x15,x14
beq x17,x0,label
add x15,x11,x14
sub x15,x30,x14
label:nop
```

- 1) 单步执行仿真，给出 execution table 截图。从这个图可以看出倒数第 3 和 2 句（beq x17,x0,label 和 add x15,x11,x14）被执行了什么特殊处理？为什么？

作答：

EXECUTION TABLE											
FULL LOOPS ▾	CPU Cycles										
Instruction	1	2	3	4	5	6	7	8	9	10	11
sd t4, 1024(a6)	F	D	X	M	W						
ld t4, 1024(a6)		F	D	X	M	W					
sub a7, a5, a4			F	D	X	M	W				
beq a7, x0, 24				F	-	D	X	M	W		
add a5, a1, a4						F					
addi x0, x0, 0							F	D	X	M	W

图一：截图 1

- (1) beq 指令被执行了 forward 处理，因为上一句 sub 中 x17 被修改，发生数据冒险，执行了前递；
- (2) add 指令被执行了 flush 处理，因为上一句 beq 分支指令预测不正确，发生控制冒险，通过 flush 清除已读取的指令。

2) 在 ripes 仿真环境下做一次五级流水线 CPU 仿真，参照表 1 前三列的前三个时钟执行过程画出完整五级流水线（F、D、X、M、W）执行表。与上面的执行表截图作对比。分析有什么异同。

作答：

表 1

Instruction\clock cycle	1	2	3	4	5	6	7	8	9	10	11
sd x29,1024(x16)	F	D	X	M	W						
ld x29,1024(x16)		F	D	X	M	W					
sub x17,x15,x14			F	D	X	M	W				
beq x17,x0,label				F	D	X	M	W			
add x15,x11,x14					F	D					
sub x15,x30,x14						F					
label:nop							F	D	X	M	W

Ripes 平台对控制冒险的检测比实验平台二靠后一个流水级，需要清除 add 和 sub 两条语句；而实验平台二选择在 beq 的 IF 级后就暂停一个周期，只用清除一条 add 指令。但是，最终执行结果和所花时间是完全相同的。

## 2.2 练习二

在五段流水线体系中执行以下 RISC-V 程序指令段：

```
addi x6,x0,1024
addi x2,x2,18
add x2,x3,x4
sw x2,0(x6)
lw x7,4(x6)
add x1,x1,x7
add x3,x5,x7
sub x3,x3,x7
lw x4,8(x6)
add x4,x4, x4
add x4,x6,x7
```

填写表 2.1 回答：

- 1) 哪几条标号指令存在数据冲突（hazard）？
- 2) 哪几条标号指令数据冲突不能通过旁路技术解决？为什么？
- 3) 如果流水线时钟周期为 100ns，计算完成这些指令所需时间。

作答:

表 2.1

序号	程序段	是否有冲突	是否可通过旁路解决冲突，为什么
1	addi x6,x0,1024	否	
2	addi x2,x2,18	否	
3	add x2,x3,x4	是	可以
4	sw x2,0(x6)	是	可以
5	lw x7,4(x6)	否	
6	add x1,x1,x7	是	需停顿一周，因为要等 mem 读数
7	add x3,x5,x7	否	
8	sub x3,x3,x7	是	可以
9	lw x4,8(x6)	否	
10	add x4,x4, x4	是	需停顿一周，因为要等 mem 读数
11	add x4,x6,x7	是	可以

执行用时：17\*100=1700ns

4) 分析程序段，以表 2.2 为模板画出你认为可能的五段流水线执行表表 2.2-1。并在 **ripes** 平台上完成仿真，以表 2.2 为模板画出实际仿真过程的执行表标为表 2.2-2，再通过 **webriscv-v** 平台进行仿真，截图得到仿真过程的执行表标为表 2.2-3，做对比分析。和你认为的执行过程是否一致？对比两个平台上的仿真执行表的异同，理解同样支持 RISC-V 指令而以不同流水线硬件结构实现时其实际执行过程的区别。

作答:

表 2.2-1

Instruction\clock cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
addi x6,x0,1024	F	D	E	M	W												
addi x2,x2,18		F	D	E	M	W											
add x2,x3,x4			F	D	E	M	W										
sw x2,0(x6)				F	D	E	M	W									
lw x7,4(x6)					F	D	E	M	W								
add x1,x1,x7						F	D	-	E	M	W						
add x3,x5,x7							F	-	D	E	M	W					
sub x3,x3,x7									F	D	E	M	W				
lw x4,8(x6)										F	D	E	M	W			
add x4,x4,x4											F	D	-	E	M	W	
add x4,x6,x7												F	-	D	E	M	W

表 2.2-2

Instruction\clock cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
addi x6,x0,1024	F	D	E	M	W												
addi x2,x2,18		F	D	E	M	W											
add x2,x3,x4			F	D	E	M	W										
sw x2,0(x6)				F	D	E	M	W									
lw x7,4(x6)					F	D	E	M	W								
add x1,x1,x7						F	D	-	E	M	W						
add x3,x5,x7							F	-	D	E	M	W					
sub x3,x3,x7									F	D	E	M	W				
lw x4,8(x6)										F	D	E	M	W			
add x4,x4,x4											F	D	-	E	M	W	
add x4,x6,x7												F	-	D	E	M	W

EXECUTION TABLE																	
FULL LOOPS ▾		CPU Cycles															
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
addi t1, x0, 1024	F	D	X	M	W												
addi sp, sp, 18		F	D	X	M	W											
add sp, gp, tp			F	D	X	M	W										
sw sp, 0(t1)				F	D	X	M	W									
lw t2, 4(t1)					F	D	X	M	W								
add ra, ra, t2						F	-	D	X	M	W						
add gp, t0, t2								F	D	X	M	W					
sub gp, gp, t2									F	D	X	M	W				
lw tp, 8(t1)										F	D	X	M	W			
add tp, tp, tp											F	-	D	X	M	W	
add tp, t1, t2													F	D	X	M	W

图二：截图 2

比较：实际执行中，两个平台停顿位置不同，ripes 在 IF 和 ID 之后停顿，而 webrisc-v 平台在 IF 阶段后就停顿。