### 实验5 基于RISC-V流水线CPU的指令执行过程在线仿真与冒险处理方式研究

**测试代码1任务1**、

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **代码段1**  模式5单周期执行周期数为\_\_\_65\_\_\_ | **模式1：**  **with forward**  **with flush** | **模式2：**  **no forward with flush** | **模式3：**  **with forward**  **no flush** | **模式4：**  **no forward**  **no flush** |
| **执行周期数** | 21 | 22 | 21 | 24 |
| **被执行forward的指令，执行forward次数和原因** | addi x6, x6, 2  **beq x6, x0, fi**  执行1次。  原因：beq x6, x0, fi依赖于addi x6, x6, -1中对x6值结果的更改，执行beq时前递了addi的aluout。 | / | addi x6, x6, 2  **beq x6, x0, fi**  执行1次。  原因：beq x6, x0, fi依赖于addi x6, x6, -1中对x6值结果的更改，执行beq时前递了addi的aluout。 | / |
| **被执行flush操作的指令和原因** | j loop  **fi:add x4, x4, x5**  原因：分支默认不跳转，而j loop发生了控制冒险，需要将已经在F阶段的add指令flush。  beq x6, x0, fi  **addi x6,x6,-1**  原因：分支默认不跳转，而退出循环时beq发生了控制冒险，需要将已经在F阶段的addi指令flush。 | j loop  **fi:add x4, x4, x5**  原因：分支默认不跳转，而j loop发生了控制冒险，需要将已经在F阶段的add指令flush。  beq x6, x0, fi  **addi x6,x6,-1**  原因：分支默认不跳转，而退出循环时beq发生了控制冒险，需要将已经在F阶段的addi指令flush。 | / | / |

在界面上依次选择模式1-5执行仿真，查看程序执行的周期数，对比分析不同模式下执行过程，对比不同模式下的异同，并与单周期执行过程和结果对比，查看并分析执行周期数不同或者结果不同的原因。指出被执行了forward 或者flush操作的具体指令和原因。 完成表1。

表1

表1

表2 测试程序1在模式1时候的执行表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 指令执行周期 | | | | | | | | | | | | | | | | | | | | | | |
| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| addi x6,x6,2 | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| loop: beq x6,x0,fi |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| addi x6,x6,-1 |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| addi x5,x5,3 |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| j loop |  |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| fi: add x4,x4,x5 |  |  |  |  |  | F | D | Flush | Flush | Flush |  | F | D | Flush | Flush | Flush | F | D | X | M | W |  |  |
| Nop  Nop |  |  |  |  |  |  | F | Flush | Flush | Flush | Flush |  | F | Flush | Flush | Flush | Flush | F | D  F | X  D | M  X | W  M | W |
| loop: beq x6,x0,fi |  |  |  |  |  |  |  | F | D | X | M |  |  | F | D | X | M | W |  |  |  |  |  |
| addi x6,x6,-1 |  |  |  |  |  |  |  |  | F | D | X | M | W |  | F | D | Flush | Flush | Flush |  |  |  |  |
| addi x5,x5,3 |  |  |  |  |  |  |  |  |  | F | D | X | M | W |  | F | Flush | Flush | Flush | Flush |  |  |  |
| j loop |  |  |  |  |  |  |  |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |

表2 测试程序1在模式2时候的执行表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 指令执行周期 | | | | | | | | | | | | | | | | | | | | | | | | |
| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| addi x6,x6,2 | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| loop: beq x6,x0,fi |  | F | D | - | - | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| addi x6,x6,-1 |  |  | F | - | - | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| addi x5,x5,3 |  |  |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| j loop |  |  |  |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| fi: add x4,x4,x5 |  |  |  |  |  |  |  | F | D | Flush | Flush | Flush |  | F | D | Flush | Flush | Flush | F | D | X | M | W |  |  |
| Nop  Nop |  |  |  |  |  |  |  |  | F | Flush | Flush | Flush | Flush |  | F | Flush | Flush | Flush | Flush | F | D  F | X  D | M  X | W  M | W |
| loop: beq x6,x0,fi |  |  |  |  |  |  |  |  |  | F | D | X | M |  |  | F | D | X | M | W |  |  |  |  |  |
| addi x6,x6,-1 |  |  |  |  |  |  |  |  |  |  | F | D | X | M | W |  | F | D | Flush | Flush | Flush |  |  |  |  |
| addi x5,x5,3 |  |  |  |  |  |  |  |  |  |  |  | F | D | X | M | W |  | F | Flush | Flush | Flush | Flush |  |  |  |
| j loop |  |  |  |  |  |  |  |  |  |  |  |  | F | D | X | M | W |  |  |  |  |  |  |  |  |

可见no forwarding相比于含forwarding模块，仅在第2、3条指令进入EX阶段前要等待addi x6, x6, 2通过WB阶段将x6寄存器的值更新，因为第2、3指令对addi x6, x6, 2有数据依赖。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **代码段1**  模式5单周期执行周期数为\_\_\_365\_\_\_ | **模式1：**  **with forward**  **with flush** | **模式2：**  **no forward with flush** | **模式3：**  **with forward**  **no flush** | **模式4：**  **no forward**  **no flush** |
| **执行周期数** | 91 | 141 | 95\* | 143\* |

表3

|  |  |  |  |
| --- | --- | --- | --- |
| PC值 | 指令 | 冒险的种类 | 执行的操作 |
| 04 | ori x4, x10, 10 | 数据冒险 | 从00指令前递x10的值 |
| 30 | sw x12, 0(x4) | 控制冒险 | flush清除指令 |
| 8c | add x18, x18, x19 | 数据冒险 | 从84指令前递x19的值 |
| 94 | bne x5, x0, loop | 数据冒险 | 从90指令前递x5的值，stall一周期 |
| 98 | slli x12, x18, 0 | 控制冒险 | flush清除指令 |
| 38 | sub x18, x19, x12 | 数据冒险 | 从34指令前递x19的值，stall一周期 |
| 40 | loop2: addi x5, x5, -1 | 数据冒险 | 从3c指令前递x5的值 |
| 44 | ori x18, x5, -1 | 数据冒险 | 从40指令前递x5的值 |
| 48 | xori x18, x18, 1365 | 数据冒险 | 从44指令前递x18的值 |
| 50 | addi x20, x19, -1 | 数据冒险 | 从4c指令前递x19的值 |
| 54 | ori x16, x20, x19 | 数据冒险 | 从50指令前递x20的值 |
| 5c | and x17, x20, x16 | 数据冒险 | 从54指令前递x |
| 68 | shift: addi x5, x0, -1 | 控制冒险 | flush清除指令 |
| 6c | slli x18, x5, 15 | 数据冒险 | 从68指令前递x18的值 |
| 70 | slli x18, x18, 16 | 数据冒险 | 从6c指令前递x18的值 |
| 74 | srai x18, x18, 16 | 数据冒险 | 从70指令前递x18的值 |
| 78 | srli x18, x18, 15 | 数据冒险 | 从74指令前递x18的值 |
| 80 | sum: add x18, x0, x0 | 控制冒险 | flush清除指令 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 代码段1 | 分析：单周期CPU架构下，执行需（ 65 ）个时钟周期，  执行后，x6=（ 0 ），x5=（ 6 ），x4=（ 6 ） | | | |
| 实际执行仿真情况 | 模式1：  with forward  with flush | 模式2：  no forward with flush | 模式3：  with forward  no flush | 模式4：  no forward  no flush |
| Reg X6= | 0 | 0 | -1 | -1 |
| Reg X5= | 6 | 6 | 6 | 6 |
| Reg X4= | 6 | 6 | 15 | 15 |
| 执行结果正确与否 | 正确 | 正确 | 错误 | 错误 |
| 若不正确如何修改程序可以获得正确结果 | \ | \ | 由于未进行flush，导致循环过程中j loop后的add x4, x4, x5每次都被错误执行了一次，退出循环时beq后的addi x6, x6, -1被错误执行一次，导致x4, x6错误。可以在j loop和beq x6, x0, fi后各添加一条addi x0, x0, 0(nop)来获得正确结果 | |
| 代码段1 | 分析：单周期CPU架构下，执行需（ 65 ）个时钟周期，  执行后，x6=（ 0 ），x5=（ 6 ），x4=（ 6 ） | | | |
| 实际执行仿真情况 | 模式1：  with forward  with flush | 模式2：  no forward with flush | 模式3：  with forward  no flush | 模式4：  no forward  no flush |
| Reg X6= | 0 | 0 | -1 | -1 |
| Reg X5= | 6 | 6 | 6 | 6 |
| Reg X4= | 6 | 6 | 15 | 15 |
| 执行结果正确与否 | 正确 | 正确 | 错误 | 错误 |
| 若不正确如何修改程序可以获得正确结果 | \ | \ | 由于未进行flush，导致循环过程中j loop后的add x4, x4, x5每次都被错误执行了一次，退出循环时beq后的addi x6, x6, -1被错误执行一次，导致x4, x6错误。可以在j loop和beq x6, x0, fi后各添加一条addi x0, x0, 0(nop)来获得正确结果 | |