练习 1 . 对于第一种情况（**通过转发解除数据冒险**）中的程序，针对两条相互依赖的指令彼此之间距离不同的情况执行与第一种情况仿真波形中相同的分析。可以通过更改两条相关add指令之间的nop数量来控制距离。

此外，需创建第一个输入操作数接收转发数据的其他示例。

还可以创建两条add指令通过I1管道执行的其他示例，但需确认行为相同。

最后，将相关add指令（add t3,t3,t4）替换为通过其他管道执行的其他相关指令并分析仿真结果。例如，可以包含以下指令之一来代替第二条add指令：

- lw t3, (t4)（强制读取值来自DCCM）

- mul t3, t3, t4

- div t3, t3, t4

新仿真程序的示例：从EX2阶段旁路到译码阶段：

1a0: 01ee8eb3 **add t4,t4,t5**

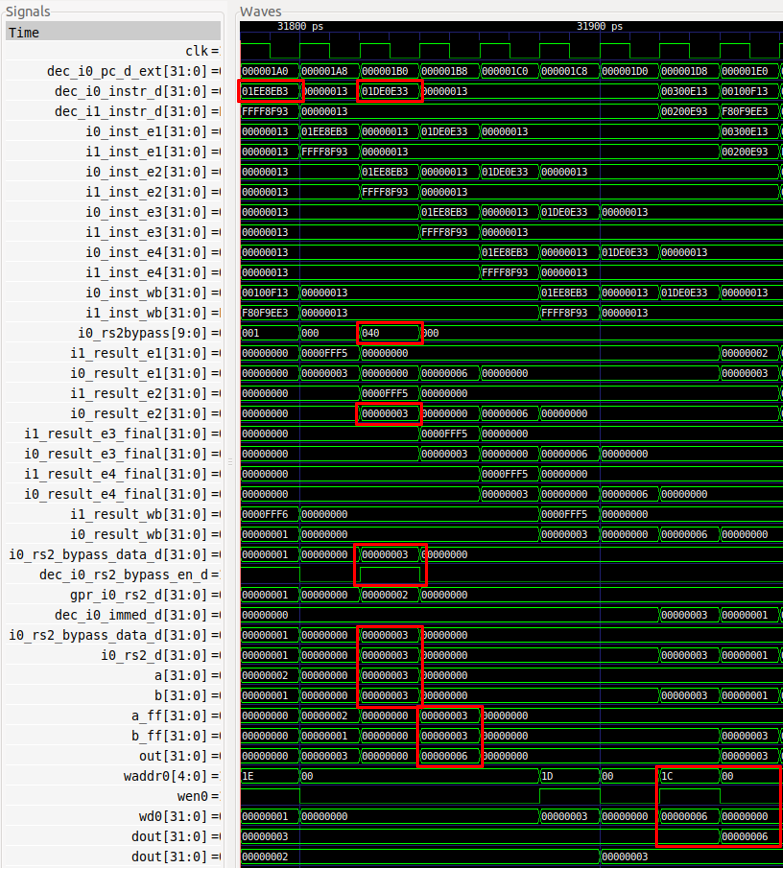
1a4: ffff8f93 addi t6,t6,-1

1a8: 00000013 nop

1ac: 00000013 nop

1b0: 01de0e33 **add t3,t3,t4**

1b4: 00000013 nop

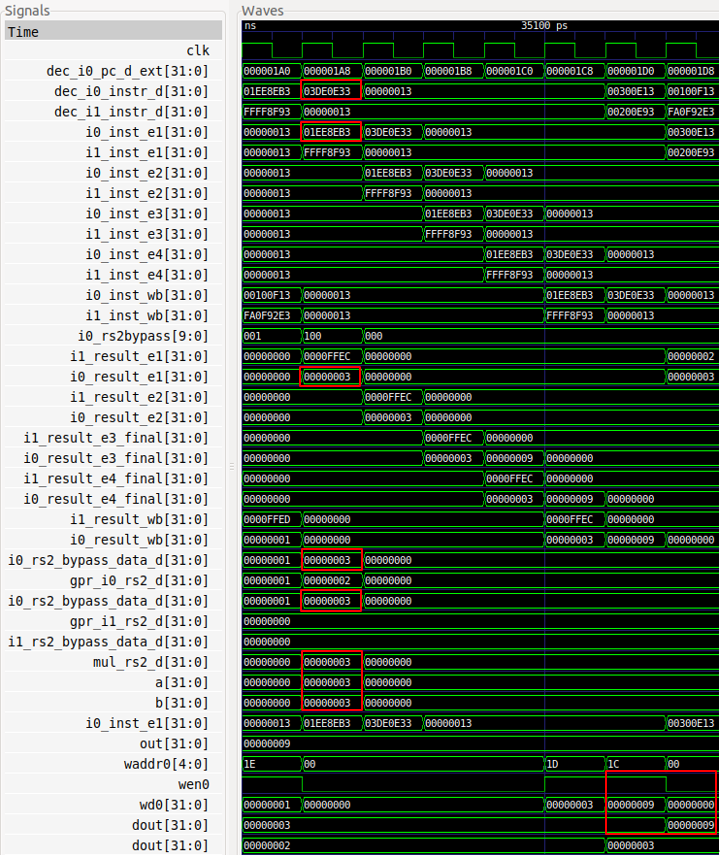


新仿真程序的示例：执行mul指令而不是第二条add指令：

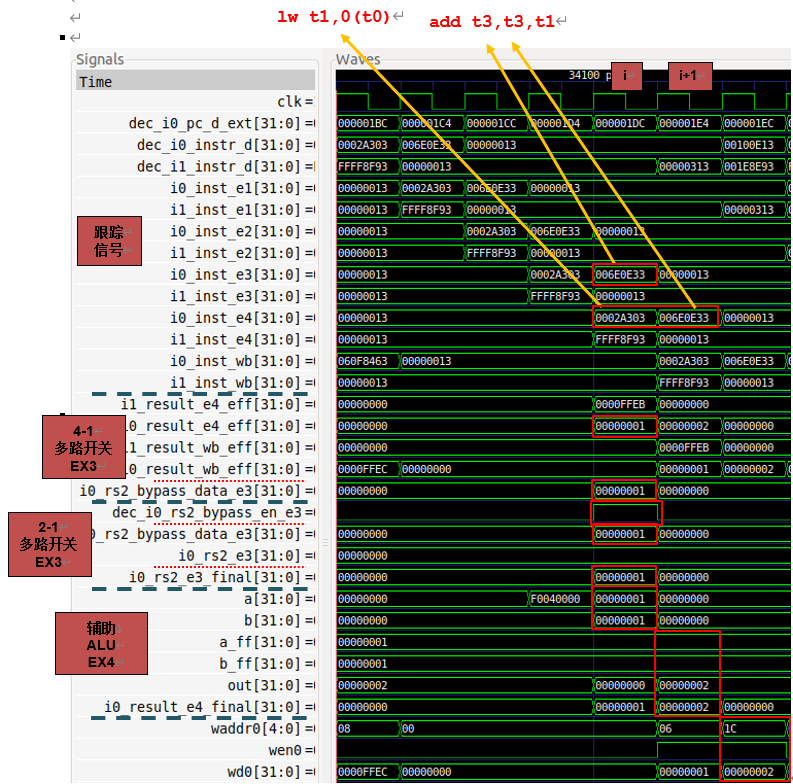
1a0: 01ee8eb3 **add t4,t4,t5**

1a4: ffff8f93 addi t6,t6,-1

1a8: 03de0e33 **mul t3,t3,t4**



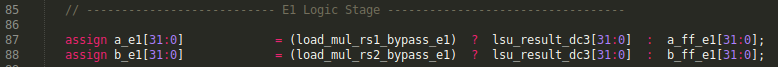
练习 2. 在自己的计算机上重复第二种情况（通过在提交阶段转发解除数据冒险）的仿真过程。可以使用以下位置提供的.tcl文件：[RVfpgaPath]/RVfpga/Labs/Lab4/Solutions/DataHazards\_Close-LW-AL/scriptLoad.tcl



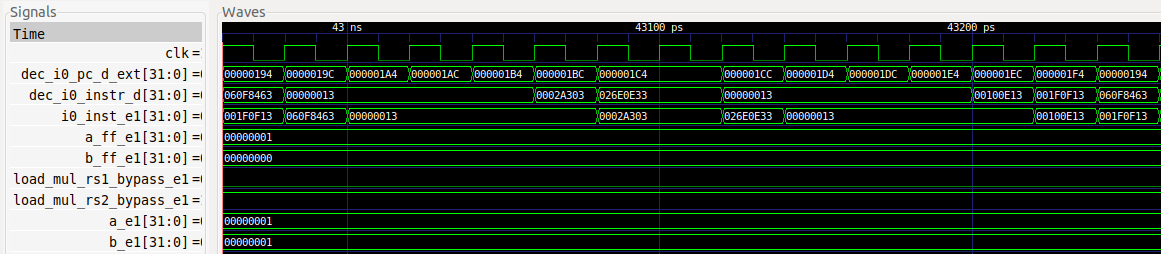
练习 3. 分析lw指令后跟与装载读取的值相关的mul指令这种情况。在第二种情况的程序中，只需将相关add指令替换为mul指令。

可以使用以下位置提供的程序：[RVfpgaPath]/RVfpga/Labs/Lab4/Solutions/DataHazards\_Close-LW-MUL

辅助ALU无法执行mul指令。乘法器（模块**exu\_mul\_ctl**）内部实现了一个新的旁路路径，用于将装载读取的值转发到M1阶段。



这样一来，由于这种RAW相关性，只有1个周期丢失。



第二个操作数从装载读取的值旁路。这样一来，由于相关性的原因，只有1个周期丢失。