Jordan Zannier 997694810
Naif Tarafdar 997590566

1. Calculated CPI with stalls = 1.6392
   CPI = total number of cycles/total number of instructions
   = (total instructions + num of 1 cycle stalls + 2 * num of 2 cycle stalls)/total instructions
   = 1.6392
   = 63.92 % slowdown

2. Calculated CPI with stalls = 1.4005
   CPI = (total instructions + num of 1 cycle stalls + 2 * num of 2 cycle stalls)/total instructions
   = 1.4005
   =40.05 % slowdown

   The two cycle stall was a load to use hazard because it's WX forwarding. The one cycle stall is a normal raw hazard which uses MX forwarding. A store can result only in a stall if the second source register is a RAW dependency, since a dependency on the first register can be bypassed (WM).

3. Calculated CPI with stalls = 1.2067
   CPI = (total instructions + num of 1 cycle stalls + 2 * num of 2 cycle stalls)/total instructions
   = 1.2067
   =20.67 % slowdown

   The one cycle stall can be caused by a WAW dependency with 1 interweaved independent instruction. The one cycle structural hazard is a memory operation followed by two independent ALU operation, where the second ALU operation will create a 1 cycle stall. The two cycle stall are two consecutive instructions with a WAW dependency or 2 or more consecutive memory operations followed by an ALU operation. In the latter case two consecutive memory operations followed by a WAW hazard will result in a 2 cycle stall WAW (initially 1 cycle stall then 1 more because of a structural hazard), while two consecutive memory operations followed by an independent ALU operation will result in a 2 cycle structural hazard stall.

Benchmark for Question 1 Explanation

We used the O1 compilation flag. We have two loops, the first loop creates a 2 cycle stall for MAX_2CYCLE iterations. The second loop creates a 1 cycle stall for MAX_1CYCLE iterations.

To calculate 2 cycle stall it would be 2 consecutive dependent instructions. First instruction writes to a register, while the second reads from the same register. Refer to line 53 which points out that variable a is being written to and then read from by the subsequent instruction on line 54. This will cause a RAW hazard that forces the pipeline to stall for 2 cycles. Refer to the comments from line 42 to 48 which has the assembly code clearly showing register 2 having the RAW hazard.

To calculate 1 cycle stall it would be 2 dependent instructions seperated by 1 independent instruction. Refer to line 72 which points out that variable h is written to, and then on line 73 an independent instruction is added for padding. The next instruction on line 74 reads from h, and causes a RAW hazard that forces the pipeline to stall for 1 cycle. Refer to the comments from line 62 to 68 which has the assembly code clearly showing register 2 having the RAW hazard.