Q2

(I) 10 clock ticks needed, 6 instructions executed, r1 is 0x15

(II) 18 clock ticks, 6 instructions executed, r1 is 0x15

(III) 10 ticks, 6 instructions executed, r1 is 0x06

The needed clock cycles in part (II) are different (more of them) because without ALU forwarding, if we use same register for evaluations in a row, we need to stall the pipeline on ID phase until value from previous instruction was written back into register.

In part (III) without forwarding or interlock we have same amount of cycles as (I) because we don't have any stalls. However, the result value in r1 is incorrect, since we use incorrect values (old ones, didn't update yet) for computations, since it hasn't been updated yet into register from previous instruction. That's why we get smaller value than expected.

Q3

38 instructions and 50 ticks, we have to stall when we don't hit branch target buffer (that happens when we have 3rd different branch, since our buffer can only have 2 entries, so that oldest one gets pushed out) and we also need to stall when LD is executed to avoid data hazard in SRLi instruction, which operates on register into which LD instruction loads, so we wait until LD result is forwarded to O1. 4 LD Stalls, 4 Stalls before branching and then it takes 4 initial ticks for first instruction to make it through the pipeline, that makes it 12 ticks difference. Since we have 4 iterations we have 4 stalls each.

We now required 53 ticks, since every time we took the branch, we stalled now, to calculate new branching address, because it's not saved in a buffer now. It's only increased by 3 ticks because we didn't take every branch in the program as well as since we had 4 iterations and on first one we stall cause buffer is empty (in (I)), therefore we only save 3 cycles in (I). Nonetheless, it's still an increase.

It take 46 ticks since now we don't have to stall for LD which was used by first(now second) shift. Instead we perform other shift and by the time we get to SRLi, value R2 is already loaded and is in O1. Therefore we reduce the time by 4 ticks (cause we had 4 iteration), which were LD stalls.