Without loop unroll:

ALU or Branch (1st slot)	LW or SW (2nd slot)
addi $s0, $s0, 4	lw $t0, 0($s0)
NOP	NOP
NOP	lw $t1, 0($t0)
bne $s0, $s2, LOOP	NOP
add $t1, $s1, $t1	NOP
NOP	sw $t1, 0($t0)

Explanation: since addi has no dependency upon the previous lines, we move it to the first row on the left. The second lw has to be stalled one cycle, and requires a forwarding from after MEM to before EX. The add after second lw also needs to be stalled for one cycle, and requires a forwarding from after MEM to before EX. Sw can be executed on the line below add, and requires a forwarding from after EX to before MEM. A delayed branch always executes the instructions in delay slots, but the instructions following the delayed slots will be affected by the branch. Since the spec requires us to use delay slots and branch takes two cycles to resolve, we put bne two cycles before the end of the loop.

We need 200*6+4=1204 cycles for 200 iterations, the plus 4 comes from finishing the last 4 stages of the last instruction.

With loop unroll:

ALU or Branch (1st slot)	LW or SW (2nd slot)
addi $s0, $s0, 8	lw $t0, 0($s0)
NOP	lw $t2, -4($s0)
NOP	lw $t1, 0($t0)
NOP	lw $t3, 0($t2)
add $t1, $s1, $t1	NOP
bne $s0, $s2, LOOP	NOP
add $t3, $s1, $t3	sw $t1, 0($t0)
NOP	sw $t3, 0($t2)

Explanation: Different from the previous implementation, here have addi to be incremented by 8, which is the sum of 4+4 of the two unrolled loops. Then, we use register renaming to have register t2 replace register t1 on the second line. The lw on the second line have its immediate to be -4, because we moved addi to the first line and we want 8-4=4, which is the correct immediate before unrolling. The lw on the fourth line also used register renaming. add and sw have dependency requirements same as the previous implementation. A delayed branch always executes the instructions in delay slots, but the instructions following the delayed slots will be affected by the branch. Since the spec requires us to use delay slots and branch takes two cycles to resolve, we put bne exactly two cycles before the end of the loop.

We need (200/2)*8+4=804 cycles for 200 iterations, the plus 4 comes from finishing

the last 4 stages of the last instruction.
.