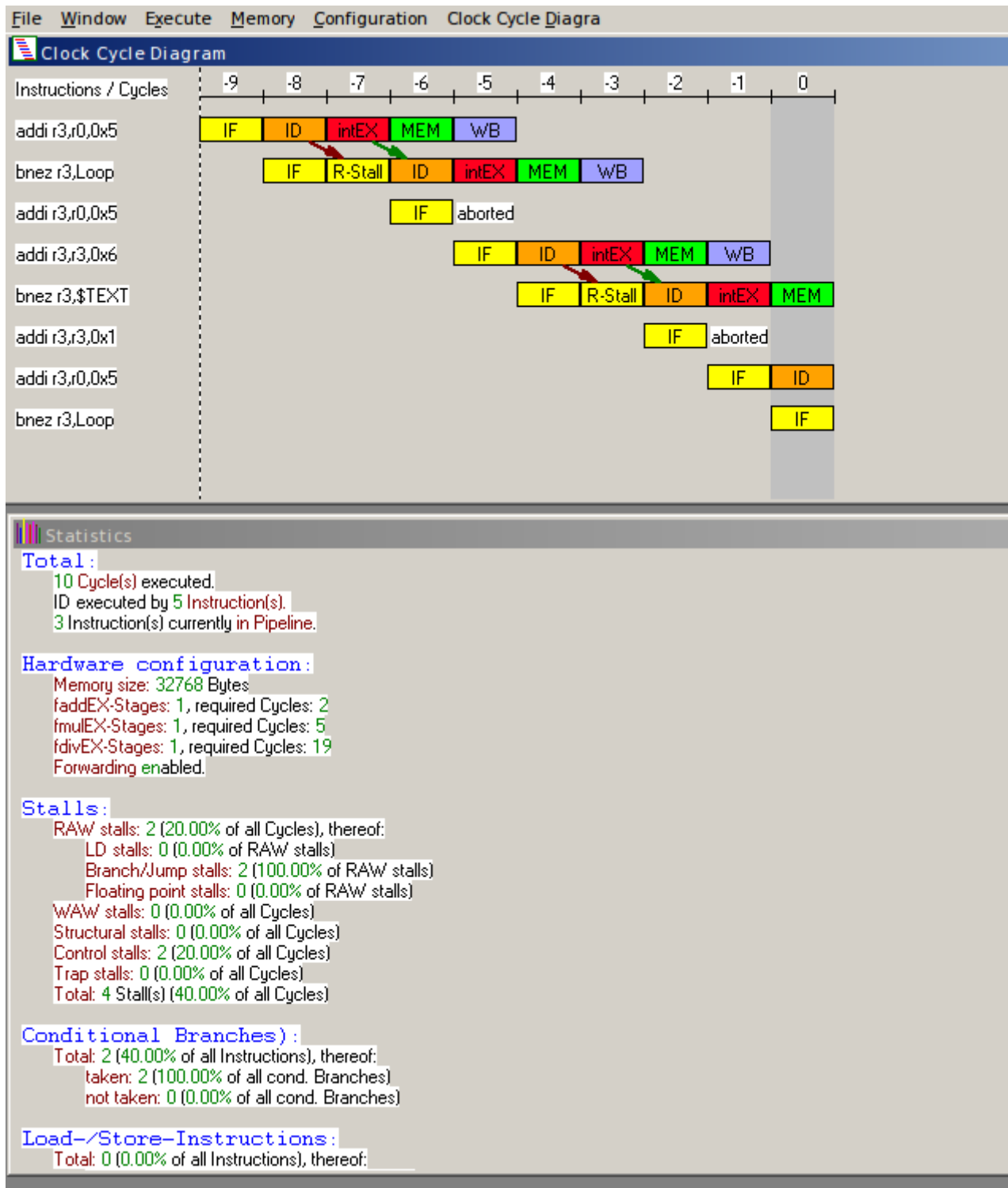


0a.



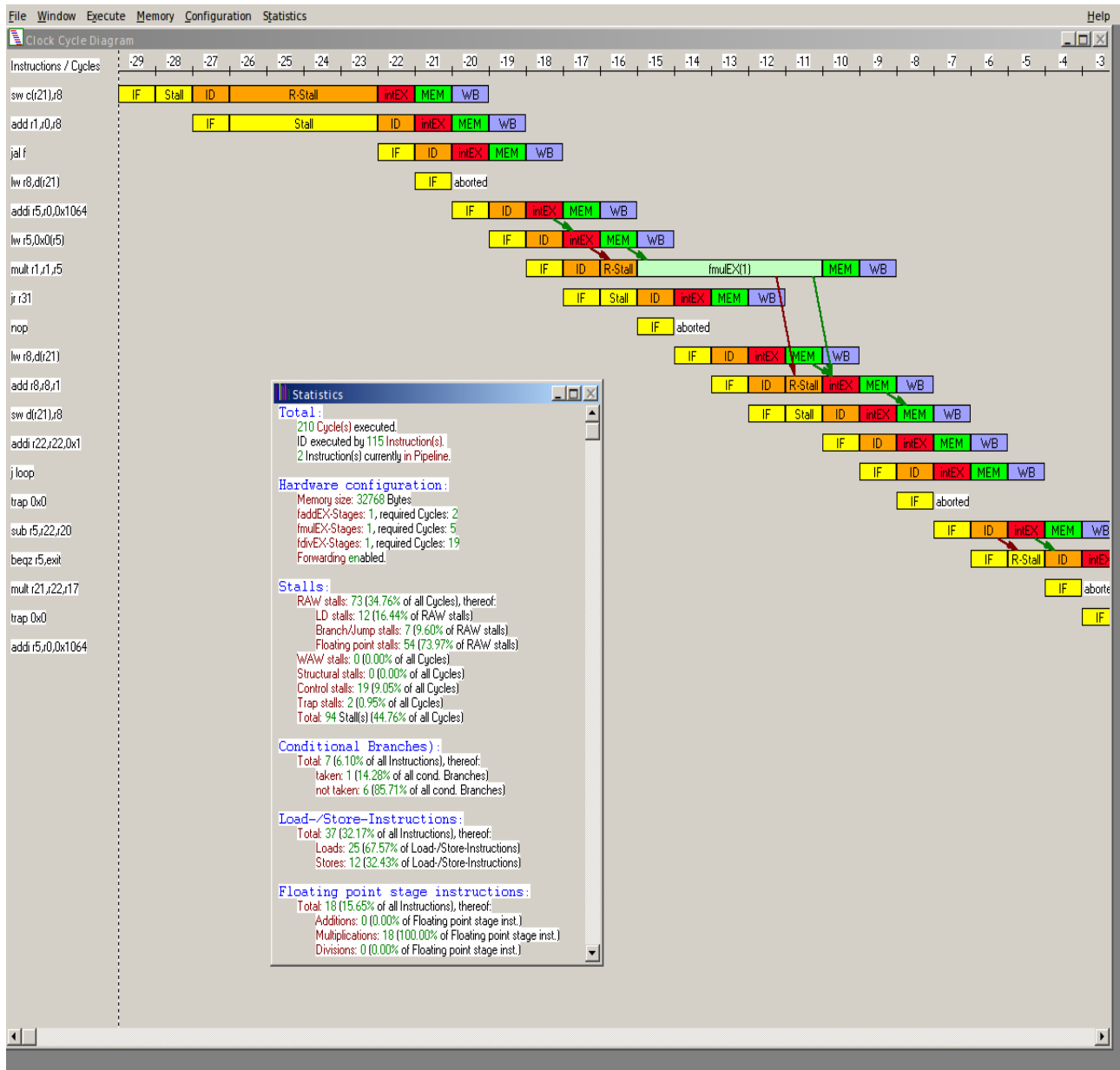
Control Stall occurs due to jump statement. The statement next to jump statement gets executed in its IF stage, and gets aborted, if the control signal jumps to some other block. Causing control stalls.

0b.

It is not possible to have stall in ID stage, as the branch instructions finishes in ID stage itself, so results gets forwarded from previos to next instruction.

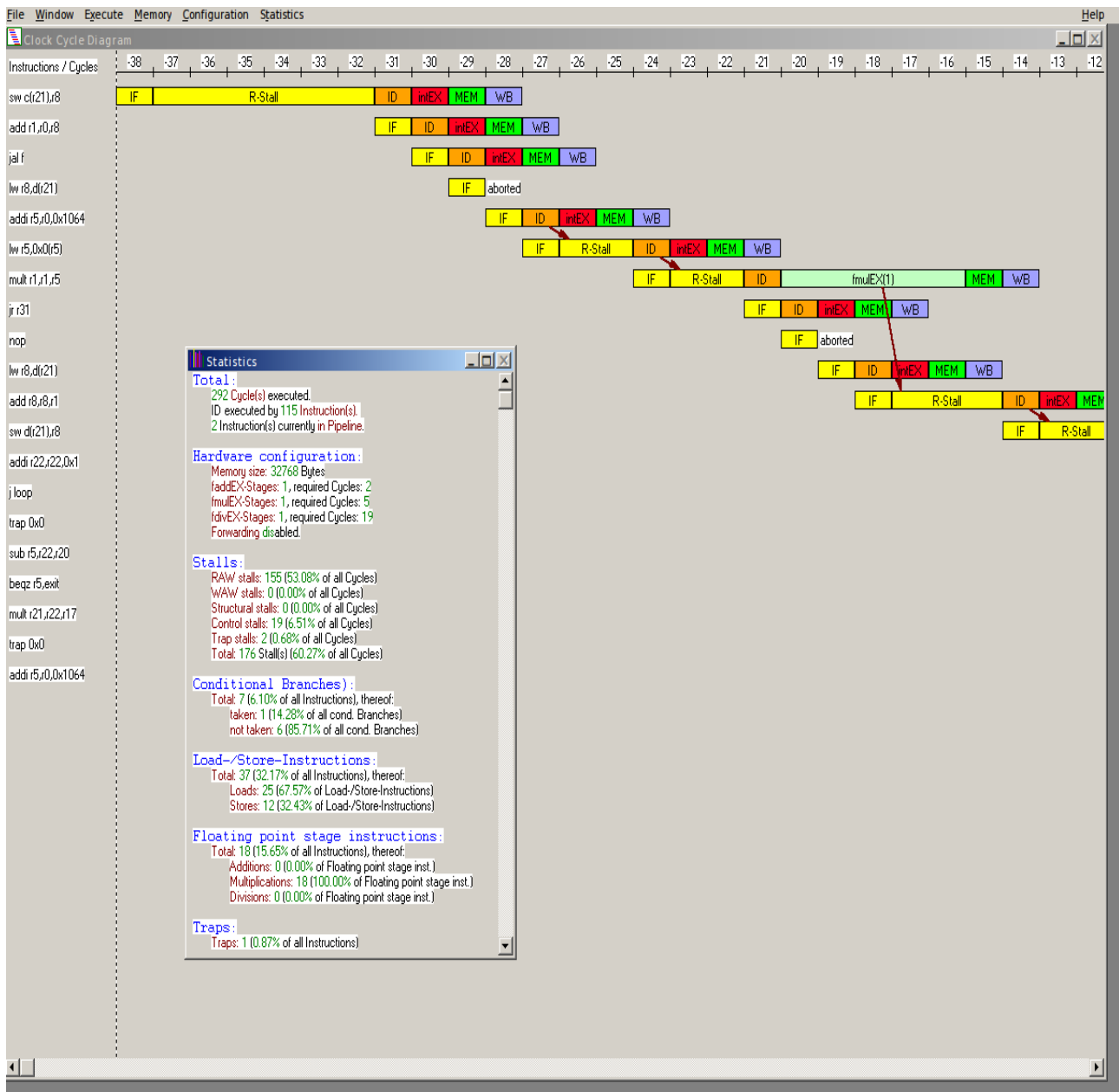
1a.

forwarding enabled: 210 cycles



1a.

forwarding disabled: 292 cycles

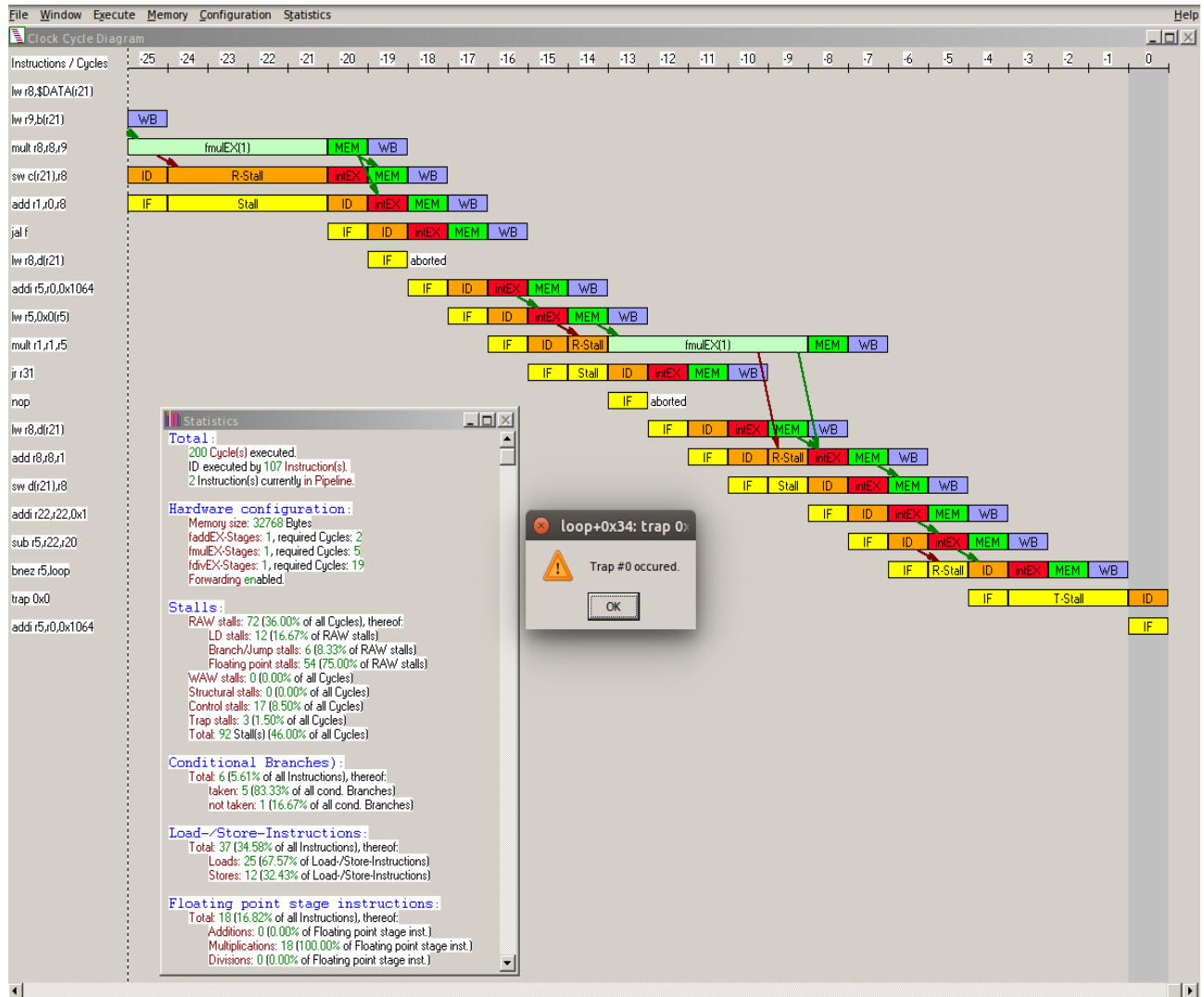


sped up due to data forwarding = $292/210$
 = 1.39047619 times dereament

percentage decrease in cycle due to forwarding = $(292-210)*100/292$
 = 28.08%

1b.
one branch statement
forwarding enabled

no of cycles = 200



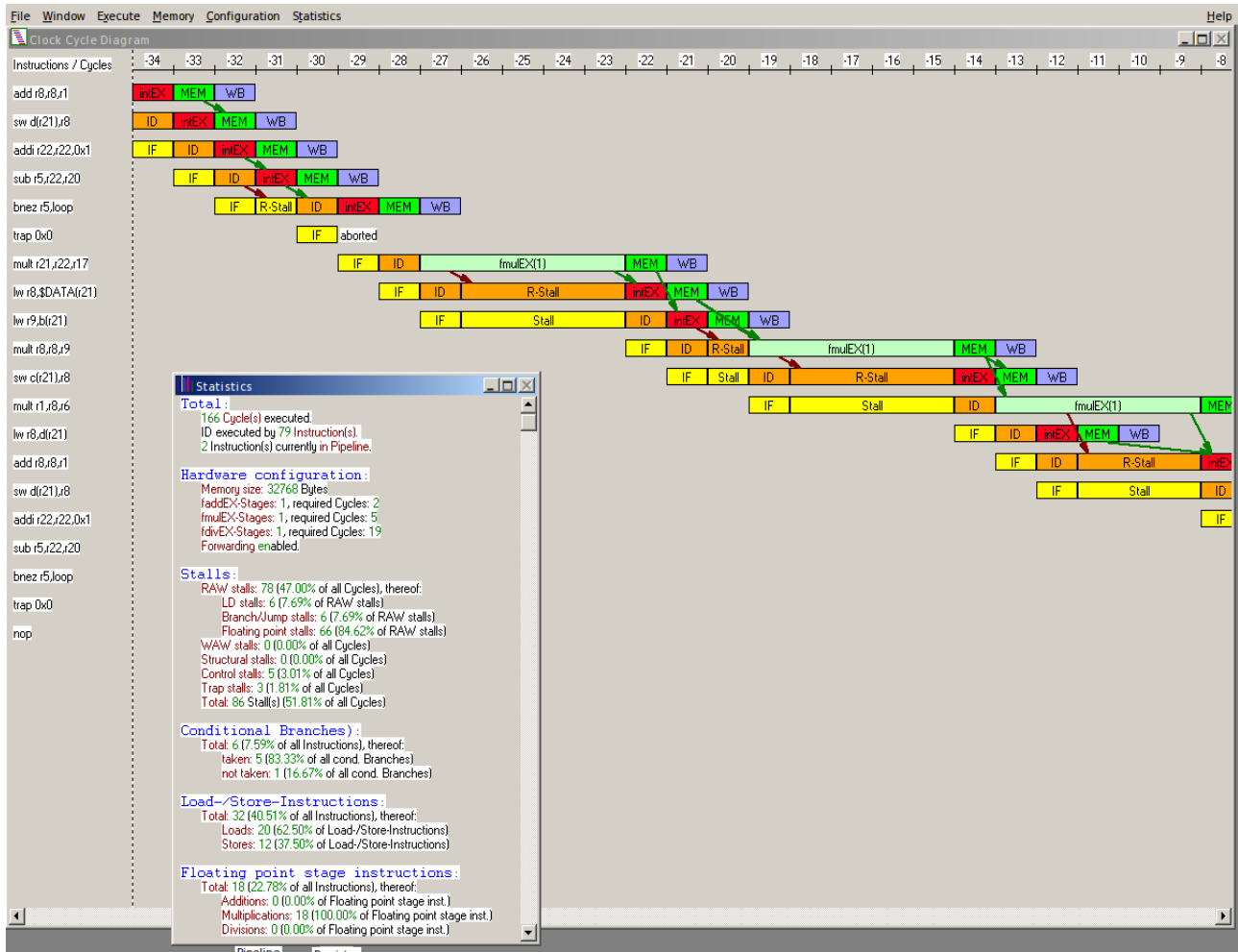
two branch instructions, forwarding enabled, no of cycles = 210
one branch instructions, forwarding enabled, no of cycles = 200

decrease in no of cycles = 10

%speed up = $(210-200) \times 100 / 210$
= 4.76% approx

2.a.
single branch instruction
forwarding enabled
inline

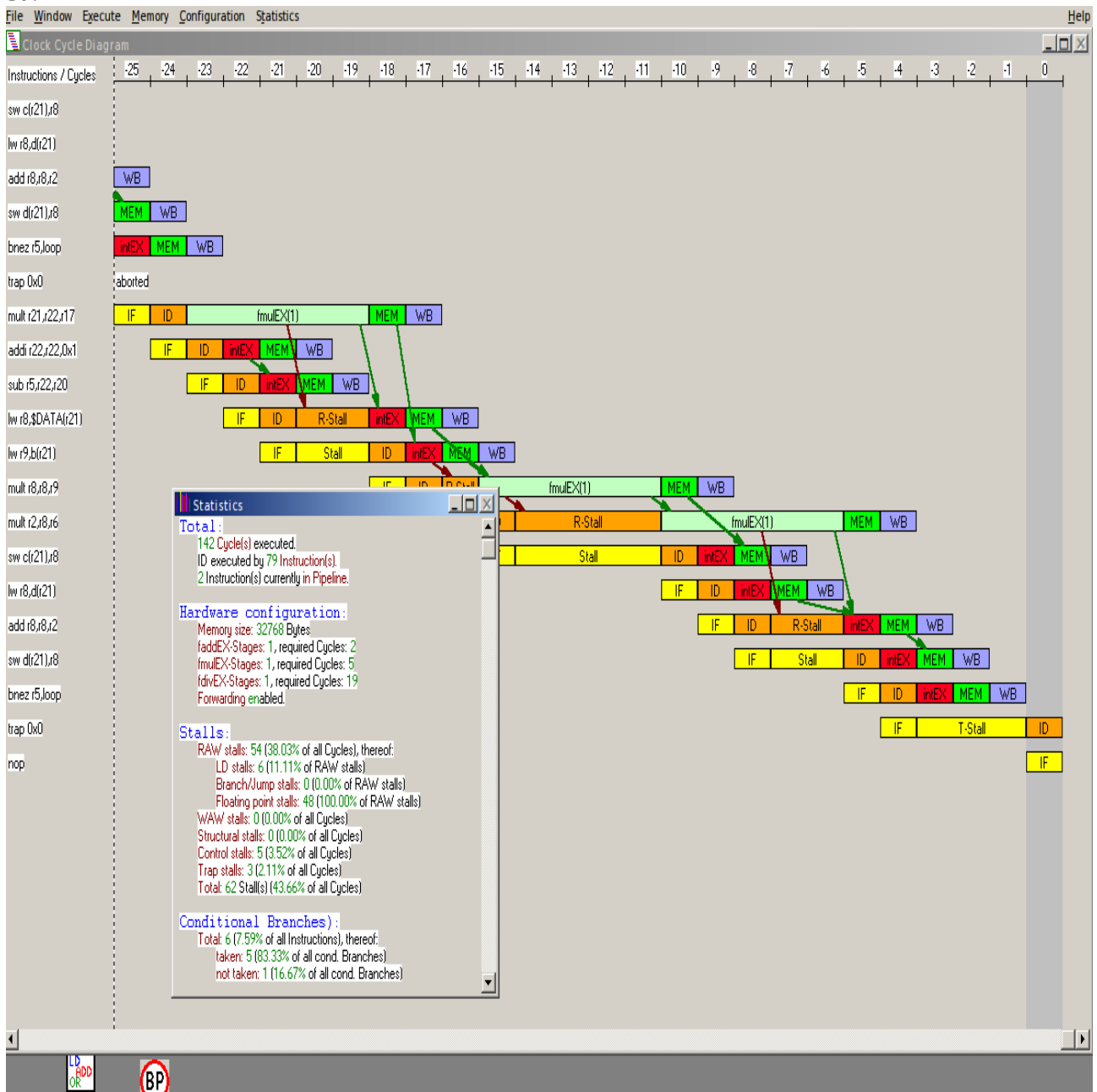
no of cycles = 166



no of cycles in 1b = 200

% decreament in cycle count from 1b = $(200-166)*100/200$
= 17%

3a.



for scheduled code:

no of cycles = 142

no of stalls = 62

for unscheduled code:

no of stall in 2a=86

no of cycles in 2a=166

decrease in number of stalls = 24

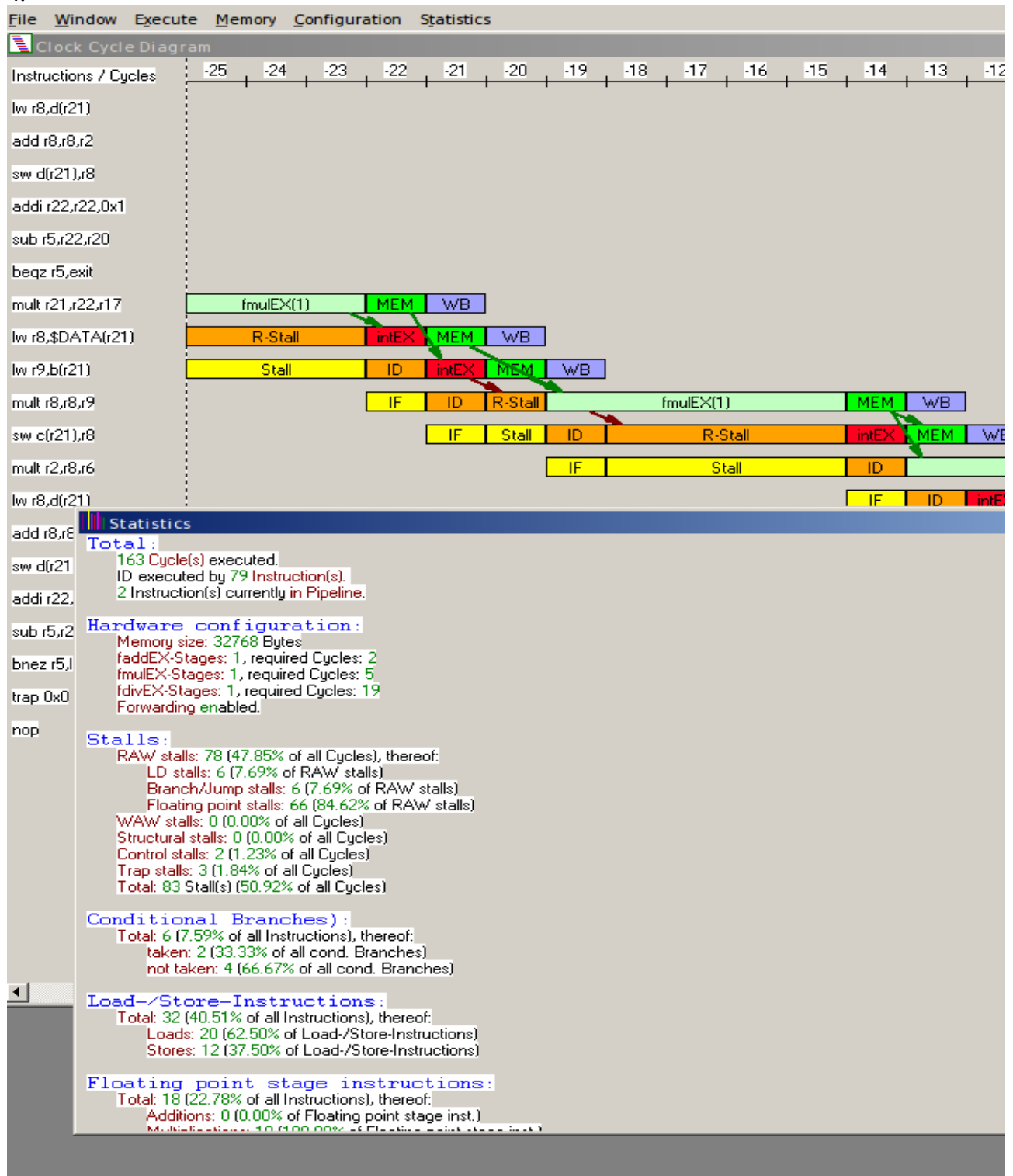
% decrease in stalls = $24 \times 100 / 86$

= 27.9% approx

3b.

Not possible to decreasing number of stalls , with scheduling of instruction, as every next instruction now requires the previos instruction to get completed for the correctness of program, thus no more scheduling is possible.

4.

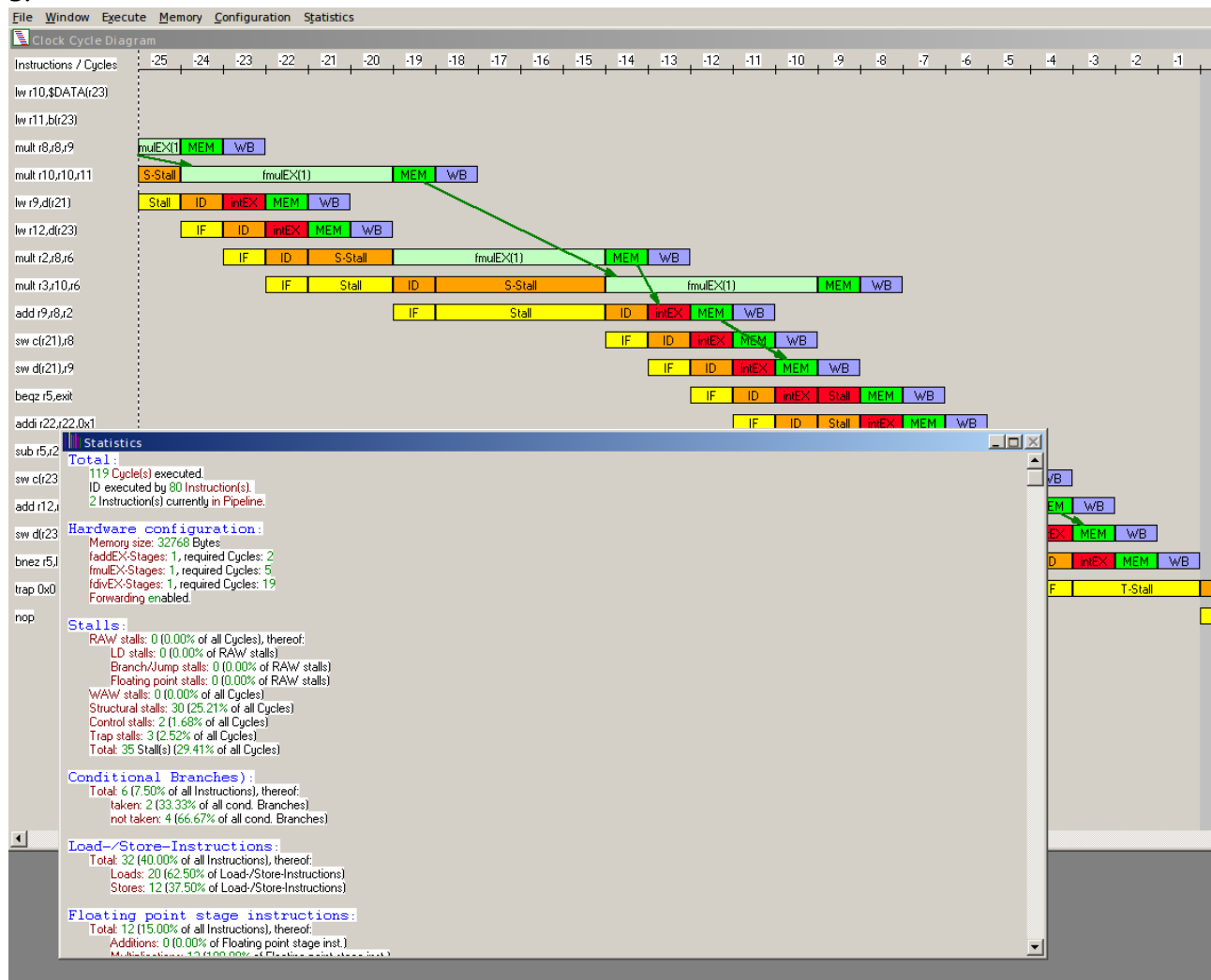


no of stalls with unrolling = 83
no of cycles with unrolling = 163

no of stall without unrolling = 86
no of cycles without unrolling = 166

cycles is reduced by 3, as stalls decreased by 3. This is because the number of jump statement executed reduced by 3.

5.



no of cycles = 119
no of stalls = 35