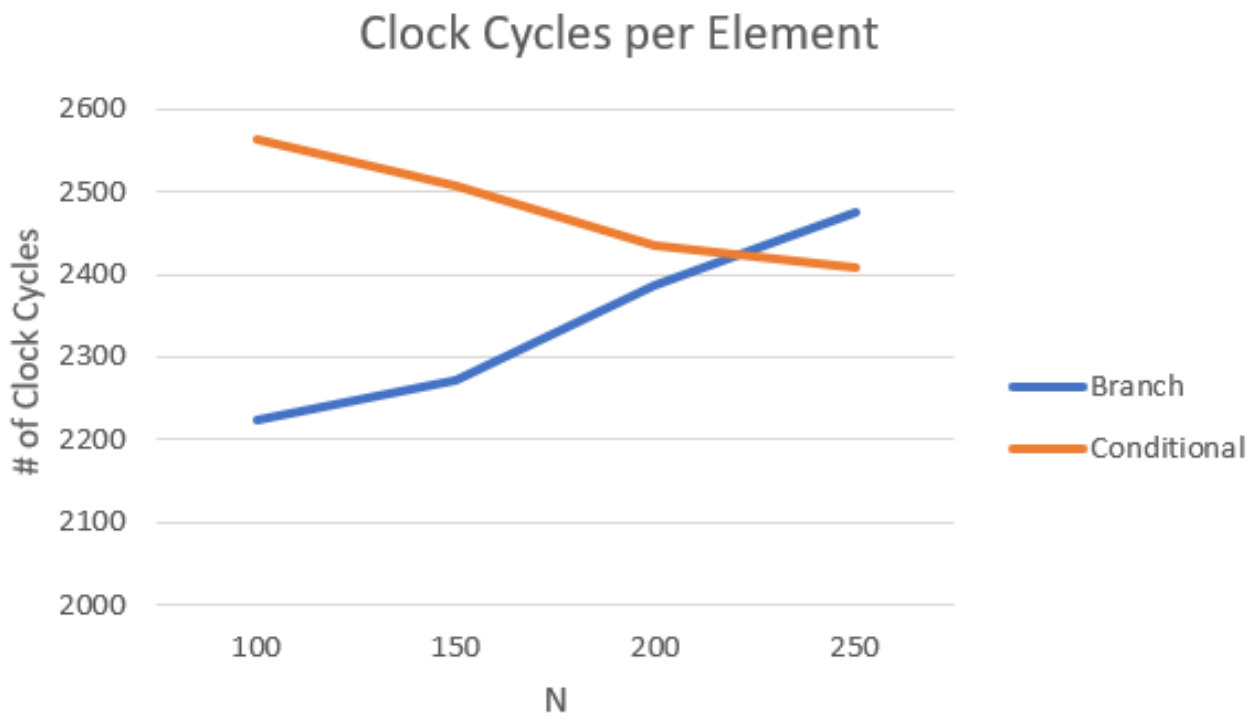
Version 1- Branch

N	t1	t2	t3	t4	t5	μ
100	1923	2037	2288	2390	2479	2223.4
150	2316	2459	2054	2088	2436	2270.6
200	2439	2392	2363	2384	2357	2387
250	2493	2511	2506	2403	2465	2475.6

Version 2- Conditional Move

N	t1	t2	t3	t4	t5	μ
100	2480	2644	2694	2582	2412	2562.4
150	2754	2404	2440	2567	2376	2508.2
200	2334	2367	2417	2403	2653	2434.8
250	2495	2433	2401	2255	2457	2408.2

*Notice: All the values in t1,t2,t3,t4,t5 and μ take the unit Clock Cycles.



Slope(Branch) =(y2-y1)/(x2-x1)

=(2475.6-2223.4)/(250-100)

=1.63 CPE (clock cycles per element)

Slope (Conditional Move) = (y2-y1)/(x2-x1)

= (2408.2-2562.4)/(250-100)

=-1.028 CPE (clock cycles per element)

My conclusion:

For version 1 – Branch, as the N size increased, the average clock cycle increased. The clock cycles increased when my size N increased using branching instructions because of the branching hazards introduced by these instructions. Branching instructions introduce unpredictable jumps in the code, making it difficult for the microprocessor to optimize the execution of instructions and causing unnecessary time execution. When a branch instruction is encountered, the processor must evaluate a condition to determine which branch to take, and this evaluation can introduce a delay or a "stall" in the pipeline, which can increase the overall execution time. As a result, the slope is positive and the clock cycle per element continues to increase as N increases.

However, for version 2- Conditional Move, as the N size increased, the average clock cycle decreased. The clock cycles decreased as N increased due to a phenomenon called "instruction-level parallelism". This allows the microprocessor to execute multiple instructions in parallel or out of order, thereby reducing the overall execution time. In particular, the use of cmov instructions may help to increase ILP by reducing the number of branch instructions in the code and does not stall the pipeline like branching instructions which can cause mispredictions and wastage of time. In summary, the decrease in clock cycles as N increases is due to a combination of factors, including the use of cmov instructions and the ability of microprocessors to exploit ILP. Hence, the slope is negative and the clock cycle per element continues to decrease as N increases.