

Senura Dissanayake

101283013

Student 1

Samuel Nieuwenhuis

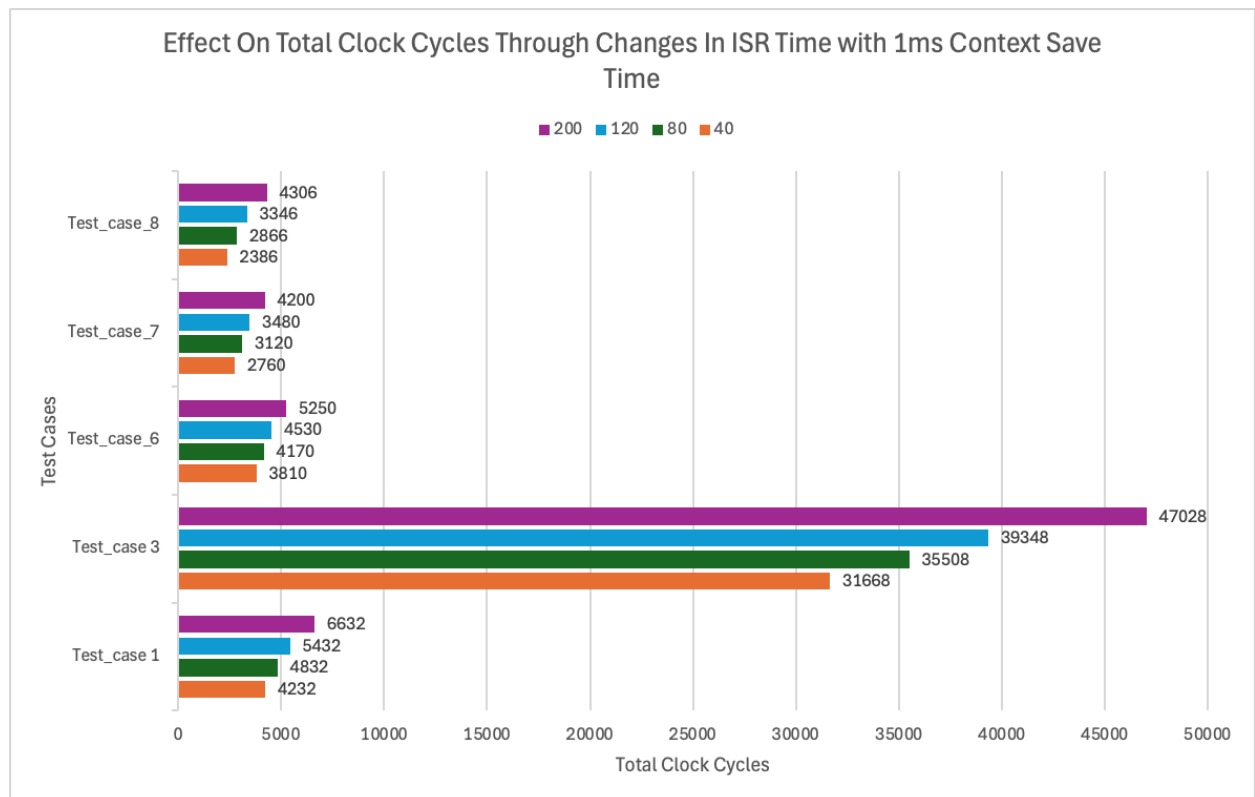
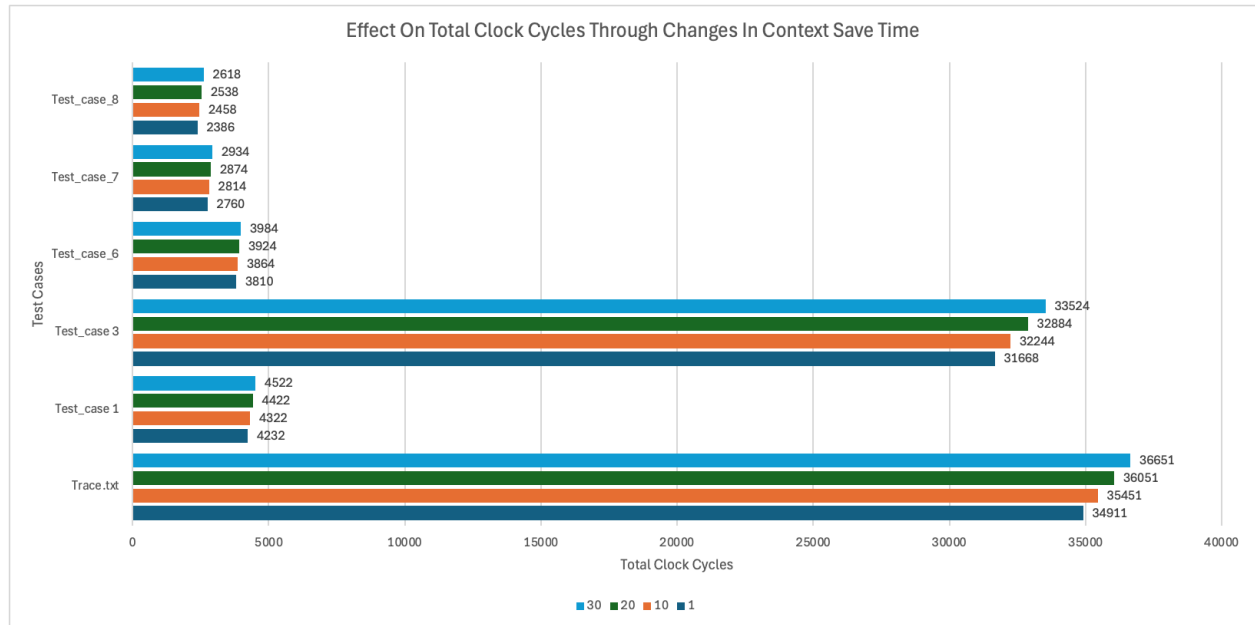
101270254

Student 2

Increasing the save/restore context time from 10 to 30 ms seemed to give us an increase in clock cycles of roughly 1.6% per extra 10 ms. For example in `trace_txt_base.txt`, we can see that running the program using `trace.txt` with no change in save/restore context time gives us a total of 34911 clock cycles. Then, as seen in `trace_txt_10ms_context.txt`, the number of clock cycles increases to 35451. Then, increasing it to 20 ms gives us a total of 36051 cycles, as shown in `trace_txt_20ms_context.txt`. Finally, an increase to 30 ms brings the total number of clock cycles to 36651, as seen in `trace_txt_30ms_context.txt`. So the increase is a fairly consistent value of around 1.6% per extra 10 ms.

When the ISR execution time increases, the system seems to spend more time handling interrupts and less time performing normal processing. As seen in `trace_1_40ms_ISR.txt`, where the ISR activity time is set to 40ms, the overall number of cycles is 4232. Then, when we increase it to 80 ms, as seen in `trace_1_80ms_ISR.txt`, it causes an increase in overall time to 4832 total cycles. Finally, when we increase ISR activity time to 200 ms, we can see from `trace_1_200ms_ISR.txt` another increase in overall time taken, bringing the total cycles to 6632.

Increasing the save/restore context time and the ISR activity duration both slow down the overall system performance. A higher context switch time means more overhead each time an interrupt occurs, reducing the CPU's availability for normal processing. Similarly, longer ISR activity times cause the processor to spend more time inside the interrupt routine, delaying the execution of regular tasks and possibly blocking other interrupts. Together, these changes increase total execution time, reduce system responsiveness, and can lead to performance bottlenecks, especially in systems with frequent interrupts or time-sensitive operations.



Link to repository: https://github.com/SamuelNieuwenhuis61003/SYSC4001_A1