PROJECT REPORT – SUSHANT TRIVEDI (ASU ID: 1213366971)

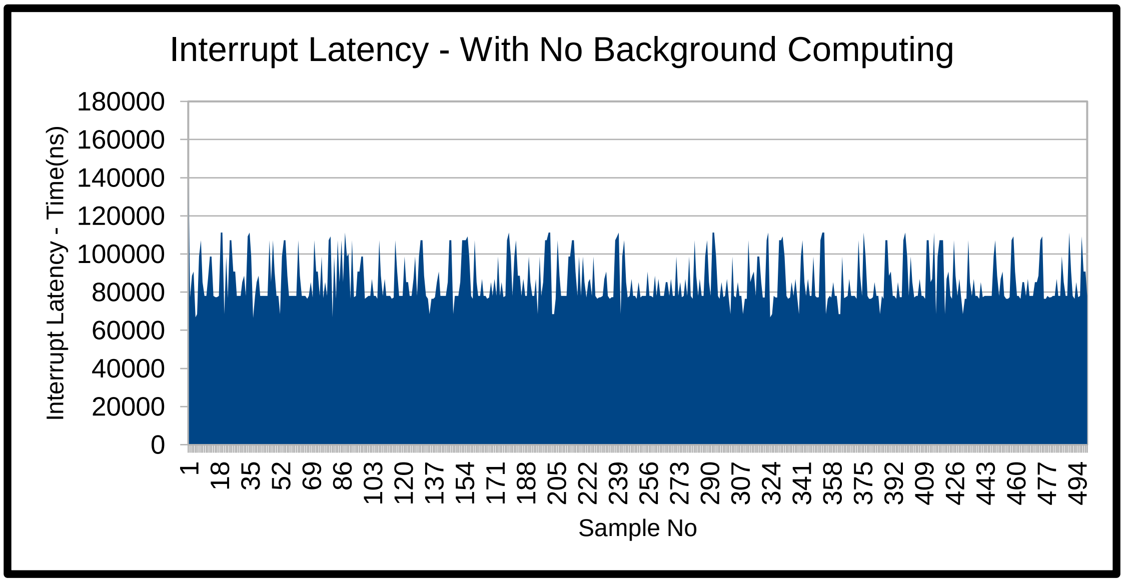
We have successfully implemented the 3 objectives of the Assignment-3

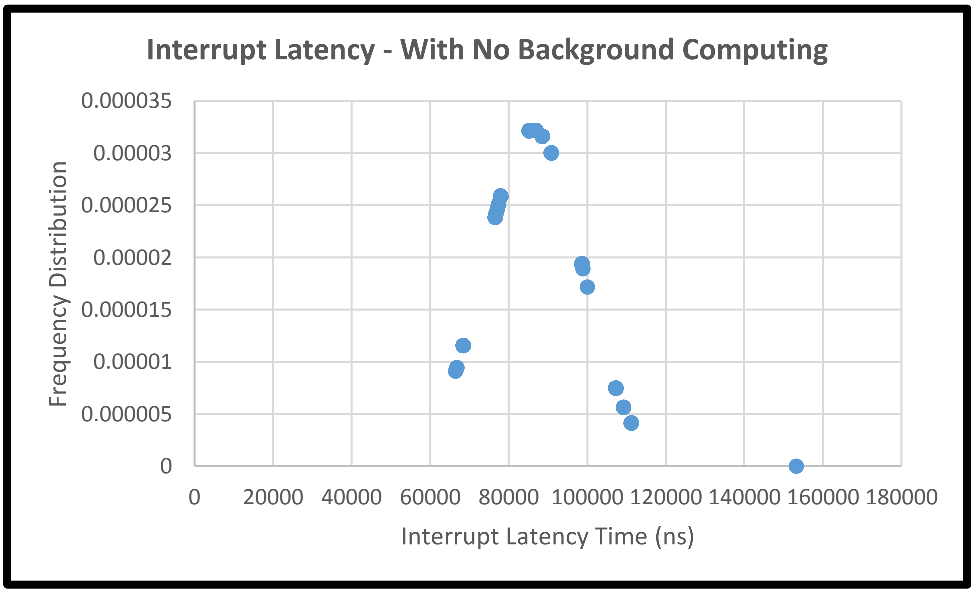
1. Interrupt Latency without Background Computing
2. Interrupt Latency without Background Computing
3. Context Switching Overheads

Interrupt Latency without Background Computing

In this part, we see that since there are no background process running in addition to the single thread the Interrupt Latency is averaging out at 80ms in my code.

This has been implemented in the same way as explained in the HW-3 Assignment problem statement. In the code this has been implemented in Function calculate_interrupt_latency() in the first part of the function.

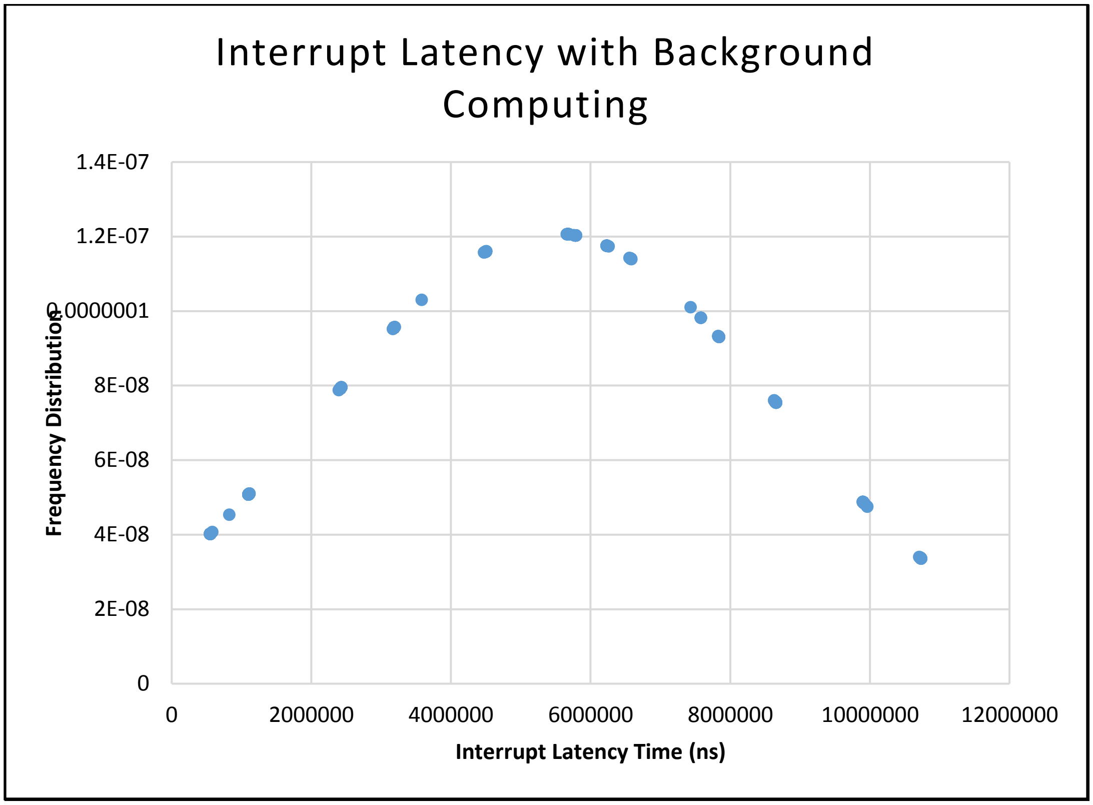




Interrupt Latency without Background Computing

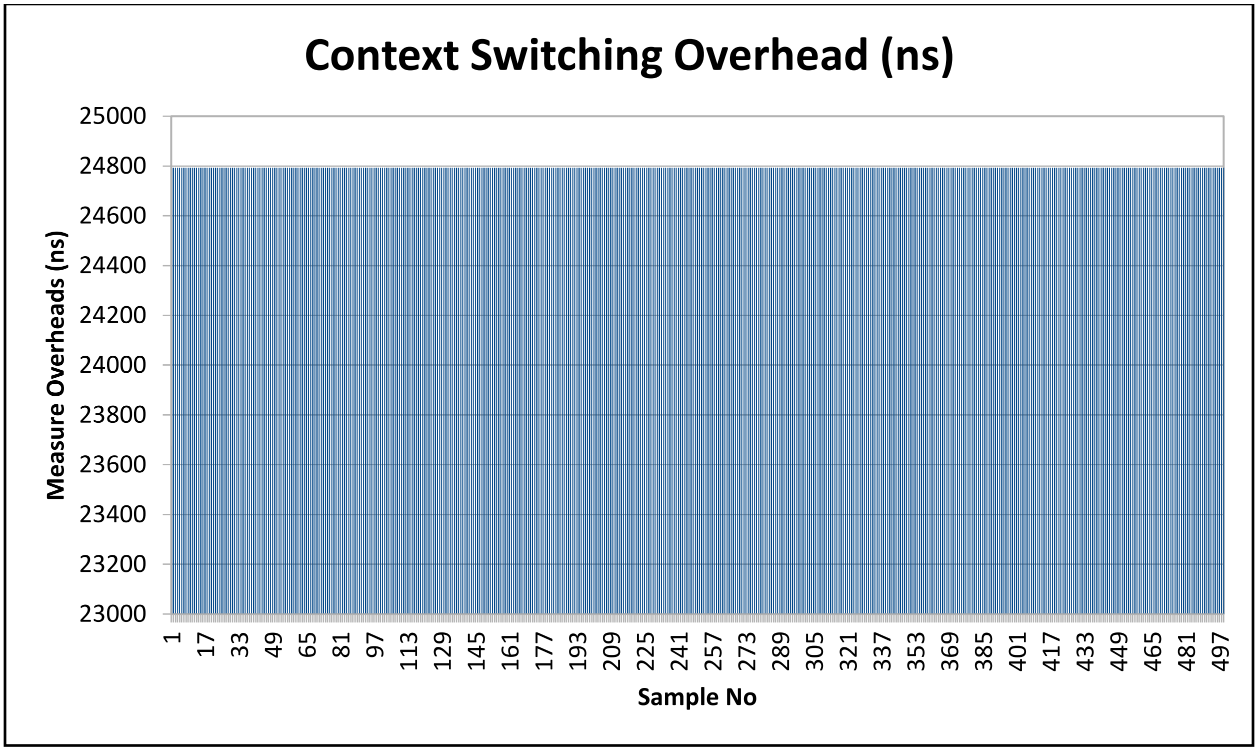
In order to implement this part, we have used two cooperative threads (sender & receiver functions) with equal priority sending message data between the threads using Message queue as defined at the start of the code.

Following is the frequency curve, where it shows Gaussian curve distribution representing the blocking of ISR by the cooperative threads operation.



Context Switching Overheads

Please note the graph below showing context switching. As we are synchronizing the context switching using mutex locks, we see almost same measurements in this. A thing to note is that we have reduced the overheads due to system calls mutex_lock() and mutex_unlock() for our measurements.


Context Switching Overhead (ns)
Measure Overheads (ns)
Sample No