Zak Rowland
CST331
Lab 1 – UART
Conclusions

a. Part one of this lab was completed without much trouble. The most difficult part was getting the UART initialized with all of the proper settings. Part two proved to be more difficult. The first issue I faced was getting the switch debounce to work properly. I tried writing the code in main() using counters which didn't work. I tried writing a separate debounce function that is called after a certain time has passed in main() which didn't work. Finally, I set up a timer to use without interrupts to control the debounce delay which works great. The other problem I faced was a strange runtime exception which was caused be overwriting shadow set settings when setting up the LED ISR. This taught me to be very careful when setting up control registers and try to combine settings into one line wherever possible.

b. The system clock used is 84MHz, PBCLK2 for the UART is 84MHz, and PBCLK3 for the LED timers is 84MHz.

c.

- Uncached latency: 94 cycles or 1.119us
- Uncached execution: 364 cycles or 4.333us
- Uncached latency was tested by making the main() routine trigger an artificial RX interrupt and measuring the time elapsed from the interrupt trigger to the first instruction of the routine.
- Uncached execution was tested by measuring the time elapsed from the first instruction of the routine to the next instruction in main(). I accomplished this by changing the receive functionality to "do-while" instead of only "while" so that it would perform a read once no matter what. I then cleared all instructions from main()'s infinite while loop except one dummy instruction in order to set the finish breakpoint.

d. i. The system performed at 114,130 baud.
ii. 1.119us + 4.333us = 5.452us = 183,418.929Hz
iii. The data needed won't necessarily always be in cache, so testing for the worst case scenario is a must.