Rose-Hulman Institute of Technology – Memorandum
October 25, 2016

To: Dr. Yoder, ECE497
From: Luke Kuza
Re: Homework 07

For homework 07 for ECE497, Embedded 32-bit Linux, the homework's goal was to write and demonstrate performance of different types of programs that copy signals on the Beaglebone Black board. The goal was to copy an input digital signal on one pin, and output it on another pin, and have as little delay as possible. There were four instructed ways to perform this task,

1. With a high level language (Java, Python, Javascript). In this implementation of the homework, Java was chosen using a modified version of the Bulldog Java API to access the BeagleBone I/O
2. With a C program using mmap to map memory and copy the signals
3. A C kernel driver running in the kernel
4. Using the onboard Texas Instruments Programmable Realtime Units (PRU)

For the first task, a very simple high level Java program was used to copy an input signal to an output signal as quickly as possible. Surprisingly, the Java program got a decent result of about 100 microseconds of delay. Java had medium CPU usage.

The second task was using mmap, and as expected, was much faster than Java. It was about 10-15 microseconds, almost 100x faster than Java. This had high CPU usage.

The third task was surprising. After verifying all the delays were removed from the kernel module template code, and ensuring it was optimized, the kernel module was found to be extremely slow. The signal was updated in *seconds* rather than mili or micro seconds. This was the most surprising out of the three results. It also had high CPU usage.

Finally, for the fourth task, the PRU was used to copy the signals. The PRU, as expected, had virtually no delay. While it was difficult to program for and setup, unlike Java, once the PRU is running, it has no additional CPU usage because it is running on its own processor. The delay is almost 0 because of this.

Scope outputs were taken for each task. Channel 1 is the input signal, and channel 2 is the output signal. The images are found in the ScopeTestImages folder in the hw07 folder and are properly named. In addition, here is a formatted table of the results.

Program Type	Signal Copy Delay
Java	106 microseconds
mmap (C)	13 microseconds
Kernel Module (C)	Over 100 milliseconds
PRU (ASM and C)	Measured 0 delay, can assume delay on the nano scale.

Table 1: Results of testing different program types and their resulting delays.

Overall, the experiment went as expected except the kernel module was a lot slower than expected. The Java high level program was a lot faster than expected as well. The PRU's incredible speed and ability to offload the CPU because it is a dedicated unit is very beneficial, and running this experiment allows for more options for the final projects at the end of the quarter. It also shows the benefits of dedicated a component to a specialized task, rather than having the CPU perform all tasks.