Chris Gordon
Stephen Fox

Project 4

For project 4, we implemented devices using a bus as well as a stopwatch assembly application. This was accomplished by only allowing one device to drive the bus at any given time, including the data memory and the processor itself. The processor drives the bus if memWrEn is active, meaning it is driving some data that will be written to a device. Memory drives the bus if memWrEn is not active and the address is in its address space. All the I/O devices have addresses that start with xF and only drive the bus if the address matches one of the devices' registers and memWrEn is not active. One tricky part of the project was finding a way to debounce the switches so as to only update the register when its state was stable. This was done by using a clock divider to get a tick every millisecond. If SW is different than the switch register it counts up to 10, and if it remains stable until then, the switch register is updated. Another tricky part of implementing devices this way was finding a way to allow certain values to be written to synchronously as well change depending on the state of other values, such as the ready and overrun bits of the timer device. This was overcome by keeping track of the past state of those other values to be able to if they had changed between clock ticks. On this project, I worked on the processor while Stephen implemented the stopwatch application.