

# 2160 Lab 89 Report

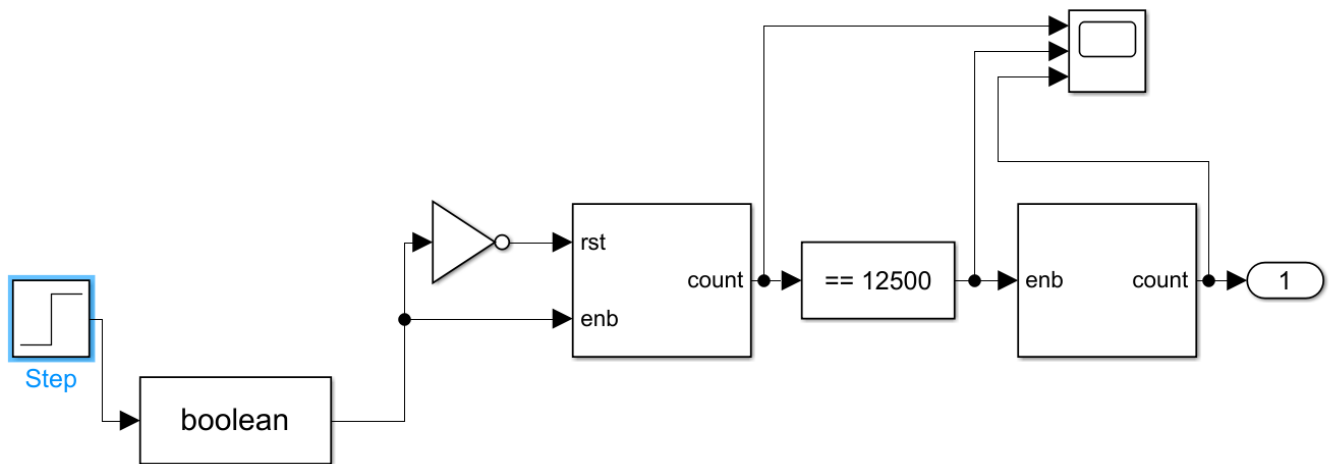
---

Patrick Taylor and James Wachala

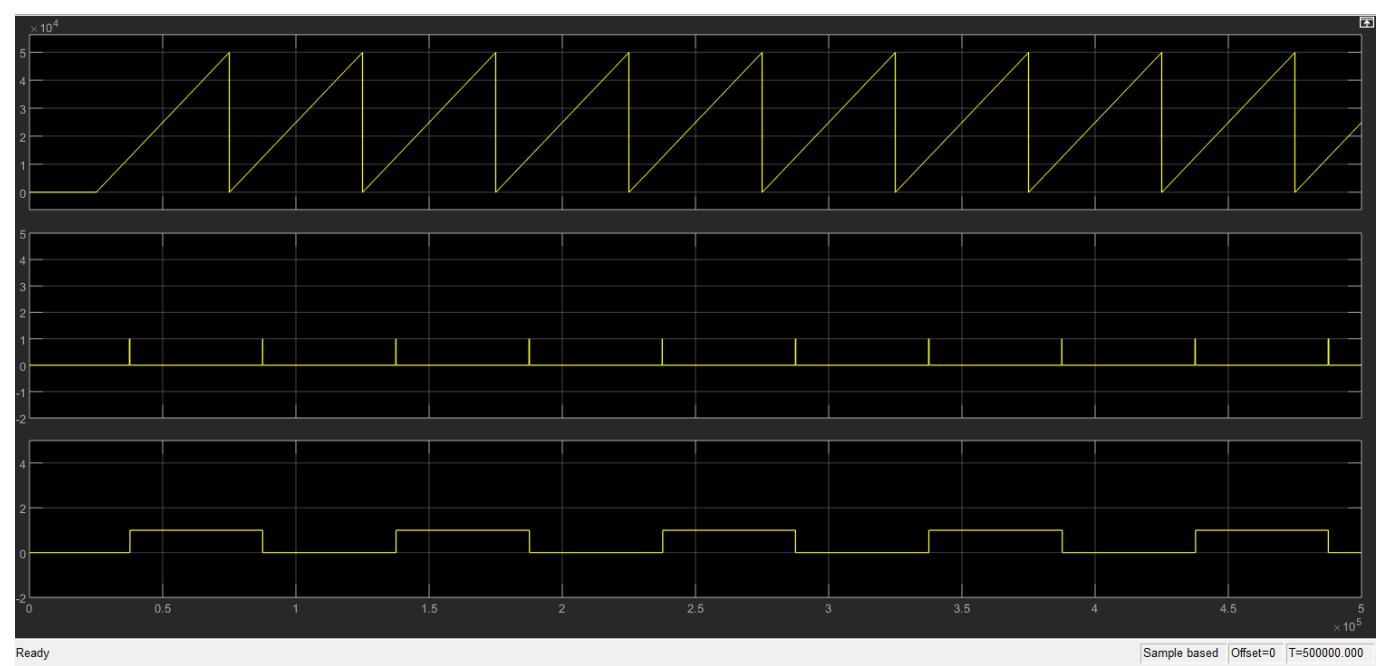
Section Number: 1

March 207th, 2019

## Prelab



The value for the comparator was chosen so that it only pulses once the button has been held down for 250ms straight. For the 50kHz test clock, this value is 12,500 as  $12500/50000 = .25$ . A value of 50000 was chosen for the first counter's upper limit so that it would reset every second. The other counter runs free, as it only toggles between 0 and 1 based upon the pulses from the comparator, which if the button is held down, is every second, as the first counter resets every second.

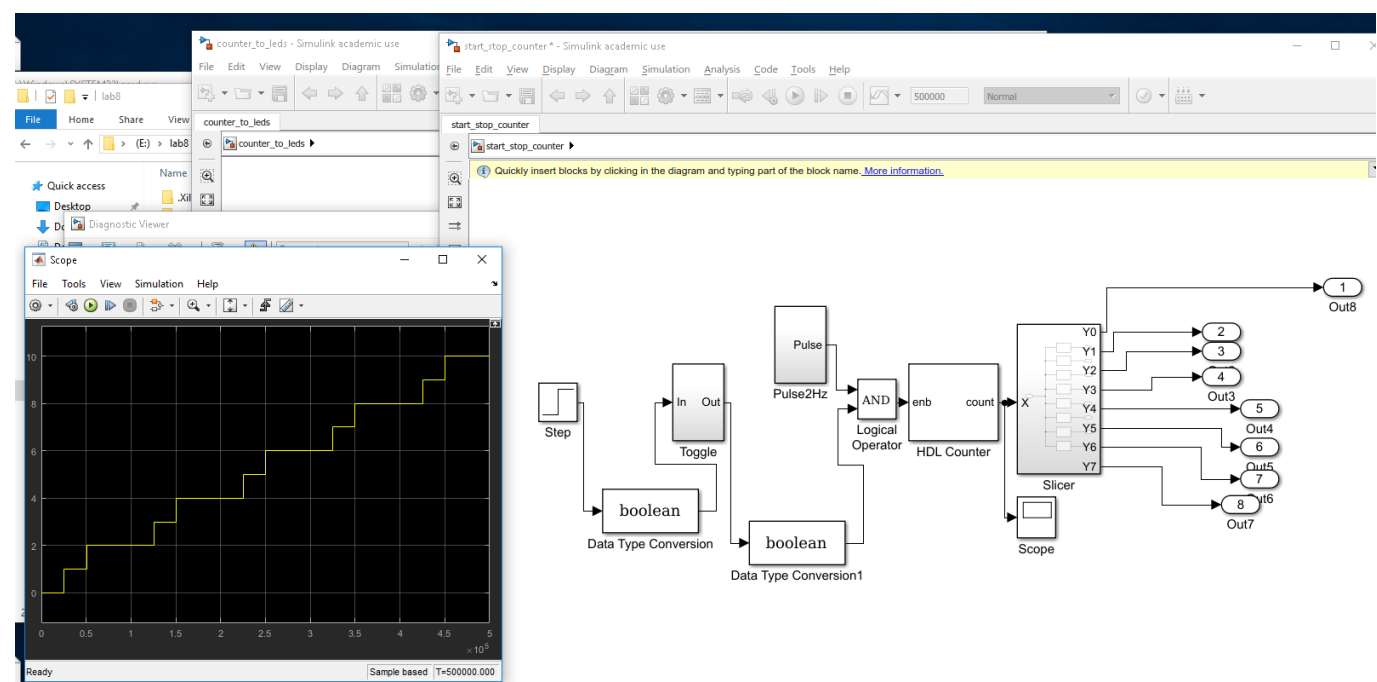


The scope provided here shows the intended behavior. The setup used to produce these graphs is simulating holding the button down. Notes that .5 on the x axis is one second on the 50kHz clock. So, it begins being held down at .5 seconds. Note that this is where graph 1 starts ticking up. .25 seconds later, the counter pulses on graph 2 (fulfilling the requirement for the button to be on for 250ms before starting). This pulse causes the output (graph 3) to turn to 1. 1 second later, the graph falls back down to zero, behaving as intened. It then cycles as intened after.

Assignment 1

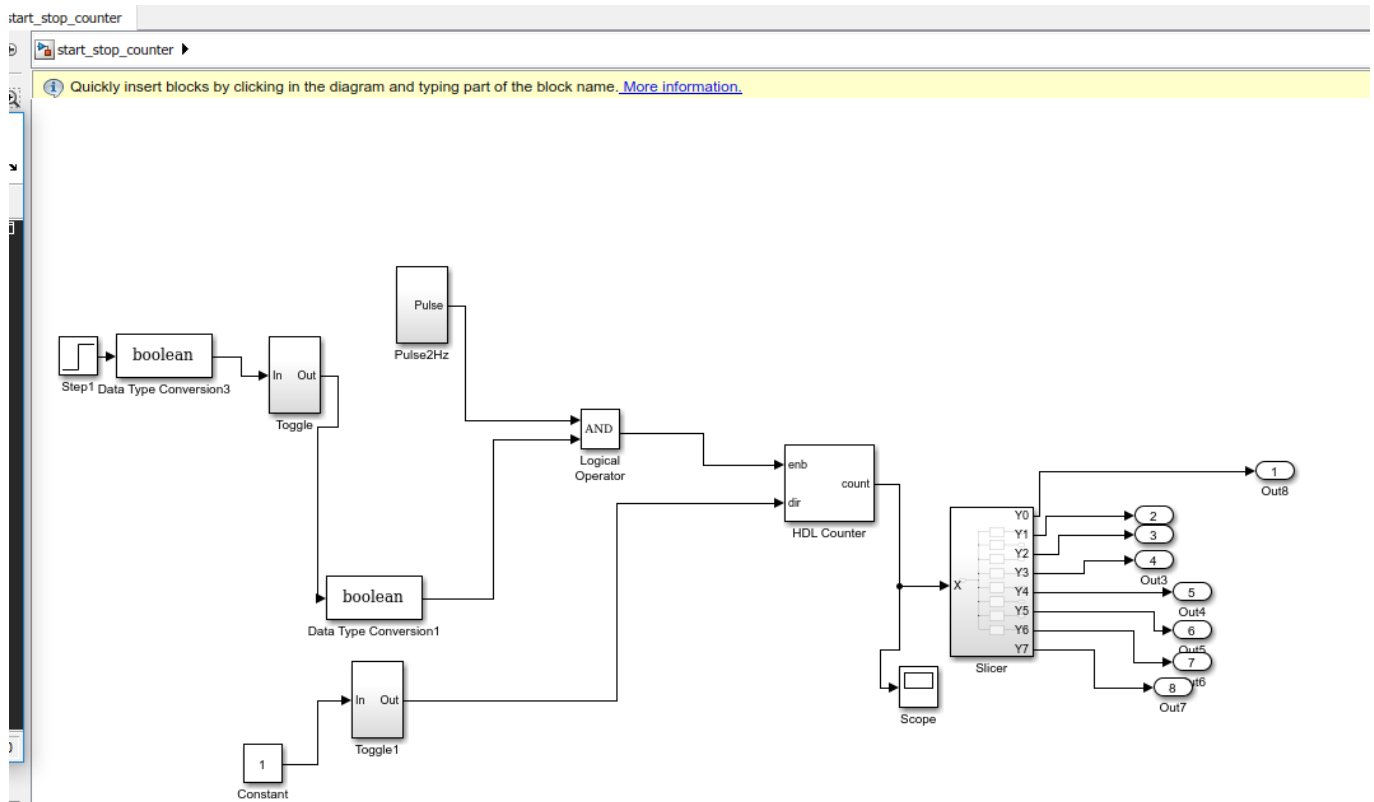
The values here were chosen because they are 1000 time greater than the ones in the test simulation, showing the shift from a 50kHz clock to a 50MHz clock, which is 1000 times faster.

Assignment 2

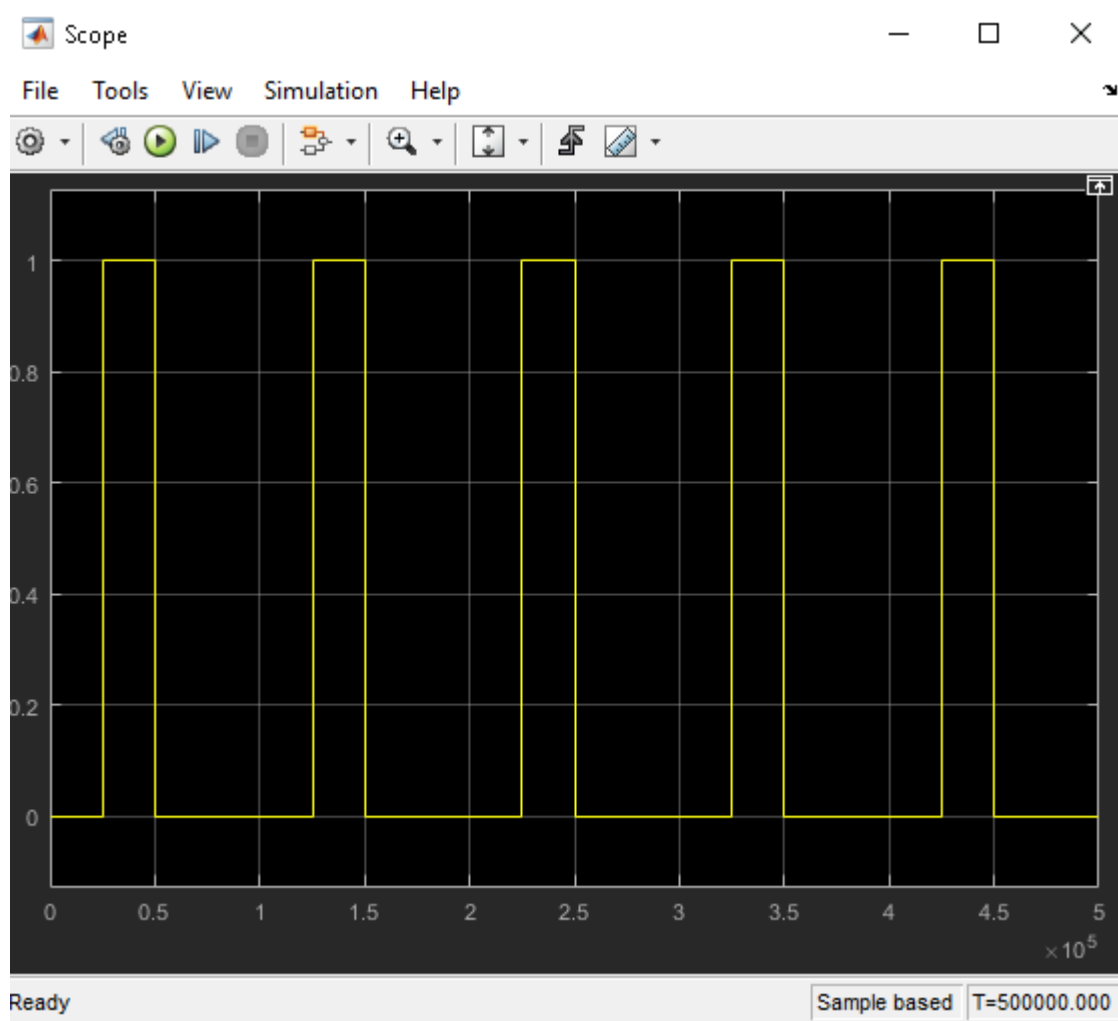


The Output cycles every 2 seconds. Each cycle it waits for a second after the toggle triggers and increments twice in the following second following that delay. Our design is correct as the Toggle triggers every other second (starting the cycle). The Pulse2Hz is triggered after Toggle triggers which increments the output twice.

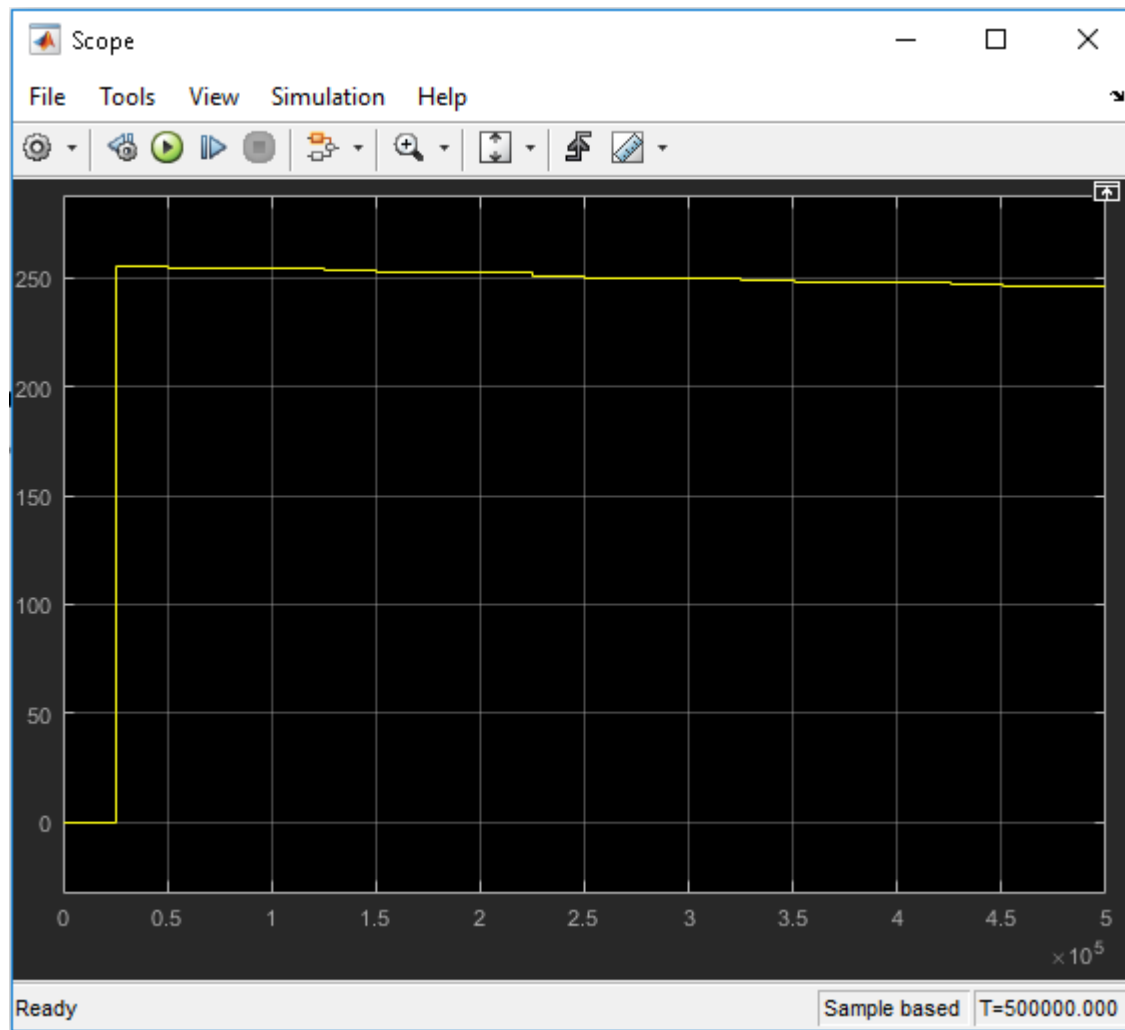
## Assignment 3



Step 1 activates after 25,000 iterations (half a second), which causes the Pulse2Hz to activate and feed into the counter to start counting, adding 1 every half second. The Constant determines whether the toggle is one or of. If it is off the counter decrements, if it is 1, the counter increments and decrements as the toggle block switches on and off. If the constant were a button and the toggle gets toggled on by briefly pressing the button the counter would increment only. The slicer then cuts the 8 bit number (ranging from 0-255) and feeds the 1-bit number to the LEDs (once the program is passed to the Zedboard)



The direction is being toggled between 1 and zero. When the direction is being fed a 1, it increments one then decrements as if gets toggled to 0.



The output of the counter being told to decrement as Toggle1 is feeding a constant value of 0 to the direction. And underflowing as the initial value is 0, it loops around to 255 then decrements.