Lab Assignment 7

Jack Fenton
fenton.j@husky.neu.edu
Nicholas Abadir
abadir.n@husky.neu.edu

Submit date: 3/29/2020
Due Date: 3/29/2020

Abstract

In this lab we continued using Simulink and learning it more in depthly. In this lab we created a simulation that would have been able to turn on an LED for one second and be off for one second on the ZEDboard. However we were unable to test it so a picture of the simulation will be present in the lab. The design would have taken a button input by holding the button down and then go through a two second cycle.

Introduction

The goal of this lab was to become more comfortable using Simulink and counters. Although the lab was originally intended to control the LEDs on the ZEDboard, the simulations below will show the functionality of each logical circuit. This logical circuit, while very basic, allows us to see how to create oscillations for machines, even without the zedboard, such as a robot arm that simply needs to move up and down depending on the input given.

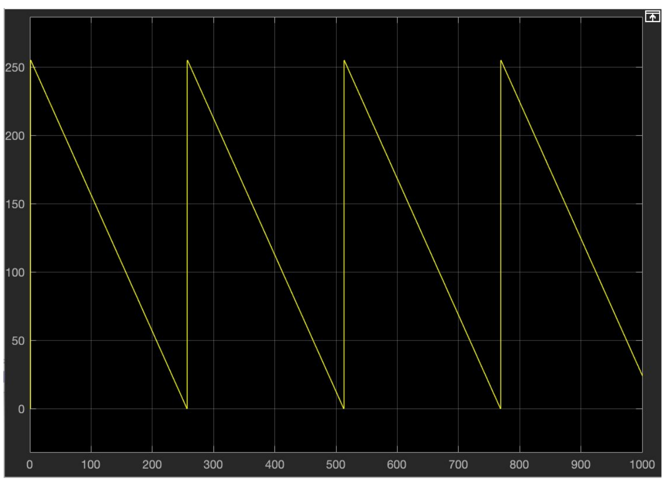
Lab Discussion

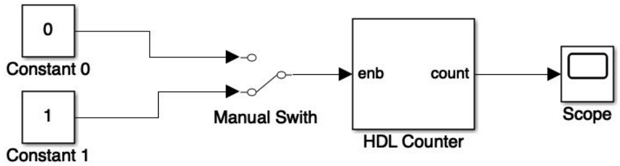
In this lab we used Simulink and used the slicer from the previous lab along with the HDL counter block to construct this oscillating logic circuit. We also used comparative blocks to control when the output was true or false.

This lab used less components than previous due to extenuating circumstances so the only components we used were our computers and Simulink through MATLAB.

Results and Analysis

Lab 7.2 Counters

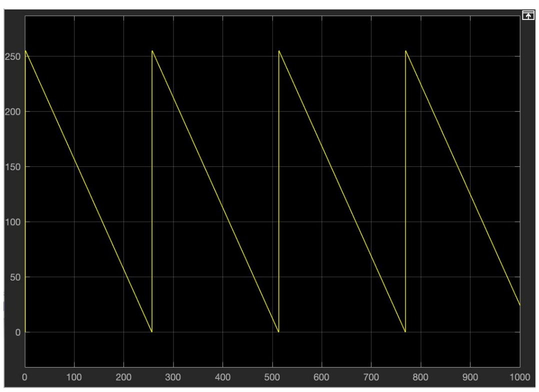


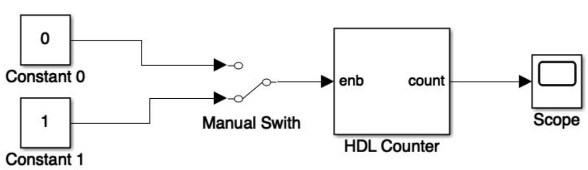


rst: The reset port is used to force the current tally in the counter to drop to the initial value.
load: When the load port is on, the value held in the counter is set to the value in port load_val
load_val: When the load port is in, this value becomes the value held in the counter
enb: The value in the counter only increments if this value is 1
dir: If this value is 1, the count increases by step value. If value is 0, this value decreases

The initial value will be the number held within the counter at the beginning of the simulation and each time the reset is triggered. The step value is how much the count changes by each time the counter is enabled, the direction of the change is determined by dir. Count to value is the maximum number the counter will hold, if greater, resets. The word length is the number of bits allocated to the number held by the counter, this caps the maximum count to value.

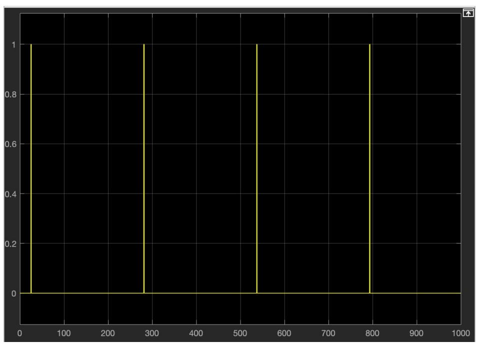
Lab 7.3 Design of a free-running counter

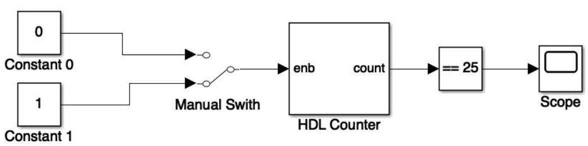




The figure above shows when the switch is set to constant 1, and when switch is set to constant the counter does nothing as the enable = 0 which means off. This makes sense as logical 1 is true and logical 0 is false. The enable requires type boolean or else en error message pops up preventing the simulation.

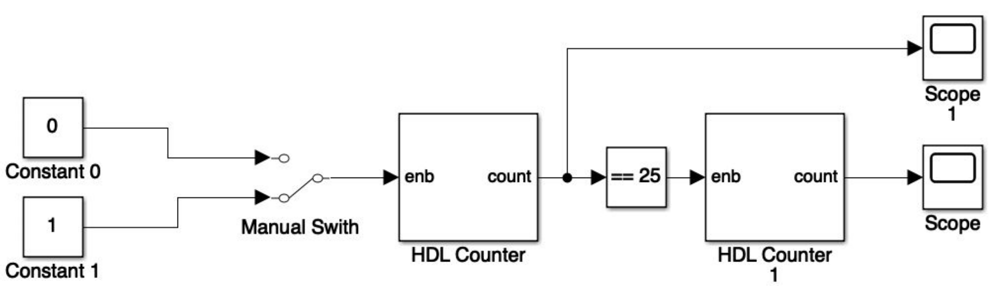
Lab 7.4 Design of a counter with comparator

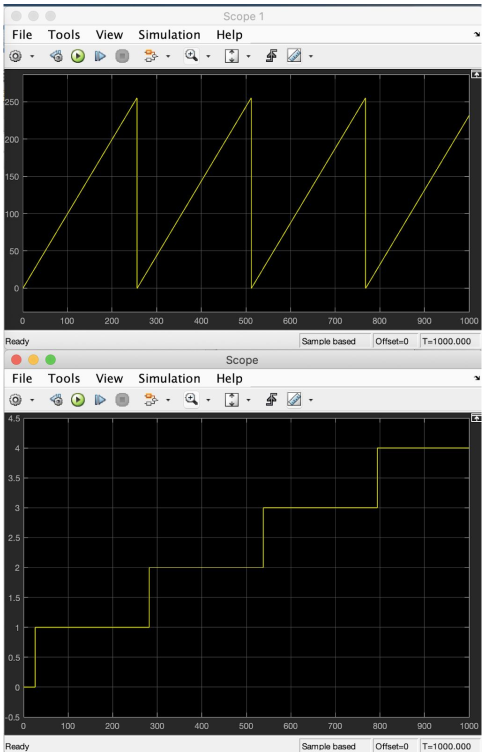




The scope makes sense as the counter goes through four iterations as seen above in 7.3 or 100/255 will hit 25 a fourth time. The scope outputs a one though when the counter hits 25 and the compare to constant becomes true outputting 1.

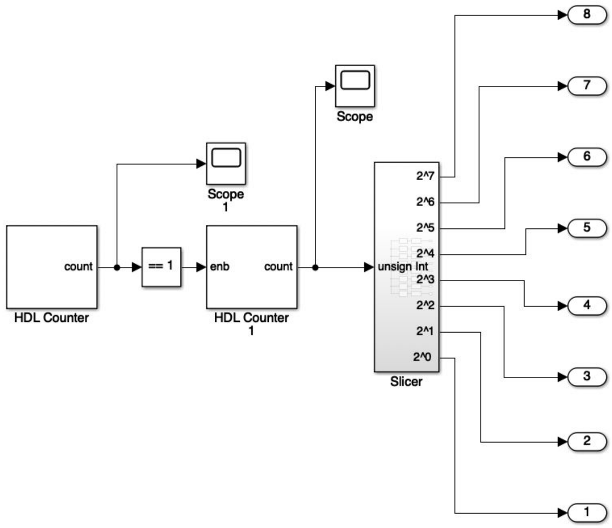
Lab 7.5 Design of a cascaded counter





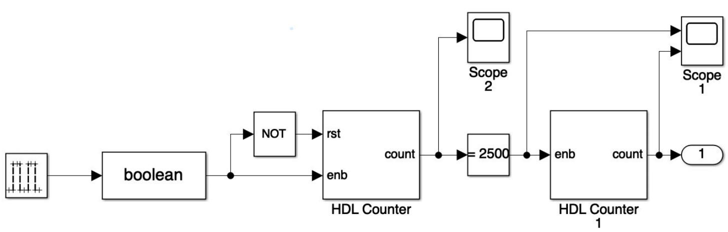
The 1st scope above represents the first counter and the 2nd scope represents the counter after the compare to constant block. As seen the first one simply counts to 255 over and over where the seconds is on when the first counter hits 25 and then turns off immediately afterwards. However it keeps its previous count number allowing us to track how many times the previous counter was equal to the constant 25. The speed of the second counter can be controlled by changing the number the first counter counts to. If higher than 255 the rate would, if lower than the rate would increase. You would do this by changing the word length of the first counter.

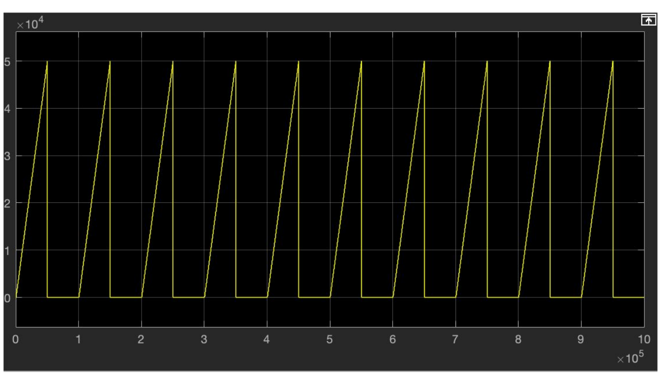
Lab 7.6 Controlling the LEDs with a cascaded counter

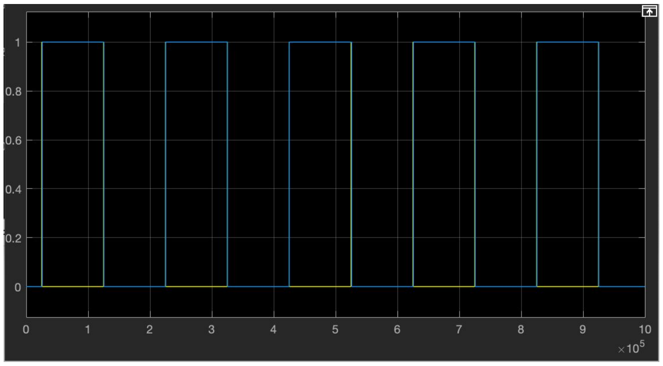


The design above is identical to the 7.5 however, the slicer has been appended to the output and so the second counter will now output the count number of how many times the first counter was equal to 1 in binary.

Lab 7.7 Designing a de-bounce circuit to manage the Push Buttons

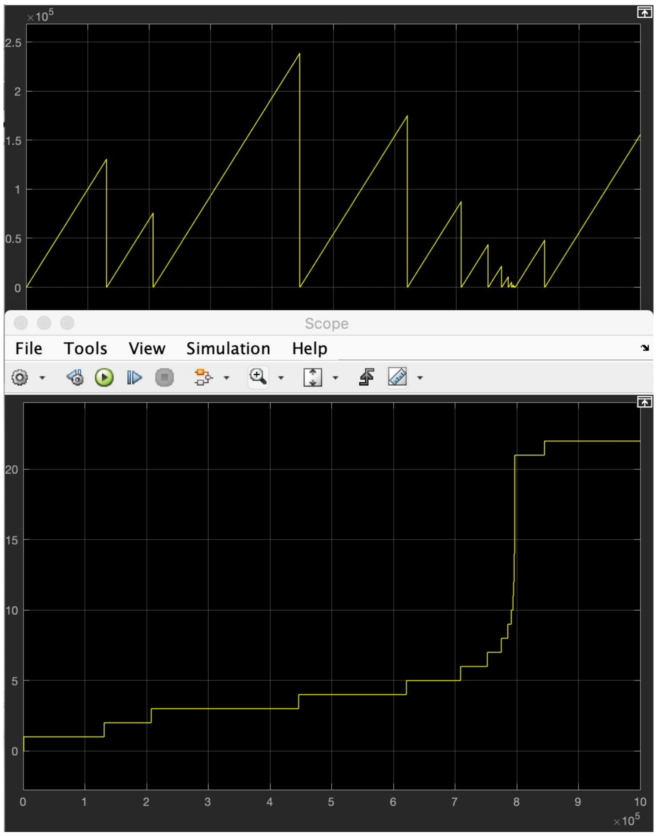


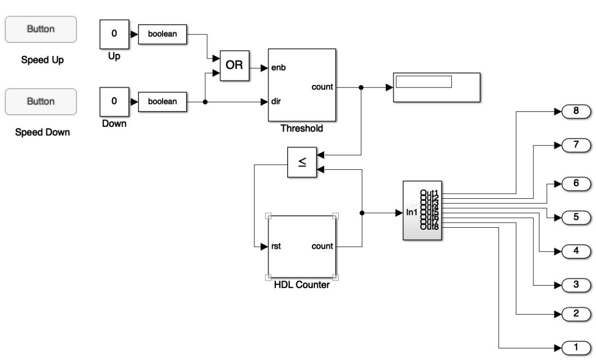




While the instructions of the lab said to use the step block, this was not possible as the step block increased to one immediately and then maintained the state, only simulating one button press and so there was no way to track the data when the button state was off. We switched the step block for a pulse block which allowed us to control the 0.25s delay between when the button was pressed and the actual running of the circuit by setting the compare to constant block to 2500. This is due to the counter running at 10 MHz instead of the 50 MHz as said in the lab. The word limit was then set accordingly to 12 as the maximum count was 4096 which includes 2500. The counter resets when the pulse is zero so as to be able to control the timing exactly. The graph above shows the counter counting up and then resetting when hits a certain value, however incorrectly shows at 5. The second graph shows how long the compare to constant is true as well as the output of the second counter. Unfortunately most of the scopes couldn't be combined as the scaling of the first graph was too large and made the other graphs seem insignificant as $10^4 > 1$ by a lot.

Lab 7.8 Change Counting Speed





The speed is controlled by putting a cap on the maximum number counted on the HDL Counter. This is controlled by comparing the current number held by the counter with a threshold value, if the counter had become greater than the threshold, then the reset port is triggered and the count begins again. The lower the threshold is, the faster the count increases. The value held in this threshold is set to a default value, but it can be changed by the button inputs. If the Speed Up button is pressed, the enb port is on and dir is off, so count and threshold continually decrease while the button is pressed. When the Speed Down button is pressed, the enb port and dir port are on, so the count and threshold continually increase while the button is pressed. The count within HDL Counter can be seen on the 1st graph above from the Scopes, when the maximum height of the ramp decreases, the speed up button has been pressed, and when the maximum height of the ramp increases, the speed down button has been pressed.

Conclusion

This lab was a continuation with Simulink and the construction of more complex circuits. The step function in 7.7 needed to be replaced with a pulse function as the step function only occurred once instantaneously and then was never altered. The pulse function however allowed us to simulate a person pushing a button in one second intervals. This gave us additional control to have the person push for ¾ of the time. This lab taught us how to control outside variables with logical circuits which directly correlate to actual circuits even though the units may be different, such as voltage versus bits and etc. This could allow us to further control response times of buttons and any sort of input devices.