**Lab 2 Report**

Jonathan Monreal (jem177) and Jiawei Wu (jxw585)

Group 8

**Design Questions**

1. ***What is the control signal frequency specification of the motor driver used on the daughter board?***

According to the L620x datasheet, the maximum frequency is 100 KHz, while the typical frequency is 30 KHz.

1. ***What is the chosen output frequency of your PWM generator?***

The output frequency of our PWM generator is ~25 KHz.

1. ***Assuming a maximum speed of 6000 rpm, what is the maximum motor speed your design will calculate with the provided 192 count/turn encoder attached to the motor?***

The maximum speed our design will calculate is ~3200 rpm.

1. ***How frequently does your design determine the motor's rate of rotation?***

The synchronous reset frequency is ~12 KHz.

1. ***Calculate approximately how long it takes to change the speed goal from 0 to full speed in one direction if the button is held constantly.***

It takes about 40 seconds for it to saturate at a maximum speed from zero.

1. ***At maximum gain, what is the pulse width when the goal is minimum and maximum for the free-running, enabled motor?***

The maximum pulse width is the input clock period of the pwm, which is ~0.04ms

The minimum pulse width is 0, since minimum goal is 0.

**Counters**

***goal-counter***

The goal counter is an 8-bit counter controlled by buttons 0-2 on the FPGA. Button 0 counts up, button 1 counts down, and button 2 resets the goal to zero. The counter module includes checks that prevent the counter from overflowing, such that the maximum value is 255. The direction is handled using a dedicated module.

***speed\_counter***

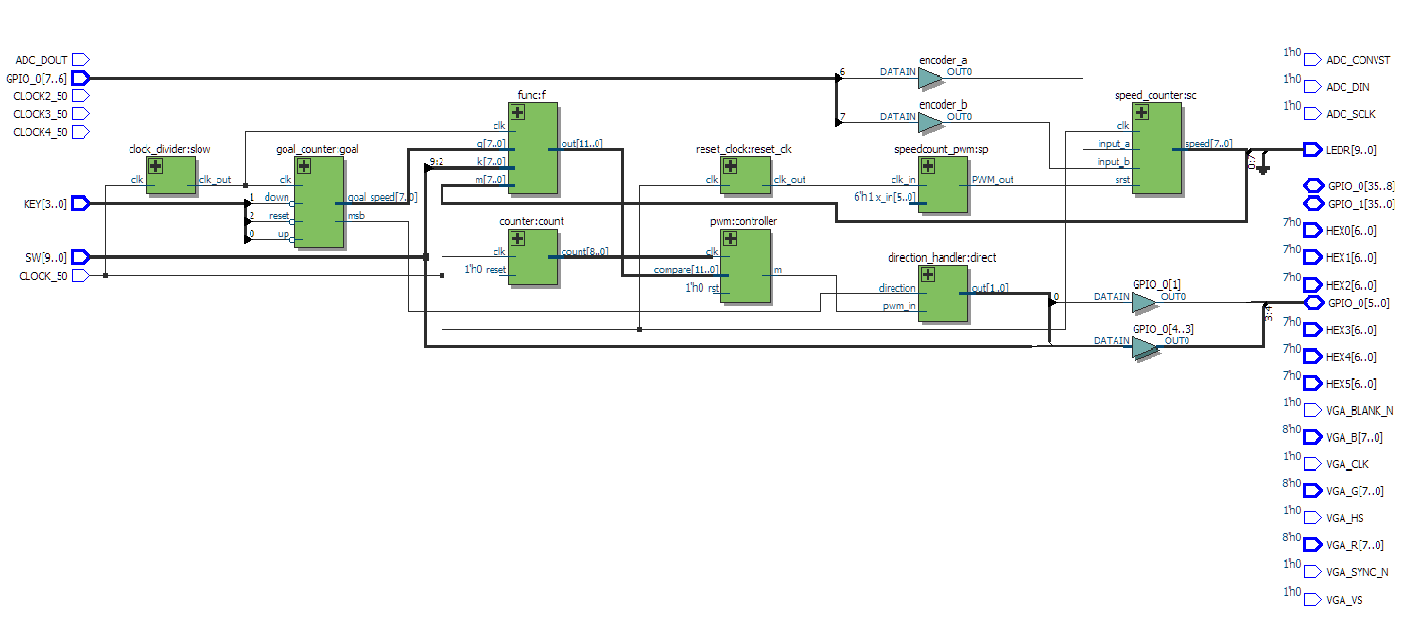
The speed counter is used to calculate a measure of speed based on the output of the rotary encoders on the motor. D-flipflops are used to capture the values of encoder\_a and encoder\_b at certain points in time, and converted into an enable equation for the counter. A synchronous reset is used where, when the reset is asserted, the value of the counter is captured in a register and the counter is reset, such that the value captured in the register is a measure of the speed of the motor.

***counter***

A simple counter used to divide the CLOCK\_50 frequency for use with the PWM module.

**Code Organization**

We followed the code organization so that the codes satisfy the requirements.

**Design Diagram and Module Functionality**

**Module clock\_divider** is a clock of low frequency that is associated with the buttons press trigger rate.

**Module goal\_counter** is the counter for goal speed from the button press, it generates a direction that is handled separately.

**Module func** is the implementation of the equation r=k(g-m) that we would use.

**Module pwm** determines how much duty cycle the motor would receive.

**Module direction\_handler** takes the direction bit from goal and determines which way the motor turns.

**Module speedcount\_pwm** is a pwm adjusting the reset/count percentage of the clock used to measure feedback speed.

**Module speed\_counter** is the motor speed feedback measurement.

**Oscilloscope Captures**

**Collaboration**

*Written by Jonathan Monreal*

The division of work was done with attention to not only the number of modules, but the relative difficulty of constructing each module. Work was shared on the top-level module, while I took on the task of working on the counters and Jiawei worked on the PWM modules.