Experiment 5
Motor Drive Control
ECE 482

The objectives of this experiment are:

- To gain experience with FPGA programming using Verilog
- To implement six-step (trapezoidal) motor control from an analog reference

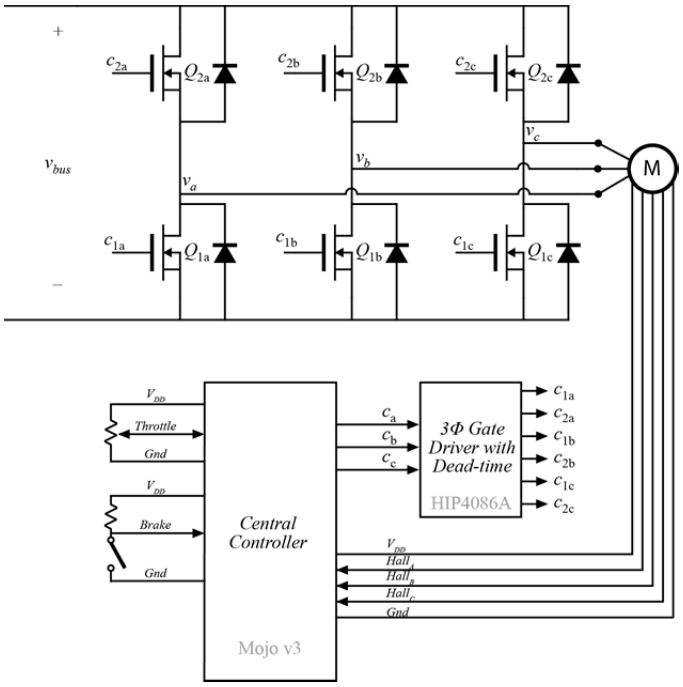

Figure 1: Motor drive power stage and control circuit schematic

In this lab, you will use Xilinx ISE Design Suite to program your central controller in order to implement an open-loop motor driver. You will use the PCB designed in Experiment 4 for the power stage.

For the central controller, we will be using the Mojo v3 FPGA development board. This board contains both a Xilinx Spartan 6 FPGA and Atmel ATmega32U4 microcontroller. The latter is used to program the FPGA on startup, handle USB communications, and provide analog-to-digital converters (ADC) accessible to the FPGA. Read through the basic tutorials on this development board here:

http://embeddedmicro.com/tutorials/mojo

Example files for the board provided in this course are written in Verilog using the Xilinx ISE.

I. Voltage-Controlled PWM

Download and run both "Pulse Width Modulation" and "Analog Inputs" projects from the tutorial website above. Once you are confident that you understand the operation of each, continue on.

Using the previous as a starting point, or starting from scratch, write code to generate a 20kHz PWM signal whose duty cycle is proportional to the signal supplied by the throttle in Figure 1. Limit the duty cycle so that at zero throttle, the duty cycle is 1% and at full throttle the duty cycle is 99%. Connect the throttle between the Mojo

3.3V supply and ground, with the output connected to one of the analog inputs, and output the PWM to a digital output so that you can view it on the oscilloscope. Verify that your code works as expected.

II. Reading Hall Sensors

The hub motor has a total of 8 electrical connections which may be made. The lower-gauge (larger diameter) wires comprise the three winding terminations of the motor. The higher-gauge wires connect internally to three digital-output hall sensors which may be used to determine rotor position. The hall sensors work by sensing the internal magnetic field generated by the rotating poles, with each of the three sensors outputting logic high when the magnetic field in its vicinity exceeds a predefined threshold. The three sensors are space 120° (electrically) apart. In this bundle of five wires, V_+ and V_- are 3.3/5V compatible supply inputs which power the hall sensors, whereas $H_{A\text{-}C}$ are the outputs of each sensor. Note that, as shown in Fig. 2, the hall effect sensor outputs are open-collector type outputs, so an external resistor R_p is necessary to convert each signal to an observable voltage waveform. This resistor should be larger than 1kΩ. The series resistance R_s protects the hall sensors, and should be (roughly) 33Ω

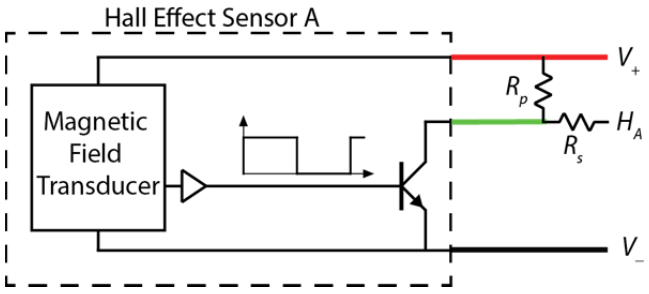

Figure 2: Motor hall sensor behavioral circuit model

Important Note: It is very easy to destroy the internal hall sensor BJT by applying a voltage at its collector. Before connecting *any* signal to the hall sensor outputs, make sure that the signal is a high impedance. For example, when reading hall sensor outputs onto an FPGA, you *must* connect a resistor in series and make sure that the FPGA pins connected are configured as *inputs*, not outputs.

The wiring of the hub motor and end connector are shown in Fig. 3. In order to determine the relationship between the hall sensors and winding phases, produce a time-aligned plot of the voltages V_A, V_B, and V_C and sensor outputs $H_{A\text{-}C}$. Divide your plot into six subintervals according to the state of each Hall output. For each subinterval, determine which single phase should have positive current, which should have negative current, and which should have zero current if the motor were to be controlled to generate a maximum positive torque. *Hint: you want power flowing to the motor*.

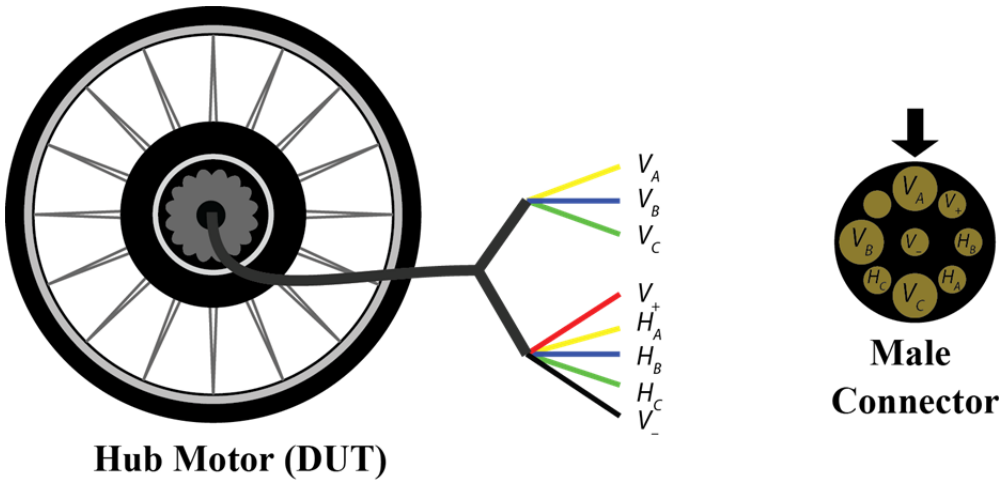

Figure 3: Motor cable and connector wiring diagram

III. Six-Step Trapezoidal PWM

Next, adjust your FPGA code to generate six output signals to control each of the six transistors in your motor drive. From the plot generated previously. That is, by connecting hall sensor inputs to digital inputs of the FPGA, generate six digital output signals which are logic high when the transistor should be "on", and logic low when it should be "off". Verify that, as you spin the wheel, these signals behave as expected and there is never an interval when both transistors in a single half bridge are turned on simultaneously. At this point, these signals should not be connected to anything other than the oscilloscope for measurement. As you spin the wheel, the six outputs should take on the form shown in Fig. 4(a).

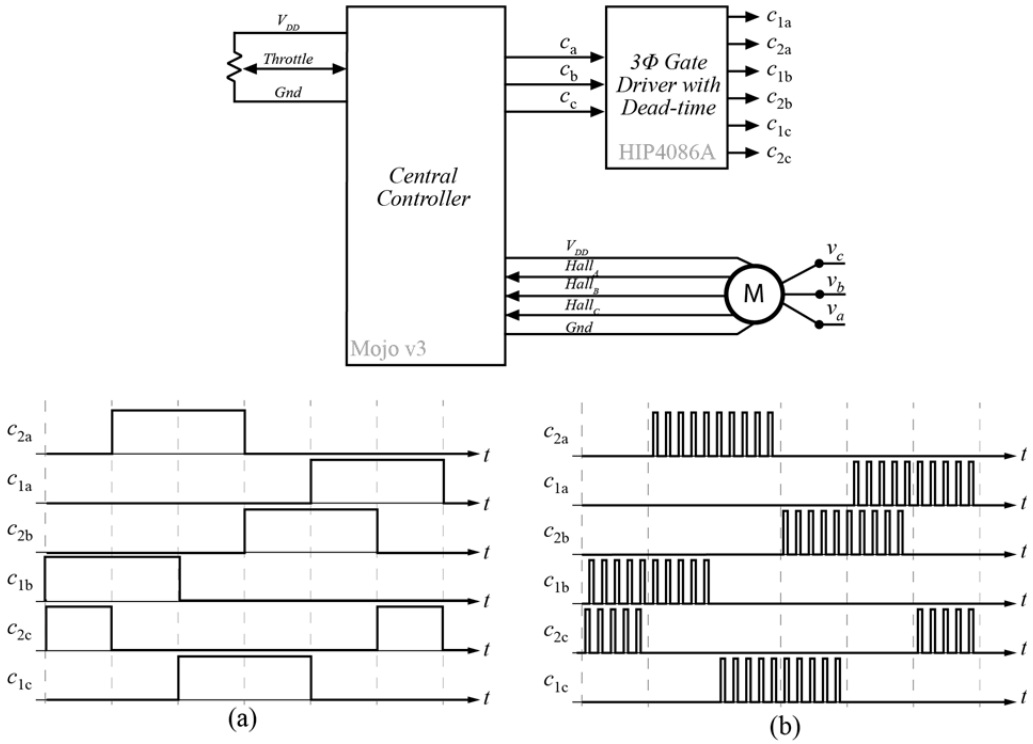

Figure 4: Traditional (a) and PWM (b) six-step transistor gate drive signals

Now, combine this code with that generated in Section I so that, instead of being "on", each of the six signals is pulse-width modulated with a duty cycle proportional to the current throttle during the interval dictated by the hall sensors, and (completely) off otherwise. As you spin the wheel, the six outputs should take on the form shown in Fig. 4(b), with the duty cycle determined by the throttle.

IV. Board Population and Testing

When your PCBs from Experiment 4 arrive, populate the motor drive stage. At this point, you do not need to populate any of the boost converter, but can use the benchtop power supplies to establish a voltage on V_{bus}. Using the same testing approaches developed so far in the course, populate one step at a time, testing as you go to ensure proper operation.

Once you have confirmed that your circuit layout is correct and the motor driver will function correctly, connect the gate drive signals generate in section III to the gate drive chip and, without any connection to the motor, confirm that the transistors are driven correctly.

Connect your motor drive and hub motor, with protection fuses in between. Leave the hub motor in the stand on the floor at all times (i.e. do not put the stand on the benchtop). Beginning with just 5V and a 1A current limit, and increasing steadily, test your motor drive by steadily applying throttle. Demonstrate to the instructor that you are able to control the speed of the hub motor using the throttle. Only test the hub motor at low speeds, as excessive speed will cause the motor to become unstable on the provided stands.

V. Speed Limiting of Motor

Adjust your FPGA code to limit the speed of the hub motor. One approach to this is to generate a low frequency clock on the FPGA, and implement a counter which measures the number of periods of this clock which occur between consecutive hall sensor edges. Knowing the number of poles and the radius of the wheel from lab one, you can estimate the speed that the bicycle will be traveling from this information. There are many other valid approaches to this task.

Demonstrate the ability to limit speed of the wheel to a rotational frequency corresponding to just 5 mph. In your final system, you can increase this limit to the legal limit of 20 mph.

VI. Deliverables

Submit the following to receive credit for this experiment:

1. Your final central controller code (as a single .zip of the project)
2. The diagram produced in Section II
3. A one-page (max) writeup detailing any problems faced in the experiment, how you solved them, and any modifications you would like to make to the system in retrospect.