What a Drag

Or Christmas in May

In a drag race two cars (or trucks or motorcycles, etc.) race down a quarter mile strip; whoever crosses the finish line first wins. Winners and losers are sometimes separated by as little as a millisecond. Therefore starting the race is important and an elaborate system has been developed.

There are three components to the starting system:

1. Two light beams and photocells across the track and separated by about 7 or 8 inches. The first one is called the "Pre-Stage" beam and the second one is called the "Stage" beam. These light beams sense the position of the front tires of the racers.
2. A stack of lights called a Christmas Tree as shown below in Figure 1; and
3. The electronics that manages the lights and keeps track of the times.

Figure 1. Drag Racing Christmas Tree

Here's how the system works.

PRE-STAGE

As you nudge your racer forward, your front tire will first interrupt the Pre-Stage beam. This will cause the top two (side-by-side) yellow lights on your side of the Christmas tree to light. Pre-Stage serves as a warning that you are getting close to the Stage beam.

STAGE

If you move your racer forward another 7 or 8 inches, you will cross the Stage beam. In turn, this will light the next two yellow lights at the top of the Christmas tree. Note that the light beams are an inch or so off the ground and at that height typical racing tires are 11 to 16 inches wide, so blocking both the Pre-Stage beam and the Stage beam is normal. The image above shows exactly this condition (both racers are Pre-Staged and Staged). A Staged racer is presumed to be ready to race.

Amber Lights

The next three lights are amber and they will start lighting in sequence one second after both racers are Staged. The sequence goes light this:

- One second after both racers are Staged, the top amber light will go on.

- After 0.5 seconds the top amber light will go off and the middle amber light will go on.
- After 0.5 seconds, the middle amber light will go off and the bottom amber light will go on.
- After 0.5 seconds, the bottom amber light will go off and the green light will go on.

Green Light
This means GO!

Red Light
The dreaded red light comes on if you move out of the Stage beam before the green line comes on. It means you've already lost! Note that it's OK for your Pre-Stage lights to go off before you get the Green Light, just not your Stage lights.

The Problem
Your job is to design a circuit (the DRCTC module) using Verilog that implements a simplified drag strip Christmas tree. To simplify the circuit, we will ignore the second car. Our circuit inputs and outputs will look like this:

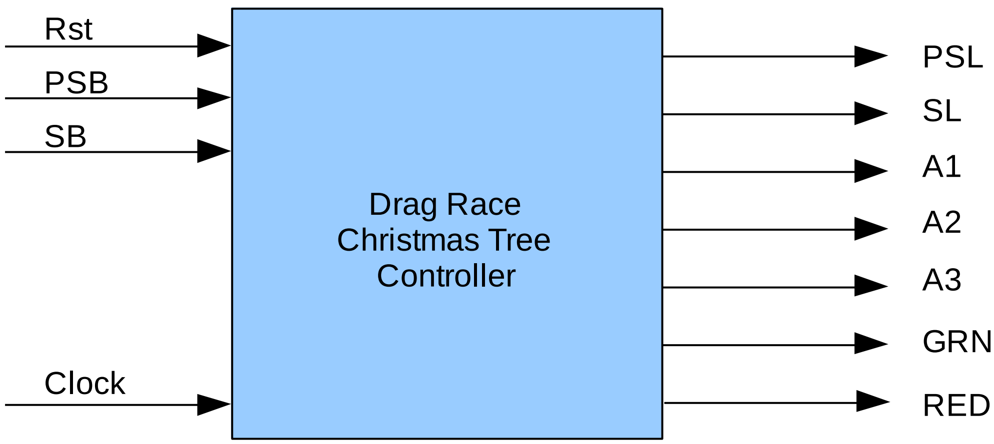

Figure 2. The DRCTC Module

Where
Inputs:

- Rst is a synchronous reset signal
- PSB is TRUE (1) if the Pre-Stage beam is broken
- SB is TRUE (1) if the Stage beam is broken
- Clock is a 50 MHz clock signal

Outputs:

- PSL is the Pre-Stage Light
- SL is the Stage Light
- A1 is the first amber light
- A2 is the second amber light
- A3 is the third amber light
- GRN is the green light
- RED is the red light

We will also keep the staging lights simple. Let

PSL = PSB
and
SL = SB

Use simple ‘assign’ statements for PSL and SL.

So now your job is to provide the logic for lights A1, A2, A3, GRN, and RED.

In the interest of uniformity, I’ll start the state transition diagram for you:

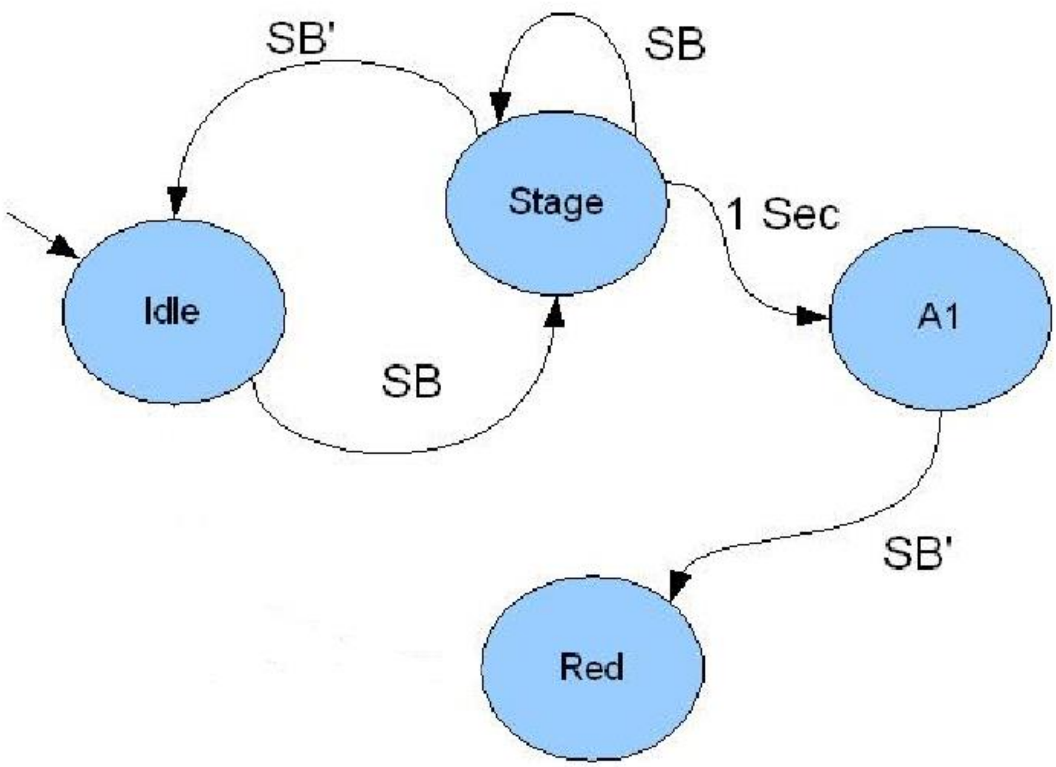

Figure 3. Beginning of the State Transition Diagram

Note that the Pre-Stage beam and light are not part of our state machine. For our purposes nothing interesting happens until the Stage beam is broken.

You need to:

- Finish drawing the state transition diagram. You don't need to include the reset (Rst) transitions.
- Set up a Quartus project called DragRace in a folder called DragRace.
- Implement a top-level module called DragRace that interfaces your DRCTC module to the DE2 board as follows:
    - Rst is derived from KEY[0]
    - PSB is SW[0]
    - SB is SW[1]
    - Clock is CLOCK_50
    - PSL is HEX7 (all segments)
    - SL is HEX6 (all segments)
    - A1 is HEX5 (all segments)
    - A2 is HEX4 (all segments)
    - A3 is HEX3 (all segments)
    - GRN is HEX2 (all segments)
    - RED is HEX1 and HEX0 (all segments)

- Implement the state machine in Verilog; use the state names suggested above;
- Implement the necessary timer module(s).
- Use my testbench (TestDragRace.v) to test your DRCTC module. Take a screen shot of the results. Note: It will take three minutes or so for this simulation to run. You can check simulation time in the lower left corner of the ModelSim GUI window. Your output should look similar to Figure 4.
- Upload your project to the DE2 board to verify its operation (fix any bugs)
- Add an interface to the GPIO pins in your top-level DragRace module as follows:

  ```
  module DragRace ( … GPIO );
  …
  output [13:1] GPIO;
  localparam Z = 1'b0;
  …
  assign GPIO = { R, Z, G, Z, A3, Z, A2, Z, A1, Z, SL, Z, PSL };
  ```

  You will be using the DE2-115 pins shown in Figure 5.
- Attach the LED circuit board to the DE2-115 as shown in Figure 6. **Don't get it turned around!** Make sure you get it lined up with pins 1-13.
- Take a photograph of your set-up with the LEDs attached to your DE2-115 board.

- Zip your entire project, including the ModelSim screenshot and the photograph from the step above, and turn it in on Canvas.

Here's a link to the Steve Oh video on You Tube.

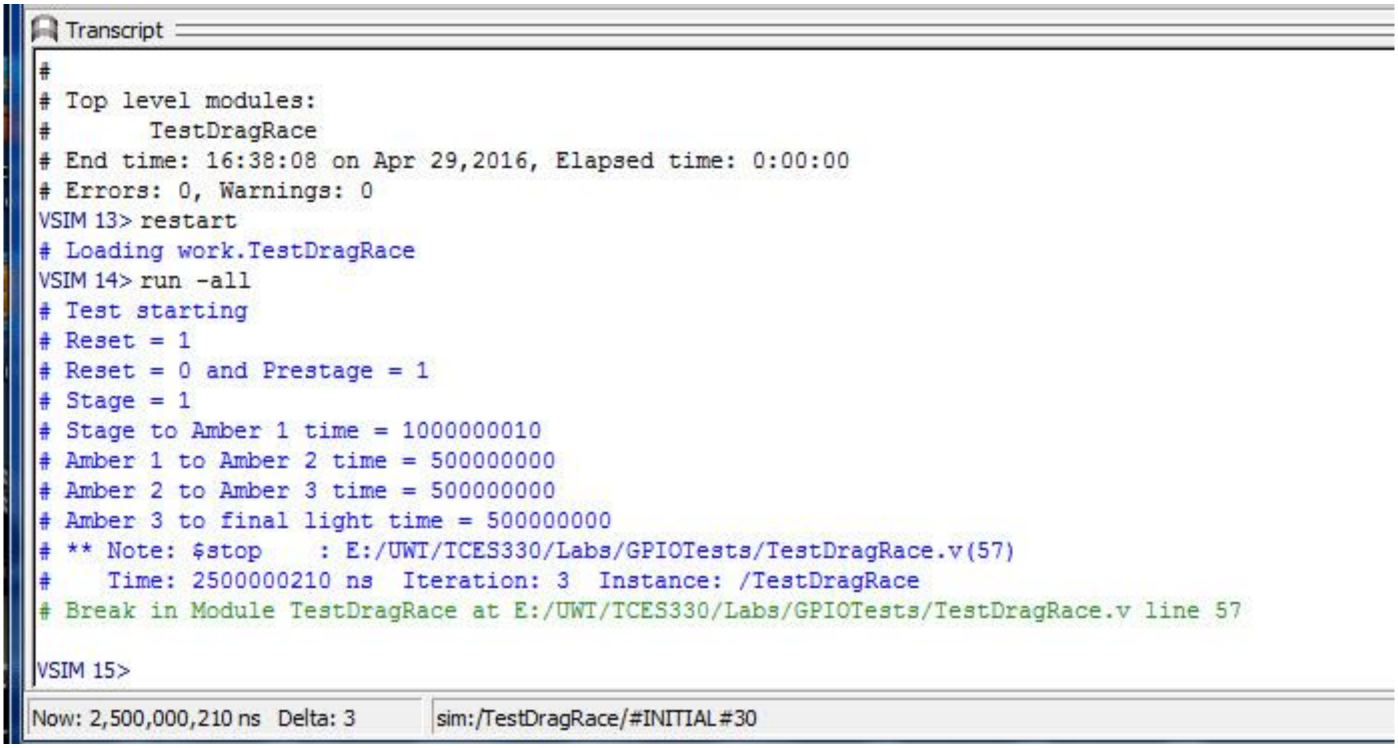

Figure 4. My ModelSim Testbench Output

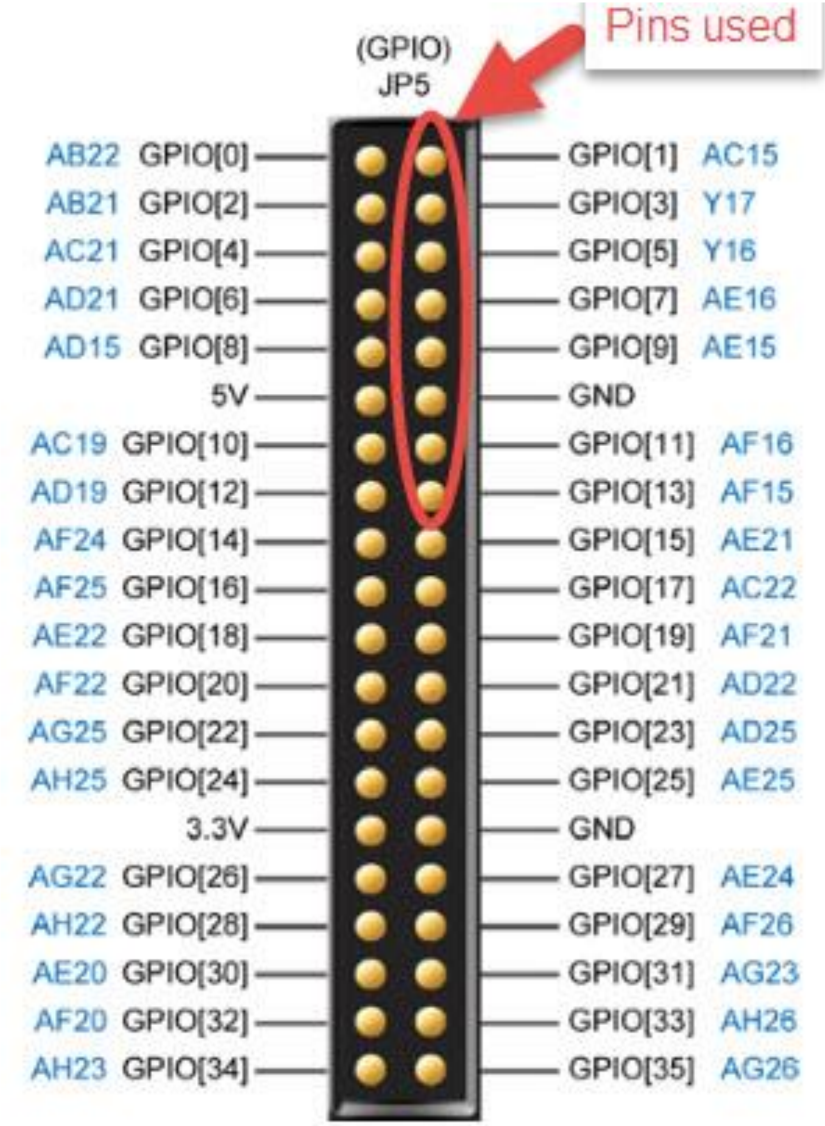

Figure 5. GPIO Pins Used

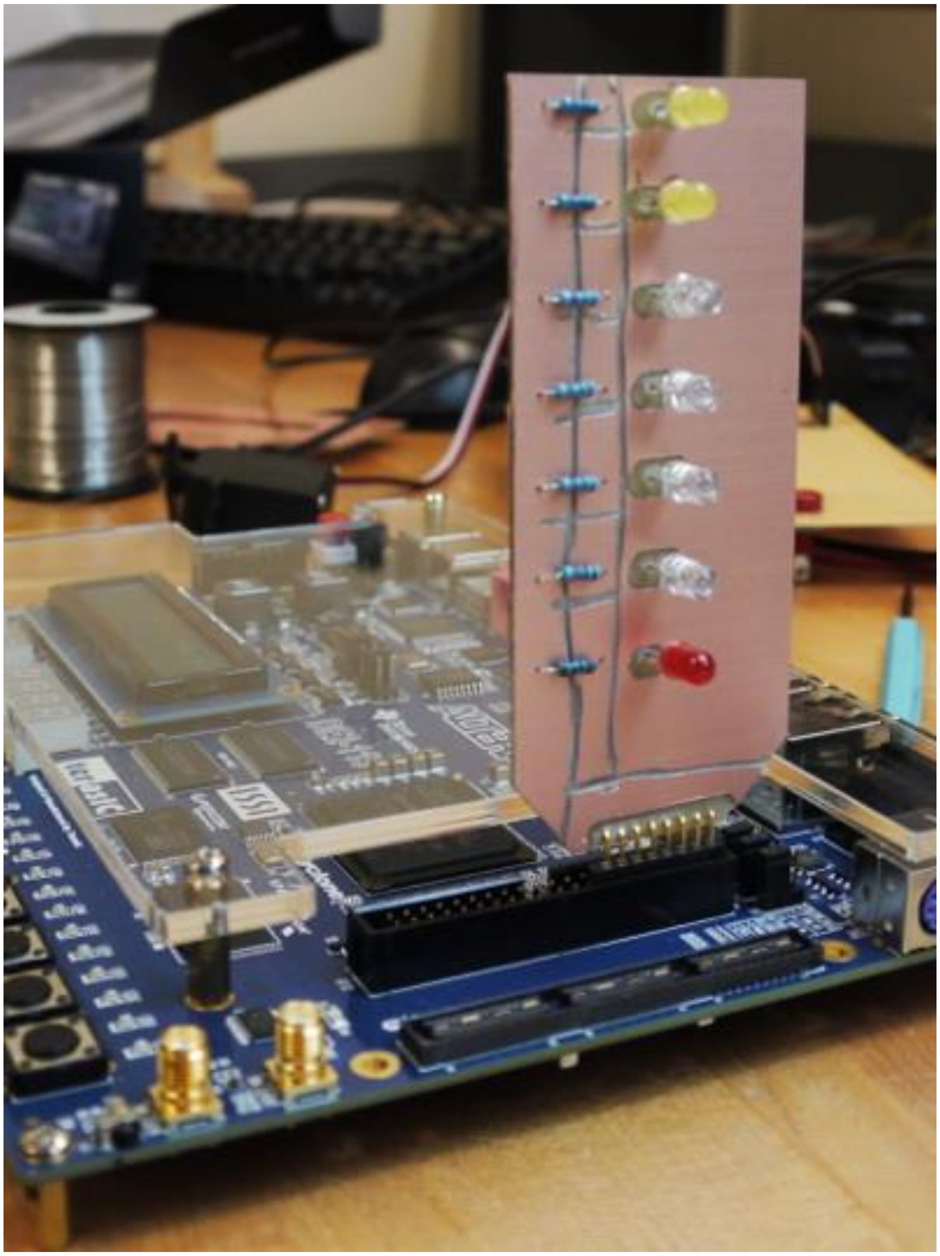

Figure 6. LEDs on the DE2-115