**Lab 3 Synthesis and Discussion**

**Synthesis:**

1. Binary State Encoding

=========================================================================

State | Encoding

=========================================================================

off | 000

l1 | 001

l2 | 010

l3 | 011

r1 | 100

r2 | 101

r3 | 110

haz | 111

-------------------

=========================================================================

1. High level units:

=========================================================================

Advanced HDL Synthesis Report

Macro Statistics

# FSMs : 1

# Counters : 1

32-bit up counter : 1

# Registers : 1

Flip-Flops : 1

========================================================================

1. Low level units:

=========================================================================

\* Final Report \*

=========================================================================

Design Statistics

# IOs : 11

Cell Usage :

# BELS : 138

# GND : 1

# INV : 2

# LUT1 : 31

# LUT2 : 1

# LUT2\_L : 1

# LUT3 : 4

# LUT4 : 21

# MUXCY : 39

# MUXF5 : 5

# VCC : 1

# XORCY : 32

# FlipFlops/Latches : 36

# FD : 3

# FDE : 1

# FDR : 32

# Clock Buffers : 2

# BUFG : 1

# BUFGP : 1

# IO Buffers : 10

# IBUF : 4

# OBUF : 6

**Device utilization summary:**

====================================

Selected Device : 3s100ecp132-4

Number of Slices: 30 out of 960 3%

Number of Slice Flip Flops: 36 out of 1920 1%

Number of 4 input LUTs: 60 out of 1920 3%

Number of IOs: 11

Number of bonded IOBs: 11 out of 83 13%

Number of GCLKs: 2 out of 24 8%

**Synthesis cont.:**

* Please see previous page for exact number of FPGA components. Overall, the final synthesis report used only a fraction of the total available resources to implement this design. The greatest number of resources used was the number of slices (30 out of 960). Everything else was a smaller fraction of usage.

**Discussion:**

1. Everything worked as planned, so there are failures to report
2. The only problems encountered in getting the design to work was when initially set up, the state machine was encoded such that when hazards were released, it would initially go back to the off state for one cycle regardless of whether a turn signal was asserted. This became immediately evident once the circuit was rigged up, and easily fixed with a line of VHDL code.
3. The only comments I would have about improving the lab would be to explain a bit more about the internal architecture of an FPGA. While it is fun to see our designs come to life, because the inner workings of an FPGA are still so abstracted there ended up not being much of a jump between test benching our VHDL code and the actual implementation. Other than that, the lab was thoroughly enjoyable.