## Guide for lab-4

**By Nilesh Modi**

- Since there is some confusion in the description of the priority conditions of inputs in Lab-4, only following conditions should be taken into consideration when you make state table and state diagram. This will reduce the confusion and also the size of your state table!

  **Note**: Students, who have already made a table with some more/different conditions than the ones mentioned here, do NOT need to change anything. This clarification is just for those people who still haven't started working on pre-lab.

- In any case, whichever state table you may have, we will grade you solely according to the consistency of your overall implementation with the state table, i.e. for checkout, we will only check for the combinations you have mentioned in your statetable.

  Different possibilities with inputs:

  1. Only LEFT turn signal is ON (Left turn sequence continues)
  2. Only RIGHT turn signal is ON(Right turn sequence continues)
  3. Both LEFT and RIGHT are simultaneously ON (the system goes in HAZARD)
  4. Only BRAKE is ON (The lights stays ON)
  5. Both LEFT and BRAKE are ON (Left lights should continue left turn sequence, while Right lights always stay at ON)
  6. Both RIGHT and BRAKE are ON (Left lights should always stay ON, while Right lights continue Right turn sequence)
  7. Only HAZARD is ON (Both left and right lights go ON and OFF in unison)
  8. Both LEFT and HAZARD are ON (Hazard overrides Left turn signal, both left and right lights go ON and OFF in unison)
  9. Both RIGHT and HAZARD are ON(Hazard overrides Right turn signal, both left and right lights go ON and OFF in unison)
  10. Both HAZARD and BRAKE are ON(Brake overrides Hazard, i.e. all turn signals should be ON)
  11. No signal is ON (All lights should go OFF)
  12. RESET is ON (All lights should go OFF)
  13. For LIGHTS, CLK and DIMCLK, you should follow the instructions in your lab-4 handout.

- After making the state table and the state diagram, you should write a verilog program simulating the above behavior. Then you should write a testbench to test your verilog program. The help for this is available on the lab datasheets webpage in the form of a verilog tutorial and a testbench file.

- On completion of the previous step, you should create an implementation constraints file, similar to the one you created in previous lab. You have to specify on which pins of FPGA board, you will be providing inputs and from which pins, you will take outputs, which will drive six LEDs.

- After that create a .bit file, with the same procedure as the previous lab. Download it to FPGA using XSLOAD. You can test your circuit by directly providing inputs to FPGA using DC supply. Please keep in mind, this step is only for your verification.

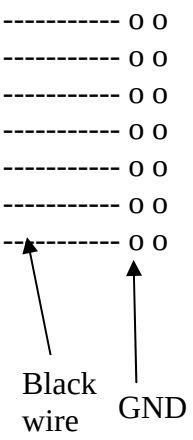
- Then, you have to write a C/C++ program that will provide clock and input controls to your FPGA board. I have attached a sample program with this mail. This should help you in creating your own program. Furthermore, there are two libraries attached with the mail, which should be added to the project (instructions on how to add are available under the document "Borland C Universal Library"). To help you setting up the libraries, I have attached both the required files "cbw.h" and "cbw32bc.lib" with this mail.

- Also, students who are using Borland C++ Builder for the first time, there is a two page starting guide on the lab datasheets webpage.

- Once you have your program ready with you, you should connect the flexible ribbon cable connector with the proper counter piece which fits in to this connector (you can get this from Electronic Shop). Also, before proceeding further, please read the document "Digital I/O Board" under datasheets section.

- After that you should mount your proper counter piece on to the breadboard, and then from there, you can connect the pins to the input pins of FPGA using normal wires. The experiment is same as the previous lab, instead here your inputs will be coming from a C/C++ program.

- The following sketch might help you locating the GND pin of the counter piece.




- Connect your circuit with the function generator and with LEDs.

## Call your TA to grade your hard work!!