ECE 270 Lab Verification / Evaluation Form

Experiment 6

Evaluation:

IMPORTANT! You must complete this experiment during your scheduled lab period. All work for this experiment must be demonstrated to and verified by your lab instructor *before the end* of your scheduled lab period.

STEP	DESCRIPTION	MAX	SCORE
Pre-Lab 1	Truth Table and Function Minimization	5	
Pre-Lab 2	Schematic	2	
1	Create Verilog Module	3	
2	Correctness of Coding Style (Dataflow/Behavioral)	3	
3	Implement Function using PLD	5	
4	Implement Function with Logic Gates	5	
5	Thought Questions	2	
	TOTAL	25	

Signature of Evaluator: _____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. ***You will not receive credit for this lab experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, I hereby certify that the work on this experiment is my own and that I have not copied the work of any other student (past or present) while completing this experiment. I understand that if I fail to honor this agreement, I will receive a score of ZERO for this experiment and be subject to possible disciplinary action."

Last Name (Printed): _____ **Lab Div:** _____ **Date:** _____

E-mail: __ __ __ __ __ __ __ __ **@**purdue.edu **Signature:** _____

Introduction to PLDs, Verilog, and ispLEVER™

Instructional Objectives:

- To learn how to create a VERILOG Module that specifies the design of a combinational logic circuit for implementation on a programmable logic device (PLD)
- To learn how to use ispLEVER™ Classic to synthesize a Verilog module and "fit" the design into a specific PLD
- To learn how to use a Universal Programmer to "burn" a fuse map (JEDEC file) produced by an HDL (hardware description language) compiler into a PLD
- To learn how to translate a word description into a logic circuit realization.

Prelab Preparation:

- Read this document in its entirety
- Complete the **Pre-lab Steps**

Lecture/Demonstration:

Your lab instructor will give a brief presentation that includes the following:

- A review of the steps for synthesizing Verilog modules and fitting it into a specific PLD
- A review of the internal structure of a 22V10 PLD
- A demonstration of how to program a PLD using the Universal Programmer
- A demonstration of the completed experiment

Experiment Description:

A distinguished (yet unnamed) instructor has asked you to design a circuit that calculates course grades, based on the letters in students' first names. The "key letters" chosen for this semester are "I", "R", "C", and "K". A course grade of "A" will be given to students who have the vowel ("I") and *all three* of the consonants ("R", "C", and "K") in their first name (e.g., "RICKY"). A course grade of "B" will be given to students who have the vowel ("I") and *either one or two* of the three consonants ("R", "C", or "K") in their first name (e.g., "KRISTEN" or "CINDY"). A course grade of "C" will be given to students who *don't* have the vowel "I" in their first name but do have *at least one* of the three consonants in their first name (e.g., "ROCKY" or "DAR"). A course grade of "F" will be given to *all other students* (i.e., those who do not receive a grade of "A", "B", or "C").

Pre-lab Step (1):

Create a truth table for the grading circuit (note that it has four inputs and four outputs), and map and minimize all four functions. For each of the four functions, compare the SP cost to the PS cost, and choose the one that is "cheapest" to implement. Show all your derivatory work on a separate sheet and have it available for your lab instructor to check. ***Keep track of the amount of time it takes you to complete this step.***

Pre-lab Step (2):

Draw a ***complete schematic*** for your circuit (i.e., one that includes switches, pull-up resistors, LEDs, etc.) using OrCAD, ***based on parts available*** in your DK-2 parts kit. Your solution will most likely utilize some combination or subset of: 74HC00, 74HC02, 74HC04, 74HC08, and 74HC10 (numerous solutions are possible). Print a copy of your schematic and have it available for your lab instructor to check.

Experiment Step (1):

Using ispLEVER, create a Verilog Module (*Dataflow Style* recommended) for implementing your grade assignment circuit. Target your design for a **GAL22V10C** device. Keep track of the amount of time it takes you to design the circuit using Verilog. Ask your lab instructor to check your completed Verilog module and the coding style before proceeding with the next step.

IMPORTANT: Do NOT put *spaces* in VERILOG project or file names.

Experiment Step (2):

Wire up your PLD-based grade assignment circuit, using a DIP switch for the four inputs and four **RED** (resistor) LEDs (connected in a ***current sourcing*** configuration) to the "A", "B" and "F" outputs. Keep track of the amount of time it takes you to wire and debug your circuit. Synthesize your Verilog module using ispLEVER™, fit it into a **GAL22V10C**, and use the Universal Programmer at your station to "burn" your design (JED file) into your ***own*** 22V10 PLD (different "22V10" variants may be included in the DK-2 Parts Kit – the current version is an Atmel **ATF22V10C-25PU**). Test and verify the operation of your circuit, and demonstrate it to your lab instructor.

CAUTION: Be sure to select the correct ***part number*** (e.g., **ATF22V10C-25**) on the Universal Programmer. (The suffix tells the programmer which algorithm to use; choosing the wrong algorithm can ***permanently damage*** your PLD.)

Experiment Step (3):

Wire up your "discrete" grade assignment circuit, using a DIP switch for the four inputs and four **RED** (resistor) LEDs (connected in a current sourcing configuration) to the "A", "B", "C" and "F" outputs. *Keep track of the amount of time it takes you to wire and debug your circuit.* Test and verify the operation of your circuit, and demonstrate it to your lab instructor.

Experiment Step (4):

Place your answers to the following "thought questions" in the space provided below:

(a) Compare the amount of time it took you to complete Pre-Lab Step (1) versus Experiment Step (1), expressed as a ratio (e.g., **2:1** or **1:3**). If there were 6 input variables instead of 4, in what way (if any) would you expect this ratio to change?

(b) Compare the amount of time it took you to complete (i.e., wire, test, and debug) Experiment Step (3) versus Experiment Step (2), expressed as a ratio. If there were 6 input variables instead of 4, in what way (if any) would you expect this ratio to change?

