Intro to Digital Logic, Lab 3
Digital Design using FPGAs

Lab Objectives

Now that you know how to develop Verilog designs and load them into the DE1 SoC board, we can now start looking at more complex designs.

Design Problem – Multi-level logic on the DE-1 FPGA.

Before going through the lab, you should review the Verilog tutorial on the website up through (but not including) Register Transfer Level (RTL) Code.

In order to speed up processing of returns at Nordstrom's, the customer service department wants an electronic detector device. Its goal is to both determine those items that have been discounted, so that the proper rebate can be calculated, as well as help find shoplifters returning their ill-gotten gains.

There are six products being sold. The UPC code for each is shown, as well as whether it was ever on sale (i.e. sold at a discounted price), and whether it is expensive, and thus is marked when sold.

Item Name	UPC Code	Discounted?	Expensive?
Shoes	0 0 0	No	Yes
Costume Jewelry	0 0 1	No	No
Christmas Ornament	0 1 1	Yes	No
Business Suit	1 0 0	No	Yes
Winter Coat	1 0 1	Yes	Yes
Socks	1 1 0	Yes	No

Note that since there are only 6 items, not all UPC codes are used. The behavior of your circuit for these situations is unimportant (i.e. Don't Care).

You will be given the UPC code of the item under test (signals "U", "P", and "C" for simplicity), and a detector will check for a secret mark ("M"). Your circuit should have one "discounted" light that lights up whenever a discounted item's UPC code is applied, as well as a "stolen" light that lights up whenever a theft is detected.

Nordstrom's has a special method for finding shoplifters. Whenever they sell an expensive item, they put a secret mark onto it. Thus, expensive items that are purchased are specially marked, while stolen expensive items will not be so marked (inexpensive items are never

marked, since it is too expensive to mark everything sold). When there is a return, we want to make sure someone didn't steal the item, then return the stolen item for cash.

With the rules given, there are four cases for the stolen light logic to consider:

- An expensive item with the mark is not stolen.
- An expensive item without the mark is stolen.
- A non-expensive item without the mark is not stolen.
- A non-expensive item with the mark will never occur, so is a Don't Care.

For this circuit, create a design by hand for each of the outputs. You will need to use K-Maps or Boolean Algebra to optimize the design (K-Maps will likely be the best way). Then, write the corresponding code in Verilog in Quartus II and simulate it. Finally, download the design to the FPGA, and use the switches and lights on the board to connect to the circuit.

Once you have the design working on the FPGA, draw the schematic of the design. This will be used for evaluating the quality of your design.

Note that for all design problems, you will be graded 100 points on correctness, style, testing, etc. For this lab, the bonus goal is to use as FEW logic gates as possible. Each gate (an individual Inverter, an individual AND, and an individual OR) appearing in your hand-drawn circuit diagram costs the same, and gates (other than inverters) can have as many inputs as you want. Unlike lab #2, inverters on the inputs DO count towards your total; each and every gate on your schematic will count as 1 gate. Note that you can only use standard gates (AND, OR, NAND, NOR, NOT, XOR, XNOR) – no AND gates with one input inverted, etc.

Please use switch Sw9, Sw8, Sw7 for U, P, C, respectively, and Sw0 for the secret Mark.

Note that with this, and all labs, keep your files when you are done with the lab – they will often get reused in subsequent labs!

Lab Demonstration/Turn-In Requirements

A TA needs to "Check You Off" for each of the tasks listed below.

- Turn in to the TA the K-maps or Boolean simplification you did to create your design.
- Turn in a printout of the Verilog code you produced for your design.
- Turn in your circuit diagram.
- Demonstrate your UPC code circuit working in simulation under ModelSim.
- Demonstrate your UPC code circuit working on the DE1 board.
- Tell the TA how many hours (estimated) it took to complete this lab, including reading, planning, design, coding, debugging, testing, etc. Everything related to the lab (in total).