18-240: Structure and Design of Digital Systems

Electrical & Computer ENGINEERING

HW3 [12 problems, 64 points]

Covers lectures L06 – L07 *Due: 25 September 2023*

Homework sets are due at 5:00PM on the due date. Upload your answers, as described below, to Gradescope by then. No late homework will be accepted. Remember, we let you drop two homework assignments over the semester.

To hand in your homework, create a PDF either by scanning your paper work, or by generating it in a software tool to start with (e.g., a word processor, LaTeX). Ensure your PDF is readable, your work legible and rotated correctly. Upload your work to Gradescope and follow the Gradescope instructions carefully to ensure your work is graded properly. Points will be deducted if you do not follow directions, for instance, by not assigning problems to pages (a five point penalty).

All other files must be submitted by running our handin script, which generates *a second PDF file* of your source code. If you have never run the handin script for *any* 18-240 assignment before, please read the *Overview of handin240* page on the course Wiki. The setup needs to be done only one time for the semester. Make sure you have a copy of all of the files that you need to submit in the same directory in your AFS space. Once you are ready to submit, run:

```
$ handin240 hw3
```

Note that everything that you type in should be lowercase. This script will do three things: (1) it will check to see if all of the required files are present, (2) it will automatically copy all of the files required for the homework to a handin directory where the TAs can run the code, and (3) it will generate a PDF of all of the required files called **`HW3_code.pdf`**. *After the script finishes, you must upload this generated PDF to Gradescope.*

Remember: **all files need to be generated and uploaded before the deadline**, so leave ample time for submission!

Discussions about homework in small groups are encouraged — think of this as giving hints, not solutions, to each other. However, homework must be written up individually (no copying is allowed). If you discussed your homework solutions with someone else, either as the giver or receiver of information, your write-up must explicitly identify the individuals and the manner information was shared.

If you use an AI assistant (ChatGPT, or others) for help on any of these problems, you must ensure that your answer is completely your work. Do not simply copy-n-paste any part of a ChatGPT conversation into your answer. And you must cite the AI assistant, with a thorough description of what help you received. For example, *"Conversation with ChatGPT 3.5 consisting of approximately 12 prompts asking for a thorough understanding of how to do Boolean proofs."*

You must show details of your work. There is no credit for just writing down an answer.

Drill Problems [32 points]

Drill problems are graded leniently based on your approach and effort; and not entirely on correctness. That means it is possible for you to have a perfect score on a problem whose answer is actually incorrect. Please check your work with the published solutions to verify correctness.

1. [3 points, Lecture 6] Pick any six decimal numbers between -127 and 127 (inclusive). Convert them to 8-bit signed magnitude and 8-bit two's complement format. Show your work.

2. [3 points, Lecture 6] Pick any six 8-bit binary numbers (other than those in Problem 1). Convert each into decimal three times; once for an unsigned binary format, a second time for a signed magnitude format and a third time for two's-complement format. Show your work.

3. [5 points, Lecture 7] The 4-input LUT on an FPGA has been configured such that TT0-TT15 = {0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1}. What will be the minimized function of the output for inputs {**A**, **B**, **C**, **D**} where **A** is the most significant?

4. [3 points, Lecture 7] What is wrong with this code for a 2-4 decoder with enable? Explain.

```
module decoder24en
  (input  logic [1:0] binary,
   output logic [3:0] onehot,
   input  logic       en_L);

  always_comb begin
    case(binary)
      2'h0 : onehot = 4'h1;
      2'h1 : onehot = 4'h2;
      2'h2 : onehot = 4'h4;
      2'h3 : onehot = 4'h8;
    endcase
    if (~en_L) onehot = 4'h0;
  end
endmodule : decoder24en
```

5. [6 points, Lecture 6] Write a SystemVerilog module that calculates the sum of two BCD numbers and a **carryIn**. The module header is:

```
module bcdAdd
    (input  logic [3:0] a, b,
     input  logic       carryIn,
     output logic [3:0] sum,
     output logic       carryOut);
```

The **sum** output must be a BCD number. **carryOut** is the carry out to the next higher digit.

▷ **Submit this as a file named `hw3prob5.sv`.**

You undoubtedly need to create a testbench to ensure this works, but we won't grade you on it.

6. [6 points, Lecture 6] In Lecture 7, you learned that a carry-lookahead adder relies upon the recurrence equation $C_{i+1} = G_i + C_i P_i$.

Expand this equation to get an expression for C_4 in terms of Ps and Gs and C_0.

Use that equation to provide a count of how many gates of each type (and with each number of inputs) is required to build the entire carry-lookahead adder (for instance, C_2 requires one 3-input OR gate, one 3-input AND gate, ...).

7. [6 points, Lecture 7] Lecture 7, slide 14 showed a circuit that I could use to determine if I could drive to school each day by generating the signal **driveToSchool.L**. It relied upon input signals **keysInPocket.H**, **carInGarage.H**, **Temp<40F.L**, and **Raining.L**.

Convert that circuit to work with input and output signals that are the opposite activity levels: **driveToSchool.H**, **keysInPocket.L**, etc. Draw a schematic of the resulting circuit, making sure that it adheres to the mixed-logic standard you saw on slide 16 of the same lecture.

Non-Drill Problems [32 points]

8. [4 points, Lecture 7] If you attach a multi-meter to the output of a tri-state driver, what voltage will it measure when the driver is not enabled? Explain.

9. [4 points, Lecture 7] If you attach a multi-meter to the the output of a tri-state driver on a a circuit board with lots and lots of chips and connections, what voltage will it measure if the driver is not enabled? Assume the circuit board is powered and functioning normally. Explain. (Note: this question is very similar to the one above. The fact that I'm asking it again must mean something is different about this scenario.)

10. [6 points, Lecture 6] The old mechanical cash registers were more complicated than you might think. They did their arithmetic in decimal. For the most part, you just added items (the costs of the items). But if you had to subtract off a discount, it used tens complement arithmetic to do the subtraction. Given a cash register whose highest sum total is $999.99, and I wanted to subtract off a discount totaling $23.37 from the current sum, what value would the machine add to the sum? i.e., what is -23.37? Just writing down the answer isn't good enough here. You must answer in terms of "complement arithmetic." e.g., use analogies to 2s complement arithmetic to explain how you calculated the answer.

11. [10 points, Lecture 6] Suppose we are working with 6-bit two's-complement numbers. Consider the two arithmetic operations below:

$$\begin{array}{rcccccc} & 1 & 1 & 0 & 1 & 0 & 0 \\ + & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ \hline \end{array} \qquad \begin{array}{rcccccc} & 0 & 1 & 1 & 0 & 1 & 0 \\ - & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ \hline \end{array}$$

Answer this: what is the largest (most positive value) and smallest (most negative value) that will not produce an overflow for each of the two operands, $a_5a_4a_3a_2a_1a_0$ and $b_5b_4b_3b_2b_1b_0$. In other words, what is the legal range of each operand, so that overflow won't occur? *Explain*.

12. [8 points, Lecture 7] What is the truth table for $F(A,B,C,D)$ from the circuit below?