

Name

CompSci/ECE350 Spring 2023 Midterm Examination 1 – 2/21/23

*This is a closed book, closed notes, closed internet exam to be taken in 75 minutes. Please answer all 5 questions. $15 \times 5 = 75$, use your time wisely! Note some questions are worth a little more, some a little less. You are permitted one (single sided) 8.5" x 11" sheet of notes. **KEEP YOUR NOTES PAGE TO AUGMENT FOR MIDTERM 2. DO NOT STAPLE. DO NOT WRITE ON THE BACK OF PAGES OF THE EXAM OR YOUR SCRATCH PAGES** - I need to scan them all into Gradescope, and double-sided scanning is fraught with peril. Try to put your final answers in the spaces provided, but at least give clear instructions there where to find the final answers. Turn in any extra scratch pages you might have if they might allow us to better understand your answers.*

Please, if accurate, sign this statement before turning in your exam¹:

I have neither given nor received any unauthorized help on this examination.

¹ If the statement is not accurate, we need to talk!

1. Combinational Logic Minimization (22 points total)

Thanks to random.org for the minterms for the function

$F(x_1, x_2, x_3, x_4) = \sum m(2, 7, 8, 9, 10, 12, 13, 15)$; true and complemented inputs are freely available.

a) [3 pts] Give the canonical sum-of-products representation of F . (The logical expression in full written-out form)

b) Give the minimal sum-of-products form of F (give the logical expression [5 pts] and draw the 2-level sum-of-products circuit **IMPLEMENTED WITH ONLY NAND GATES** [3 pts]).

expression

Logic diagram

Name

c) Give the minimal product-of-sums form of this function (give the logical expression [5 pts] and draw the 2-level product-of-sums circuit [3 pts]).

expression

Logic Diagram

d) [3 pts] Compare the cost (number of gates plus number of inputs) of the minimal POS and SOP solutions, assuming complemented and uncomplemented versions of primary inputs are available at no extra cost.

Name

2. Priority Encoders (24 points total)

We discussed regular encoders way back in lecture 3; an 8:3 encoder, for instance, was constrained to always have exactly one of its 8 inputs activated; the 3-bit output would reflect which input (0-7) was currently active. This constraint is a severe one; there is no “no input” condition, and we must always have exactly one input active. Let’s relax the latter condition by building an 8:3 Priority Encoder, where the output of the encoder is the ID (0-7) of the highest priority input currently active. Input 0 is the highest priority, input 7 is the lowest. We still insist that at least one input be active at all times. Assuming you have logic gates of all standard types with up to 8 inputs freely available, implement an 8:3 priority encoder. Hint: slow, i.e. bad way is to construct a 256 entry truth table for the 8 inputs.... Show the (not necessarily minimized) logic equations (no need to draw the gate level design), and give a brief explanation of how you arrived at this. You can assume true and complemented versions of the 8 primary inputs x_0-x_7 are available. (NOTE! There are solutions with fewer than a dozen gates).

Explanation of design [6 pts]

Design Equations [18 pts]

Name

3. 9's and 10's Complement Arithmetic (16 points total)

[4 points each] 9's and 10's complement numbers for base 10
addition/subtraction

Completely analogous with 1's and 2's complement representations for binary numbers, there are 9's and 10's complement representations for base 10 numbers. As with binary, we can replace subtraction for decimal numbers with addition of the 9's or 10's complement of the number. (I taught a dorm mate of mine in House CC who was troubled by subtraction how to do this and he seemed happy...). In a world of ***n decimal digit numbers***, the 9's complement of a number is formed by subtracting it from $10^n - 1$ (i.e. n 9's). The 10's complement of a number is formed by subtracting it from 10^n (1 followed by n 0's). Negative numbers in 9's and 10's complement begin with the leading digit 9. The same rules for handling overflows out of the most significant digit that apply for 1's and 2's complement apply for 9's and 10's complement respectively.

So the 4 digit 9's complement representation of -343 would be 9656, and the 4 digit 10's complement representation of the same number would be 9657. Of course, the 4 digit 9's and 10's complement representation of +343 would just be 0343.

In a universe of **4-digit decimal numbers**, complete the following problems with 9's and 10's complement representations - give the final answer as a regular decimal number (like -543).

	Original problem	4-digit 9's complement version	4-digit 10's complement version
a	154 - 376		
b	(-245) - (-154)		

Name

4. Fun with Adders (18 points total)

- A. [4 pts] Give the design equations and a gate level implementation of a full adder cell (Primary inputs only are available - A, B, C_{in})



- B. [7 pts] We have noted in class that our FPGAs use Look Up Table devices rather than conventional logic gates to implement logic; LUTs are just like using multiplexers to implement functions with the limitation that only the constants "0" and "1" can appear at the inputs of the LUTs (i.e. no signal values). Use 3-input LUTs to implement a full adder cell.



Name

C. [7 pts] Due to a stockroom accident, you only have full adder chips available. Design a 3-input AND gate using only one or more full adders; the only other things you have access to are the primary inputs (true form only, no complements), power and ground (1 and 0), and wires! Draw your implementation.



Name

5. Booth Multiplication (20 points total)

In a **6-bit starting universe**, let's multiply $(-17) \times 22$ using (original) Booth's algorithm in whatever format works best for you, but be sure to show what happens on each step, even if it is to "do nothing", and write out the bit strings for each step). I want the full 12-bit answer.