

# Assignment 1: Combinational Circuits

**Out on:** January 29, 2018

**Due by:** February 9, 2018 before 10:00 PM

**Collaboration:** None

**Grading:** Packaging 10%, Style 10%, Correctness 80%

## Overview

The first assignment is mostly about simple combinational circuits of the kind we discussed in class: gates, decoders, adders, etc. It's possible for you to submit the assignment without drawing a single circuit, simply by writing out the correct formulas, discussing how you derived them, and describing what their characteristics would be as circuits. If you want to include actual circuit diagrams, please either "draw" them as ASCII art or include a PDF file with clearly labeled diagrams.

## Problem 1: Universal Gates (30%)

In lecture we mentioned that NAND (and NOR) gates are **universal** in the sense that **any** combinational circuit can be built **exclusively** out of NAND (or NOR) gates without any other gates.

For this problem, first show how you can implement the standard AND, OR, and NOT gates using **only** NAND gates. Then show how to do the same using **only** NOR gates. Optionally draw those circuits.

Since the set of AND, OR, NOT is intuitively universal as well, once you've done this problem you should be **very** much convinced that NAND and NOR are indeed universal.

**Note:** Just to make sure, you'll need to hand in six circuits for this problem; three will contain only NAND gates, and the circuits will implement AND, OR, and NOT; three will contain only NOR gates, and those circuits will implement AND, OR, and NOT. I hope this is clear?

## Problem 2: XOR and XNOR Gates (20%)

Below you'll find the truth tables for XOR and XNOR gates. Show how you can implement XOR and XNOR gates in terms of just AND, OR, and NOT gates.

a	b		a	XOR	b
-----	-----	+	-----	-----	-----
0	0		0		
0	1		1		
1	0		1		
1	1		0		

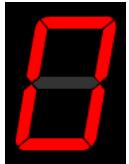
  

a	b		a	XNOR	b
-----	-----	+	-----	-----	-----
0	0		0	1	
0	1		0	1	0
1	0		1	0	0
1	1		1	1	1

You should use the "design process" we talked about in lecture: From the truth tables, derive a formula for the circuit in DNF or CNF (and optionally draw the circuit). **Do not simplify the circuits, leave them in CNF or DNF!**

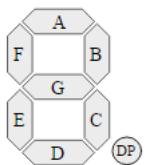
## Problem 3: Seven-Segment Display (50%)

Seven-segment displays can be found in a huge number of electronics applications, including on some fancy PC motherboards (where they usually indicate errors). Everybody knows what they look like, but just in case:



(The Seven Segment GIF Image is located at  
[https://en.wikipedia.org/wiki/File:7-segments\\_Indicator.gif](https://en.wikipedia.org/wiki/File:7-segments_Indicator.gif))

Each of the seven display elements can be switched on and off independently through one of seven pins. The pins are labeled A to G according to the following scheme (forget the decimal point, we won't use it):



Here is how the hexadecimal digits from 0 to F are usually represented on a seven-segment display:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

For this last problem, you have to design a circuit that takes a binary digit 0-7 encoded on three lines and displays it correctly on a seven-segment display by outputting 0/1 on the correct pins A-G of the seven-segment display. Your circuit takes the

inputs X, Y, and Z on which the digits 0-7 are supplied as follows:

Digit	X	Y	Z
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

The outputs of your circuit are A-G connected to the seven-segment display as shown above. Good luck! :-D

You should once again use the "design process" we talked about in lecture: From the truth tables, derive formulas for the circuit(s) in DNF or CNF (and optionally draw the circuit). **Do not simplify the circuits, leave them in CNF or DNF!**

**Note:** If you're really into circuits, also design the circuit for 4-bit numbers. But this is **in addition** to the one for 3-bit numbers!

## Deliverables

Please turn in a [pdf](#) of your assignment to **Gradescope**, join the course with entry code **9RW4WE** (if you don't have a student account on Gradescope yet, please register for one now); the filename should be `HW1_<your_name>.pdf` with *name* replaced by the first part of the email address you used to register on **Piazza** (so I would use `HW1_kwu34.pdf`).

## **Grading**

For reference, here is a short explanation of the grading criteria; some of the criteria don't apply to all problems, and not all of the criteria are used on all assignments. **Packaging** refers to the proper organization of the stuff you hand in, following the guidelines for Deliverables above. **Style** refers to either to programming style if we are talking about a programming problem, or to the clarity and readability of your solution for a written problem.

**Make sure to include your name and email address in the header of the PDF file you turn in.**