LAB 3 // Hardware

HARDWARE I/O AND KARNAUGH MAPS

INSTRUCTIONS and RUBRIC

CSE 2301 – Fall 2024

INTRODUCTION

Prerequisites

At this point, you should have received your kit, protoboard, logic probe, and wire cutters, all of which you will need for this and future forays into hardware. You should have also reviewed modules 5 and 5.5 (on Karnaugh maps and troubleshooting).

Objectives

The objective of this assignment is to design and build a code converter that will take the POSTNET 2-out-of-5 code as input and deliver an XS3 code as output. You will also learn the fundamentals of hardware. Much like programming, if you do not take the time to plan out your circuit, it will make your implementation and debugging process much harder.

Background

First, we will be building the input and output parts of the hardware circuit, which we will be using for the rest of the semester. We need input and output in each lab, so it is crucial that you not only build the circuits correctly, but understand how they work. We **STRONGLY RECOMMEND** that you read all of the theory in this document. It will make your circuit-building process far less painful, as you will have the knowledge to debug intelligently.

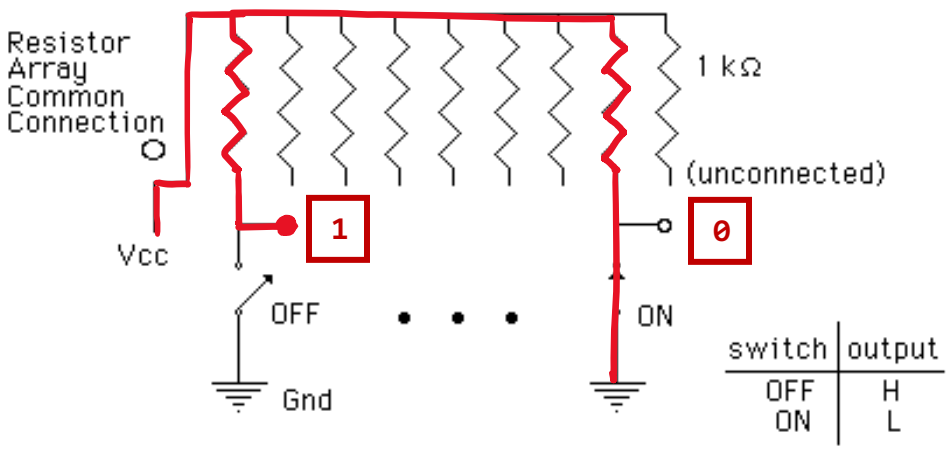
Due to the time investment required for this lab, it will be split into two weeks. In the first week, you will go over the theory for the Karnaugh Maps and complete the hardware input/output circuit. In the second week, you will be given some time to work on the hardware.

PART 1:

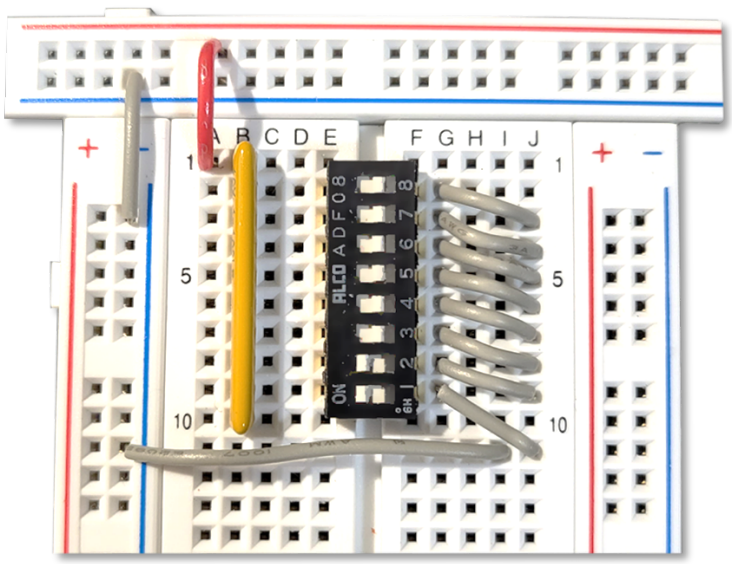
Building the I/O Hardware

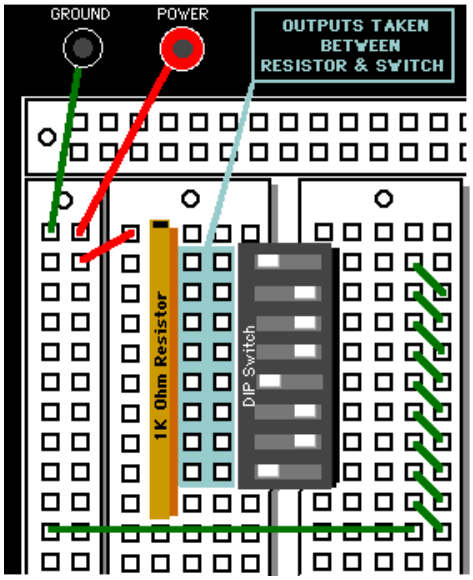
The Input Mechanism

The input circuit really just a switch, but it needs to be powered and grounded in order to function – that is, to supply a current in each line. Your kit includes 1kΩ and 330Ω SIP (single in-line) resistors. The 1kΩ resistor will be used for the input subcircuit, in conjunction with your DIP (dual in-line) switch. Current from Vcc (power) travels through the single in-line resistor to each pin. If the switch is "off", it is disconnected from the line, which actually means that the current goes to "out", and drives the line high (logic = 1). Conversely, when the switch is "on", it connects the line to ground, and since there is negligible resistance on the grounded path, the "out" line is considered low (logic = 0). This leads to the common misconception that logic is "inverted".



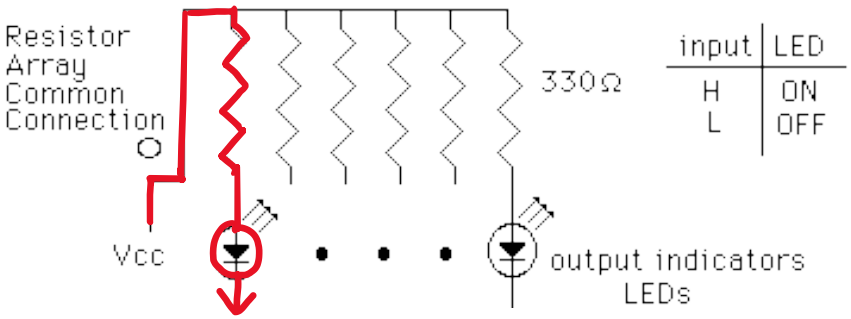
YOUR TASK – Build the input circuit using the image below as a guide.
Your circuit should be practically identical to the physical implementation.



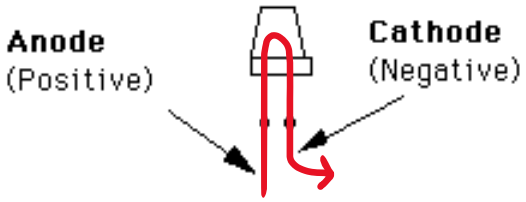


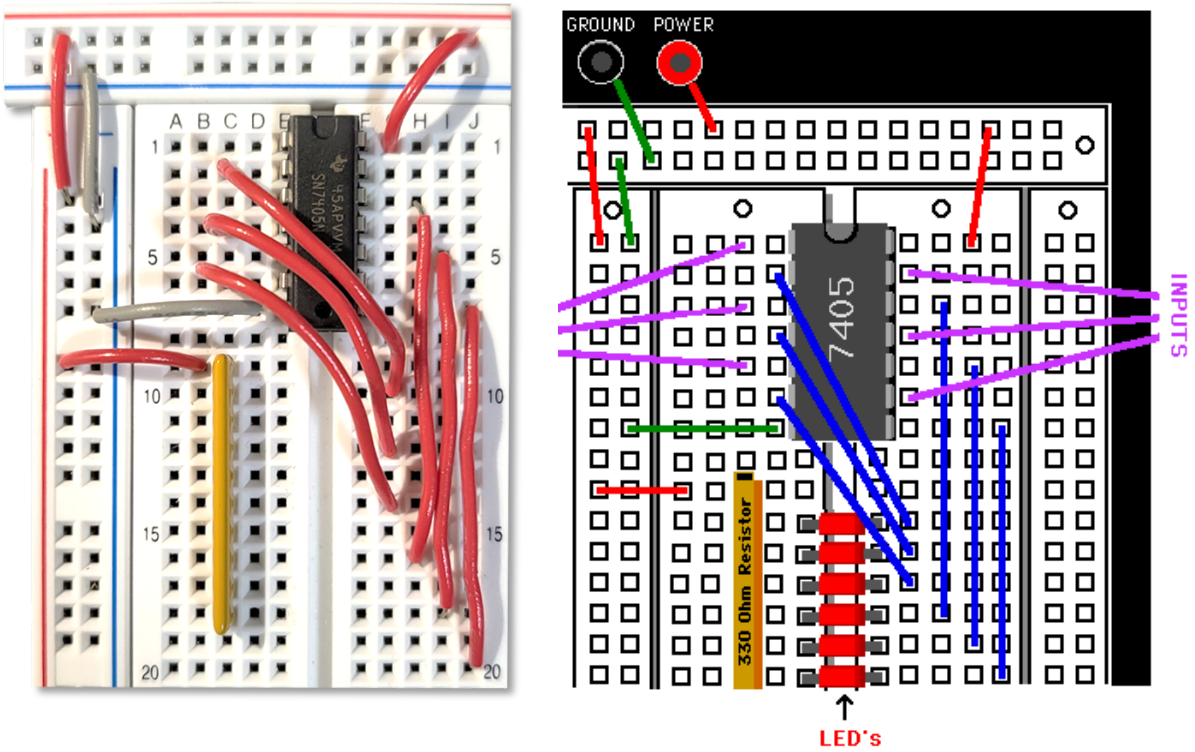
The Output Mechanism

Our output subcircuit allows you to monitor the logic in six output lines at once, but no more. During the final project you will expand this. Here we will use a 7405 chip (known as an integrated circuit, or IC) which is a special type of inverter. Whereas a normal inverter like a 7404 would take a **0** and turn it into a **1**, or vice versa, when a 7405 receives a **0** (low), an internal capacitor generates high impedance at the associated pin, preventing any current flow from the other side. Alternatively, when the input is **1** (high), impedance is low, and the current from the other side is free to flow into the 7405, to ground. This is known as an "open collector gate", and it is necessary for driving an external load (where power to the LEDs does not come from the 7405). In the diagram below, recognize that power is coming from the SIP resistor.



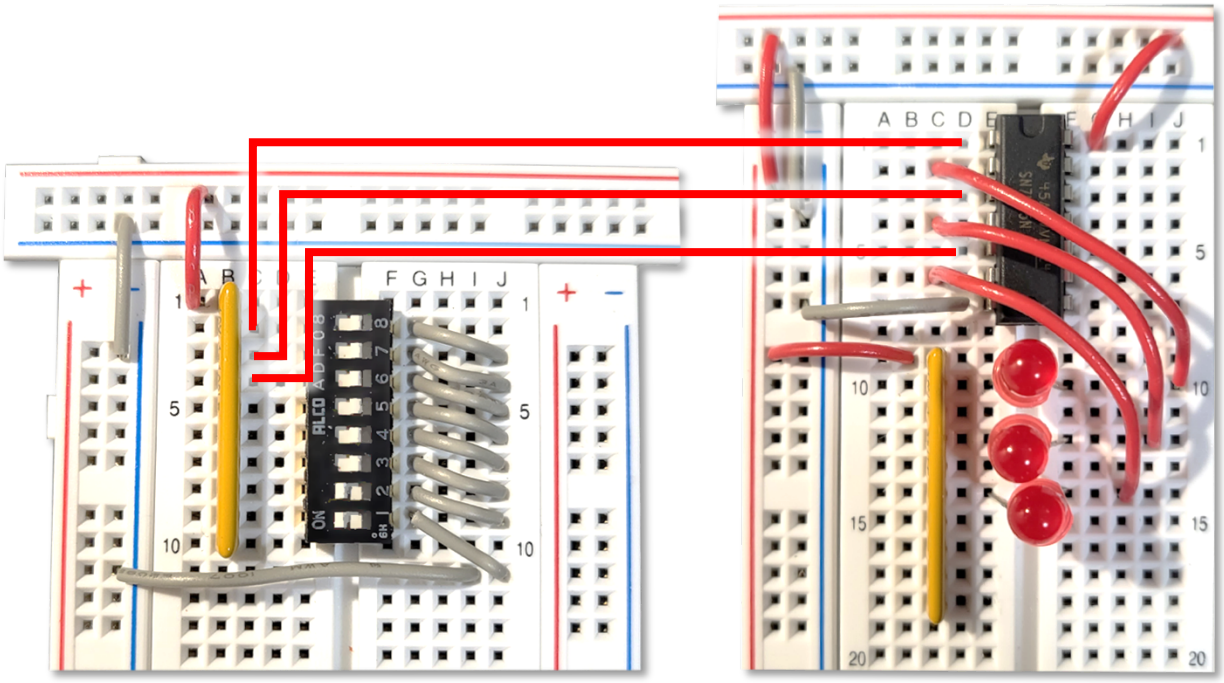
YOUR TASK – Build the output circuit using the images below as a guide.
Unlike resistors, LEDs are sensitive to the direction of current flow. Consequently, the anode (long leg) must be placed on the resistor side, whence the power is coming.





Testing the implementation

Wire six of the switch outputs to the six available pins of the inverter (7405). On the switch side, place wires between the resistor and the DIP switch itself. On the inverter side, insert each wire one pin above a line connecting to an LED. For example, if a red wire is in line 2, then it must be the inversion of line 1, so I'll place a green wire in line 1.



PART 2:

Karnaugh Maps

POSTNET code is weighted 74210. It is this peculiarity which allows it to represent every value 0-9 with 2-out-of-5 bars (except for zero, which is arbitrarily given 11000). A table indicating all allowable POSTNET codes is shown below. Your first task will be to fill out the VWXYZ (POSTNET) to DCBA (XS3) conversion table also modelled here. Recall that XS3 is just unsigned binary plus three.

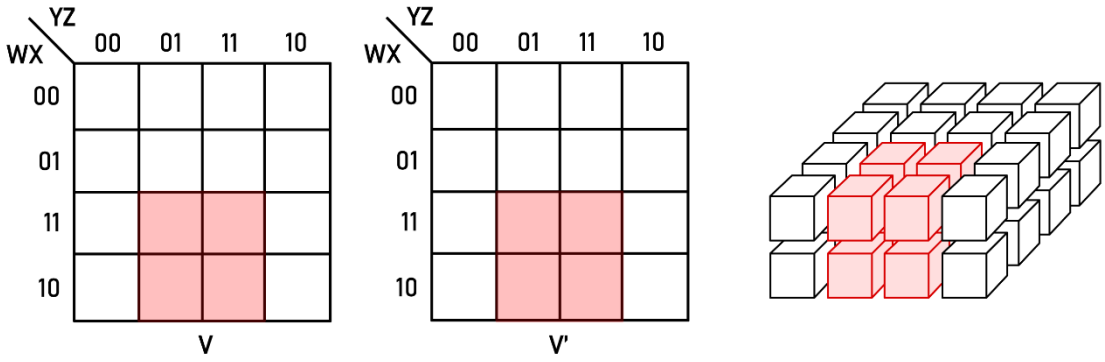
Term	*V*	*W*	*X*	*Y*	*Z*
0	1	1	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	1	0	0	0	1
8	1	0	0	1	0
9	1	0	1	0	0

	POSTNET					*XS3*			
Weight	**7**	**4**	**2**	**1**	**0**	**8**	**4**	**2**	**1**
Term	**V**	**W**	**X**	**Y**	**Z**	**D**	**C**	**B**	**A**
0	1	1	0	0	0	0	0	1	1
1	0	0	0	1	1	0	1	0	0
2	*…continued to 9…*								

Now you will be tasked with creating a circuit the converts between these two systems. Alas, using the technique that we employed for Lab 2 with binary to ternary conversion would be needlessly complex (and rather arduous to wire), so we will use 5-variable Karnaugh Maps instead. If you have not done so already, it is absolutely critical that you review Module 5 before continuing.

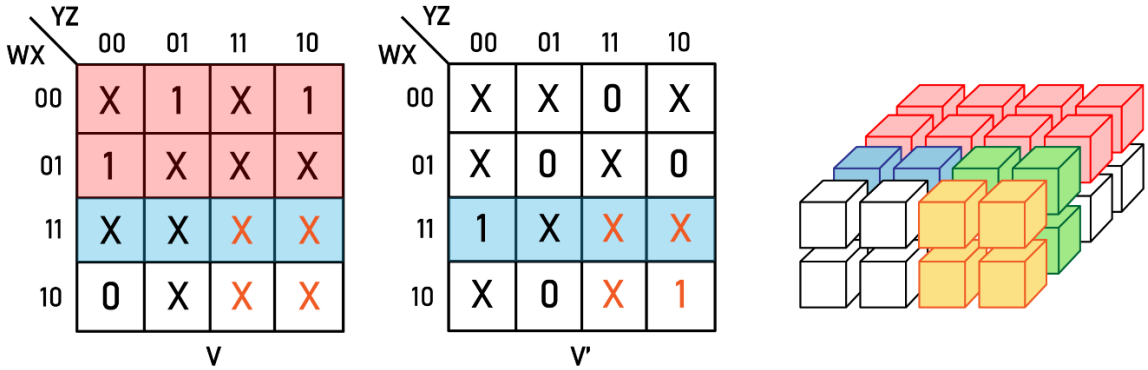
YOUR TASK – First, complete the table converting from POSTNET to XS3. Only fill out the table for decimal 0-9, as POSTNET cannot represent further values anyhow. This will give you the values you require to develop the Karnaugh map. Then complete all four K-maps.

Since we are converting from a 5-variable system to a 4-variable system, we will have five inputs and four outputs, so we will need a 5-variable Karnaugh maps for each XS3 binary digit (D, C, B, and A). Set up your tables like the one seen below. A visualization is given to show you how 5-variables should be considered a "stack" of two 4-variable maps.



Now, place each value from the POSTNET to XS3 table you created onto a Karnaugh map. Make sure it's the right one, though! As an example, for the decimal value `0` the VWXYZ code (POSTNET) is `11000`, and DCBA (XS3) is `0011`, so on the Karnaugh map for the variable D, we would place a `0` in $VW\overline{XYZ}$ box. After you have finished placing all of the ones and zeroes, you will be left with blank spaces, where you should place X's to indicate that these are don't care states; we will never need these values for out `0`-`9` representation, and therefore they can be treated as zeroes or ones on the map (whichever is more convenient). Below is the completed map for D. Do it yourself and check your result against ours to be sure that you're working through the problems properly.

Note the use of colored letters to indicate selection. The equation is D = $V\overline{W} + WX + WY$.



PART 3:

LogicWorks Mock-Up

YOUR TASK – You must now build the circuit in LogicWorks using the four equations you've developed. Use the techniques you learned in labs 1 and 2. Once you've built it, test each valid input (as indicated on your table) exhaustively to make sure that you are obtaining the expected outputs. Then demo to your TA. This concludes the week 1 tasks.

SCORING

Demo Requirements [5 pts]:

- ✓ Complete the input and output hardware.
- ✓ Connect six switches to the 7405 inverter and demonstrate that flipping each switch turns the associated LED on/off. **Demo (3 pts).**
- ✓ Finish the POSTNET to XS3 table.
- ✓ Draw out all four 5-variable Karnaugh maps and derive the DCBA equations.
- ✓ Create a LogicWorks circuit mirroring the equations. **Demo (2 pts).**

Report Requirements [3.8 pts]:

- ✓ [1.7] Theory:
  - [1.3] Explain the functionality of the 7405 in your own words. Why are we using a 7405 inverter instead of a 7404? What is an open collector? What do we mean by "driving an external load?" Our instructions provide an overview of the fundamentals, but you will need to do some research of your own to receive full credit. Be thorough.
  - [0.4] How much current will each LED in your output circuit be taking?
- ✓ [1.7] Deliverables:
  - [0.5] Include your full table of POSTNET (VWXYZ) to XS3 (DCBA) conversions from decimal 0 to 9.
  - [1.2] Show all four of your 5-variable Karnaugh maps and the work you did to minimize each equation. This can just be a photo of your work, but try to keep it readable. Also include the final, minimized equations for D, C, B, and A.
- ✓ [0.4] Discussion section. Should conform to standard lab report guidelines.

Practice Question [1.2 pt]:

- ✓ [0.5] Question 1:
  Create a 4-variable Karnaugh map for $\sum m\,(0, 2, 3, 4, 6, 7, 8, 10, 12)$ and reduce it as much as possible. Suboptimal solutions will receive partial credit.
- ✓ [0.7] Question 2:
  Convert the number -19.5_{10} to IEEE-754 floating point. Include only the first 8 bits of the mantissa. Show *all of your work.*