Lab 2 Displaying Data

Deadlines & Grading

- **Prelab**: Thursday, February 6, 2014 at 11:59 PM – 25 points, **to be done individually**
- **In-Lab Exercises**: Monday, February 10 / Tuesday, February 11 – 45 points, **in groups of 2**
- **Report**: Thursday, February 20 / Friday, February 21 at 11:59 PM – 30 points, **in groups of 2**

Section I: Overview

While it can often be overlooked, data visualization remains an important part of building any circuit. One of the simplest such representations is to show a number as a status. Since circuits use *Boolean logic* (which represents everything as zeroes and ones), we must translate all of our calculations within the circuit from a *binary number system* to a *decimal number system*, which is more easily readable by humans. In this lab, we will design a circuit that can take in a 4-bit binary number and translate this number into a displayable digit. This digit will be displayed by turning on the individual LEDs of a *seven-segment display*.

Section II: Background

As our reliance on technology continues to grow, it can be helpful to take a look back at how these concepts were enabled. One of the most fundamental enablers has been the use of the binary number system, which allowed us to connect the concept of electrical switches (which can only be on or off) to a mathematical representation. As a result, we can harness electricity to do calculations, which are at the heart of all digital circuits.

English mathematician George Boole developed a system of logic in 1854 (in a work titled *An Investigation of the Laws of Thought*) that allowed logical arguments to be made using mathematical constructs; after his death, this became known as *Boolean algebra*.[1] This foundation was not exploited using electricity until several decades later, when two engineers independently created groundbreaking works. John Atanasoff had an epiphany at a bar in 1937, making the connection that an electrical calculator could be created if we use the binary number system. Starting from the initial design on the back of a cocktail napkin, Atanasoff's idea went on to become the Atanasoff-Berry Computer (Figure 1), the first digital

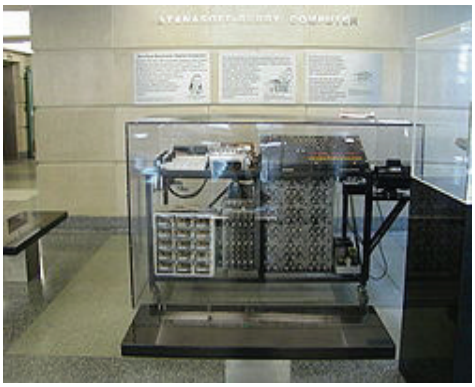


Figure 1. Replica of the Atanasoff-Berry Computer at Iowa State University.[3]

computer.[2] Almost simultaneously, Claude Shannon wrote his Master's thesis, *A Symbolic Analysis of Relay and Switching Circuits*, which demonstrated that *relays* (electromechanical switches) could be used to implement Boolean algebra.[4]

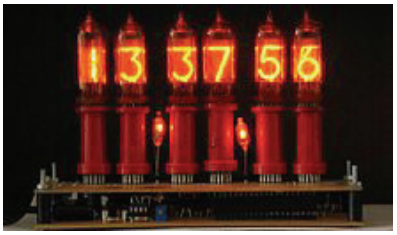


Figure 2. Nixie tube clock.[9]

Of course, there needed to be some way to represent the outputs of these machines. In fact, even before digital logic was born, the *seven-segment display* was born, allowing the use of seven incandescent lights turned on and off behind a plate of black-painted glass to represent the different digits. The first documented use of a seven-segment display goes back to 1910, to display information at a power plant.[5] Since this was a rudimentary technology, engineers for some time turned to mechanical alternatives. Railway stations would use rapid split-flap displays, while buses used flip-disc displays, both of which allowed the user to rotate through a series of painted numbers and letters.[6,7] The Nixie tube (Figure 2), used as early as 1955, provided 10 coils (like a light bulb), with each coil in the shape of a number – if the correct pair of contacts were connected, the desired number would light up.[8]

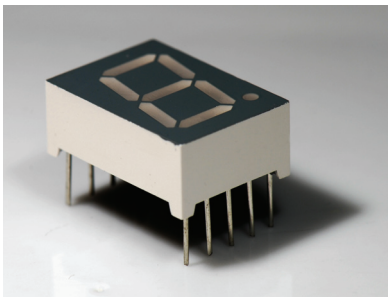


Figure 3. Seven-segment display.[10]

Once *light-emitting diodes* (LEDs) were invented in 1962, seven-segment displays began to take off (Figure 3). With LEDs, the display could be made compact, and the number shown was very legible. Over the years, seven-segment displays have become pervasive. Today, these displays can be found on digital clocks and watches, in microwaves and cars, and on audio-video electronics. Variations of seven-segment displays were used to express numbers and letters in different languages. For example, Russian and Bengali numbers were better represented using nine-segment displays,[11,12] while 14- and 16-segment displays (Figure 4) allow us to show the entire Latin alphabet by providing greater flexibility.[13]

Figure 4. 16-segment display.[14]

Today, along with *n*-segment displays, we have more complex display types that allow us to express richer information. *Dot-matrix displays* (Figure 5) contain rows and columns of dots, which can be turned on and off by a built-in controller; the circuit generating the output communicates with the controller to inform it of what text to display.[15] Computers (and their derivatives, such as smartphones) use monitors to display even more detailed information. Like a dot-matrix display, a monitor is simply told which dots (*pixels*) to turn on and off (as well as which color to show), but unlike the dot-matrix display, it is up to the computer itself to figure out how to arrange these pixels to display desired text and images. Even with these advances, seven-segment displays are still used due to their low cost and high visual contrast.

Figure 5. Dot-matrix display.[16]

Section III: Prelab

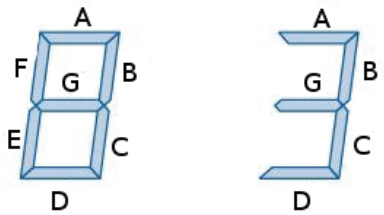
A. Designing a Decoder for a Seven-Segment Display

The purpose of this assignment is to design and implement a decoder for the seven-segment display that will allow us to display the numbers 0 through 9. The decoder that you are asked to design takes a four-bit input, ***W***, ***X***, ***Y*** and ***Z*** (which represents a single decimal digit), and outputs seven one-bit signals, ***A***, ***B***, ***C***, ***D***, ***E***, ***F***, and ***G***, that control turning on and off the appropriate LED segments of the seven-segment display. The outputs ***A***, ***B***, ***C***, ***D***, ***E***, ***F***, and ***G*** are assigned to the segments of a seven-segment display as shown in Figure 6a.

W	X	Y	Z	Digit
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Table A. Encoding for the seven-segment display.

When any of the variables ***A***, ***B***, ***C***, ***D***, ***E***, ***F***, or ***G*** are set high, then the corresponding segment will turn on. (This type of a circuit is often referred to as *active high*. For *active low* logic, segments are turned on when the corresponding inputs are low.) For example, if we set (***ABCDEFG***) = (1111001), the digit displayed is '3' – see Figure 6b. Figure 7 shows how all ten decimal digits are displayed.



(a) *(b)*

Figure 6. Outputs labeled on the seven-segment display.[17]

Note that since we are encoding decimal digits, we only need 10 out of the possible 16 combinations of the 4-input signals. Table A shows how the inputs map to digits. The output for the remaining encodings can be treated as *don't cares*, which means they can output either logic "0" or logic "1." These don't cares will allow you to simplify the circuit that you build, which you can take advantage of using the Karnaugh maps.

Your goal is to design an active high driver circuit for the seven-segment display. Driving active high means your circuit must output logic "1" in order to get a segment to turn on.

NOTE 1 The only parts that you can use to build your decoder are the 74LS00 (2-input NAND), 74LS10 (3-input NAND), 74LS20 (4-input NAND), 74LS30 (8-input NAND), and 74LS04 (NOT). The pinouts for these parts can be found in Appendix A (or online, by searching for 74LSxx).

Figure 7. Decimal digits on the seven-segment display.[17]

ASSIGNMENT 1 Make the truth table and Karnaugh map for the Boolean expressions associated with each segment of the display. You **must** design your circuit so the seven-segment outputs look **exactly** like those in Figure 7. As an example, we have made the Karnaugh map for segment ***A***:

WX / YZ	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	0	X	X

ASSIGNMENT 2 Using the Karnaugh maps, find minimized sum-of-products (SOP) equations for the expression derived in Assignment 1. For example, the Karnaugh map above results in the following minimized SOP (color-coded by grouping above):

$$A = W + X'Z' + XZ + YZ$$

ASSIGNMENT 3 Implement the expressions found in Assignment 2 using NOT and NAND gates found in the *integrated circuits* (ICs, or chips) listed in NOTE 1. Do this by writing out Boolean algebra. **Drawings of gates will not be accepted.** For our example segment ***A***, we get the following expression:

$$A = (W'(X'Z')'(XZ)'(YZ)')'$$

ASSIGNMENT 4 Using your answers for Assignment 3, write out a list of which chip pins will be connected together (as well as to the inputs and outputs). Keep the number of chips used at a minimum. To organize the list, we recommend describing specific pins using the format <chip type>–<chip count>–<pin #> (e.g., if we want to talk about pin 3 on the second 4-input NAND chip that we are using, we would label that as 74LS20-2-3). Make sure to write down which connections will be used for which segments (you can reuse connections for multiple segments). For segment ***A***, we have the following:

W → 74LS04-1-1
X → 74LS04-1-5
Y → 74LS04-1-13
Z → 74LS04-1-9
74LS04-1-6 → 74LS00-1-1
74LS04-1-8 → 74LS00-1-2
X → 74LS00-1-4
Z → 74LS00-1-5

Y → 74LS00-1-13
Z → 74LS00-1-12
74LS04-1-2 → 74LS20-1-1
74LS00-1-3 → 74LS20-1-2
74LS00-1-6 → 74LS20-1-4
74LS00-1-11 → 74LS20-1-5
74LS20-1-6 → ***A***

B. Prelab Deliverables

DELIVERABLE 1: prelab2.pdf *due February 6*
Submit answers for **Assignments 1-4** in PDF format. This file must contain the truth table and *all* Karnaugh maps, showing groupings of terms that you used in the minimized SOP equations. For each map, type up the *minimized* SOP equations.

Make sure to submit the above file to CMS (http://cms.csuglab.cornell.edu).

Section IV: Lab Exercises

In the lab you will build the seven-segment display circuit according to the design that you submitted in Section I using components listed in NOTE 1. Bring a copy of your prelab submissions (**updated with corrections based on our feedback**) with you to lab. In particular, make sure you have a copy of your wiring list (from **Assignment 4**) that you can check off.

When you come into lab, find a partner (who cannot be your Lab 1 partner), and compare your prelab to theirs. Pick the one that is both correct and simpler to build (if both are not the same). Then, start by assembling the logic for each segment, using your wiring list from **Assignment 4**. We recommend checking off each wire as you place it on the protoboard. In addition, **it is strongly recommended to build your circuit and get checked off one segment at a time.**

Some things to keep in mind while you're working on this lab:
- Make sure all switches on the proto-board are set to **TTL** or **+5** (which is our value of V_{CC}).
- Always make sure to hook up GND and V_{CC} to the chips properly. Also, never connect the output of a chip to either GND or V_{CC}.
- Remember to straddle all chips across the U-channel of the breadboard, to prevent pin shorts.

We have built the remaining part of the display driver for you, which you should see on your protoboard. Notify a TA immediately if you do not see this on your board. You are responsible for taking each output of your decoder and connecting it to the proper pin on the driver that we have built. For this, you should connect your outputs ***A*** through ***G*** to the 74LS241 chip, as shown in Figure 8. Feel free to ask a TA if you want to know how the driver circuitry works.

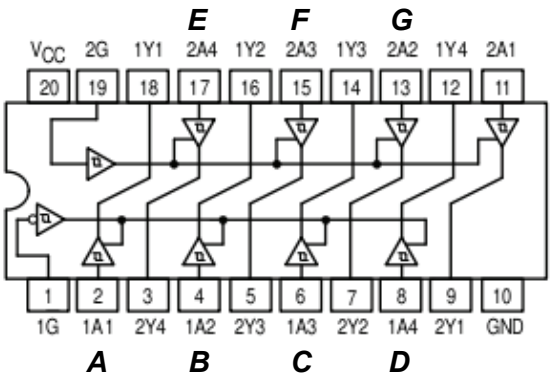


Figure 8. Pins to use on the 74LS241 chip.

The inputs to your logic circuit, ***W***, ***X***, ***Y***, and ***Z***, will come from the toggle switches at the bottom of the proto-board. You should use **SW1** for ***W***, **SW2** for ***X***, **SW3** for ***Y***, and **SW4** for ***Z***. For final verification, you should make sure that all of the values shown in Table A translate into a proper display on the seven-segment display.

Before leaving the lab:

- Make sure that you have had a TA check off all parts of your lab, including parts that aren't fully working.
- Turn off the proto-board, using the large switch at the top left.
- If you have used the computer, make sure to **save all files remotely** and then log off. Upon logging off, the computer automatically deletes any files that you created, so be careful!
- Clean your bench.
  - **DO NOT REMOVE ANY CHIPS OR WIRES THAT WERE ALREADY ON THE PROTOBOARD WHEN YOU CAME INTO LAB!**
  - Use a small flat-head screwdriver to gently pop all chips out of the protoboard.
  - Place all wires in the plastic box at the front of class. Return cutters and screwdrivers.
  - **Return all ICs to their correct slot in the box**, and return the box to a TA.
  - Throw out any stripped insulation or other garbage.

Section V: Lab Report

DELIVERABLE 2: lab2report.pdf *due **ten days after lab***

You will submit a single report for your group. **Each partner must share in the writing of the report.** Please refer to the Lab Guidelines on Blackboard for general information on what to include. The report must be a PDF file, named **lab2report.pdf**, and should be submitted through CMS (http://cms.csuglab.cornell.edu).

Groups in the Monday lab sessions must submit their report by **Thursday, February 20 at 11:59 PM**. Groups in the Tuesday evening lab session have until **Friday, February 21 at 11:59 PM**.

In general, lab reports for ENGRD 2300 should do the following:

- Assume that the reader has a general ECE knowledge, but has never seen this handout (and does not have access to it). This means you must summarize the purpose of this project, as well as the tasks you have been asked to complete. This also means you cannot refer to any portion of this document in your write-up.
- Include some sort of introduction.
- Use section headings to organize the document.
- Provide enough detail so we can reproduce your lab *exactly* as you designed it without looking at your submission.
- Include a section on the distribution of work for the jointly-done parts of the lab.

For this particular lab report, you should remember to discuss the following:

- Include updated versions of the Karnaugh maps and SOP equations from Assignments 1-2.
- Discuss the choices you made in designing your logic.
- Compare your prelab design with your partner's. If they differ significantly, please explain why, as well as the reasons you selected one over the other for implementation.
- Include complete schematics of your final design as built in lab (i.e., draw the logic gates out and show how they connect together). Do not include the wiring list from Assignment 4.

Appendix A: Selected 7400-Series Pin Diagrams

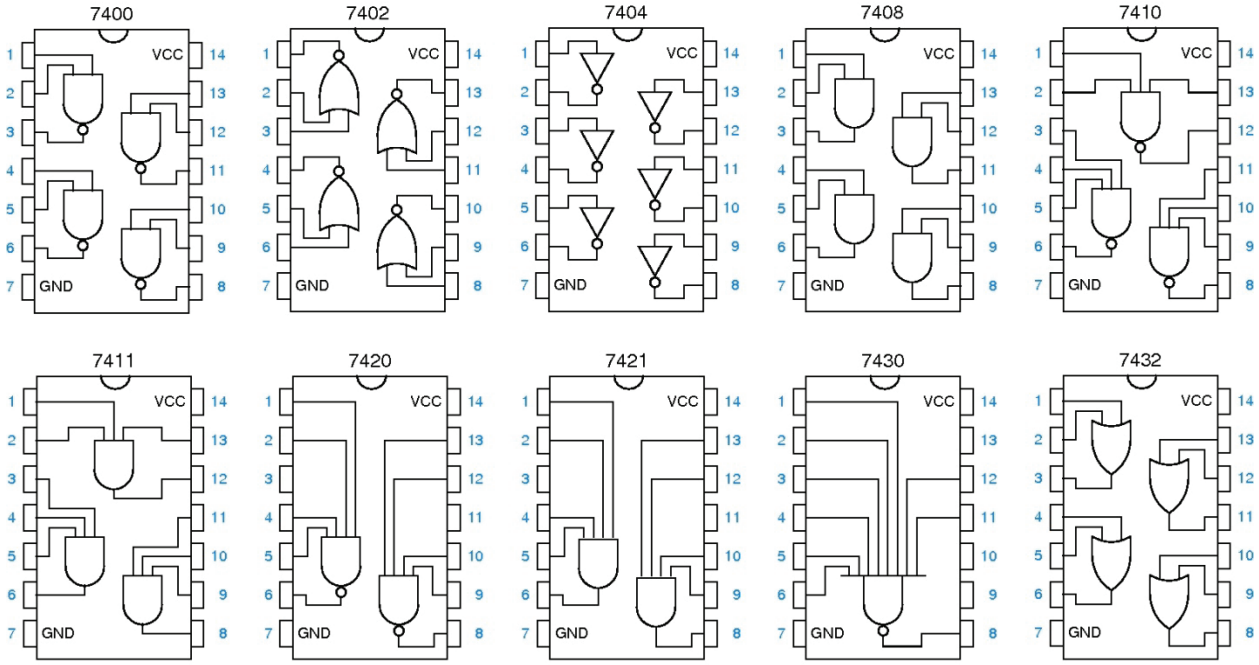


Figure 9. Selected pin diagrams for 7400-series integrated circuits.[17]

Appendix B: Frequently Asked Questions

When simplifying the Karnaugh maps for the prelab (Assignment 2), how do we know which don't cares to circle?

Do not circle don't cares unless they help make your circles with "1"s bigger. Also, remember that you can only make a circle that holds 2^n "1"s. If your circles follow these guidelines, and you still end up with multiple possibilities, then you likely have nothing to worry about. Multiple simplified equations just mean something that may have mattered in one combination does not matter in a different combination, but something else does matter to that new combination.

If two of my SOPs both contain an identical minterm, do I have to wire up separate gates for each of them?

No – in fact, if you're confident enough, you can use one single gate for both SOPs. For example, let's say we have two SOPs: $J = ((MN')'(Q'P')'T')'$, and $K = ((MN)'(Q'P')'(SV)'T)'$. Both J and K have the 2-input NAND gate $(Q'P')'$ in them. You could simply wire up a single NAND gate using a 74LS00 chip, by passing Q' to pin 1 and P' to pin 2. Now, instead of having one wire coming out of pin 3, you have two wires – one that goes to the 3-input NAND gate for J, and a second wire that goes to the 4-input NAND gate for K.

References

[1] Barnett, J. H. (2013). *Origins of Boolean Algebra in the Logic of Classes: George Boole, John Venn, and C.S. Peirce.* Mathematical Association of America. Retrieved January 31, 2014, from http://www.maa.org/publications/periodicals/convergence/origins-of-boolean-algebra-in-the-logic-of-classes-george-boole-john-venn-and-c-s-peirce

[2] *Milestones: Atanasoff-Berry Computer, 1939* (2013). IEEE Global History Network. Retrieved January 31, 2014, from http://www.ieeeghn.org/wiki/index.php/Milestones:Atanasoff-Berry_Computer,_1939

[3] *Atanasoff-Berry Computer at Durhum Center* (2006). Wikimedia Commons. Retrieved January 17, 2013, from https://commons.wikimedia.org/wiki/File:Atanasoff-Berry_Computer_at_Durhum_Center.jpg

[4] *How Do Digital Computers "Think"?* (2014). Computer History Museum. Retrieved January 31, 2014, from http://www.computerhistory.org/revolution/digital-logic/12/269

[5] Rogers, W. O. (1910, February 1). *Power Plant Signaling System*. Power and the Engineer: 32 (5) (pp. 204-206). Retrieved October 18, 2012, from http://books.google.com/books?id=0sIfAQAAMAAJ&pg=PA204

[6] *Supplemental Fact Sheet: Ferry Building Flap Sign* (2013, February 13). Metropolitan Transportation Commission. Retrieved January 31, 2014, from http://www.mtc.ca.gov/pdf/Flap_sign_fact-Sheet_02-13-13.pdf

[7] Ball, N. R. and Vardalas, J. N. (1994). *Ferranti-Packard: Pioneers in Canadian Electrical Manufacturing*. Kingston, Ontario: McGill-Queen's University Press. Retrieved October 31, 2012, from http://books.google.com/books?id=CrzgS5SoMzcC&pg=PA29

[8] Greenhill, L. (2012, January 16). *The Return of the Nixie Tube*. Stereophile. Retrieved January 31, 2014, from http://www.stereophile.com/content/return-nixie-tube

[9] *Nixie clock* (2007). Wikimedia Commons. Retrieved October 31, 2012, from https://commons.wikimedia.org/wiki/File:Nixie_clock.jpg

[10] *Seven segment 02 Pengo* (2007). Wikimedia Commons. Retrieved October 18, 2012, from https://commons.wikimedia.org/wiki/File:Seven_segment_02_Pengo.jpg

[11] Davie, A. (2005). *Elektronika 4-71b*. Museum of Soviet Calculators on the Web. Retrieved January 31, 2014, from http://www.taswegian.com/MOSCOWx/4-71b.html

[12] Mahmud, S., Chowdhury, T., and Hasan, M. (2005). *Designing a 9 Segment Display for Bangla and English Numerals*. Asian Journal of Information Technology: 4 (pp. 689-691).

[13] Embedded Artists AB (2013, November 25). *LPC812 MAX Experiment: 7-Segment Display*. mbed. Retrieved January 31, 2014, from http://mbed.org/users/embeddedartists/notebook/lpc812-max-experiment_7-segment-display/

[14] *Sixteen-segment display field* (2006). Wikimedia Commons. Retrieved October 31, 2012, from https://commons.wikimedia.org/wiki/File:Sixteen-segment_display_field.svg

[15] *Lab 12: Basics of LED Dot Matrix Display* (2011, April 28). Embedded Lab. Retrieved January 31, 2014, from http://embedded-lab.com/blog/?p=2478

[16] *3D dot-matrix font with reflection*. Shutterstock.

[17] Wakerly, J. F. (2005). *Digital Design: Principles and Practices*. Upper Saddle River, NJ: Prentice Hall.