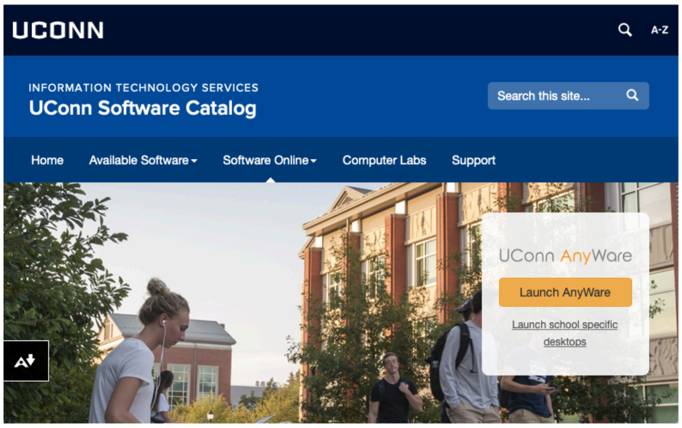
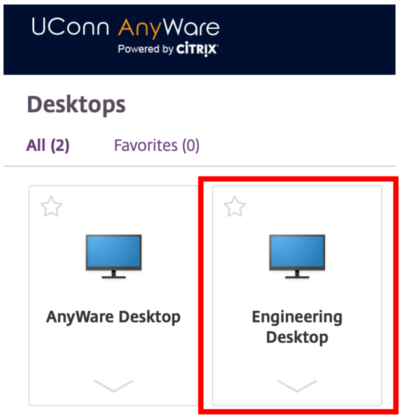
Introduction to LogicWorks

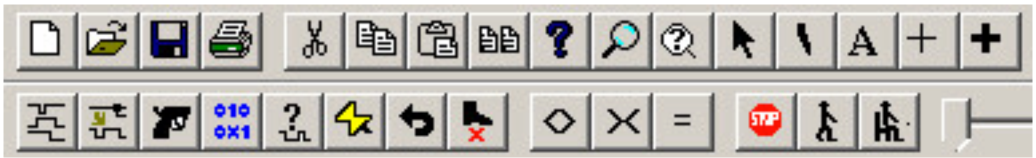
Exploring numbers

Your task is to learn the basics of LogicWorks simulation program. There are two exercises in this module. The first is to become familiar with basic circuit creation and the second is to create a functional logic circuit.





Access to the LogicWorks software is through UConn Anyware - Skybox (https://software.uconn.edu/uconn-software-online) – Engineering Desktop [https://confluence.uconn.edu/ikb/desktop-support/software-and-applications/anyware-desktop-first-time-set-up-how-to-access] - instruction



For those of us who use the Mac version, the icon palette is a little different but has the same functions.

Prerequisites

Knowledge

Since this is your initial foray into the digital design process, there aren't any knowledge prerequisites to complete this lab, except an understanding of number systems with respect to their radices and the functions of the basic logic gates.

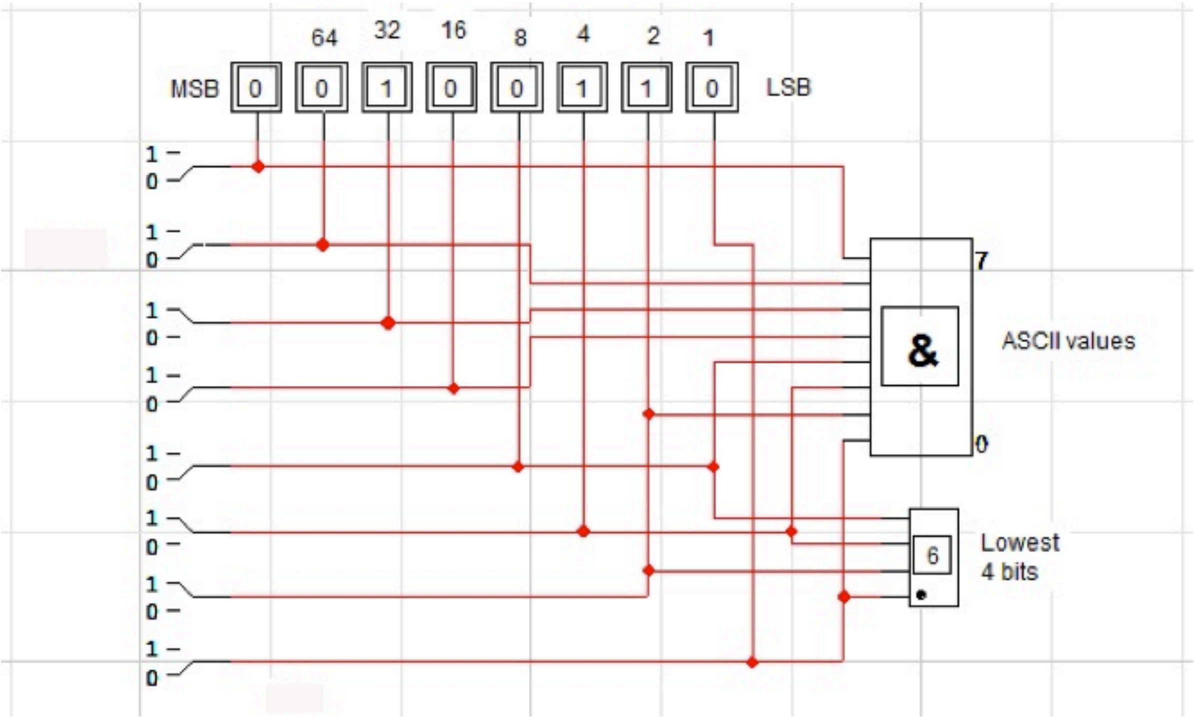
Software The only thing necessary is the LogicWorks simulation package and possibly access to a browser.

Objectives

You will learn how to manipulate signal lines, input switches, display devices (including hex and ASCII displays), basic gates, and labeling. You will experience exhaustive testing which is rarely possible in large systems incidentally.

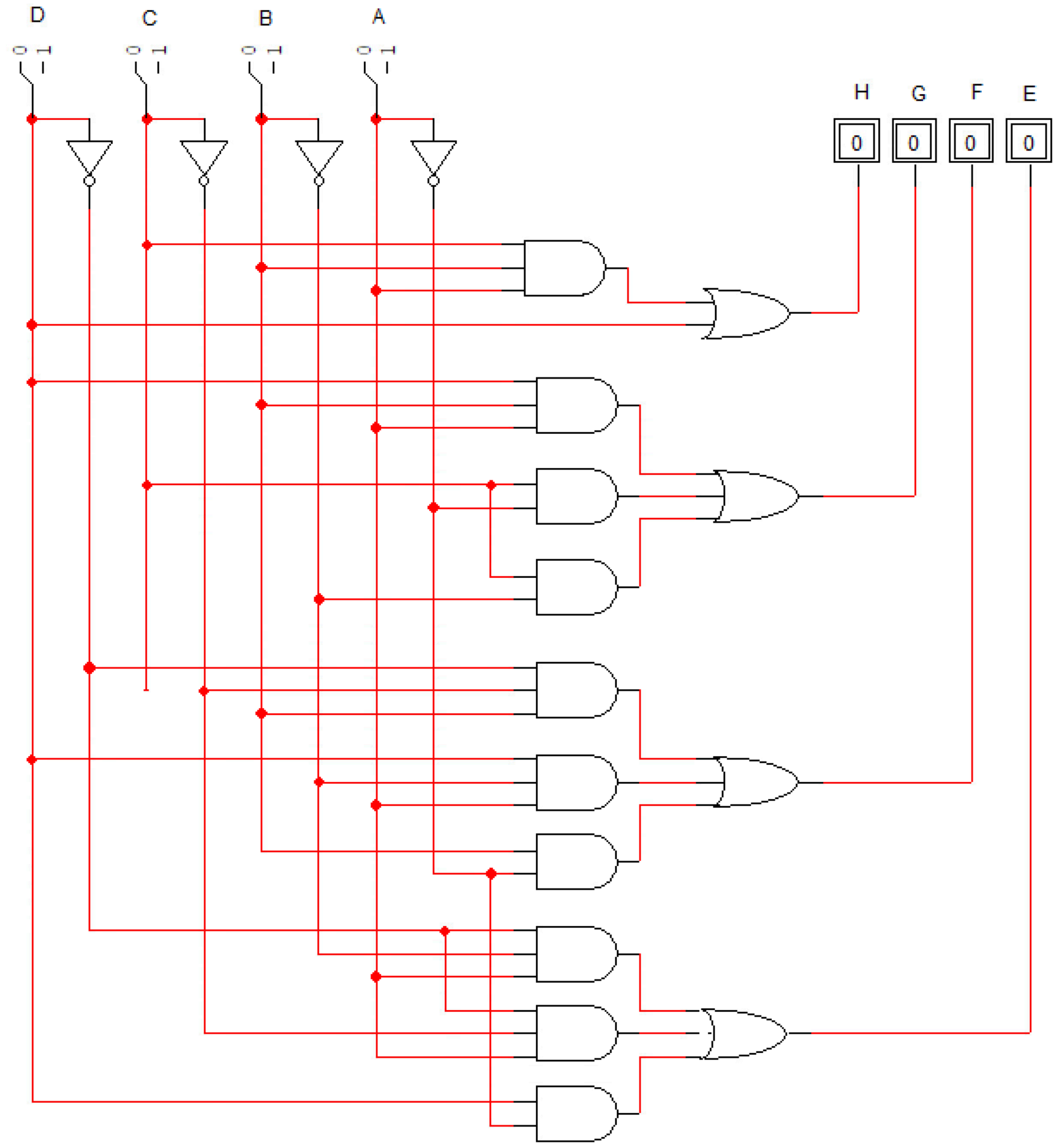
Exercise 1

The circuit below is an untidy example of one that shows the conversion of an 8-bit input to an ASCII value as well as showing the hex value of the lowest bits. MSB stands for most significant bit, LSB for least significant bit. Your task is to build a similar circuit taking care for tidiness that allows the circuit to be clearly analyzed or debugged. Troubleshooting is a significant skill that you need to develop as an engineer or computer scientist.



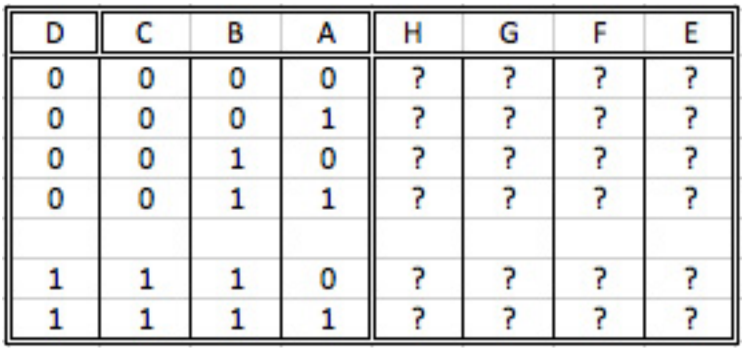
Exercise 2

The circuit below is a binary (DCBA) to 7421-weighted (HGFE) decoder. Your task is to create a circuit that performs the same as this one and to test it exhaustively. In other words there are 15 different combinations (0000 to 1110) of inputs that need to be checked. In this exercise you will use gates from the LogicWorks menu. You will need to develop a sense of tidiness again to be able to debug a faulty circuit path. One useful hint is to click on any line and it will show, in yellow, all other connections to it.



Testing

To test this circuit you should exhaustively try every possible input. Since this circuit is designed to convert a four-bit binary format to a different number system, only sixteen possible combinations normally exist. But in this case you cannot get 15_{10} in a weighted 7421 system. A recommendation for systematically testing your circuit is to start at '0' (0000) and count to '15' (1111). As you input each binary number, make a table as follows:



D	C	B	A	H	G	F	E
0	0	0	0	?	?	?	?
0	0	0	1	?	?	?	?
0	0	1	0	?	?	?	?
0	0	1	1	?	?	?	?
1	1	1	0	?	?	?	?
1	1	1	1	?	?	?	?

Deliverables

You should submit the answers to the following questions. Use HuskyCt to submit your file. A formal lab report is not required for this lab.

Reference

The 7421 code will be used when we consider the POSTNET system later in the semester.

Questions

What value is output on HGFE for DCBA = 1111?
Why do you use DCBA as the inputs rather than ABCD?