Computer Systems, Spring 2020

Week 2 Lab

Numbers and Logic Gates

This lab exercise is unassessed: there is nothing to hand in. It covers some basic concepts that will be used throughout the course, including the assessed quizzes, program and examination.

Problems to work out on paper

1. Convert the 8-bit word 0100 1101 to decimal, assuming binary representation.
2. Convert 89 to an 8-bit binary number.
3. Consider the 8-bit word 1001 0110.
   (a) Convert it to decimal, assuming binary representation.
   (b) Convert it to decimal, assuming two's complement representation.
   (c) Explain, in your own words, why a word (such as 1001 0110) has no inherent meaning, but it has multiple meanings corresponding to different number systems (e.g. binary, two's complement, and others that we haven't discussed).
4. Calculate 23+59 by converting both numbers to 8-bit binary numbers and doing binary addition. Check the result by converting back to decimal.
5. State whether each of the following conversions is possible, and give the reason it is or isn't possible. (But you don't have to do the actual conversion!)
   (a) Convert 12 to a 4-bit binary number.
   (b) Convert 12 to a 4-bit two's complement number.
   (c) Convert 73 to a 5-bit binary number.
   (d) Convert -5 to an 8-bit binary number.
   (e) Convert -128 to an 8-bit two's complement number.
   (f) Convert 128 to an 8-bit two's complement number.
6. (Optional) This experiment is a small introduction to research—see if you can make a surprising discovery! Recall that (1) we have seen in the lectures how to add binary numbers; (2) we have seen that binary and two's complement are different number systems; and (3) we have *not* discussed in lectures how to add two's complement, or how to subtract anything at all. Now, try the following. (In some cases you may observe a carry of 1 from the leftmost position—if that happens simply ignore the carry and just consider the 8-bit sum result you get.)
   (a) Convert 95 and -30 to 8-bit two's complement numbers.

(b) Add these 8-bit words using the method for binary addition, *even though they are NOT binary numbers.*

(c) Convert the result back to decimal. What do you observe?

(d) Try the same thing with some different values, such as $24-59$, $-9+28$, and other examples you make up.

Do your experiments suggest a hypothesis? Why is the hypothesis surprising? Can you think of any practical applications?

7. Write down the truth tables for the following logic gates: inv, and2, or2, xor2. You need to know these!

Problems using the computer

The aims are to learn how to use a "schematic capture" software tool for drawing a digital circuit and simulating it, and to learn how to design and simulate simple combinational circuits using basic logic gates.

You are encouraged to work with friends, and to receive and offer help. If you're not sure how to get the software to work, ask your tutor or your neighbours for advice. You don't have to create anything to hand in; what's important is that you understand how circuit simulation works.

Download LogicSim from Moodle and unzip it. Enter the Windows folder and click on LogicSim.exe.

1. Create a little circuit that contains an Input, an inverter, and an Output. Follow these steps:

   (a) Click Not and you'll see an inverter in the upper left. Drag the Inverter to somewhere around the middle of the canvas. (You can resize the window to make it bigger.)

   (b) Click Output, and drag the box somewhere to the right and below the inverter (leave some clear space between the components.)

   (c) Click the little greenish circle to the right of the inverter, then click the little circuit on the output. This will create a wire connecting them.

   (d) Click Input, and drag the box somewhere to the left of the inverter.

   (e) Create a wire connecting the input box to the input port of the inverter.

   (f) So far we have been drawing and editing the circuit; we need to change mode to simulate it. Click Start Simulation

   (g) Notice that the Input value is 0, but the Output is U. The reason for this is that it takes time (a gate delay) for the inverter to put the right value on its output. Click Gate Delay and you'll see that the inverter is now producing a 1.

   (h) Click the Input box; this will toggle it from 0 to 1. And the output is again U. Click Gate Delay again, and the Output becomes 0.

(i) Try clicking the Input again. Notice that the status of each wire is indicated by its colour: a blue wire is 0, a red wire is 1, and a grey wire is Invalid. As you click Gate Delay, the effect is that grey wires will change to blue or red.

2. Now create a circuit that contains an and2 gate and connect it to two inputs. Again, simulate the circuit. Try it with all four combinations of inputs 00, 01, 10, 11 and write down a truth table showing the results.

3. Test the or2 and xor2 logic gates the same way. In each case, build a truth table using the simulator, and compare it with the truth table given in the lecture notes.

4. Test the half adder circuit. Click Load and then click halfAdd. Generate a truth table for the circuit (just as you did for the logic gates). Check that the half adder is working correctly.

5. Do the same for the full adder circuit: load it, simulate it, and check that it's working correctly.

6. Do the same for the multiplexer circuit mux1 (click Load, and the circuit is in a subfolder called Reusables).