Lab 2: Data Representations & Transistors

Due Date: Thursday 2/9/2017 11:59PM

This lab covers the material in lectures three and four.

If you’re unsure about a result, be sure to explain your thinking. In general, you should show your work. A well-reasoned but wrong answer (for example with a minor error) will receive more points than a wrong answer with no context at all.

Problems (50 points total)

1. Suppose I have a 8-bit signed, 2’s complement integer -23, which would be represented as 11101001₂ in binary.
   a) What would the binary representation be if I sign extended (SEXT) this number to 16 bits?

   Should be 1111111111101001₂.

   b) Taking the original 8-bit number, add −110₁₀ (again, 8 bits) to it in binary. Is this an overflow? How do you know?

   −110₁₀ is represented in 8-bit binary as 10010010₂. When we add −23 to this we end up with 01111011₂, which is +123. This switch from negatives to a positive tells us we have overflowed.

2. Convert the following decimal numbers to 8-bit, unsigned binary integers. Show your work.
   a) 127

   01111111₂.

   b) 68

   01000100₂.

   c) 17

   00010001₂.
3. Write the following in hexadecimal notation. Use 4 hexadecimal digits (recall this corresponds to 16 binary digits). For example, \(10_{10}\) or \(1010_2\) would be written as \(0x000A\).

a) \(63_{10}\)

\(0x003F\).

b) \(00101011100110_2\)

\(0x2AE6\).

c) \(-23_{10}\) (HINT: start with the signed 2's complement representation in binary)

\(0xFFE9\).

4. Assume a raw digital image with a resolution of 1024x768. Also assume 8-bit color depth and that we’re using a 3-channel (RGB) scheme to represent pixels.

a) Ignoring image metadata, how big will an image be in bytes?

\[1024 \times 768 \times 3 = 2359296 \text{ bytes (2MB)}.\]

b) How many unique colors can we represent with one pixel?

\(2^{24} \ (16 \text{ million}).\)

5. In a CMOS transistor circuit, which transistor type is always connected to ground? Which is always connected to \(V_{dd}\)?

\(nMOS \text{ connects to ground. } pMOS \text{ ties to } V_{dd}.\)

6. When designing integrated circuits in practice, one must take into account the fact that real-world components are not ideal. That is, they do not behave in the perfect way we’d imagine them to in our logical circuit diagrams. One real-world imperfection is the attenuation or weakening of signals on long wires. If you try to drive a wire from one end of a microprocessor to the other, the signal will be pretty much useless by the time it reaches the other end. To fix this, we use the fact that transistors are active devices, i.e. they get a “fresh” output signal from \(V_{dd}\), which means we can use them to effectively amplify their input. Our first stab would be to put an inverter on the wire, which will amplify the input, but it has the unfortunate property that it also inverts the signal on the other end! Draw a transistor circuit such that at the end of the wire we end up with the original (non-inverted) input. We call this a digital repeater.
7. Assume CMOS transistors again.
   
   a) Write out the truth table for a 3-input NOR.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>NOR(A,B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

   b) Draw a CMOS transistor circuit that implements it. Note the total number of transistors you use.

   c) Now write out a truth table for a 3-input OR.
d) Draw the CMOS transistor circuit for this truth table. Don’t forget to adhere to the pMOS/nMOS connection rules! **Hint: you’ll need to add inverters to inputs.**

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>OR(A,B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```

e) Instead of the above, draw an OR circuit that connects an inverter to the output of your previous NOR.
f) How many transistors have you saved by using the inverted NOR gate?

You should have gone from a 12T to an 8T circuit, thereby saving 4 transistors.


Anything along the lines of “The number of transistors that can fit on a chip will double every 18 months” or “The performance of chips will double every 2 years” is fine here. Whatever trend, it should look like a straight line on a log plot. Either way it is not continuing at this pace anymore!

9. In a transistor, current flows from the ____ to the ____.

From the source to the drain.

10. Under what condition will the transistor allow current to flow? Put another way, under what condition will a transistor form a closed circuit?

When \( V_g \) (Gate voltage or input voltage) is greater than \( V_t \) (threshold voltage).

Coding Assignment (50 points total)

Get the lab2 code by logging into fourier and running the following:

```
$> cp /home/khale/HANDOUT/lab2.tgz .
$> tar xvzf lab2.tgz
$> cd l2
$> make
$> gcc -Wall -std=c99 -lm -o lab2 lab2.c
$> ls
examples lab2 lab2.c Makefile test_harness
```

You’ll see a few extra files this time. You can see how to run the lab2 program by running it without any arguments:

```
$> ./lab2
```

Your job here is to fill in the functions in lab2.c which have FILL ME IN comments written in them. See the comments in the code for further instructions. I’ve also included a test harness for you to test your code. You can run it like so:
Notice that the code will initially fail all the test cases. Your job is to make it pass them all. These are example cases, and I’ve included them in a separate file called examples. You are free to modify this file and add your own tests. The format is:

[option] [input] [expected output]

For example, to test my program with the \(-f\) flag (which invokes the bit flipping routine), I can include a line like this:

\[
f \ 0x0000 \ 0xffff
\]

And it will add another test. You can then rerun make test to see whether or not your program passes it. Note that we will be using different test cases than the ones in this example file, so make sure to add your own tests to convince yourself that your code works!

DO NOT MODIFY THE test_harness SCRIPT! Or, do so at your own risk.

Hand-in Instructions

Make sure to put your name on your submission. Submissions without names will be given zero points! For code, this means put a comment at the top of your C file with your name on it.

For the code, you must hand it in digitally. I’ve made this a bit easier this time. Once you’re happy with your code, in the directory where your code is, run the following:

\[
>$> \text{make handin}
\]

For the problems, the following still apply:

**Physical** : If you’re submitting a written copy, hand it to one of the TAs or to the instructor. You can also leave it in the instructor’s mailbox in the CS department office, but make sure to get it time stamped when you do (see the “Submitting Work” section of the syllabus).
Digital: If you would like to submit an electronic copy, note that I will only accept PDF files (no Word docs please). Again, see the “Submitting Work” section of the syllabus. Please do not take a poorly lit picture of your assignment. Your grade will suffer commensurately with our inability to read your work. Once you have a PDF, you should submit it on fourier. You should name your file yourid-lab2.pdf where yourid is the thing in front of the @hawk.iit.edu in your e-mail address.

You can first get your PDF (for example, for me it might be called kh123-lab2.pdf) onto fourier like so:

[me@mylocalmachine]$ scp kh123-lab2.pdf kh123@fourier.cs.iit.edu:

Then you can login to fourier via ssh and submit it:

[kh123@fourier]$ cp kh123-lab2.pdf /home/khale/HANDIN/lab2