Lab 4: Combinational Logic, Storage, and C Pointers & Structs

Due Date: Thursday 2/23/2017 11:59PM

This lab covers the material in lectures 7 and 8, with additional C-centric material that will help you in subsequent labs. You will want to leverage your TAs for this one, as in the lab session they will be introducing you to the concepts of pointers and structs.

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 (100 points total)

1. Draw an 8 input multiplexer (MUX) using AND and OR gates. There will only be one output!
2. I have an 8x8 matrix of LEDs that I’d like to use as a monochromatic (one color) display. This is typically called a dot matrix display. Ours will be a very limited display, in that it can only light up one “dot” at a time. I’d like to be able to turn a particular LED on based on a 2-tuple that consists of a row number and a column number \((r, c)\). All I care about is the dot selection logic, i.e. given two numbers \(r\) and \(c\) select one dot in the matrix. You do not need to worry about how the LED is wired etc. Show how I can accomplish this using two decoders. **Hint:** remember how the DRAM chip I talked about in lecture was organized. You do not need to draw this at the level of gates, i.e. you can treat decoders as black boxes with \(n\) inputs and \(2^n\) outputs.
3. I want to try to build a storage element using the following truth table:

<table>
<thead>
<tr>
<th>Input</th>
<th>Current Output</th>
<th>Next Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

a) Draw this circuit. **Hint:** you only need one gate.
b) Will this circuit work as a storage element? Why or why not?

No, the output is unstable (it will oscillate)

4. A 4-2 multiplexer takes 4 bits of data input \( X[0 : 3] \) and uses 1 bit of selector input \( S \) to produce 2 bits of output \( Y[0 : 1] \). If \( S = 0 \), then \( Y_0 = X_0 \) and \( Y_1 = X_1 \); if \( S = 1 \), then \( Y_0 = X_2 \) and \( Y_1 = X_3 \). Implement such a multiplexer using two 2-1 multiplexers.

5. Write some C declarations and code to establish the memory diagram below (there is more than one correct answer). \( p, q, r, \) and \( s \) should be pointers to integers.
One potential answer:

```c
int b[5] = {8, 3, 2, 7, -9};
int *s = b;
int *p = s + 1;
int *q = s + 2;
int *r = q;
```

6. Using the memory diagram from Problem 5, answer the following question for each of the expressions below: Does it cause a compile-time warning or error (and if so, which one), or does it cause a runtime error (if so, which one), or does it evaluate to true or false (1 or 0)? **Hint:** write your answer to Problem 5 as a program, then try adding these expressions and compiling and running them.

a) \( p < q < r \)

*Warning:* cannot compare integer with pointer. Note this is because the first expression evaluates to an integer!

b) \( s != q && *q == (*s >> 2) \)

*True*

c) \( q-b == &q[1] - &s[0] \)

*False*

d) \( s[1] == q[-1] \)

*True*

e) \( *r == *q \)

*True*
7. Consider the C declarations and code below.

```c
int b[4] = {12, 13, 14, 15};
int u = 20, v = 30, *x = &u, *y, *z;
y = &u;
z = &b[2];
// <------ position 1
++ *x; // (i.e., *x = *x + 1)
y = &v;
--z;
z[1] = 20;
// <------ position 2
```

a) Draw a memory diagram that shows the state at position 1.

![Memory Diagram for Position 1](image1)

b) Draw a memory diagram that shows the state at position 2.

![Memory Diagram for Position 2](image2)

8. Consider the C declarations and code below. Draw a memory diagram that shows the state at position 1. It is a bit tricky but try to label all the different parts of the struct array fields (x[0].a, x[0].b[0], ..., x[0].str, x[1].a, ...).
struct foo {
    int a;
    int b[3];
    char *str;
};

struct foo x[2], *p = NULL;
char *s1 = "hello";
int i;
for (i = 0; i < 3; i++) {
    x[0].b[i] = i*i;
    x[1].b[i] = -i;
}
p = &x[0];
p->a = 12;
p->str = s1;
p++;
p->a = -23;
p->str = s1;
int *q = &x[1].a;
// <------ position 1
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.

**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-lab4.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-lab4.pdf) onto fourier like so:

[me@mylocalmachine]$ scp kh123-lab4.pdf kh123@fourier.cs.iit.edu:

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

[kh123@fourier]$ cp kh123-lab4.pdf /home/khale/HANDIN/lab4