Name: ______________________________

# Computer Science 352

Fall 2016 Final

Doodle here.

This is a one-person exam. Please do not discuss your work with any other student. Do not solicit or share solutions. Failure to do so will result in the consequence that someone who did likewise will be responsible for building the software that navigates your self-driving car.

1. *Assembly Instructions*

   (a) Write in ARM assembly a function `bigger2`, which accepts a standard 4-byte integer as its only parameter. Return the nearest power of 2 greater than the parameter. For example, `bigger2(10)` $\rightarrow$ 16, `bigger2(32)` $\rightarrow$ 64, `bigger2(5)` $\rightarrow$ 8, and `bigger2(15000)` $\rightarrow$ 16384.

   Consider what a power of 2 looks like in binary. Use the `CLZ rd, rs` instruction in your solution. It counts the number of leading zeroes in `rs` and stores the result in `rd`. This can be solved in 6 instructions or fewer.

   (b) Write in ARM assembly a function `ispow2`, which accepts a standard 4-byte integer as its only parameter. Return a 1 if the parameter is a power of 2, and 0 other otherwise. For example, `ispow2(10)` $\rightarrow$ 0, `ispow2(32)` $\rightarrow$ 1, and `ispow2(5)` $\rightarrow$ 0. Assume the parameter is not 0.

   Consider what a power of 2 looks like in binary. Consider also what a number one less than a power of two looks like in binary. What can you say about the corresponding bits between a power of 2 and its predecessor? Consider the parameter and its predecessor to determine whether or not the parameter is a power of 2. This can be solved in 6 instructions or fewer.

(c) Write in ARM assembly a function `ceil2`, which accepts a standard 4-byte integer as its only parameter. Return the nearest power of 2 **greater than or equal** to the parameter. For example, `ceil2(10)` $\rightarrow$ 16, `ceil2(32)` $\rightarrow$ 32, `ceil2(5)` $\rightarrow$ 8, and `ceil2(64)` $\rightarrow$ 64. Assume the parameter is not 0.

Use your helper functions to solve this.

2. *The Plurality Is Near*

   (a) When the University of Wisconsin–Water Street opened, it did not have enough faculty to teach both lower- and upper-division courses. So, it only matriculated one class of 3000 students at a time. That class would go through its first, second, third, and fourth years without any new students coming to campus and with the same small faculty teaching all their courses. The class would graduate, and a new class of students enrolled. This lasted for 20 years. What was the latency of the original UWWS educational program? How many students graduated? Show your work and units.

   (b) A casino was built in the basement of the Pickle Student Union, and the university was able to hire enough faculty that a new class of 3000 students could matriculate each year—without disrupting the schedule of the students already enrolled. This pipelining started just after the last one-at-a-time class graduated. What was the latency of their new system? How many students fully graduated in the first 20 years of this new system? Show your work and units.

3. *Fine for Not Parking Here*

   You are at the edge of campus. If you drive directly to and park in the Phillips Lot, it takes 5 minutes to park and get to your first class. If there's no parking space, you try the Hibbard lot. If you find a spot there, it takes a total of 15 minutes to get to class—with some of that time burned by searching the Phillips lot. If there's no spot in the Hibbard lot, you try Siberia, which is what students call the lot on the frozen Chippewa River. If you have to fall back to Siberia, it will take you 54 minutes to get to class. (The dogsleds help.)

   (a) Suppose 40% of the time there's a space available in the Phillips lot, and 70% of the time you can park in Hibbard. How much time should you expect to take to get to class? Show your work.

   (b) Suppose the Phillips lot can either by expanded so that 50% of the time you can find a space or its hit time can be reduced to 2 minutes. Which modification leads to a better mean-time-to-class? Show your work.

4. *Three Ways to Not Be There*

   Define in your own words the following caching terms. Also explain how each can be avoided, if possible.

   (a) Conflict miss

   (b) Compulsory miss

   (c) Capacity miss

5. *Clunks for Cachers*

   Consider the three caching strategies we discussed in lecture. As in lecture, suppose blocks from main memory are 10 bytes in size and that our cache has 8 lines. When you read byte X, its surrounding block is drawn into cache and placed in a cache line determined by the caching strategy. For each strategy, provide a sequence of byte addresses that will demonstrate poor caching—such that *each and every* request will be a miss. Include at least 16 requests in each sequence and minimize the number of compulsory misses.

   (a) Fully associative:

```
blockID = address / blockSize

for each line in lines
  if line.tag == blockID
    return lines[address % blockSize]   # A hit! :)

# A miss. :(
if cache is full
  kick out the least-recently-used line

retrieve block from memory
place it in random empty line
return line[address % blockSize]
```

   (b) Direct mapped:

```
blockID = address / blockSize
lineID = blockID % nlines

if lines[lineID].tag != blockID
  retrieve block from memory
  place it in lines[lineID]

return lines[lineID][address % blockSize]
```

(c) 4-way set associative (meaning 2 sets of 4 lines each):

```
blockID = address / blockSize
setID = blockID % nsets

for each line in set
  if line.tag == blockID
    return line[address % blockSize]   # A hit! :)

# A miss. :(
if set is full
  kick out the least-recently-used line

retrieve block from memory
place it in random empty line within set
return line[address % blockSize]
```