

CprE 381 Homework 0

[Note: This homework first covers some of the background material that you are expected to know from CprE281 and then covers material from the first few class periods.]

1. Review

- a. I want to select one of 6 different 2-bit inputs and output it. How many bits do I need to specify which input? Draw an implementation of this using the minimum number of 2:1 Bus MUXes and no other logic.
- b. Use one appropriately sized decoder and a small number of AND, OR, and NOT gates to implement the functionality from 1.a.
- c. Design a circuit using AND, OR, and NOT gates that takes a 4-bit two's complement number as input and outputs whether the number is positive (p), negative (n), or zero (z).
- d. How many bits are required to encode the number of ps in a minute?
- e. Convert 0x38157EE9 into binary and decimal. You must show each step to receive credit.

2. Introduction to computers

- a. Using your favorite computer as an example (cannot be the Amazon Dash button or the Blue Waters supercomputer), map at least 8 different subcomponents (e.g., chips and antennas) into one of the five main components of a computer presented in the textbook or pre-recorded lecture videos. Use online descriptions, teardowns, and component data sheets to help you with this task, but cite your sources.
- b. Look up one stored program computer from each of the last 9 decades (1940's-2020's) and report its primary/main memory capacity (in B and then the most appropriate term such as Gibibyte). Do NOT list the same computers as your friends, although you may work together to find multiple computers per decade. Also do NOT use a single source for the entire list (this is cheating and not helpful for your learning)—you must CITE the source for *each* computer. For extra enrichment (i.e., not graded): See if you can find the physical size and cost of the systems in today's (i.e., 2023) dollar.
- c. Implement a program in C to calculate the average of an array of N integers (you may assume `int`-sized values whose initialization values are up to your testing approach). (1) Report the time it takes you to (correctly) write and test this program. (2) Report the program size in number of characters. Compile the program to an executable object file. (3) Report the executable size and the architecture for which you compiled it. Finally, run the program and (4) report the time it takes to execute for 3 different test arrays that differ in size by orders of magnitude. Include the program files (C language and executable object code) in a tarball (HW00_2c.tgz).

3. MS Teams

- a. Login to MS Teams and post an interesting, non-technical fact about yourself or your interests in the random channel. Use this channel for any vaguely relevant discussions (e.g., new chips or computing paradigms) or just course comradery posts.

Extra credit: Start and/or contribute to a 2050 Coover playlist integrated into MS Teams.