CSCI 2321 (Computer Design), Spring 2020

Review for Exam 2

The exam will be conducted as a timed take-home exam, limited to 2 hours. I will have several two-hour "office hours" sessions, starting Thursday May 7 and ending Tuesday May 12, and if you work during one of those I will be available for questions; otherwise I may not be. In any case you must turn in the exam by 11:59pm Tuesday May 12.

I'll put a PDF of the exam in the shared Google-Docs folder for the course shortly before the start time(s). You will create a PDF containing your answers. It can be an edited copy of my PDF (if you have software to do that) or something generated by a word processor. You will turn this in via e-mail or by placing it in your "graded work" folder. *Note* that based on what happened with Exam 1, trying to edit my PDF (or to copy it in) often produces pretty ugly results. I'd rather have just a PDF with your answers! but it's up to you.

Like the quizzes and Exam 1, it's "open book / open notes", which means you can consult paper or electronic copies of the textbook and your notes, sample solutions *from this year only*, your own graded work, and anything on the course Web site. You may not use other books, materials from this course from previous years, a calculator or computer (except as needed to consult allowed sources), or (of course) each other's papers.

Questions will mostly be similar in format to the ones in quizzes, non-opinion minute essays, and homeworks; difficulty/length will mostly be somewhere between quiz questions and homework problems. There will also likely be a few multiple-choice or true/false questions.

1 Lecture topics to review

You are responsible for all material covered in class or in the assigned reading from Chapters 3 and 4 and Appendix B of the textbook. You should review in particular the following topics. It would probably also be helpful to review sample solutions for the quizzes, assignments, and any minute essays that have well-defined answers.

- Binary, decimal, and hexadecimal number systems; two's complement notation.
- Floating-point representation and some of its limitations.
- Combinational-logic blocks versus state elements; the purpose/role of clocking.
- AND and OR gates and inverters, and how to use them to implement Boolean functions.
- Design of an ALU, as discussed in Appendix B.
- Design of a datapath for a single-cycle implementation of our selected subset of MIPS instructions: what elements are needed, how to connect them, what control signals are needed.
- Generating control signals for our single-cycle implementation — what elements we need (two combinational-logic blocks), their inputs and outputs, how outputs depend on inputs (expressed via truth tables and/or Boolean functions).
- How the completed single-cycle implementation (datapath and control) executes example instructions.
- Why a single-cycle implementation isn't really practical.

- A little about pipelining — how it can help, the basic idea (assembly-line analogy), what makes it tricky (the "hazards"), basic idea of how it's implemented (divide single-cycle design into stages and add "pipeline registers").

2 Reading to review

You should have read, or at least skimmed, all of the assigned reading from Chapters 3 and 4 and Appendix B, but the focus will be on material presented or at least mentioned in class.