463-1 Fall 2017   REVIEW SHEET FOR EXAM #1

About the exam:

The exam will include conceptual questions and problems to work.

Most or all of the exam will be graded by Scantron, so make sure to bring a pencil.

Also remember that according to the syllabus, there are no makeup exams.

The conceptual questions below are a complete guide to the conceptual part of the exam. However, the format will probably be different. The exam may include any of the following: multiple choice, short answer, true/false and similar types of questions.

The homework problems are your best guide to the problem section of the exam. *The most important thing you can do to study for this portion of the exam is to review the questions you got wrong.* After your TA has graded the homework, I will post the answers on Scribd for you to check against.

Material that is not on the homework or the review sheet will not appear on the exam, i.e., there are a few things in the slides that are just background information for you.

You will need to know the black box representation (i.e., the inputs and outputs) of the components we have studied. You will not be asked to draw the internals of any component; however, you will find it important to be able to recognize them.

You will need to be able to read circuit diagrams, e.g., diagrams like 3e-37b without the key (xy, etc.) on the right-hand side. You should understand how each circuit we have studied works.

You may need to draw simple circuits such as the ones in the homework.

I will give you a chart of Boolean identities and a table of powers of 2.

No calculators or other electronics will be permitted on this exam, as there will be no calculations. If future exams have calculations, simple four-function or scientific calculators such as those permitted on the ACT and AP exams will be permitted at that time. Using a cell phone or other device as a calculator will *not* be permitted under any circumstances.

Part I. Conceptual questions on history

1. What is a stored program computer? Who was Atanasoff? Who were Mauchley and Eckert?

2. Who was von Neumann? What is the von Neumann architecture? What are its major components? What is the CPU? What is the ALU?

3. What happens in each step of the fetch/decode/execute cycle?

4. Why were transistors an improvement over vacuum tubes?

5. Integrated circuits were an improvement over transistors because they took up less space. What is a VLSI chip? What does VLSI stand for? What significant development in computer design did VLSI make possible?

6. What business decision(s) did IBM make with regard to architecture that made the company so successful?

7. Backward compatibility is the ability to run old programs on new hardware. The modern software industry would not be possible without backwards compatible design because users want to spend their money on their core business, not on rewriting or converting software. Name one or two hardware families where backward compatibility has significantly influenced design.

8. What is Moore's Law? Why can't it hold true forever?

Part II: Gates

1. Draw the symbols for the following gates: A AND B, A OR B, NOT A, A XOR B, A NAND B, A NOR B. Give the truth table for each gate.

2. Make sure you know the order of operations for Boolean expressions and the various symbols that can be used.

3. How many rows are there in a truth table for an expression with n variables? What does it mean when there is a 1 in the result column?

4. What is sum-of-products form? Why is it useful? If an expression has n variables, how many variables will be in each term of the sum-of-products form? What is the maximum number of terms there could be? How can you derive the sum-of-products form from the truth table?

5. Why are Boolean identities useful in hardware design?

6. What is the dual of an expression? Why is it useful?

7. What is a universal gate? Which gates are universal gates? Why are universal gates useful?

Part III: Combinational (combinatorial) circuits

1. What is a half adder? How many bits can a half adder add? Show a black box representation.

2. What is a full adder? How is it different from a half adder? Show a black box representation of a full adder. Show how to implement a full adder with two half adders.

3. Show how to implement an adder for 2-bit binary numbers using two full adders.

4. How many full adders would it take to add two 16-bit binary numbers? If you used one half adder, how many full adders would you need? Which column would you use the half adder for?

5. What are the inputs and outputs of a ripple-carry adder? (i.e., draw a black box representation). What does a ripple-carry adder do? What components is it made of?

6. What does a decoder do? Draw a black box diagram. What is it useful for?

7. If you have 8-bit addresses, how many input lines does your decoder need? How many output lines? If the input is 0000 0110, which output line(s) will be activated?

8. What does a multiplexer do? Draw a black box diagram.

9. If a multiplexer has 8 selection lines (control lines), how many input lines and how many output lines can it have?

10. What does a shifter do? If a shifter has 8 input lines, how many output lines will it have? How many control lines? What do(es) the control line(s) indicate?

11. What is an ALU? What kinds of operations does it typically handle?

12. If an ALU can handle 16 operations on 32-bit numbers, how many control lines does it need? How many data lines does it need for each input? How many data lines does it need for the output?

Part IV. Sequential circuits

1. What is the difference between a sequential and a combinatorial circuit? Why does a sequential circuit need a clock line?

2. What is a clock line? What is the relationship between the clock and state changes in a sequential circuit?

3. What is the rising edge? falling edge? edge-triggered circuit? level-triggered circuit? What type of circuits are the ones in this course?

4. What is feedback? Why is it useful?

5. What is an SR flipflop? What is a JK flipflop? Why is the JK flipflop an improvement over the SR flipflop?

6. What is a D flipflop? Why is it useful?

7. What is a characteristic table? How is it different from a truth table?

8. Which type of flipflop is the basic building block of a register? Why? How many are needed to build an n-bit register?

9. How many selector lines are needed for a 2^n word memory? What is a write enable bit? When is it on? How many flipflops are needed for a 2^n word memory where every word contains m bits?

10. How many flipflops are needed to build an n-bit counter? What is the smallest number this counter can represent? The largest? What does the count enable line do?