Review for Second MidTerm

Exam guidelines

The second midterm will be Wednesday May 29th. You may bring one single-sided sheet of handwritten notes on US Letter paper. There will be no calculators allowed. There will be three or four questions taken from the topics described below, although knowledge of any of the topics covered in class is expected. Please see the review for the first midterm as well.

Flip-flops and Counters

Know the truth tables for the operation of “Set-Reset,” “D-type,” and “J-K” flip-flops. Understand the difference between “jam-type”, level-sensing (gated), and edge-triggered flip-flops, and know how and when the outputs change as a function of the inputs.

Know how to combine flip-flops to form useful circuits, such as:

- “debouncers”
- ripple counters
- shift registers

Be sure to understand the difference between synchronous and asynchronous circuits. Be able to analyze a synchronous or asynchronous circuit, and be prepared to draw a timing diagram. Know how to use flip-flops and counters to implement simple circuits which do not need a full-blown state machine. Examples might be stopwatches, gated timer circuits (start/stop logic), camera shutters, etc. Anything which needs a trigger can generally be controlled effectively by one or two flip-flops.

Synchronous Logic

Understand how synchronous logic can be used to implement a functional circuit using Moore-style state machine logic. Know the state machine design methodology and be able to implement the needed logic using D-type flip flops or registers using Karnaugh maps. Understand the functional difference between state machine input/output lines and the internal state variables held in the register. Understand the architecture of a Moore-style machine, and realize that the outputs are only a function of the current state. Review the state machine examples we have covered in class and on the homework. I will not ask you to design a state machine using J-K flip flops.
Possibilities for the midterm include:

- modulo-n counters,
- up/down counters,
- pulse-train generators,
- pattern-matching logic,
- stop-watch logic,
- ...

Understand how a counter can be used to implement simple sequential state machines, like the stepper motor controller. Also, understand how state machines can be interfaced to external devices like counters or display circuits to make more complex systems.

**Analog to Digital Convertors**

Review the design of the two major ADC types covered in class: the flash and successive approximation ADCs. Be able to design a simple ADC circuit from component parts, and understand which pieces limit the ultimate performance. In particular, understand how comparators work conceptually, know how the comparator outputs are used to get a digital output in both the flash and SAR ADC examples.

**Programmable Logic**

Understand how an EEPROM can be used to implement a truth table, and be able to specify EEPROM address locations using hex values. Understand conceptually how Verilog implements different logic units and defines the input and output lines for each block. Be able to make a schematic sketch of the logic implemented in a Verilog file. You will not need to write any Verilog code, but you may be asked to analyze a simple Verilog file and describe conceptually what it does.

I will not ask you any questions about CUPL or ask you to implement something using a PLD (Programmable Logic Device).