3827 OH

Eumin Hong (eh2890)

Columbia University

March 1, 2022

Overview

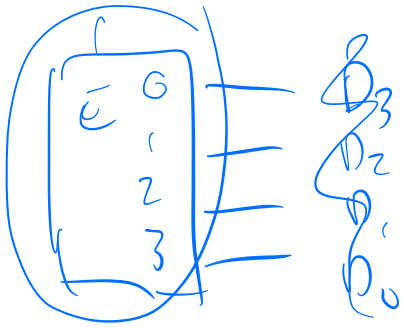
Announcements

Announcements: Upcoming Exams and Homework

- Midterm is March 10 or 11 (depends on section/form you filled out at the beginning of the semester)
- HW3 graded, will have HW4 graded ideally over the weekend (after all submissions received)

Announcements: Homework 3 Feedback

- Common errors from Homework 3:
  - Be sure to label the inputs/outputs of your decoders, MUXes, and similar circuitry (better safe than sorry)
  - Use "don't cares" in Q3 – whenever you have various possible inputs combinations for the same output, think about using "don't cares" (i.e. 10 and 11 both result in red lamp color)
  - For questions like Q4/Q5, try creating some additional example strings
    - Exam will not provide examples of all edge cases (but the behavior will be well-defined)



multiple inputs → same output
- input "don't cases"

combination of inputs is not used → output "don't cares"

Announcements: Feedback

- Form: `https://forms.gle/cnUmKVNYN7WvRbHA6`

Midterm Review

Midterm Review: Overview

- The following slides list out the topics and techniques for notable homework questions and all exam questions
  - Not going to talk about Warmup Problems – those are pretty straightforward and covered in other problems
- You should know how to solve all of the HW problems and exam problems
- All references to lectures and slides are based on the GitHub repository (updated as of March 1)

Midterm Review: Homework 1

- All conversions to and from binary should be second nature
- Q3: should know how to correctly detect overflow for both binary and 2's-complement addition algorithms (Lecture 01, Slides 67-68)
  - 
- Q5: should know how to compare numbers (in all forms of binary, but the structure of floating point numbers makes it easy to compare) (Lecture 01, Slide 79)

Midterm Review: Homework 2

- Q1/Q2: should be able to simplify to minimize literals (Lecture 02, Slides 16-30)
  - Should also be second nature (except detecting XORs/XNORs, which take more time)
  - If stuck, try using K-maps
- Q3/Q4/Q5: De Morgan's Law (Lecture 02, Slides 22-29)
- Q6/Q7/Q8: Boolean expressions/minterms to K-maps to Boolean expressions (Lecture 03, Slides 32-67)
  - "Don't cares" (Lecture 03, Slides 71-81)
  - Filling out a K-map should take no longer than 30 seconds (rough estimate)
  - Practice – use friends or online resources to check your work
    - Online K-map resource: `http://www.32x8.com/index.html`

Midterm Review: Homework 3

- Q1: using smaller combinational circuitry to build larger combinational circuitry (Lecture 04, Slides 52-97)
- Q2: meaning of minterms (Lecture 03, Slides 16-18)
- Q3/Q4/Q5: word problem to truth table/K-map to Boolean expressions (Lecture 04, Slides 98-104)
  - Verify that you know what your inputs and outputs are (e.g. inputs are $ABCD$, outputs are NSH, NSL, EWH, EWL

Midterm Review: Homework 4

- Q2/Q3: word problem to truth table/K-map to Boolean expressions for FSMs (Lecture 06, Slides 24-42 and 43-71)

Midterm Review: Spring 18 Midterm

- Q1: word problem to truth table/K-map to Boolean expressions (Lecture 04, Slides 98-104)
  - If input is "unused", think "don't care"
- Q2: unsigned binary comparison
  - Q2b: using pre-defined circuits to implement more complex circuit (means that this is independent of implementation in Q2a for base case)
  - Q2c: MUXes to handle casework
  - Q2d: 2's complement binary comparison
  - Q2e: similar to ripple carry adder (Lecture 04, Slides 115-117)
- Q3: not relevant

Midterm Review: Fall 18 Midterm

- Q1:
  - Q1a: binary addition (Lecture 01, Slides 10-21)
  - Q1b: MUXes to handle casework
- Q2: 1's and 2's complement
  - Q2a: conversion from binary to 1's complement and binary to 2's complement (Lecture 01, Slides 37-46)
  - Q2b: k-bit addition and reusing inputs (Lecture 04, Slides 105-114)
  - Q2c: MUXes to handle casework
- Q3: word problem to truth table/K-map to Boolean expressions (Lecture 04, Slides 98-104)

Midterm Review: Spring 19 Midterm

- Q1:
  - Q1a: unsigned binary overflow (for addition, not subtraction: Lecture 01, Slides 29-36)
  - Q1b: detecting overflow (Lecture 01, Slides 67-68)
  - Q1c: unsigned binary subtraction overflow
  - Q1d: MUXes to handle casework
  - Q1e: "A Slick MUX trick" (Lecture 04, Slides 78-82)
- Q2: identifying behavior of latch given inputs (Lecture 05, Slides 12-17, 18-24, and 25-27)
- Q3: word problem to truth table/K-map to Boolean expressions for FSMs (Lecture 06, Slides 24-42 and 43-71)
  - Q3a: determining FSM (Lecture 06, Slides 24-32 and 44-50)
  - Q3b: filling in truth table for given flip-flop and getting Boolean expression (Lecture 06, Slides 33-35, 36-37, 38-42, and 56-71)

Midterm Review: Spring 21 Midterm

- Q1: full adder (Lecture 04, Slides 105-114)
  - Q1a: contraction (Lecture 04, Slides 127-134)
  - Q1b: MUXes to handle casework
- Q2:
  - Q2a: conversion from binary to 1's complement and binary to 2's complement (Lecture 01, Slides 37-46)
  - Q2b: Range of representation of binary numbers (Lecture 01, Slides 58) and 2's complement overflow (Lecture 01, Slide 68)
- Q3:
  - Q3a: determining FSM (Lecture 06, Slides 24-32 and 44-50)
  - Q3b: filling in truth table for given flip-flop and getting Boolean expression (Lecture 06, Slides 33-35, 36-37, 38-42, and 56-71)

Midterm Review: Other Important Topics and References

- Bitwise Operations (Lecture 02, Slides 39-42)
- XOR Gates (Lecture 02, Slides 9-10 and 54-55)
- NAND and NOR Gates (Lecture 02, Slides 43-45)
- SoP and PoS (Lecture 03, Slides 4-9)
- Minterms and Maxterms (Lecture 03, Slides 10-28)
- Implicant Terminology (Lecture 03, Slide 57)
- "Standard Circuits" (Lecture 04, Slides 35-51)
- Adder Circuits (Lecture 04, Slides 105-114 and 115-126)
- Contraction (Lecture 04, Slides 127-134)
- Latches (Lecture 05, Slides 4-52)
- Flip Flops (Lecture 05, Slides 53-83)
- Finite State Machines (Lecture 06, Slides 18-71)