3827 OH

Eumin Hong (eh2890)

Columbia University

February 1, 2022

Overview

Introductions

Introductions: About Me

- My name is Eumin Hong (UNI: eh2890)
- I am a junior in Columbia College studying computer science
- I am from New York
- This is my fourth time TAing for 3827 (second time with Professor Rubenstein)
- I took this course with Professor Kim in Fall 2020
  - Took 1004, 3134, 3251 before 3827; took 3827 concurrently with 3203
- Only need intro computer science course (how to code) and some high school math (mostly logic) to succeed

Introductions: About You

- Please introduce yourself and mention
  1. Your name
  2. Year
  3. School
  4. Hometown
  5. Anything else that you would like to share

Logistics

Logistics: Course Grading

- Raw score S formula:

$$S = 0.1H + \max\left(0.3M + 0.6F, 0.45M + 0.45F\right)$$

where H is homework, M is midterm, and F is final

Logistics: Homework

- Homework will be submitted via Gradescope (entry code: 6P28R7)
- Scored out of 3
- Relatively loose rubric – check solutions or ask in OH for more detailed explanations
- Extensions are handled by me (not Professor Rubenstein)
  - On the TA side, there is a spreadsheet with the latest possible submission date – I cannot give out extensions past this date unless there is an emergency (in which case
  - Please contact me in advance about extensions – psychologically, I prefer that you ask for an extension and not use it rather than have you ask me for an extension on the original due date
  - I will grant extensions (that come in advance) in general, possible reasons include a lot of work/stress, interviews, upcoming tests/exams/essays, etc.

Logistics: P-credit

- P-credit is a way to boost your grade, scored out of 3
- Homeworks and exams count for less – see `3827_Lecture_00.pdf` for figures
- Obtain maximum P-credit by attending OH, participating, etc.

OH Logistics

OH Logistics: Structure

- Attendance will be taken via Zoom (either chat or participant history)
- After covering announcements (upcoming homeworks, exams, etc.) and notable concepts, it is whatever you would like to do
  - May have additional problems to highlight edge cases/tricks, or can end early
- Make use of this time and prepare your questions in advance – can be about previous/current homeworks, current course content, etc.

OH Logistics: Switching OH Sections

- For both permanent and temporary switches
- For different TA section: email TA whose section you would like to switch to, and **CC me**
  - If you do not email any TA and attend other OH section, P-credit cannot be guaranteed
  - If you miss my OH for the week and want to attend other OH section, email me first (to acknowledge missing OH) and then email TA and CC me
    - No P-credit deduction for first couple times since stuff comes up
  - Email in advance
- For my other section: email me
  - To switch to my other section is not too big of an issue; I will probably be using the same Zoom link and the sections are back-to-back

OH Logistics: OH Mode

- Either online or in-person
- Will definitely have OH in-person for MIPS (topic after the midterm)
- Pros and cons to either mode
- Will probably go with online OH for majority of semester for notes, convenience, and health
- OH will not be recorded, but blank slides will be on GitHub before each OH (ideally) and notes from OH will be on GitHub

OH Logistics: Communication

- If you want to contact me directly, send an email to eh2890@columbia.edu
  - Please add **[3827-oh]** tag to email subject
- Announcements will be via email
- Also open to a group messaging system for OH (will send announcements here as well if created)
  - Options include Slack, etc.
  - Will probably merge my OH sections

OH Logistics: Feedback

- This semester, I will be trying anonymous feedback
- Form: `https://forms.gle/cnUmKVNYN7WvRbHA6`
- You can also provide feedback via email or include your name in the form

OH Logistics: Resources

- OH GitHub: `https://github.com/eh2890/CSEE_W3827_S2022`
  - Will have course lectures slides, OH slides, OH notes, etc. here
- All OH Sections: `http://uribe.cs.columbia.edu/sched/assigned-slotorder.html`
- My OH Sections:
  - Section #11: Tuesday 3pm
  - Section #10: Tuesday 4pm
- Course Google Calendar: `https://calendar.google.com/calendar/u/0/embed?src=16uc8kberc2b6dtltq6e49dv3k@group.calendar.google.com&ctz=America/New_York`
  - Additional OH will be posted here and also announced

Notable Concepts

Notable Concepts: Binary Syntax

When writing in binary, syntax can help everyone trying to read it. We usually deal with binary numbers that have a number of bits divisible by 4 (e.g. 8-bit, 32-bit, 64-bit). Compare the (identical in value) two 32-bit numbers:

00000000110111111111001101011101

0000 0000 1101 1111 1111 0011 0101 1101

The first is hard to read. The second is easier to read. Group bits by four for readability – it's easier to see when you've missed a bit, and it's easier for me to grade.

Notable Concepts: Binary Representation

MSB: bit k-1

What is the value of the 8-bit number `1001 0110`? Is it 150 (unsigned)? Is it -22 (signed magnitude)? Is it -105 (1's complement)? Is it -106 (2's complement)? Is it û (ASCII)?

LSB is bit 0

Binary numbers are just an ordering of bits; their meaning comes from how they are interpreted.

The LSB is bit 0, the MSB is bit $k-1$ for a k-bit binary number. For review:

- Unsigned magnitude: bit i is 2^i, nothing special about MSB
- Signed magnitude: MSB does not have value, simply indicates the sign of the number
- 1's complement: MSB indicates sign – if MSB is 1, then flip all the bits and negate the result
- 2's complement: MSB indicates sign – if MSB is 1, then flip all the bits and add 1

Note: signed magnitude and 1's complement result in two ways to represent 0 (think $+0$ and -0).

Notable Concepts: 2's Complement Subtraction

Subtracting is the same as adding a negative number:

$$X - Y = X + (-Y)$$

Notable Concepts: Detecting Overflow

- Unsigned
  - Addition: overflow occurs if final carry is 1
  - Subtraction: overflow in $X - Y$ occurs if $X < Y$
- 2's complement
  - Overflow occurs if the two final carries (most significant) do not match

$$
\begin{array}{rccc}
 & \boxed{P} & \boxed{Q} & \\
 & A & B & C \\
+ & X & Y & Z \\
\hline
\end{array}
$$

P == Q?

Notable Concepts: Boolean Complement Syntax

Consider the expressions below:

$$\overline{XY} = \overline{X} + \overline{Y}$$
$$\overline{X}\,\overline{Y} = \overline{X + Y}$$

space

The expressions $\overline{XY}$ and $\overline{X}\,\overline{Y}$ are different – the space between the complement bars is crucial.

\overbar{X} \overbar{Y}

Notable Concepts: XOR

For two inputs, the XOR operator ensures that **exactly one** of the inputs is 0 (and **exactly one** of the inputs is 1 as well). Some definitions:

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

$$X \oplus Y = X\overline{Y} + \overline{X}Y$$

Important properties:

- Commutativity: $X \oplus Y = Y \oplus X$
- Associativity: $X \oplus (Y \oplus Z) = (X \oplus Y) \oplus Z$
- Identity: $X \oplus 0 = X$
- Inverse: $X \oplus X = 0$
- Negation: $X \oplus Y = \overline{X} \oplus \overline{Y} \neq \overline{X \oplus Y} = X \overline{\oplus} Y$

XNOR $\odot$

It follows that for n inputs, $X_1 \oplus \cdots \oplus X_n = 1$ for an odd number of X_i. $=1$

Notable Concepts: XNOR

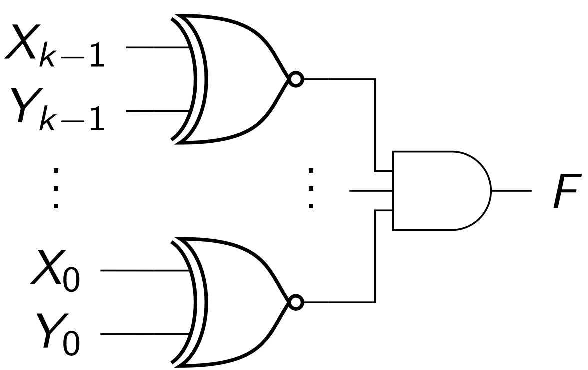
If 2-input XOR checks for inequality, XNOR checks for equality:

X	Y	$X \oplus Y$	$X \overline{\oplus} Y$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

If we perform bit-wise XNOR on two k-bit numbers X and Y and then AND all the results, we can check if $X == Y$:

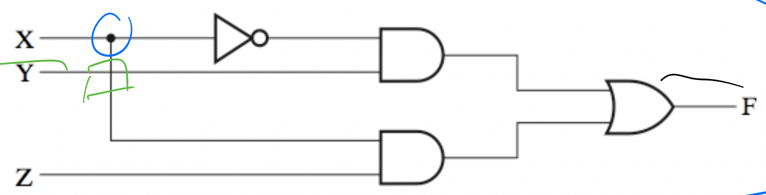
$X: X_{k-1}\ X_{k-2} \cdots X_1\ X_0$

$Y: Y_{k-1}\ Y_{k-2} \cdots Y_1\ Y_0$

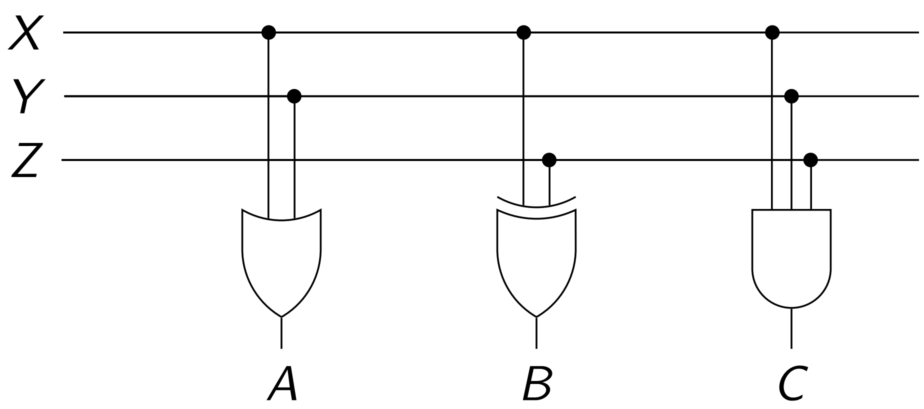


Notable Concepts: Drawing Circuits

Be sure that your circuits are not ambiguous. Use dots to indicate which wires are connected, like so:[1]



You may consider something like:



[1] From Lecture 02, Slide 35

Homework 1

Homework 1: Problem 1

Assume an architecture where all numbers are to be represented using 8 bits. What are the (base 10) values of the 8-bit binary numbers when interpreted using (i) unsigned, (ii) signed magnitude, (iii) 1's complement, (iv) 2's complement form:

a. 00110011
b. 10000000
c. 11111111
d. 10011011
e. 10001010

Homework 1: Problem 2

Convert the following (base 10) numbers to their binary representation using 8-bit (i) signed magnitude, (ii) 1's complement, (iii) 2's complement forms:

- (a) -1
- (b) -15
- (c) -67
- (d) -127

i) $\underbrace{1,0001111}$

7 bits for "15"

$15-8=7-4=3-2=1$

$8+4+2+1$

$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$1 \quad 1 \quad 1 \quad 1$

$2^4=16=10000_2$

$\hookrightarrow 16-1=1111_2$

ii) $15 \rightarrow 00001111$

$\Downarrow$ flip all bits

$-15 = 11110000$

iii) add 1 to 1's comp.: $-15 = 11110001$

Homework 1: Problem 3

For each pair of x and y below, convert x and y to their 8-bit 2's-complement forms, subtract y from x (so you need to negate y and then add). Indicate whether an overflow occured (and show clearly how you know), and convert the solution back to base 10 (even when an overflow occured and the solution is wrong, i.e., your answer should fit in the 8-bit representation of that form.).

a. $x = 10, y = -13$
b. $x = 117, y = 35$
c. $x = 117, y = -35$
d. $x = -117, y = 35$
e. $x = -117, y = 11$

Homework 1: Problem 4

Given the bit pattern 1010 1101 0001 0000 0000 0000 0000 0010, what does it represent, assuming that it is

(a) a two's complement integer?

(b) an unsigned integer?

(c) a single precision (32-bit) floating-point number?

Homework 1: Problem 5

Represent the following numbers, given here in binary form, as floating point numbers using IEEE 754 floating- point standard representation:

a. 11010.1110
b. -11011.10
c. 0.000101
d. -1.010101
e. 10000000010