CS1800
Fall 2025
Recitation 3 - Practice Questions for Homework 2
September 24 & 25, 2025

Recitations

CS1802 Recitations are dedicated time set aside to work on practice problems that specifically prepare you for the current homework or upcoming quiz.

Recitations are in-person and attendance is expected.

The solutions are published at the same time as the problems, so you can check your work. There is no need to submit anything.

Approaching the Problems

These practice problems are labelled according to which Homework or Quiz topic they will help you prepare for. You do not need to complete every practice question; we encourage you to do at least one per topic, and to prioritize the topics you would like to practice.

Instructors & Teaching Assistants

Your recitation is led by a Khoury College professor, assisted by a knowledgeable and wonderful Teaching Assistant. Professors and TAs are fantastic resources, and you have the opportunity in recitation to work with them in a smaller group -- I strongly recommend you take advantage of the time to review your solutions to these practice problems, ask for help on the homework, or review material from lecture.

Practice Problems for Unsigned Numbers (HW2, Question 1)

Convert each of the following (unsigned) numbers from its original base to the given base.

A $010001_2 =$ _____$_{10}$

B $418_{10} =$ _____$_2$

C $42011_5 =$ _____$_{10}$

D $10011100_2 =$ _____$_{16}$

E $3AB2_{16} =$ _____$_2$

Practice Problems for Two's Complement (HW2, Question 2)

A What are the highest positive and lowest negative values we can store in 5-bit two's complement? Give your answer in both two's complement binary and decimal.

B Convert both decimal numbers to 5-bit two's complement, perform the arithmetic in the given base (output must have the same number of digits as input), and report the final answer in decimal.

$13_{10} + 2_{10}$

C Convert both decimal numbers to 5-bit two's complement, perform the arithmetic indicated, and report the final answer in decimal. If overflow occurs, state it in your answer. (Recall that we don't actually perform the subtraction operation; we add negative values instead!)

$13_{10} - 10_{10}$

D Convert both decimal numbers to 5-bit two's complement, perform the arithmetic indicated, and report the final answer. If overflow occurs, state it in your answer.

$13_{10} - 2_{10}$

E Would the following operation result in overflow in 5-bit two's complement?

01110 + 00110

F Would the following operation result in overflow in **unsigned** binary?

01110 + 00110

Practice Problems for Base Conversions (HW2, Question 3)

Suppose you are given an **unsigned** number; the base isn't specified, but you know that the base ranges are 2-16 (i.e., the smallest base you'll see is 2 and the largest is 16). These unsigned numbers might use the digits 0-9 and the characters A-F.

A Is it possible that the digit 3 appears in unsigned binary? Why or why not?

B What is the **smallest** possible base for the unsigned number 1054?

C What is the **largest** possible base for the unsigned number 10110?

D What is the **smallest** possible base for the unsigned number $937A1028$?

E What is the **only** possible base for the following unsigned numbers (both are the same base) if we know that $10 + 11 = 101$?

Fun fact: bases can also be negative! Base -2 is called *negabinary*.

To convert to a negative base like negabinary, we can use the same division algorithm we used for positive bases. However, when dividing by a negative base, we must choose quotients such that the remainder is non-negative. As an example, we will use the division algorithm to convert 10_{10} to negabinary:

$$10 \div -2 = -5\, r\, 0$$
$$-5 \div -2 = 3\, r\, 1$$
$$3 \div -2 = -1\, r\, 1$$
$$-1 \div -2 = 1\, r\, 1$$
$$1 \div -2 = 0\, r\, 1$$

From this algorithm, we see that $10_{10} = 1110_{-2}$

A Convert 34_{10} to negabinary

B Convert 17_{10} to negabinary

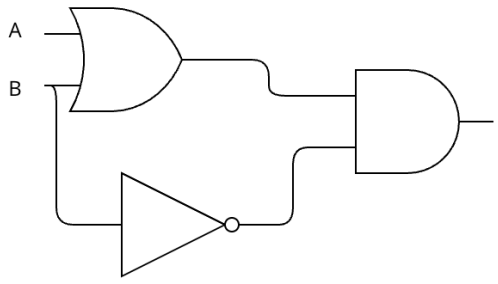
C Convert 00111_{-2} to decimal.

D Can a decimal number have the same binary and negabinary representation? Give an example if Yes, or justify if No.

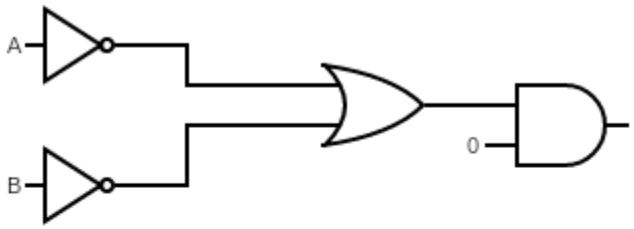
Practice Problems for Circuits & Logic (HW2, Question 4)

A Draw a circuit that represents the logic statement $(A \wedge F) \vee T$. Do not simplify the expression; your circuit should implement the statement directly!

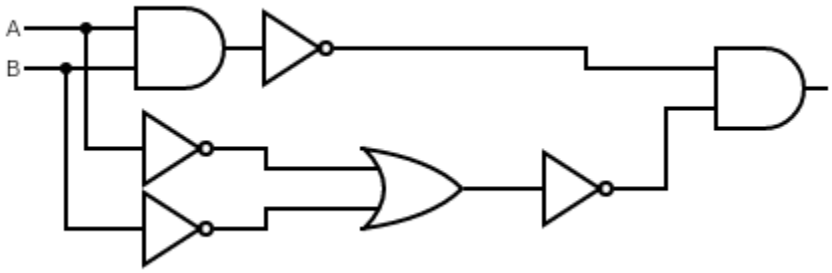
B Write the truth table for the circuit shown below, including all intermediate steps.



(For Parts C-G, Consider the two circuits below.)



Circuit #1



Circuit #2

C Write out logical expressions for both circuits

D Show that the two circuits are equivalent using the laws of logical equivalence. As always, apply one law at a time, take small steps and label each one.

E There are many other circuits that would be equivalent to these two. Draw one that uses three AND gates, one NOT gate, and no other gates.

F Write out a logical expression for your new circuit.

G Simplify your statement by applying the laws of logical equivalence. Clearly state every law applied.

Practice Problems for Circuit Operations (HW2, Question 5)

A Sometimes, we might not have the particular logic gate we need available. Luckily, we can replicate the behavior of some logic gates using other gates. Draw a circuit with the same behavior as an AND gate with only the OR and NOT gates**.**

Write a logical expression that represents your circuit, and use a truth table to confirm it produces the same output as an AND gate.

(Parts B-E are directly useful for homework 2 and should be completed in order :)

B What is the result of adding together the following unsigned bits? Append as many leading bits as necessary; we're not worried about overflow in this case.

0 + 0 = _____

0 + 1 = _____

1 + 0 = _____

1 + 1 = _____

C Now suppose we have only one bit to store the result of the addition, and if we need more than one bit, we save it as a "carry."

0 + 0 sum _____ carry _____

0 + 1 sum _____ carry _____

1 + 0 sum _____ carry _____

1 + 1 sum _____ carry _____

D Write those same answers in a truth table, with A and B as input values and *sum* and *carry* as output values. Using only $\wedge$, $\vee$, $\neg$ operators, give two logical expressions that represent the truth table (one for sum and one for carry).

A	**B**	**Sum**	**Carry**

D Great, that's how we do addition with circuits! Now let's expand this so we're adding columns of bits like we did in the binary-number exercises above. Suppose we're adding the two unsigned binary numbers $110 + 111$.

Every column we add has a *carry-in* (from the previous column) and a *carry-out* (to the next column). For each column below, specify the:

- carry-in
- sum
- carry-out.

	1	1	0
+	1	1	1