15213 - Lecture 2 - POGIL Activity

Introduction

In this activity you will learn about binary numbers. This activity was based on material developed by Professor Saturnino Garcia of the University of San Diego. It is used here with permission.

Before you begin, take a minute to assign roles to each member in your group. Try to switch up the roles as much as possible: please don't pick a role for yourself that you have already done more than once. Below is a summary of the four roles; write the name of the person taking that role next to the summary.

If your group only has three members, combine the roles of Facilitator and Process Analyst.

For this and all future activities, the Facilitator should also take the role of the reader and read the questions aloud to the group.

- **Facilitator**: Reads question aloud; keeps track of time and makes sure everyone contributes appropriately.
- **Quality Control**: Records all answers & questions, and provides team reflection to team & instructor.
- **Spokesperson**: Talks to the instructor and other teams. Compiles and runs programs when applicable.
- **Process Analyst**: Considers how the team could work and learn more effectively.

Fill in the following table showing which group member is performing each role:

Role	**Person**
Facilitator	
Quality Control	
Spokesperson	
Process Analyst	

Model 1: Binary Number Representation

Questions: 11 | Allocated Time: 15 minutes

We have all been trained to think about numbers in terms of a decimal system. We can look at a number like 872 and break it down into ones, tens, and hundreds places and almost instantly understand the quantity with which we are dealing. Unlike humans, computers don't tend to have ten fingers so they use a different representation of numbers: binary representation. Binary is very similar to decimal in many respects with the most obvious difference being that binary systems use only two characters (0 and 1) instead of 10 (0-9).

The following table shows the binary values for the integers zero to ten.

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

1. How many *binary digits* (i.e. *bits*) are used in the representation of the following decimal numbers:

   - 2:
   - 5:
   - 7:
   - 10:

2. List the decimal numbers where the binary representation requires one more *binary digit* (i.e. *bit*) than the previous number.

3. For the numbers you just listed, give the *binary representation* of the number that came before it. For example, if 4 (i.e. 100) was one of your answers, give the binary representation of the number 3 (i.e. 11).

4. What is the **smallest** number that will *require* the following number of bits. Give your answer in decimal representation.

   - 5 bits:
   - 6 bits:
   - 7 bits:

5. Let X be the first number to require N bits. In terms of X, what will be the first number to require $N + 1$?

6. Describe the pattern of the *rightmost* bit as you go from zero to ten.

7. What pattern do the *second* rightmost bit follow when going from zero to ten? To help you see the pattern, add a 0 to the front of both zero and one, making them 00 and 01.

8. **How often** does the *third* rightmost bit transition from 0 to 1 and from 1 to 0? Again, add 0's to the front of the binary representation to get enough bits if needed.

9. Complete the following table to show the binary representation of the numbers 11 to 15, using your knowledge of binary patterns you explored in the previous questions.

Decimal	Binary
11	
12	
13	
14	
15	

10. The decimal number 64 is represented as 1000000 in binary. Fill in the following table with the binary representation of the given numbers, again using the patterns you have discovered up to this point.

Decimal	Binary
64	1000000
65	
66	
67	
68	

11. *(Advanced)* Fill in the following table to indicate **how often** each bit position transitions from 0 to 1 or 1 to 0 (i.e. the frequency of transitions).

Bit	Transition Frequency
Rightmost	1
2nd rightmost	2
3rd rightmost	
4th rightmost	
5th rightmost	
Nth rightmost	

TIME REQUIRED:

Model 10: The Value of a Binary Number

Questions: 12 | Allocated Time: 20 minutes

The following example illustrates how we convert a decimal number to a value:

Digit	8	9	5	9
Index	3	2	1	0
Weight	10^3	10^2	10^1	10^0

$8959 = 8 \times 10^3 + 9 \times 10^2 + 5 \times 10^1 + 9 \times 10^0$

12. What *weighted* value is associated with only the digit at index 2?

13. Does adding a `0` to the *left* of a decimal number (i.e. at the beginning) change its value? For example, changing 759 to 0759. Explain.

14. Does adding a `0` to the *right* of a decimal number (i.e. at the end) change its value? For example, changing 759 to 7590. Explain.

15. Complete the following table that will help convert a binary number to its equivalent decimal value.

Bit	1	1	0	0	1
Index	4	3	2	1	0
Weight					2^0

16. What is the decimal representation of the binary number in the previous question (i.e. 11001)?

17. Does adding a `0` to the *left* of a binary number (i.e. at the beginning) change its value? For example, changing 101 to 0101. Explain.

18. Does adding a `0` to the *right* of a binary number (i.e. at the end) change its value? For example, changing 101 to 1010. Explain.

19. Convert the following binary numbers to decimal:

    - 100101
    - 111011
    - 1000001

20. We can calculate the value of an m digit decimal number according to the following equation:
    $\sum_{i=0..m} d_i \times 10^i$
    Where d_i is the digit at index i (e.g. d_3 in the example above is 8).

    Write a similar equation, except to handle binary instead of decimal numbers.

21. One side effect of the equation you wrote in the last question is that any value can be represented as a sum of powers of 2. For example, 3 is $2^1 + 2^0$ (i.e. 2 + 1) What powers of 2 add together to equal the following numbers.

    - 4:
    - 6:
    - 7:
    - 20:
    - 48:

22. Fill in the following table of bits for the decimal number 48.

Bit							
Index	6	5	4	3	2	1	0
Weight							2^0

23. *(Advanced)* While it might be easy to remember combinations of the powers of 2 for smaller numbers, this approach does not scale well. Briefly describe an algorithm you could use to determine which powers of 2 add to equal a given decimal value.

24. *(Advanced)* The `unsigned char` is an integer datatype in C is usually represented as a single *byte*, where a byte is defined as 8 bits. What is the maximum value of an `unsigned char`, in both binary and decimal representations?

    - Binary Representation:
    - Decimal Representation:

TIME REQUIRED:

Group Reflection

1. What is the range of numbers we can represent with a N bit binary number?
2. Convert the binary number 1110011 into its equivalent decimal representation.
3. Convert the decimal number 99 into its equivalent binary representation.

Model 0x1: Hexidecimal Representation

Questions: 13 | Allocated Time: 15 minutes

One of the downsides of binary numbers is that they require more digits to represent a number than decimal. Hexidecimal numbers are a sort of middle ground between binary and decimal numbers. Below is a table of the hexidecimal representation of the integers 0 to 20.

Dec.	Hex.	Dec.	Hex.
0	0	11	B
1	1	12	C
2	2	13	D
3	3	14	E
4	4	15	F
5	5	16	10
6	6	17	11
7	7	18	12
8	8	19	13
9	9	20	14
10	A		

Note that we'll often use the C notation of adding a '0x' prefix to a hexidecimal number to differentiate it from decimal (e.g. 150 vs 0x150).

1. What possible values can a single hexidecimal digit have? For example, a single bit can have the value 0 or 1, while a single decimal digit can have values between 0 and 9.

2. How many distinct values can a single hexidecimal digit represent? For example, a decimal digit can have 10 distinct values while a binary digit (i.e. bit) can have 2 distinct values.

3. What is the *weight* of the *rightmost* hex digit? Recall that the rightmost decimal digit had a weight of 10^0.

4. What is the weight of the hex digit just to the left of the rightmost digit (i.e. the second rightmost)?

5. Give the hexadecimal and decimal representations of the first number to *require* 3 hex digits?

   - Hexidecimal:
   - Decimal:

6. Which representation will be more compact (i.e. require fewer digits) for large numbers: hexidecimal or decimal?

7. What is the hexidecimal representation of the decimal number 21?

8. What is the hexidecimal representation of 32?

9. Complete the following table by converting the hex numbers into their equivalent decimal representation:

Hex.	Dec.
0x30	
0x5A	
0x11D	

10. One benefit of hexidecimal representation is the ease with which we can convert between it and binary representation. Complete the table below, showing the conversion between a hex digit and binary.

Dec.	Hex.	Bin.
0	0x0	0000
1	0x1	0001
2	0x2	0010
3	0x3	0011
4	0x4	0100
5	0x5	0101
6	0x6	0110
7	0x7	0111
8	0x8	1000
9	0x9	
10	0xA	
11	0xB	
12	0xC	
13	0xD	

14	0xE
15	0xF

11. There is a trivial algorithm to convert from hex to binary: just convert each hex digit to its equivalent 4-bit binary representation. For example, 0xA1 is 1010 (A) followed by 0001 (1), i.e. 10100001. What is the binary representation of the following hex numbers?

    - 0xCAFE:
    - 0x8BADF00D:

12. Devise an algorithm for converting any binary number into its equivalent hexidecimal representation. Your algorithm should be a direct conversion: you should **NOT** convert to any intermediate representation (e.g. decimal).

13. Convert the following binary numbers into their hexidecimal representation:

    - 110100011110:
    - 110111:

TIME REQUIRED:

Model 1+1: Adding Binary Numbers

Questions: 15 | Allocated Time: 15 minutes

The rules of binary addition are very similar to the rules you learned in elementary school for adding decimal numbers.

14. Perform the following decimal addition using the trusty gradeschool algorithm of adding digit-by-digit, carrying over from one column to the next when necessary. Make sure you write down all your work (including carries).

	8	1	5	2	9
+	1	2	2	5	6

15. What is the maximum number that can result from adding two 1 decimal digit numbers together?

16. As a result of your answer to the previous question, decimal additional occasional requires that we *carry* a value from one column to the next. What is the maximum carry value?

17. Complete the following table to indicate the result of adding two 1 bit numbers (b_1 and b_2). The result of addition should also be written in binary notation.

b_0	b_1	$b_0 + b_1$
0	0	0
0	1	

1	0
1	1

18. What is the maximum number of bits required to represent the value of adding two 1-bit numbers?

19. Like decimal addition, binary addition includes the possibility of having to *carry* a 1 value over to the next column and therefore have to add three values together instead of two. Complete the following table to show all possible combinations of adding two 1-bit numbers (b_0 and b_1) with a carry bit (c).

c	b_0	b_1	$c + b_0 + b_1$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	10
1			
1			
1			
1	1	1	

20. What is the maximum number of bits required to represent the value of adding two 1-bit numbers and a carry?

21. Apply what you have learned from the previous questions to add the following two binary numbers together.

	1	0	0	1
+	0	1	0	1

22. Convert the three binary numbers from the previous question (i.e. the two numbers being added together and the result) to their decimal representation. Was the result what you expected?

23. Add the following two binary numbers together.

	1	0	1	0
+	1	1	0	0

24. How many bits were required for the *result* in the previous question?

25. How does the number of bits needed for the result compare to the number of bits in the numbers

you added together?

26. Come up with an equation that describes the maximum number of bits required hold the result of adding two N-bit numbers together.
27. What happens if you only have N bits to hold the result of two N-bit numbers?

TIME REQUIRED:

Group Reflection

1. Convert the hexadecimal number 0x6FA into its equivalent binary representation.
2. Convert the binary number 1111011 to its equivalent hex representation.

Model 010: Representing Negative Values in Binary

Questions: 20 | Allocated Time: 20 mins

The binary representation we have seen thusfar is exactly the representation used in C for "unsigned" data types (e.g. `unsigned int`). The downside of this representation–and consequently unsigned numbers–is that we can only represent non-negative integers. In this model we'll look at how to represent *both* postive *and* negative integers. Later we'll look at how to represent fractional numbers (e.g. 7.31).

In our decimal system a negative number is indicated by the presence of a special symbol: `-`. In our binary system we do not have the luxury of adding another symbol: we are stuck with just 0 and 1. While we could take a number of approaches to introduce negative binary numbers, we'll focus on the dominant representation: *two's complement*.

The two's complement representation is capable of representing both positive and negative numbers. To get the two's complement representation of a non-negative number, is a simple two step process.

- **Step 1:** Start by finding the binary representation of the *magnitude* of the number (like we've done before).
- **Step 2:** Place a 0 to the front of the number.

For example, to get the two's complement representation of 5 we first convert to binary like we did before. This gives us 101. Next, we place a 0 to the front, giving us 0101. That's it!

16. What is the leftmost bit in a non-negative number?
17. What is the *two's complement* representation of the following numbers?
    - 32:
    - 42:

The process of getting the two's complement representation of a negative number is slightly more complex.

- **Step 1:** Start by finding the binary representation of the *magnitude* of the number (like we've done before).
- **Step 2:** Place a 0 to the front of the number. This gives you the **positive** representation of the number.

- **Step 3:** Invert all the bits ($0 \rightarrow 1$, and $1 \rightarrow 0$).
- **Step 4:** Add 1, ignoring the leftmost bit if the addition requires more bits than the positive representation. This gives you the **negative** representation of the number.

You might have noticed that the first two steps are the same for both negative and non-negative values, so its really just two more steps to go from non-negative to negative.

The following table shows the decimal numbers 0 to 10, their magnitude representation, their positive binary representation, the positive number with bits inverted, and the negative representation (i.e. inverted + 1). These columns correspond to steps 1 to 4 from above.

Decimal	Magnitude	Positive	Invert	Negative
0	0	00	11	00
1	1	01	10	11
2	10	010	101	110
3	11	011	100	101
4	100	0100	1011	1100
5	101	0101	1010	1011
6	110	0110	1001	1010
7	111	0111	1000	1001
8	1000	01000	10111	11000
9	1001	01001	10110	10111
10	1010	01010	10101	10110

18. What is the two's complement representation of of the following decimal numbers?

    - 3:
    - -8:

19. Based on the table above, is there an easy way to determine if a two's complement number is positive or negative?

20. Follow the four steps above to find the representation(s) of 15 in two's complement.

    - Step 1:
    - Step 2:
    - Step 3:
    - Step 4:

21. Fill in the following table by performing steps 3 and 4 from above (i.e. invert bits and add 1) but this time *starting with* the **negative** representation.

Decimal	Negative	Invert	Add 1
1	11		
2	110		
3	101		
4	100		

5	1011

22. From the previous question: How does the "Add 1" column compare to the "Positive" column from the earlier table?

23. What is the decimal representation of the following two's complement binary numbers:

    - 010111
    - 111010

24. Complete the following table by converting the two's complement representation to the equivalent decimal representation.

Two's Comp	Decimal
011	
0011	
00011	
101	
1101	
11101	

25. How would you make an N-bit two's complement number into an $(N+1)$-bit two's complement number *without* changing its value?

26. Some of the entries in the "Negative" column in the first table have more bits than *required* to represent that number. Identify these entries and give the minimal two's complement representation (i.e. minimize the number of bits without changing the value that it represents).

27. Complete the following table to indicate the most positive (i.e. largest) and most negative (i.e. smallest) number that can be represented with a given number of bits *when using two's complement representation*. Use the table above (which you simplified by removing unnecessary bits) to help you answer this question.

Bits	Most Positive	Most Negative
1	0	-1
2	1	-2
3		
4		

28. Use your answer from the previous question to find an expression that gives the *most positive* number that can be represented by a N-bit two's complement number. Hint: This will be related to a power of two in some way.

29. Use your answer from the previous questions to find an expression that gives the *most negative* number that can be represented by a N-bit two's complement number. Hint: This will be related to a power of two in some way.

30. How does the *magnitude* of the most positive number compare with the *magnitude* of the most negative number in for an N-bit two's complement number?

31. *How many* distinct integers can be represented by an N-bit two's complement representation (i.e. what is the difference between the most positive and the most negative)?

32. Perform binary addition on the two's complement numbers below:

```
  1001
+ 0011
------
```

33. What are the decimal representations of your inputs (1001 and 0111) and of your solution from the previous question? Is this the answer you expected?

34. Devise an algorithm for performing *binary subtraction* but using only addition.

35. An alternative to two's complement is *sign-magnitude* representation. In this representation, you first get the binary representation of the magnitude (like step 1 of two's complement). To get the positive version, you simply place a 0 to the front of the magnitude. To get the negative version, you simply place a 1 to the front of the magnitude. What potential drawbacks does sign-magnitude representation have?

TIME REQUIRED:

Group Reflection

1. Add the following two binary numbers together:

```
  1111000
+ 0100111
---------
```

   Confirm that the result is correct by converting all the binary numbers to their decimal representation.

2. Give the decimal representation of the following two's complement binary numbers:

   - 1111111111
   - 0110100010

3. Give the two's complement representation of the following binary numbers:

   - 105
   - -482

4. What is the range of numbers that can be represented by a 32-bit two's complement number?