For this lab, you will need to know a little bit about binary numbers. This course will cover binary numbers more in-depth in the future, but this introduction should get you through lab 1. The two important things we need to know are: 1.What are binary numbers? 2.How do we convert numbers between decimal and binary?

1. What Is a Binary Number?

Why Do We Need Binary Numbers?

- Binary numbers are essential in digital systems because they provide a simple, reliable way to interpret electronic signals as data. To interpret voltages from our circuits as bits, we use a "Logic High Output/Input Range" and a "Logic Low Output/Input Range" to categorize voltages. As discussed in Lecture 2, If the detected voltage is within the "Logic High Output/Input Range" it is interpreted as logic 1. Similarly, if the detected voltage is within the "Logic Low Output/Input Range" it is interpreted as logic 0. These two ranges are far enough apart that the detected voltage may fluctuate slightly without being interpreted as the wrong signal (i.e, even if a low voltage rises slightly above 0, it is still interpreted as logic 0). Because this categorization only uses two states, 0 and 1, we use binary numbers to represent the information being sent and received via physical voltages, which has formed the backbone of modern computing technology.

How Do Binary Numbers Work?

- Even if you are not familiar with binary numbers, you are likely familiar with decimal numbers (whether you know it or not)! The decimal number system is the default number system used in our everyday lives, so it's simplest to explain binary numbers by drawing parallels between the two.

- Numbers in the Decimal Number System are conveyed using combinations of the magnitudes 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. With the prefix "deci-" meaning "one-tenth", notice there are 10 unique numbers, or digits, that can make up a "decimal number". It's important to know this numbering system uses base 10.
  - Examples of Decimal Numbers: 0, 1, 2, 27, 101, 956, and 26,340

  When counting in decimal, we start at 0 and increment through each of the 10 possible magnitudes for a digit until we reach 9, after which comes the number 10. While the number 9 only has one digit, its neighbor, 10, requires two digits– why?

  This is because 9 is the greatest magnitude a single digit may convey, so we need an additional digit to display any larger values. Thus, the 9 in the ones place "rolls over" to a 0, and a 1 is introduced in the tens place– the next position to the left of our existing digit. Similarly, past the number 99, we need to introduce a third digit in order to convey the next value up, 100. Both of the 9's "roll over" into 0's while a 1 is introduced in the hundreds place.

- Numbers in the Binary Number System are conveyed using combinations of the magnitudes 0 and 1. With the prefix “bi-” meaning “two”, notice there are 2 unique numbers, or bits, that can make up a “binary number”. It’s important to know this numbering system uses base 2.
    - Examples of Binary Numbers: 0, 1, 101, 1101, 1010101
      Note: Although 101 is both a binary number and a decimal number, it represents different values in the different numbering systems

When counting in binary, we start at 0 and increment through the 2 possible magnitudes for a bit. So, starting with 0 the next number up is 1. But what comes next? We cannot increment the 1 to a 2, since binary numbers can only be represented using 1’s and 0’s, so now what?

Well, in binary, a 1 is the greatest magnitude a single bit can convey (like how 9 is the greatest magnitude a single digit can convey). So just like a 9 in decimal, to display the next number up after 1 in binary, we need an additional bit! Our 1 will now “roll over” to a 0, and a new bit with magnitude 1 is introduced to the left of our existing bit. So, we’ve gone from 0 to 1 to 10. What do you think the next incremented number will be?

Examining 10, we do not need to introduce another bit yet because the 0 in the rightmost position can still increment to a 1. So, 10 turns into 11– what comes next? Like in the case of 99, both of our bits display the highest possible magnitude, so both of the 1’s “roll over” into 0’s while a 1 is introduced in the next position to the left, giving us 100. Following this pattern, you can now count in binary!

Decimal:	0	1	2	3	4	5	6	7	8	9	10	11
Binary:	0	1	10	11	100	101	110	111	1000	1001	1010	1011

Now you see, 101 in decimal represent “one hundred & one” while 101 in binary represents “five”

Side By Side Comparison	
Decimal Numbers	**Binary Numbers**
Base 10	Base 2
Consist of “Digits”	Consist of “Bits”
Digits can have magnitude 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9	Bits can have magnitude 0 or 1

- It’s also important to be able to determine the position of a digit or a bit in a number, and luckily bits & digits follow the same procedure. Simply start at 0, then count from right to left until you reach the number of interest. We call the leftmost bit the Most Significant Bit(MSB), and we call the rightmost bit the Least Significant Bit(LSB).

- Examples of Finding Position of Digits/Bits:
    - 956 in decimal (has 3 digits) ⇒ the 6 is in position 0, the 5 is in position 1, the 9 is in position 2.
        - ***Note: 6 is in position 0 because it is the rightmost digit, and we start counting from right to left. 5 is in position 1 because it is the next digit to the left, etc.***
    - 1101 in binary (has 4 bits) ⇒ the rightmost 1 is in position 0, the 0 is in position 1, the middle 1 is in position 2, the leftmost 1 is in position 3.
        - ***Note: the 1 in position 0 is the Least Significant Bit while the 1 in position 3 is the Most Significant Bit***

2. Converting Binary Numbers to Decimal

Converting to Decimal

- If you only have one takeaway from this section, remember this formula which can be used to convert a number to its decimal value**:**

$$\text{Decimal Value} = \Sigma^{j}_{i=0} (\text{Magnitude}_i \text{ x } \text{Base}^{\text{Position } i}),$$

where j is the position of the leftmost bit/digit

Simply find the base of the number you are looking at, the magnitude and position of each bit/digit, plug the information for each bit/digit into this formula , then sum up those values!

Examples

- Let's take a look at the decimal number "2300". Try to find its base as well as the magnitude and position of each digit.
    - Because it's a decimal number, our number has base 10.
    - Now, gather your information about the 4 digits in our number:

Magnitude	**Position**	**$\text{Base}^{\text{Position}}$**
2	3	10^3 = 1,000 (The Thousands Place)
3	2	10^2 = 100 (The Hundreds Place)
0	1	10^1 = 10 (The Tens Place)
0	0	10^0 = 1 (The Ones Place)

Let's apply our formula to this information:

- Σ (Magnitude x $\text{Base}^{\text{Position}}$)
  = $(2 \text{ x } 10^3) + (3 \text{ x } 10^2) + (0 \text{ x } 10^1) + (0 \text{ x } 10^0)$

= (2 x 1,000) + (3 x 100) + (0 x 10) + (0 x 1)
= 2,000 + 300 + 0 + 0 = 2300

Using this formula, you can see that breaking down the decimal number 2300 yields the decimal number 2300 again. This process may seem intuitive, as you have likely worked with decimal numbers before. Though it may be less intuitive, this same process will work to convert a binary number to a decimal number– Let's try an example!

- Let's take a look at the binary number "11100110". Try to find its base as well as the magnitude and position of each bit.
  - Because it's a binary number, our number has base 2.
  - Now, gather your information about the 8 bits in our number:

Magnitude	**Position**	**$Base^{Position}$**
1	7	$2^7 = 128$
1	6	$2^6 = 64$
1	5	$2^5 = 32$
0	4	$2^4 = 16$
0	3	$2^3 = 8$
1	2	$2^2 = 4$
1	1	$2^1 = 2$
0	0	$2^0 = 1$

Let's apply our formula to this information:

- Σ (Magnitude x $Base^{Position}$)
  = $(1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$
  = (1 x 128) + (1 x 64) + (1 x 32) + (0 x 16) + (0 x 8) + (1 x 4) + (1 x 2) + (0 x 1)
  = 128 + 64 + 32 + 0 + 0 + 4 + 2 + 0 = 230

Using this formula, you can see that breaking down the binary number "11100110" actually yields the decimal number 230!
Note: In this lab, you will not have to convert binary numbers larger than 5 bits

- You can also think of this process as summing all of the active positions, where an active position is denoted by a bit set to 1. The reason it's important to be able to determine the position of each bit is because the value of each active position is actually equal to $2^{position}$. This is because the magnitude is only ever 0(which means you can ignore that position) or 1(which means you just need to add the weight of that position itself). Knowing this makes converting numbers from binary to decimal much easier.

- Take another look at the binary number "11100110". Try converting to decimal with our new approach:

Bits:	1	1	1	0	0	1	1	0
Positions:	7	6	5	4	3	2	1	0

Adding Active Bits: $2^7 + 2^6 + 2^5 \quad + 2^2 + 2^1 = 128 + 64 + 32 + 4 + 2 = 230$

3. Converting Decimal Numbers to Binary

The Division Method

There are multiple ways to convert from decimal to binary, which you will learn later on, but we will only be discussing one here– the division method. Start with the decimal number you're interested in and divide it by 2, making note of whether the remainder is 0 or 1. Then, you take your result and divide it by 2, noting the remainder again. Continue this process until you reach "1 ÷ 2" which will always give you a remainder of 1. Now, rewrite your list of remainders in the same order with the most recent one (from "1 ÷ 2") on the left(MSB) and the first remainder on the right(LSB).

Examples

- Try converting the decimal number 23 to binary using the division method:

Division	Result	Remainder (R)
23 ÷ 2 =	11	1 (⇐ LSB)
11 ÷ 2 =	5	1
5 ÷ 2 =	2	1
2 ÷ 2 =	1	0
1 ÷ 2 =	0	1 (⇐ MSB)

Now, let's rewrite our remainders such that the most significant bit is the most recent remainder that was found ⇒ 10111

You can check your work by converting the number back to decimal, and if you applied the division method correctly, it should equal 23.

Bits:	1	0	1	1	1
Positions:	4	3	2	1	0

Adding Active Bits: $2^4 \quad + 2^2 + 2^1 + 2^0 = 16 + 4 + 2 + 1 = 23$ 🙂

- Try converting the decimal number 49 to binary using the division method:

Division	**Result**	**Remainder (R)**
49 ÷ 2 =	24	1 (⇐ LSB)
24 ÷ 2 =	12	0
12 ÷ 2 =	6	0
6 ÷ 2 =	3	0
3 ÷ 2 =	1	1
1 ÷ 2 =	0	1 (⇐ MSB)

Rewriting our remainders, we get ⇒ 110001

Checking our work:

Bits:	1	1	0	0	0	1
Positions:	5	4	3	2	1	0

Adding Active Bits: $2^5 + 2^4 + 2^0 = 32 + 16 + 1 = 49$ (Wahoo!)

4. Additional Resources

- This tutorial should get you through lab 1, and if you have any questions on information mentioned above don't hesitate to post on Ed or ask a TA! You shouldn't have to convert numbers with more than 5 bits in this particular lab, but it's helpful to know your powers of 2 up to 2^{10} for this course off the top of your head.
- For additional information, see Chapter 1 of your textbook
- For confusion on the conversion methods discussed, see the following videos
  - Binary to Decimal
  - Decimal to Binary (Starting at 7:27)