**CPSC111 Spring 2017**

**Binary Number System:** <http://sandbox.mc.edu/~bennet/cs110/textbook/module1.html>

**Binary negative numbers**

<http://sandbox.mc.edu/~bennet/cs110/textbook/module3_2.html>

**Binary addition and subtraction**

<http://sandbox.mc.edu/~bennet/cs110/textbook/module3_1.html>

**Decimal to Binary Conversion**

(<http://www.wikihow.com/Convert-from-Decimal-to-Binary>)

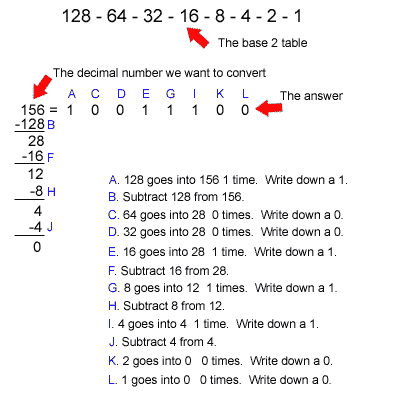
The [**decimal**](http://www.wikipedia.org/wiki/Decimal) (*base ten*) [**numeral system**](http://www.wikipedia.org/wiki/Numeral_system) has ten possible values (0,1,2,3,4,5,6,7,8, or 9) for each place-value. In contrast, the [**binary**](http://www.wikipedia.org/wiki/Binary_numeral_system) (*base two*) **numeral system** has two possible values, often represented as 0 or 1, for each place-value.

To avoid confusion while using different numeral systems, the [base](http://www.wikipedia.org/wiki/Radix) of each individual number may be specified by writing it as a subscript of the number. For example, the decimal number 156 may be written as 15610 and read as "one hundred fifty-six, base ten". The binary number 10011100 may be specified as "base two" by writing it as 100111002.

Since the binary system is the internal language of electronic computers, serious computer programmers should understand how to convert from **decimal to binary**. Although, converting in the opposite direction, from [binary to decimal](http://www.wikihow.com/Convert-from-Binary-to-Decimal), is often easier to learn first.

Comparison with descending powers of two and subtraction

1. List the powers of two in a "base 2 table" from right to left. Start at, evaluating it as "1". Increment the exponent by one for each power (). The list, to ten elements, would look like this: 512, 256, 128, 64, 32, 16, 8, 4, 2, 1
2. **For this example, let's convert the decimal number 15610 to binary**. What is the greatest power of two that will fit into 156? Since 128 fits, write a 1 for the leftmost binary digit, and subtract 128 from your decimal number, 156. You now have 28.
3. **Move to the next lower power of two**. Can 64 fit into 28? No, so write a 0 for the next binary digit to the right.
4. **Can 32 fit into 28**? No, so write a 0
5. Can 16 fit into 28? Yes, so write a 1, and subtract 16 from 28. You now have 12.
6. Can 8 fit into 12? Yes, so write a 1, and subtract 8 from 12. You now have 4.
7. Can 4 (power of two) fit into 4 (working decimal)? Yes, so write a 1, and subtract 4 from 4. You have 0.
8. Can 2 fit into 0? No, so write a 0.
9. Can 1 fit into 0? No, so write a 0.
10. Since there are no more powers of two in the list, you are done. You should have 10011100. This is the binary equivalent of the decimal number 156. Or, written with base subscripts: 15610 = 100111002



**Short division by two with remainder**

This method is much easier to understand when visualized on paper. It relies only on division by two.1

**For this example, let's convert the decimal number 15610 to binary**. Write the decimal number as the dividend inside an upside-down "long division" symbol. Write the base of the destination system (in our case, "2" for binary) as the divisor outside the curve of the division symbol.

2)1562

**Write the integer answer (quotient) under the long division symbol, and write the remainder (0 or 1) to the right of the dividend**.

2)156   0  
   78

**Continue downwards, dividing each new quotient by two and writing the remainders to the right of each dividend**. Stop when the quotient is 0.

2)156   0  
 2)78   0  
 2)39   1  
 2)19   1  
  2)9   1  
  2)4   0  
  2)2   0  
  2)1   1  
    04

**Starting with the bottom remainder read the sequence of remainders upwards to the top**. You should have 10011100. This is the binary equivalent of the decimal number 156. Or, written with base subscripts: 15610 = 100111002

**How to Convert a Negative Binary to Decimal**Instructions

Step1:

Write out the binary number in a column running down a piece of paper. This will allow the value of each bit to be determined and then the total summed.

Step2:

Take the most significant bit and remove it from the column before converting it. If the **left most bit in the word is zero the number is positive** and if the **left most bit is a one**, then the **number is negative**.

Step3:

Write down next to each bit on the piece of paper the value of that position in the word. Each position in the binary word has a value equal to two raised to the power of the bit position. For example, the value of bit one is one, value of bit two is two and bit three is four. Continue down the list of bits and enter the value of each one.

Step4:

Multiply the value of the bit in the original binary number and the value of the position as found in Step 2. This will give a piece of the puzzle. Continue to evaluate each binary bit in the word. Once you have completed this process, add up the list of numbers.

Step5:

Apply the sign to the total value found in Step 4 to find the decimal equivalent of the binary number.

Instructions through an example:

Step1:

To convert from binary to decimal is a simple process. For an example, let's use the binary 101010 as an unsigned.

Step2:

First, let's list the binary number vertically. List them in order from RIGHT TO LEFT

0  
1  
0  
1  
0  
1

Step3:

Now let's fill in the values

0 x 1 = 0  
1 x 2 = 2  
0 x 4 = 0  
1 x 8 = 8  
0 x 16 = 0  
1 x 32 = 32

Step4:

Now just add the values together

0 + 2 + 0 + 8 + 0 + 32 = 42

And that's you're answer. 101010 in binary is 42 in decimal.

Now let’s convert 1011 0101 as a signed binary number. Since the left most bit is 1 then the decimal number is negative. Now, let’s try to find the decimal number from the remaining 7 bits:

011 0101 will correspond to = 0+32+16+0+4+0+1 = 53 and as we had the left most bit a 1 so the final answer is -53

Or I can solve it this way:

1\*1 = 1

0\*2 = 0

1\*4 = 4

0\*8 = 0

1\*16= 16

1\*32= 32

0\*64= 0

Total = 53 and since bit 8 (left most) was1the decimal is -53

**Binary To Hex Conversions**

Conversion between hex and binary is easy. Simply substitute four-bit groups for the hex digit of the same value. Specifically:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Hex Digit: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Bit Group: | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| Hex Digit: | 8 | 9 | a | b | c | d | e | f |
| Bit Group: | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

For conversion from hex to binary, simply string together the bits for each hex digit. For instance, 0x509d7a is binary 10100001001110101111010. To wit:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hex Number: | 5 | 0 | 9 | d | 7 | a |
| Binary Number: | 0101 | 0000 | 1001 | 1101 | 0111 | 1010 |

To convert the other way, break the binary number into groups of four then replace each one with its hex digit. Group the digits *starting from the right*. If you don't have a complete group of four, when you reach the left, then *pad with zero bits on the left* to fill the last group. For instance, binary 111011011111110001 is 0x3b7f1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Binary Groups: | 0011 | 1011 | 0111 | 1111 | 0001 |
| Hex Digits: | 3 | b | 7 | f | 1 |

Because this conversion is so easy, the easiest way to convert between binary and decimal is usually to go through hex. It generally requires fewer operations, and hex numbers are easier to work with because they are shorter Also, it's easier to remember where you are when scanning a hex number, since the digits differ more.

**Decimal to Two's Complement Conversion**

These examples show conversion of a decimal number to 8-bit twos complement. The bit size is always important with twos complement, since you must be able to tell where the sign bit is.

The steps are simple. First, you convert the magnitude of the number to binary, and pad to the word size (8 bits). If the original number was positive, you are done. Otherwise, you must negate the binary number by inverting the bits and adding 1.

* Convert -72 to an 8-bit, twos complement binary number.
  1. Convert the magnitude, 72 to binary. The easiest way is to convert it to hex first. 72÷16 = 4 remainder 8, so 7210 = 4816 = 10010002.
  2. Pad to 8 bits: 01001000
  3. Negate the number by inverting the bits and adding 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| ¬ | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

* 1. So, -7210 is 10111000 as an eight-bit, two's complement number.
* Convert 47 to an 8-bit, twos complement binary number. This is positive, so all that is needed is to convert to binary and pad to eight bits. 47÷16 = 2 remainder 15, so 4710 = 2f16 = 1011112. So 47 as an 8-bit two's complement number is just 00101111.
* Convert -109 to an 8-bit, twos complement number. Again, the magnitude: 109÷16 = 6 remainder 13, so 10910 = 6d16 = 11011012.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| ¬ | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

* Convert -67 to an 8-bit, twos complement number. 67÷16 = 4 remainder 3, so 6710 = 4316 = 10000112.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| ¬ | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

* Convert 81 to an 8-bit, twos complement number. Since this is positive, it's just a matter of converting to binary and padding to 8 bits. 81÷16 = 5 remainder 1, so 8110 = 5116 = 10100012, giving 01010001.

**Two's Complement to Decimal Conversion**

These are examples of converting an eight-bit two's complement number to decimal. To do this, you first check if the number is negative or positive by looking at the sign bit. If it is positive, simply convert it to decimal. If it is negative, make it positive by inverting the bits and adding one. Then, convert the result to decimal. The negative of this number is the value of the original binary.

* Interpret 11011011 as a two's complement binary number, and give its decimal equivalent.
  1. First, note that the number is negative, since it starts with a 1.
  2. Change the sign to get the magnitude of the number.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| ¬ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| + |  |  |  |  |  |  |  | 1 |
|  | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

* 1. Convert the magnitude to decimal: 001001012 = 2516 = 2×16 + 5 = 3710.
  2. Since the original number was negative, the final result is -37.
* Interpret 01101001 as a two's complement binary number, and give its decimal equivalent. The number is positive, so simply convert it to decimal: 011010012 = 6916 = 6×16 + 9 = 10510.
* Interpret 11110010 as a two's complement binary number, and give its decimal equivalent.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| ¬ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| + |  |  |  |  |  |  |  | 1 |
|  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

* 000011102 = e16 = 0×16 + 14 = 1410. Answer: -14.
* Interpret 10011100 as a two's complement binary number, and give its decimal equivalent.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| ¬ | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| + |  |  |  |  |  |  |  | 1 |
|  | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

* 011001002 = 6416 = 6×16 + 4 = 10010. Answer: -100.
* Interpret 01010111 as a two's complement binary number, and give its decimal equivalent. 010101112 = 5716 = 5×16 + 7 = 8710.

**Two's Complement Binary Addition Examples**

Here are some examples of eight-bit, twos complement binary addition. In each case, we compute the sum, and note if there was an overflow. If there was a carry out, the extra bit is shown on the next line. (It's falling into the bit bucket, where it will never be heard from again.) You can also look at the [rules](http://sandbox.mc.edu/~bennet/cs110/tc/orules.html) for determining overflow.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| • -39 + 92 = 53:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 |  | 1 | 1 |  |  |  |  | |  | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | | + | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | |  | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |   Carryout without overflow. Sum is correct. | • -19 + -7 = -26:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 |  |  | 1 |  | |  | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | | + | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | |  | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |   Carryout without overflow. Sum is correct. | • 44 + 45 = 89:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | 1 |  | 1 | 1 |  |  |  | |  | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | | + | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | |  | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |   No overflow nor carryout. |
| • 104 + 45 = 149:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 1 |  | 1 |  |  |  |  | |  | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | | + | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | |  | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |   Overflow, no carryout. Sum is not correct. | • -75 + 59 = -16:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | 1 | 1 | 1 | 1 | 1 | 1 |  | |  | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | | + | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | |  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |   No overflow nor carryout. | • -103 + -69 = -172:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 |  | 1 | 1 | 1 |  | 1 | 1 |  | |  | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | | + | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |   Overflow, with incidental carryout. Sum is not currect. |
| • 10 + -3 = 7:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 |  |  |  |  | |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | | + | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | |  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |   Carryout without overflow. Sum is correct. | • 127 + 1 = 128:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | |  | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | + | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   Overflow, no carryout. Sum is not correct. | • -1 + 1 = 0:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | + | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   Carryout without overflow. Sum is correct. |