O(N) CS LESSONS

Two's Complement

The Challenge of Representing Negative Numbers in Binary

By John B. Owen

Positive Numbers In Binary

- Expressing positive values in binary, or even zero, is a pretty straight forward thing.

Positive Numbers In Binary

- By using the place value system, it is fairly simple to see that 23_{10} converts to 10111_{2}, where the places values of 16, 4, 2 and 1 indicated by the 1s from left to right all add up to 23.

```
1   0   1   1   1
16  8   4   2   1
```

Negative Numbers In Binary

- Representing negative numbers in binary is a little trickier and was a challenge in the beginning stages of modern computing, with several attempts tried and abandoned.

Sign-Magnitude Representaion

- This was the first and most obvious method, making the leading bit of a binary string the sign bit, and just changing it to represent positive or negative, as we do in base 10, like this:

$$23_{10}$$

$$-23_{10}$$

Sign-Magnitude Representaion

- The value 23_{10} in an 8-bit representation is actually **0**0010111$_2$, where the leading zero "sign" bit indicates it is positive.
- To represent the negative number, take the corresponding positive number (its *magnitude*) and use 1 as the "sign" bit.
- **1**0010111$_2$,

Sign-Magnitude Representaion

- This seems to work fine, but it poses two significant problems
    - Arithmetic is very tricky (we won't go into that)
    - There are two versions of zero,

      00000000 and 10000000

      which is very confusing

1's Complement

- To fix the arithmetic problem, this next system was devised, where all bits are just reversed.
- The opposite, or negative of 23_{10}, or

$$00010111_2$$

was 11101000_2

1's Complement

- The reason for the name, 1's complement, is the other way to get the negative number, and that is to subtract the positive number from 11111111_2

```
 11111111
-00010111
=11101000
```

1’s Complement

- This system solved the arithmetic issue, but still had problems with zero, in that there were still two versions:

  00000000 and 11111111

- Theoretically, they are equal, but technically, they are not! Very confusing!

2’s Complement

- To fix this issue, the 2’s complement system was devised.
- Most modern computers these days use this system, which essentially divides the range of possible values of an integer into two parts, the negatives and the non-negatives.

2’s Complement

- Every non-negative value (including zero) has a negative complement.
- The complement of zero is -1
- The complement of 1 is -2
- The complement of 2 is -3

2’s Complement

- Here is a number line to show this relationship:

  -3 -2 -1 | 0 1 2

- Do see the pattern?

2’s Complement

-3 -2 -1 | 0 1 2

- The base 10 rule to think of for finding complement values is ***“opposite, minus 1”***
- The opposite of 0 is 0, minus 1 is -1
- The opposite of -1 is 1, minus 1 is 0

2’s Complement

-3   -2   -1 | 0   1   2

- The opposite of 1 is -1, minus 1 is -2
- The opposite of -2 is 2, minus 1 is 1

2’s Complement

-3 -2 -1 | 0 1 2

- The opposite of 2 is -2, minus 1 is -3
- The opposite of -3 is 3, minus 1 is 2

2’s Complement

____. . . -3   -2   -1 | 0   1   2 . . .____

- What values are next on both sides?
- If you thought -4 and 3, you are correct!

2’s Complement

- The symbol used in Java for the complement of a value is the “~” symbol.
- The expression ~0 means “the complement of zero”, and has a resulting value of -1
- ~ -2 results in the value 1
- Keep thinking, “opposite, minus 1”

2’s Complement

- What is the result of ~10?

2’s Complement

- What is the result of ~10?
- If you thought -11, you are correct!
- The opposite of 10 is -10, minus 1 is -11.
- It is very simple when you think of it in base 10.

2's Complement

- Try these:

~23 is _____

~ -51 is _____

~100 is _____

~ -17 is _____

2’s Complement

- Answers:

~23 is -24

~ -51 is 50

~100 is -101

~ -17 is 16

2’s Complement

- Now, how are 2’s complement values expressed in binary form?
- In binary, the 2’s complement representations are simply flipped versions of each other.
- Zero is 00000000,

  and negative 1 is 11111111

2’s Complement

- The value 1_{10} is 00000001_2
- The complement value is -2_{10} is 11111110_2
- Do you see that these two binary strings are just flipped versions of each other?

2’s Complement

- What is the complement value of 23_{10}, or 00010111_2?

2's Complement

- What is the complement value of 23_{10}, or 00010111_2?
- Just flip all the bits to get -24_{10}!
- 11101000_2

2’s Complement

- What if you wanted to express a base ten negative number in two’s complement binary form, directly from the value?
- For example, -24_{10} from the previous example is 11101000_2

2's Complement

- We've seen the indirect way to do this, and that is to think of the complement of -24, or 23, express it in binary, and then flip all of the bits.
- 23_{10} is equivalent to 00010111_2
- The flipped version of this is 11101000_2 which represents the value -24.

2’s Complement “Ripple” technique

- The direct way to do this is to express positive 24_{10} in binary, which is 00011000_2, and then apply the “ripple” technique, which takes two steps:

1. Starting from the right end of the number, keep all zeroes unchanged until you reach the first ‘1’.
2. Leave it also, but reverse all the remaining digits to the left.

2's Complement "Ripple" technique

~~0001~~1000_2

Flip Keep

11101000_2

Two's Complement binary representation of -24_{10}

Try these using the “ripple” technique…

$-42_{10} \rightarrow 42_{10} = 00101010_2 \rightarrow$ ______________

$-95_{10} \rightarrow 95_{10} = 01011111_2 =$ ______________

$-121_{10} \rightarrow 121_{10} = 01111001_2 =$ ______________

$-125_{10} \rightarrow 125_{10} = 01111101_2 =$ ______________

Answers

$-42_{10} \rightarrow 42_{10} = 00101010_2 \rightarrow 11010110_2$

$-95_{10} \rightarrow 95_{10} = 01011111_2 = 10100001_2$

$-121_{10} \rightarrow 121_{10} = 01111001_2 = 10000111_2$

$-112_{10} \rightarrow 112_{10} = 01110000_2 = 10010000_2$

Lesson Summary

- Two’s Complement is the third and most successful attempt at representing negative values in binary
- The “complement” operator in Java is “~”
- The base 10 way to find complement is to think, “opposite, minus 1”

Lesson Summary

- The indirect way to find the two’s complement binary representation of a negative base ten value is to:
    1. Think of the complement of that negative base ten value
    2. Express it in binary
    3. Flip all of the bits

Lesson Summary

- The direct way to find the two's complement binary representation of a negative base ten value is to:
    1. Express the positive value in binary
    2. Apply the two-step "ripple" technique