

Being Positive About Negative Numbers: How to Win Compliments When Teaching 2's Complements

Gordon Grimsey

Auckland University of Technology
Faculty of Business
Auckland, New Zealand
Gordon.Grimsey@aut.ac.nz

Keywords

Binary, octal, hexadecimal, nine's complement, ten's complement, one's complement, two's complement, sign-and-magnitude, mantissa, excess-127, equal additions, decomposition.

ABSTRACT

This paper discusses various techniques which have been used in the classroom to explain how negative numbers are represented inside a digital computer. The method starts by examining the decimal system and common ways of depicting negative numbers using familiar examples. The concept of the nine's and ten's complement is introduced by way of a car odometer. Base ten subtraction techniques are examined, leading to using ten's complement numbers to do subtraction. Rules for doing signed addition and subtraction are revised using a "human simulation".

The concept of the two's complement is introduced leading to its application in addition, subtraction, and multiplication of signed numbers. A novel method of converting 2's complement numbers to base ten is demonstrated. Finally, the subtraction technique is generalised to octal and hexadecimal numbers.

1. INTRODUCTION

Many introductory computing courses teach basic number systems involving binary, octal and hexadecimal numbers. Part of this body of knowledge includes the concept of representing negative binary numbers using the 2's complement format. The author has found that many students have extreme difficulty understanding how the 2's complement system works. Part of this problem relates to their exclusive exposure to the "sign-and-magnitude" notation for representing negative numbers in base ten, and partly to the confusing array of methods which are employed in a typical digital computer for representing numerical values. For example, it is not uncommon to have:

- ◆ unsigned binary integer,
- ◆ signed two's complement binary integer,
- ◆ packed-decimal (hex F/C/D "overpunched" on the right-most digit), and
- ◆ IEEE-754 floating point format which uses both sign-and-magnitude (for the mantissa), and excess-127 (for the exponent)

-all within one machine. Of course, to be complete, one must also mention the "one's complement" as a stepping stone towards the "two's complement"! ...

Another possible explanation for this difficulty is the students' lack of numeracy, which seem to have accompanied the "calculators in schools" generation.



2. GETTING STARTED

2.1 Exposure to Negatives

The guiding principle in teaching anything new, is to draw on your students' previous experiences and relate this to the new concepts. In teaching binary numbers (or any non-denary number system), this usually means demonstrating the concepts in base ten before introducing the binary equivalent.

In anticipation of the variety of "negative" notations that will be encountered while studying computer architecture, it is instructive to see how many different ways your students already know for representing negative values. Write a negative number on the board: **-456**, for example. You could mention that this format is often referred to as "sign-and-magnitude" because the two attributes of the numeric value are clearly distinguishable. Now ask the class what other ways this negative value could be written. With a little prompting (e.g. "What does your bank balance look like on your monthly statement...?"), you can usually build up the following list:

- ◆ 456- (Common in accounting reports as it makes the "exceptional" values stand out)
- ◆ (456) (A refund from the IRD?)
- ◆ 456DB (a Debit)
- ◆ 456DR (The British version of DB)
- ◆ 456OD (A typical bank statement?)
- ◆ 456CR (Have fun explaining why this means "negative")
- ◆ 456- (Red ink!)

2.2 Rewinding Speedos

Your class may think that they have exhausted ALL of the possibilities, but you should point out that computers use even more "exotic" techniques for representing negative values. Indeed, in a common everyday situation, yet another method of representing negatives exists, which mirrors the computer's two's complement technique. I refer, of course, to the odometer "mileage" recorder found in every car. Put a scenario to your class - 'Imagine that you have just bought a new car, and taken it on a long trip. On returning from your journey, you decide that you don't want to keep it. So you decide to "rewind the speedometer" in an attempt to make the look "as good as new"! But, by accident, you rewind the odometer one revolution past zero. What do you think the "speedo" will read now?' Over many years of putting

this question, I have yet to find a class that didn't reply "999999" (assuming that the odometer has 6 digits).

This immediately establishes a most useful result: As turning the odometer backward by one turn is equivalent to subtracting 1, we have established that $000000 - 1 = 999999$. Of course, everyone knows that $0 - 1 = -1$, so we have $99999 = -1$.

To assist this discussion, I usually employ a visual aid, as depicted in figure 1.



Figure 1. The Decimal Handy Calculator.

This hand-held mechanical calculator is basically a counter which adds a "1" by pressing the button over the appropriate digit "window". By pulling the top lever to the right of the display, you put the machine into "reverse" – subtract mode. With a small positive number loaded in the units column, put the machine into "reverse", and press the units button to successively reduce the counter to zero And then ask the class to predict what the next iteration will produce.

Continuing with this line of argument, the class will quickly establish that:

- ◆ $999998 = -2$
- ◆ $999997 = -3$
- ◆ $999996 = -4$

At this point students are usually not convinced that the "string of 9's" notation is a valid represent of a negative number. So, try some simple addition problems: e.g. $-4 + 6 = +2$. (Hopefully, the class will be able to give you this result. If not, draw a number line on the board and demonstrate this by adding the number "vectors") Now do the same problem, but this time using the "string of 9's" format for -4 : e.g. $999996 + 6 = 1|000002$. Point out that because the odometer has a fixed number of digits (as in a digital computer), the leftmost "1" cannot be stored, leading immediately to the result $999996 + 6 = 000002$.

If your class is still not convinced that this technique is valid, you can try adding some negative numbers: e.g. $-3 + -4 (= -7)$. Using “speedo” arithmetic: $999997 + 999996 = 1999993 = 999993$

Ask the class what the answer means. Alternatively, you can claim that the result is -7 , and prove it by adding 7 to 999993 to get zero. At this point, you can further “legitimise” this new number system by giving it a name – “ten’s complement” (If it has a name it MUST be “legitimate”!).

An even better aid is an old mechanical calculator, depicted in figure 2.

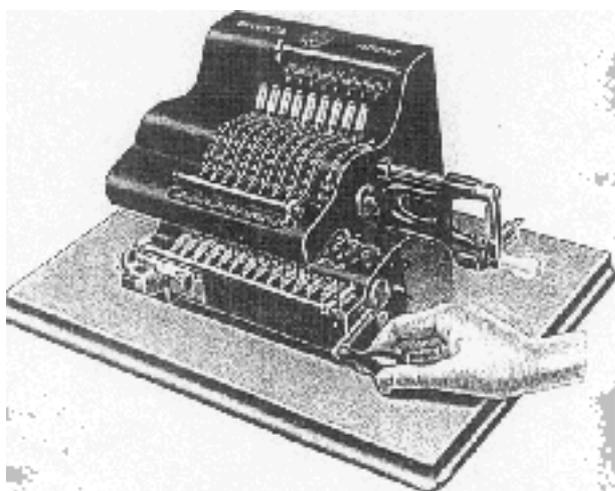


Figure 2. A Mechanical Calculator.

This machine can be used to demonstrate the ten’s complement much more easily, and has a bell which rings as you pass zero, giving a much more dramatic effect! Time permitting, you can digress into “computing history”, and demonstrate multiplication, division, and even extraction of square roots using the shift register.

The main drawback of both of these devices is their size – too small for all but the front row of your class to see. Nevertheless, you can get some of your students to verify the results displayed. Another option is to develop a simple Visual Basic or Delphi application which simulates the odometer. This can be used with a datashow projector for a classroom demonstration, or students may run the application independently as an exercise. Of course, the “soft odometer” can be programmed to emulate any number base system, so switching from decimal to binary, octal or hexadecimal complements will be a bonus.

3. FORWARD TO SUBTRACTION

Before leaping into binary subtraction, it is wise to review subtraction in base ten. Do a simple 4 digit subtraction on the board which does not involve “borrowing” e.g. $5684 - 3251$. After completing the problem, ask the class why it was so easy to do.... Clearly, all of the digits being subtracted were small enough to avoid the complex “borrowing” procedure which (hopefully), your class is familiar with. Now try a harder example: e.g. $5123 - 4678$. This time warn the class that you will solve the problem TWICE, using two different methods. Instruct them to follow your steps and decide which method corresponds most closely to their method of subtraction.

Method 1: (Decomposition)

This method, I believe, is now the most commonly taught subtraction method in New Zealand schools. Previously (thirty years ago, perhaps), method 2 – “equal additions” - was the standard method. Both methods start from the rightmost digit. With decomposition, because you can’t “subtract 8 from 3”, you borrow from the “2” in the ten’s column of the subtrahend. This leaves a “1” in the ten’s column, but gives you “13” in the units column. In effect, you are using the place value system to “decompose” the number you are subtracting from.

Method 2: (Equal Additions)

Starting from the right again, this method adds a “1” to the ten’s column of the minuend, and ten 1’s are added to the units column of the subtrahend. Clearly, equal amounts are being added to the numbers being subtracted, so the net result is not affected. Unfortunately, this technique gained a bad reputation as the “rule of thumb” that often accompanied the method – “BORROW a one from the bottom line, add it to the top then “PAY BACK the bottom line ...” – didn’t make much sense when you tried to analyse why the method worked. Perhaps this was the reason why the method is no longer taught (or was it to conform with American written textbooks?).

Having demonstrated both methods, survey the class to see the proportion of “decomposers” vs “equal adders”. Over the years, I have found that the number of “equal adders” has decreased to the point where I have only one or two of them in my class. You may of course find someone who uses a totally different method (get them to demonstrate to the class).

It may be of interest to find out where the “equal adders” went to Primary school – and WHEN (but that may be considered an invasion of privacy!).

The “punchline” is that no matter which of the above methods your students prefer, they will need to learn yet another method to understand how computers subtract – everyone starts off on an equal footing. Well, almost.... As I favour the decomposition method, and there are very few “equal adders” left, I tend to explain subtraction in terms of decomposition. It may help the “equal adders” to recall feet and inches, and/or pounds-shilling-and-pence calculations (they may be old enough!). If so, they probably did these subtractions using the decomposition method. I find it hard to believe, but students find it very difficult to follow the logic of the decomposition method if they have been taught equal additions (and vice versa).

4. SUBTRACTING SIGNED NUMBERS

A further step towards two’s complement subtraction is to revise subtraction of signed numbers. Ask the class to consider simple subtraction of integers: e.g. $3 - -4$, $-3 - -6$, ... Some members of the class will be able to give correct answers, but many will not! The objective here is to establish that “subtracting a negative value is equivalent to adding a positive”; i.e. $3 - -4 = 3 + 4$

To help the “numerically challenged” in the class, (and perhaps to add a bit of comic relief) a number-line simulation of the above problem can be useful. In this simulation, YOU represent the numbers being added/subtracted. The rules are as follows:

- ◆ If I am adding/subtracting a positive number, I will face right
- ◆ If I am adding/subtracting a negative number, I will face left
- ◆ When adding, I will walk forward (one pace = +1)
- ◆ When subtracting, I will walk backwards (one pace = -1)

Now draw a number-line on the board ($-8 \rightarrow +8$) so that each integer is positioned at approximately 50cm (or 1 pace). To demonstrate $3 - -4$, position yourself at +3 (facing right as you are a positive number). To subtract -4 , face left (to become a negative number), and walk backwards 4 paces to complete the subtraction. By keeping a close eye on your number-line, you should arrive at +7.

You should try a variety of problems (e.g. $3 - 5$; $3 + -5$; $-2 - -5$; $-4 - -3$; $-4 + 3$) in order to reinforce the more general rule “Subtraction is equivalent to adding the opposite of ...”. In other words, you can always turn a subtraction problem into an addition.

5. SUBTRACTION USING THE TEN’S COMPLEMENT

5.1 Finding the Ten’s Complement of a Number

If you haven’t already done so, you need to demonstrate how to find the ten’s complement. Formally, this is done by subtracting the number (let’s say 5432) from 10000. This, of course, leads to a complex subtraction. Having completed that you could suggest an easier method – subtract from a string of 9’s and then add 1 – MUCH easier. The name “nine’s complement” may be introduced at this point so that an analogy with the “one’s complement” can be drawn later.

Some teachers may prefer to start with the nine’s complement, and do some subtraction examples before moving to the ten’s complement. Such an approach is taken by Englander (2000). This approach involves introducing an “end-round-carry” into the subtraction algorithm, which I would prefer to avoid.

5.2 The Negative of a Negative is Positive

You should now demonstrate the (obvious?) fact that $-(-N) = N$, using a ten’s complement number.

e.g. $10\text{comp}(10\text{comp}(N)) = 10\text{comp}(10000 - N) = 10000 - (10000 - N) = N$

(Using a specific value is probably more effective than doing the algebraic manipulation)

5.3 Examples

Now repeat the examples done in section 4, but use the “ten’s complement” to replace any negative numbers. e.g.

- ◆ $3 - 5 = 3 + -5 = 3 + 10\text{comp}(5) = 3 + 99995 = 99998$.
(Looking up the result established earlier, it is easy to verify that this is the value -2)
- ◆ $-2 - -5 = 99998 - 99995 = 99998 + (99999 - 99995 + 1) = 99998 + 5 = 1|00003$.

At this point you can suggest that the leftmost digit represents the “sign” of the number. Thus a “0” signals a positive value, and a “9” indicates a negative value. (Strictly speaking, when the leftmost digit is ≥ 5 , the number is considered to be negative so that the bottom half of the number range is positive, and the top half is negative. However, you may not want to mention this, unless a “bright student” queries this point).

This example also highlights the need to establish a fixed length register to ensure that the leftmost “1” is truncated. The truncation sometimes worries students, as they cannot see why the technique should be mathematically sound. The truncation effect can be demonstrated to be mathematically valid using a simple numerical example:

$$\text{e.g. } 6 - 7 = 6 + (10000 - 7) - 10000 = -1$$

The last step of subtracting 10000 is the “truncation”. Clearly this is merely compensating for adding 10000 to the minuend during the complementation process.

Finally, before leaving base ten, you may want to impress your class with your mathematical “genius” by means of a simple “party trick”. Ask two members of your class to give you a six digit number. Let’s say these are 125684 and 476934, which you write down on the board. You now add a (carefully chosen) number of your own – for this example it will be “874315”. Ask for another number from the class (let’s say 367893), and you respond with “523065”. The number sequence is now:

125684 <- student number 1 (SN1)

476934 <- student number 2 (SN2)

874315 <- My number 1 (MN1)

367893 <- Student number 3 (SN3)

523065 <- My number 2 (MN2)

You now ask the class to add up numbers as quickly as they can. But, before they have time to pick up their pen (or fetch their calculator), you announce the sum as “2367891”! So how does it work? If you look at MN1, you will note that I have chosen it to be the 9’s complement of the SN1. MN2 was chosen to be the 9’s complement SN2. So, when adding this column of figures, my two numbers will effectively cancel out the first 2 student numbers, leaving only SN3. But, the complement process will generate two extra 1’s in the seventh column – hence the leading “2” in my answer. Finally, you will note that my answer is actually 2 less than SN3. This is the result of using the 9’s complement rather than the ten’s complement. If had used the 10’s complement, for my values, then the answer would have been identical to the SN3 (except for the leading “2”). But that might make the technique a little too obvious.

If you can complete this trick without making a blunder, you may be able to convince your class that the complementation method really does work!

6. SUBTRACTION IN BINARY

6.1 Binary Subtraction Using the Decomposition Method

Before introducing the two’s complement method, you should demonstrate that the “normal” subtraction algorithms (decomposition/equal additions) work in binary. After doing a simple example – e.g. 10111 - 01101, where only a single “borrowing operation” is required, you could try a more complex one where multiple “borrowing” is required – e.g. 100010 - 001111. Before tackling the second example, you may want to do a similar example in base ten (e.g. 200000 - 12345). Using the decomposition method, the first digit subtracted, causes a “ripple effect” on the subtrahend, effectively converting 20000 to 199990 + 10.

When you do this in binary, a similar “ripple effect” occurs, but you will get a string of “1s”. Students find this most confusing., Decomposing the binary value 10000, quickly becomes a sea of ones and zeros overlaid with crossing out, as the borrowing operation ripples across the subtrahend. Doing one example of this type is enough to convince most people that there MUST be a better way! And there is.... (The “equal additions” lobby – if any are present – will proudly announce that this is not a problem at all.) Ignoring such claims, you should now point out the “real” method of subtraction – the one used in computer systems – avoids all “borrowing” complexities.

6.2 Subtraction Using the Two’s Complement Method

6.2.1 What is a Two’s Complement?

By analogy with the ten’s complement method used previously, you may assert that a similar method exists for base two. Hopefully, your class will “guess” that the analogous method is the “two’s complement” method! If, not, spell it out for them. The motivation for this new method is three-fold:

- ◆ It will simplify subtraction by avoiding the complex borrowing operations.
- ◆ Computers don’t have subtraction circuitry - they convert subtractions into “addition of the complement”. To understand how computers work, you need to understand complementary arithmetic.
- ◆ Two’s complement computers do not need to have special circuitry to define how mixed sign arithmetic should be processed. The well known results for addition, subtraction, multiplication, and division are handled correctly (assuming overflow does not occur of course).

The first step, is to show how to get a “two’s complement” by subtracting a binary value from 10000 ...

This is, of course, even harder than finding the ten's complement. By reminding students of the nine's complement technique, the class may be able to suggest the "one's complement" method – subtract the value from a string of '1s and add 1". Once this is established, see if the class can "discover" that finding the one's complement is equivalent to reversing the bit values.

6.2.2 Two's complement Examples

Try a few examples involving addition and subtraction of signed numbers: e.g. $2 - 5$; $-2 + -3$; $-3 - -7$

Complete these in base ten, and then convert each problem into two's complement binary format. In each case, verify that the correct result has been obtained by comparison with the base ten results.

7. MULTIPLICATION USING TWO'S COMPLEMENT

The proof of any scientific theory is to demonstrate that it works in all cases. So, if the two's complement method of storing negative numbers is valid, it must be able to handle multiplication of signed numbers. Try a few multiplications: e.g. -5×4 ; 6×-4 ; $-3 \times -4 \dots$

Hopefully, the class will be able to tell you the answers to these problems in base ten. (If not, you may need to go back to the number line and demonstrate the effect of multiplying by -1 as being equivalent to rotating a unit vector through 180 degrees on the number line.)

For these examples, restrict your "register" to 7 or 8 bits, otherwise the calculations become extremely tedious (and error prone). The final example - demonstrating that two's complement arithmetic can take two negatives and produce the correct positive result, should convince most students that the method is sound.

8. CONVERTING NEGATIVE TWO'S COMPLEMENT VALUES TO BASE TEN.

Another stumbling block for many students is interpreting a negative two's-complement number. When faced with interpreting "10001011", many will assume that the result is either a large unsigned value, and convert the binary directly to base ten (147), or, interpret this as a "sign-and-magnitude" value, and convert it to a small negative number(-11). The standard method of "find the two's complement" and then convert to base ten, doesn't seem

to come to mind! For some students, an alternative method may be more successful. For small numbers (up to 8 bits), many student convert to base ten by writing the powers of two above each digit. For the value mentioned above, the powers of two would be 128, 8, 2 and 1 – if the number were unsigned. But, for a 2's complement number, you may assign a weight of -128 to the leading digit, leaving all of the others as they were. The conversion then becomes $8 + 2 + 1 - 128 = - 117$

You should note that this technique can also be employed if you are using the nested method for conversion: e.g. $10001011 = ((((((-1 \times 2 + 0) \times 2 + 0) \times 2 + 0) \times 2 + 1) \times 2 + 0) \times 2 + 1) \times 2 + 1$

9. COMPLEMENTS IN OTHER NUMBER BASES

All of the concepts and techniques relating to base ten and binary can equally be applied to other number bases such as octal and hexadecimal. I tend not to venture into eight's and sixteen's complements unless a student raises the issue. Certainly, subtraction can be achieved in octal and hexadecimal, using the decomposition method – more easily than in binary. Complementary subtraction, is however, harder, as you cannot just "invert the digits ..." as you would in binary. The preferred method is to find the fifteen's complement (or is that the "F's" complement?) and add 1 to get the sixteen's complement. Also, determining whether a number is positive or negative is not simply looking at the leftmost digit to see if it is zero.... For students who prefer the complementation method over decomposition, you could suggest that they convert the octal or hex to binary, thereby avoiding the complexity of yet another type of complement.

10. CONCLUDING COMMENTS

The concepts covered in this paper are presented to DT100 classes at AUT over 2 x 1 hour sessions. Hopefully, some of the techniques mentioned may be of use in your teaching. The author would be interested to hear of other techniques that have been used in teaching this topic, and how successful they were.

11. REFERENCES

Englander, I. (2000) "The Architecture of Computer Hardware and Systems Software: An Information technology Approach", second edition. John Wiley & Sons, New York.