Binary Abacus Instructions

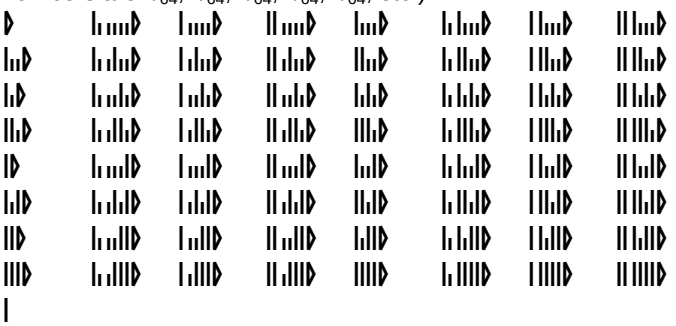
A binary abacus is a table-top device that can be used to perform complicated numerical calculations manually. Because this digital calculator has no internal parts, the operator is completely aware of how every calculation is performed. Use of the binary abacus, therefore, greatly enhances ones understanding of numerical calculations.

The Binary Number System

The binary abacus makes use of the Binary Number System (or binary for short) as opposed to the more conventional Decimal Number System (or decimal for short). First, let me show you some binary numbers. Here are consecutive whole numbers beginning with zero. (In decimal these are 0, 1, 2, 3, 4, etc.)

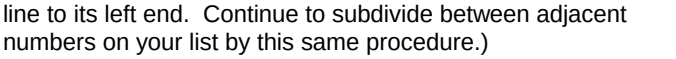
They may appear to be complete gibberish, yet they are indeed numbers. They follow a well defined pattern. Take some time to study the pattern. It may help you to write these numbers for yourself. (To follow the pattern from one number to the next, do as follows. Make the first short line tall and all lines to its left short. Everything else remains the same. If there are no short lines, you must imagine one on the right end. Notice, zero does not fit into the pattern.)

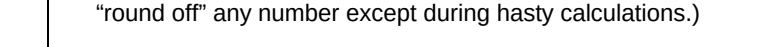
Now, let me show you some fractions. Here are some numerically ordered equally separated fractional numbers beginning with zero and ending with one. (In decimal, these numbers are $^0/_{64}$, $^1/_{64}$, $^2/_{64}$, $^3/_{64}$, $^4/_{64}$, etc.)

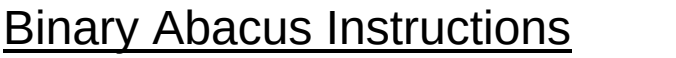


Take some time to study this pattern. This pattern may be more difficult to grasp. (To follow the pattern, do as follows. Make a list with zero at the beginning and one at the end. Now produce the number exactly in-between as follows. Copy the first number. Add short lines to its left end until it has at least as many symbols as the second number. Finally add one tall line to its left end. Continue to subdivide between adjacent numbers on your list by this same procedure.)

Now let me show you how to write mixed numbers. Mixed numbers can be written by simply placing the fractional part on the left and the whole part on the right. Here are some examples.





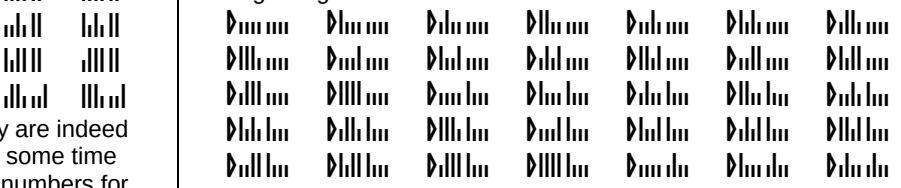


Now let me show you the guiding principles behind the representations you have seen so far. This should make things much simpler.

In the non-zero whole numbers shown thus far, the triangle () had been omitted. Every number, in fact, has one and only one triangle. When not written, the triangle is understood to be at the left end. For example, actually represents (and can be written as) .

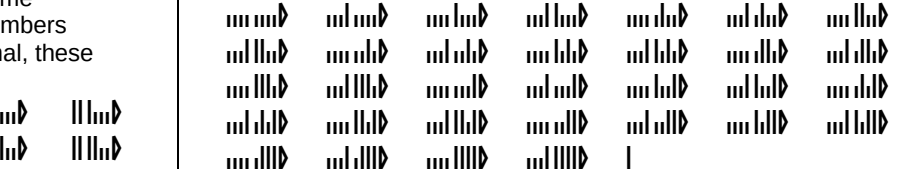
All of the numbers seen thus far have an endless series of short lines on both ends. All of these were omitted. Thus  represents ……and represents ……. Unless it is specifically noted otherwise (as you will see later) one is always to understand that there are an endless series of short lines on both ends of a number. Thus can be written as , , or (among others).

Now lets look at some numbers again and see if they make more sense. Once again, here are consecutive whole numbers beginning with zero.



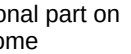
Notice how the line on the right of the triangle changes every time and all of the other lines change only when the line on their left changes from tall to short.

Also again, here are some numerically ordered equally separated fractional numbers beginning with zero and ending with one.



Notice how this pattern is similar to the last. As you hopefully can now see, the lines represent binary digits, the triangle (place holder) identifies the unit digit, and each digit carries twice the weight of the digit on its left. This is the simple beauty of the Binary Number System. Now let me show you a few notional conveniences.

When a number is too long to write conveniently, excessive digits can be replaced with a truncation mark (-). For example,  can be written as or (among others) and as or (among others). A truncation mark is always understood to represent an endless series of unidentified digits. Of course, using the truncation mark does not communicate the number exactly. (Note, one should never "round off" any number except during hasty calculations.)

Additionally, some numbers do not have an endless series of short digits on both ends. Such numbers can still be written exactly if a repeating pattern can be identified. For example, …  can be written as . When a group of digits is underlined it is understood to repeat endlessly. Here are some more examples.

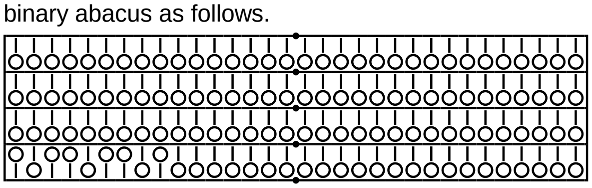
If no repeating pattern can be found, one must use the truncation mark as before. As you have hopefully noticed, clumping digits into groups of four makes them much easier to read.

As promised, I will now show you how to write negative numbers. To reverse the sign of a number (make a positive number negative or vise versa) simply reverse all of its digits. Thus the opposite of [illegible] is [illegible], of [illegible] is [illegible], and of [illegible] is [illegible].

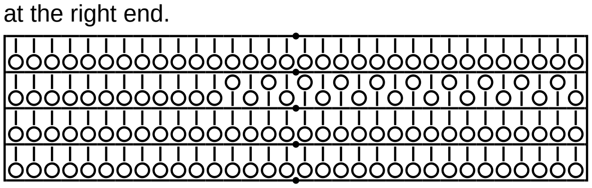
Finally, I would like to touch on how one might say these numbers. The two binary digits could be named any of many things (off and on, down and up, zero and one, yin and yang, black and white, short and long, or small and big). However, to facilitate quick pronunciation other syllables must be chosen.

Putting Numbers into your Binary Abacus

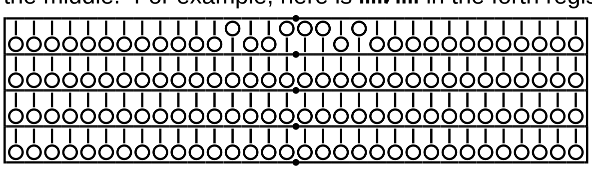
Your binary abacus has four registers (rows), each for storing one number. Each register has thirty-two digits (beads). The digits of a register correspond to the digits of a binary number. For example, [illegible] can be stored in the first register of your binary abacus as follows.



In this example, the place holder is at the left end of the abacus. The place holder goes at the left end when working with only whole numbers. When working with only fractional numbers, it goes at the right end. For example, here is [illegible] stored in the third register with the place holder at the right end.

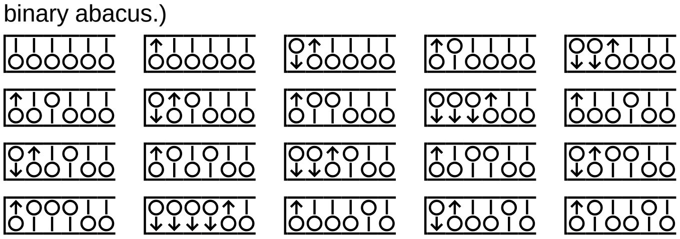


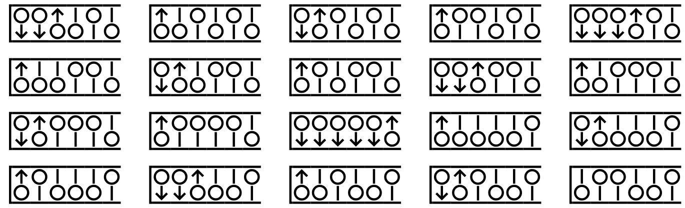
When working with mixed numbers, the place holder goes in the middle. For example, here is [illegible] in the forth register.



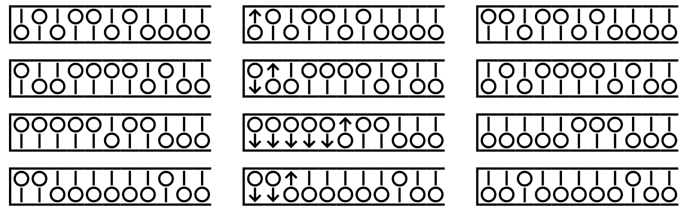
Counting on a Binary Abacus

First, let's learn how to count using a binary abacus. The following illustrates counting beginning with zero. (It is recommended that you perform all illustrated exercises on your binary abacus.)



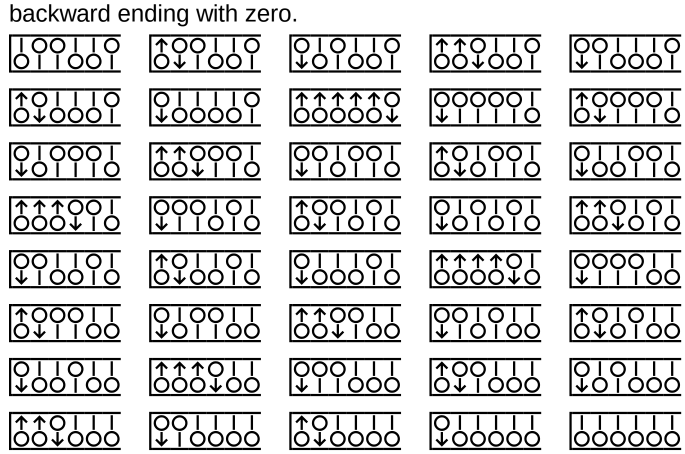


As you will notice, these are consecutive whole numbers. The general procedure for counting (adding one) is as follows. Begin with the unit digit. If this digit is down, simply reverse it and you are finished. If this digit is up, reverse it, move to the next digit (to the right), and repeat. Here are some more examples (the first, [illegible] + one = [illegible]).

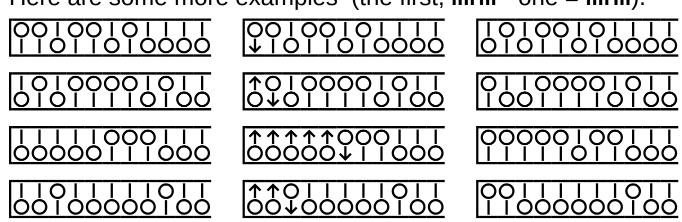


Counting Backward on a Binary Abacus

To count backward (subtract one) on a binary abacus, simply use the procedure for counting forward except exchange the words "up" and "down". The following illustrates counting backward ending with zero.

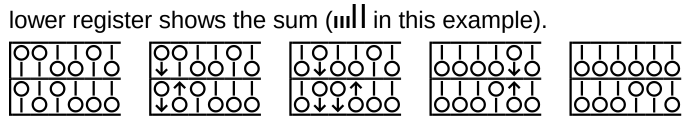


Here are some more examples (the first, [illegible] - one = [illegible]).

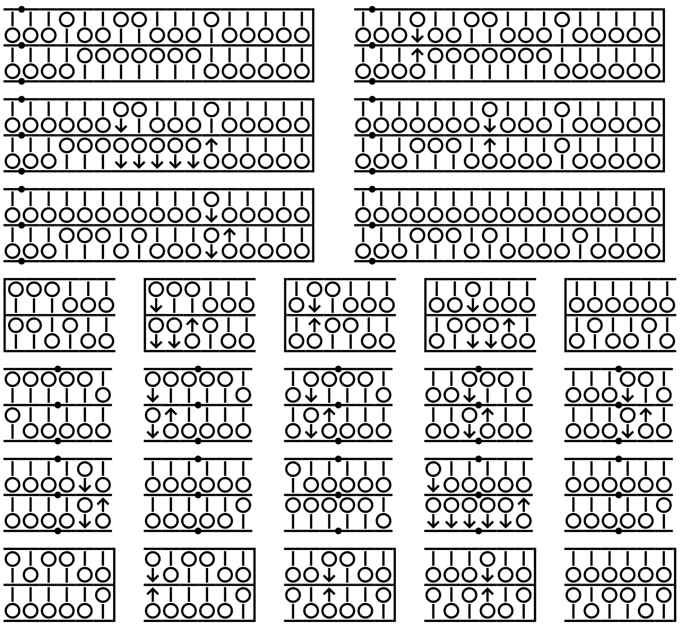


Addition on a Binary Abacus

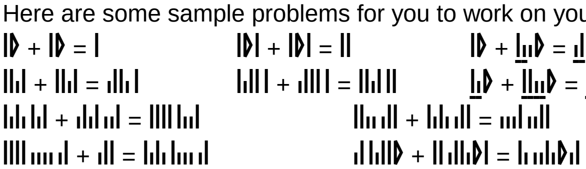
Now, lets learn how to add with a binary abacus. When adding on a binary abacus, two registers are used. (See the illustration below.) First, the numbers to be added ([illegible] and [illegible] in this example) are entered into the registers. Next, each up digit in the upper register is reversed using the "L-shaped maneuver". Finally the upper register shows zero, and the lower register shows the sum ([illegible] in this example).



After each step, the sum of both registers remains unchanged. To perform the "L-shaped maneuver" do as follows. Reverse the up digit in the upper register. Then, perform the "add one" procedure beginning with the same digit in the lower register. Here are some more examples (the first, + =).

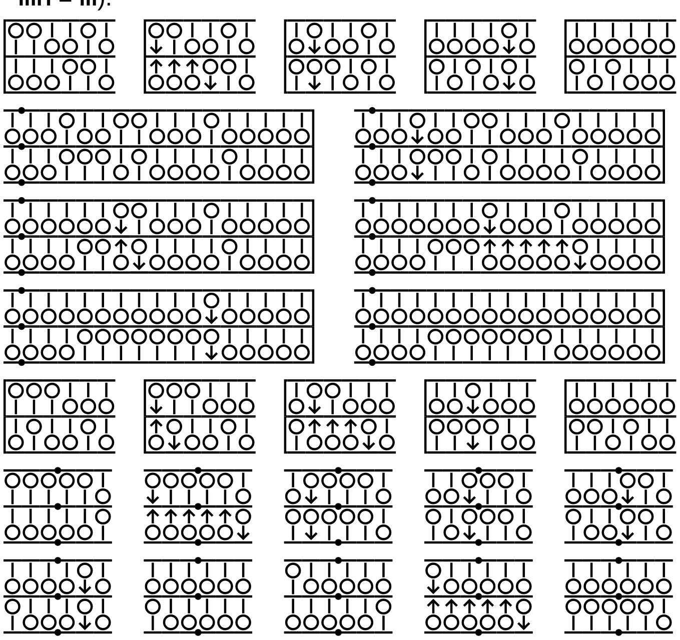


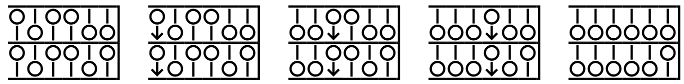
Here are some sample problems for you to work on your own.



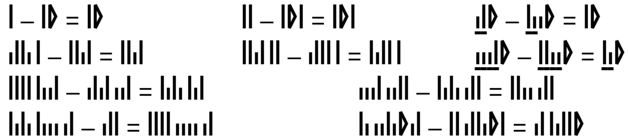
Subtraction on a Binary Abacus

To subtract on a binary abacus, simply use the procedure for adding except during execution of the "L-shaped maneuver" replace the "add one" procedure with the "subtract one" procedure. Also, make sure that the number to be subtracted is in the upper register. Here are some examples (the first, - =).



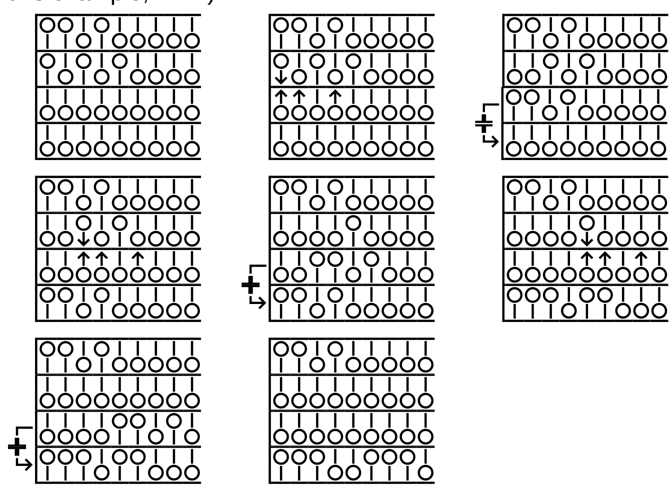


Here are some sample problems for you to work on your own.

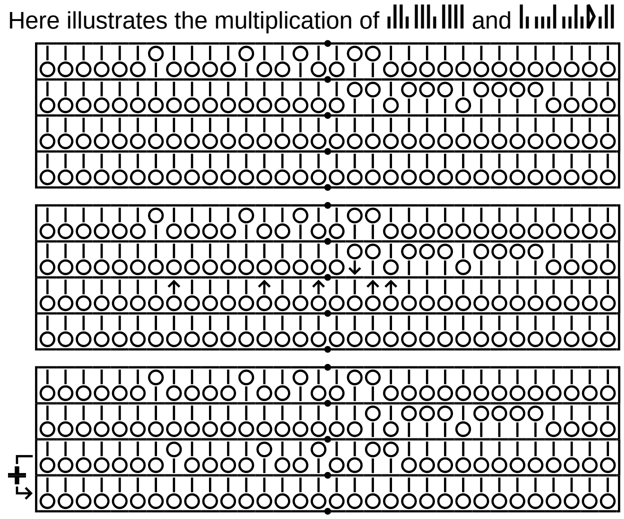


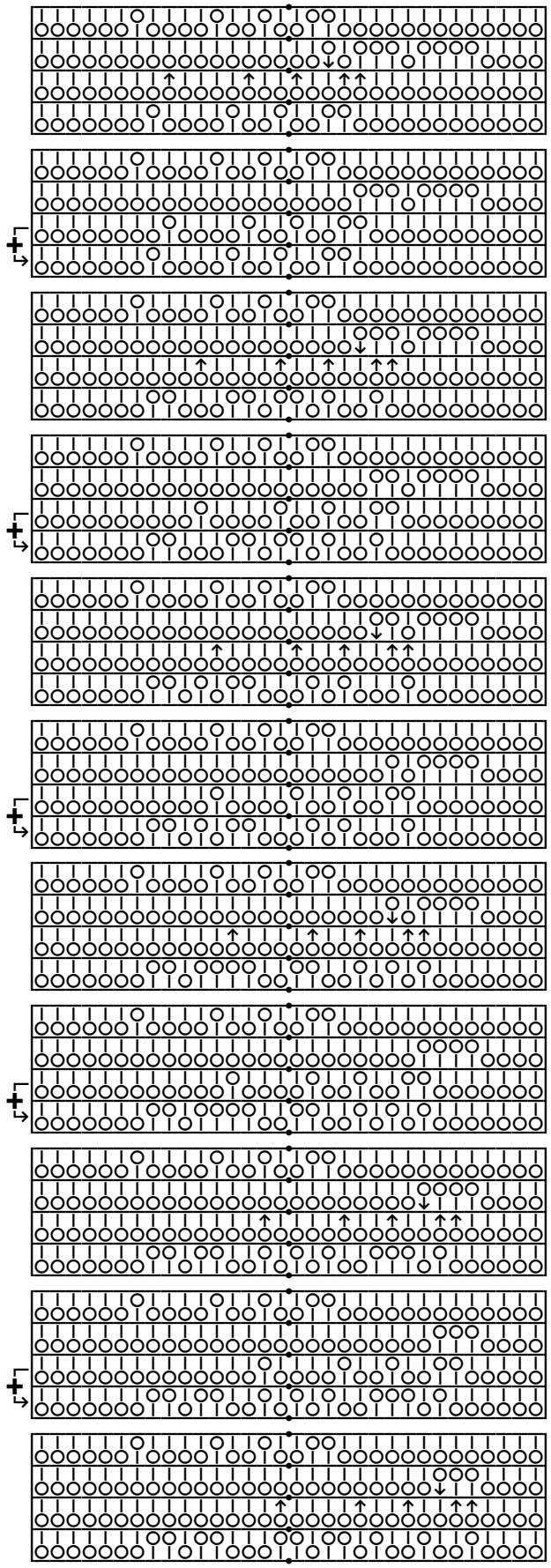
Multiplication on a Binary Abacus

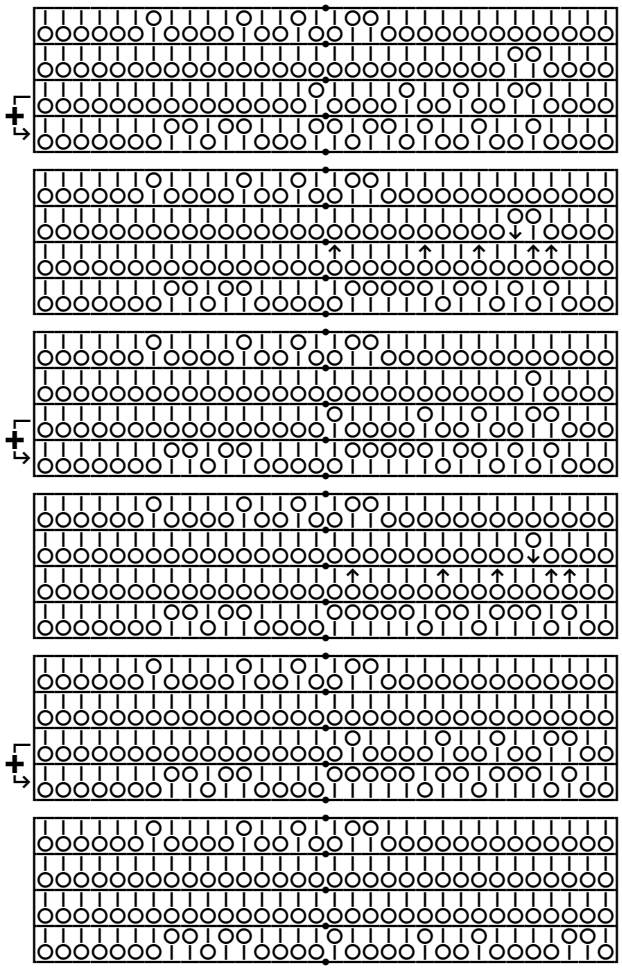
Multiplication on a binary abacus requires four registers. The guiding principle is that the sum of the first register, the second register, and the product of the third and forth registers remains unchanged after every step. Here is how it works. (See the illustration below.) First, the numbers to be multiplied (in this example, and) are put in the third and forth registers. The first and second registers should begin at zero. Next, each up digit in the third register is reversed using the "T-shaped maneuver" (a two step process). Finally, the third register shows zero and the first register shows the answer (in this example,).



The "T-shaped maneuver" requires two steps. The first step is as follows. An up digit in the third register is reversed and into the second register is copied the contents of the forth register with the unit digit directly below the digit just reversed. For the second step, the second register is added to the first register. Here illustrates the multiplication of and







As you can see, the answer is .