Decimal/Binary Conversion on the Soroban

Conversion of a whole number from decimal to binary

This method uses successive divisions by two, in place, utilizing a simple right-to-left algorithm. The division algorithm works as follows:
* Start with the rightmost rod and work toward the left
* If the number on the current rod is even, subtract half that number from the current rod.
* If the number on the current rod is odd, decrement it by one, move the heaven bead on the right neighbor rod to the counter beam, and subtract half the remaining number on the current rod.

The conversion method is to check whether the decimal number is odd or even and set the next binary digit (bit) of the binary number to a one or zero accordingly. The decimal number is then divided by 2 and checked for odd or even and the next more significant bit of the binary number is set, repeating until the decimal number has been reduced to zero. Note that any time the decimal number is odd, when the rightmost rod is decremented to make it even, it is not necessary to set the heaven bead on the right neighbor (to the right of the unit rod), because that heaven bead would be cleared immediately after setting the corresponding "one" bit in the binary number Because we may have strings of multiple consecutive zeros in the binary number, we use the heaven bead as a marker to mark a completed bit and the top earth bead is used to indicate whether that bit is a one or zero.

Example: convert 485 to binary

Clear the soroban and set 485 on the left side with the 5 on a units rod.
4850000000000
The number is odd, so set the least significant bit of the binary number to one and set the heaven bead as a completed bit marker. Decrement the decimal number to make it even.
4840000000006
Now divide by 2
2420000000006
This is even, so set only the marker bead on the next binary bit
2420000000056
and divide the decimal number by 2
1210000000056
Continue in this manner
0600000000656
0300000005656
0150000055656
0070000655656
0030006655656
0010066655656
0000666655656
Now clearing the marker beads, the binary number is
0000111100101 (1E5 in hexadecimal or 745 in octal).

Conversion of a whole number from binary to decimal

This method uses successive multiplications by two, adding the next less significant binary bit between each multiplication. In order to speed up the process, the binary number is first converted to an octal number and each multiplication is by eight rather than two, with the next octal "digit" (character) added between each multiplication. For those not familiar with octal, each group of three binary bits can be represented by an octal number between 0 and 7, as follows:
000 = 0
001 = 1
010 = 2
011 = 3
100 = 4
101 = 5
110 = 6
111 = 7

It is easiest to perform the multiplication by eight "in place" by adding seven times the current decimal number to itself.

Example: convert 111100101 to decimal
Grouping this number and converting to octal, we have 111 100 101 = 745 octal

clear the soroban and enter the octal number on the left side
7450000000000
Add the most significant octal character to the decimal number on the units rod and clear the octal character
0450000000007

Add 7*7 to the decimal number (to multiply the 7 by 8)
0450000000007
+ 49

0450000000056

Add the next octal character (4) to the units rod and clear it from the octal number
0050000000060

Add 7*60 to the decimal number
0050000000060
+ 42

0050000000480
+ 00

0050000000480

Add the next octal character and clear it
0000000000485
finished

Conversion of a decimal fraction to a binary fraction

This method uses successive multiplications by two, using the "overflow" to the units digit to determine the next bit. We will use octal here, as in the previous method in order to speed up the conversion. The algorithm is to multiply the fraction by 8, in place, take the units digit as the next less significant octal character, then clear the units digit - repeat until you have as many octal characters as you want (in general, the octal number is infinitely long).

Example: convert 0.485 decimal to octal.

The decimal number is specified to three significant places, or an accuracy of +/- 1/2 out of 485 = +/- 1 part in 970, so we only need the same level of accuracy in the octal number. Four octal characters are more than sufficient for this, so we will convert to five octal characters and then round off to four.

Clear the soroban and enter the decimal fraction on the left with the most significant digit just to the right of a unit rod.
0004850000000

multiply by 8
00048500000000
+ 28

00328500000000
+ 56

00384500000000
+ 35

00388000000000

Overflow to the units digit is 3, so take that as the first octal character (placing it just to the right of a unit rod), then clear it from the decimal number
00088000030000

multiply the decimal number by 8
00088000030000
+ 56

00648000030000
+ 56

00704000030000

Overflow to the unit rod is 7, which becomes the next octal character
00004000037000

multiply by 8
00004000037000

+ 28

00032000037000

No overflow, so the next octal character is 0

multiply by 8
00032000037000
+ 21

00242000037000
+ 14

00256000037000

Overflow is 2, place it to the right of the previous 0 in the octal number
00056000037020

multiply by 8
00056000037020
+ 35

00406000037020
+ 42

00448000037020

The next (and last required) octal character is 4
00048000037024

The last character is 4; i.e., 4/8, which will cause the 2 to be rounded up to 3. The rounded number is then 0.3703 octal, or 0.011111000011 binary.

Conversion of a binary fraction to a decimal fraction

The method here is to use successive divisions by two and adding the next most significant bit of the binary fraction between each division. We can speed this up by using octal, with division by 8 for each octal character (but see the note below). We start by adding the least significant octal character to a unit rod, then divide in place by 8. Then add the next most significant octal character and divide again, and continue until all characters have been added, ending with a division by 8 after adding the last character.

Note - because division by 2 in place is so simple, it seems easiest to replace each division by 8 with three divisions by 2.

Example: Convert 0.3703 octal to decimal.

As a decimal digit has more resolution than an octal character, 4 decimal digits should provide sufficient accuracy. This means we don't have to continue the divisions beyond that point, greatly reducing the work required for the conversion.

Clear the soroban and enter the octal number to the left
3703000000000
Add the least significant character to a unit rod and clear it from the octal number
3700000300000
Now divide by 2 in place 3 times (divide by 8)
3700000150000
3700000075000
3700000037500

Add the next octal character (0) and divide by 8 again
3700000037500
3700000018750
3700000009370 Note - from this point on we are not keeping track of digits which overflow to the right of the 7
3700000004680

Add the next octal character and divide by 8 again
3000000704680
3000000352340
3000000176170
3000000088080

Add the last octal character and divide by 8
0000000388080
0000000194040
0000000097020
0000000048510

The decimal number is 0.4851

Steve Treadwell
September, 2011