Concepts of Conversion of Base-2 Floating Point Binary Number to a Base-10 Decimal Number

Sean Rodby, Luke Snyder, Autar Kaw
University of South Florida
United States of America
kaw@eng.usf.edu
Website: http://numericalmethods.eng.usf.edu
Version: Mathcad 14; Last Revised: August 28, 2008

Introduction

The following worksheet illustrates how to convert a base-2 floating point binary number to a base-10 decimal number using loops and various conditional statements. The user inputs total number of bits, decimal sign bit value, exponent sign bit value, mantissa bits entry, and exponent bits entry in the *Input* section of the program. The program will then convert the floating point binary number into a decimal number.

Section 1: Input

This is the only section where the user interacts with the program.

The generalized formula for floating point format is given by $y = \sigma \cdot m \cdot 10^{e}$

- Enter the total number of bits

  tot_bits := 9

- Enter the sign of the number: '0' if the number is positive, '1' if the number is negative.

  dec_sign := "0"

- Enter the sign of the exponent: '0' if the number is positive, '1' if the number is negative.

  exp_sign := "0"

- Enter the mantissa bits

  Mant_bits := "1011"

- Enter the exponent bits

  Exp_bits := "101"

Section 2: Procedure

We must first check to see if the number of bits entered by the user corresponds with the total number of bits specified. If *Totalbits* is not equal to *tot_bits* then their is an error in the number of bits entered.

$a := \mathrm{strlen}(\mathrm{dec\_sign})$

$b := \mathrm{strlen}(\mathrm{exp\_sign})$

$c := \mathrm{strlen}(\mathrm{Mant\_bits})$

$d := \mathrm{strlen}(\mathrm{Exp\_bits})$

$\mathrm{Totalbits} := a + b + c + d = 9$

Using *if* statements to designate the signs of both the decimal number to be approximated and the exponent.

$$\mathrm{Dec\_sign} := \left| \begin{array}{l} 1 \ \text{ if } \ \mathrm{dec\_sign} = \text{"0"} \\ -1 \ \text{ if } \ \mathrm{dec\_sign} = \text{"1"} \end{array} \right.$$

$$\mathrm{Exp\_sign} := \left| \begin{array}{l} 1 \ \text{ if } \ \mathrm{exp\_sign} = \text{"0"} \\ -1 \ \text{ if } \ \mathrm{exp\_sign} = \text{"1"} \end{array} \right.$$

Here we initialize the *Mant_sum* value at one to account for the *1*2^0* term that always exists in the floating point approximations.

```
Mant_sum := | mant_sum ← 1
            | for  i ∈ 0 .. c − 1
            |   mant_sum ← mant_sum + str2num(substr(Mant_bits, i, 1))·$2^{-(i+1)}$
            | mant_sum
```

Using the loop to create a number array from the exponent character array.

```
Exp_sum := | exp_sum ← 0
           | for  i ∈ 0 .. d − 1
           |   exp_sum ← exp_sum + str2num(substr(Exp_bits, i, 1))·$2^{d-(i+1)}$
           | exp_sum
```

Determining the final base-10 number.

$$Final\_Dec\_Num := Dec\_sign \cdot Mant\_sum \cdot 2^{Exp\_sign \cdot Exp\_sum} = 54$$

Conclusion

This worksheet illustrates the use of Mathcad to convert a floating point binary representation to a base-10 number. Recall that floating point representation is used more often than fixed point representation due to two primary advantages: floating point representation supports a much larger range of values while maintaining a relative error of similar magnitude for all numbers.

References

Floating Point Representation.
See:
http://numericalmethods.eng.usf.edu/mcd/gen/01aae/mcd_gen_aae_txt_floating point.pdf