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: *Mathematica 6*; Last Revised: August 28, 2008

Initialization

Clearing the definitions of all symbols in the current context:

```
ClearAll[Evaluate[Context[] <> "*"]]
```

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 Data

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

- Enter the total number of bits

```
totbits = 9;
```

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

```
decsign = "0";
```

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

```
expsign = "0";
```

- Enter the mantissa bits

```
Mantbits = "1011";
```

- Enter the exponent bits

```
Expbits = "101";
```

This is the end of the user section. All information must be entered before proceeding to the next section. **RE-EVALUATE THE NOTEBOOK**.

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.

```
a = StringLength[decsign];
b = StringLength[expsign];
c = StringLength[Mantbits];
d = StringLength[Expbits];
Totalbits = a + b + c + d;
If[totbits ≠ Totalbits, Print[
  "nThere is an error in the number of bits entered. The total number of bits counted
does not correspond with the total number of bits entered by the user. Please make sure
the addition of all the bits specified equals the total number of bits specified."],
 Print["The total number of counted bits is equal to the number of total
    number of bits entered by the user. The simulation will now begin"]]
```

```
The total number of counted bits is equal to the number of
  total number of bits entered by the user. The simulation will now begin
```

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

```
If[decsign == "0", Decsign = 1;]
If[decsign == "1", Decsign = -1;]
If[expsign == "0", Expsign = 1;]
If[expsign == "1", Expsign = -1;]
```

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

```
Mantsum = 1;
Do[Mantsum = Mantsum + ToExpression[StringTake[Mantbits, {i}]] * 2^-i, {i, 1, c}]
```

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

```
Expsum = 0;
Do[Expsum = Expsum + ToExpression[StringTake[Expbits, {i}]] * 2^(d-i), {i, 1, d}]
```

Determining the final base-10 number.

```
FinalDecNum = Decsign * Mantsum * 2^(Expsign*Expsum)
```

```
54
```

Conclusion

This worksheet illustrates the use of *Mathematica* 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/nbm/gen/01aae/nbm_gen_aae_txt_floatingpoint.pdf