**First Name:** Johannes
**Last Name:** Öhlböck
**Date:** November 14, 2004
**Homework Number:** 1
**Homework Title:** Exercise 1.20

## Problem description:

What is the IEEE single-precision binary floating-point representation of the decimal fraction 0.1

**(a)** with chopping?

**(b)** with rounding to nearest?

## Problem solution:

Get the base-2 expansion of 0.1:

```
x = 0.1;
while post decimal positions ≠ 0 {
    x × 2;
    If x < 1
      Add 0 to the binary number;
    Else {
      Add 1 to the binary number;
      Shift the Integer digit 1 by 0;
    }
}
```

In our case the algorithm gives $000\overline{1100}$ while the last four digits repeat infinit times.

Getting the three parts of an IEEE SP floating point number:

**Sign:** The fraction 0.1 is positively so we use a binary 0 digit for the sign.

**Exponent:** The count of 0's from the head of the base-2 expansion is three. We add a decimal one for the first 1 and so we get minus 4 as exponent. To get the IEEE SP binary representation of this exponent we have to add 127 befor transformation. So

$$E = (-4) + 127 = 123 = (01111011)_2$$

**Mantissa:** We take the above representation of 0.1 and delete the 0's and the first 1 of the head (normalize).

**case a:** From the resulting binary number we cut out the first 23 digits and get for our Mantissa

$$M_a = (10011001100110011001100)_2$$

**case b:** We have to take a look on the 24th digit of the resulting binary number. If the digit is 0 we can use the first 23 digits as mantissa. If the digit is 1 and any other digit following is 1 we have to add a binary 1 to the last position of the first 23 digits (round up). If the digit is 1 and all other digits following are 0 we have a case of tie. In our Example we have:

$$M_b = (|10011001100110011001100|_{22}\, 110011...)_2$$

So the 24th digit is a 1 followed by another 1. In this case we have to add a binary 1 to the 23th digit of our mantissa and get:

$$M_a = (10011001100110011001101)_2$$

## Results:

| Method | Sign | Exponent | Mantissa |
|---|---|---|---|
| chopping | 0 | 01111011 | 10011001100110011001100 |
| rounding to nearest | 0 | 01111011 | 10011001100110011001101 |

## Discussion and Comments:

A comparison of the decimal values from the two binary numbers above with the frction 0.1 gives the following differences:

**a:** $(00111101110011001100110011001100)_{IEEE-SP} = (0.09999999403953552)_{10}$

**b:** $(00111101110011001100110011001101)_{IEEE-SP} = (0.10000000149011612)_{10}$

$$|0.1 - a| = 5.960464483090178 \cdot 10^{-9}$$

$$|0.1 - b| = 1.490116113833650 \cdot 10^{-9}$$

As we can see, the methode of rounding to nearest leads to a better approximation.