

Computer Architecture – Assignment 11

Bonus

Claudio Maggioni Tommaso Rodolfo Masera

1 Bonus Question 1

1.1 Decimal to Hexadecimal

All the numbers are going to be first converted to binary and then to hexadecimal for an easier calculation

a) **11**;

First we find how 11 is expressed in binary which is 1011 and then we convert it to scientific base 2 notation: $1011 = 1.011 \times 2^3$.

The sign bit is 0 since $11 > 0$ and therefore the number is positive.

To determine the exponent we take the single precision bias (127) and we add to it the amount of times we moved a bit past the floating point: $127 + 3 = 130$.

We then convert 130 to decimal by successive halving:

Fraction	Rest
$130/2 = 65$	0
$65/2 = 32$	1
$32/2 = 16$	0
$16/2 = 8$	0
$8/2 = 4$	0
$4/2 = 2$	0
$2/2 = 1$	0
$1/2 = 0$	1

By reading from most significant bit to least significant bit we get 10000010 for our exponent.

We then finally assemble the floating point binary number with the parts we have converted:

$$0\ 10000010\ 011000000000000000000000 = 01000001001100000000000000000000$$

And we convert to hexadecimal by splitting the number in 4 bit groups and converting each group to its equivalent hexadecimal digit:

$$0100\ 0001\ 0011\ 0000\ 0000\ 0000\ 0000\ 0000 = 4\ 1\ 3\ 0\ 0\ 0\ 0\ 0 = 41300000.$$

From now on the passages are going to include only the calculations to avoid verbosity. Nonetheless, the same reasoning will be applied for each conversion.

b) **5/64**;

Sign bit = 0

$$\frac{5}{64} = 0.078125$$

Conversion to binary:

$$- 0.078125 \times 2 = \mathbf{0} + 0.15625;$$

$$- 0.15625 \times 2 = \mathbf{0} + 0.3125;$$

$$- 0.3125 \times 2 = \mathbf{0} + 0.625;$$

$$- 0.625 \times 2 = \mathbf{1} + 0.25;$$

$$- 0.25 \times 2 = \mathbf{0} + 0.5;$$

$$- 0.5 \times 2 = \mathbf{1} + 0;$$

Result of conversion: 0.000101.

Normalization to scientific notation: $1.\mathbf{01} \times 2^{-4}$.

Exponent bits in decimal = $127 - 4 = 123$

Conversion of exponent:

Fraction	Rest
$123/2 = 61$	1
$61/2 = 30$	1
$30/2 = 15$	0
$15/2 = 7$	1
$7/2 = 3$	1
$3/2 = 1$	1
$1/2 = 0$	1

Exponent bits in binary = 01111011

Final binary number = 0 01111011 010000000000000000000000 =
00111101101000000000000000000000

Conversion to hexadecimal:

0011 1101 1010 0000 0000 0000 0000 0000 = 3 D A 0 0 0 0 0 =
3DA00000.

c) **-5/64**;

With this being the negative counterpart of the previous number we just need to flip the sign bit of $\frac{5}{64}$.

With that we get: 10111101101000000000000000000000.

Which, converted to hexadecimal, gives:

1011 1101 1010 0000 0000 0000 0000 0000 = B D A 0 0 0 0 0 =
BDA00000.

d) **6.125**;

Sign bit = 0

Conversion to binary:

Integer part = $6_{10} = 110_2$

Fractional part = 0.125_{10} :

$$- 0.125 \times 2 = \mathbf{0} + 0.25;$$

$$- 0.25 \times 2 = \mathbf{0} + 0.5;$$

$$- 0.5 \times 2 = \mathbf{1} + 0;$$

$$= 001_2.$$

$$\text{Normalized binary} = 110.001 = \mathbf{1.10001} \times 2^2$$

$$\text{Exponent bits in decimal} = 127 + 2 = 129.$$

Fraction	Rest
129/2 = 64	1
64/2 = 32	0
32/2 = 16	0
16/2 = 8	0
8/2 = 4	0
4/2 = 2	0
2/2 = 1	0
1/2 = 0	1

$$\text{Exponent bits in binary} = 10000001$$

$$\text{Final binary number} = 0\ 10000001\ 100010000000000000000000 = 01000000110001000000000000000000.$$

$$\text{Conversion to hexadecimal: } 0100\ 0000\ 1100\ 0100\ 0000\ 0000\ 0000\ 0000 = 4\ 0\ C\ 4\ 0\ 0\ 0\ 0 = 40C40000.$$

1.2 Hexadecimal to Decimal

All the numbers are going to be first converted to binary and then to decimal for an easier calculation

a) $\mathbf{42E48000}_{16}$

Conversion to binary:

$$\begin{aligned} 42E48000 &= 0100\ 0010\ 1110\ 0100\ 1000\ 0000\ 0000\ 0000 = \\ &0\ 10000101\ 110010010000000000000000 = \\ &01000010111001001000000000000000 \end{aligned}$$

From this we can observe that the sign bit is 0 (i.e. positive).

Conversion of exponent from binary to decimal:

$$10000101 = 2^7 + 2^2 + 2^0 = 128 + 4 + 1 = 133$$

To find how many bits have been moved beyond the floating point we now find the exponent for our notation:

$$133 - 127 = 6$$

And we then take the mantissa to multiply it by 2^6 (where 6 is the number we found through the last passage) to get the final binary number to convert to decimal:

$$1.110010010000000000000000 \times 2^6 = 1.11001001 \times 2^6 = 1110010.01$$

Conversion to decimal:

$$1110010.01 = 2^6 + 2^5 + 2^4 + 2 + 2^{-2} = 64 + 32 + 16 + 2 + 0.25 = 114.25$$

b) **3F880000**₁₆

Conversion to binary:

$$3F880000 = 0011\ 1111\ 1000\ 1000\ 0000\ 0000\ 0000\ 0000 =$$

$$0\ 01111111\ 000100000000000000000000 =$$

$$00111111100010000000000000000000$$

Sign bit = 0

Conversion of exponent from binary to decimal:

$$01111111 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2 + 1 = 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$$

$$\text{Power of 2} = 127 - 127 = 0$$

Final binary number:

$$1.0001 \times 2^0 = 1.0001$$

Conversion to decimal:

$$1.0001 = 2^0 + 2^{-4} = 1 + 0.0625 = 1.0625$$

c) **00800000**₁₆

Conversion to binary:

$$00800000 = 0000\ 0000\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000 =$$

$$0\ 00000001\ 000\ 00000000000000000000 =$$

$$00000000100000000000000000000000$$

Sign bit = 0

Conversion of exponent from binary to decimal:

$$00000001 = 2^0 = 1$$

$$\text{Power of 2} = 1 - 127 = -126$$

Final binary number:

$$1.000000000000000000000000 \times 2^{-126}$$

Conversion to decimal:

$$1.000000000000000000000000 \times 2^{-126} = 2^{-126} = 1.1754944 \times 10^{-38}$$

d) **C7F00000**₁₆

Conversion to binary:

$$\text{C7F00000} = 1100\ 0111\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000 =$$

$$1\ 10001111\ 111000000000000000000000 =$$

$$11000111111100000000000000000000$$

Sign bit = 1

Conversion of exponent from binary to decimal:

$$10001111 = 2^7 + 2^3 + 2^2 + 2^1 + 2^0 = 128 + 8 + 4 + 2 + 1 = 143$$

$$\text{Power of 2} = 143 - 127 = 16$$

Final binary number:

$$1.111000000000000000000000 \times 2^{16} = 1.111 \times 2^{16} =$$

$$1111000000000000.0$$

Conversion to decimal:

$$1111000000000000.0 = 2^{16} + 2^{15} + 2^{14} + 2^{13} = 65536 + 32768 + 16384 + 8192 = 122880 \text{ and, since the sign bit is 1 (i.e. negative), } = -122880.$$

2 Bonus Question 2

Loss of significance can occur via subtraction of values that are close to each other so that most of the significant digits get cancelled.

For example let $x = 1.5 \times 2^0$ and $y = 1.0 \times 2^{26}$ such that the sequence of operations $r = (x + y) - y$ results in $r = 0.0$ which is incorrect as the actual value should be 1.5.

This is due to catastrophic cancellation which occurred during the subtraction. This kind of occurrences can't be identified unless data is further examined leaving no way of checking that 0.0 is the wrong result.