Number Systems

A) The Decimal System

- Used in everyday life to represent numbers
- Based on 10 digits \( \{0, 1, 2, \ldots, 9\} \)
- Consider, for example, the number 9734 (Nine thousands, 7 hundreds, thirty four):

\[
9734 = (9 \times 1000) + (7 \times 100) + (3 \times 10) + 4
= 9 \times 10^3 + 7 \times 10^2 + 3 \times 10 + 4 \times 10^0
\]

- The base (radix) is 10 → each digit in the number is multiplied by 10 raised to a power corresponding to that digit's position.

- Same thing holds for decimal fractions but we powers of 10 are used.

\[
0.485 = 4 \times 10^{-1} + 8 \times 10^{-2} + 5 \times 10^{-3}
\]

- In general:

\[
X = \ldots d_2 d_1 d_0 \cdot d_{-1} d_{-2} d_{-3} = \left( \sum_{i} d_i \times 10^i \right)_{10} \text{ notation to indicate decimal}
\]
B) Other Number Systems (\(r\)

<table>
<thead>
<tr>
<th>System</th>
<th>Radix</th>
<th>Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>10</td>
<td>0, 1, 2, ..., 9</td>
</tr>
<tr>
<td>Binary</td>
<td>2</td>
<td>0, 1</td>
</tr>
<tr>
<td>Octal</td>
<td>8</td>
<td>0, 1, 2, ..., 7</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>16</td>
<td>0, 1, 2, ..., 9, A, B, C, D, E, F</td>
</tr>
</tbody>
</table>

Ex. 1
Convert \((735)_8\) to decimal.

\[
(735)_8 = 7 \times 8^2 + 3 \times 8 + 5 \times 8 = (477)_{10}
\]

Ex. 2
Convert \((A3.F2)_{16}\) to decimal.

\[
(A3.F2)_{16} = 10 \times 16 + 3 \times 16 + 15 \times 16^{-1} + 2 \times 16^{-2}
= (163.9453125)_{10}
\]

Ex. 3
Convert \((10110.011)_2\) to decimal.

\[
(10110.011)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}
= (22.375)_{10}
\]
c) Converting from Decimal to Binary:

\[(\ldots)_10 \rightarrow (\ldots)_2\]

- The binary system is used to represent numbers in the computer since it uses only 2 digits (0, 1 called bits) and therefore simpler to represent using 2 voltage levels (High, Low). Besides, it requires simpler circuits to manipulate binary numbers.

- To convert a number from decimal to binary, the integer part is handled in a different way than the fraction part. For example, converting \((30)_{10}\):

\[30 = 2^4 + 2^3 + 2^1\]

Repetitive division by 2 and recording the remainders in order of appearance from the decimal point and moving left:

\[
\begin{array}{c|c c c c c}
30 & 2 & 3 & 1 & 0 & 10 \\
\hline
5 & 1 & 0 & 1 & 0 & 10 \\
25 & 1 & 0 & 1 & 0 & 10 \\
12 & 1 & 0 & 1 & 0 & 10 \\
6 & 1 & 0 & 1 & 0 & 10 \\
3 & 1 & 0 & 1 & 0 & 10 \\
1 & 1 & 0 & 1 & 0 & 10 \\
0 & 1 & 0 & 1 & 0 & 10 \\
\end{array}
\]

\[X = (\ldots d_4 d_3 d_2 d_1 d_0 \cdot d_{-1} d_{-2} d_{-3} \ldots)_{10}
\]
Ex. 4 Convert \((23.375)_{10}\) to binary.

Solution:

\[
\begin{array}{c|cccc}
2 & 23 & 11 & R_1 \\
2 & 5 & R_1 \\
2 & 2 & R_1 \\
2 & 1 & R_0 \\
& 0 & R_1 \\
\end{array}
\]

\[0.75 = 2 \times 0.375\]
\[1.5 = 2 \times 0.75\]
\[1.0 = 2 \times 0.5\]
\[0.0 = 2 \times 0.0\]

Remainders: 10111 = 0110...

Thus, \((23.375)_{10} = (10111.011)_2\)

Ex. 5 Convert \((9.2)_{10}\) to binary.

Solution:

\[
\begin{array}{c|cccc}
2 & 9 & & & \\
2 & 4 & R_1 \\
2 & 2 & R_0 \\
2 & 1 & R_0 \\
& 0 & R_1 \\
\end{array}
\]

\[0.4 = 2 \times 0.2\]
\[0.8 = 2 \times 0.4\]
\[1.6 = 2 \times 0.8\]
\[1.2 = 2 \times 0.6\]
\[0.4 = 2 \times 0.2\]

It repeats afterwards:

\[(9.2)_{10} = (1001.0011)_2\]
Ex. 6  (Used to be a job interview question at a major software company).

Consider the following program segment:

```plaintext
num = 0.0;
for i = 1 to 10 do
  num = num + 0.1;
if (num = 1)
  print("OK");
else
  print("Not OK");
```

What should the program print?

Solu: The program prints "Not OK". The reason is that the fraction part \(0.1\) has no exact binary representation (see previous example) and is subject to truncation error due to finite precision. Adding \(0.1\) ten times will therefore result in a value < \(1.0\).
D) Operations in Binary Number System

\[
\begin{align*}
0 + 0 & = 0 \\
+ 1 & = 1 \\
+ 0 & = 1 \\
1 + 1 \text{ (carry)} & = 0 \\
0 + 0 & = 0 \\
1 & = 1 \\
0 & = 0 \\
1 \times 0 & = 0 \\
1 \times 0 & = 0 \\
1 \times 1 & = 1 \\
1 \times 1 & = 1 \\
0 \times 1 & = 0 \\
1 \times 0 & = 0 \\
1 & = 1
\end{align*}
\]

Ex. 7
Carry out the following operations in binary system:

\[
\begin{align*}
11 & + 110.1 \\
\hline
10001.11
\end{align*}
\]

\[
\begin{align*}
16 \times 21 & = 11 \\
16 \times 11 & = 11
\end{align*}
\]
ii) \[ 100101 - 1011 \]

\[
\begin{array}{cccc}
0 & 1 & 0 & 0 \\
+ & 0 & 0 & 0 \\
\hline
1 & 0 & 1 & 1
\end{array}
\]

\[ 1011 - 11010 \]

iii) \[ 1101 \times 101 \]

\[
\begin{array}{cccc}
1 & 1 & 1 & 0 & 1 \\
\times & 1 & 0 & 1 & 1 \\
\hline
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
+ & 1 & 1 & 1 & 0 & 1 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}
\]
iv) \[ 10010001 \div 101 \]

\[ \begin{array}{c}
101 \\
\hline
11101 \\
101 \\
\hline
101 \\
- 101 \\
\hline
101 \\
- 101 \\
\hline
0110 \\
- 101 \\
\hline
0111
\end{array} \]

Thus, \[ 10010001 \div 101 = 11101 \]