3. More numbers

- Review
  - Number representations
  - Addition, Subtraction
  - Sign extension
  - Overflow

- Logical Operations
  - AND, OR, NOT

- Hexadecimal representation
  - Fractions, large numbers
  - Floating point representation
  - Examples

- Representing characters
  - ASCII
REVIEW

2's COMPLEMENT

\[ 0001 = \frac{1}{16} \]

To get \(-1\), complement bits, add 1

\[ 1111 = -1 \]

SHORTCUT:

1. Copy bits from R to L till the first "1"
2. Then, flip remaining bits to the left

EXAMPLE

\[ 011010000 = X \]

\[ 100110000 = -X \]

1st "1"
SIGN EXTENSION

\[
\begin{array}{ll}
4 \text{ BITS} & 8 \text{ BITS} \\
+4_{10} & 0100 \\
-4_{10} & 1100 \\
\end{array}
\]

OVERFLOW

\[
\begin{array}{ll}
5 \text{ BITS} & 8 \text{ BITS} \\
2\text{'s complement} & 9_{10} = 01000 \\
& 9_{10} = 01001 \\
+17_{10} & 10001 \\
\hline
-8_{10} & 11000 \\
-9_{10} & \Phi0111 \\
\hline
01111 & = +15_{10}
\end{array}
\]

RULES: SIGN OF RESULT IS DIFFERENT

(OR)

CARRY INTO MSB ≠ CARRY OUT OF MSB
### Logical Operations

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A AND B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A XOR B (SUM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### NOT A

<table>
<thead>
<tr>
<th>A</th>
<th>NOT A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Examples of Logical Operations

AND  USEFUL AS "MASKS"
  • useful for clearing bits
    ➢ AND with zero = 0
    ➢ AND with one = no change

  AND  11000101
       00001111
    AND  00000101

OR
  • useful for setting bits
    ➢ OR with zero = no change
    ➢ OR with one = 1

  OR  11000101
      00001111
    OR  11001111

NOT
  • unary operation -- one argument
  • flips every bit

  NOT  11000101
       00111010
FRACTIONS \\
(FIXED POINT)

\[ 0.5_{10} = \frac{1}{2} = 2^{-1} \]

\[ = 0.1 \]

\[ = 0.1 + \frac{1}{4} = 1.25_{10} \]

\[ \rightarrow 01.01 = \frac{1}{2} + \frac{1}{4} = 1.25_{10} \]

\[ 2.75_{10} = 10.11_2 \]

\[ 0.3_{10} = 0.01011_2 \]

\[ \begin{array}{c}
0.3 \times 2 \\
0.6 \\
0.12 \\
0.4 \\
0.8 \\
1.6 \\
1.0
\end{array} \]

\[ = 0.0101101 \]

\[ = 0.010011001 \]
Very Large and Very Small: Floating-Point

**Scientific Notation**

For example: $6.023 \times 10^2 \Rightarrow$ Needs 7.9 bits

$6.626 \times 10^{-34} \Rightarrow$ Needs 110 bits

$F \times 2^E \Rightarrow$ Exponent

**Fraction**
IEEE 754 Floating Point Standard

Single Precision

\[ N = (-1)^S \times \begin{cases} 0. \text{fraction} \times 2^{\text{exponent} - 127}, & 1 \leq \text{exponent} \leq 254 \\ \text{EXCEPTION} & \end{cases} \]
Floating Point Example

Single-precision IEEE floating point number:
10111111010000000000000000000000

\[
\begin{align*}
\text{sign} & : \text{NEGATIVE} \\
\text{exponent} & : 126_{10} \\
\text{fraction} & : 0.1_{2} \Rightarrow 0.5_{10}
\end{align*}
\]

\[
\text{VALUE} = -1.5 \times 2^{(126-127)} = -1.5 \times 2^{-1} = -0.75
\]
IEEE-754 Floating-Point Conversion
From Decimal Floating-Point
To 32-bit and 64-bit Hexadecimal Representations
Along with Their Binary Equivalents

Enter a decimal floating-point number here,
then click either the Rounded or the Not Rounded button.

Decimal Floating-Point: 2.325

Rounding from floating-point to 32-bit representation uses the IEEE-754 round-to-nearest-value mode.

Results:

Decimal Value Entered: 2.325

Single precision (32 bits):

<table>
<thead>
<tr>
<th>Bit 31</th>
<th>Bit 30 - 23</th>
<th>Bit 22 - 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Bit</td>
<td>Exponent Field</td>
<td>Significand</td>
</tr>
<tr>
<td>0: +</td>
<td>Decimal value of exponent field and exponent</td>
<td>Decimal value of the significand</td>
</tr>
<tr>
<td>1: -</td>
<td>128 - 127 = 1</td>
<td>1.625000</td>
</tr>
</tbody>
</table>

Hexadecimal: 4014CCCD  Decimal: 2.3250000

Double precision (64 bits):

<table>
<thead>
<tr>
<th>Bit 63</th>
<th>Bit 62 - 52</th>
<th>Bit 51 - 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Bit</td>
<td>Exponent Field</td>
<td>Significand</td>
</tr>
<tr>
<td>0: +</td>
<td>Decimal value of exponent field and exponent</td>
<td>Decimal value of the significand</td>
</tr>
<tr>
<td>1: -</td>
<td>1024 - 1023 = 1</td>
<td>1.16250000000000</td>
</tr>
</tbody>
</table>

Hexadecimal: 4002999999999999A  Decimal: 2.32500000000000000000

[ Convert IEEE-754 32-bit Hexadecimal Representations to Decimal Floating-Point Numbers. ]
[ Convert IEEE-754 64-bit Hexadecimal Representations to Decimal Floating-Point Numbers. ]