Binary and Hexadecimal

CSC 225

Christopher Siu

Number Bases

Definition

The **base** of a positional numeral system is its number of unique symbols used to represent numbers.

- The base is also called the **radix**.

Example

Babylonian mathematicians used “sexagesimal”, base 60.

Example

Mayan mathematicians used “vigesimal”, base 20.

Decimal

Definition

The **decimal** system is a positional notation using base 10.

- Uses 10 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
- Decimal is likely a result of most humans' having 10 fingers.

Example

In decimal, each **digit** represents a quantity of a power of 10:

$$3027 = 3 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

Thus, $3027 \neq 3207$.

Binary

Definition

The **binary** system is a positional notation using base 2.

- Uses 2 symbols: `0` and `1`.
- Binary is straightforward to implement in electronics.

Example

In binary, each **bit** represents a quantity of a power of 2:

$$\texttt{10110} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

Thus, $\texttt{10110} = 22$, $\texttt{11001} = 25$, and $\texttt{10110} \neq \texttt{11001}$.

Binary Addition

Binary numbers may be added in the same way as decimal numbers:

1. Write the numbers one above the other, right-aligned.
2. Add the bits in each column.
   - If a column's sum reaches 2, write a 0 and carry a 1.
   - If a column's sum reaches 3, write a 1 and carry a 1.
3. Repeat from right to left.

Example

To add 100110 and 111:

$$\begin{array}{r} 100110 \\ +\quad 111 \\ \hline 101101 \end{array} \quad \textit{equivalent to} \quad \begin{array}{r} 38 \\ +\quad 7 \\ \hline 45 \end{array}$$

Two's Complement Notation

Definition

A **byte** is composed of 8 bits.

- Bits are typically numbered right-to-left from 0.
  - The left-most bit is called the **most significant bit**, or "MSB".
  - The right-most bit is called the **least significant bit**, or "LSB".
- Computers have a finite number of bits for each value.

Example

Consider:

7	6	5	4	3	2	1	0
0	0	1	0	1	1	0	1

...this 8-bit value's most significant bit is bit 7, a 0.

Two's Complement Notation

Definition

In **two's complement** notation, n-bit integers with a MSB of '1' are negative and count up from -2^{n-1}.

- In other words, the "upper half" of the representable number line is placed "before" 0.
- An integer representation allowing for negative numbers is called **signed**. One that does not, **unsigned**.

Example

The possible values of a signed 3-bit integer are:

Binary	100	101	110	111	000	001	010	011
Decimal	-4	-3	-2	-1	0	1	2	3

Two's Complement Notation

There are 2^n unique n-bit binary values, thus:

- An unsigned n-bit integer has a minimum value of 0.
- An unsigned n-bit integer has a maximum value of $2^n - 1$.

Example

The possible values of an unsigned 8-bit integer are $\{0, \ldots, 255\}$.

- A signed n-bit integer has a minimum value of -2^{n-1}.
- A signed n-bit integer has a maximum value of $2^{n-1} - 1$.

Example

The possible values of a signed 8-bit integer are $\{-128, \ldots, 127\}$.

Carries

Definition

A **carry** occurs when an operation requires more bits than are available.

Example

Consider the following sum of (signed) 8-bit integers:

$$\begin{array}{rr} & \texttt{1001 1101} \\ + & \texttt{1000 0110} \\ \hline & \texttt{0010 0011} \end{array} \quad \textit{equivalent to} \quad \begin{array}{rr} & -99 \\ + & -122 \\ \hline & 35 \end{array}$$

There is no space to store the left-most '1'.

Overflows

Definition

An **overflow** occurs when an operation results in a value too large or too small to be represented.

Example

Consider the following sum of (signed) 8-bit integers:

$$\begin{array}{rr} & \texttt{0101 1110} \\ + & \texttt{0011 0001} \\ \hline & \texttt{1000 1111} \end{array} \quad \textit{equivalent to} \quad \begin{array}{rr} & 94 \\ + & 49 \\ \hline & -113 \end{array}$$

The left-most '1' has overflowed into the sign bit.

Binary Subtraction

Example

To subtract 5 from 6 using signed 4-bit integers:

$$\begin{array}{rr} & \texttt{0110} \\ + & \texttt{1011} \\ \hline & \texttt{0001} \end{array} \quad \textit{equivalent to} \quad \begin{array}{rr} & 6 \\ + & -5 \\ \hline & 1 \end{array}$$

Example

To subtract 6 from -1 using signed 4-bit integers:

$$\begin{array}{rr} & \texttt{1111} \\ + & \texttt{1010} \\ \hline & \texttt{1001} \end{array} \quad \textit{equivalent to} \quad \begin{array}{rr} & -1 \\ + & -6 \\ \hline & -7 \end{array}$$

Hexadecimal

Definition

The **hexadecimal** system is a positional notation using base 16.

- Uses 16 symbols: `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `A`, `B`, `C`, `D`, `E`, `F`.
- By convention, preceded by "`0x`" or "`x`".

Example

In hexadecimal, each digit represents a quantity of a power of 16:

$$\texttt{0x950B} = 9 \cdot 16^3 + 5 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0$$

Thus, $\texttt{0x950B} = 38155$.

Hexadecimal

- Because $2 = 2^1$ and $16 = 2^4$, binary numbers can be converted to hexadecimal in groups of 4 bits.

Example

Given the binary number 1001010100001011:

Binary	1001	0101	0000	1011
Hexadecimal	9	5	0	B

Thus, 1001010100001011 = 0x950B:

- $2^{15} + 2^{12} + 2^{10} + 2^8 + 2^3 + 2^1 + 2^0 = 38155$
- $9 \cdot 16^3 + 5 \cdot 16^2 + 11 \cdot 16^0 = 38155$