Numbers

COMP370 Introduction to Computer Architecture

Goals

- Understand binary and hexadecimal numbers
- Be able to convert between number bases
- Understand binary fractions

Unary Numbers

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Unary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>111</td>
</tr>
<tr>
<td>4</td>
<td>1111</td>
</tr>
<tr>
<td>5</td>
<td>11111</td>
</tr>
<tr>
<td>6</td>
<td>111111</td>
</tr>
</tbody>
</table>

Decimal Numbers

- Decimal numbers work well for humans.
- Each position is 10 times the one to the right

3,148 is

1000’s  100’s  10’s  1’s

3  1  4  8
Binary Numbers

- Computers usually use binary numbers, base 2
- Binary numbers only have two digits, 0 and 1
- Each position is 2 times the one to the right

\[010011_2 (19_{10})\] is

\[
\begin{array}{cccccc}
32's & 16's & 8's & 4's & 2's & 1's \\
0 & 1 & 0 & 0 & 1 & 1
\end{array}
\]

Hexadecimal Numbers

- It can be convenient to represent values in base 16 or hexadecimal.
- Hex numbers have the digits

\[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ A \ B \ C \ D \ E \ F\]
- Each position is 16 times the one to the right

\[31E8_{16} (12,776_{10})\] is

\[
\begin{array}{cccc}
4096's & 256's & 16's & 1's \\
3 & 1 & E & 8
\end{array}
\]

Indicating the Number Base

- The number or radix base can be specified by a subscript after the number.

\[47_{10} \ 101111_2 \ 2F_{16}\]
- Computer languages use a prefix

\[47 \ 0B101111 \ 0X2F\]
- Binary starts with zero “B” \(\text{\textit{(C++ only)}}\)
- Hexadecimal starts with zero “X”

Converting Decimal to Binary

- Start with the largest even power of 2 bigger than the decimal number

\[
\begin{array}{cccccccc}
4096 & 2048 & 1024 & 512 & 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1
\end{array}
\]
- For each power of two

\[
\begin{array}{l}
\text{if the number} \geq \text{power of 2} \\
\text{put a 1 bit to the right;}
\end{array}
\]

\[
\begin{array}{l}
\text{Subtract power of 2 from the number;}
\end{array}
\]

\[
\begin{array}{l}
\text{else} \\
\text{put a 0 to the right;}
\end{array}
\]
- Repeat for all powers of 2
### Decimal to Binary Example

- Convert 317 to binary

<table>
<thead>
<tr>
<th>Number</th>
<th>4096</th>
<th>2048</th>
<th>1024</th>
<th>512</th>
<th>256</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- The answer is 0000100111101

### Decimal-to-Binary Conversion

Convert the following decimal numbers to binary:

- $12 = 8 + 4 = 2^3 + 2^2$ → 1100
- $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0$ → 11001
- $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^0$ → 111010
- $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^0$ → 1010010

### Repeated Division Conversion

- To convert a decimal number to any base, $R$
- Generated digits are right to left, least to most

```java
number = decimal number;
while (number > 0) {
    rem = number % R;
    Next digit is rem;
    number = number / R; // integer division
}
```

### Decimal-to-Binary Conversion - Repeated Division-by-2 Method

Convert 12 to binary:

- $12 \div 2 = 6$, remainder 0
- $6 \div 2 = 3$, remainder 0
- $3 \div 2 = 1$, remainder 1
- $2 \div 2 = 1$, remainder 1
- $1 \div 2 = 0$, remainder 1

Stop when the whole-number quotient is 0.

MSB → LSB

1 1 0 0
**Decimal-to-Binary Example**

Convert 19 to binary

\[
\begin{align*}
19 & = 9 & \text{Remainder} \\
9 & = 4 \\
4 & = 2 \\
2 & = 1 \\
1 & = 0
\end{align*}
\]

\[10011\]

**Converting Decimal to Hex**

- Start with the largest even power of 16 bigger than the decimal number
- Repeat for all powers of 16, big to small

```java
if the number >= power of 16 {
    quotient = number / power of 16;
    append quotient to result;
    number = number % power of 16;
}
Add the remainder to the result;
```

**Decimal to Hex Example**

- Convert 317 to Hex

\[
\begin{align*}
1048576 & = 0 & \text{Remainder} \\
65536 & = 0 \\
4096 & = 0 \\
256 & = 1 \quad 317/256 = 1, \text{ rem}=61 \\
16 & = 3 \quad 61/16=3, \text{ rem}=13 \\
1 & = D
\end{align*}
\]

- The answer is \(13D\)

**Decimal-to-Hex Conversion**

Repeated Division-by-16 Method:

- Convert 299 to hex.

\[
\begin{align*}
299 & = 18 & \text{Remainder} \\
18 & = 1 \\
1 & = 0
\end{align*}
\]

Stop when the whole-number quotient is 0.
Binary-to-Decimal Conversion

• For each 1 bit, add the value of that bit.

Convert the binary whole number 1101101 to decimal

Weight: \(2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0\)

Binary number: \(1\ 1\ 0\ 1\ 1\ 0\ 1\)

\[1101101 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0\]
\[= 64 + 32 + 8 + 4 + 1\]
\[= 109\]

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
</tr>
</tbody>
</table>

Online Quiz

• Complete the online quiz on base number conversion on the assignments page of the website, http://williams.comp.ncat.edu/comp370/

• Due by 5:00pm on Friday, January 16, 2009

• Your score is the percentage right minus the number of seconds over 90.

• You may take the exam as often as you like. Your grade will be determined by your best score.

\[This\ is\ not\ easy.\]
Converting Hex to Binary

- Each hexadecimal digit converts to four bits.
- Convert 2A7C to binary

```
2   A   7   C
0010 1010 0111 1100
```

Converting Binary to Hex

- Group the binary bits into groups of four.
- Each group of four bits converts to a single hexadecimal digit.
- Convert 1101000101101 to hexadecimal

```
1 1010 0010 1101
1    A     2     D
```

Decimal Fractions

- To the right of the decimal or radix point, each digit is \( \frac{1}{10} \)th of the digit to the left.

```
10's 1's . 1/10's 1/100's 1/1000's
1   2   .   3   7   5
```

Binary Fractions

- To the right of the radix point, each digit is \( \frac{1}{2} \) of the digit to the left.
- The decimal number 12.375 is

```
8's 4's 2's 1's . 1/2's 1/4's 1/8's
1   1   0   0   .   0   1   1
```
Example Binary Fractions

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>0.25</td>
<td>0.01</td>
</tr>
<tr>
<td>0.125</td>
<td>0.001</td>
</tr>
<tr>
<td>0.75</td>
<td>0.11</td>
</tr>
<tr>
<td>0.1</td>
<td>0.00011001100110011…</td>
</tr>
</tbody>
</table>

Binary-to-Decimal Conversion

Convert the Fractional binary number 0.1011 to decimal

\[
\begin{align*}
\text{Weight:} & \quad 2^{-1} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4} \\
\text{Binary number:} & \quad 0.1 \quad 0 \quad 1 \quad 1 \\
0.1011 &= 2^{-1} + 2^{-3} + 2^{-4} \\
&= 0.5 + 0.125 + 0.0625 \\
&= 0.6875
\end{align*}
\]

Decimal to Binary Conversion

```plaintext
number = decimal fraction;
while (number > 0) {
    number = number * 2;
    if (number >= 1) {
        append 1 to result;
        result = result - 1;
    } else {
        append 0 to result;
    }
}
```

Decimal to Binary Example

Convert 0.3125 to binary

\[
\begin{align*}
0.3125 \times 2 &= 0.625 & \text{result} = 0.0 \\
0.625 \times 2 &= 1.25 & \text{result} = 0.01 \\
0.25 \times 2 &= 0.50 & \text{result} = 0.010 \\
0.5 \times 2 &= 1.00 & \text{result} = 0.0101
\end{align*}
\]