Number Systems

Introduction / Number Systems
Data Representation

- Data representation can be *Digital* or *Analog*
- In Analog representation values are represented over a *continuous* range
- In Digital representation values are represented over a *discrete* range
- Digital representation can be
  - Decimal
  - Binary
  - Octal
  - Hexadecimal

We need to know how to use and convert from one to another!
Using Binary Representation

- Digital systems are binary-based
  - All symbols are represented in binary format
  - Each symbol is represented using Bits
  - A bit can be one or zero (on or off state)

- Comparing Binary and Decimal systems:
  - In Decimal a digit is [0-9] – base-10
  - In Binary a digit is [0-1] – base-2
  - In Decimal two digits can represent [0-99] \( \rightarrow 10^2-1 \)
  - In Binary two digits can represent [0-3] \( \rightarrow 2^2-1 \)
Binary Counting

<table>
<thead>
<tr>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
<th></th>
<th>2^{-1}</th>
<th>2^{-2}</th>
<th>2^{-3}</th>
<th>2^{-4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.125</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.375</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.625</td>
</tr>
</tbody>
</table>

Binary Representation: 1 0 0 1

Decimal Representation: 9.625
Counting in Different Numbering Systems

- **Decimal**
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ..., 19, 20, 21, ..., 29, 30, ..., 39, ...

- **Binary**
  - 0, 1, 10, 11, 100, 101, 110, 111, 1000, ...

- **Octal**
  - 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, ..., 17, 20, 21, 22, 23, ..., 27, 30, ...

- **Hexadecimal**
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, ..., 1F, 20, ..., 2F, 30, ...

Remember: **aa.bb**
- **aa** is the whole number portion
- **bb** is the fractional portion
- "." is the radix point

**Demonstrating different number base or radix**
Learning Number Conversion
Binary-to-Decimal Conversions

11011 \rightarrow \text{Decimal}

\begin{align*}
1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 &= 16 + 8 + 2 + 1 = 27_{10} \\
(n_{N-1}n_{N-2}...n_3n_2n_1n_0)_b & \overset{\text{Convert}}{\rightarrow} n_0 \times b^0 + n_1 \times b^1 + n_2 \times b^2 + n_3 \times b^3 ... + n_{N-1} \times b^{N-1}
\end{align*}

In the above example:
Binary is base-2 (b=2)

- n_0 = 1
- n_1 = 1
- n_2 = 0
- n_3 = 1
- n_4 = 1

Q: What is 11011.11 in Decimal?

Ans: $= 27 + (1 \times 2^{-1} + 1 \times 2^{-2})$
    $= 27 + 0.5 + 0.25$
    $= 27.75$
Decimal-to-Binary Conversions

**Quotient + Remainder**

65 / 2 = 32 + Remainder_of_1
32 / 2 = 16 + Remainder_of_0
16 / 2 = 8 + Remainder_of_0
8 / 2 = 4 + Remainder_of_0
4 / 2 = 2 + Remainder_of_0
2 / 2 = 1 + Remainder_of_0
1 / 2 = 0 + Remainder_of_1

65 → Binary

1000001

MSB  LSb

1- Save the remainder
2- Continue until Quotient = 0

What if you are using a calculator?

65/2 = 32.5
0.5 x 2 (base-2) = 1

LSB = Least Significant Bit
MSB = Most Significant Bit

Last one should be “0”
Decimal-to-Binary Conversions

0.125 $\rightarrow$ Binary

Radix point + The whole portion + The fractional portion

$0.125 \times 2 = 0 + \text{fractional portion of } 0.25$

$0.25 \times 2 = 0 + \text{fractional portion of } 0.5$

$0.5 \times 2 = 1 + \text{fractional portion of } 0.0$

$0.001$

Last one should be “1”
Octal/Decimal Conversions

Binary is base-8 (b=8)

\[ 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = 3 \times 64 + 7 \times 8 + 2 \times 1 = 250_{10} \]

\[ (n_{N-1}n_{N-2}...n_3n_2n_1n_0)_b \xrightarrow{\text{Convert}} n_0 \times b^0 + n_1 \times b^1 + n_2 \times b^2 + n_3 \times b^3 \ldots + n_{N-1} \times b^{N-1} \]

What about 372.28?

Ans: \( = 250 + (2 \times 8^{-1}) = 250.25 \)

Decimal-to-Octal:

266 \rightarrow Octal

\[ \frac{266}{8} = 33 + \text{Remainder of } 2 \]
\[ \frac{33}{8} = 4 + \text{Remainder of } 1 \]
\[ \frac{4}{8} = 0 + \text{Remainder of } 4 \]

MSB  LSB
Hex-to-Decimal Conversions

Binary is base-16 (b=16)

2AF \rightarrow \text{Decimal}

\[ 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = 512 + 160 + 16 = 687_{10} \]

\[ (n_{N-1} n_{N-2} \ldots n_3 n_2 n_1 n_0)_b \rightarrow n_0 \times b^0 + n_1 \times b^1 + n_2 \times b^2 + n_3 \times b^3 \ldots + n_{N-1} \times b^{N-1} \]

In the above example:
- Binary is base-16 (b=16)
- \( n_0 = F \) which is 15
- \( n_1 = A \) which is 10
- \( n_2 = 2 \)

Remember A=10, F=15
Decimal-to-Hex Conversions

423 → Hex

423/16 = 26 + Remainder \_of\_ 7
26/16 = 1 + Remainder \_of\_ 10
1/16 = 0 + Remainder \_of\_ 1

MSB 1
LSB A

1A7

Remember 10 in Hex is A

214 → Hex

214/16 = 13 + Remainder \_of\_ 6
13/16 = 0 + Remainder \_of\_ 13

MSB D
LSB 6

D6

Remember 13 in Hex is D

Last one should be “0”
Converting from Hex-to-Octal

$124_{Hex} \rightarrow ??_{Oct}$

Always convert to Binary first and then from binary to Oct.

Ans:  
= 124 Hex  
= 0001 0010 0100  
= 000 100 100 100  
= 0 4 4 4  
= 444 Oct
Counting

- **Decimal**
  - \(0,1,2,3,4,5,6,7,8,9,10,11,12\ldots,19,20,21,\ldots,29,30,\ldots,39\ldots\)

- **Binary**
  - \(0,1,10,11,100,101,110,111,1000,\ldots\)

- **Octal**
  - \(0,1,2,3,4,5,6,7,10,11,12\ldots,17,20,21,22,23\ldots,27,30,\ldots\)

- **Hexadecimal**
  - \(0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,\ldots,1F,20,\ldots,2F,30,\ldots\)
Converting to BCD and ASCII

- We use Hex and Octal numbers to simplify number representation
- Any symbol can be represented by a code
  - Example: *American Standard Code for Information Interchange* (ASCII)
    - Each symbol is represented by a seven-bit code (How many symbols can be represented? – 127)
    - Example: A=100 0001 = 41 in Hex, 1=011 0000 = 31 in Hex, $=010 0100 = 24 in Hex (What is “DAD” in ASCII?)

Look at the ASCII code listing – Don’t memorize!
Converting to BCD and ASCII

- We use Hex and Octal numbers to simplify number representation
- Any symbol can be represented by a code
  - Example: *Binary-Coded-decimal* (BCD)
    - Each digit has its own binary code
    - Example: $6_{10}=0110$, $16_{10}=0001 0110$ (In binary 16 is?)
  - BCD can be packed or unpacked
    - $12 \rightarrow$ Packed=$0001 0010$; unpacked=$0000 0001 0000 0010$
Terminologies

- **BYTE**
  - 8 bits is equivalent to one byte

- **NIBBLE**
  - 4 bits is equivalent to one nibble

- **WORD**
  - 16 bits is equivalent to one word

1. 128 bits is equivalent to how many bytes? \( \frac{128}{8} = 16 \)
2. What is the maximum number that can be represented by 1 byte? \( 2^8 - 1 = 255 \)
Switch State

In each case we have 16 switches.
1- What Binary/Decimal/Hexadecimal number does each switch represent?
2- What is the maximum binary number we can represent using these switches?

Maximum number: $2^{16}-1=65536-1=65535=64\text{K}$ in Computer terms!