**Representing Numbers**
CST111—Introduction to Information Technology

**Numbering Systems** (Page 1)
- Five elements that are common to every numbering system (binary, octal, decimal, hexadecimal, etc.):
  1. There are as many digits as *name* of base
     - I.e. base 10 digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  2. Each place value is a power of the base
     - I.e. base 10 places are: 1000’s = 10^3, 100’s = 10^2, 10’s = 10^1, 1’s = 10^0
     - \(10^3\ 10^2\ 10^1\ 10^0\)

**Numbering Systems** (Page 2)
- Five numbering system elements (con.):
  3. Every digit *holds a place* (304 or 3 4)
  4. That digit show the *quantity* of that place (in 304 there are 3 hundreds, 0 tens, 4 ones)
  5. The *sum of multiplying digit times place value* (called the total place value)
     represents the total value of the number (i.e. 304) ...
     - \(3 \times 100 + 0 \times 10 + 4 \times 1 = 304\)
     - \(300 + 0 + 4 = 304\)

**Binary Number System** (Page 1)
- Binary number system is base 2 so there are two digits (0 and 1)
- The place values all are powers of base 2
  - \(2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0\)
  - \(128\ 64\ 32\ 16\ 8\ 4\ 2\ 1\)

**Binary Number System** (Page 2)
- To convert a binary number to decimal:
  - Multiply each digit by its place value
  - Sum all the products
  - Example binary number 10011₂:
    - \(1 \times 16 = 16\)
    - \(0 \times 8 = 0\)
    - \(0 \times 4 = 0\)
    - \(1 \times 2 = 2\)
    - \(1 \times 1 = 1\)
    - The sum of 16 + 0 + 0 + 2 + 1 = 19 = 10011₂

**Binary Number System** (Page 3)
- Shortcut to convert binary to decimal:
  - Add the *place values* of only the one (1) digits
  - Example binary number 10011₂:
    - \(16\ 8\ 4\ 2\ 1\)
    - \(1\ 0\ 0\ 1\ 1\)
    - The sum of 16 + 2 + 1 = 19 = 10011₂

**Binary Number System** (Page 4)
- To convert a decimal number to binary:
  - Continually divide by base two (2) until the quotient is zero (0)
  - The *remainder* of the divisions in reverse order is the resulting binary number—reminders only can be zero (0) or one (1)
  - The *last division* is always \(1 \div 2\) equals zero (0) with a remainder of one (1)
To convert decimal to binary (con.):

- Example decimal number 27:

<table>
<thead>
<tr>
<th>Division</th>
<th>Quotient</th>
<th>Remainders</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 ÷ 2</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>13 ÷ 2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>6 ÷ 2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3 ÷ 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 ÷ 2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Remains in reverse order:
  - 11011_2

13 **Hexadecimal Number System (Page 1)**

- Hexadecimal system is base 16 so there are sixteen digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) ...
  - The characters A to F are the hexadecimal equivalent digits for number 10 to 15
- The place values all are powers of base 16
  - 16^3  16^2  16^1  16^0
  - 4096  256  16  1

14 **Hexadecimal Number System (Page 2)**

- An important use of hexadecimal (also called hex) is as a shorthand for binary
  - Each hex digit can represent four binary digits
- Therefore the following four-byte word (instruction) values are equivalent:
  - 0011 1010 1111 0010 1101 0111 1011 0001_2
  - 3AF2D7B1_{16}

15 **Hexadecimal Number System (Page 3)**

<table>
<thead>
<tr>
<th>Hex</th>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0_{16}</td>
<td>0</td>
<td>0000_2</td>
</tr>
<tr>
<td>1_{16}</td>
<td>1</td>
<td>0001_2</td>
</tr>
<tr>
<td>2_{16}</td>
<td>2</td>
<td>0010_2</td>
</tr>
<tr>
<td>3_{16}</td>
<td>3</td>
<td>0011_2</td>
</tr>
<tr>
<td>4_{16}</td>
<td>4</td>
<td>0100_2</td>
</tr>
<tr>
<td>5_{16}</td>
<td>5</td>
<td>0101_2</td>
</tr>
<tr>
<td>6_{16}</td>
<td>6</td>
<td>0110_2</td>
</tr>
<tr>
<td>7_{16}</td>
<td>7</td>
<td>0111_2</td>
</tr>
<tr>
<td>8_{16}</td>
<td>8</td>
<td>1000_2</td>
</tr>
<tr>
<td>9_{16}</td>
<td>9</td>
<td>1001_2</td>
</tr>
<tr>
<td>A_{16}</td>
<td>10</td>
<td>1010_2</td>
</tr>
<tr>
<td>B_{16}</td>
<td>11</td>
<td>1011_2</td>
</tr>
<tr>
<td>C_{16}</td>
<td>12</td>
<td>1100_2</td>
</tr>
<tr>
<td>D_{16}</td>
<td>13</td>
<td>1101_2</td>
</tr>
<tr>
<td>E_{16}</td>
<td>14</td>
<td>1110_2</td>
</tr>
<tr>
<td>F_{16}</td>
<td>15</td>
<td>1111_2</td>
</tr>
</tbody>
</table>

16 **Hexadecimal Number System (Page 4)**

- To convert a hex number to decimal:
  - Multiply each digit by its place value
  - Sum all the products
- Example hexadecimal number A2C_{16}:
  - A (10) × 256 = 2560
  - 2 × 16 = 32
  - C (12) × 1 = 12
  - The sum of 2560 + 32 + 12 = 2604 = A2C_{16}
Hexadecimal Number System (Page 5)
- To convert a decimal number to hex:
  - Continually divide by base sixteen (16) until the quotient is zero (0)
  - The remainder of the divisions in reverse order is the resulting hexadecimal number

Hexadecimal Number System (Page 6)
- To convert decimal to hex (cont.):
  - Example decimal number 312:
    | Division | Quotient | Remainders |
    |----------|----------|------------|
    | 312 ÷ 16 | 19       | 8          |
    | 19 ÷ 16  | 1        | 3          |
    | 1 ÷ 16   | 0        | 1          |
  - Remainders in reverse order: 138

Binary Addition
- First convert the decimal numbers to binary
- Add digits from right to left
- Carry digits to next position to the left

Hexadecimal Addition
- First convert the decimal numbers to hexadecimal
- Add digits from right to left
- Carry digits to next position to the left

Negative Number Representation
- Numbers occupy a fixed-length in memory:
  - I.e. 8-bit byte or 16-bit word
  - Sign (positive or negative) is stored in left-most bit position (effectively using one less bit for storing the number):
    - Zero (0) is positive
    - One (1) is negative

Sign-Magnitude Representation
- Simply change sign, keep all number digits the same:
  - $16 = 00010000_2$ (first bit is sign)
  - $-16 = 10010000_2$
- Two problems:
  - Arithmetic is not easily hardware compatible
  - Two different representations for zero (0):
    - $00000000_2$
    - $10000000_2$

One's Complement Representation (Page 1)
- Invert all digits (zeros become ones and ones become zeros)
  - $16 = 00010000_2$ (first bit is sign)
  - $-16 = 11101111_2$
- Solves arithmetic problem ...
  - Add as usual; if there is a carry digit to the left, it must be added back into right digit

One's Complement Representation (Page 2)
- There still are two different representations for zero (0):
  - $00000000_2$
  - $11111111_2$

Two's Complement Representation (Page 1)
- Add 1 to the 1’s complement of the number:
  - $16 = 00010000_2$ (first bit is sign)
• 1 110 1111₂ (invert all digits)
• -16 = 1 111 0000₂ (add 1 to previous)

- Solves problem of two values for zero (0):
  • 0 = 0 000 0000₂ (first bit is sign)
  • 1 111 1111₂ (invert all digits)
  • 0 = 0 000 0000₂ (add 1 to previous)

Two's Complement Representation (Page 2)
- There is one negative number for which there is no positive number:
  • -128 = 1 000 0000₂ (first bit is sign)
  • 0 111 1111₂ (invert all digits)
  • -128 = 1 000 0000₂ (add 1 to previous)

Two's Complement Representation (Page 3)
- A shortcut to convert to 2's complement:
  • Starting from right-most digits, keep all zero (0) digits unchanged until first 1
  • Also leave the first one (1) digit from the right unchanged
  • Invert remaining left-most digits
  • 16 = 0 001 0000₂ (first bit is sign)
  • -16 = 1 111 0000₂

Subtraction with Two's Complement
- This is the way that IBM PC and most other computers perform subtraction
  • Convert number to be subtracted (second number) to 2's complement and then add the two numbers

Two’s Complement Parts 1 and 2—An Introduction
- http://www.youtube.com/watch?v=9W67T2zzAfo&NR=1
- http://www.youtube.com/watch?v=Hof95YLQk0&feature=related