COMPSCI 210 S1T Computer Systems

Data Representation

Number Representation

Agenda & Reading

- Agenda:
  - Number Representation
    - Asian Numbers
    - Roman Numerals
    - Computer Based Numbers
  - Number Base
    - Binary
    - Decimal
    - Octal
    - Hexadecimal
  - Conversion
    - Converting from internal form into a base
    - Converting from a base into internal form
    - Converting between Binary, Octal and Hexadecimal
- Macromedia Director Shockwave Movie
  - Conversion.html
- Java Examples:
  - 01\Bin2Dec.html, 01\Dec2Bin.html, Converting.java

Number Representation

- Number
  - A number is a mathematical concept used to describe and assess quantity.
  - It is an abstract entity representing a quantity, used to express how many things are being referred to, or how much there is of some thing or property; an arithmetical value corresponding to a particular quantity of something.
  - There are many alternative ways of representing numbers:
    - Asian:
      - Chinese: 一, 二, 三 ...
      - Korean: 영, 일, 이 ...
      - Japanese: ???
    - Roman
      - I, II, III ...
    - Computer Based Numbers
      - 1, 0

Asian Numbers

	Chinese	Korean
0	零	영
1	一	일
2	二	이
3	三	삼
4	四	사
5	五	오
6	六	육
7	七	칠
8	八	팔
9	九	구

- Asian numbers are divided into two categories:
  - 1-9
  - 10, 100, 1000, 10^4
- Build up numbers out of these.
- Example:

	Chinese	Korean
10	十	십
100	百	백
1000	千	천
10000	万	만

百	十	万	千	百	十	
1	2	3	4	5	6	7

  - Chinese: 一百二十三万四千五百六十七
  - Korean:백이십삼만사천오백육십칠
- Note: If a digit is 0,
  - Japanese/Korean: We miss out both the digit, and the power of ten marker (2003 ->이십삼)
  - Chinese: We mark by a 零. (2003 ->二千零三)

Exercise: 1024 in Chinese and Korean?

- Note:
  - Korean: the 1 in front of a power of ten is always omitted
  - Japanese: a 1 is omitted in front of 10, 100, optional in front of 1000, and compulsory in front of 10000
  - Chinese: a leading 1 is usually explicitly indicated by a 一, except for numbers 10 to 19

Roman Numerals

1	I
5	V
10	X
50	L
100	C
500	D
1000	M

- Numbers are represented by writing down the following symbols multiple times, so the sum represents the number.
- Example:
  - 632 -> DCXXXII
- Note:
  - Digits corresponding to 4 and 9 are represented by preceding the representation of 5 or 10 by the representation of 1
  - Example: 40-> XL, 900 -> CM
  - Examples:
    - 1 to 10: I, II, III, IV, V, VI, VII, VIII, IX X
    - 10, 20 to 100: X, XX, XXX, XL, L, LX, LXX, LXXX, XC, C
    - 100, 200, to 1000: C, CC, CCC, CD, D, DC, DCC, DCCC, CM, M

Exercise: 1024 in Roman?

Computer Based Numbers

- Computers are built of electronic circuitry. Information is represented by currents.
- Two states: On/Off
  - Binary (base 2): 0/1
  - Hexadecimal (base 16)
  - Octal (base 8)
- Represent Number in any base :
  - A digit sequence

    $a_{n-1}\ a_{n-2} \dots a_2\ a_1\ a_0$

    represents the integer

    $a_{n-1} * base^{n-1} + a_{n-2} * base^{n-2} + \dots + a_2 * base^2 + a_1 * base^1 + a_0 * base^0$

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	a	1010
11	b	1011
12	c	1100
13	d	1101
14	e	1110
15	f	1111

Number Bases

A system of counting in tens

- Base 10 (Decimal)
  - Available digits : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Weighting - the quantity you multiply by to find the true value = 10
  - $123 = 1 * 10^2 + 2 * 10^1 + 3 * 10^0$
- Binary (Base 2)
  - Available digits : 0, 1
  - Weighting = 2
  - $101 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0$
- Base 8 (Octal)
  - Available digits : 0, 1, 2, 3, 4, 5, 6, 7
  - Weighting = 8
  - $123 = 1 * 8^2 + 2 * 8^1 + 3 * 8^0$
- Base 16 (Hexadecimal)
  - Available digits : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - Weighting = 16
  - $123 = 1 * 16^2 + 2 * 16^1 + 3 * 16^0$

Each binary digit is called a bit.

Conversion

- Convert from a base into internal form
  - Multiply each of the powers by its appropriate digit and add the values
  - Examples:
    - Binary to Decimal
    - Octal to Decimal
    - Hexadecimal to Decimal
- Convert from internal form into a base
  - Set the working value V to the number to convert. Then calculate
    - *d* = *V* % *base*; and *V* = *V* / *base* until (*V* = 0)
  - Examples:
    - Decimal -> Binary
    - Decimal -> Octal
    - Decimal -> Hexadecimal
  - Shockwave Movie: Conversion.html

Binary/Octal/Hex -> Decimal

- Binary to Decimal
  - **01 011 111$_2$**
    $= 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
    $= 0 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 =$ **95**
  - **11 010 001$_2$**
    $= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 1 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 =$ **209**
- Octal to Decimal
  - **127$_8$**
    $= 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0$
    $= 64 + 16 + 7 = 87$
  - **235$_8$**
    $= 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0$
    $= 128 + 24 + 5 = 157$
- Hexadecimal to Decimal
  - **1A7$_{16}$**
    $= 1 \times 16^2 + 10 \times 16^1 + 7 \times 16^0$
    $= 256 + 160 + 7 = 423$
  - **23E$_{16}$**
    $= 2 \times 16^2 + 3 \times 16^1 + 14 \times 16^0$
    $= 512 + 48 + 14 = 574$

01 101 001$_2$ = ??

151$_8$ = ??

15B$_{16}$ = ??

Decimal ->Binary/Octal/Hex

- Click here to view
- Base 10 -> 2
  - 60
  - 21
  - Exercise: 156
- Base 10 -> 8
  - 122
  - 220
  - Exercise: 1234
- Base 10 -> Hex
  - 1940
  - 175
  - Exercise: 689

Between Octal, binary & Hex

- Base 2 -> 8
  - 0101 1111
  - 1101 0001
  - Exercise: 101101001
- Base 2 – 16
  - 0101 1111
  - 1101 0001
  - Exercise: 101001101
- Base 8 -> 2
  - 363
  - 247
  - Exercise: 123
- Base 8 -> 16
  - 363
  - 247
  - Exercise: 123
- Base 16 -> 2
  - EA3
  - 2A7
  - Exercise: 1F3
- Base 16 -> 8
  - EA3
  - 2A7
  - Exercise: 1F3

Helper

```
Examples:
java Converting
Bin2Dec.html (applet)
Dec2Bin.html (applet)
```

- Calculator
  - You can use the Calculator program in the Scientific mode to perform conversion calculation.
    - Start -> Programs -> Accessories ->Calculator
- Java Program
  - Numbers
    - Octal numbers are always begin with a zero.
    - Hex numbers are always preceded by 0x.
  - You can also use the Java Integer Wrapper class to output binary, hex and octal Strings
  - You can use Integer.parseInt(String s, int base) method to parse the string argument as a signed integer in the base specified by the second argument.

```
int n = 061;  //n = 49 (decimal)
int n = 0x31;  // n = 49 (decimal)
```

```
Integer.toBinaryString(octalNum);   // output = 11010101
Integer.toOctalString(octalNum);   // output = 325
Integer.toHexString(octalNum);   // output = D5
```

```
int n = Integer.parseInt(“1010”, 2); //n=10
int n = Integer.parseInt(“FF”, 16);  //n=255
int n = Integer.parseInt(“-FF”, 16); //n=-255
```