Part 2: Number Systems

* Number system
  + A number is represented by a string of digits
    - An-1An-2…. A0 where An-1, An-2, …. A0 is one of possible digit number in a number system
    - Then we said that An-1 is the most significant digit of the number and A0 is the least significant digit of the number.
  + Decimal number
    - Use 10 discrete digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) – Base 10
    - A number is represented by a string of digits
    - Depending on its position in the string, each digit has an associated value of an integer raised to the power of 10
    - Ex.: 1209.5 = 1 x 103 + 2 x 102 + 0 x 101 + 9 x 100 + 5 x 10-1
  + Binary number
    - Use 2 discrete digits (0, 1) – Base 2
    - A number is represented by a string of digits
    - Depending on its position in the string, each digit has an associated value of an integer raised to the number of 2
    - Ex) (1001.1)2 = 1 x 23 + 0 x 22 + 0 x 21 + 1 x 20 + 1 x 2-1 = 8 + 0 + 0 + 1 + 0.5 = (9.5)10
  + Octal number
    - Use 8 different digits (0, 1, 2, 3, 4, 5, 6, 7) – Base 8
    - A number is represented by a string of digits
    - Depending on its position in the string, each digit has an associated value of an integer raised to the power of 8
    - Ex) (725.3)8 = … = 448 + 16 + 5 + 0.375 = (469.375)10
  + Hexadecimal number
    - Use 16 different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) – Base 16
    - A number is represented by a string of digits
    - Depending on its position in the string each digit has an associated value of an integer raised to the power of 16
    - Ex) (B65F)16 = 11 x 163 + 6 x 16 2 + 5 x 161 + 15 x 160 = 45056 + 1536 + 80 + 15 = (46687)10
* Converting number systems
  + Division by base and reading remainders from the bottom up!
  + Decimal to Binary
    - 1310 = 8 + 4 + 1 = 23 + 22 + 20 = (1101)2
    - 2010 = 16 + 4 = (10100)2
  + Decimal to Octal
    - 15310:
      * 153/8 = 19, remainder = 1
      * 19/8 = 2, remainder = 3
      * 2/8 = 0, remainder = 2
    - 15310 = 2318
    - We can also convert a decimal number to octal indirectly (decimal to binary to octal)
  + Decimal to Hexadecimal
    - Convert a decimal 239810 to hexadecimal
      * 2398 / 16 = 149, remainder = 14 = E
      * 149 / 16 = 9, remainder = 5
      * 9 / 16 = 0, remainder = 9
      * 239810 = 95E16
  + Decimal Fraction to Binary
    - Divide by 2-1 which multiplies by 2 and take the number behind the decimal reading to next line
    - read top to bottom
  + Binary and Hexadecimal
    - Since one digit hexadecimal number can convert to 4 digit binary number, conversion between binary and hexadecimal is very simple
    - Conversion from 10010010101001 to hexadecimal
      * Divide a binary number into 4 groups with 4 bits (divide from least significant bit)
      * Now each group convert to hexadecimal number for each group
* Arithmetic in Binary
  + Similar in decimal
    - Decimal: 10 discrette symbols (0,1,2,3,4,5,6,7,8,9)
    - Binary: 2 discrete symbols (0,1)
  + Decimal addition and binary addition: carrying
  + Subtraction: borrowing
* Number ranges
  + In digital computers, the range of numbers that can be represented is based on the number of bits available in the hardware structures that store and process information
  + Integer: positive or negative
  + Integer representation: unsigned integer and signed integers
* Formats (see notebook for more and practice
  + Unsigned: ALL POSITIVE
    - 1. Convert to binary
    - 2. Add ‘0’ up to N (number of bits in memory)
  + Sign and magnitude: Positive and negative numbers
    - 1. Convert to binary
    - 2. Add 0 up to N-1
    - 3. Use MSB for sign
    - Add:
      * 0 for positive
      * 1 for negative
  + One’s complement: positive and negative numbers
    - Convert to binary
    - Add 0 up to N / stop here if positive
    - Compliment the number if negative (interchange 0s and 1s)
  + Two’s complement: positive and negative numbers
    - Convert to binary
    - Add 0s up to N / stop here if positive
    - Calculate one’s compliment
    - Add 1 to one’s compliment
* Denoting One’s compliment
  + N = 8 bits
    - 00000011 – 0 is the most significant bit, so its positive = +3
    - 11111101 – 1 is MSB, so its negative – 00000010 = -2
  + If positive, just convert from binary and add +
  + If negative, complement the number and convert from binary and add –
* More about One’s Complement
  + Not used to store numbers in computers today
  + Foundation of two’s complement
  + Interesting for data communication application such as error detection and correction
* Denoting Two’s Complement
* Overflow with Two’s Complement
  + Rules for detecting overflow in a Two’s Complement sum:
    - If the sum of two positive numbers yields a negative result, the sum is overflowed
    - If the sum of two negative numbers yields is a positive result, the sum has overflowed
    - Otherwise, the sum has not overflowed.
  + The reason for the rules is that overflow in twos complement occurs, not when a bit is carried out of the left column, but when one is carried into it
  + That is when there is a carry into the sign
  + The rules detect this error by examining the sign of the result, a negative and positive added together cannot overflow, because the sum is between the addends
  + Since both addends fit within the allowable range of numbers, and their sum is between them, it must fit as well
* Binary Coded Decimal (BCD)
  + Another way to save decimal information with binary numbers (0.1)
  + Input-Output equipment generally uses decimal numbers. Because most logic circuits only accept two-value signals, the decimal numbers must be coded in terms of binary signals.
  + Decimal representation with BCD digit
    - (9821)10 = (1001 1000 0010 0001)­BCD
    - (12003)10 = (0001 0010 0000 0000 0011)BCD
  + BCD number need more memory to store numbers
  + The results is quite different than that obtained by converting the number as a whole into binary
  + Because there are only 10 decimal digits 1010 through 1111 are not valid BCD codes
* Counting table for reference

