Number System Conversion
Binary Arithmetic

Michael Brodskiy
Professor: S. Shazli

January 18, 2023

- Binary Numbers
  - All computers work with 0's and 1's so it is like learning alphabets before learning a language
- Number Systems
  - There is more than one way to express a number in binary, so 1010 could be -2, -5, or -6, and it is necessary to know which one
- A/D and D/A Conversion
  - Real world signals come in continuous/analog format, and it is good to how they become 0's and 1's (and vice versa)
- Digital = Discrete
  - Binary Codes
    - Represent symbols using binary digits (bits)
  - Decimal digits 0-9
- Digital Computers
  - I/O is digital
  - Internal representation is binary
  - Sometimes use hexadecimal (base 16) for large binary numbers
- Bases we will use
  - Binary: Base 2

  - Octal: Base 8
  - Decimal: Base 10
  - Hexadecimal: Base 16

- Positional Number System
  - $101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
  - $63_8 = 6 \cdot 8^1 + 3 \cdot 8^0$
  - $A1_{16} = 10 \cdot 16^1 + 1 \cdot 16^0$

- Conversion from binary to octal/hex
  - Binary: 10011110001
  - Octal: $10|011|110|001 = 2361_8$
  - Hex: $100|1111|0001 = 4F1_{16}$

- Conversion from binary to decimal
  - $101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_{10}$
  - $63.4_8 = 6 \cdot 8^1 + 3 \cdot 8^0 + 4 \cdot 8^{-1} = 51.5_{10}$
  - $A1_{16} = 10 \cdot 16^1 + 1 \cdot 16^0 = 161_{10}$

- Decimal to binary/octal/hex
  - Binary: 56 becomes 111000_2

	Quotient	Remainder
56/2 =	28	0
28/2 =	14	0
14/2 =	7	0
7/2 =	3	1
3/2 =	1	1
1/2 =	0	1

  - Octal: 56 becomes 70_8

	Quotient	Remainder
56/8 =	7	0
7/8 =	0	7

  - Each successive divide releases another LSB (least significant bit)

- Negative Numbers
  - Sign-and-magnitude

    - The most-significant bit (MSB) is the sign digit, where 0 is positive and 1 is negative
    - The remaining bits are the numbers magnitude
  - Ones-complement
    - Negative number: a bitwise complement positive number
  - Twos-complement
    - Most important option
    - Simplifies arithmetic
    - Used almost universally
    - Negative number: a bitwise complement plus one
      - $0011 = 3_{10}$
      - $1101 = -3_{10}$
    - Only one zero
    - MSB is the sign digit, where 0 is positive and 1 is negative
    - Overflow: summing two positive numbers gives a negative result and vice versa

- Signed-Complement Arithmetic
  - Addition
    - Add the numbers including the sign bits, discarding a carry out of the sign bits (2's Complement)
    - If the sign bits were the same for both numbers and the sign of the result is different, an overflow has occurred
    - The sign of the result is computed in step 1
  - Subtraction
    - Form the complement of the number you are subtracting and follow the rules for addition