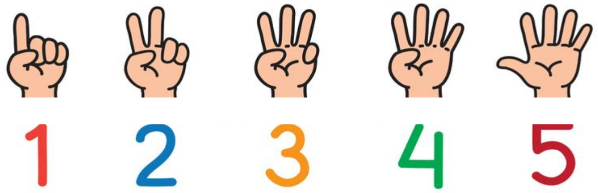
CPEG323: Intro. to Computer System Engineering

Lecture 02: Number Representation

Decimal Numbers : Base 10



- Natural to human beings
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Inefficient for computer
  - They store data as an ordered sequence of binary digits : on and off signal, true and false, 1 and 0.

How to represent integers in binary?

Positional Number System

- Decimal representation example:

$$4231$$

$$4000 + 200 + 30 + 1$$

$$4 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$$

Positional Number System Cont.

- **Terminology:** Digit *d* and Base *B*
  - In base B : B symbol per digit *d*
    - Decimal ➔ *B*=10 and 10 symbols per *d*
  - **n digits in base B can represent B^n different numbers**

- **Representation**:

$$\boxed{\text{n digit number in base } B = d_{n-1}\, d_{n-2} \;\ldots\; d_1\, d_0}$$

$$\Downarrow$$

$$\boxed{\text{Value} = d_{n-1} \times B^{n-1} + d_{n-2} \times B^{n-2} + \ldots + d_1 \times B^1 + d_0 \times B^0}$$

Commonly used number bases

- Decimal (Base 10)
  - 10 symbols per digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Notation: 1234_{ten} = 1234
- Binary (Base 2)
  - 2 symbols per digit: 0, 1
  - Binary digits are called bits
  - Notation: 1110_{two} = 0b1110
- Hexadecimal (Base 16)
  - 16 symbols per digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - Notation: $1A5E_{hex}$ = 0x1A5E
  - Each hexadecimal digit can be converted to 4 bits

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion Examples

$(1010)_{two}$

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ ➡ $(1010)_{two} = 10$

$(1010)_{hex}$

$1 \times 16^3 + 0 \times 16^2 + 1 \times 16^1 + 0 \times 16^0$ ➡ $(1010)_{hex} = 4112$

Questions

- $(111)_{two}$ = ________?
- 11 = 0b________?
- $(1010\ 0001\ 0011)_{two}$ = 0x____________?
- 0xB9EF = (____________)$_{two}$?
- $(1000001)_{two}$ = ________?

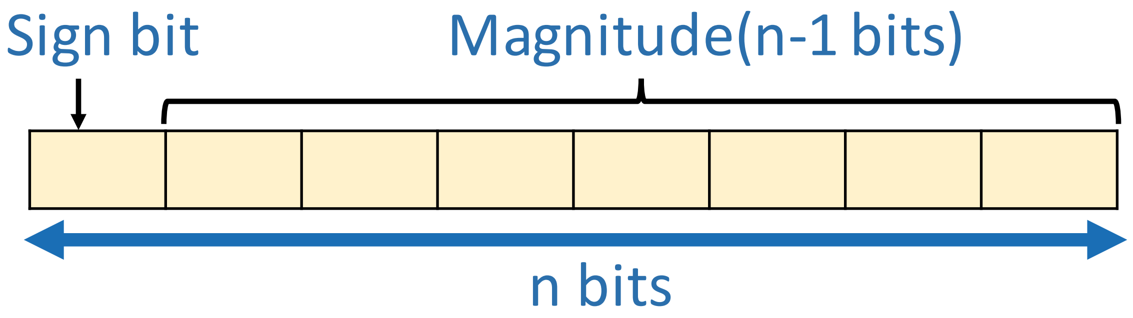
Unsigned v.s. Signed

- Unsigned: n digits in base B can represent B^n different numbers
- Signed: positive and negative numbers

- **Challenge:** Represent signed numbers in a way that
  both positive and negative numbers can be calculated
  using the same hardware

Sign and Magnitude

- Separate sign bit : 0 for positive and 1 for negative



- 8 bit representation example:
  - 8 bit representation of 12 : 0000 1100
  - 8 bit representation of -12: 1000 1100
- What are the shortcomings?
  - Arithmetic circuits get complicated
  - Two zeros

One’s Complement

- To negate a positive number: flip all of its bits
- 8 bit representation example :
  - $12 = (0000\ 1100)_{two} \rightarrow -12 = (1111\ 0011)_{two}$

- Positive numbers have leading 0s
- Negative numbers have leading 1s

- Any shortcoming?
  - Still have two zeros

Two’s complement

- To negate a positive number: Flip all of its bits and add 1
- 8 bit representation example :
  - 12 = $(0000\ 1100)_{two}$ ➔ Flip: $(1111\ 0011)_{two}$ + 1 ➔ -12 = $(1111\ 0100)_{two}$

An n bit two’s complement number value

$$= -2^{n-1} \times d^{n-1} + 2^{n-2} \times d^{n-2} + \ldots + 2^{1} \times d^{1} + 2^{0} \times d^{0}$$

- Almost split numbers evenly between positive and negative:
  - One negative number that has no corresponding positive number. Which one?
  - The single pattern for zero belongs to positive numbers

Questions

- 6-bit two’s complement representation of -19 ?
- 5-bit two’s complement representation of -3 ?
- What range of numbers can be presented by a 4-bit two’s complement?

    A. [-15 , 15]

    B. [-8, 8]

    C. [0, 15]

    D. [-8, 7]

    E. [-16, 15]

Overflow

- A condition where the result of an arithmetic operation cannot be represented by the hardware
  - Number of bits in the binary representation is insufficient to show the result

- Example

$$\begin{array}{r} -7 \\ + \quad -6 \\ \hline -13 \end{array} \qquad \begin{array}{r} 1\ 0\ 0\ 1 \\ + \quad 1\ 0\ 1\ 0 \\ \hline 0\ 0\ 1\ 1 \end{array} = +3 \ ✗$$

Reading assignment

- Section 2-4