

## Basic Concepts (cont.)

---

### Overflow

- Calculate  $127 + 127$  by using 8-bits 2's complement

0111 1111  $\leftrightarrow$  127

0111 1111  $\leftrightarrow$  127

-----

1111 1110  $\leftrightarrow$  -2

(usually the calculation result uses the same representation as the operands;  $\text{int} + \text{int} = \text{int}$ )

- The right answer should be 254, which is out of the range of 8-bits in 2's complement.
  - The error is caused by overflow.
  - 2's complement can be thought as a clock, half of it is positive values and the other is negative value. If a positive value exceeds the boundary, it actually enters in the half for the negative values. So it generates an incorrect value.
  - This is why the InfiniteLoop example we discussed in the first class generates a negative value and does not loop forever.
- 

### Other Useful Representations

- **Octal** number
  - only need 8 numbers: 0~7
  - use 3 bits to represent one digit
- **Hexadecimal** number
  - need 16 numbers: 0~9 and A-F
  - use 4 bits to represent one digit
- Example: 2024 in binary, octal, and hexadecimal (use unsigned number)
  - binary: (111 1110 1000)<sub>2</sub>
  - octal: 011 111 101 000 (3 7 5 0)<sub>8</sub>
  - hexadecimal: 0111 1110 1000 (7 E 8)<sub>16</sub>
- **ASCII**: a type of character code, to map character onto integers
  - Each computer has a set of characters that it uses, such as 26 uppercase letters and 26 lowercase letters. In order to transfer these characters into computer, each one is assigned a number.
  - These numbers can be converted into binaries so that characters can be stored in a computer too.
  - Originally, each ASCII character has 7 bits, allowing for 128 characters in all. Then, to support more characters, the table is extended to 255 characters, using 1 byte to represent.
  - Online ASCII table: <http://www.asciitable.com>
- Images: pixels (RGB values)
- Videos: flows of images and voice data