1 Беззнаковые целые числа

Чтобы найти 6-разрядную двоичную форму целого числа без знака N (где $0 <= N < 64$), мы многократно проверяем, соответствуют ли степени двойки (от 2^5 до 2^0) числу N, вычитая, когда они совпадают, и ставя "1", или ставя "0", если совпадают нет.

1.1 Примеры

- $N = 0_{10}$

$$0 \div 2^5 (= 32) = 0 \Rightarrow 0_{10} = 000000_2$$

- $N = 13_{10}$

$$\begin{aligned}
13 \div 2^5(= 32) = 0 &\quad (0 \cdot 2^5 = 0)\\
13 \div 2^4(= 16) = 0 &\quad (0 \cdot 2^4 = 0)\\
13 \div 2^3(= 8) = 1 &\quad (1 \cdot 2^3 = 8)\\
5 \div 2^2(= 4) = 1 &\quad (1 \cdot 2^2 = 4)\\
1 \div 2^1(= 2) = 0 &\quad (0 \cdot 2^1 = 0)\\
1 \div 2^0(= 1) = 1 &\quad (1 \cdot 2^0 = 1)\\
13_{10} = 001101_2 &
\end{aligned}$$

- $N = 24_{10}$

$$\begin{aligned}
24 \div 2^5(= 32) = 0 &\quad (0 \cdot 2^5 = 0)\\
24 \div 2^4(= 16) = 1 &\quad (1 \cdot 2^4 = 16)\\
8 \div 2^3(= 8) = 1 &\quad (1 \cdot 2^3 = 8)\\
0 \div 2^2(= 4) = 0 &\quad (0 \cdot 2^2 = 0)\\
0 \div 2^1(= 2) = 0 &\quad (0 \cdot 2^1 = 0)\\
0 \div 2^0(= 1) = 0 &\quad (0 \cdot 2^0 = 0)\\
24_{10} = 011000_2 &
\end{aligned}$$

- $N = 63_{10}$ Поскольку $63_{10} = 2^6 - 1$, все биты равны 1:

$$63_{10} = 111111_2$$

1.2 Примеры

В 6-битном представлении с дополнительным знаком:

- Неотрицательные числа преобразуются так же, как и в случае с беззнаковыми числами (просто нужно убедиться, что они помещаются в 6 бит).

- Отрицательные числа требуют:
  (1) записи двоичной формы модуля числа,
  (2) инверсии всех битов (дополнение до единицы), затем
  (3) прибавления 1.

16_{10}

$$16_{10} \div 2^5 (= 32) = 0, \quad 16 \div 2^4 (= 16) = 1, \cdots \Rightarrow 16_{10} = 010000_2$$

-2_{10}

$$d2_{10} = 000010_2$$
$$\text{Инвертируем биты} \Rightarrow 111101_2$$
$$\text{Прибавляем } 1 \Rightarrow 111110_2$$

-31_{10}

$$31 = 00011111_2 \text{ (но используем только 6 бит)} \Rightarrow 011111_2$$

-32_{10}

32 не помещается в 5 бит, но в 6 битах это 100000_2.

2 Преобразование 6-битных двоичных чисел в десятичные

Рассматриваются как *беззнаковое* (unsigned), так и *дополнительные* (two's complement) представления.

$$\text{Беззнаковая формула: Value} = \sum_{i=0}^{5} b_i \cdot 2^i$$

$$\text{Формула дополнительного кода: Value} = -b_5 \cdot 2^5 + \sum_{i=0}^{4} b_i \cdot 2^i$$

Для каждого 6-битного числа:

- **000101**:

$$\text{Беззнаковое} = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 = 1 + 4 = 5$$

$$\text{Знаковое (старший бит = 0)} = 5$$

- **101011**:

$$\text{Беззнаковое} = 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 32 + 8 + 2 + 1 = 43$$

$$\text{Знаковое} = -1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = -32 + 11 = -21$$

- **111111**:

$$\text{Беззнаковое} = 63$$

$$\text{Знаковое} = -32 + 31 = -1$$

- **100000**:

$$\text{Беззнаковое} = 1 \cdot 32 + 0 = 32$$

$$\text{Знаковое} = -32 + 0 = -32$$

3 Преобразование десятичных в 8-битные шестнадцатиричные

Чтобы получить 8-разрядную шестнадцатеричную форму, надо преобразовать десятичную систему счисления в двоичную (до 8 разрядов), затем сгруппировать каждый элемент (4 бита) и преобразуйте каждый элемент в отдельную шестнадцатеричную цифру.

- 7_{10}:

$$7_{10} = 00000111_2 \Rightarrow 0x07$$

- 240_{10}:

$$240_{10} = 11110000_2 \Rightarrow 0xF0$$

- 171_10:

$$171_{10} = 10101011_2 \Rightarrow 0xAB$$

- 126_{10}

$$126_{10} = 01111110_2 \Rightarrow 0x7E$$

4 Преобразование из шестнадцатиричной в 8-битную двоичную

Каждая шестнадцатеричная цифра преобразуется в 4-разрядный двоичный код. Например, 3 = 0011, C = 1100 и т.д.

- $0x3C$:

$$3_{16} = 0011_2,\, C_{16} = 1100_2 \Rightarrow 00111100_2$$

- $0x7E$:

$$7_{16} = 0111_2,\, E_{16} = 1110_2 \Rightarrow 01111110_2$$

- $0xFF$:

$$F_{16} = 1111_2 \Rightarrow 11111111_2$$

- $0xA5$:

$$A_{16} = 1010_2,\, 5_{16} = 0101_2 \Rightarrow 101001101_2$$

5 Negate двоичных чисел

Чтобы negate 8-битное двоичное значение b, мы переворачиваем все биты и добавляем 1.

- $0x3C$:

$$3C_{16} = 00111100_2 \Rightarrow\ = 11000011_2,\ +1 = 11000100_2$$

- $0x7E$:

$$7E_{16} = 01111110_2 \Rightarrow\ = 10000001_2,\ +1 = 10000010_2$$

- $0xFF$:

$$FF_{16} = 11111111_2 \Rightarrow\ = 00000000_2,\ +1 = 00000001_2$$

- $0xA5$:

$$A5_{16} = 101001101_2 \Rightarrow\ = 010110010_2,\ +1 = 010110011_2$$

6 $Big-Endian\ vs.\ Little-Endian(0xDEADBEEF)$

Конвертация Endian'а описывает, как многобайтовые данные хранятся в памяти:

- $Big-Endian$:

  The most significant byte (MSB) is stored at the lowest memory address.

$$\text{Byte order: } \underbrace{DE}_{MSB} \mid \text{AD} \mid \text{BE} \mid \underbrace{EF}_{LSB}$$

- $Little-Endian$:

  The least significant byte is stored at the lowest memory address.

$$\text{Byte order: } \underbrace{DE}_{LSB} \mid \text{AD} \mid \text{BE} \mid \underbrace{EF}_{MSB}$$

7 Преобразование десятичных значений в 5-битные двоичные значения и дополнение их знаковыми и нулевыми битами до 8-битных двоичных значений.

Convert the given decimal numbers to 5-bit two's complement (if negative) or pure binary (if positive). Then:

- **Sign Extention**:

  For negative values, fill the higher bits with 1.

- **Zero Extention**:

  For non-negative values, fill the higher bits with 0.

- 7_{10}:
  $7 = 00111_2(5 - bit).\ Sign - extend \Rightarrow 00000111_2(8 - bit)$
- 15_{10}:
  $15 = 01111_2(5 - bit) \Rightarrow Extended = 00001111_2$
- -16_{10}:
  $16_{10} = 10000_2(5 - bits).\ Flip = 01111_2,\ +1 \Rightarrow 10000.\ Extend(negative) = 11110000_2$
- -5_{10}:
  $5_{10} = 00101_2(5 - bit).\ Flip = 11010_2,\ +1 \Rightarrow 11011.\ Extend(negative) = 11111011_2$

8 Преобразование пар десятичных чисел в 4-битные двоичные и их сложение

We convert each decimal to 4-bit representation, then add.

- **Unsigned:** $7 + 9$:
  $$7_{10} = 0111_2 \quad 9_{10} = 1001_2$$
  $$0111_2 + 1001_2 = 10000_2$$
  5 bits if there is no any space
- **Signed:** $4 + (-5)$:
  $$4_{10} = 0100_2 \quad -9_{10} = 1011_2$$
  $$0100_2 + 1011_2 = 1111_2$$
  In 4-bit two's complement, 1111_2 represents 1, which is the correct result for $4 + (-5) = -1$.