Base conversion

Base 2

Assembly language programming
By xorpd

xorpd.net

Objectives

- You will learn about three ways to convert between different number representations.
- You will understand how to choose the right way for every case of conversion.
- You will experience conversion of numbers between the Binary and Decimal systems, and vice versa.

Why is this important?

- You work with decimal numbers most of your life. Sometimes you would like to know what a binary number looks like in decimal.
- You need to know how to decode decimal numbers into base 2, to share them with the processor.
- Understanding conversion methods will unexpectedly give you some new insights about numbering systems.

Binary to Decimal

- We are given a binary representation of some quantity. We want to find its decimal representation.
- Conversion is pretty straightforward. We calculate the value of the number according to the definition.
- $1011_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 = 1 + 2 + 0 + 8 = 11_{10}$
- $11110_2 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 0 + 2 + 4 + 8 + 16 = 30_{10}$
- This side of the conversion is easy, because we are used to calculations in base 10.
- Works for other arbitrary bases too:
  - $12011_3 = 1 \cdot 3^0 + 1 \cdot 3^1 + 0 \cdot 3^2 + 2 \cdot 3^3 + 1 \cdot 3^4 = 1 + 3 + 0 + 54 + 81 = 139_{10}$

Decimal to Binary -- Direct evaluation

- We are given a Decimal representation of quantity, and we want to find the Binary representation.
- Example: What is the binary representation of 13_{10}?
- **Direct evaluation:**
  - $13_{10} = 3 \cdot 10^0 + 1 \cdot 10^1 = 11_2 \cdot {1010_2}^0 + 1_2 \cdot {1010_2}^1 = 11_2 + 1010_2 = 1101_2$
  - We had know that $10_{10} = 1010_2$, and how to multiply and exponent in base 2.
  - This is just like the case of converting Binary to Decimal, however here the calculation is not comfortable.
  - Not fun for someone who is used to base 10. (Might be fun for a computer, though).

Decimal to Binary -- Finding largest power

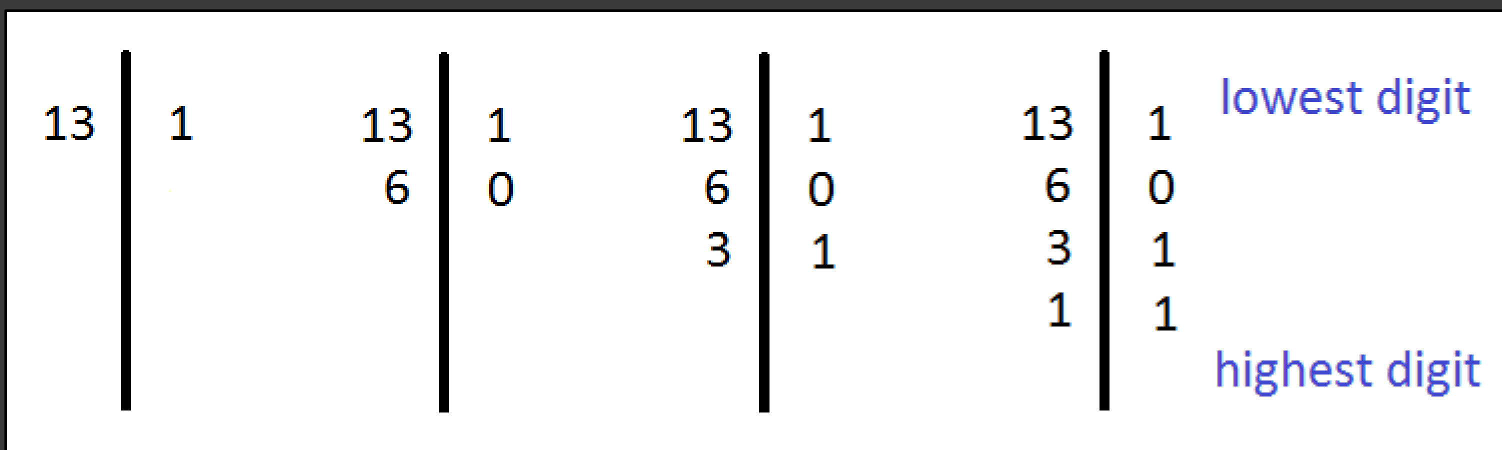
- We are given a Decimal representation of quantity, and we want to find the Binary representation.
- Example: What is the binary representation of 13_{10}?
- **Finding largest power:**
  - We know that the binary representation is just sums of powers of 2.
  - We will try to find the largest power of 2 that fits into 13_{10}: That would be $2^3 = 8$. ($2^4 = 16$ is too large). We are then left with $13_{10} - 2^3 = 5_{10}$.
  - The largest power of 2 that fits into 5_{10} is $2^2 = 4$. We are then left with $5_{10} - 2^2 = 1_{10}$.
  - The largest power of 2 that fits into 1_{10} is $2^0 = 1$. We are then left with $1_{10} - 2^0 = 0$, and we are done.
  - We conclude that $13_{10} = 2^3 + 2^2 + 2^0 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$.
- Interesting fact: Every number could be composed to a sum of distinct powers of 2 in a unique way.

Decimal to Binary -- Remainder evaluation

- We are given a decimal representation of some quantity, and we want to find the Binary representation.
- Example: What is the binary representation of 13_{10}?
- **Remainder evaluation**:
- Finding the first digit:
  - We want some number $(\dots b_3\, b_2 b_1 b_0)_2 = 13_{10}$
  - $(\dots b_3\, b_2 b_1 b_0)_2 = b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 + \cdots = 13_{10}$
  - Could the number 13_{10} reveal anything about the values $b_0, b_1, b_2, \dots$?
  - 13_{10} is odd ($13_{10} \equiv 1(mod\ 2)$). Therefore the sum on the left must be odd. Hence $b_0 = 1$.
  - We could remove b_0 from both sides, and then divide by 2. We will then obtain: $(\dots b_3 b_2 b_1)_2 = b_1 \cdot 2^0 + b_2 \cdot 2^1 + b_3 \cdot 2^2 + \cdots = \frac{13-1}{2} = 6_{10}$
- Finding the second digit:
  - Could the number 6_{10} reveal anything about the values $b_1, b_2, b_3, \dots$?
  - 6_{10} is even. ($6_{10} \equiv 0(mod2)$). Therefore the sum on the left must be even. Hence $b_1 = 0$.
  - We could remove b_0 from both sides, and then divide by 2. We will then obtain: $(\dots b_4 b_3 b_2)_2 = b_2 \cdot 2^0 + b_3 \cdot 2^1 + b_4 \cdot 2^3 + \cdots = \frac{6-0}{2} = 3_{10}$.
- We keep going until we find all digits.

Decimal to Binary -- Remainder evaluation (Cont.)

- We usually use a table for the remainder evaluation method.
- In every step we check the remainder of the number in the left when divided by 2. We write the remainder on the right side of the line, and the result of division by 2 in the left side of the line.



- Again we get that $13_{10} = 1101_2$.

Which method to use?

- We can convert a number from any base to any base using each of the 3 mentioned methods.
- In some cases one method might be more comfortable than another. (Given that you are a human being).
- Conversion from a small base to a larger base: use **Direct evaluation.**
  - **Example: from base 2 to base 10.**
- Conversion from a large base to a smaller base: use **Remainder evaluation**.
  - Example: from base 10 to base 2.

Example (Decimal to Binary)

- How to represent 30145_{10} in base 2?
  - We will use the **remainder evaluation** method, because 10 > 2.
  - We obtain: $30145_{10} = 111010111000001_2$

30145	1
15072	0
7536	0
3768	0
1884	0
942	0
471	1
235	1
117	1
58	0
29	1
14	0
7	1
3	1
1	1

Example (Binary to Decimal)

- How to represent 1010111_2 in base 10?
  - We will use the **direct evaluation** method, because 2 < 10.

- $1010111_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 =$

$$2^0 + 2^1 + 2^2 + 2^4 + 2^6 =$$

$$1 + 2 + 4 + 16 + 64 = 87_{10}$$

Example (Decimal to Ternary(3))

- How to represent 101_{10} in base 3? (The ternary system).
  - We will use the remainder evaluation method, because 10 > 3.
  - This time we will divide by 3, and calculate remainder modulo 3.

101	2
33	0
11	2
3	0
1	1

- We obtain: $101_{10} = 10202_3$

Exercises

- Conversion between bases.
- Some interesting questions.