**IS 312 Information Technology Handout 1: Binary Number System (V. 6)**

*Dr. Yüe “Jeff” Zhang*

Binary number system is the system used inside computers. It is important for one who is learning about computer applications to understand the binary system. Binary system can be understood with our knowledge in the decimal system. This handout takes this approach.

I. The essence of the decimal system

Meaning of a decimal number

We “live in” the decimal system that we never thought how we interpret the numbers expressed in the decimal system. Now let’s take a closer look:

Suppose we want to express the number “one thousand, four hundred, and ninety-two,” we would write the number 1492. This is indeed putting the four digits in a decimal “scale”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Ten Thousands** | **Thousands** | **Hundreds** | **Tens** | **Ones** |
| **Powers of 10** | 10000 (**104**) | 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |
| Digit in place – Example 1492 | (in 1492, there is no ten thousand) | 1 | 4 | 9 | 2 |
| Meaning/Value | “0 unit of a ten thousand” | “1 unit of a thousand” | “4 units of a hundred” | “9 units of a ten” | “2 units of a one” |
| Value =  | (0 X 1000 +) | 1 X 1000 + | 4 X 100 + | 9 X 10 + | 2 X 1 |

You all agree with the above – just that you never thought about that.

Note and summary 1: decimal numbers are the number of units of “powers of 10”, added together. **NOTE: Put the digits in the decimal scale from the RIGHT END (the Unit) ! !**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 10000 (**104**) | 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |
|  | **Scale from right to left!! 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 From right to left** |
| Do we do this? Or, | (0) | 1 | 4 | 9 | 2 |
| *… do we do this?* | *1* | *4* | *9* | *2* |  |

Counting in the decimal system

We also need to note the way we count in decimal system: after we exhaust all available single digit numbers, we go from “9” to “1”and “0” – “10”, the BEGINNING of the next higher level; and when we exhaust all available two-digit combinations, we go from “99” to “1” and “00” – “100”, the BEGINNING of the next higher level.

In a somewhat more “mathematical” language:

the next number after the largest single digit number (0~9, in the order of magnitude **100**), is the first number in the next order of magnitude (**101** level - 10~99);

the next number after the order of magnitude **101** or 11~99, is in the order of magnitude **102** or 100~999;

the next number after the order of magnitude **102** or 100~999, is in the order of magnitude **103** or 1000~9999; and so on.

Note and summary 2: when we exhaust all available numbers in the previous order of magnitude, we move up to the next order of magnitude - after 9 it’s 10 (**101**), after 99 it’s 100 (**102**), after 999 it’s 1000 (**103**), etc.

II. The essence of the binary system 【Think in PARALLEL of we just learned above】

A. Meaning of a **BINARY** number

Just as we analyzed in last page, the binary system takes similar form as the decimal:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Powers of “10” (i.e., 2 base 10: 2**10** ) | 10000 (**104**) | 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |
|  | **Scale from right to left!! 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 From right to left** |
| The above “translated” into decimal | 16 | 8 | 4 | 2 | 1 |
| Digit in place: Example – 1 0 0 1 1 | 1 | 0 | 0 | 1 | 1 |
| Meaning/Value | “1 unit of a sixteen” | “0 unit of an eight” | “0 unit of a four” | “1 unit of a two” | “1 unit of a one” |
| Value = 10011**2**= | 1 X **104** + | 0 X **103** + | X 10**2** + | 1X 10**1** + | 1X 10**0** |
| (explained in decimal) 19**10**  | 1X 16 + | 0 X 8 + | 0 X 4 + | 1 X 2 + | 1 X 1 |

III. Conversion between binary and decimal (will be tested)

A. Binary to decimal: Use **addition**. Examples:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Note: binary “10” now means decimal 2!! | 16 | 8 | 4 | 2 | 1 |  |
| Powers of 10 | 10000 (**104**) | 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |  |
| Remind the last time: | **Scale from right to left!! 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 🡸 From right to left** |  |
| Example 1: 10010 | 1 | 1 | 1 | 0 | 0 |  |
| Value = 10010**2**= | 1 X 16 + | 0 X 8 + | 0 X 4 + | 1 X 2 + | 0 X 1 | =18**10** |
| Example 2: 10011 | 1 | 0 | 0 | 1 | 1 |  |
| Value = 10011**2**= | 1 X 16 + | 0 X 8 + | 0 X 4 + | 1 X 2 + | 1 X 1 | =19**10** |

Tips of doing this right:

1. Have the scales right: 1, 2, 4, 8, 16, 32,… AND:

2. Have the scale numbers filled out FROM RIGHT! – Right most: 1, then to its left 2, then to its left 4, then to its left 8, etc.

B. Decimal to binary: use **subtraction**. Examples – convert 20, 28, 29 to binary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | Has | BinScale | Remain |  | 28 | Has | BinScale | Remain |  | 29 | Has | BinScale | Remain |
| 20 | 1 | 16 | 20-16 =4 |  | 28 | 1 | 16 | 28-16 =12 |  | 29 | 1 | 16 | 29-16 =13 |
| 4 | 0 | 8 | 4 |  | 12 | 1 | 8 | 12-8=4 |  | 13 | 1 | 8 | 13-8=5 |
| 4 | 1 | 4 | 0 |  | 4 | 1 | 4 | 4-4=0 |  | 5 | 1 | 4 | 5-4=1 |
| 0 | 0 | 2 |  |  | 0 | 0 | 2 | 0 |  | 1 | 0 | 2 | 1 |
| 0 | 0 | 1 |  |  | 0 | 0 | 1 |  |  | 1 | 1 | 1 |  |
|  |  | Result: | 10100 |  |  |  | Result: | 11100 |  |  |  | Result: | 11101 |

 Read: Read: Read:

Explanation of example 1 above (converting 20 to binary number):

1. Take the number 20, ask the question “does 20 contain 16 (the highest in the current binary scale; it can be 32 or 64 or even higher, but here we only go as high as 16 given the decimal numbers that need to be converted);
2. The answer is “Yes”; so there is ONE (1) 16 in the decimal number 20; we log a “1” in the column to the left;
3. Since a 16 has been accounted for in the number 20, it has to be “removed” from the decimal number 20: 20-16 = 4; log that number in the “Remainder” column;
4. Then we ask: “Does 4 contain 8, the next (lower) number on the binary scale?”;
5. The answer is “No”, so there is ZERO (0) 8 in the decimal number 4; we log a “0” in the column to the left;
6. Since there is no binary scale number was accounted for, the decimal number 4 is “intact”, and we still have the same number, i.e., 4, in the “Remainder” column;
7. We continue on the next step as ask: “Does 4 contain 4, the next (lower) number on the binary scale?”;
8. The answer is “Yes”; so there is ONE (1) 4 in the decimal number 4; we log a “1” in the column to the left;
9. Since a 4 (binary scale number) has been accounted for in the decimal number 4 (the remainder of the decimal number from previous step), it has to be “removed” from the decimal number: 4-4 = 0; log that number in the “Remainder” column; …
10. Repeat the above procedure until the lowest level of the binary scale (20 = 1) is being compared.

Now, do you feel comfortable?

Also, observe the last digit of the above binary numbers, what observation may you make regarding even or odd decimal numbers?

IV. Counting in the binary system

Remember in decimal we count from the smallest single-digit number: 0, 1, 2, 3, …, 8, 9, - what next? We exhausted all numbers in the order of magnitude of **100**, so we have to go up to the next order of magnitude **101**: (1,2,...8, 9, ) 🡪 10, then 11, 12, .., 98, 99 – again, we ran out of the numbers in the order of magnitude of **101**, so we go up to the next order of magnitude **102**: (98, 99, ) 🡪 100, then 101, … 998, 999 🡪 go up to the next order of magnitude **103**: 1000, 1001, …

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order of | Thousands | Hundreds | Tens | Ones |
| Magnitude | 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |
| **100** |  |  |  | 0, 1, ..9, done! |
| **101**(10, 11,…, 98,99) |  |  | 11…9 | 01…9 |
| **102**(100,101,…998,999) |  | 11…9 | 00…9 | 01…9 |
| **103**(1000,1001,…9998,9999) | 11…9 | 00…9 | 00…9 | 01…9 |

Similarly, in binary:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order of Magnitude | 8 | 4 | 2 | 1 |
| 1000 (**103**) | 100 (**102**) | 10 (**101**) | 1 (**100**) |
| **100** |  |  |  | 0, 1 (done!) |
| **101**(10, 11) | – 11 in binary is like 99 in decimal: no more 2-digit numbers available!! |  | 11 | 01 |
| **102**(100,101,110,111) |  | 1111 | 0011 | 0101 |
| **103**(1000,1001,1010,1011,1100,1101,1110,1111) | 11111111 | 00001111 | 00110011 | 01010101 |