ASIAN INSTITUTE OF COMPUTER STUDIES

Bachelor of Science in Computer Science

Course Modules

CS316–Architecture & Organization

3rd Year – 1st Semester

MODULE 5:Binary Numbers

WEEK 4

Learning Outcomes:

After completing this course you are expected to demonstrate the following:
Explain data explain data representation and data storage, discuss bits, bytes and words, understand and convert between number systems: decimal, binary

I. Engage

INTRODUCTION

Data, in principal, is stored in 1s and 0s. Computers make use of data structures that describe a specific sequence for data to be organized. This allows the computer to understand how the different bits of data are related and to interpret the data correctly.

EXPLAINING BINARY NUMBERS

https://www.youtube.com/watch?v=LpuPe81bc2w

In order to communicate data and instructions, data such as numbers, letters, characters, special symbols, sounds/phonics, and images are converted into computer-readable form (binary). Once the processing of this data is complete, it is then converted into a format that people can understand. The processed data becomes meaningful information. The information becomes knowledge when it is understood and used by people for different purposes.

II. Explain

Bits and bytes

Have you ever heard someone saying that a movie is 700 megabytes (MB) big? Basically, what they are saying is that, the movie consists of 700 million bytes of data.

Words

A word is a fixed-sized piece of data handled as a unit by the instruction set or the hardware of the processor. The number of bits in a word (the word size, word width, or word length) is an important characteristic of any specific processor design or computer architecture.

DATA REPRESENTATION

Data representation refers to the form in which data is stored, processed and transmitted. Digital devices store data using the binary number system.

1 B	8 b	b = bit
1 KB	1024 B	B = byte
1 MB	1024 KB	KB = Kilobyte
1 GB	1024 MB	MB = Megabyte
1 TB	1024 GB	GB = Gigabyte
1 PB	1024 TB	TB = Terabyte
		PB = Petabyte

BITS AND BYTES

You can combine several bits (0s and 1s). If you combine 8 bits together, it is called a ''byte''. A byte can represent any number between 0 (represented in bits as 00000000) and 255 (represented in bits as 11111111).

HOW COMPUTERS WORK: ***https://qrs.ly/57ab1zm***

DECIMAL and BINARY

DECIMAL NUMBERING SYSTEM (BASE 10)

The numbering system we use is called the decimal system because the prefix 'deci' means 10, and there are 10 numbers in the decimal system: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

In the decimal system, each time the value of a number increases by 1, you increase the size of the digit by 1. This works well until you reach the largest digit possible that is 9. Once you are at 9, there are no larger digits left, so the next time you increase the value of your number by 1, you need to add a new digit to your number. The value of this digit increases from 0 to 1, and you restart the first digit at 0:

…
07
08
09
10

When you reach the number 19 and increase the value by 1, the second digit goes up again, and you start with 20. This means that the value of any digit in the second position is 10, with

20 equal to 2 × 10. As you continue counting, the value of your second digit eventually reaches 9 (for example in 90), and when you need to increase it again, (for example when adding 1 to 99) you find that impossible, so you reset both 9's to 0 and add a third digit, that is 100.

The place value of the third digit is thus always a 100, or 10 × 10 or 10^2. If you need to know the value of the number 742, you can calculate it as follows:

7	4	2
$7 \times 10^2 = 700$	$4 \times 10^1 = 40$	$2 \times 10^0 = 2$
		= 700 + 40 + 2 = 742

As your number increases in value, the value of each digit is 10 times larger than the values of the previous digit. The fourth digit is thus worth 1 000 (or 10^3), the fifth digit is worth 10 000 (or 10^4), and so forth.

Take note

Any number raised to the power of zero, for example, 10^0, equals 1. It does not matter how big or small the number is. So even $1\,000^0 = 1$.

So, to calculate the value of an eight digit number (for example, 51 240 887), you do the following:

5	1	2	4	0	8	8	7
$5 \times 10^7 =$ 50 000 000	$1 \times 10^6 =$ 1 000 000	$2 \times 10^5 =$ 200 000	$4 \times 10^4 =$ 40 000	$0 \times 10^3 =$ 0	$8 \times 10^2 =$ 800	$8 \times 10^1 =$ 80	$7 \times 10^0 =$ 7
							= 50 000 000 + 1 000 000 + 200 000 + 40 000 + 0 + 800 + 80 + 7 = 51 240 887

Once you understand that this is how counting works when you have ten unique numbers available, counting in any other numbering system is easy to grasp.

BINARY NUMBERING SYSTEM (BASE 2)

The prefix bi- means two (as in bicycle, biplane, or bilingual), so the binary numbering system is a numbering system in which there are only two unique values: 0 and 1. To count in binary, you use the same logic you use to count in the decimal system. You increase the value of a digit until it reaches the largest digit possible (1), then add a new digit and restart the previous digit.

To see how this works, try to count from 0 to 16 in binary.

Table 5.1: Counting in binary

DECIMAL	BINARY	DESCRIPTION
0	0_2	Start with the lowest possible value. Note the subscript 2 to show we are working in base 2
1	1_2	You increase the value of the digit by 1.
2	10_2	Since you cannot increase the value of the first digit any more, you reset it to 0 and increase the value of the second digit.
3	11_2	You increase the value of the first digit by 1 again.
4	100_2	Since you cannot increase the value of the first digit any more, you try to increase the value of the second digit, but this is also impossible, so you reset both values and add a third digit.
5	101_2	You increase the value of the first digit by 1.
6	110_2	Since you cannot increase the value of the first digit, you reset it and increase the value of the second digit.
7	111_2	You increase the value of the first digit by 1.
8	1000_2	Since you cannot increase the value of any digits, you reset them all and add a fourth digit.
9	1001_2	You increase the value of the first digit by 1.
10	1010_2	You reset the first digit and increase the value of the second digit.
11	1011_2	You increase the value of the first digit by 1.
12	1100_2	You reset the first and second digit and increase the value of the third digit.
13	1101_2	You increase the value of the first digit by 1.
14	1110_2	You reset the first digit and increase the value of the second digit.
15	1111_2	You increase the value of the first digit by 1.
16	10000_2	Since you cannot increase the value of any of the digits, you reset all of them and add a fifth digit.

CONVERTING FROM BINARY TO DECIMAL

How can you calculate the decimal (Base 10) value of an existing binary number? Although one could count to any number in this way, it is not very efficient and would take a long time with large binary numbers. A better way is to use the same technique you used with the decimal system and find out what the place value of each digit is before adding them all together.

To see how we do this, let us look at what the value of each of the digits is on their own.

Table 1: *The place value of binary digits*

BINARY	PLACE VALUE	DESCRIPTION
000001_2	$1 \times 2^0 = 1$	The place value of the first digit is 1.
000010_2	$1 \times 2^1 = 2$	The place value of the second digit is 2.
000100_2	$1 \times 2^2 = 4$	The place value of the third digit is 4.
001000_2	$1 \times 2^3 = 8$	The place value of the fourth digit is 8.
010000_2	$1 \times 2^4 = 16$	The place value of the fifth digit is 16
100000_2	?	Based on this pattern, what is the place value of the sixth digit?

In the **decimal numbering system**, each digit can hold **ten unique values** and each new digit has a place value that is ten times as large as the previous digit. In the **binary system**, each digit can only hold **two unique values**, so each new digit has a place value that is twice as large as the previous digit!

Just as with the decimal numbering system, you can calculate the value of a binary number by calculating the value of the individual digits and adding them together. Let's work through the following examples to ensure you understand how to do this.

Convert the binary number 00101101^2 to a decimal number.
To convert this binary number to a decimal number, you can use the following steps:

- Step 1: Start by calculating the place value of each position of the binary number.
- Step 2: Add the place values of the digits that are 1 together.
- Step 3: Ignore the place value of the digits that are 0.

Did you know

Remember that multiplying any number with 0 is 0.

You can lay your working out in this manner:

Binary number	0	0	1	0	1	1	0	1
Place value	128	64	32	16	8	4	2	1
Computed value	0	0	32	0	8	4	0	1
Value base 10								$= 32 + 8 + 4 + 1 = 45$

The binary number 00101101_2 is therefore 0 + 0 + 32 + 0 + 8 + 4 + 0 + 1 = 45.

Example 2

Convert the binary number 11000011_2 to a decimal number.

Using the same technique as in the previous example:

- Step 1: Start by finding the place value of each digit.
- Step 2: Add those place values together for the digits equal to 1.

- Step 3: Ignore the place value of the digits that are 0.

Binary number	1	1	0	0	0	0	1	1
Place value	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Computed values	1×128	1×64	0×32	0×16	0×8	0×4	1×2	1×1
Decimal value								$= 128 + 64 + 2 + 1 = 195$

The binary number 11000011_2 is therefore equal to 195.

CONVERTING FROM DECIMAL TO BINARY

To convert a decimal number into binary, you can use the following steps:
Step 1: Find the largest power of 2 that is smaller than or equal to the decimal number.
Step 2: Divide the decimal number by this power.
Step 3: Write down the answer of the division (1) underneath the power.
Step 4: Rewrite the decimal number in terms of the division and its remainder.
Step 5: Repeat this process with the remainder until there is no remainder left.
Step 6: Write down 0 under all the powers that were not used.

For example, to convert 37 into binary, start by writing down the factors of 2:

							$37 = ?$
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

1. Find the largest power of 2 that is smaller than or equal to 37.
2. The answer is 32, since $37 \div 32 = 1$ with a remainder of 5.
3. Write the number 1 under 32 and rewrite 37 as $1 \times 32 + 5$:

							$37 = 1 \times 32 + 5 = ?$
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
		1					

1. Find the largest power of 2 that is smaller than or equal to 5.
2. The answer is 4, where $5 \div 4 = 1$ with a remainder of 1.

							$37 = 1 \times 32 + 1 \times 4 + 1 \times 1 = ?$
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
		1			1		

3. This means you can write a 1 under the number 4, and rewrite 5 as $1 \times 4 + 1$:

The final remainder is 1, and the power of 2 smaller than or equal to 1 is also 1.

Since 1 ÷ 1 = 1 with no remainder, you can write the number 1 under the 1 in the table.

You should also add 0 under all the powers of 2 you did not use.

The 1s and 0s you have written in your table gives you your binary number:

							$37 = 1 \times 32 + 1 \times 4 + 1 \times 1 = 00100101_2$
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	1	0	0	1	0	1

Work through the following examples to make sure you understand how to convert a number from decimal to binary.

A. Evaluate

I. Convert the following binary numbers to decimal numbers:

a. 00001001_2
b. 10001011_2
c. 01001110_2
d. 10001000_2
d. 00010001_2
f. 11101111_2
g. 01010100_2
h. 10011001_2
i. 10011111_2
j. 11100000_2

II. Convert the following decimal numbers to binary numbers.

a. 155_{10}
b. 10_{10}
c. 98_{10}
d. 25_{10}
e. 200_{10}
f. 255_{10}
g. 75_{10}
h. 101_{10}
i. 175_{10}
j. 99_{10}

Contact Information of the Facilitator

Name : *Sheila B. Hernandez*
MS Teams Account (email) : *sheila.hernandez@gmail.com*
SmartPhone Number : *09564019225*

1. Binary Number: https://www.siyavula.com/read/it/grade-10-it/data-representation-storage-and-social-implications/02-data-representation-storage-and-social-implications?id=sec2-2#sec2-2
2. en.wikipedia.org/wiki/Word_(computer_architecture)
3. **https://www.youtube.com/watch?v=LpuPe81bc2w**
4. *How Computer Works*:***https://qrs.ly/57ab1zm***