DOI: 10.14421/ijid.2020.09102 IJID (International Journal on Informatics for Development), e-ISSN: 2549-7448
Vol. 9, No. 1, 2020, Pp. 9-14

Bit Manipulation: Conditional Statement using Bit-wise operators with C++

Rahmawati Nafi'ah[1], Wakhid Kurniawan[2], Johan Setiawan[3], Khoirul Umam[4]
[1, 2, 3, 4]Student of Department of Informatics
Informatics Graduate Program, Faculty of Science and Technology
UIN Sunan Kalijaga Yogyakarta
[2]PT. Cobra Dental Indonesia
[3]Solusi Pembayaran Elektronik
[4]Coding Collective
[1]19206050011@Student.uin-suka.ac.id, [2]19206050003@Student.uin-suka.ac.id, [3]19206050010@Student.uin-suka.ac.id,
[4]19206050015@Student.uin-suka.ac.id

Article History
Received May 19th, 2020
Revised June 23rd, 2020
Accepted July 21st, 2020
Published July 21st, 2020

***Abstract*—All of information that manipulated by a computer is represented in the form of bits, so in the programming language it is necessary to understand bitwise operations at the first. This paper aims to create a concept of making Conditional Statements with Bitwise operators in C ++. By doing so, we hope that people is easy to understand the operation behind conditional statements. A conditional operator is also known as a ternary operator. It takes three operands. A conditional operator is closely related with if else statement. The method used is a literature study studying the bit manipulation algorithm in the C ++ language. The results obtained are a function using bitwise operations in C ++ that implement conditional statements.**

Keywords-bit-manipulation; conditional_statement; bitwise_operation; c++; ternary_operator.

1 INTRODUCTION

All of information that manipulated by computer is represented by bit. Some information with the word yes and no, can be replaced with 1 and 0. Logic in a computer is represented by a two-state element, namely an off state (no current) and an on state (there is a current) [1]. The bit is like a switch that can be switch On or Off. If it is turned off it will be like the computer reading the value 0 and if it is turned on it will read the valuo of 1 [2]. Bit also can be described as true or false, live or dead, yes or no, etc.

Modern computers store and process information represented as 2-valued signals. It can be easily represented, stored, and sent on noisy and inaccurate cables. Electronic circuits for storing and counting 2-valued signals are so simple and reliable, so they can be integrated into billions of data on a single silicon chip. The binary system only recognizes code 1 (high) and 0 (low) to describe a machine condition, so it is very easy to process it according to its designation.

Most text characters use the ASCII standard which each character is represented by a unique byte-sized integer value. All of information inside the system such as disk files, programs, users data where saved in memory, and data were transferred over the network also represented by many bits [3]. In a program with C language, the programmer creates a source program (or source file) with an editor and saves it in the form of a text file. The source program is a sequence of bits, each with a value of 0 or 1, organized in 8-bit chunks called bytes. Each byte represents some text character in the program [4].

Programmer needs to understand the representation of numbers from machines, because they are not the same as integers and real numbers. The source code of program can be read and understood by humans in the form of C ++ language. In order for the machine to understand it, the source program is translated into low-level machine language sequences [5].

The number system used by humans everyday is decimal, whereas computers tend to be easier to process binary number systems [6]. Some literature about bits mentions that bits refer to "binary digits". Mugivane said that a bit is a binary digit, the smallest unit of information that can be stores either as 1 or 0 [7][8]. Bit is short for "Binary Digit", which means binary digit. Binary digits are the smallest units in digital computing. A computer does not use decimal numbers to store data. All computer data is stored in binary numbers. Only 2 different values can be expressed as one bit, either 0 or 1. In digital telecommunications too, all voltage levels are converted to binary data. The term "binary digit" or "bit" was introduced by John Tukey in 1947, when he was working as a scientist at Bell Laboratories [9].

According to Laudon, a binary digit representing the smallest unit of data in a computer system. It can only have one of two states, representing 0 or 1 [2]. Binary numbers are two radix (base) numbers, the lowest step is 0 and the highest number is 1. Every time you calculate by adding one unit, you will get: 0, 1, 10, 11, 100, 101, 110, 111, 1000 etc [1].

Meanwhile, the term byte is a grouping of binary digits (0 or 1) which represent an information [7]. Laoudon said byte is a string of bits, usually eight, used to store one number or character in a computer system [2]. One Byte is 8 bit units combined into one. Therefore, Byte is a unit of information that is larger than bits. The first term 'byte' was coined and used by Dr. Werner Buccholz in 1956, when he was working as a scientist at IBM [9].

The decimal number system is based on 10 numbers (0 to 9), the binary number system has only 2 numbers namely 0 and 1. All data processed by a computer has the form of numbers 0 and 1. In digital communication, these two values represent voltage level. One application, binary value 0 is used to express the absence of voltage, and binary value 1 for constant positive voltage.

The big letter ‘B’ represents bytes, while the small letter ‘b’ denotes bits. So uppercase letters mean larger bytes or units, while lowercase letters mean smaller bits or units. Usually transfer speed (bandwidth) is expressed in terms of bits, while data storage capacity is usually expressed in terms of bytes [9]. In personal computers, data is stored and processed in 8-bit units called bytes. In ASCII code, each alphanumeric character is represented by a unique sequence of 7 bits. Although bits are used to measure digital transmission speed (bit rate), the capacity of storage (disk, files, databases, etc.) is measured in bytes [10][11][12].

Their basic types also called Intrinsic Types [13]. The first five types are integers of non-descending length. For example, the exact int length of each type depends on the implementation; for example int can be 16, 32 or 64 bits. All of these types can be qualified as signed or not signed. The first has no effect on integer numbers (except characters) because the numbers are signed by default. When declaring an integer type as unsigned, there will not be a negative value but twice as many positive values (plus one when we consider zero as positive or negative). The char type can be used in two ways: for letters and numbers that are rather short. Char almost always has a length of 8 bits.

Logic values are represented as BOOLs. Boolean variables can store true and false. The length property that does not decrease applies in the same way as a floating-point number: float is shorter than or equal to double length, which in turn is shorter than or equal in length to double length. Typical sizes are 32 bits for float, 64 bits for double, and 80 bits for double length [13].

The bitwise operations can be applied for integer data type only. The C ++ programming language is very efficient to manipulating bits. Work on byte, or data type like char, short, int, long, float, double, long double, pointer [14][11]. The operations with bits are used in data compression to reduce space, encrypt the data for security problems. This operation is faster and closer to the system and optimizes the effectiveness of the program to a better level [15].

Bit manipulation is the act of manipulating at the bit level (0 and 1). C++ programming language is one programming language that provides bit manipulation operators. According e-

book Computer systems : a Programmer's Perspective, one useful feature of C++ is that it supports bit-wise Boolean operations [4]. The symbols have used for the Boolean operations are exactly those used by C++: | for or, & for and, ~ for Not, and ^ for Exclusive-Or. These can be applied to any "integral" data type, that is, one declared as type char or int, with or without qualifiers such as short, long, long long, or unsigned [15].

The & (bitwise AND) in C or C++ takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1. Table 1 shows this.

Table 1 Bitwise AND

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

The | (bitwise OR) in C or C++ takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 if any of the two bits is 1. Table 2 shows this.

Table 2 Bitwise OR

A	B	A | B
0	0	0
0	1	1
1	0	1
1	1	1

The ~ (bitwise NOT) in C or C++ takes one number and inverts all bits of it . We can see this in Table 3.

Table 3 Bitwise NOT

A	~A
0	1
1	0

The ^ (bitwise XOR) in C or C++ takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different [16]. Table 4 shows this operation.

Table 4 Bitwise XOR

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

C++ also provides a set of logical operators ||, &&, and !, which correspond to the Or, And, and Not operations of logic. These can easily be confused with the bit-level operations, but their function is quite different. The logical operations treat any nonzero argument as representing True and argument 0 as representing False. They return either 1 or 0, indicating a result of either True or False, respectively. C++ also provides a set of shift operations for shifting bit patterns to the left and to the right [4]. Operator << shows left shift and operator >> shows right shift [17].

The << (left shift) in C or C++ takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift. The >> (right shift) in C or C++ takes two numbers, right shifts the bits of the first operand, the second operand decides the number of places to shift. The right most bits are drop and new bits are inserted in from left. On the bases of which bit to be inserted from the left, right shift is divided into two type, logical right shift and arithmetic right shift [18][17].

Table 5 illustrates a set of functions that manipulate and test a set of bits. This function is a function obtained from the assignment of Computer Systems and Organization course. The functions that included in Bit-Level Manipulation are bitXor, allOddBits, isAsciiDigit, conditional, logicalNeg. Each function has a different description, rating and max operator. The "description" provides an overview of the functions that must be created. "Ranking" shows the level of difficulty (number of points). "Max ops" gives the maximum number of operators that you can use to implement this function.The function is written from the lowest rating of 1 to the highest rating of 4. Rating 1 is given to the bitXor function, rating 2 on the allOddBits function, rating 3 on the isAsciiDigit and conditional functions, and rating 4 on the logicalNeg function [19].

Table 5 Bit-Level Manipulation Functions

Name	Description	Rating	Max Ops
bitXor(x,y)	x ^ y using only & and ~	1	14
allOddBits(x)	Are all odd-numbered bits set to 1?	2	12
isAsciiDigit(x)	Is x an ASCII digit?	3	15
conditional(x, y, z)	Same as C's "x ? y : z"	3	16
logicalNeg(x)	Compute !x without using ! operator	4	12

This paper focuses on the conditional function. In the function there are three variables, namely *x*, *y* and *z*. The description of this function is the same as C's "*x* ? *y* : *z*". The "Rating" for the conditional function is three. The "Max ops" for the conditional function is sixteen. A C programming conditional operator is also known as a ternary operator. It takes three operands. The conditional operator is closely related with `if..else` statement. Syntax of C programming conditional operator is (*condition*) ? *expression1* : *expression2*. If the

condition is true then *expression1* is executed else *expression2* is executed [20].

Our aim to conduct this research is to implement the conditional operation without using the conditional operator. Instead, we should manipulate bits by applying several types of bit operators to perform the conditional operation. Thus, at the end we will understand more bit manipulations.

In some cases, bit manipulation can avoid or reduce the need to repeat data structures and can provide a lot of folding speed, because bit manipulation is processed in parallel, but the code can be more difficult to write and maintain. Hopefully, this article will make us more accustomed to and understand bit-level representation and manipulation.

2 METHOD

In this study, we use a research approach in the form of literature study. Library studies relating to theoretical studies and other references relating to values, culture and norms that develop in the social situation under study, besides library studies are very important in conducting research, this is because research will not be separated from scientific literature [21]. Some references used are from several other researchers. The use of some of these references helps in deepening research. mainly concerned with references about manipulation of bits, bytes and bits.

3 RESULT AND DISCUSSION

The implementation of bitwise operators has been described in the journal Application of Bitwise Operators in C [18]. In that journal, bitwise applications are implemented as a modulus operator and as Boolean flags and bit masks. This research will explain the application of bitwise operators on the conditional function. This operation is hard to understand for some people so this this research explains it in the understandable form.

3.1 Analysis

The program code is in the bits.c file provided by the lecturer in Computer Systems and Organization. In making this function, there are some rules that must be fulfilled, such as expressions in functions that can only use integers 0 to 255 (not allowed to use large constants like 0xffffffff), function arguments and local variables (no global variables), unary integer operations (! ~), binary integer operations (& ^ | + << >>). Programs are strictly prohibited from using control constructs (if, do, while, for, switch, etc.), defining or using macros, defining additional functions into files, calling other functions, using other operations (&&, ||, -, or ?:), using all forms of casting, using data types other than Integer (arrays, structs, or unions).

More details about the behavior of conditional functions are explained in comments on bits.c. The bits.c file gives an example of using a condition function, for example the result of a conditional function (x, y, z) where x = 2, y = 4 and z = 5 is 4. The operator for this function is ! , ~, &, ^, | , +, <<, >> . The "Rating" ranks the difficulty level for this function is three. The "Max ops" gives the maximum number of operators that you can use to implement this function is sixteen.

In this program, the variables are local, not global. Local variables are variables that can only be used where the variable is declared in a function scope. Global variables have the ability to be recognized by all existing program code whether in class or in the main program (main program). It is because the declaration of global variables is done outside the class and outside the main program, but you should reduce the use of global variables because there is a possibility that this variable will modified by instructions that use globally declared variables. Each function has a different number of variables so that they should be declared locally so that they don't interfere when modified.

3.2 Result

Based on the analysis, the formulas that were implemented in C ++ and obtained in the form of functions can be seen in Figure 1. The ternary operator *x* ? *y* : *z* will return *y* if x is true and *z* otherwise. In order to map an integer value to Boolean, 0 will be converted to false while any other values will be converted to true, then create an algorithm to map *x* to a new domain. Figure 1 depicts this function.

```
4  /*
5   * conditional - same as x ? y : z
6   *   Example: conditional(2,4,5) = 4
7   *   Legal ops: ! ~ & ^ | + << >>
8   *   Max ops: 16
9   *   Rating: 3
10  */
11 int conditionals(int x, int y, int z) {
12   x = ((!x) << 31) >> 31;
13   return (~x & y) | (x & z);
14 }
```

Figure 1. Conditional Function.

Regarding operators, the program has fulfilled the requirements because it uses a legal operator (! << >> ~ & |) and does not exceed the maximum amount (7 times). Figure 1 explains that there are three variables with integer types namely *x*, *y* and *z*. It works by turning variable *x* into a mask: change all of its bit into 1 if *x* is false, otherwise change all the bits into 0 if *x* is true. The next operation is either negate *x* and apply AND operation to negated *x* and *y* or apply AND operator to *x* and *z*. At the end of this function, one of those operations' result will be returned.

In bits.c functions are used in the programming structure not in the procedure. In the procedure there are void keywords and in the body there is no return value. Whereas in function there is a data type followed by the name of the function and in the

body part there is a return value. Function writing begins with a data type that is `int`, which means integer followed by the name of the function that is conditional and parentheses, which contain a list of parameters. There are three parameters in the function and all three use the `int` data type. This function will get the values x, y and z from the input in the main program. In the body section, there is a return value that will later be displayed in the main program.

Figure 2 explains the program code for conditional function calls. In C ++ programming, the program code `cout` or character-out to display output to the screen. After << symbol the programmer can write the text that will be displayed to the screen, followed by quotation marks, text, quotation marks and at the end of the command there is a semicolon (;) A sign `\n` is a symbol to create a new line. An `int` variable is a variable declaration with an integer data type.

```
17  int main(){
18      cout << "========================================\n";
19      cout << "               conditional              \n";
20      cout << "========================================\n";
21          int xa;
22          int xb;
23          int xc;
24
25      cout << "Input value x : "; cin >> xa;
26      cout << "Input value y : "; cin >> xb;
27      cout << "Input value z : "; cin >> xc;
28      cout << "\nResult  : ";
29      cout << conditionals(xa,xb,xc);
30      cout << "\n========================================\n";
```

Figure 2. Main Script.

In this program, there are three variables, namely *xa*, *xb* and *xc*. `cin` command or character-input is a function to take input from the keyboard. >> symbol is followed by a storage location variable, and a screen will appear for inputting data through the keyboard. In the program the value *x* will be represented by *xa* variable, the value of *y* is represented by *xb* variable and the value of *z* is represented by *xc* variable. After inputting of all these numbers, the program will send data to the conditional function with *xa*, *xb* and *xc* parameters.

3.3 Discussion

In Figure 3 you can see the results of the conditional function program. After running the program will bring up the title, then followed the writing enter the value x : . The user will input the value, for example 2. Continued with the display Enter the value of y, then the user enters the value of *y* that is 4. input will appear last Enter the value of z and the user enters the value 5. After completing inputting, the screen will appear writing The result is: followed the conditional result is 4 based on the input provided.

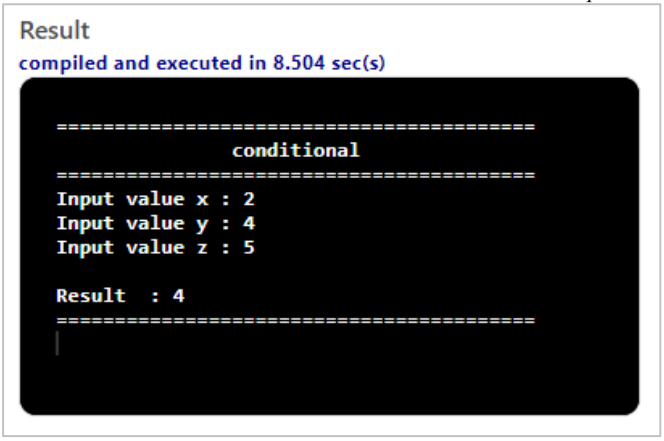

Figure 3. Result of programs.

4 CONCLUSION

To represent bits in C ++, the first must understand bitwise operations, as a basis for constructing conditional statement structures and algorithms if x is false then 1, if x is correct then 0. Based on the discussion in the previous session it produces an equation if x is not wrong, don't omit the bit y, if x is not correct, don't lose the z bit. From the concept or formula produced a function that can represent conditional statements in C ++.

In this article, we hope that there will be articles that discuss the more fundamental methods with more mature optimization methods, and also develop articles that discuss comparative studies of several languages used by compiler / interpreter types. It is hoped that readers can see a very useful thing in the development of the system so that the non-functionality of the application built can be achieved indirectly.

ACKNOWLEDGMENT

This paper is one of the assignment from Computer Systems and Organization course, the Department of Informatics Graduate Program, Faculty of Science and Technology UIN Sunan Kalijaga Yogyakarta.

REFERENCES

[1] F. Gulo, "Aplikasi Pembelajaran Hidrografi Menggunakan Metode Computer Assisted Instruction (CAI)," *J. Ris. Komput.*, vol. Vol. 3 No., no. Desember, pp. 34–37, 2018.

[2] Laudon, *Management Information Systems Thirteenth Edition Global Edition*. 2014.

[3] S. Janakiraman, K. Thenmozhi, J. B. B. Rayappan, and R. Amirtharajan, "Lightweight chaotic image encryption algorithm for real-time embedded system: Implementation and analysis on 32-bit microcontroller," *Microprocess. Microsyst.*, vol. 56, pp. 1–12, 2018.

[4] R. E. Bryant and D. R. O'Hallaron, *Computer Systems : a Programmer's Perspective -Second edition*. 2011.

[5] A. Sanchez-Stern, P. Panchekha, S. Lerner, and Z. Tatlock, "Finding root causes of floating point error," *ACM SIGPLAN Not.*, vol. 53, no. 4, pp. 256–269, 2018.

[6] K. Ismail and N. A. Rachman, "Dekoder Empat Bit Binary Code Decimal (Bcd) Untuk Dua Digit Seven Segment," *Semin. Nas. Telekomun. dan Inform.*, vol. Bandung, 2, no. Selisik, pp. 325–330, 2018.

[7] F. I. Mugivane, *Introduction To Computer*. 2014.

[8] M. Sari and A. Prasetyo, "Program Konversi Sistem Bilangan Desimal ke Biner dan Oktal Menggunakan Pemrograman C++ Berbasis Codeblock," Dec. 2018.

[9] "Pengertian & Perbedaan antara bit (b) dengan Byte (B) – JalaWave Connection," Nov-2016. [Online]. Available: https://www.jalawave.net.id/?p=413. [Accessed: 10-Jun-2020].

[10] E. Isbn, "Table of Contents Getting Help in an Integrated Development Environment," no. August 1997, 2003.

[11] R. E. Bryant and D. R. O. Hallaron, *Computer Systems. A Programmer's Perspective [3rd ed.]*. Boston: Pearson, 2016.

[12] C. Sinthanayothin and W. Bholsithi, "Dental application: The steps toward the implementation of the cephsmile plus services," *Int. J. Adv. Comput. Technol.*, vol. 3, no. 3, pp. 210–221, 2011.

[13] Peter Gottschling, *Discovering Modern C++. An Intensive Course for Scientists, Engineers,and Programmers*. Pearson Education, 2016.

[14] U. Sidi, M. Ben Abdellah, M. Fez, D. Chenouni, M. Berrada, and A. Tahiri, "Paper—A Serious Game for Learning C Programming Language Concepts Using Solo Taxonomy A Serious Game for Learning C Programming Language Concepts Using Solo Taxonomy Alaeeddine Yassine," *iJET*, pp. 110–127, 2017.

[15] P. Lindstrom, S. Lloyd, and J. Hittinger, "Universal coding of the reals: Alternatives to IEEE floating point," *ACM Int. Conf. Proceeding Ser.*, no. March, 2018.

[16] K. Yordzhev, "The Bitwise Operations Related to a Fast Sorting Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 9, 2013.

[17] I. Corporation, "[3]Intel ® 64 and IA-32 Architectures Software Developer ' s Manual Documentation Changes," *System*, vol. 3, no. 253665, 2011.

[18] R. Chandra, S. Rawat, and T. Jain, "Application of Bitwise Operators in C," *Int. J. Sci. Eng. Res.*, vol. 4, no. 11, pp. 1–4, 2013.

[19] A. Burud and P. Bhaskar, "Design and Implementation of FPGA Based 32 Bit Floating Point Processor for DSP Application," *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018*, no. ref 15, pp. 1–5, 2018.

[20] "C Programming Conditional Operator (?:) - Trytoprogram." [Online]. Available: http://www.trytoprogram.com/c-programming/c-conditional-operator/ . [Accessed: 10-Jun-2020].

[21] U. Suharsaputra, *Metode Penelitian Kuantitatif, Kualitatif, dan Tindakan*. Bandung: Alfabeta, 2012.