Developing Visualization for Learning Algorithm: Decimal to Binary Visualization

Minze Xiao [a,*], Yuan Chang [b]
[a,*]Xidian University, China, min.xiaoze@yahoo.com
[b]Xidian University, China, changyu12@yahoo.com

Abstract

This research presents an innovative approach to visualizing the decimal to binary conversion algorithm using Adobe Animate, a versatile tool known for its animation and graphics capabilities. The Digital Dif-ferential Analyzer is a fundamental algorithm in computer graphics, and our work focuses on creating an engaging and educational visualization that demystifies its step-by-step execution. The project begins by designing algorithm that outlines the key components and stages of the decimal to binary conversion algorithm. Leveraging Adobe Animate's user-friendly interface, we implement frame-by-frame animations that dynamically illustrate the algorithm's progression. Each frame depicts a specific step in the conver-sion process, with of binary animation of bits. The visualization incorporates interactive elements, allow-ing users to control parameters such a decimal input. By utilizing Adobe Animate's scripting capabilities, we enable real-time updates and dynamic conversion results, enhancing user engagement and under-standing. The inclusion of sliders, buttons, and dynamic text labels empowers users to actively manipu-late and observe the algorithm in action. Summarizing the key findings and emphasizing the importance of visualizing binary numbers in algorithm education. Encouraging educators to adopt these methods to make algorithmic concepts more accessible and enjoyable for students.

Keywords: algorithm, decimal to binary, image, visualization

I. Introduction

Images Learning is a "self-generating" need for life, which strives for itself, because from birth humans have the urge to live their lives, towards certain goals, consciously or unconsciously. Man's efforts to survive originate from himself, and as a social creature he must also defend his life. There are two main drives in humans, namely the drive to grow and develop, and the drive to defend themselves. Humans learn continuously to be able to achieve independence and adapt to various environmental changes. [1]

Computer education [2] is currently in great demand by students. The development of information technology and the presentation of communicative messages through multimedia is increasingly rapid, so multimedia-based learning is being held for several courses in higher education. Currently, the courses held by study programs are almost all passive because lecturers and instructors provide lecture or practical material while students only receive lecture material, theoretical, minimal practical skills, and a lack of individual learning which leads to the development of creativity such as visualization of binary numbers, and others.

This paper endeavors to facilitate students' understanding of algorithms by introducing a visual approach centered around binary numbers. Recognizing the foundational role binary numbers play in computer science, the paper explores the significance of their representation and manipulation in algorithmic processes. Through innovative visualizations such as binary charts and trees, students are guided in comprehending the conversion between decimal and binary, foster-

Received 9 October 2023, Revised 21 December 2023, Accepted 14 January 2024, Available online 5 February 2024, Version of Record 14 January 2024.

ing a more intuitive grasp of these fundamental concepts. The paper also delves into the practical applications of binary numbers in algorithms, accompanied by interactive learning tools to enhance engagement. By showcasing successful case studies and addressing potential challenges, this paper advocates for the integration of visualizations as a pedagogical strategy to make algorithmic education more accessible and enjoyable for students.

A. Learning

Learning is not something done to students, but rather something students themselves do. It is the direct result of how students interpret and respond to their experiences. Learning is "a process that leads to change, which occurs as a result of experience and increases the potential for improved performance and future learning" [3]. The change in the learner may happen at the level of knowledge, attitude or behavior. Competency of learners can be achieved through learning process.

Competency can be defined in two kind of meanings, one refers to the outputs, or results of training, and the other refers to the inputs, or underlying attributes, required of a person to achieve his/her performance. The meanings of competency show that they depend on the purpose for which it is used [4]. Approaching this is related to individual performance to the goals of the business. Several organizations use competencies as the integrated knowledge, skill, judgment, and attributes that people need to perform a job effectively. By having the competencies, organization team members can work and achieve their objectives more effectively [5]. Everyone can enhance his/her competency by learning in class or online training.

B. Visualization of Binary Number Conversion

Binary numbers are a number system with base 2 used for data processing on computers, namely numbers expressed as 0 or 1. Meanwhile decimal numbers are base 10 numbers which use the digits 0, 1, .. 9. Visualization of converting decimal numbers to binary is part of computer-assisted learning (Computer-Aided Learning). Interactive visualizations [6] can be created using Adobe Animate [7] where the user enters decimal numbers, then the computer converts them into binary numbers. Adobe Animate is not only software with professional standards for creating animations, but is equipped with ActionScript programming which is capable of creating animations and visualizations from mathematical and logical calculations.

Computer-assisted learning has been widely used. There is a lot of software for making visual aids to display subject matter easily. This presentation includes a linear presentation, namely students as passive listeners or viewers. Besides that, there are non-linear or interactive presentations where the audience can take part in learning by entering data, then the computer carries out the process and displays the results on the display screen. Interactive presentations can be equipped with visualizations so that the audience can understand them more easily. [5]

Numerical simulations use hundreds or even thousands of data to provide an overview of the results of an evaluation. The data is processed to obtain results in visual form. Visualization of graphs and processes for data sets from scientific, engineering, and medical data is usually called scientific visualization. Creating visualizations including creating graphs, charts, surface rendering, animation and multimedia can be done using computer software. [8]

Number system

Computers do not work with decimal numbers, but binary numbers. The decimal number system is a number with base 10, and each digit position has its own weight. The weights in decimal numbers consist of the weights of ones, tens, hundreds, and so on. The sum of all the digits multiplied by their respective weights is the value of the decimal number. For example, the rightmost digit has a weight of 100 (ones), the second digit from the right has a weight of 101 (tens), the third digit has a weight of 102 (hundreds), and so on. [2]

Examples of decimal numbers:
167 = 1 x 102 + 6 x 101 + 7 x 100 = 1 x 100 + 6 x 10 + 7 x 1
= 100 + 60 + 7

Binary numbers, also called base 2 numbers, have a weight equal to the power of the number 2. As can be seen in the following example, the weights are 20, 21, 22, and so on. The decimal equivalent of a binary number is equal to the sum of all the binary numbers multiplied by their respective weights.

Example of binary numbers

10011 = 1 x 24 + 0 x 23 + 0 x 22 + 1 x 21 + 1 x 20
= 16 + 0 + 0 + 2 + 1
= 19

Thus the binary number 10011 is equivalent to decimal 19. The binary number 10011 can be written in 1 byte (8 digits) as 00010011 as in Figure 1. The 0th bit has the value 1, the 1st bit has the value 1, the 2nd bit has value 0, and so on. Adding three 0 values to the 5th bit, 6th bit and 7th bit does not change the meaning of the binary number, as in Figure 1.

167 = 1 x 102 + 6 x 101 + 7 x 100 = 1 x 100 + 6 x 10 + 7 x 1
= 100 + 60 + 7

0	0	0	1	0	0	1	1
b7	b6	b5	b4	b3	b2	b1	b0

Figure 1 Arrangement of binary digits in one byte consisting of 8 digits

Convert decimal numbers to binary

To convert a decimal number to binary, you can use the double dabble method. This method uses repeated division of the converted decimal number, writing down the quotient and remainder each time. The example in Figure 2 shows how to convert the decimal number 14 to a binary number.

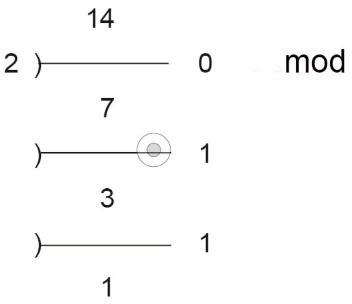


Figure 2 Converting decimal numbers to binary using double dabble

Once completed, the results are obtained by reading them from bottom to top. Thus the decimal number 14 is equivalent to the binary number 1110. The algorithm for converting a decimal number to binary can be written as follows:

1. Determine the decimal number.
2. Define an array to store the remainder of the division.
3. The decimal number is divided by 2, the remainder of the division is stored in the array as the 0th element.
4. The quotient is divided by 2, and the remainder of the division is stored in the array as the 1st element.
5. Repeat number 4 as long as the quotient is greater than 1.
6. The conversion result is obtained by reading the remainder of the division from the last sequence to the earlier sequence

II. Method

In number conversion using a programming language, usually someone enters a decimal number, then sees the results on the computer screen. Visualization is used to clarify understanding with interesting animations, making it easier to understand. The visualization used in this research is one of the visualizations of the decimal to binary number conversion algorithm. This visualization is very simple, namely giving the user the opportunity to enter a decimal number in the input box. By pressing a button, the computer will display the digits 0 and 1 which form a binary number. The number of digits is 8, so the binary number resulting from the conversion is a binary number with a length of 1 byte (8 bits), as in Figure 3. As an alternative, also write a binary number with a sufficient number of digits. For example, the decimal number 14 is converted into binary to become 00001110 or 1110.

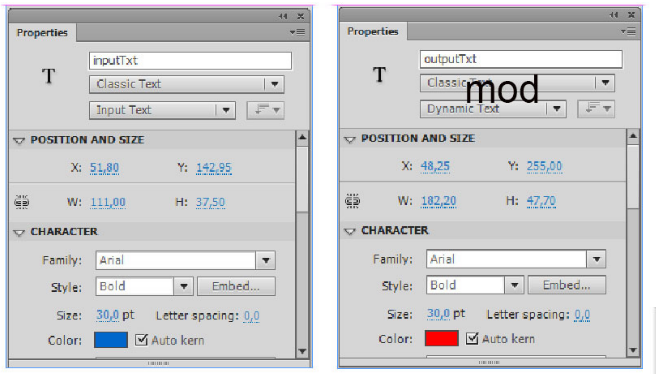
1. Enter the decimal number
2. Convert to binary
3. Animate bits 0 to bit 7
4. Show onversion results

Figure 3 Algorithm for the decimal to binary conversion process

III. Results and Discussion

Many decimal to binary number conversions have been made using programming languages such as Pascal, C, Visual Basic, Delphi, and others. To create a visualization of decimal numbers to binary, you can do the following:

1. Create a new Animate file by selecting ActionScript 3.0.
2. Create a text field for input using the Text Tool, and in Properties select Classic Text and Input Text, with the name inputTxt.
3. Create a text field for output as in number 2, but select Dynamic Text with the name outputTxt, as in Figure 4.



Properties
inputTxt
Classic Text
Input Text
POSITION AND SIZE
X: 51,80 Y: 142,95
W: 111,00 H: 37,50
CHARACTER
Family: Arial
Style: Bold Embed...
Size: 30,0 pt Letter spacing: 0,0
Color: 🟦 ☑ Auto kern

Properties
outputTxt
Classic Text
Dynamic Text
POSITION AND SIZE
X: 48,25 Y: 255,00
W: 182,20 H: 47,70
CHARACTER
Family: Arial
Style: Bold Embed...
Size: 30,0 pt Letter spacing: 0,0
Color: 🟥 ☑ Auto kern

Figure 4 Setting the input text (left) and output text (right) in the Properties Window

4. Create a button with the instance names enterBtn and resetBtn.
5. Create a new layer with the name «Action», as in Figure 5. In frame 1, create the following script:

```
import flash.text.*;

var binaryArray:Array = new Array();
var num:Number;
var binaryString:String;
var bString:String;
var bNumber:Number;
var len:int = 0;
inputTxt.type = TextFieldType.INPUT;
inputTxt.restrict = “0-9”;
addChild(inputTxt);
inputTxt.addEventListener(FocusEvent.FOCUS_IN, startFocus);
inputTxt.addEventListener(FocusEvent.FOCUS_OUT, endFocus);

function checkEnterKey(e:KeyboardEvent):void{
        if(inputTxt.text != ‘’ && e.keyCode == Keyboard.ENTER){
                num = Number(inputTxt.text);
                //trace(e.keyCode + “ : “ + tf.text);
                bString = convertDecimalToBinary(num);
                bNumber = Number(binaryString);
        }
}

function getTextInput(e:KeyboardEvent):void{
        num = Number(inputTxt.text);
        //trace(e.keyCode + “ : “ + tf.text);
        bString = convertDecimalToBinary(num);
        bNumber = Number(binaryString);
}

function convertDecimalToBinary(num:Number):String{
        if(num == 0)
                return String(“0”);

        binaryString = ‘’;   //start with empty string
        while(num > 0){
                num /= 2;
                if(Math.floor(num) == num)
                        binaryArray.push(0);
                else
                        binaryArray.push(1);

                trace(num);
                num = Math.floor(num);
        }

        for(var i:int = binaryArray.length - 1; i >= 0; i--){
                binaryString += binaryArray[i];
                // clear array to prepare for next input
                binaryArray.pop();
        }
        trace(binaryString);
        return binaryString;
}

function startFocus(e:FocusEvent):void{
        stage.addEventListener(KeyboardEvent.KEY_UP, getTextInput);
}

function endFocus(e:FocusEvent):void{
        stage.removeEventListener(KeyboardEvent.KEY_UP, getTextInput);
}
```

```
function readText(evt:Event):void {
        bString = convertDecimalToBinary(num);
        bNumber = Number(binaryString);
        outputTxt.text = bString;
}

function resetText(evt:Event):void {
        inputTxt.text = " ";
        outputTxt.text = " ";
}

enterBtn.addEventListener(MouseEvent.CLICK,
readText);
resetBtn.addEventListener(MouseEvent.CLICK,
resetText);
```

6. Select the Control menu > Test Movie to see the results, as can be seen in Figure 8.5.
7. Save the file.

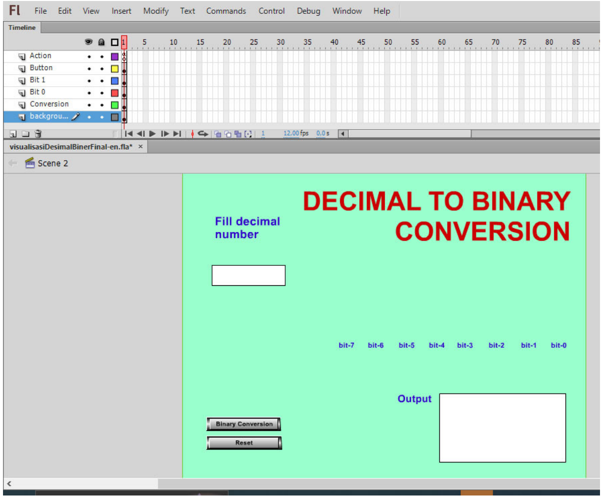
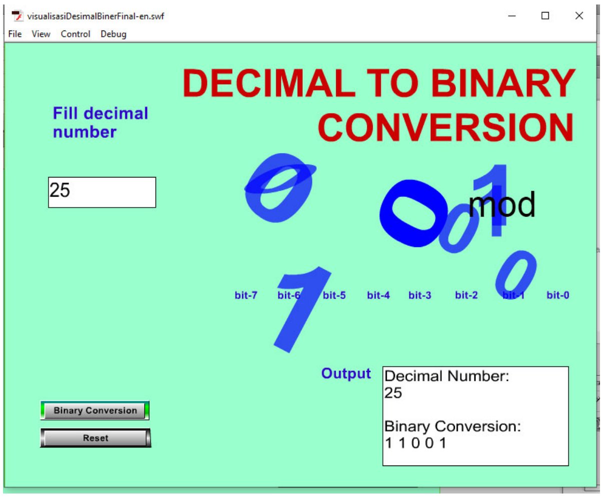


Figure 5 View on stage

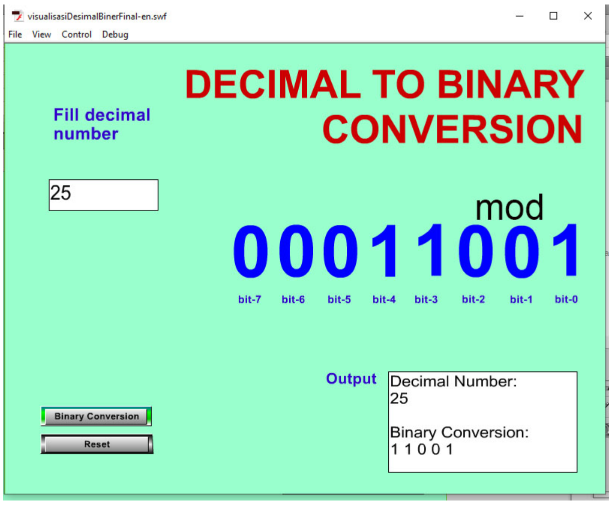


25

Decimal Number:
25

Binary Conversion:
1 1 0 0 1

Figure 6 Animation of bits in binary numbers when the movie is running



25

Decimal Number:
25

Binary Conversion:
1 1 0 0 1

Figure 7 Results of converting decimal numbers to binary

IV. Conclusion

This paper introduces a visualization of the decimal-to-binary algorithm, shedding light on essential graphic programming details crucial for student learning. The exploration extends to the thorough testing of visualization and interactivity, conducted by students at the university Department of Informatics. Leveraging Adobe Animate, a timeline-based authoring tool with object-oriented programming capabilities, the paper advocates its utility for developing scientific visualizations, further enriching the educational experience in algorithmic understanding. Summarizing the key findings and emphasizing the importance of visualizing binary numbers in algorithm education. Encouraging educators to adopt these methods to make algorithmic concepts more accessible and enjoyable for students.

References

[1] K. Mcgonigal, "From Learning Theory to Teaching Strategies," Newsl. Teach., vol. 14, no. 2, p. 5, 2005.

[2] H. L. Capron, Computers Tools for an Informatoon Age. Upper Saddle River, NJ, 2000.

[3] J. C. Sun, Y. J. Oh, H. Seli, and M. Jung, "Learning Behavior and Motivation of At-Risk Learning Class," J. At-risk Issues, vol. 20, no. 2, pp. 12–24, 2017.

[4] R. Sullivan and N. McIntosh, "The competency-based approach to training," Med. J. Indones., vol. 5, no. 2, pp. 95–98, 1996, doi: 10.13181/mji.v5i2.853.

[5] A. A. Rahman, “Assessment Practices for Competency Based Education and Training in Vocational College, Malaysia,” 2013, [Online]. Available: https://ac.els-cdn.com/S1877042814012889/1-s2.0-S1877042814012889-main.pdf?_tid=cece6267-6e88-4d3b-85a4-74f1ee5af78b&acdnat=1545535028_438ef5e574a25a372bff4f87225d03f.

[6] D. Hearn and P. Baker, Computer Graphics. Englewood Cliffs: Prentice Hall International, Inc., 2006.

[7] R. Chun, Adobe Animate CC Classroom in a Book. California: Adobe Press, 2019.