Computer Architecture

Assignment 2

Converting Numbers & Truth Tables

Bryan Yang

Table of Contents

[Part 1 – Number Systems 3](#_Toc111010843)

[Part 2 – Truth Tables and Logic Gates 5](#_Toc111010844)

[Part 3 – Reflection 6](#_Toc111010845)

[References: 7](#_Toc111010846)

## Part 1 – Number Systems

For the operations below, be sure to show your work! Do not simply put the numbers through a calculator or online tool and post the results. We should be able to follow your logic as you solved the problem.

1. Complete the following addition problem in hexadecimal: 32154AAAA + FEDCBA092. Show the answer in hexadecimal and in decimal.

32154AAAA16 = (3 x 168) + (2 x 167) + (1 x 166) + (5 x 165) + (4 x 164) + (10 x 163) + (10 x 162) + (10 x 161) + (10 x 160)

32154AAAA16 = 12884901888 + 536870912 + 16777216 + 5242880 + 262144 + 40960 + 2560 + 160 + 10

32154AAAA16 = 1344409873010

FEDCBA09216 = (15 x 168) + (14 x 167) + (13 x 166) + (12 x 165) + (11 x 164) + (10 x 163) + (0 x 162) + (9 x 161) + (2 x 160)

FEDCBA09216 = 64424509440 + 3758096384 + 218103808 + 12582912 + 720896 + 40960 + 144 + 2

FEDCBA09216 = 6841405454610

1344409873010 + 6841405454610 = 8185815327610

8185815327610 % 16 = F16 #reminder – last digit  
8185815327610 / 16 = 51161345710 #quotient  
8185815327610 / 16 = 51161345710 #quotient  
51161345710 % 16 = 116 #reminder  
51161345710 / 16 = 3197584110 #quotient  
3197584110 % 16 = 116 #reminder  
3197584110 / 16 = 199849010 #quotient  
199849010 % 16 = A16 #reminder  
199849010 / 16 = 12490510 #quotient  
12490510 % 16 = E16 #reminder  
12490510 / 16 = 780610 #quotient  
780610 % 16 = 716 #reminder  
780610 / 16 = 48710 #quotient  
48710 % 16 = E16 #reminder  
48710 / 16 = 110 #quotient  
110 % 16 = 116  #reminder – first digit  
110 / 16 = 010  #quotient  
8185815327610 = 1E7E9A11F16 #result

1. Convert the Decimal number 4048891811 to hexadecimal.

404889181110 % 16 = 316 #reminder – last digit

404889181110 / 16 = 25305573810 #quotient

25305573810 % 16 = A16 #reminder

25305573810 / 16 = 1581598310 #quotient

1581598310 % 16 = F16 #reminder

1581598310 / 16 = 98849810 #quotient

98849810 % 16 = 216 #reminder

98849810 / 16 = 6178110 #quotient

6178110 % 16 = 516 #reminder

6178110 / 16 = 386110 #quotient

386110 % 16 = 516 #reminder

386110 / 16 = 24110 #quotient

24110 % 16 = 116 #reminder

24110 / 16 = 1510 #quotient

1510 % 16 = F16  #reminder – first digit

1510 / 16 = 010  #quotient

404889181110 = F1552FA316 #result

1. Convert the Octal number 2114112 to Decimal.

21141128 = (2 x 86) + (1 x 85) + (1 x 84) + (4 x 83) + (1 x 82) + (1 x 81) + (2 x 80)   
21141128 = 524288 + 32768 + 4096 + 2048 + 64 + 8 + 2  
21141128 = 56327410

1. Expand the following table to include decimal numbers from 11 to 16, and expand the table to include hexadecimal numbers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Binary** | **Octal** | **Decimal** | **Hexadecimal** |
| 000 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 |
| 010 | 2 | 2 | 2 |
| 011 | 3 | 3 | 3 |
| 100 | 4 | 4 | 4 |
| 101 | 5 | 5 | 5 |
| 110 | 6 | 6 | 6 |
| 111 | 7 | 7 | 7 |
| 1000 | 10 | 8 | 8 |
| 1001 | 11 | 9 | 9 |
| 1010 | 12 | 10 | 10 |
| 1011 | 13 | 11 | A |
| 1100 | 14 | 12 | B |
| 1101 | 15 | 13 | C |
| 1110 | 16 | 14 | D |
| 1111 | 17 | 15 | E |
| 10000 | 20 | 16 | F |

## 

## Part 2 – Truth Tables and Logic Gates

State the Boolean Expression and truth table for the following logic gates:

* NAND

|  |  |  |
| --- | --- | --- |
| A | B | X |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

* NOR

|  |  |  |
| --- | --- | --- |
| A | B | X |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

* XOR

|  |  |  |
| --- | --- | --- |
| A | B | X |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

* NOT

|  |  |
| --- | --- |
| A | X |
| 0 | 1 |
| 1 | 0 |

* 3-input AND gate (inputs A, B, C, and output X)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | X |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## Part 3 – Reflection

Answer the following question, providing examples and citing relevant research. Your reflection should be between 500 and 750 words:

* Why is it important to study how to manipulate fixed-point numbers? Provide examples of these systems in use.

Although floating point numbers allows us to use much wider range of values than fixed points with the ability to represent very small numbers and very large numbers, it is important to study how to manipulate fixed-point numbers (Specbranch, 2018). Float and double are the most used number representations with signed and un-signed values.

To understand the importance of the fixed-point numbers, we need to discuss the definition and difference between fixed- and floating-point numbers. Fixed point numbers have fixed number of digits after the decimal point. Meaning, no matter how large or small the numbers are, there will always be same numbers of bits for each portion (Specbranch, 2018). That is why fixed-point arithmetic is done using integers where the decimal part is stored in a fixed amount bit (the number is multiplied by how many digits of decimal precision would be needed). In contrast, floating point numbers have a varying number of digits after the decimal point (Specbranch, 2018). Floating point numbers reserves a certain number of bits for digits (called the mantissa) and the certain number of bits tells us within which number the decimal place sits (called the exponent).

With the characteristics of fixed points created a few problems for its usage for every day use compared to flexible nature of floating-point number representations. First, successful execution of fixed-point arithmetic operations depends on constructing correct chains of operations that fit together in sequence. This often is very time consuming to tract the dynamic range of the values in the operation chain to avoid overflows and thereby unnecessary shift operations (Specbranch, 2018). Therefore, it often takes a lot of developers and engineers to work out the correct digit for each variable and add shifts when needed by using the mathematical computing software (Specbranch, 2018).

Since the founding of new number representations such as float and double, fixed-point numbers have been overhauled and narrowed downed to few niches and industries. As mentioned previously, fixed point computation uses integer side of computation processing unit (Specbranch, 2018). This allows the fixed point to take advantage of faster instructions and more available instruction ports. Therefore, financial industry often makes the use of certain form of decimal fixed-point numbers to represent money. The banking systems use 2 integers to present cents or small units of currencies and four decimal places to represent units (thousand, millions, billions, etc.). This use case allows exchanges to have fixed precision across the range of prices (Specbranch, 2018).

To summarize the use cases, fixed point can be found in places where resources are tightly constrained such as banking and hardware audio-processing industries. This also comes with some speed benefits on server-scale system and allow the user to do non-integer math on systems without floating point hardware (Specbranch, 2018). However, this benefit often cost a lot of men hours (engineers and developers). It is a great tool for certain instances when we want to compute as fast as possible and give up some dynamic range. There are still invaluable tool for modern society.

## References:

Meghalee (2020). *Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers*. Study.Com. https://study.com/academy/lesson/converting-floating-point-values-in-the-binary-numerical-system.html

Goodman, S. (2020). *Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers*. Study.Com. https://study.com/academy/lesson/converting-between-binary-decimal-octal-hexadecimal-numbers.html

Arizzi, R. (2020). *Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers*. Study.Com. https://study.com/academy/lesson/electric-gate-symbols-for-boolean-functions.html

Alcocer, Y. A. Y. (2020). *Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers*. Study.Com. https://study.com/academy/lesson/propositions-truth-values-and-truth-tables.html

*Fixed Point Arithmetic - Speculative Branches*. (2018). Specbranch. https://specbranch.com/posts/fixed-point/