

605.611 - Foundations of Computer Architecture

Assignment 01

Sabbir Ahmed

February 1, 2021

1. Convert the following from decimal to binary. Express numbers in 16 bits.

(a) 14

$$\begin{array}{r} 7 \\ \hline 2 \overline{)14} \\ \hline 14 \\ \hline 0 \end{array} \quad \begin{array}{r} 3 \\ \hline 2 \overline{)7} \\ \hline 6 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ \hline 2 \overline{)3} \\ \hline 2 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ \hline 2 \overline{)1} \\ \hline 1 \end{array}$$

$$\Rightarrow (1110)_2$$

(b) 97

$$\begin{array}{r} 48 \\ \hline 2 \overline{)97} \\ \hline 8 \\ \hline 17 \end{array} \quad \begin{array}{r} 24 \\ \hline 2 \overline{)48} \\ \hline 4 \\ \hline 08 \end{array} \quad \begin{array}{r} 12 \\ \hline 2 \overline{)24} \\ \hline 2 \\ \hline 04 \end{array} \quad \begin{array}{r} 6 \\ \hline 2 \overline{)12} \\ \hline 12 \\ \hline 0 \end{array} \quad \begin{array}{r} 3 \\ \hline 2 \overline{)6} \\ \hline 6 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ \hline 2 \overline{)3} \\ \hline 2 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ \hline 2 \overline{)1} \\ \hline 1 \end{array}$$

$$\Rightarrow (0110\ 0001)_2$$

(c) 947

$$\begin{array}{r}
 \begin{array}{r}
 \overline{473} \\
 2 \overline{)947} \\
 \underline{8} \\
 14 \\
 \underline{14} \\
 07 \\
 \underline{6} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 \overline{236} \\
 2 \overline{)473} \\
 \underline{4} \\
 07 \\
 \underline{6} \\
 13 \\
 \underline{12} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 \overline{118} \\
 2 \overline{)236} \\
 \underline{2} \\
 03 \\
 \underline{2} \\
 16 \\
 \underline{16} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 \overline{59} \\
 2 \overline{)118} \\
 \underline{10} \\
 18 \\
 \underline{18} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 \overline{29} \\
 2 \overline{)59} \\
 \underline{4} \\
 19 \\
 \underline{18} \\
 1
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{r}
 \overline{14} \\
 2 \overline{)29} \\
 \underline{2} \\
 09
 \end{array}
 \quad
 \begin{array}{r}
 \overline{7} \\
 2 \overline{)14} \\
 \underline{14} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 \overline{3} \\
 2 \overline{)7} \\
 \underline{6} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 \overline{1} \\
 2 \overline{)3} \\
 \underline{2} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 \overline{0} \\
 2 \overline{)1} \\
 \underline{0} \\
 1
 \end{array}
 \end{array}$$

$$\Rightarrow (0011 \ 1011 \ 0011)_2$$

(d) 32768 (What is special about this number that makes it easy to calculate?)

32768 is a power of 2 i.e. 2^{15} . This decimal integer is very easy to calculate since only the 15th bit is active and therefore $(32768)_{10} = (1000 \ 0000 \ 0000 \ 0000)_2$.

(e) 32767 (What is special about this number and the last one that makes it easy to calculate?)

32767 is 1 less than 32768 which is a power of 2 i.e. $2^{15} - 1$. This decimal integer is very easy to calculate since all the 15 bits are active and therefore $(32767)_{10} = (0111 \ 1111 \ 1111 \ 1111)_2$.

2. Convert the following numbers from binary to decimal

(a) 0000 0000 0000 1010

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 10$$

(b) 0000 0000 1001 1010

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 154$$

(c) 0000 0001 1011 1100

$$1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 444$$

3. Convert the following numbers from binary to hexadecimal

(a) 0000 0000 0000 1010

$$1010 = A$$

(b) 0000 0000 1001 1010

$$1001 = 9$$

$$1010 = A$$

$$= 9A$$

(c) 0000 0001 1011 1100

$$0001 = 1$$

$$1011 = B$$

$$1100 = C$$

$$= 1BC$$

4. Convert the following numbers from hex to binary.

(a) 0x000C

$$0 \cdot 16^3 + 0 \cdot 16^2 + 0 \cdot 16^1 + 12 \cdot 16^0 = 12$$

(b) 0x07AD

$$7 \cdot 16^3 + 10 \cdot 16^2 + 0 \cdot 16^1 + 13 \cdot 16^0 = 1965$$

(c) 0xA49F

$$10 \cdot 16^3 + 4 \cdot 16^2 + 9 \cdot 16^1 + F \cdot 16^0 = 42143$$

5. Convert the ASCII string “Java.2” to binary. You must include the character for the period.

Looking up the ASCII values of the string:

$$J = 74$$

$$a = 97$$

$$v = 118$$

$$. = 46$$

$$2 = 50$$

And converting the decimals to binary,

$$\begin{array}{r}
 37 \quad 18 \quad 9 \quad 4 \quad 2 \quad 1 \quad 0 \\
 \hline
 74 = 2 \overline{)74} \quad 2 \overline{)37} \quad 2 \overline{)18} \quad 2 \overline{)9} \quad 2 \overline{)4} \quad 2 \overline{)2} \quad 2 \overline{)1} \\
 \hline
 6 \quad 2 \quad 18 \quad 8 \quad 4 \quad 2 \\
 \hline
 14 \quad 17 \quad 0 \quad 1 \quad 0 \quad 0 \\
 \hline
 14 \quad 16 \\
 \hline
 0 \quad 1
 \end{array}$$

$$\Rightarrow (0100 \ 1010)_2$$

$$97 = (0110 \ 0001)_2 \text{ (from part 1B)}$$

$$\begin{array}{r}
 59 \quad 29 \quad 14 \quad 7 \quad 3 \quad 1 \quad 0 \\
 \hline
 118 = 2 \overline{)118} \quad 2 \overline{)59} \quad 2 \overline{)29} \quad 2 \overline{)14} \quad 2 \overline{)7} \quad 2 \overline{)3} \quad 2 \overline{)1} \\
 \hline
 10 \quad 4 \quad 2 \quad 14 \quad 6 \quad 2 \\
 \hline
 18 \quad 19 \quad 09 \quad 0 \quad 1 \quad 1 \\
 \hline
 18 \quad 18 \quad 8 \\
 \hline
 0 \quad 1 \quad 1
 \end{array}$$

$$\Rightarrow (0111 \ 0110)_2$$

$$\begin{array}{r}
 23 \quad 11 \quad 5 \quad 2 \quad 1 \quad 0 \\
 \hline
 46 = 2 \overline{)46} \quad 2 \overline{)23} \quad 2 \overline{)11} \quad 2 \overline{)5} \quad 2 \overline{)2} \quad 2 \overline{)1} \\
 \hline
 4 \quad 2 \quad 10 \quad 4 \quad 2 \\
 \hline
 06 \quad 03 \quad 1 \quad 1 \quad 0 \\
 \hline
 6 \quad 2 \\
 \hline
 0 \quad 1
 \end{array}$$

$$\Rightarrow (0010 \ 1110)_2$$

$$\begin{array}{r}
 25 \quad 12 \quad 6 \quad 3 \quad 1 \quad 0 \\
 \hline
 50 = 2 \overline{)50} \quad 2 \overline{)25} \quad 2 \overline{)12} \quad 2 \overline{)6} \quad 2 \overline{)3} \quad 2 \overline{)1} \\
 \hline
 4 \quad 2 \quad 12 \quad 6 \quad 2 \\
 \hline
 10 \quad 05 \quad 0 \quad 0 \quad 1 \\
 \hline
 10 \quad 4 \\
 \hline
 0 \quad 1 \quad 5
 \end{array}$$

$$\Rightarrow (0011 \ 0010)_2$$

Therefore, the ASCII string “Java.2” converted to binary is

(01001010 01100001 01110110 01100001 00101110 00110010)₂

6. Is the binary number 1001 1100 0111 1011 1101 1001 odd or even?

Since the binary number ends with a 1, it indicates that $2^0 = 1$, an odd integer, was added to powers of 2. Since adding an odd integer to an even integer makes it an odd integer, this binary number is odd.

7. What is the base 10 value of the following two’s complement numbers.

(a) 01001 0111

(b) 1101 1000

8. Add the following numbers in 4-bit two’s complement format. Specify if there is an overflow or not.

(a) 5+3

(b) 7+5

(c) 7+ (-5)

(d) (-7) + (-5)

(e) (-5) + (-3)

9. Using only the bitwise complement operation and addition, implement a function in the language of your choice to calculate the negative value of an int. The bitwise complement is the `~` operator in Java and C/C++. The program should be properly formatted and documented, including:

(a) A preamble statement that includes the name of the program, the purpose of the program, the author, and the data that it was written. The author attribute in Java can be documented using the `@author` Javadoc attribute.

- (b) Functions should include a purpose and any parameters and return values. The parameters and return values can be documented using the @param and @return Javadoc attributes.
 - (c) Proper style including variable naming, indentation, placement of brackets, etc.
 - (d) Students unfamiliar with proper style should refer to: <https://www.oracle.com/technetwork/java/150003.pdf>
10. Implement the following multiplication problems using only bit-shift operations and addition.
- (a) $7 * 4$
 - (b) $8 * 3$
 - (c) $9 * 13$
11. Implement the division of $13/4$ using bit shift operations. What happens to the remainder. What does this tell you about the efficient implementation integer division? Does this match what you know about integer division in Java or C/C++?
12. Can $14 / 3$ be implemented using simply bit shift operations? Why or why not?
13. Write a program in Java or C/C++ to convert a character to upper case, regardless of whether it started as upper or lower case. This is simply OR'ing the byte for the character 0x20. The bitwise OR operation in Java or C/C++ is "`|`". The same rules of program style apply to the program as for Problem 9.
14. Write a program in Java or C/C++ to convert a character to lower case, regardless of whether it started as upper or lower case. This is simply AND'ing the byte for the character 0xCF. The bitwise AND operation in Java
15. Write a program in Java or C/C++ to convert a numeric value in a String to a valid integer. You must use the following algorithm, you may not use functions like "parseInt" or any other Java or C/C++ (or any language) libraries. The same rules of program style apply to the program as for Problem 9. The algorithm for this operation is:

```
int toInt(char[] inputNumber) {  
    retval == 0;  
    foreach character c in the inputNumber {  
        retval = retval * 10;  
        retval = retval + c - '0';  
    }  
    return retval;  
}
```