|  |  |  |  |
| --- | --- | --- | --- |
| Instructor | ***Luke Papademas*** | Due Date | **6/16** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 2 in your Computer Architecture and Organization textbook.

**Part 1 Glossary Terms - Data Representation**

Define, in detail, each of these glossary terms from the realm of computer architecture and organization, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or the Internet by visiting Web sites such as:

[**http://www.ask.com**](http://www.askjeeves.com) or [**http://www.webopedia.com**](http://www.webopedia.com/)

**(a) ASCII**

|  |
| --- |
| **A 7 bit character code used to represent numeric, alphabetic, special printable, and control characters. The eighth bit of an ACSII code is used for parity, which means that the bit is either on or off depending on whether the sum of the bits is even or odd.** |

**(b) BCD**

|  |
| --- |
| **Binary coded decimal is a method that converts each digit of a decimal number into a 4 bit binary form. For example, 146 encoded in binary coded decimal would be represented as 0001 0100 0110. There are multiple variants of BCD, including packed BCD which stores two digits per byte and zoned-decimal format.** |

**(c) Error Detection and Correction**

|  |
| --- |
| **Error detection and correction is the concept of handling a reasonable number of the reasonably expected errors. There are different error detection and correction methodologies, including –**   * **Cyclic Redundancy Check (CRC) – a type of checksum that determines whether an error has occurred within a large block or stream of information. It is a type of systematic error detection scheme.** * **Hamming Codes – an adaptation of the concept of parity. They are used in situations where random errors are more likely to occur.** * **Reed-Solomon – a cyclic redundancy check that operates over entire characters instead of only a few bits** |

**(d) Floating - Point Number**

|  |
| --- |
| **A number that can have a decimal point move anywhere relative to the digits of the number. It consists of three parts, a sign bit, an exponent part, and a signifcand.** |

**(e) Unicode**

|  |
| --- |
| **A 16 bit alphabet that is downward compatible with ACSII and the Latin-1 character set. It has the capacity to encode most characters used in every language throughout the world.** |

**Part 2 Exercises - Data Representation**

Provide a brief but complete answer for each of these exercises or find the solution of the given problem.

**(1)** Perform the following base conversions using subtraction or division – remainder.

(a) 588 10 = \_210210\_\_ base 3

(b) 2254 10 = \_33004\_\_\_ base 5

(c) 652 10 = \_1621\_\_\_\_ base 7

(d) 3104 10 = \_4228\_\_\_\_ base 9

**(2)** Represent the following decimal numbers in binary using 8 - bit signed magnitude, one's complement and two's complement and excess - 127 representation.

(a) 77

signed magnitude: \_01001101\_

one's complement: \_01001101\_

two's complement: \_01001101\_

excess - 127: \_11000110\_

(b) − 42

signed magnitude: \_10101010\_

one's complement: \_11010101\_

two's complement: \_11010110\_

excess - 127: \_01010101\_

(c) 119

signed magnitude: \_01110111\_\_

one's complement: \_01110111\_\_

two's complement: \_01111000\_\_

excess - 127: \_11110110\_\_

(d) − 107

signed magnitude: \_11101011\_

one's complement: \_10010100\_

two's complement: \_10010101\_

excess - 127: \_00010100\_\_

**Part 3 Exercises - Data Representation**

**(1)** Suppose a computer uses 4 - bit one’s complement representation. Ignoring overflows, what value will be stored in the variable j after the following pseudocode routine terminates?

( Complete the following table to show your responses. )

**0 → j // store 0 in j**

**−3 → k // store −3 in k**

**while k ≠ 0**

**j = j + 1**

**k = k − 1**

**end while**

|  |  |  |  |
| --- | --- | --- | --- |
| **j** | **binary** | **k** | **binary** |
| **0** | **0000** | **− 3** | **1100** |
| **1** | **0001** | **-4** | **1011** |
| **2** | **0010** | **-5** | **1010** |
| **3** | **0011** | **-6** | **1001** |
| **4** | **0100** | **-7** | **1000** |
| **5** | **0101** | **7** | **0111** |
| **6** | **0110** | **6** | **0110** |
| **7** | **0111** | **5** | **0101** |
| **-7** | **1000** | **4** | **0100** |
| **-6** | **1001** | **3** | **0011** |
| **-5** | **1010** | **2** | **0010** |
| **-4** | **1011** | **1** | **0001** |
| **-3** | **1100** | **0** | **0000** |

**(2)** Assume a 24 - bit word on a computer. In these 24 bits, we wish to represent the value 295 .

(a) How would the computer represent the string 295 ?

(b) If our computer uses 8 - bit ASCII and even parity, how would the computer represent the string 295 ?

(c) If our computer uses packed BCD with zero padding, how would the computer represent the number + 295 ?

Binary Value: \_\_0000000100100111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ASCII: \_\_101100100011100100110101\_\_\_\_\_\_\_\_\_\_\_\_\_

Packed BCD: \_\_0010100101011100\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 4 Exercises - Data Representation**

**(1) ( Mayan Number Systems )**

The ancient Mayans used a base 20 number system. Research their particular number system and then answer each of these:

1. Convert the base ten number 2010 into a Mayan numeral.



1. Convert the Roman Numeral MCXXVII into a Mayan numeral.



**(c)** In your opinion, what were some advantages and disadvantages of the Mayan Number System.

There are only a few symbols to memorize, and it makes sense that increments of 5 have a line, 0 is a shell, and 1-4 are dots. A disadvantage is that to write large numbers, you must use many more characters to represent the number.