**Student Name: Federico Watkins**

**CSIS 1800: Introduction to Computer Science and Information Systems**

**Chapter number: 3**

**Assignment number: 3**

1. How many things can be represented with:

🡪 To answer the following I use the fact that in general n-bits can represent 2n things.

(Dale and Lewis, Computer Science Illuminated, p. 57-59)

a. 2 bits

2n so 🡪 22 = 4

b. 10 bits

2n so 🡪 210 = 1024

c. 16 bits

2n so 🡪 216 = 65536

d. 32 bits

2n so 🡪 232 = 4294967296

2. Give the expressions in DEC (power of 10) and BIN (power of 2) for the following values:

(Dale and Lewis, Computer Science Illuminated, p. 54-59)

1. 1024 Byte

1024 byte = 8192 bit = 100100 🡪 in DEC (power of 10) = 36

1024 byte = 8192 bit 🡪 in BIN (power of 2) = 100100

1. 128 Mbyte

1024 byte = 1024000000 bit = 1111000000 🡪 in DEC (power of 10) = 960

1024 byte = 1024000000🡪 in BIN (power of 2) = 1111000000

1. 16 Gbyte

1024 byte = 128000000000 bit = 1111000000 🡪 in DEC (power of 10) = 960

1024 byte = 128000000000🡪 in BIN (power of 2) = 1011000000000

1. Provide 2’s complement of the following DEC numbers. Use 16-bits to represent each number.   
   (Notice the difference between the positive and negative numbers with the same absolute value? What is the difference?)

To calculate the two’s complement I will invert the bits and add 1. So, I’m taking the positive value and changing all the 1 bits to 0 and all the 0 bits to 1, and then adding 1.

(Dale and Lewis, Computer Science Illuminated, p. 59-63)

1. 1024

102410 = 00000100000000002 = 2’s complement 🡪 00000100000000002

1. -1024

-102410 = 00000100000000002 = 2’s complement 🡪 11111100000000002

1. 128

12810 = 00000000100000002 = 2’s complement 🡪 00000000100000002

1. -128

-12810 = 00000000100000002 = 2’s complement 🡪 11111111100000002

The difference between the positive and negative numbers with the same absolute value seems to be that the positive has all 0’s its significant digits and the negative number has all 1’s after its significant digits

4. Perform the following operations on DEC numbers using two's complement.  
 1) Use 8 bit to represent each number in the first calculation

2) Use 16bit to represent each number in the second calculation  
  
Show all details of the calculations and determine are your results correct? To validate your answer, perform an opposite calculation.

(Dale and Lewis, Computer Science Illuminated, p. 59-63)

1. 127 01111111 🡨 8-bits

+ 1 + 00000001

128 10000000

127 0000000001111111 🡨 16-bits

+ 1 + 0000000000000001

128 0000000010000000

128 0000000010000000 🡨 Validation

- 127 - 0000000001111111

1 0000000000000001

1. -127 10000001 🡨 8-bits

- 1 11111111

-128 10000000

-127 1111111110000001 🡨 16-bits

- 1 1111111111111111

-128 1111111110000000

-128 1111111110000000 🡨 Validation

+ 127 0000000001111111

-1 1111111111111111

1. -128 10000000 🡨 8-bits

+ 127 + 01111111

- 1 11111111

-128 1111111110000000 🡨 16-bits

+ 127 + 0000000001111111

* 1 1111111111111111

- 1 1111111111111111 🡨 Validation

+ 128 + 0000000010000000

127 0000000001111111

1. 128 ~~0~~10000000 🡪OVERFLOW 🡨 8-bits

- 127 10000001

1 00000001

128 0000000010000000 🡨 16-bits

- 127 1111111110000001

1 0000000000000001

1 0000000000000001 🡨 Validation

+ 127 1111111110000001

128 0000000010000000

5. Perform the following operations and express your answers in terms of power of 2:

(Dale and Lewis, Computer Science Illuminated, p. 63-67)

1. 512 KByte \* 16

(512KByte × 210 ) × 16 = 512000byte × 16= 4096000bit × 16 = 65536000bit

1. 16 MByte \* 32

(16MByte × 220 ) × 32 = 16000000byte × 32= 128000000bit × 32 = 4096000000bit

1. 8 GByte \* 16

(8GByte × 230 ) × 16 = 8000000000byte × 16= 64000000000bit × 16 = 1024000000000bit

6. Express in ASCII your first and last names. Use 2 row-table to indicate the correspondence between the letter and the code.

(Dale and Lewis, Computer Science Illuminated, p. 66-69)

(http://www.asciitable.com/)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Chr🡪 | F | e | d | e | r | i | c | o |  | W | a | t | k | i | n | s |
| Dec🡪 | 70 | 101 | 100 | 101 | 114 | 105 | 99 | 111 |  | 87 | 97 | 116 | 107 | 105 | 110 | 115 |

7. How would the following string of characters be represented using run length encoding? What is the compression ratio?

(Dale and Lewis, Computer Science Illuminated, p. 69-71)

AAAABBBBBBBCCCcccCCCCDDDDD the ball EEEFFFFZZZZZZZFFF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AAAA | BBBBBBB | CCC | ccc | | CCCC | DDDDD | | ------ | EEE | | FFFF | ZZZZZZZ | FFF |
| A4 | B7 | CCC | ccc | | C4 | D5 | | ------ | EEE | | F4 | Z7 | FFF |
| \*A4\*B7CCCccc\*C4\*D5 | | | | the Ball | | | | | | EEE\*F4\*Z7FFF | | | |
|  | | | | | | | | | | | | | |
| Compression ratio = Encoded string length ÷ Original string length | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| Encoded string length = 18 | | | | | | | Encoded string length = 12 | | | | | | |
| Original string length = 26 | | | | | | | Original string length = 17 | | | | | | |
| Compression ratio = 0.692 | | | | | | | Compression ratio =0 .706 | | | | | | |

8. Given the following Huffman encoding table, decipher the bit string:

(Dale and Lewis, Computer Science Illuminated, p. 71-73)

Huffman Code Character

00 A

11 E

010 T

0110 C

0111 L

1000 S

1011 R

10010 O

10011 I

101000 N

101001 F

101010 H

101011 D

Encode the following words using Hoffman table (separate the digits from each other):

(Dale and Lewis, Computer Science Illuminated, p. 71-73)

1. Door

|  |  |  |  |
| --- | --- | --- | --- |
| D | O | O | R |
| 101011 | 10010 | 10010 | 1011 |

1. Nonsense

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | O | N | S | E | N | S | E |
| 101000 | 10010 | 101000 | 1000 | 11 | 101000 | 1000 | 11 |

1. Dani

|  |  |  |  |
| --- | --- | --- | --- |
| D | A | N | I |
| 101011 | 00 | 101000 | 10011 |

1. Short

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | H | O | R | T |
| 1000 | 101010 | 10010 | 1011 | 010 |

9. What is the sampling rate per second that is enough to create reasonable sound reproduction?

(Dale and Lewis, Computer Science Illuminated, p. 73-76)

The sampling rate per second that is enough to create a reasonable sound reproduction is 40kHz or 40,000Hz. Due to human anatomy limitations any rate beyond that is considered inaudible to humans.

10. Determine the total capacity of memory in bytes ( use power of 2 calculations):

(Dale and Lewis, Computer Science Illuminated, ch. 2-3)

a. 16 Kbytes \* 1w (w = word = 64 bits)

16Kbytes × 8byte = 128000byte

b. 64 Gbytes \* 1w (w = word = 32bits)

64Gbyte × 4byte = 256000000000byte

c. 4 Tbytes \* 1w (w = word = 16 bits)

4Tbyte × 2byte = 8000000000000byte