Computer organization HW 1

1. a. -5 + (-63)

510 = 0000 01012 -510 = 1111 10112

6310 = 0011 11112 -6310 = 1100 00012

1111 1011 + 1100 0001 = 1011 1100

N = 1 C = 1 Z = 0 V = 0

b. -127 + -128

-12810 = 1000 00002

-12710 = 1000 00012

1000 0001 + 1000 0000 = 0000 0001

N = 0 C = 1 Z = 0 V = 1

c. 105 – (-71)

10510 = 0110 10012 -10510 = 1001 01112

7110 = 0100 01112 -7110 = 1011 10012

0110 1001 + 0100 0111 = 1011 0000

N = 1 C = 0 Z = 0 V = 0

d. -29 – (+101)

2910 = 0001 11012 -2910 = 1110 00112

10110 = 0110 01012  -10110 = 1001 10112

1110 0011 + 1001 1011 = 0111 1110

N = 0 C = 1 Z = 0 V = 1

1. 6810 = 0100 01002 -6810 = 1011 1100

3010 = 0001 11102

12710 = 0111 11112 102310 = 011 1111 11112

127 + -68:

0111 1111 + 1011 1100 = 0011 1011

1023 + -68:

011 1111 1111 + 1011 1100 = 100 1011 1010

127 + 30

0111 1111 + 0001 1110 = 1001 1101

1023 + 30

011 1111 1111 + 000 0001 1110 = 100 0001 1101

1. UTF-8

The UTF-8 is one of the character encodings that uses variable length 8-bit bytes to represent all possible characters in Unicode. The functionalities of this character encoding provides advantageous features such as backwards compatibility due to its variable length encoding. Backwards compatibility, also known as downward compatibility, allows compatibility with input developed for older products.

1. 0027 + 0102 – 015 = (2 \* 70) + (1 \* 21) – (1 \* 50) = 2 + 2 – 1 = 3

3 = 0103

1. 255.75 + 65.0625

255.7510 = 1111 1111.112

1.111111111 \* 27

127 + 7 = 134

A 🡪 0 10000110 1.11111111100000000000000

65.062510 = 0100 0001.00012

1.0000010001 \* 26

127 + 6 = 133

B 🡪 0 10000101 1.00000100010000000000000

Shift significand of lower number

B’ 🡪 0 10000101 0.100000100010000000000000

Add significands

11111111100000000000000 + 100000100010000000000000 = 11000000110100000000000

0 10000101 0.110000001101000000000000

255.75 + 65.0625 = 0 10000101 1.10000001101000000000000

1200) 434F 4D50 5554 5245

1208) FFF2 3508 0100 57FC

1210) 7892 4123 0000 1234

Decode the hexadecimal data, assuming that it is a sequence of:

* 8 ASCI encoded characters – write the characters

43 4F 4D 50 55 54 52 45

C O M P U T R E

* one 2’s complement 16-bit number – present the decimal number

FFF2

1111 1111 1111 00102 = -1410

* IEEE Floating point single precision number –present the binary scientific notation of the number ;

3508 0100

0 01101010 00010000000000100000000

EXP = 106 – 127 = -21

1. 00010000000000100000000 \* 2-21

* one byte binary number in octal system - present the octal number;

57 = 0101 0111

001 010 111 = 127

The octal number representation is 127

* one 8-bit number presented in Excess 127 code - present the decimal number

FC = 1111 1100 12710 = 0111 1111

1111 1100 + 0111 1111 = 0011 1011 (with carry 1)

0011 10112 = 5910

* 4 bytes natural BCD number – present the encoded decimal number

7892 4123 = 0111 1000 1001 0010 0100 0001 0010 00112

0111 1000 1001 0010 0100 0001 0010 00112 =202285085110

202285085110 as BCD = 0010 0000 0010 0010 1000 0101 0000 1000 0101 0001

* Special IEEE Floating point number – present the binary scientific notation

0000 1234 = 0 00000000 0.00000000001001000110100

This special IEEE floating point case shows that this number is in denormalized form

. 00000000001001000110100 \* 20

1. Presentation of images. Explain:

Raster form of images:

This form of imaging uses an immense amount of colored pixels that develop the whole image. Due to their fixed amount of pixels there seem to be some problems in regards to resizing these images. Stretching these images may cause them to be grainy and become distorted.

Vector form of images:

This form of imaging, opposed to raster imaging, use mathematical formulas to construct an image as they construct line segments, polygons, and points. Vector imaging allows for resizing of the image without degrading the quality of an image.