1. **Finish your homework independently**
2. **Convert this docx to pdf: “stuID\_name\_csapp2.pdf”**

**Example: ”2017010000\_zhangsan\_csapp2.pdf”**

1. **Submit this pdf: learn.tsinghua.edu.cn**

**2.82 ◆◆**

Consider numbers having a binary representation consisting of an infinite string

of the form 0*.y y y y y y . . .*, where *y* is a *k*-bit sequence. For example, the binary

representation of is 0*.*01010101 *. . .* (*y* = 01), while the representation of is

0*.*001100110011 *. . .* (*y* = 0011).

A. Let that is, the number having binary representation *y*. Give

a formula in terms of *Y* and *k* for the value represented by the infinite string.

Hint: Consider the effect of shifting the binary point *k* positions to the right.

Formula:

Let value=0. y y y y y. . . , Y = unsigned(y).

Shift the binary point *k* positions to the right, y. y y y y y. . .=value\*2^k.

Also, value + Y = y. y y y. . . = value\*2^k.

It gives :

B. What is the numeric value of the string for the following values of *y*?

(a) 001

(b) 1001

(c) 000111

**2.85** ◆

Intel-compatible processors also support an “extended precision” floating-point

format with an 80-bit word divided into a sign bit, *k* = 15 exponent bits, a single

*integer* bit, and *n* = 63 fraction bits. The integer bit is an explicit copy of the

implied bit in the IEEE floating-point representation. That is, it equals 1 for

normalized values and 0 for denormalized values. Fill in the following table giving

the approximate values of some “interesting” numbers in this format:

|  |  |  |
| --- | --- | --- |
|  | Extended precision | |
| Description | Value | Decimal |
| Smallest positive denormalized |  |  |
| Smallest positive normalized |  |  |
| Largest normalized |  |  |

bias=2^{14} -1=16383

Smallest positive denormalized:

E=1-16383=-16382, M=0.00…1=2^{-63}

Smallest positive normalized:

Exp=000…1=1, E=exp-16383=-16382, M=1.000…0=1

Largest normalized:

Exp=11…10, E=exp-16383=0111…11=2^{14}-1=16383, M=1.1111…111=2-2^{-63}

**2.87** ◆◆

Consider the following two 9-bit floating-point representations based on the IEEE

floating-point format.

**1.** Format A

There is one sign bit.

There are *k* = 5 exponent bits. The exponent bias is 15.

There are *n* = 3 fraction bits.

**2.** Format B

There is one sign bit.

There are *k* = 4 exponent bits. The exponent bias is 7.

There are *n* = 4 fraction bits.

Below, you are given some bit patterns in Format A, and your task is **to convert**

**them to the closest value in Format B**. If rounding is necessary, you should ***round***

***toward* +∞**. In addition, give the values of numbers given by the Format A and

Format B bit patterns. Give these as whole numbers (e.g., 17) or as fractions (e.g.,

or).

|  |  |  |  |
| --- | --- | --- | --- |
| Format A | | Format B | |
| Bits | Value | Bits | Value |
| 1 01110 001 |  | 1 0110 0010 |  |
| 0 10110 101 |  | 0 1110 1010 |  |
| 1 00111 110 |  | 1 0000 0111 |  |
| 0 00000 101 |  | 0 0000 0001 |  |
| 1 11011 000 |  | 1 1110 1111 |  |
| 0 11000 100 |  | 0 1111 0000 |  |

Format A

bias=2^{4}-1=15,

1 01110 001, s=1, E=14-15=-1, M=1.001=9/8

0 10110 101, s=0, E=22-15=7, M=1.101=13/8

1 00111 110, s=1, E=7-15=-8, M=1.110=14/8

0 00000 101, s=0, E=1-15=-14, M=0.101=5/8

1 11011 000, s=1, E=27-15=12, M=1.000=8/8

0 11000 100, s=0, E=24-15=9, M=1.100=12/8=3/2

Format B

bias=2^{3}-1=7

1 0110 0010, s=1, E=6-7=-1, M=1.0010=9/8

0 1110 1010, s=0, E=14-7=7, M=1.1010=13/8

1 0000 0111, s=1, E=1-7=-6, M=0.0111=7/16

0 0000 0001, s=0, E=1-7=-6, M=0.101=1/16

1 1110 1111, s=1, E=14-7=7, M=1.1111=31/16

0 1111 0000, inf