CSPB 2400 - Park - Computer Systems

Started on	Wednesday, 24 January 2024, 4:47 PM
State	Finished
Completed on	Wednesday, 24 January 2024, 5:12 PM
Time taken	25 mins 23 secs
Marks	10.60/11.00
Grade	**9.64** out of 10.00 (**96**%)

Question **1**

Correct

Mark 1.00 out of 1.00

It is always true that "a + b + c" and "c + b + a" will produce the same result for IEEE floating values "a", "b" and "c".

Select one:

- True
- False ✔

Question **2**

Correct

Mark 1.00 out of 1.00

It is always true that "a * b" and "b * a" will produce the same result for IEEE floating values "a" and "b".

Select one:

- True ✔
- False

Question **3**

Correct

Mark 1.00 out of 1.00

Convert the binary fractional number 11.0001_2 to a decimal representation. You can use fractions or floating point values, but your answer must be precise. You can use expressions if that's useful.

3.0625

Your last answer was interpreted as follows: 3.0625

Correct answer, well done.

Question **4**

Correct

Mark 1.00 out of 1.00

Convert the binary fractional number 110.10_2 to a decimal representation. You can use fractions or floating point values, but your answer must be precise. You can use expressions if that's useful.

6.5

Your last answer was interpreted as follows: 6.5

Correct answer, well done.

Question **5**

Correct

Mark 1.00 out of 1.00

Consider a 12-bit floating-point representation based on the IEEE floating-point format, with one sign bit, 5 exponent bits and 6 fraction bits. The exponent bias follows the IEEE standard.

Interpret the bitstring $1\_01001\_100000_2$ using this 12-bit floating-point representation and fill in the table below.

You must express your results precisely (e.g. using rationals (M/N) and exponents (M^N), rather than approximate decimal fractions such as 0.ABCD). You may use expressions if it's useful.

Field	Mean	Value
e	The value represented by considering the exponent field to be an unsigned integer (as a decimal value)	9 Your last answer was interpreted as follows: 9
E	The value of the exponent after biasing (as a decimal value)	-6 Your last answer was interpreted as follows: -6
2^E	The numeric weight of the exponent (as a decimal value)	1/64 Your last answer was interpreted as follows: $\frac{1}{64}$
f	The value of the fraction (as a fraction such as 3/4 or the exact floating point number)	1/2 Your last answer was interpreted as follows: $\frac{1}{2}$
M	The value of the significand (as a fraction such such as 7/4 or the exact floating point number)	3/2 Your last answer was interpreted as follows: $\frac{3}{2}$
$s*2^E * M$	The value of the number in decimal. The 's' is equal to +1 if the number is positive and -1 if it is negative.	-1*(1/64)*(3/2) Your last answer was interpreted as follows: $-1 \cdot \left(\frac{1}{64}\right) \cdot \left(\frac{3}{2}\right)$

Correct answer, well done.
Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Question **6**

Correct

Mark 1.00 out of 1.00

Consider a 8-bit floating-point representation based on the IEEE floating-point format, with one sign bit, 4 exponent bits and 3 fraction bits. The exponent bias follows the IEEE standard.

Interpret the bitstring $0\_0000\_100_2$ using this 8-bit floating-point representation and fill in the table below.

You must express your results precisely (e.g. using rationals (M/N) and exponents (M^N), rather than approximate decimal fractions such as 0.ABCD). You may use expressions if it's useful.

Field	Mean	Value
e	The value represented by considering the exponent field to be an unsigned integer (as a decimal value)	0 Your last answer was interpreted as follows: 0
E	The value of the exponent after biasing (as a decimal value)	-6 Your last answer was interpreted as follows: -6
2^E	The numeric weight of the exponent (as a decimal value)	1/64 Your last answer was interpreted as follows: $\frac{1}{64}$
f	The value of the fraction (as a fraction such as 3/4 or the exact floating point number)	1/2 Your last answer was interpreted as follows: $\frac{1}{2}$
M	The value of the significand (as a fraction such such as 7/4 or the exact floating point number)	1/2 Your last answer was interpreted as follows: $\frac{1}{2}$
$s*2^E * M$	The value of the number in decimal. The 's' is equal to +1 if the number is positive and -1 if it is negative.	1*(2^-6)*(1/2) Your last answer was interpreted as follows: $1 \cdot 2^{-6} \cdot \left(\frac{1}{2}\right)$

Correct answer, well done.
Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Question **7**

Correct

Mark 1.00 out of 1.00

Consider a 9-bit floating-point representation based on the IEEE floating-point format, with one sign bit, 4 exponent bits and 4 fraction bits. The exponent bias follows the IEEE standard.

Convert the decimal floating point number -0.0029296875 (or $-\frac{3}{1024}$) to the appropriate IEEE representation. Enter the binary representation of each field (sign, exp, fraction)

s	e	f
1 Your last answer was interpreted as follows: 1	0000 Your last answer was interpreted as follows: 0	0011 Your last answer was interpreted as follows: 11

Correct answer, well done.
Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Question **8**

Correct

Mark 1.00 out of 1.00

Consider a 10-bit floating-point representation based on the IEEE floating-point format, with one sign bit, 4 exponent bits and 5 fraction bits. The exponent bias follows the IEEE standard.

Convert the decimal floating point number -0.046875 (or $-\frac{3}{64}$) to the appropriate IEEE representation. Enter the binary representation of each field (sign, exp, fraction)

s	e	f
1 Your last answer was interpreted as follows: 1	0010 Your last answer was interpreted as follows: 10	10000 Your last answer was interpreted as follows: 10000

Correct answer, well done.
Correct answer, well done.

Correct answer, well done.

Correct answer, well done.

Question **9**

Correct

Mark 1.00 out of 1.00

Show how the following binary fractional values would be rounded to the nearest half (one bit to the right of the binary point), according to the round-to-even rule. In each case, show the numeric values, both before and after rounding and the bit pattern after rounding.

Finally, please DO NOT USE fractions and type the decimal values very precisely (accurate to the last decimal place).

Binary Pattern	Value before rounding	Bit pattern rounded to the nearest half	Value after rounding
10.010	2.25 ✔	10 ✔	2 ✔
10.111	2.875 ✔	11 ✔	3 ✔
10.110	2.75 ✔	11 ✔	3 ✔
11.001	3.125 ✔	11 ✔	3 ✔

Question **10**

Partially correct

Mark 1.60 out of 2.00

Work through question number 2.88 in the *Computer Systems: A Programmer's Perspective, 2nd Edition* textbook:

We are running programs on a machine where values of type int have a 32- bit two's-complement representation. Values of type float use the 32-bit IEEE format, and values of type double use the 64-bit IEEE format. We generate arbitrary integer values x, y, and z, and convert them to values of type double as follows:

```
/* Create some arbitrary values */
int x = random();
int y = random();
int z = random();
/* Convert to double */
double dx = (double) x;
double dy = (double) y;
double dz = (double) z;
```

For each of the following C expressions, you are to indicate whether or not the expression always yields 1. If it always yields 1, describe the underlying mathematical principles. Otherwise, give an example of arguments that make it yield 0. Note that you cannot use an IA32 machine running gcc to test your answers, since it would use the 80-bit extended-precision representation for both float and double.

A. (float) x == (float) dx Yes. Converting to float could cause rounding, but both x and dx will be rounded the same way. ✔

B. dx-dy == (double) (x-y) No. Let x = 0 and y = TMin32 ✔

C. (dx + dy) + dz == dx + (dy + dz) No, loss of precision in addition. ✖

D. (dx * dy) * dz == dx * (dy * dz) No, floating point multiplication not associative. ✔

E. dx/dx == dz/dz No, let x=0 and z = 1d ✔