Tutorial 2

Part A: Multiple Choice Question (MCQ)

Q1 Convert the following 8 bits two's complement binary numbers to 10 bits system E = 0011 1100 and F =1101 0011

A. E = 10 0011 1100 and F = 01 1101 0011

B. E = 00 0011 1100 and F = 11 1101 0011

C. E = 11 0011 1100 and F = 00 1101 0011

D. E = 01 0011 1100 and F = 10 1101 0011

Q2 Perform the following numbers addition using 10 bit binary system based on one's complement:

10 1100 0011 + 11 1000 1100

A. 11 0100 1111

B. 10 0101 0000

C. 10 0100 1111

D. 01 1100 111

Q3 Perform the following numbers subtraction using 7 bit binary system based on two's complement: 21 - 7

E – F = S

What is the binary values of E, -F and S?

A. 111 0101, 100 0111, 011 1100

B. 001 0101, 111 1001, 000 1110

C. 001 0101, 100 0111, 001 1100

D. 001 0101, 000 0111, 001 1100

Q4 The first bit is used for the sign of the number, the next three for the biased exponent and the next three for the magnitude of the mantissa. What is decimal number represent by 0010110 ?

A. 3.5

B. 0.375

C. 1.5

D. 0.1875

Q5 Why do we applied biased exponent in standard floating point?

A. To represent negative exponents

B. It is a process of deleting the zeroes

C. It is one step to normalize the number

D. To avoid overflow

Q6 The digit after the radix point is called as __________.

A. Determinant

B. Exponent

C. Mantissa

D. Bias value

Q6 For addition and subtraction in floating point, it is necessary ________.

A. to ensure that each operands have different exponent value

B. to ensure that both operands have the same number of digit value after the radix

C. to ensure that both operands have the same exponent value

D. to ensure that both operands need to be absolute

Q7 Express (-1/32) using single precision floating point format.

A. 1 10000110 00000000000000000000000

B. 1 00101010 00000000000000000000000

C. 1 01100100 00000000000000000000000

D. 1 0111010 00000000000000000000000

Part B: Structure/Explanation

2.1 Represent the following decimal numbers in both binary sign/magnitude and two's complement using 16-bits binary system:

(a) +512 (b) –29

2.2 Represent the following two's complement values in decimal:

(a) 1101011_2 (b) 0101101_2

2.3 Convert the following 8-bit two's complement value to 16-bits binary system:

(a) 11001100_2 (b) 00101110_2

2.4 Assume numbers are represented in 8-bits two's complement representation. Show the calculation of the following:

(a) 6 + 13 (c) 6 – 13

(b) –6 + 13 (d) –6 – 13

2.5 Find the following differences using two's complement arithmetic:

(a) $111000_2 - 110011_2$

(b) $11001100_2 - 101110_2$

(c) $111100001111_2 -$ $1100\ 1111\ 0011_2$

2.6 Perform the subtraction in two's complement representation for $37_{10} - 17_{10}$ in 8 bit and 16 bit binary system.

2.7 In 4-bit binary arithmetic, find the multiplication of 5_{10} with 4_{10} using the 1st version of highly optimized multiplication hardware.

2.8 In 6-bit binary arithmetic, find the multiplication of 21_{10} with 14_{10} using the 1st version of highly optimized multiplication hardware.

2.9 In 6-bit binary arithmetic, find the multiplication of 21_{10} with (-14_{10}) by using:

a) the two's complement binary numbers. Proof that it yields incorrect result.

b) the 1st version of highly optimized multiplication hardware.

2.10 Using a 4-bit binary arithmetic, find the division of (-7_{10}) by 2_{10} with the 1st version of highly optimized division hardware.

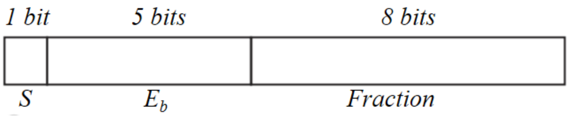
2.11 Using a 4-bit binary arithmetic, find the division of the following numbers with the 1st version of highly optimized division hardware.

a) 6_{10} by 3_{10}

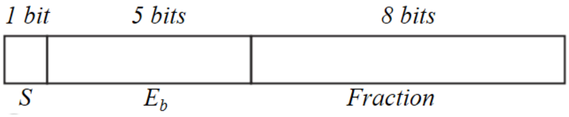
b) 6_{10} by (-3_{10})

c) (-12_{10}) by 5_{10}

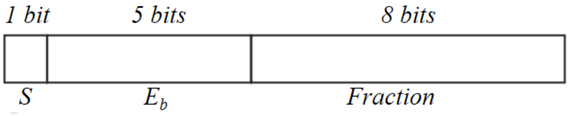
2.12 Convert 0.1110111 x 2^2 to a normalized 14-bit format with a bias of 16. Show your working.



2.13 Transform (-33.625_{10}) to floating point using the following format (radix 2). Show your working.



2.14 Transform (-0.03125_{10}) to floating point using the following format (radix 2). Show your working.



2.14 Complete the table with the normalized binary number and its exponent respectively using single precision floating-point.

	Binary Values	Normalized as	*Exponent (e')*
(a)	1101.101		
(b)	0.00101		
(c)	1.0001		
(d)	10000011.0		

2.15 Complete the table with the *biased exponent* (E_B) and binary representation for each number using the type of floating-point respectively.

	Exponent (e')	*Biased Exponent (E_B)*			
		Single Precision		Double Precision	
		(Dec)	*(Bin)*	*(Dec)*	*(Bin)*
(a)	3				
(b)	– 3				
(c)	0				
(d)	7				

2.16 Convert (-0.75_{10}) to single precision floating-point.

1 bit	*8 bits*	*23 bits*
Sign	*Biased Exponent*	*significand / fraction*

2.17 Convert the following number to single precision floating-point.

(a) (-33.625_{10})

(b) 0.03125_{10}

2.18 Complete the table with all *sign (S)*, *exponent (e')* and *fraction (F)* values if single precision floating-point applied.

		(Decimal)		*(Binary)*	
	Binary Values	E_B	S	E_B	*Fraction*
(a)	-1.11				
(b)	+1101.101				
(c)	-0.00101				
(d)	+100111.0				
(e)	+0.0000001101011				

2.19 Convert the following number to double precision floating-point.

(a) (-0.75_{10})

(b) 10.4_{10}

2.20 What is the decimal number represented by this double precision float?

1 bit	*11 bits*	*52 bits*
0	0 1 1 1 1 1 1 1 1 1 0	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 . . . 0

2.21 Add these two binary floating-point numbers.

$0.6015625_{10} + 0.78125_{10} = \_\_\_\_\_\__2$

2.22 Add the following binary numbers as represented in a normalized single precision format.

0	1 1 0 0 1 0 0 0	1 1 1 1 0 1 0 1 0 0 ..... 0 0 0	+
0	1 0 0 1 1 0 1 0	0 1 0 1 0 0 1 1 0 0 ..... 0 0 0	

2.22 Given two numbers 0.5_{10} and -0.4375_{10}.

(a) Multiply the numbers in binary.

(b) Converting to decimal to check the results.

Show your workings.

2.23 What are the three component parts of a floating-point number?

2.24 How many bits long is a double-precision number under the IEEE-754 floating-point standard?

2.25 Perform the following binary multiplications:

a) 1100 x 101

b) 10101 x 111

c) 11010 x 1100

2.26 Perform the following binary divisions:

a) 101101 ÷ 101

b) 10000001 ÷ 101

c) 1001010010 ÷ 1011

2.27 Express the following numbers in IEEE 32-bit floating-point format:

(a) – 8

(b) – 7

(c) – 2.5

(d) 384

(e) 1/16

(f) – 1/4

2.28 The following numbers use the IEEE 32-bit floating-point format. What is the equivalent decimal value?

(a) 1 10000000 11000000000000000000000

(b) 0 01111111 00000000000000000000000

(c) 0 10000011 10100000000000000000000

2.29 Consider a floating-point format with 8 bits for the biased exponent and 23 bits for the significand. Show the bit pattern for the following numbers in this format:

a) – 720

b) 0.645

2.30 Show how the following floating-point calculations are performed (where significands are truncated to 4 decimal digits). Show the results in normalized form.

(a) $7.286 \times 10^{2} + 7.847 \times 10^{2}$

(b) $3.314 \times 10^{1} + 8.227 \times 10^{-2}$

(c) $(8.954 \times 10^{1}) \times (1.324 \times 10^{0})$

2.31 Assume we are using a floating-point representation uses a 14-bit format, 5 bits for the exponent with a bias of 16, a normalized mantissa of 8 bits, and a single sign bit for the number):

(a) Show how the computer would represent the numbers 100.0 and 0.25 using this floating-point format.

(b) Show how the computer would add the two floating-point numbers in part (a) by changing one of the numbers so they are both expressed using the same power of 2.

(c) Show how the computer would represent the sum in part (b) using the given floating-point representation. What decimal value for the sum is the computer actually storing? Explain.