

**ISTANBUL TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT**

BLG 212E

**Microprocessor Systems
Homework I**

150190739 : Yasin Abdulkadir YOKUŞ

November 23, 2020

FALL 2020

Contents

FRONT COVER

CONTENTS

1	QUESTION I	1
1.1	1.a.1	1
1.2	1.a.2	1
1.3	1.a.3	2
1.4	1.a.4	2
1.5	1.b.1	3
1.6	1.b.2	4
1.7	1.b.3	4
2	QUESTION II	5
2.1	2.a. 1, 2, 3	5
2.2	2.b	6

1 QUESTION I

1.1 1.a.1

To convert from decimal to hexadecimal.

Conversion steps:

- Divide the number by 16.
- Get the integer quotient for the next iteration.
- Get the remainder for the hex digit.
- Repeat the steps until the quotient is equal to 0.

Table 1: Conversion of 123_{10} to hex

Division by 16	Quotient (integer)	Remainder (decimal)	Remainder (hex)	Digit #
123/16	7	11	B	0
7/16	0	7	7	1

Answer : 7B

1.2 1.a.2

To convert from base 3 to decimal, we need to find the sum of the digits multiplied with powers of 3. For base 3 number with n digits $d_{n-1}...d_3d_2d_1d_0$, multiply each digit of the base 3 number with its corresponding power of 3 and sum:

$$Result(Decimal) = d_{n-1} * 3^{n-1} + \dots + d_3 * 3^3 + d_2 * 3^2 + d_1 * 3^1 + d_0 * 3^0 \quad (1)$$

Result in base 10 is equal to each digit of "02102210" multiplied with its corresponding powers of 3.

$$Result = 0 * 3^7 + 2 * 3^6 + 1 * 3^5 + 0 * 3^4 + 2 * 3^3 + 2 * 3^2 + 1 * 3^1 + 0 * 3^0 \quad (2)$$

$$Result = 1458 + 243 + 54 + 18 + 3 = 1776 \quad (3)$$

Answer : $02102210_3 = 1776_{10}$

1.3 1.a.3

To convert "01010100" from base 2(binary) to hexadecimal, we need to find the sum of the digits multiplied with powers of 2. This gives us intermediate result of decimal base, then we convert this result into hexadecimal base. Other solution method we can use is we separate the given binary into groups of 4 digits. Then write corresponding hexadecimal values for separated groups. Key thing to consider here is if we don't have 4, 8, 12... digits then we replace 0s to left of the number.

$$Result(\text{Hexadecimal}) = 01010100 \quad (4)$$

$$Result = 0101 \quad 0100 \quad (5)$$

$$Result = 5 \quad 4 \quad (6)$$

$$Result = 54 \quad (7)$$

Answer : 54

1.4 1.a.4

Single precision has 32 bits total which are sign (1 bit), exponent (8 bits), and mantissa (23 bits).

We separate the whole and the decimal part of the number. Whole number portion is "7" and a decimal number portion is "0.325".

First, let us convert the whole number into binary.

Table 2: Conversion of 123_{10} to hex

Whole Number Division	Result	Remainder
$7/2$	3	1
$3/2$	1	1
$1/2$	0	1

From down to top, $7_{10} = 111_2$

Second, let us convert the decimal portion into binary. This would be the 0.325 from 7.325, which will be 0.0101001100110011001 when converted into binary by creating a table like the one above.

$$0.325_{10} = 0.0101001100110011001_{2}$$

Third, combine the two parts of the number that have been converted into binary. The number 7 in binary is 111 and the decimal portion 0.325 in binary is 0.0101001100110011001. When we combine them using a decimal point, we end up with 111.01010011001100110011001 as our final answer. Let us convert the binary number into base 2 scientific notation.

$$111.01010011001100110011001 = 1.110101001100110011001 * 2^2$$

Since the original number is positive, this number is also positive and sign bit is 0. This will be the first bit out of the 32 total bits in our IEEE 754 single precision representation. There is set bias for single precision. The exponent bias for single precision is 127, which means we must add the base 2 exponent found previously to it. Thus, the exponent we will use is 127+2 which is 129. Then we turn the exponent into binary, so that it could be used in the IEEE 754 conversion. $129_2 = 10000001_2$

Last but not least, we determine the mantissa. The mantissa part of the IEEE 754 conversion, is the rest of the number after the decimal of the base 2 scientific notation.

$$1101010011001100110011001$$

$$\text{Answer (as Sign, Exponent and Mantissa)} = 0 \ 10000001 \ 11010100110011001100110 \quad (8)$$

1.5 1.b.1

Since X and Y are unsigned numbers, overflow is detected from the end carry-out bit of the MSB.

$$X = 1011 \ 0010 \quad (9)$$

$$Y = 0110 \ 1101 \quad (10)$$

$$X + Y = 1 \ 0001 \ 1111 \quad (11)$$

Answer : There is an Overflow.

1.6 1.b.2

When two signed numbers are added, the sign bit is treated as part of the number. End carry does not indicate overflow. Overflow cannot occur if one number is positive and the other is negative. Overflow occurs if the sign bit changes for the addition of two positive, or two negative number.

$$X = 0101 \quad 1100 \quad (12)$$

$$Y = 0011 \quad 0101 \quad (13)$$

$$X + Y = 1001 \quad 0001 \quad (14)$$

Sign bit changes for the addition of two positive numbers from 0 to 1.

Answer : There is an Overflow.

1.7 1.b.3

While subtracting in signed number, we add two numbers including their sign bits, then discard the carry bit. Negative numbers are represented in 2s complement.

$$X = 1110 \quad 0011 \quad (15)$$

$$Y = 0111 \quad 1111 \quad (16)$$

$$2s \text{ complement of } Y = 1000 \quad 0000 + 1 \quad (17)$$

$$2s \text{ complement of } Y = 1000 \quad 0001 \quad (18)$$

Since $X - Y$ equals $X + 2^3$'s complement of Y ,

$$X + 2s \text{ complement of } Y = 1110 \quad 0011 + 1000 \quad 0001 \quad (19)$$

$$X - Y = 1 \quad 0110 \quad 0100 \quad (20)$$

Discard the carry bit. If the sum of two negative numbers yields a positive result, the sum has overflowed.

$$X - Y = 0110 \quad 0100 \quad (21)$$

Sign bit changes from 1 to 0.

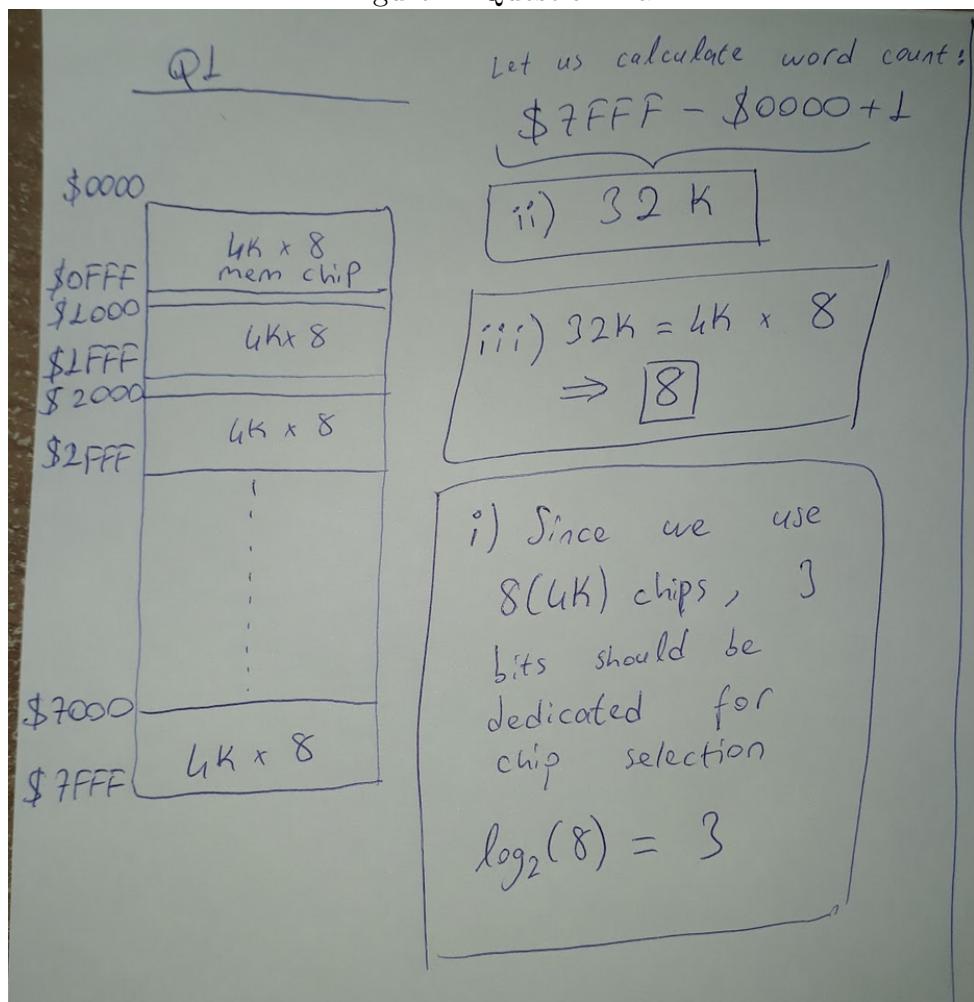
Answer : There is an Overflow.

2 QUESTION II

2.1 2.a. 1, 2, 3

- i) Since we use 8(4K) chips, 3 bits should be dedicated for chip selection $\log_2(8) = 3$.
- ii) Let us calculate the word count from hexadecimal to decimal, $\$7FFF - \$0000 + 1 = 32K$.
- iii) $32K = 4K * 8$ Each rectangle is 4K in the given memory map.

Figure 1: Question 2.a



Answer for i, ii, iii in order: 3, 32K, 8

2.2 2.b

From the previous problem, we know that we need 8 4K chips for this design. Since these memory chips 4K address lines, $2^2 * 2^{10} = 2^{12}$, we will use 12 bits for address location selection within the chip. Remaining 4 bits $A_{12} \dots A_{15}$ can be used in chip selection.

Then we connect 3 bits $A_{12} \dots A_{14}$ to decoder inputs and also A_{15} to decoder select. As you can see, we connected 8 bit data bust and RD,WR to memory chips also. One last choice is using active high as select signals, which we did for memory chip and decoder select connections. Depending on memory chip inputs we start selecting chips addressing from \$0000 to \$0FFF, and so on.

Figure 2: Question 2.b

