#cec32xNotes, cec32x_lab_24_Lab4–fixed-point_expression_and_operations

Lab 4. Fixed-Point Expression and Operations

Introduction

We have learned in Module 4 that we can use the fixed-point expression to express real numbers using integers. We have also learned addition, subtraction, and multiplication operations using fixed-point expressions.

In this lab, we verify these expressions and operations using programming.

The key points of this lab are as follows:

- Expressing two real numbers in the Qm.n format. These two real numbers are given in the program as floating-point numbers.
- Printing out the Qm.n expression of these two numbers.
- Performing addition, subtraction, and multiplication of these two real numbers directly using floating-point numbers.
- Performing addition, subtraction, and multiplication of these two real numbers in Qm.n expression.
- Printing out the results from the above two different approaches and see the differences.

During the programming, we use the Q7.8 expression to express the real numbers. To help you get started, we use the following code in `main.c`:

```
#include <stdio.h>
#include <math.h>

float f1 = 3.1415;
float f2 = -10.5;
float f_sum, f_sub, f_mul;

int16_t A1, A2, A_sum, A_sub, A_mul;

// We use Qm.n format here:
int m = 7;
int n = 8;

int main(void) {

}
```

Note that:

- You need to use the `pow` function prototyped in the `math.h` file to calculate as shown in the Example of Section 2.3 on Page 9 of the class notes of Module 4.
- You need to use the `round` function to round a floating-point number to an integer. This is also prototyped in the `math.h` file.

Lab Tasks

There are FIVE tasks in this lab, as detailed below.

Task 1. Convert ''f1'' and ''f2'' to integers

(15 points).

You should obtain `A1` and `A2` from `f1` and `f2`, respectively, as shown in the Example of Section 2.3 on Page 9 of the class notes of Module 4. For example, if `f1` is -3.1415926, you should have `A1  =` -12868 if the fixed-point expression is Q3.12.

Task 2. Display the fixed-point expression in hexadecimal form

(15 points).

Now, you need to display the values of the above `A1` and `A2` in hexadecimal form.

Note that the computer expresses the numbers using the binary format, and the decimal and hexadecimal formats are just for our humans to see the numbers in easier ways.

Task 3. Calculate results using floating-point numbers directly

(10 points).

Calculate the summation, subtraction, and multiplication of `f1` and `f2` directly and save the results in `f_sum`, `s_sub`, and `f_mul`, respectively.

Task 4. Calculate results using fixed-point numbers

(20 points).

Calculate the summation, subtraction, and multiplication of `f1` and `f2` in fixed-point format using the approaches shown on Page 10 of the class notes for Module 4. Save the results in `A_sum`, `A_sub`, and `A_mul`, respectively.

Note that when you calculate `A_mul`, there might be an overflow issue depending on how you do it. A hint is to cast `A_1` to a 32-bit integer before doing the multiplication to make sure the multiplication result is 32 bits. Multiplying two 16-bit numbers will not have an overflow problem when the result is expressed in 32 bit. Note that you need to guarantee there is no overflow with `A_mul` is `f_mul` is not beyond the range of Qm.n expression.

Task 5. Print out the calculation results

(20 points).

Now display and contrast the calculation results using floating-point numbers and fixed-point numbers. The corresponding values should be close. If not, there may be some errors.

Note that `A_sum`, `A_sub`, and `A_mul` are all **16-bit integers** and they **represent** real numbers in the fixed-point form. To print out these real numbers, you need to **place** the point to the appropriate position by multiplying a scalar . Note also that when you multiplying the above scalar, you should not use the approach of the division of two integers; this will lead to a loss of accuracy as the fraction is removed. You have to think about this carefully and a discussion is expected in the report.

If you have questions regarding the details, please ask.

Submission of Lab Report

The demo of the results of this lab is not that crucial as the correct results will be embedded in the lab report, which need to include the following.

- The code snippets of all your code. Note that you need to write the code with clear indentation and comments so that a fellow programmer can understand your code easily.
- A screenshot showing your results. Note that you need to print out your name(s) in one or another. Note that the correct results from the screenshot are expected to justify the points for the five tasks.

Note that you need to name your project and pdf file name according to the naming convention we have been using for the previous labs.

There are 20 points given to the following sections of the report:

- (10 points) Explanation of how you did the calculation of the multiplication using Q7.8.
- (10 points) Explanation of how you printed out `A_sum` as a real number.