* General description or introduction of the problem and your solution

The problem has two requirments. Firstly, it requires us to implement gaussian blur algorithm including file in and out functions in c++ code. We can verify the program by comparing the difference between provided image and generated image. Secondly, we need to transfer our code to systemC processes with the following requirements.

* + - A Input process to parse image format and send pixels to next process
    - A Calculation process to receive pixels and do Gaussian blur. Then pass results to next process. In this part, Calculation process will send all necessary pixels for each Gaussian blur calculation.
    - A Output process to dump processed pixels as a BMP image.
    - Use SystemC datatype as possible for Calculation process.
    - Implement handshaking between Input and Calculation processes, and also Calculation and Output processes.

We will discuss my solution in next section.

* Implementation details (data structure, flows and algorithms)

My implement is based on provided files in lab02-Sobel Filter with FIFO interface with some necessary modifications.

Text

Description automatically generated

Main.cpp

In main.cpp file, most of the code remain unchanged. I only add three sc\_fifo: i\_r, i\_g i\_b as fifo between filter and testbench.

Text

Description automatically generated

Testbench.cpp

In Testbench.cpp, I alter do\_sobel function by removing “total” variable and replacing it with three variables: o\_r, o\_g, o\_b. Moreover, I apply min and max function in target\_bitmap to avoid values over 255 or under 0. I made those modification so that my program can generate colored images rather than gray images. Besides, values form filter process will be added bias and be divided by factor to meet the requirements of gaussian blur. By using fifo my program can do the filtering in another systemC process.

Graphical user interface, text, application

Description automatically generated

SobelFilter.cpp

In SobelFilter.cpp, I changed the mask and the calculation process so that it can apply the mask on red, green, and blue separately, instead of making those colors gray in the previous process.

* Additional features of your design and models

My program can generate colored gaussian blur.

* Experimental results (on different benchmarks and settings)

lena\_std\_short.bmp

A picture containing person, clothing, hair, indoor

Description automatically generatedA picture containing clothing, person, person, hair

Description automatically generated

Image on the left is the original one and the one on the right is after the gaussian blur process. Although, the effect is not obvious, we can still see the difference in red circuit. The right one is vaguer than the left one

lena\_sobel\_short.bmp

Diagram

Description automatically generatedA picture containing text, linedrawing

Description automatically generated

I apply my program in image after sobel process. The difference is obvious. The one on the left in before and the one on the right is after.

* Discussions and conclusions.

This homework is quite hard, but it is without a doubet a good practice for us to learn systemC. Besides, we have to write our report in English and submit our homework to GitHub, which are not familiar to me. Anyway, thank you for providing us this challenge.