Implement of Many-Core System Homework 1

109061564 馬婕芸

Introduction

In this homework, there are two main parts. One is to build a Gaussian blur filter using C/C++ codes based on the given sobel C source codes. Another is to convert implemented Gaussian blur C codes into SystemC process using SystemC datatypes. The implementation details and simulation results of both will be discussed in the following sections.

Implementation details

1. Gaussian blur filter is an image filter in digital signal processing. In this homework, we will implement a 3x3 Gaussian blur filter. The kernel is defined as shown below:

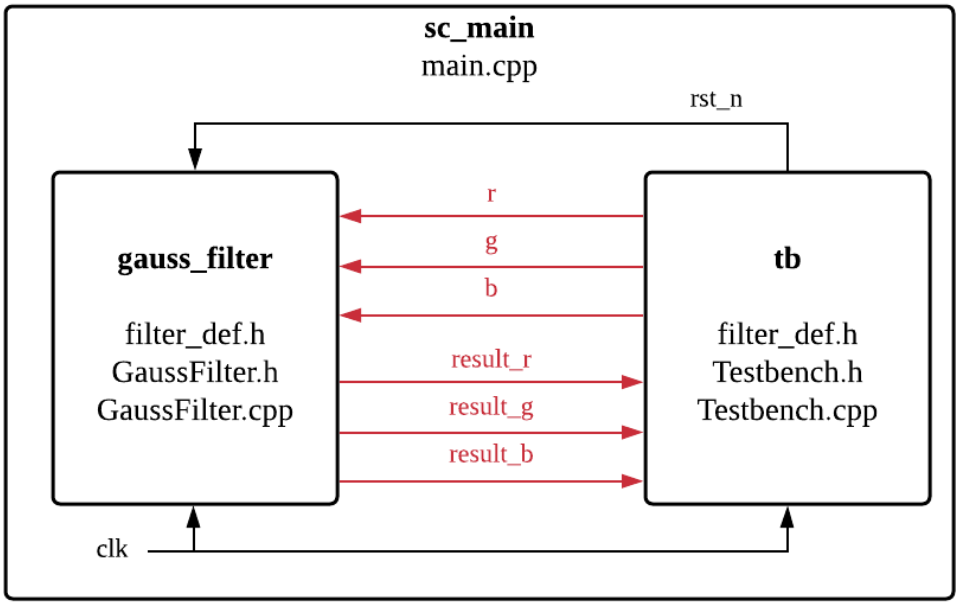
$$\text{kernel} = \begin{bmatrix} 1, & 2, & 1 \\ 2, & 4, & 2, \\ 1, & 2, & 1 \end{bmatrix}$$

After 2-D convolution with the original image (lena.bmp), the Gaussian blur process finishes.

2. C/C++ codes: Given the kernel, the GaussFilter.cpp code does the 2-D convolution with the original image. Also, sine there are three values (r, g, b) in one pixel, the convolution will be done on each value one by one.

$$\text{result}[m,n] = \sum_{j}\sum_{i} \text{original}[i,j] \cdot \text{filter}[m-i, n-j], \qquad 0 \le m,n,i,j \le 2$$

3. SystemC process: After implementing Gaussian blur C codes, I convert it into SystmeC process based on the source codes given in Lab02 (SystmC Process, Events, and Channel). The architecture of the Gaussian blur filter and testbench is shown as follows.

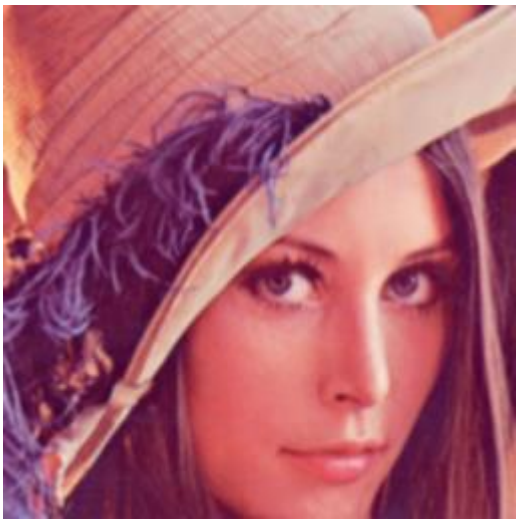


Both gauss_filter and tb modules are instantiated in sc_main and connected to each other based on FIFO (first in, first out) channels. For the file description, the main.cpp define sc_main, two sc_module and the connection between two modules. The filter_def.h defines the dimension of kernel. The GaussFilter.h and Testbench.h define variables and functions used in two modules.

Lastly, the GaussFilter.cpp and Testbench.cpp define function behavior in two modules.

Results

1. lena_std_short.bmp:

Before Gaussian blur	After Gaussian blur
	

2. lena.bmp

Before Gaussian blur	After Gaussian blur
	

Discussion

Since SystemC, a set C/C++ classes, provides an event-driven simulation interface, we are able to simulate concurrent process and software part simultaneously. In this homework, the testbench is written in software whose functions are not affected by events. Also, the Gaussian blur is simulated under concurrent process, so we can know the approximate simulation time in the real design by simply executing C/C++ codes.