UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Object-Oriented Modeling of Electronic Circuits, Spring 1400
Computer Assignment 4, Week 14-16
Abstract Data Handling and Use of Channels

Name:
Date:

In this assignment, you are to design a system that detects edges in a picture. There are two accelerators that communicate via a SystemC channel that you are to develop. The accelerators are a gray scaler and an edge detector.

The system is comprised of two modules. The first one is called *GrayScaler*, and it converts an RGB image to grayscale using the following *average method*. Every pixel of a picture has three colors (Red, Green, and Blue) each specified by an 8-bit word. The grayscale equivalent is the average of the three colors.

$$grayscale = \frac{R + G + B}{3}$$

The second module, named *EdgeDetector*, is responsible for detecting edges in the grayscale image. It convolves the received segment of the image with the following *kernel* to detect the edges.

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

GrayScaler Accelerator

The values for the three RGB pixel colors are written in three separated files. Assume for this problem that the picture size is 512×512. The GrayScaler accelerator reads all pixel values from external image files and stores them in separated arrays. The image read by the GrayScaler is divided into 64 segments each of which consisting of 8 image rows. This accelerator calculates the gray scale values and sends the data to the edge detector.

The gray scaler sends a segment that has been processed to the edge detector. This is done through a channel that connects the two accelerators. While GrayScaler is waiting to a segment sent to the EdgeDetector to be received, it processes the next segment of data. The channel that you develop handles this transfer of data.

Below is a summary of steps taken by the gray scaler.

a. Read {red_pixel, green_pixel, blue_pixel}.txt files and store their values in separate arrays.
b. Greyscale the first segment of the received image based on the *average* method.
c. Send the gray scaled segment to EdgeDetector.
d. If segments are not finished, return to step b.
e. Show the original picture using "illustrate.py."

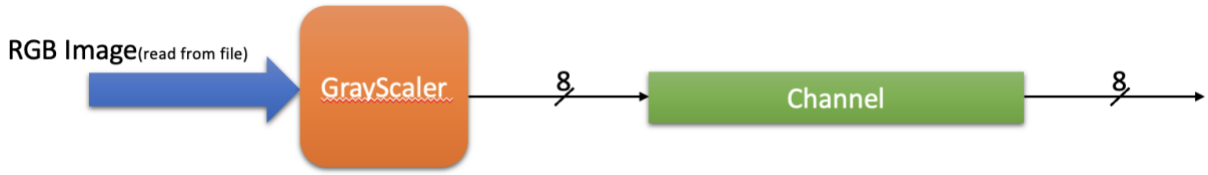

Figure 1 GrayScaler

EdgeDetector Accelerator

After receiving a segment of a picture, this module performs matrix convolution, as depicted in Figure 2. When the processing reaches a row that belongs to the next segment of an image, the edge detector requests another segment from the channel connecting the two accelerators. The data of the previous segment that is no longer needed will be overwritten and processing of edge detection continues. This continues until all data have been processed by the edge detector. The complete data will be saved in the EdgeDetector accelerator and will be dumped to an external file at the end of the processing.

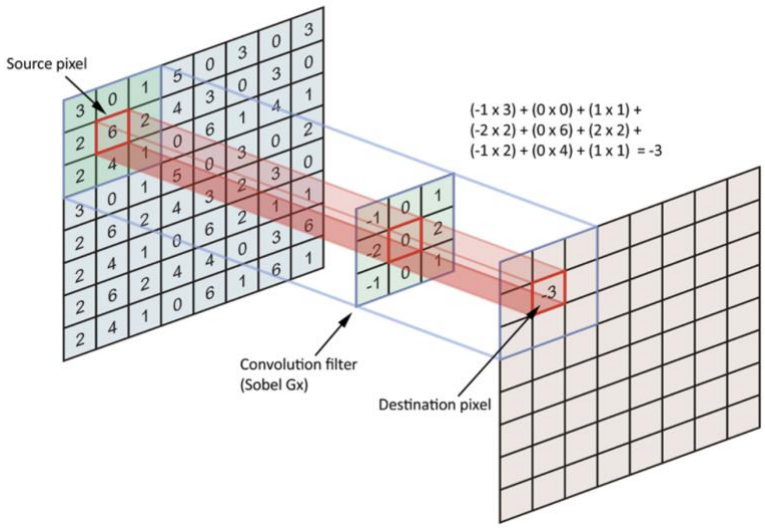

Figure 2 Matrix Convolution

Below is a summary of steps taken by the edge detector.

a. Assume two rows of 0's for the top and bottom, and two columns of 0's for the right and left.
b. Get the first segment.

c. Perform convolution and save the results internally.
d. Continue receiving the segments until all segments have been received.
e. Show the new picture using “read_file_show_img.py.”