University of Tehran

Electrical and Computer Engineering Department

Object Oriented Modeling of Electronic Systems
ECE 342 – Spring 2024
Computer Assignment 5, 6
Due Date: June 3

Description:

In this assignment, you are to implement a module that searches for dedicated patterns in its input pictures, namely Pattern Finder Circuit. This module is constructed from three convolutional blocks with 3*3 kernels. A convolutional block searches for a certain pattern in a 4*4 input image. Convolutional block is followed by a ReLU activation function and a 2*2 Max Pool layer.

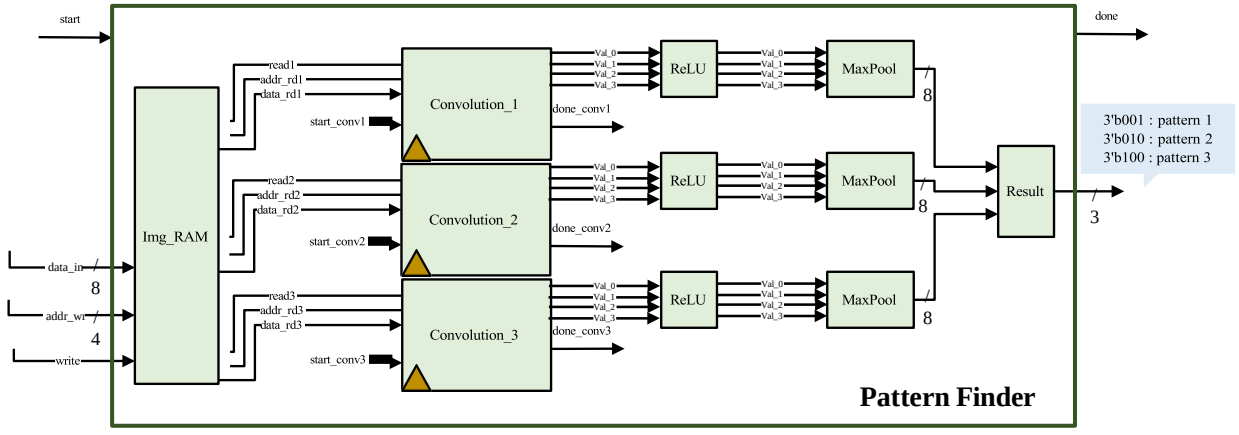

Figure 1. Schematic of the circuit

A convolutional block is a building block used in a convolutional neural network (CNN) for image recognition. It is made up of one or more convolutional layers, that are used to extract features from the input image. The convolutional layers are typically followed by one or more pooling layers, which are used to reduce the spatial dimensions of the feature maps while retaining the most important information.

The basic structure of a convolutional block is as follows:

The input image is passed through a convolutional layer, where filters are applied to the image to extract features such as shapes. The output of this stage sometimes passes through an activation function, such as a ReLU. In Figure 2, you can see the result of applying ReLU on an example matrix. In Figure 3, the functionality of convolution with kernel size of 3 and stride of 1 is illustrated.

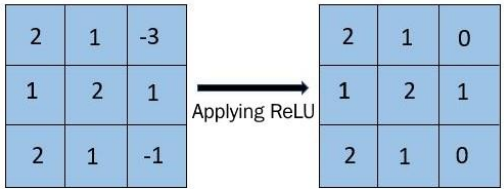

Figure 2. ReLU

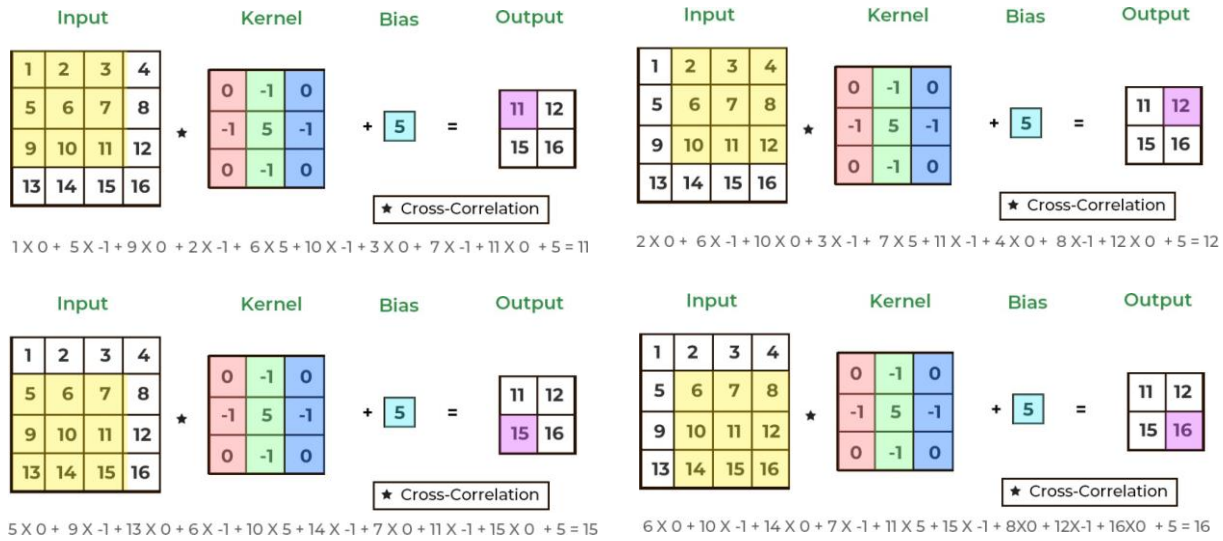

Figure 3. Convolution

The output of the convolutional layers is then passed through a pooling layer, which is used to down-sample the feature maps. This reduces the spatial dimensions of the feature maps, making the network more computationally efficient and reducing the risk of overfitting. Figure 4 shows the result of applying the Max Pooling function with 2*2 stride on a matrix.

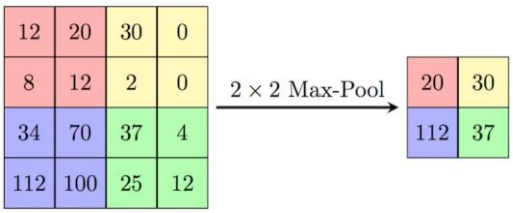

Figure 4. Max Pooling

Convolutional blocks are used in the initial stages of a CNN, where the primary goal is to extract features from the input image. They are typically repeated multiple times in the network, with each block extracting more complex features from the output of the previous block. This hierarchical structure allows the network to learn increasingly complex features, leading to improved image recognition performance.

Design phase:

In Figure 1, you can see the complete schematic of this circuit. You are to write both **bus-functional** model and **RT Level** model **SystemC code** of your design. Each part of it is as described below:

- A RAM block with 16 rows (8-bit) is needed for saving the input image in the beginning of this task. After a complete positive pulse on *start* input signal, convolutional blocks start their computations concurrently. After all computations, *done* output signal will be asserted.

- Three convolutional blocks are needed. Instantiation of this block must be done using for loops and the number of them must be configurable with generic parameters. This convolutional block is a sequential module that uses only one adder and one multiplier. It gets its kernel's values using 9 generic parameters. This module starts its computations after a complete pulse on *start_conv* input signal. Also, *done_conv* output signal must be asserted when the results are ready. The kernels used in this project is as followed:

Pattern 1	Pattern 2	Pattern 3
0 1 0 1 1 1 0 1 0	1 1 1 1 0 0 1 1 1	1 1 1 0 1 0 0 1 0
Bias = -1	Bias = -2	Bias = -2

Figure 5. Patterns

- ReLU block is a combinational block that receives 4 outputs of a convolutional block and puts 4 values on its output.

- A max pooling block is a combinational module that reads the 4 values on its input and puts the largest one on its output.

- A combinational logic, the *result* unit is needed to check the results generated by max pooling blocks. In the output, instead of the largest value, '1' must be asserted and for others '0' must be asserted. This output represents the pattern found by the circuit.

- The input matrices required to be tested is as followed:

0	0	0	0
1	1	1	1
1	0	0	1
1	1	1	1

1	1	1	0
1	0	0	0
1	1	1	0
0	0	0	1

0	1	1	1
0	0	1	0
0	0	1	0
0	0	1	0

0	1	1	1
0	0	1	0
0	0	1	0
0	0	0	0

0	0	1	0
0	1	0	1
0	0	1	0
0	0	0	0

0	0	0	1
0	1	0	0
1	1	1	1
0	1	0	0

Figure 6. Input matrices for testing

Deliverables:

a. All SystemC codes, both bus-functional model (BFM) and RT level (RTL) model in separate files, with appropriate names.
b. A complete report. Your report should include enough design illustration, description, actual data, and output justification. Note that your reports should be well-organized.

Attention:

a. Your report should include adequate design illustration, description, actual data, and output justification. Note that reports should be well organized.
b. For drawing schematics, use Visio or other graphic tools.
c. Make sure you do an independent work and what you submit as your final report includes none but your own work.
d. Compress all files and documents mentioned in the Deliverables section into a zip file and upload on Elearn. The name of the zip file must be in this format with the structure bellow:
"YourFirstNameYourLastName-CA#5,6.zip".