Computer Aided Design

Instructor: Dr.Beitollahi

Homework 3

Topic: Intermediate VHDL (2)

Lectures: 3-4-5-6

Erfan Hemati – Zahra Alizadeh

Release: 1403/2/12

Deadline: 1403/2/21

1) Write a program that converts every 3-bit binary number to the corresponding unary number. (10 points)

- ❖ Example: binary: 101 → unary: 00011111

2) Write a program that counts the number of 1's between the leftmost 1 and the rightmost 1 in a given 16-bit string. (10 points)

- ❖ Example: bit string: 00<u>1</u>0101010111010 → count: 6

3) Do the following operations over a set of given numbers: (6 numbers of 5 bits each) (15 points)

- Find the *Maximum* and *Minimum* number.
- *Sort* the numbers ascendingly.
- Find the number with the *most frequency*.

- ❖ The given bit strings represent ***unsigned*** numbers.

4) Calculate the following functions for the 8-bit input x: (15 points)

- $f(x) = ceil(log_2^x)$
- $g(x) = x \; mod \; 7$

5) Convolution is one of the main operations between square matrices. the first a kernel, and the second a channel, convolution is the process of flipping both the rows and columns of the kernel and multiplying locally similar entries and summing. You have to implement a unit that receives two matrices and performs convolution operations by the kernel on the input channel. The amount of zero-padding should be 0 (don't add any padding to the channel matrix). The implemented unit should be generic and receive the kernel and channel size from the input. The output of the designed unit is the resulting matrix of convolution. The reset signal has an async effect on the execution of the operation. The clock signal is another input of this unit. (25 points)

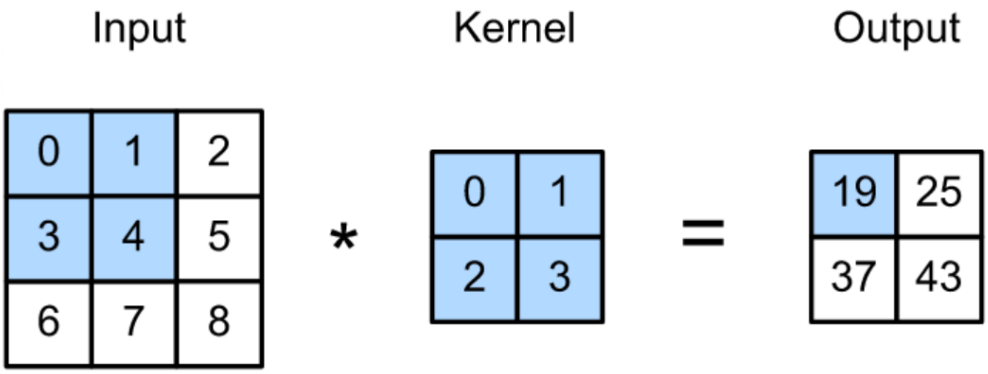
- Tip:

The pixel's values should be 8-bit data.
To receive input matrices, you can receive them as a one-dimensional vector. For example, if the input kernel is a 3x3 matrix, you can get a std_logic_vector that has 72 bits. That is, 9 8-bit data, and then in the designed architecture, use a type that represents a square matrix and stores the data of the matrix fields in A.

For example:

```
type matrix_array is array (0 to MATRIX_SIZE-1, 0 to MATRIX_SIZE-1) of unsigned(7 downto 0);
signal my_matrix: matrix_array;
```

An example of Matrix convolution:



As you can see the pixels of the kernel are multiplied one by one with corresponding pixels on the channel matrix :

(0 x 0) + (1 x 1) + (3 x 2) + (4 x 3) = 19

6) Design a unit that receives one 8-bit number serially. The designed unit contains only one single-bit input signal that receives data bit by bit and a clock which it uses to read data at the right time. Now, for the receiver to understand when the input 8-bit number data starts, a start bit should be considered for the data, which is always 0. This means that the receiver must wait for a falling edge on your data input signal when it sees the start bit for the first time, it must read each data in the middle of each bit, and when it reads 8 data bits, it must wait for the stop bit, which is equal to 1. (25 points)

The location of the green arrows is when the data should be read, which is the rising edge of the clock.

- Tip:

Before seeing the start bit, the data on the input signal must be 1 so that the receiver notices the start of new data.
After fully receiving 1 byte of the desired data, the output should be produced as follows:

1 < number of 1's in binary > 0 <number of 0's in binary>

- The output is 10 bits long, since you need 4 bits to show the number of 1's and 4 other bits to show the number of 0's.

and the data is output in parallel.

Your test bench is an important part of this question and will be evaluated.