**Team 8 Hardware Documentation**

**Important Note:**

This project uses fixed point as agreed with Eng Abotaleb

Numbers are represented as 16 bits (1 for sign, 3 for integer, 12 for decimal)

**For Example:**

16’h0800 = 0.5

16’hF800 = - 0.5

**Parallel convolution with many filters**

Number of cycles = No of input images

This module perform convolution process on a certain number of images using certain number of filters

**As Example**

Image:

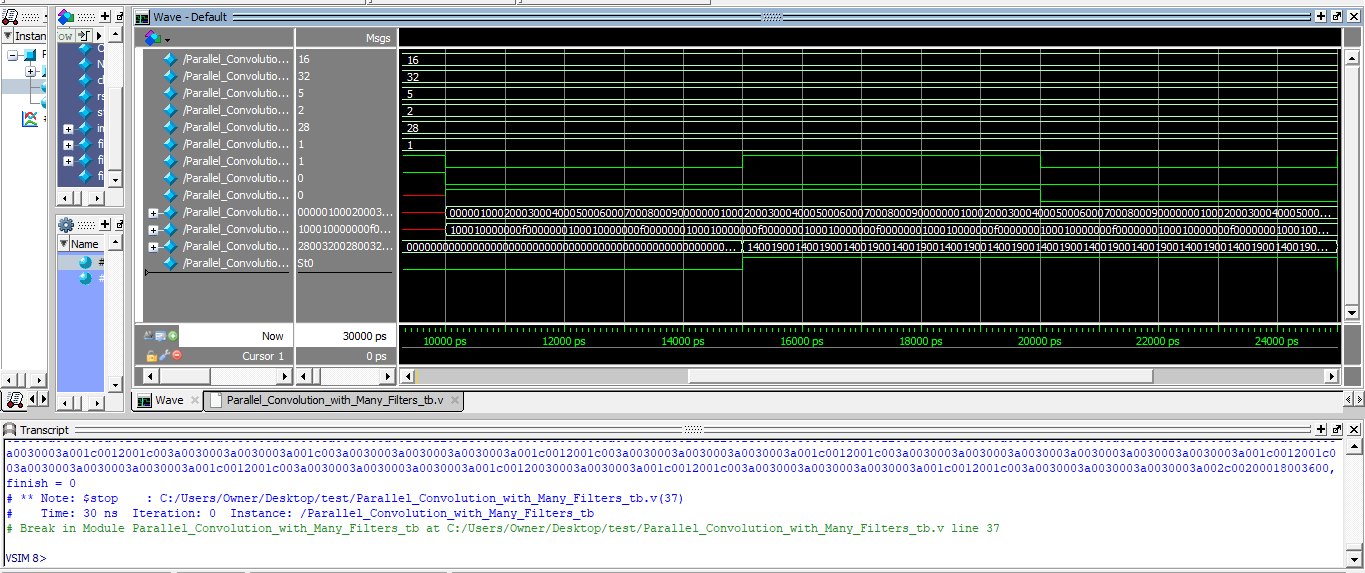
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 = 0 | 0100 = 0.0625 | 0200 = 0.125 | 0300 = 0.1875 | 0400 = 0.25 |
| 0.125 | 0.1875 | 0.25 | 0500 = 0.3125 | 0600 = 0.375 |
| 0.25 | 0.3125 | 0.375 | 0700 = 0.4375 | 0800 = 0.5 |
| 0.375 | 0.4375 | 0.5 | 0900 = 0.5625 | 0 |
| 0.5 | 0.5625 | 0 | 0.0625 | 0.125 |

Filters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1000 = 1 | 0000 = 0 | F000 = -1 | 0000 = 0 | 1000 = 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |

Output = 1.25 = 1400

**Output is correct**



**Tanh activation function**

Number of cycles = 1 cycle

This module gets tanh the input

**As Example**

0100 = 0.0625 in decimal

tanh(0.0625) = 0.06242 approx = 0.06245 , so output is 0100

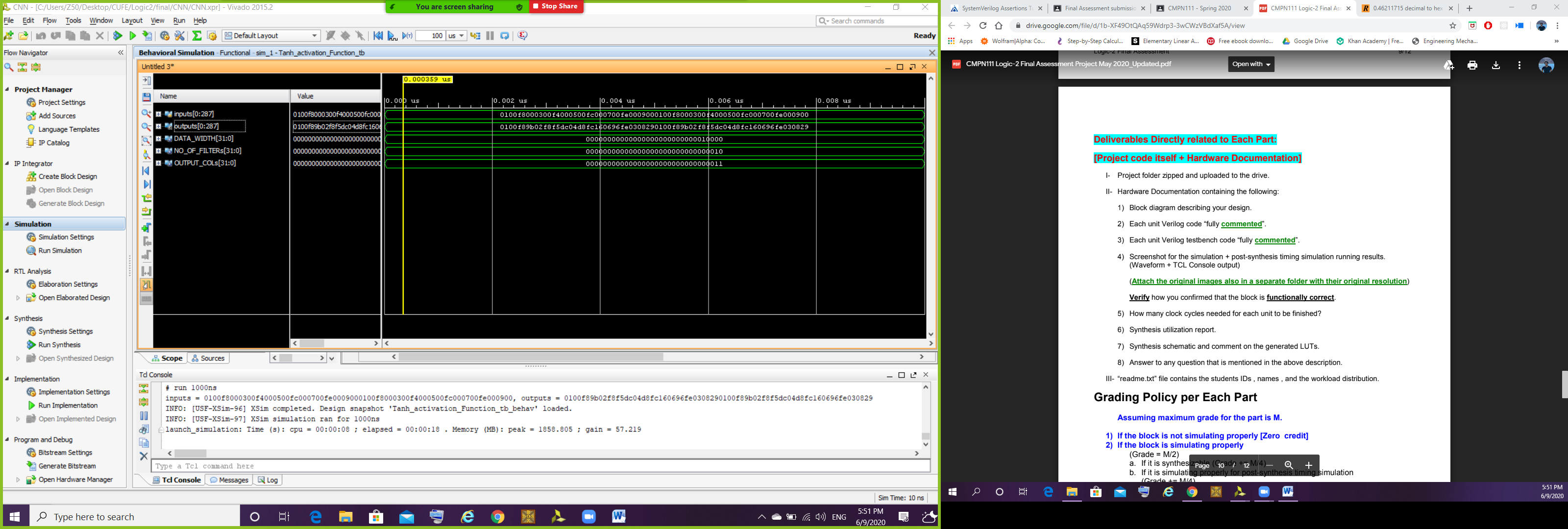
f800 = -0.5 in decimal

tanh(-0.5) = - 0.46212 , so output is approx f89b

inputs = 0100f8000300f4000500fc000700fe0009000100f8000300f4000500fc000700fe000900

outputs = 0100f89b02f8f5dc04d8fc160696fe0308290100f89b02f8f5dc04d8fc160696fe030829

**Output is correct**



**Softmax activation function**

Number of cycles = 1 cycle

This module gets exponential the number (we had permission to neglect dividing the exponential by the total summation)

**As Example**

0100 = 0.0625 in decimal

e^0.0625 = 1.06449 = 1108

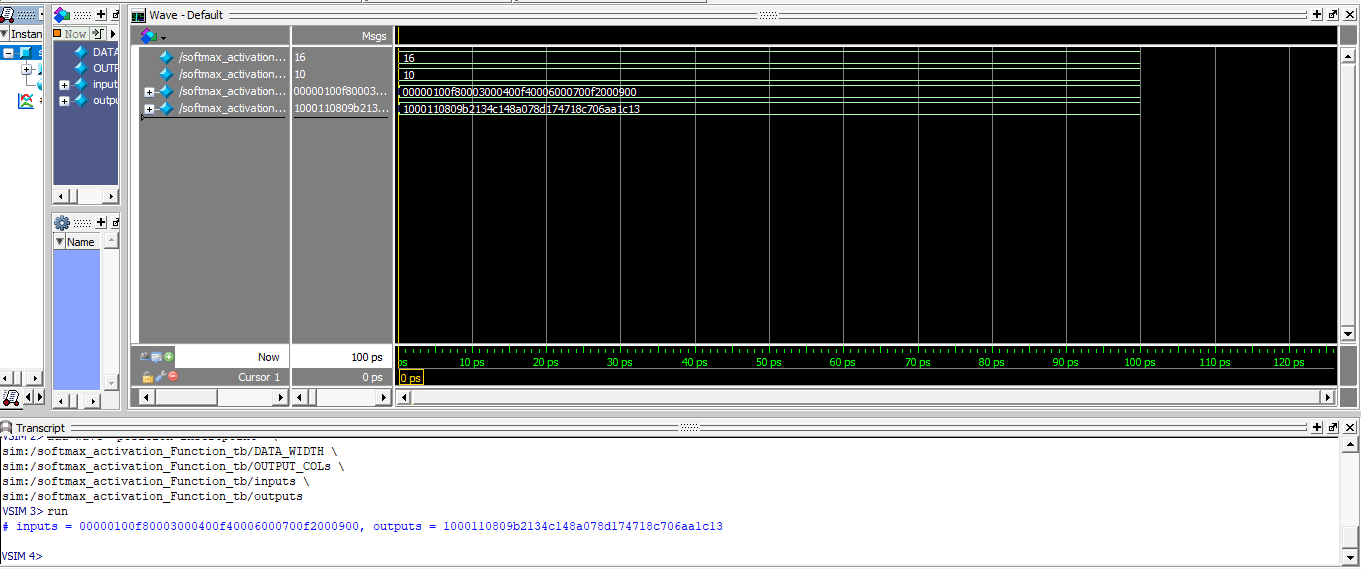
F800 = -0.5 in decimal

e^-0.5 = 0.6065306  = 0.9b4

inputs =   0000 0100 f800 0300 0400 f400 0600 0700 f200 0900

outputs = 1000 1108 09b2 134c 148a 078d 1747 18c7 06aa 1c13

**Output is correct**



**Average pool layer**

Number of cycles = 1 cycle

This module gets the average of 4 pixels

**As Example**

Input: Image 10\*10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0100= 0.0625 | F800= -0.5 | 0300= 0.1875 | F400= -0.75 | 0500= 0.3125 | FC00= -0.25 | 0700=0.4375 | FE00= -0.125 | 0900= 0.5625 |
| 0000 | 0100 | F800 | 0300 | F400 | 0500 | FC00 | 0700 | FE00 | 0900 |

.

.

. repeated 4 times

.

Average of 4 pixels in each box

Output: Image 5\*5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.03125 = 0080 | - 0.15925 = FD80 | -0.21875 = FC80 | 0.09375 = 0180 | 0.21875 = 0380 |

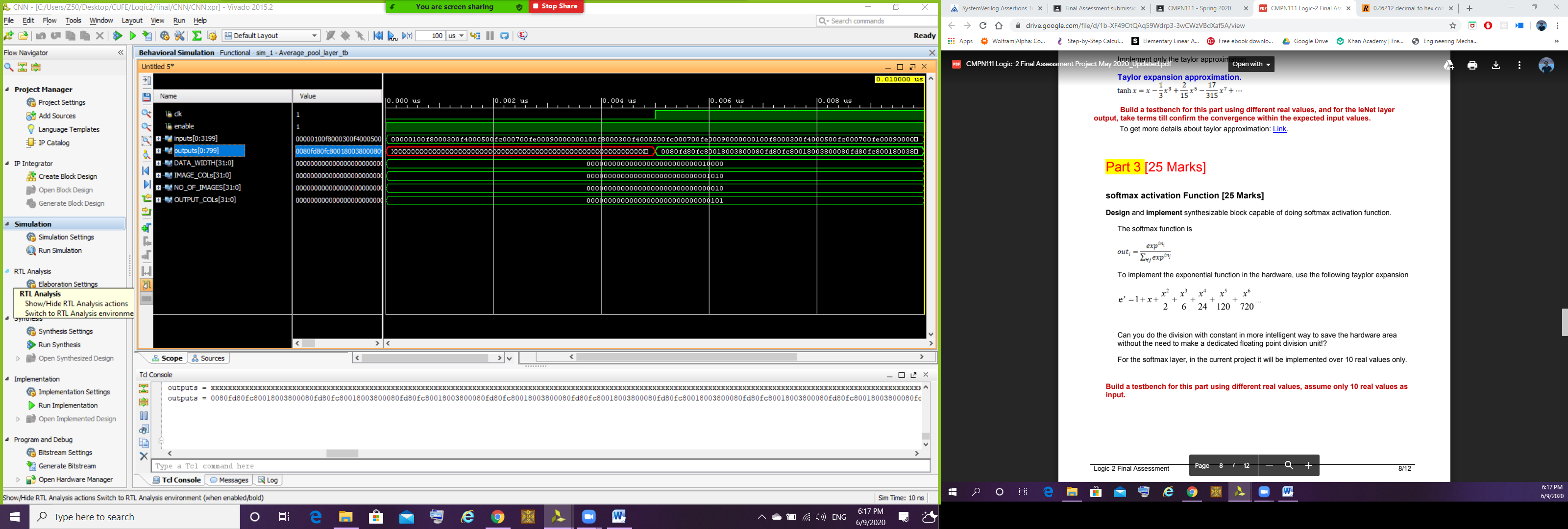
.

.

. repeated 4 times

.

**Output is correct**



**CNN\_Main**

Number of cycles = 5 + Number of filters of first Conv + Number of filters of second Conv + Number of filters of third Conv + Number of Weights of first FC

As agreed with Eng Abotaleb we will make a small verification using small weight files and different number of filters

Number of cycles in testbench = 17

Image 32\*32 :

Each row contains 32 pixels “0200” = 0.125

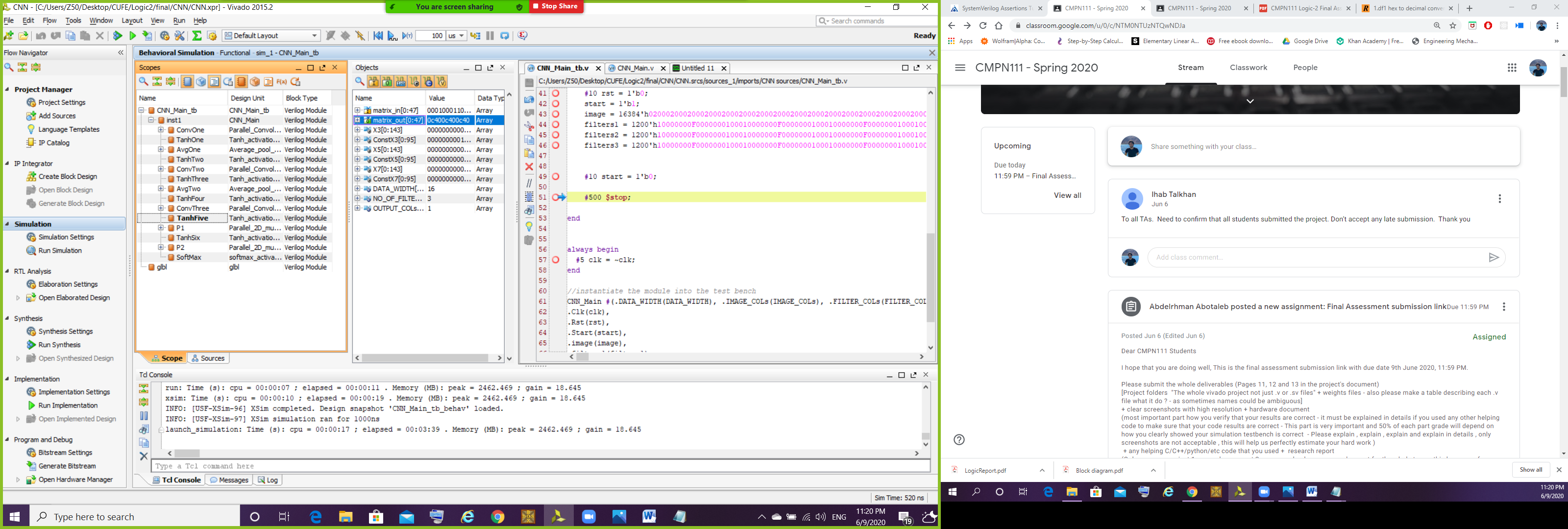
And Each convolution has 3 filters each like the following

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |
| 1 | 0 | -1 | 0 | 1 |

Convolution, tanh, softmax modules are tested and verified before

The following is the input of the first FC (0C40 0C40 0C40) which is (0.765625 0.765625 0.765625)

FC is working properly from phase 1



Output of the CNN = 0b2a

