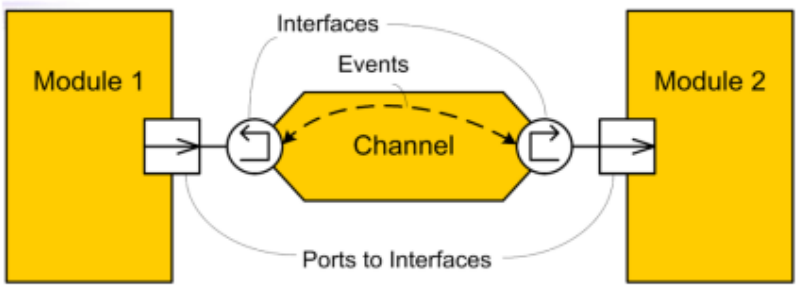
Machine Learning Intelligent Chip Design

Homework1 Implementation of AlexNet in SystemC

Description

Implement the AlexNet CNN architecture using SystemC. The model consists of convolutional layers, max-pooling layers, and fully connected layers, following its original design for image classification tasks.

In SystemC, **interface** is an abstract class that inherits from sc_interface. Interface can be used to define communication protocols between different modules. It only describes the communication methods and protocols and does not involve specific data transmission. A SystemC **channel** is a class that implements one or more SystemC interface classes and inherits from either sc_channel or sc_prim_channel. The following figure shows some primitive channels and interfaces defined by SystemC library:



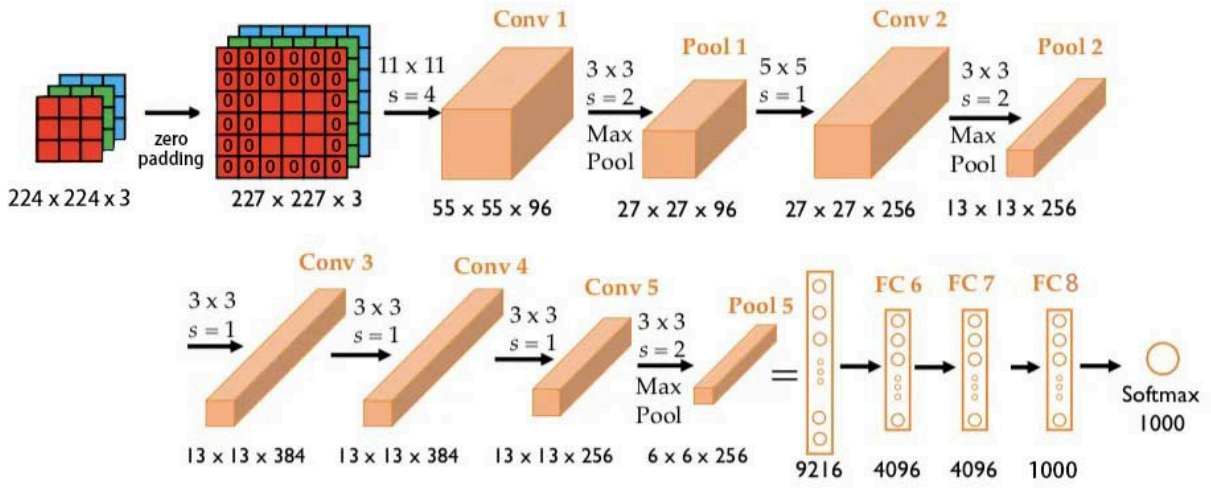
- Channel
  - sc_signal<T>
  - sc_signal_resolved
  - sc_signal_rv<W>
  - sc_buffer<T>
  - sc_fifo<T>
  - sc_mutex
  - sc_semaphore
- Interface
  - sc_fifo_in_if
  - sc_fifo_out_if
  - sc_mutex_if
  - sc_semaphore_if
  - sc_signal_in_if
  - sc_signal_out_if

Implementation Details

The purpose of this homework assignment is to givean opportunity to practice using **sc_signal**, **sc_buffer**, and **sc_fifo** to establish a communication mechanism between different modules. You are required to divide the AlexNet model implemented in HW1 into several sub-modules (not necessarily distinguished by layer) and then connect them using these three SystemC channels. The result of the model's execution should be identical to that of hw1.

Alexnet Training Model

The AlexNet model is shown in the diagram below. Before the input image enters the first convolutional layer, **zero padding** should be applied: **two rows on the top and left, and one row on the bottom and right**. Additionally, since the results need to be displayed as probabilities, please apply softmax at the final stage.



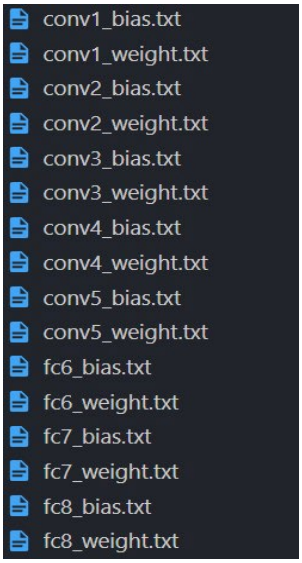
The Pre-trained AlexNet Model Information

Layer	Type	Description	Output Dimension
0	Input Layer	Zero Padding 224 to 227	227x227x3
1	Convolutional Layer 1 + ReLU	64 kernels of 11x11, stride 4	55x55x64
1b	Max Pooling 1	3x3, stride 2	27x27x64
2	Convolutional Layer 2 + ReLU	192 kernels of 5x5, stride 1, padding 2	27x27x192
2b	Max Pooling 2	3x3, stride 2	13x13x192
3	Convolutional Layer 3 + ReLU	384 kernels of 3x3, stride 1, padding 1	13x13x384
4	Convolutional Layer 4 + ReLU	256 kernels of 3x3, stride 1, padding 1	13x13x256
5	Convolutional Layer 5 + ReLU	256 kernels of 3x3, stride 1, padding 1	13x13x256
5b	Max Pooling 5	3x3, stride 2	6x6x256
6	Fully Connected Layer 6 + ReLU	4096 neurons	4096
7	Fully Connected Layer 7 + ReLU	4096 neurons	4096
8	Fully Connected Layer 8	1000 neurons	1000
9	Softmax Layer	Converts logits to probabilities	1000

Provided Data Description

Values in the pre-train model in Pytorch are floating points with 16 digits after the decimal. We export these values as txt file for you. Values in these txt files are floating point but rounded to the sixth decimal place.

- **Model layer parameters**



- **imagenet_classes.txt**
  https://gist.github.com/ageitgey/4e1342c10a71981d0b491e1b8227328b

Input image and weight reading format

In this assignment, you need to read the image and weight txt files provided by the TA to perform calculations. The reading method follows the **raster scan order**, as illustrated in the diagram below.

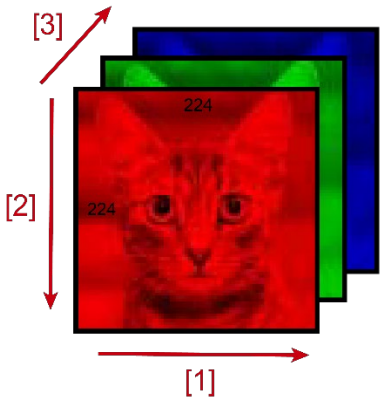
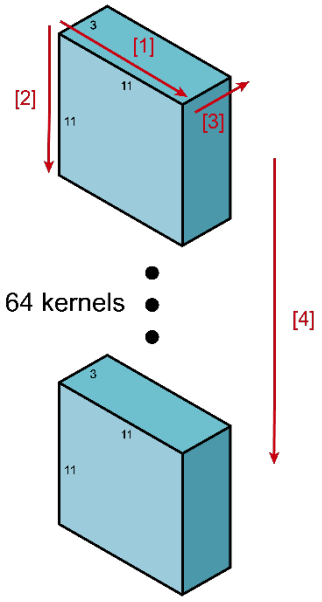


Image reading format



Weight reading format

Output Result Format

- **Simulation results of AlexNet executed in Python**
  This part is for reference only; you do not need to run Python yourself.
  Dog

```
In [20]: runfile('C:/Users/micha/Desktop/use_dataset_alexnet_model_instance.py', wdir='C:/Users/micha/Desktop')
Predicted class: golden retriever
```

Cat

```
In [9]: runfile('C:/Users/micha/Desktop/use_dataset_alexnet_model_instance.py', wdir='C:/Users/micha/Desktop')
Predicted class: Egyptian cat
```

- **Simulation results of AlexNet executed in SystemC**

Dog:

```
Top 100 classes:
================================================
  idx |      val | possibility | class name
------------------------------------------------
  207 |    16.59 |       38.63 | golden retriever
  175 |    15.57 |       13.86 | otterhound
  220 |    15.36 |       11.26 | Sussex spaniel
  163 |    15.00 |        7.86 | bloodhound
  219 |    14.59 |        5.22 | cocker spaniel
```

Cat:

```
Top 100 classes:
================================================
  idx |      val | possibility | class name
------------------------------------------------
  285 |    20.21 |       96.38 | Egyptian cat
  281 |    16.14 |        1.65 | tabby
  282 |    15.73 |        1.10 | tiger cat
  287 |    14.79 |        0.43 | lynx
  728 |    14.41 |        0.29 | plastic bag
  330 |    12.73 |        0.05 | wood rabbit
```

The example output above only displays the top 5 labels with the highest probability. However, in this assignment, you need to display the **top 100 labels with the highest probability**, and the **output format and layout must be identical to the example provided by the TA**. For example, both output values of the val and possibility should always **display two decimal places, regardless of the number of digits in the integer part.**

Below, we provide formatting code for reference. Please note that this code is for reference only. If you choose to use it, make sure to modify variable names and other details as needed.

```cpp
cout << fixed << setprecision(2);
cout << "Top 100 classes:" << endl;
cout << "================================================" << endl;
cout << right << setw(5) << "idx"
     << " | " << setw(8) << "val"
     << " | " << setw(11) << "possibility"
     << " | " << "class name" << endl;
cout << "------------------------------------------------" << endl;

for (int i = 0; i < 100; i++) {
    file.clear(); // Clear any potential error flags
    file.seekg(0, ios::beg); // Seek back to the beginning of the file
    int index = top_5_val[i].second;
    string line;
    for (int j = 0; j <= index; j++) {
        getline(file, line);
    }

    cout << right << setw(5) << index
         << " | " << setw(8) << top_5_val[i].first
         << " | " << setw(11) << (top_5_pos[i].first) // Assuming softmax outputs probabilities
         << " | " << line << endl;

}
cout << "================================================" << endl;
```

Implement Notes

- **Module Boundaries:**
  Ensure that each submodule has clearly defined boundaries and that any communication and interaction between them is clearly specified. This helps reduce coupling between modules, making the code easier to understand and debug.
- **Module functionality**
  Each submodule should have specific functionality, and these functionality should be as independent and reusable as possible. This makes the code more modular and individual modules can be easily replaced or modified when needed without affecting other modules.
- **Testing and Validation**
  Each submodule is unit tested to ensure it functions properly and as expected. In addition, integration testing should be performed on the entire system to verify the interfaces and interactions between modules.

Submission Guidelines & Grading Policy

- **The grading breakdown for this assignment is as follows:**
  - **Report : 40%**
    - Simulation results demonstrate the predicted output for the provided input data.
    - Your implementation approach, challenges faced, and any observations or insights gained during the implementation and simulation process.
    - Observe the differences among these three channels and your analysis of them.
    - If the code is not submitted, **no points will be awarded for the report!**
  - **Simulation Result : 60%**
    - You need to complete the AlexNet architecture using these three channels—**signal, buffer**, and **fifo**—within their respective folders. During the demo, we will **check whether you have used them. If not, you will not receive any points.**
    - We will separately demo the three versions of the assignment. The test data will be the same as in HW1, consisting of two test cases per version, **each worth 10 points**, for a **total of 60 points**. The simulation results must **display the correct output on the terminal, and the formatting must match the provided example exactly.**
- **For the code submission, please use the compression and submission command provided by the TA.**
- **Please submit the report file to the new E3. The name of your report file is report_mlchipXXX.pdf (XXX is your account ID). If the file violates the naming rule and the file format, 10 points will be deducted.**
- **Ensure that your code is well-commented and organized for clarity and understanding.**
- **Plagiarism is forbidden, otherwise you will get 0 point!!!**
- **You can't modified Makefile TA's provided, otherwise you will get 0 point !!!.**

Submission & demo command

- **Please use make cat、make dog commands to execute your SystemC code. These commands correspond to running cat.txt and dog.txt, respectively. This means you need to modify how input files are handled by passing "dog" and "cat" as parameters to read the corresponding image files. We will provide a main function example and a parameter-passing example, as below.**

```cpp
int sc_main(int argc, char* argv[]) {
    sc_clock clk("clk", 1, SC_NS);
    sc_signal<bool> reset;

    if (argc != 2) {  // Ensure exactly one filename is provided
        std::cerr << "Usage: " << argv[0] << " <file>" << std::endl;
        return 1;
    }

    std::string file = argv[1];  // Get the filename from the argument
    top->input_layer->load_data_and_pad("./data/" + file);
    return 0;
}
```

main function example

```cpp
void load_data_and_pad(const string& input_file_path){
    ifstream input_file(input_file_path.c_str());
}
```

parameter-passing example

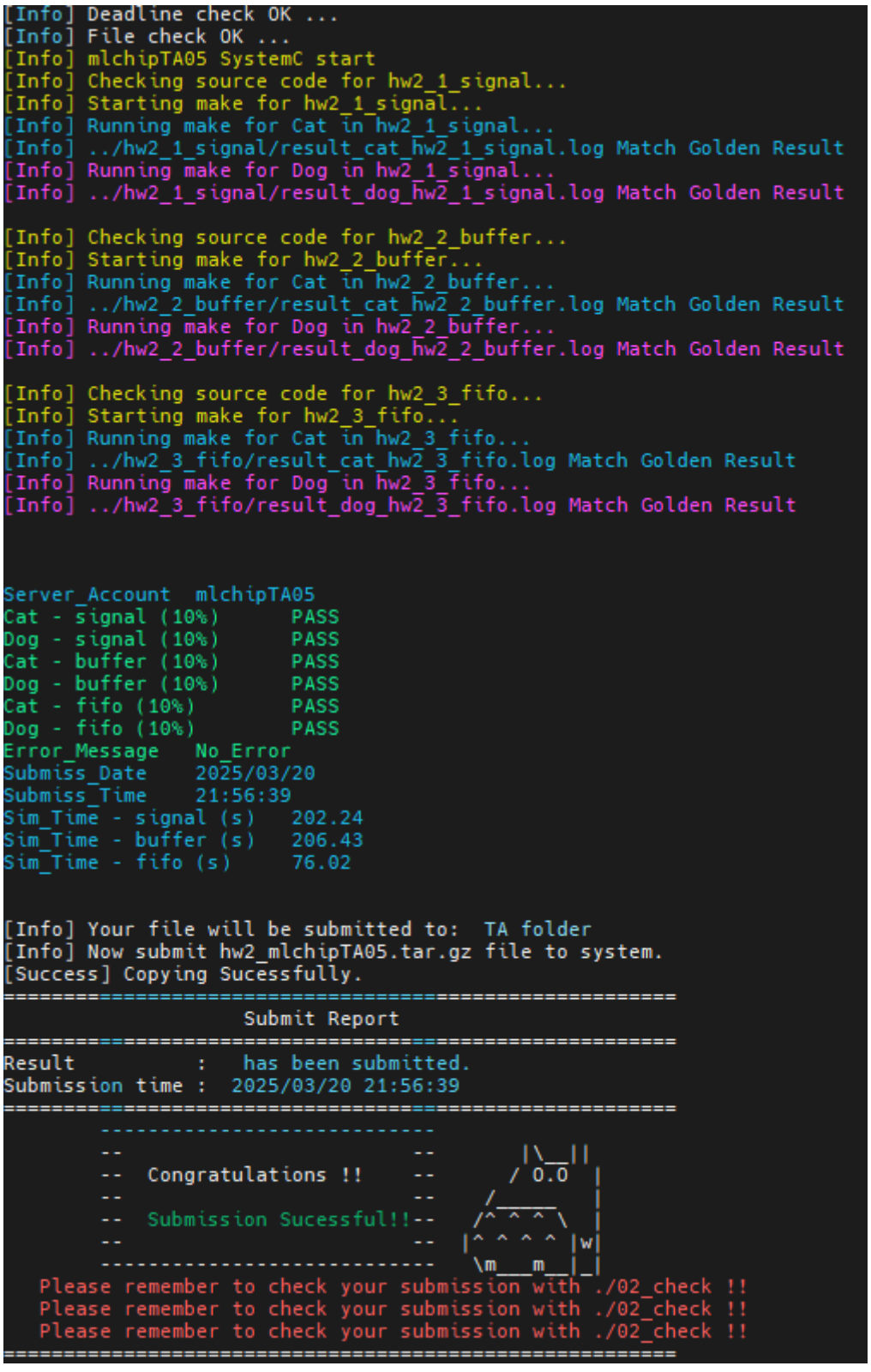
- **Follow following command in 09_SUBMIT folder to submit your code and demo.**

1. **./00_tar**

```
[Info] Top folder created: hw2_mlchipTA05
[Info] Directory created: hw2_mlchipTA05/hw2_1_signal_mlchipTA05
[Info] Copied: ../hw2_1_signal/alexnet.cpp -> hw2_mlchipTA05/hw2_1_signal_mlchipTA05/alexnet_signal_mlchipTA05.cpp
[Info] Directory created: hw2_mlchipTA05/hw2_2_buffer_mlchipTA05
[Info] Copied: ../hw2_2_buffer/alexnet.cpp -> hw2_mlchipTA05/hw2_2_buffer_mlchipTA05/alexnet_buffer_mlchipTA05.cpp
[Info] Directory created: hw2_mlchipTA05/hw2_3_fifo_mlchipTA05
[Info] Copied: ../hw2_3_fifo/alexnet.cpp -> hw2_mlchipTA05/hw2_3_fifo_mlchipTA05/alexnet_fifo_mlchipTA05.cpp
[Info] Creating tar archive: hw2_mlchipTA05.tar.gz
hw2_mlchipTA05/
hw2_mlchipTA05/hw2_1_signal_mlchipTA05/
hw2_mlchipTA05/hw2_1_signal_mlchipTA05/alexnet_signal_mlchipTA05.cpp
hw2_mlchipTA05/hw2_2_buffer_mlchipTA05/
hw2_mlchipTA05/hw2_2_buffer_mlchipTA05/alexnet_buffer_mlchipTA05.cpp
hw2_mlchipTA05/hw2_3_fifo_mlchipTA05/
hw2_mlchipTA05/hw2_3_fifo_mlchipTA05/alexnet_fifo_mlchipTA05.cpp
[Success] hw2_mlchipTA05.tar.gz created successfully.
```

2. **./01_submit**

- **We will run make dog、make cat command to verify the correctness of your assignments. Simulation examples of correct and incorrect results are shown below.**



Correct example

```
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] mlchipTA05 SystemC start
[Info] Checking source code for hw2_1_signal...
[Error] No sc_signal connection in ../hw2_1_signal
[Info] Checking source code for hw2_2_buffer...
[Error] No sc_buffer connection in ../hw2_2_buffer
[Info] Checking source code for hw2_3_fifo...
[Info] Starting make for hw2_3_fifo...
[Info] Running make for Cat in hw2_3_fifo...
[Info] ../hw2_3_fifo/result_cat_hw2_3_fifo.log Match Golden Result
[Info] Running make for Dog in hw2_3_fifo...
[Info] ../hw2_3_fifo/result_dog_hw2_3_fifo.log Match Golden Result



Server_Account  mlchipTA05
Cat - signal (10%)      FAIL
Dog - signal (10%)      FAIL
Cat - buffer (10%)      FAIL
Dog - buffer (10%)      FAIL
Cat - fifo (10%)        PASS
Dog - fifo (10%)        PASS
Error_Message   No sc_signal connection in hw2_1_signal, No sc_buffer connection in hw2_2_buffer
Submiss_Date    2025/03/20
Submiss_Time    22:17:40
Sim_Time - signal (s)   NaN
Sim_Time - buffer (s)   NaN
Sim_Time - fifo (s)     76.26


[Info] Your file will be submitted to:  TA folder
[Warning]  demo has been submitted.
[Warning]  It will overwrite your original file.
[Info] Now submit hw2_mlchipTA05.tar.gz file to system.
[Success] Copying Sucessfully.
============================================================
                    Submit Report
============================================================
Result          :   has been submitted.
Submission time :  2025/03/20 22:17:40
============================================================
        ----------------------------
        --                        --       |\__||
        --  Congratulations !!    --      / O.O  |
        --                        --    /_____   |
        --  Submission Sucessful!!--   /^ ^ ^ \  |
        --                        --  |^ ^ ^ ^ |w|
        ----------------------------   \m___m__|_|
   Please remember to check your submission with ./02_check !!
   Please remember to check your submission with ./02_check !!
   Please remember to check your submission with ./02_check !!
============================================================
```

Incorrect example

3. **./02_check**

```
hw2_mlchipTA05.tar.gz  has been downloaded!
demo_result_hw2_mlchipTA05.csv  has been downloaded!
```