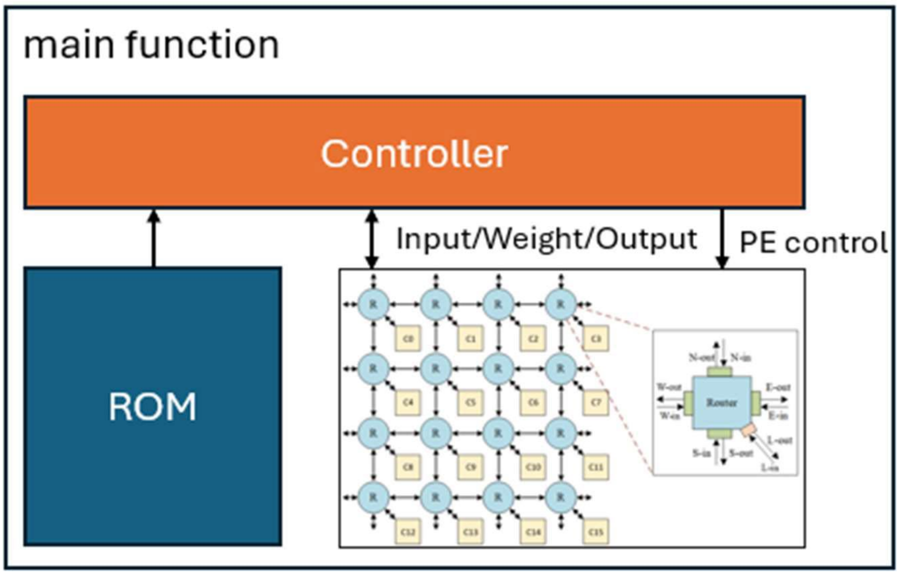
Machine Learning Intelligent Chip Design

Homework4 NoC-based CNN architecture

Description

In HW1, you build an Alexnet model by SystemC. In HW2, you can partition your neural network model by channel. In HW3, you use the 4x4 mesh-based NoC architecture to transfer packets from PE to PE. In this HW4, you should replace the PEs in HW3 with neurons from HW2 or HW1 and integrate them into the NoC, therefore, you can build a NoC-based CNN architecture.

Implementation Details



- **PE:**
  You should place the partition NN layer on each PE and implement a register in the PE to record the destination of the packet. The destination can be defined and assigned in your own way.
- **Controller:**
  The controller will initialize the result destination PE for each packet. It reads the ROM data and transfers it to the PE. After the calculation is complete, the controller prints the results.
- **ROM:**
  This module will read the input data, weight, and bias of each layer. The ROM will be provided by the TA. You may refer to the comments in the .h file or the content in the .cpp file to understand how to access the data. **However, please note that modifying either ROM.h or ROM.cpp is strictly prohibited. Any changes that result in a demo failure will lead to an automatic score of zero.**

- **Router:**

  You should connect the router to the controller with the same communicate protocol in the NoC. The controller will config the PE control information by this way.

Implement Notes

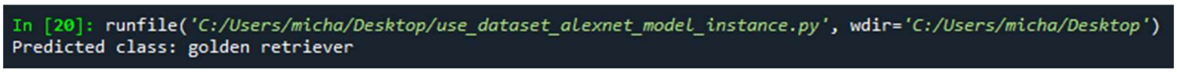
There are 16 PEs and routers in HW3. Not all PEs and routers should be used with this homework. If your model layer partitions don't require as many resources, you can leave some PEs and routers unused.

Output Result Format

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

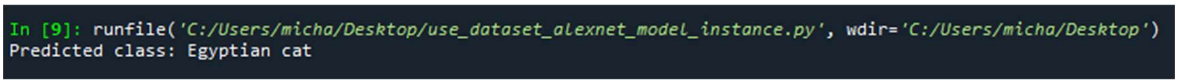
  This part is for reference only; you do not need to run Python yourself.

  Dog



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

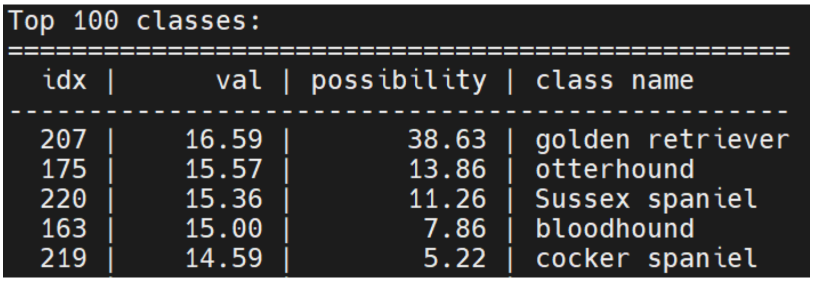
  Cat



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

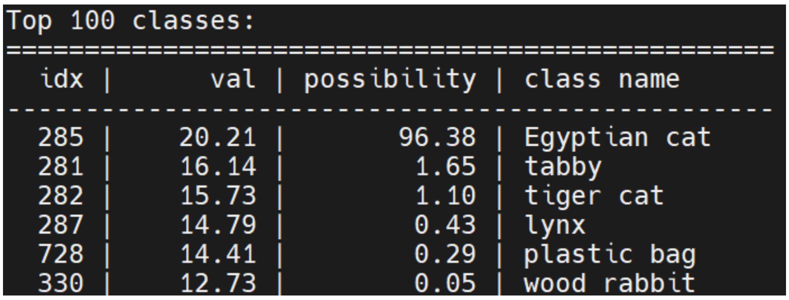
- **Simulation results of AlexNet executed in SystemC**

  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
```

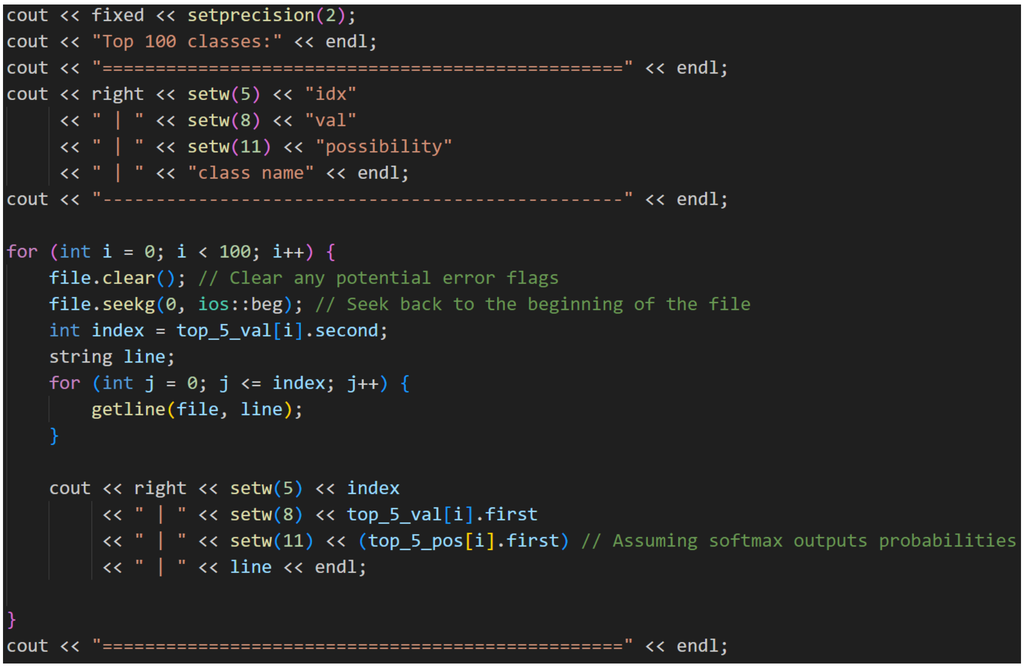
  Cat



```
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.



```
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;
```

Submission Guidelines & Grading Policy

- **The grading breakdown for this assignment is as follows:**
  - **Report: 30%**
    - Simulation results demonstrate the predicted output for the provided input data.
    - How do you design the router and NI? What routing algorithm do you use? What is the depth of the buffer? Do you use virtual channels?
    - Your implementation approach, challenges faced, and any observations or insights gained during the implementation and simulation process.
    - If the code is not submitted, **no points will be awarded for the report!**
  - **Simulation Result: 70%**
    - You need to complete a NoC-based CNN architecture similar to Figure 1. After executing the 'make cat' and 'make dog' commands, the terminal must **display the correct output on the terminal, and the formatting must match the provided example exactly.**

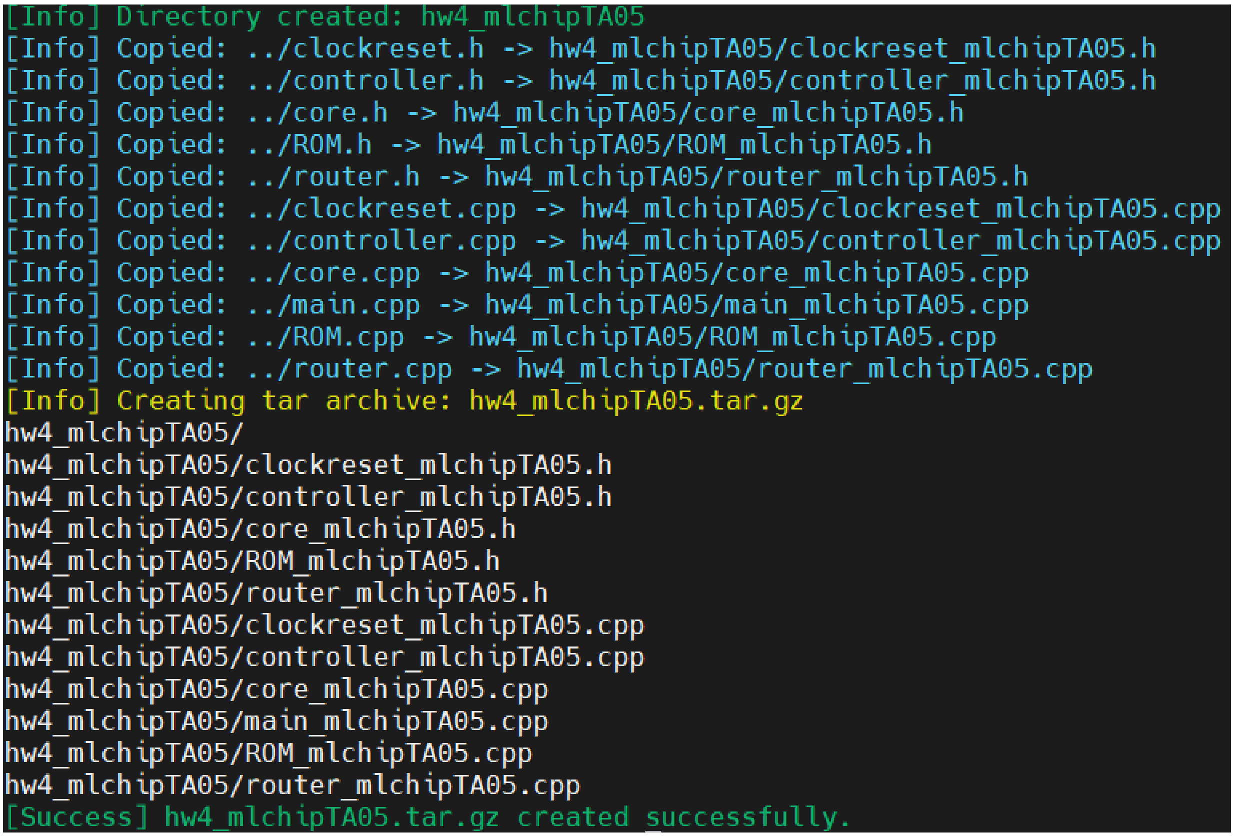
- **For the code submission, please use the compression and submission command provided by the TA.**
- **Please submit the report file to the new E3. The name of your report file is report_mlchipXXX.pdf (XXX is your account ID). If the file violates the naming rule and the file format, 10 points will be deducted.**
- **Ensure that your code is well-commented and organized for clarity and understanding.**

- **The following are rules that must be absolutely adhered to. Any violation will result in 0 points.**
    - **Plagiarism is forbidden.**
    - **You cannot modify the provided Makefile, ROM files.**
    - **You cannot use any Chinese characters in the assignment, including within comments.**
    - **You are not allowed to read image files (e.g., cat.txt or dog.txt), bias files, or weight files directly in your own design. All such data must be accessed exclusively through the ROM provided by the TA.**

Submission & demo command

- **Please use the make cat 、 make dog commands to execute your SystemC code.**
- **Follow the following command in the 09_SUBMIT folder to submit your code and demo.**

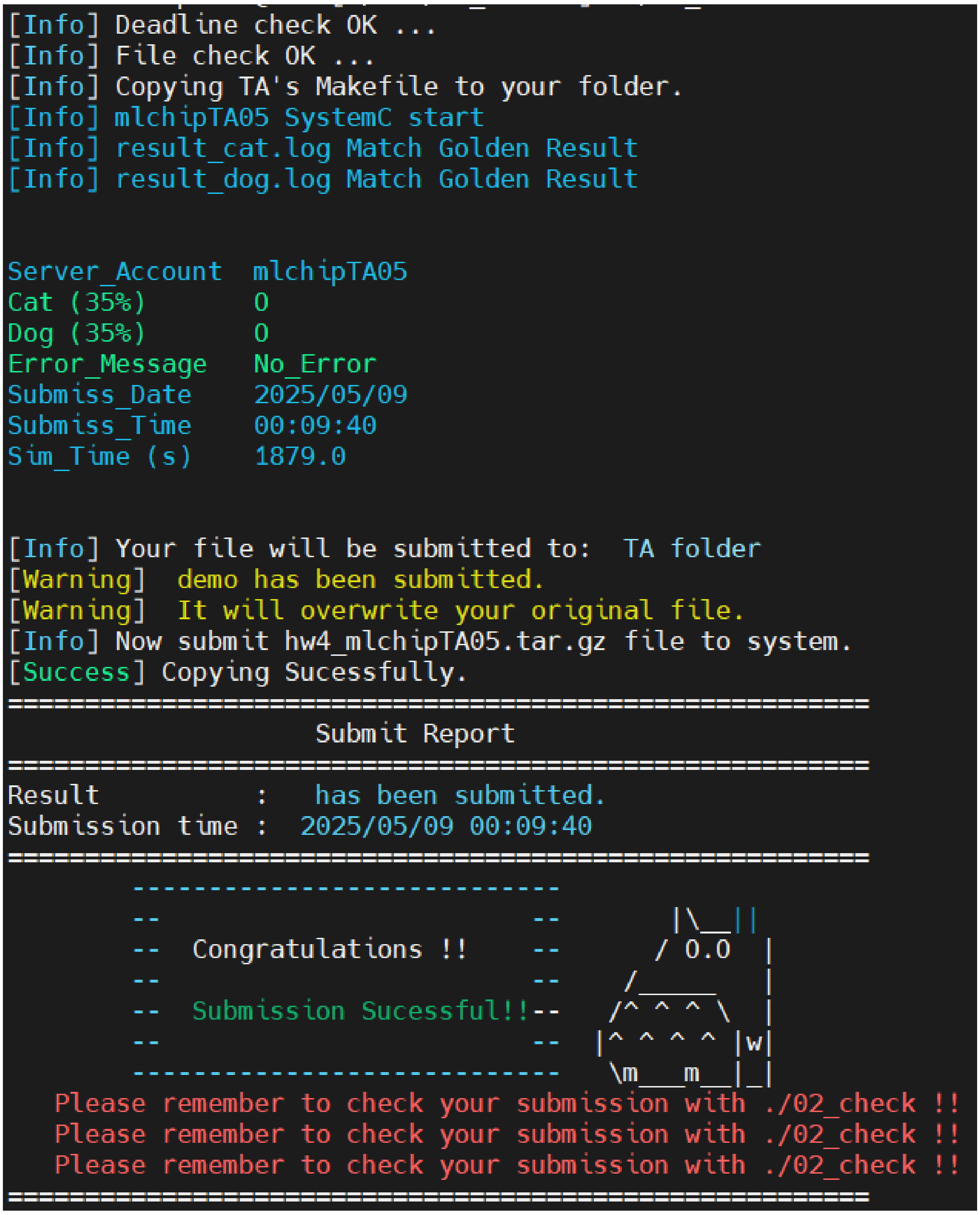
1. **./00_tar**



```
[Info] Directory created: hw4_mlchipTA05
[Info] Copied: ../clockreset.h -> hw4_mlchipTA05/clockreset_mlchipTA05.h
[Info] Copied: ../controller.h -> hw4_mlchipTA05/controller_mlchipTA05.h
[Info] Copied: ../core.h -> hw4_mlchipTA05/core_mlchipTA05.h
[Info] Copied: ../ROM.h -> hw4_mlchipTA05/ROM_mlchipTA05.h
[Info] Copied: ../router.h -> hw4_mlchipTA05/router_mlchipTA05.h
[Info] Copied: ../clockreset.cpp -> hw4_mlchipTA05/clockreset_mlchipTA05.cpp
[Info] Copied: ../controller.cpp -> hw4_mlchipTA05/controller_mlchipTA05.cpp
[Info] Copied: ../core.cpp -> hw4_mlchipTA05/core_mlchipTA05.cpp
[Info] Copied: ../main.cpp -> hw4_mlchipTA05/main_mlchipTA05.cpp
[Info] Copied: ../ROM.cpp -> hw4_mlchipTA05/ROM_mlchipTA05.cpp
[Info] Copied: ../router.cpp -> hw4_mlchipTA05/router_mlchipTA05.cpp
[Info] Creating tar archive: hw4_mlchipTA05.tar.gz
hw4_mlchipTA05/
hw4_mlchipTA05/clockreset_mlchipTA05.h
hw4_mlchipTA05/controller_mlchipTA05.h
hw4_mlchipTA05/core_mlchipTA05.h
hw4_mlchipTA05/ROM_mlchipTA05.h
hw4_mlchipTA05/router_mlchipTA05.h
hw4_mlchipTA05/clockreset_mlchipTA05.cpp
hw4_mlchipTA05/controller_mlchipTA05.cpp
hw4_mlchipTA05/core_mlchipTA05.cpp
hw4_mlchipTA05/main_mlchipTA05.cpp
hw4_mlchipTA05/ROM_mlchipTA05.cpp
hw4_mlchipTA05/router_mlchipTA05.cpp
[Success] hw4_mlchipTA05.tar.gz created successfully.
```

2. **./01_submit**

- **Simulation examples of correct and incorrect results are shown below.**

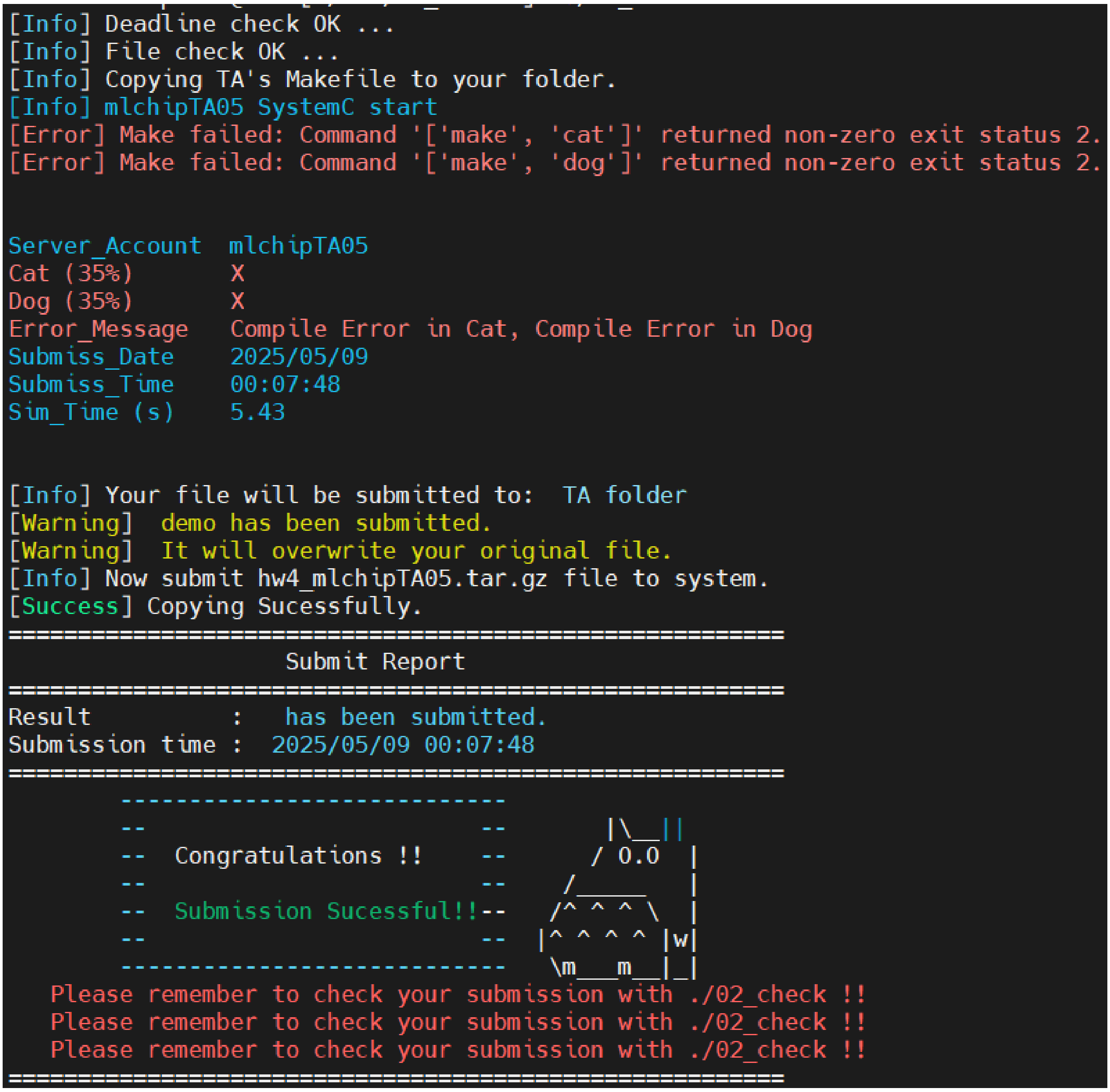


```
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] Copying TA's Makefile to your folder.
[Info] mlchipTA05 SystemC start
[Info] result_cat.log Match Golden Result
[Info] result_dog.log Match Golden Result


Server_Account  mlchipTA05
Cat (35%)       0
Dog (35%)       0
Error_Message   No_Error
Submiss_Date    2025/05/09
Submiss_Time    00:09:40
Sim_Time (s)    1879.0


[Info] Your file will be submitted to:  TA folder
[Warning]  demo has been submitted.
[Warning]  It will overwrite your original file.
[Info] Now submit hw4_mlchipTA05.tar.gz file to system.
[Success] Copying Sucessfully.
========================================================
                    Submit Report
========================================================
Result          :   has been submitted.
Submission time :  2025/05/09 00:09:40
========================================================
        ----------------------------
        --                        --        |\__||
        --  Congratulations !!    --      / 0.0  |
        --                        --    /______  |
        --  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 !!
========================================================
```

Correct example



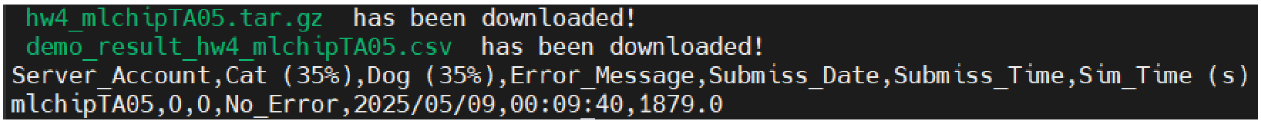
```
[Info] Deadline check OK ...
[Info] File check OK ...
[Info] Copying TA's Makefile to your folder.
[Info] mlchipTA05 SystemC start
[Error] Make failed: Command '['make', 'cat']' returned non-zero exit status 2.
[Error] Make failed: Command '['make', 'dog']' returned non-zero exit status 2.


Server_Account  mlchipTA05
Cat (35%)       X
Dog (35%)       X
Error_Message   Compile Error in Cat, Compile Error in Dog
Submiss_Date    2025/05/09
Submiss_Time    00:07:48
Sim_Time (s)    5.43


[Info] Your file will be submitted to:  TA folder
[Warning]  demo has been submitted.
[Warning]  It will overwrite your original file.
[Info] Now submit hw4_mlchipTA05.tar.gz file to system.
[Success] Copying Sucessfully.
========================================================
                    Submit Report
========================================================
Result          :   has been submitted.
Submission time :  2025/05/09 00:07:48
========================================================
        ----------------------------
        --                        --       |\__||
        --  Congratulations !!    --      / 0.0  |
        --                        --    /______  |
        --  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



```
 hw4_mlchipTA05.tar.gz  has been downloaded!
 demo_result_hw4_mlchipTA05.csv  has been downloaded!
Server_Account,Cat (35%),Dog (35%),Error_Message,Submiss_Date,Submiss_Time,Sim_Time (s)
mlchipTA05,0,0,No_Error,2025/05/09,00:09:40,1879.0
```