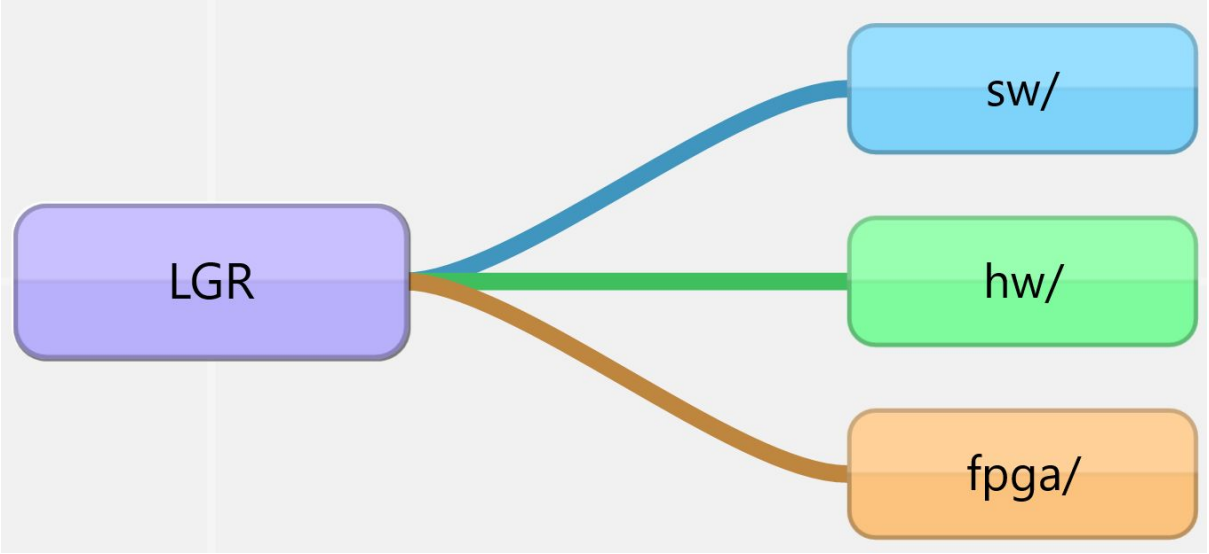
Quang’s Directory in Server: "ndnquang@140.116.245.125:/home/ndnquang/lgr/"

Ai’s Directory in Server: “

Github link: https://github.com/icecream81092/lgr.git

(Uploading)

Outline (LGR)



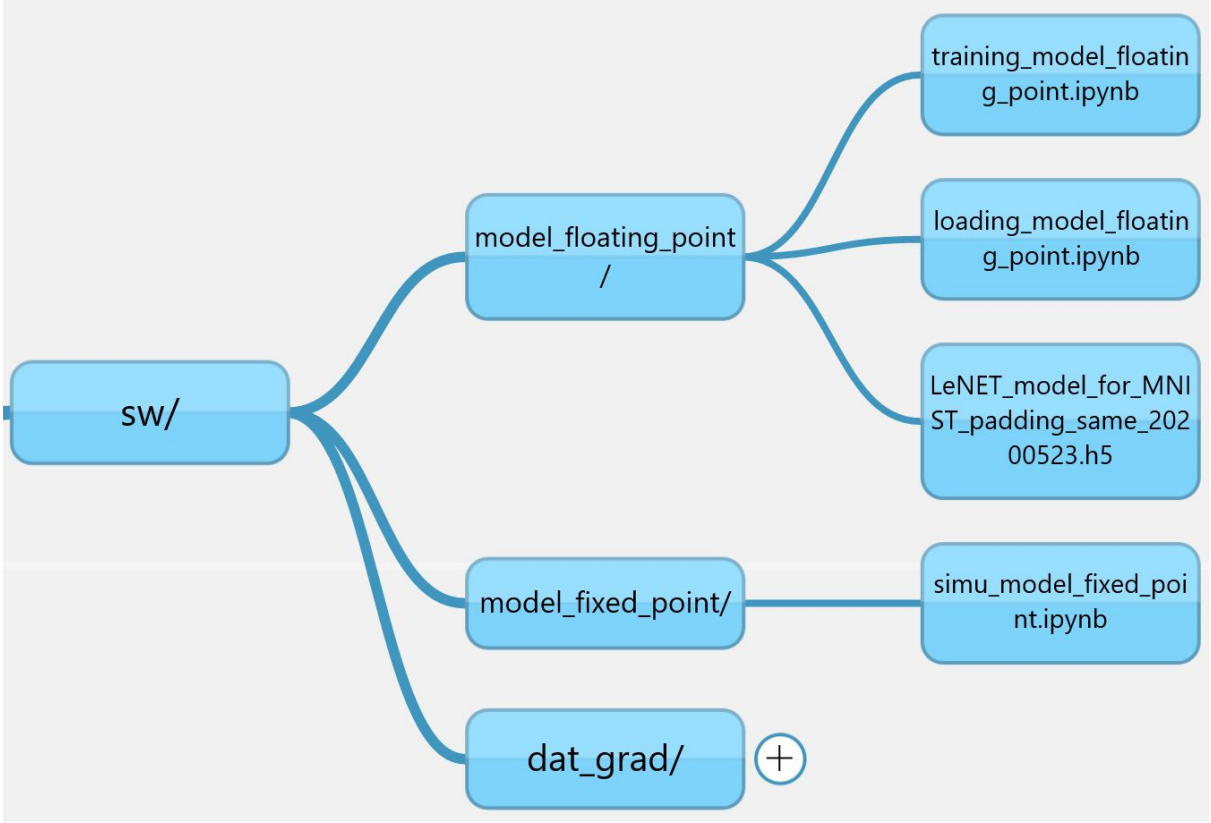
*Install Google Colabe for Chrome

Software (sw)

Only a few important files are listed here, the others are too many and do not need to be modified, so they are not listed one by one

Open

Folder and file



Step 1: Training

Google Keras for MNIST
Reference https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py
Open training_model_floating_point.ipynb

+ Code + Text

```
import keras
import numpy as np
np.random.seed(1337)
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

batch_size = 128
num_classes = 10
epochs = 20

# dimension
rows, columns = 28, 28
```

Save the model file: `'LeNET_model_for_MNIST_padding_same_%Y%m%.h5'`

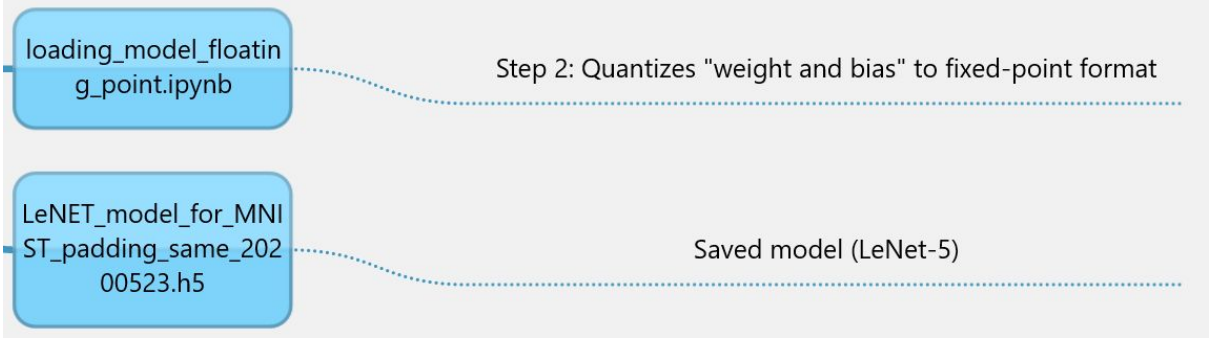
```
from datetime import date
today = date.today().strftime('%Y%m%d')
model_name = 'LeNET_model_for_MNIST_padding_same_'+str(today)+'.h5'
print(model_name)

model.save(model_name)
print("Saved model to disk")
```

```
LeNET_model_for_MNIST_padding_same_20200523.h5
Saved model to disk
```

Step 2: Quantization

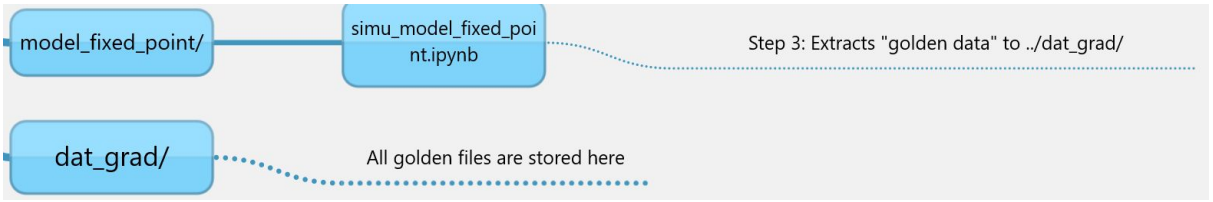
Open the file: loading_model_floating_point.ipynb



```
def float_to_hexadecimal(number):
    intBits=4
    decBits=4
    if decBits == 0:
        mx = pow(2,intBits-1) - 1 # maximum number
    else:
        mx = pow(2,intBits-1) - pow(2,-1*decBits) # maximum number
    mn = -1*pow(2,intBits-1) # minimum number
    if number > mx:
        print ("number:" + str(number) + " has been truncated to: " + str(mx))
        number = mx
    elif number < mn:
        print ("number:" + str(number) + " has been truncated to: " + str(mn))
        number = mn
    n = []
    m = 0
```

Save the weights and bias into the fixed/ directory. sw/model_floating_point/fixed.

Step 3: Extract golden file



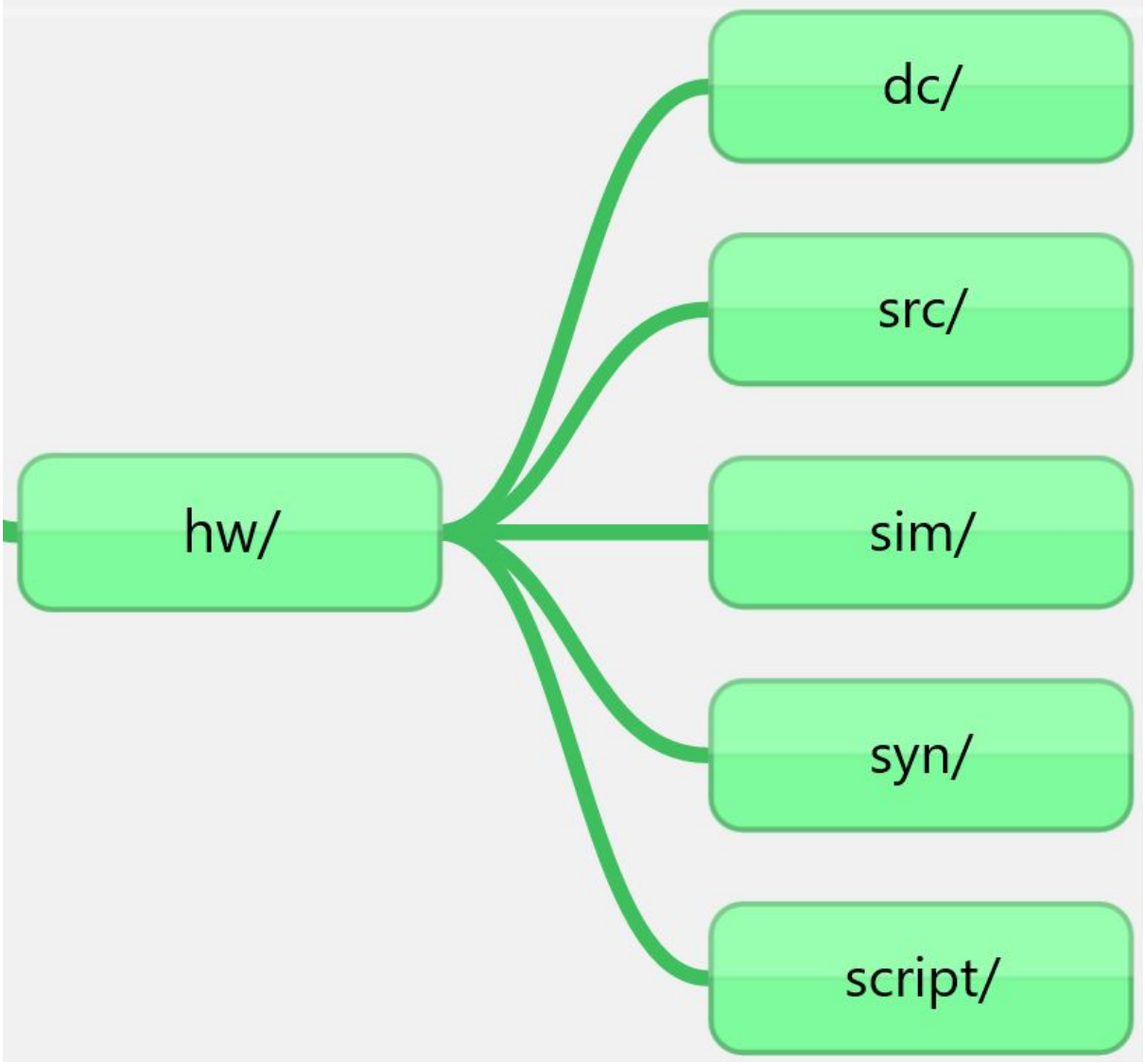
Run the code:

```
def floatToTwosComplementDecimal(intBits,decBits,number):
    if decBits == 0:
        mx = pow(2,intBits-1) - 1 # maximum number
    else:
        mx = pow(2,intBits-1) - pow(2,-1*decBits) # maximum number
    mn = -1*pow(2,intBits-1) # minimum number
    if number > mx:
        print ("number:" + str(number) + " has been truncated to: " + str(mx))
        number = mx
    elif number < mn:
        print ("number:" + str(number) + " has been truncated to: " + str(mn))
        number = mn
    n = []
    m = 0
```

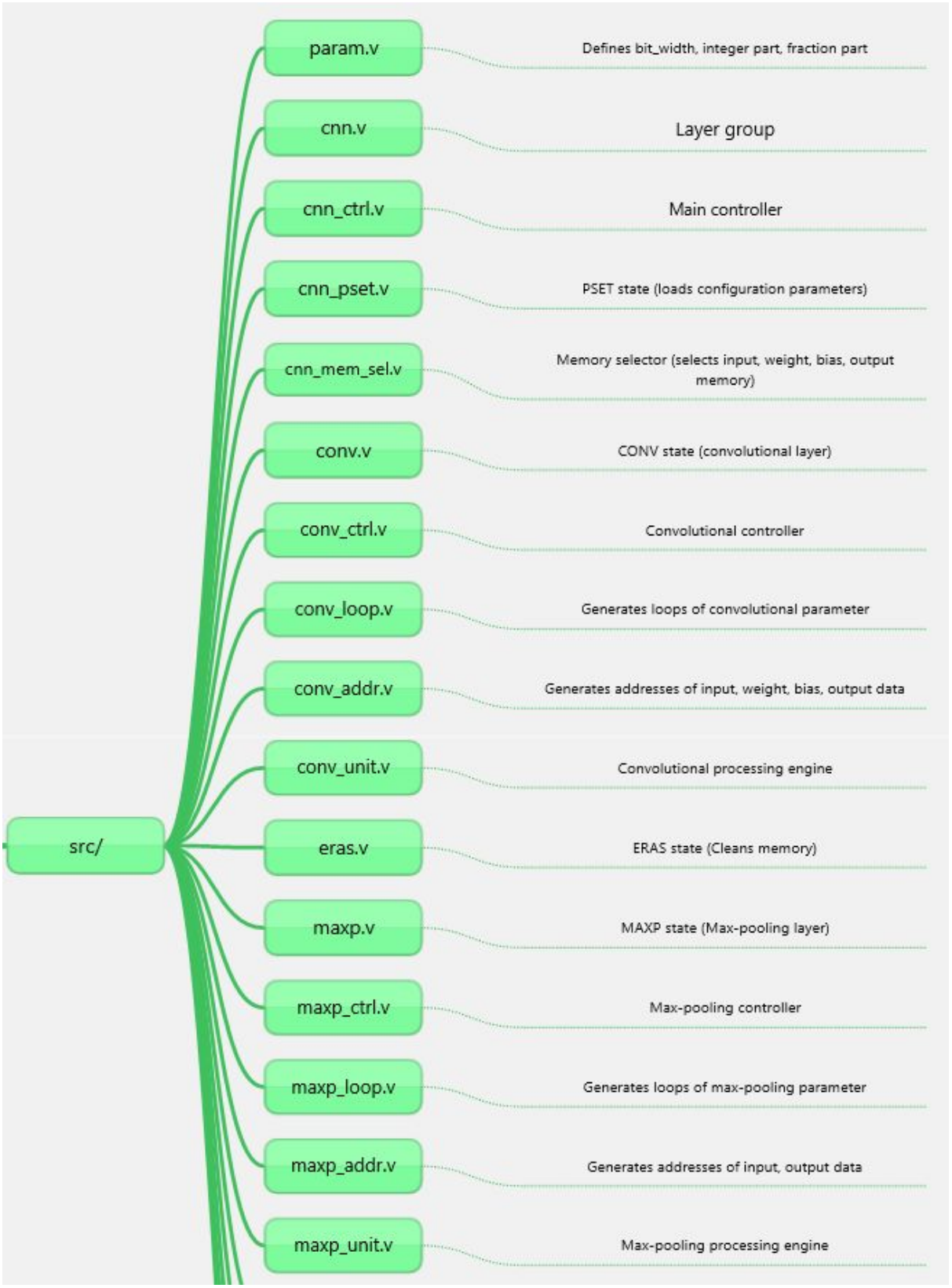
L1_C1_b_fixed: bias. Layer 1 convolution 1 bias fixed.
L1_C1_w_fixed: weights Layer 1 convolution 1 weights fixed.

Hardware (hw)

Folder and file



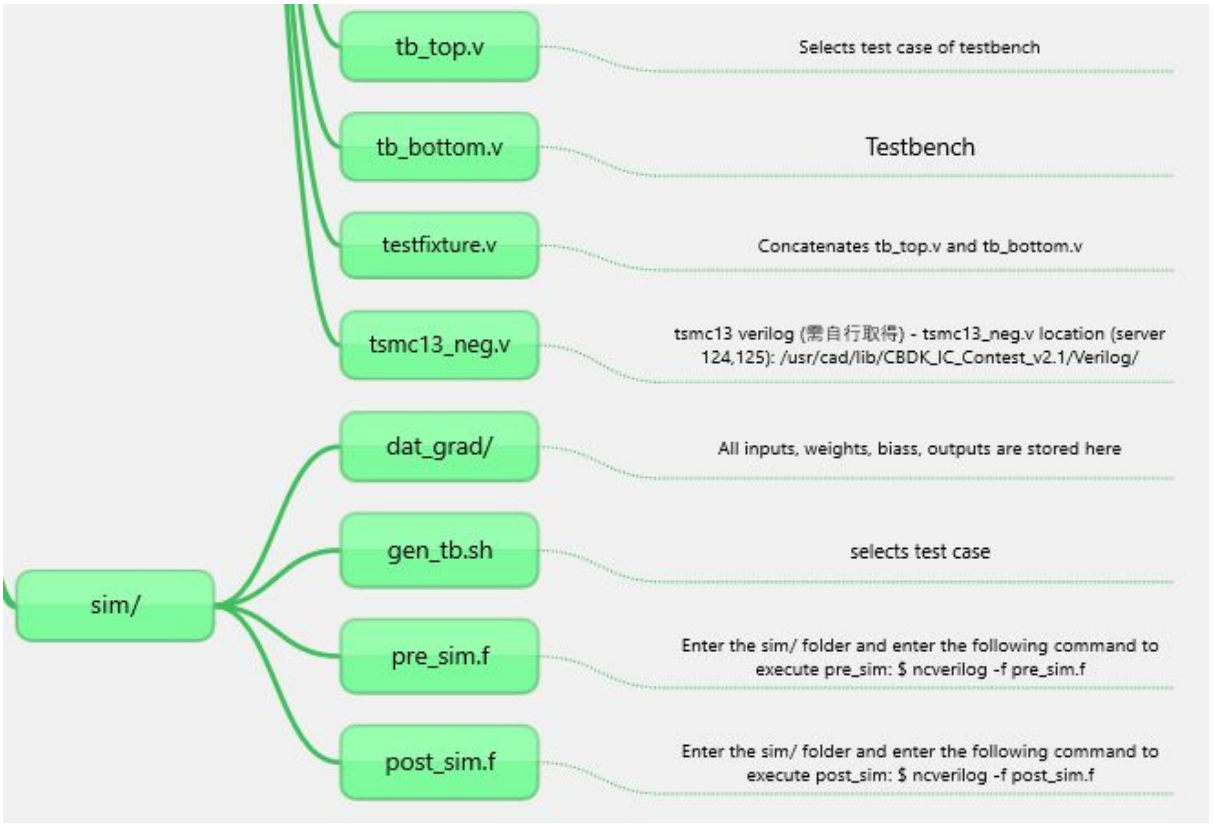
Circuit



Simulation

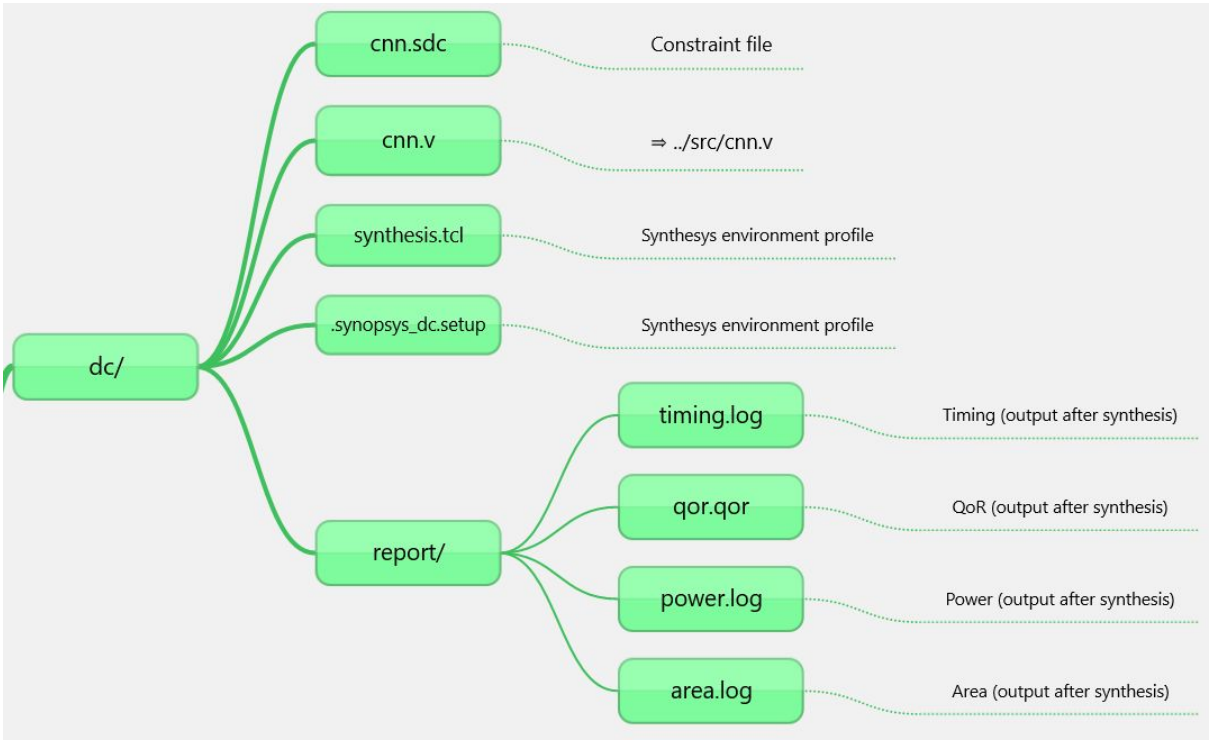
$ cd script/

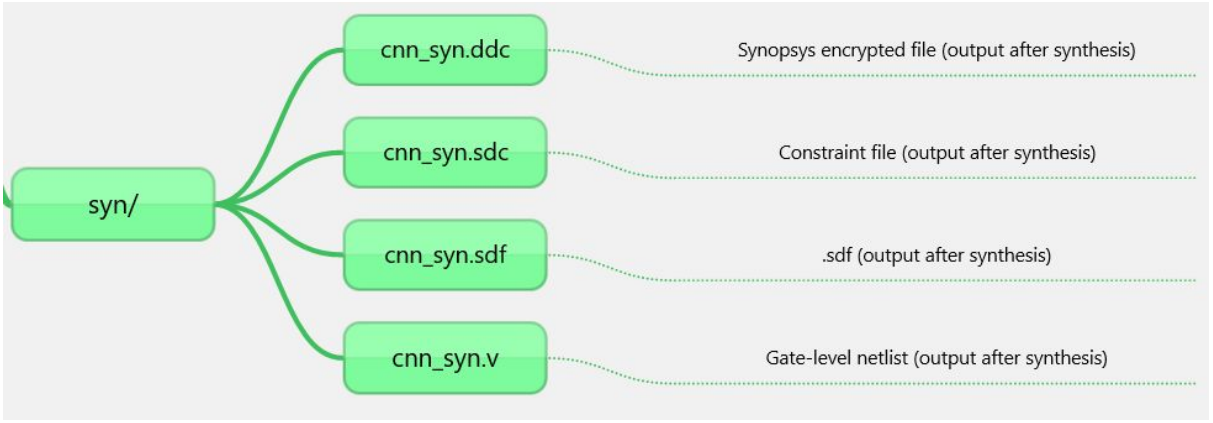
$ ./run_ncverilog_all.sh (pre_sim)



Synthesys

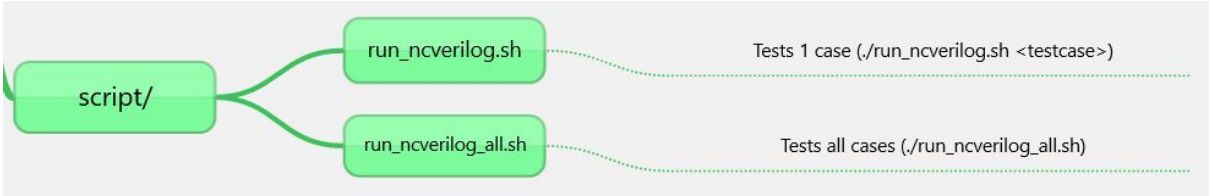
$ cd ../dc/
$ dc_shell
$ source synthesis.tcl
$ cd ../sim/
$ ncverilog -f post_sim.f





Test more cases

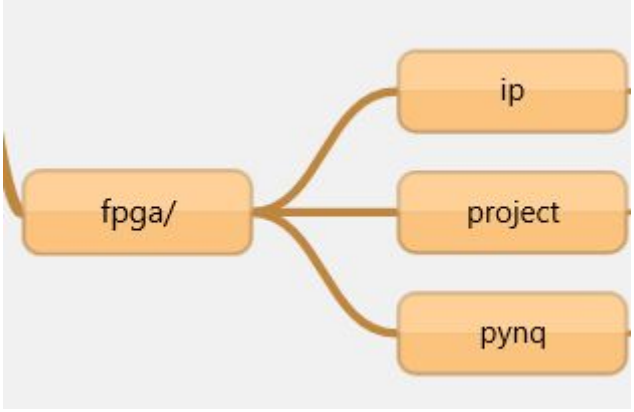
$ ./run_ncverilog.sh 00
$ ./run_ncverilog_all.sh



FPGA (fpga)

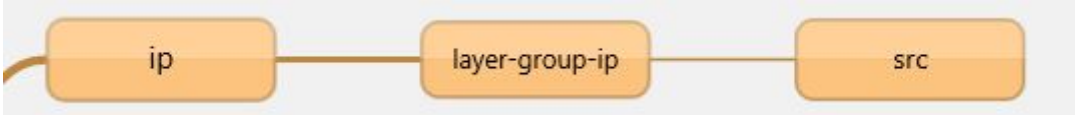
Folder and file

- ip: contains layer-group-ip
- project: contains lenet project
- pynq: contains jupyter notebook code

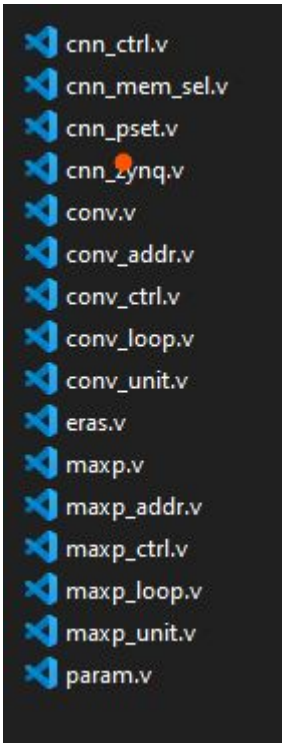


Create and Pakage new IP

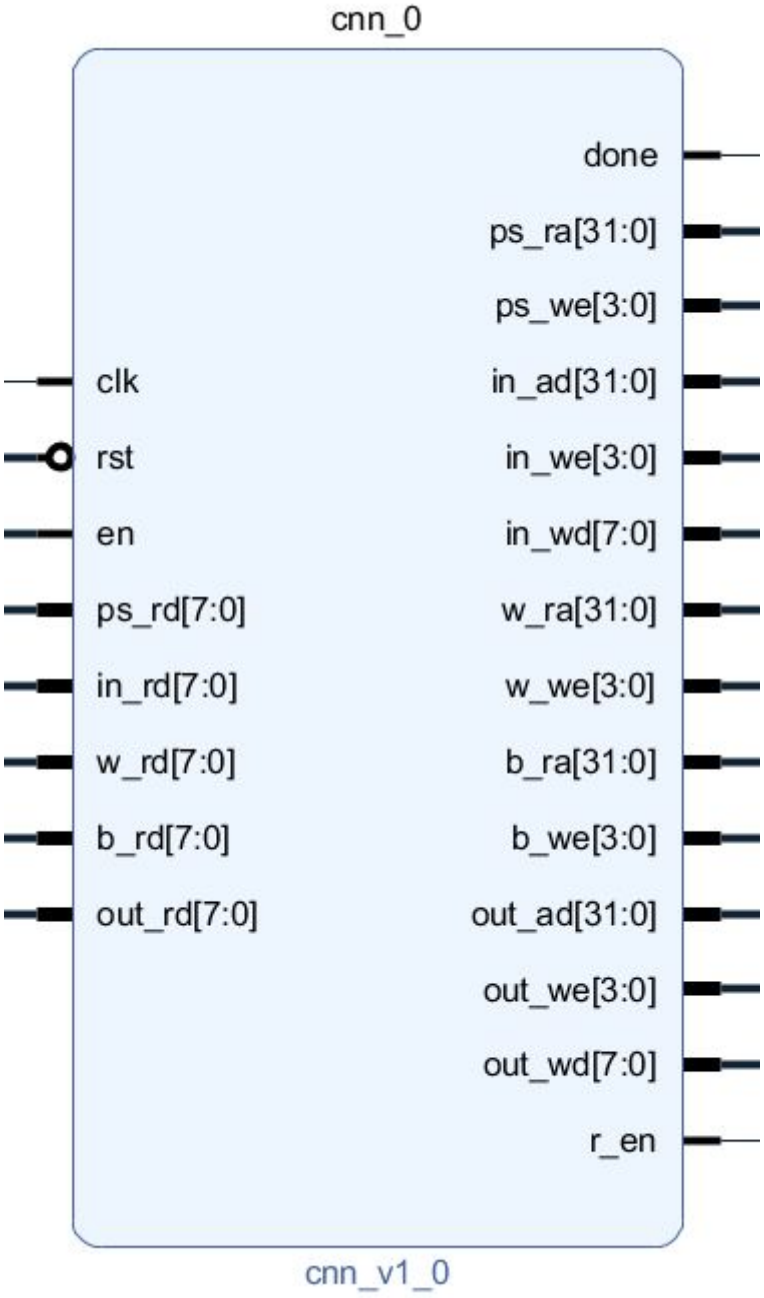
Please follow the homework 2 (DEEP LEARNING INTEGRATED CIRCUIT DESIGN AND ACCELERATION)



Notice: Change cnn.v to cnn_zynq.v

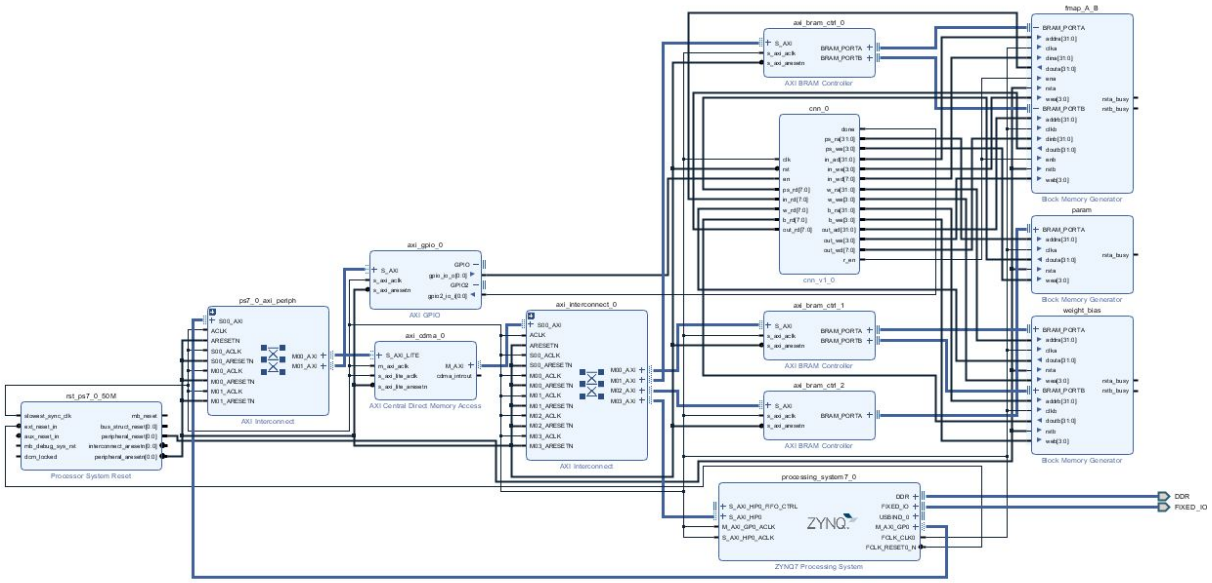


This is the IP:

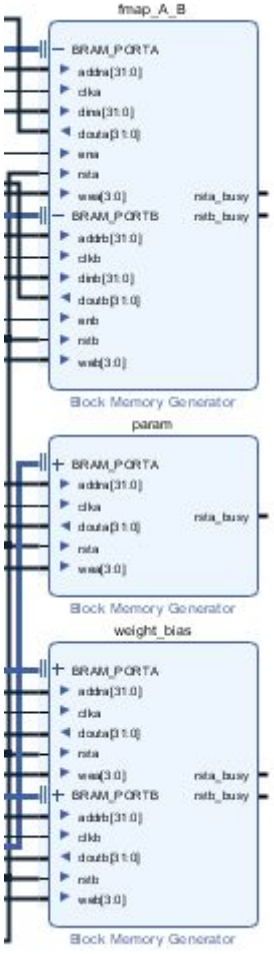


Block design

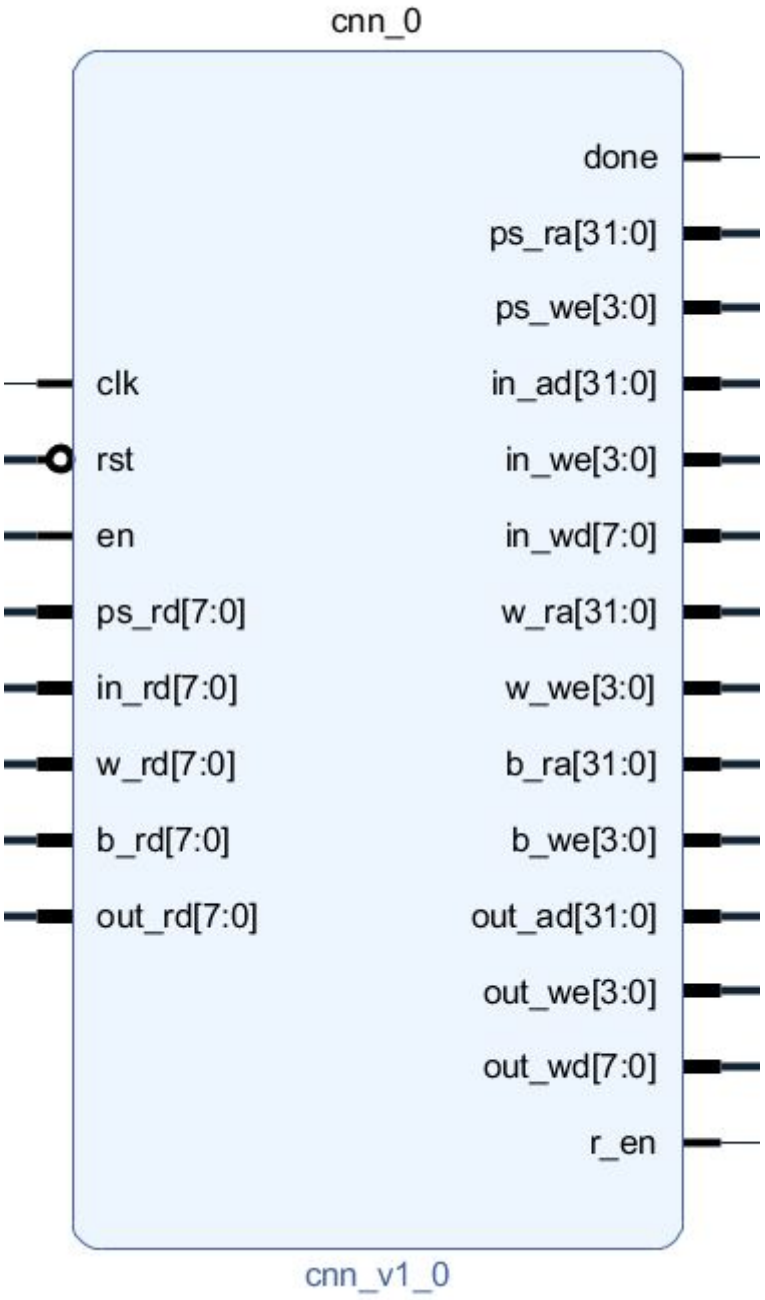
Follow homework 3 and 4 (DEEP LEARNING INTEGRATED CIRCUIT DESIGN AND ACCELERATION)
Open lenet.xpr



There are 5 BRAMs: fmap_A_B BRAM (dual), weight_bias BRAM (dual), param BRAM (single):



CNN IP:



Pynq

Pynq Z2 board: http://140.116.245.116:7777/tree?
User and password: default setting
Open mnist_main.ipynb
Execute: Ctrl+Enter