IC Design

Homework # 4

(Due on 2019/01/04, 13:20, Verilog code and report upload to CEIBA)

✧ Plagiarism is not allowed. 10% penalty for each day of delay.

✧ Any further questions, you can send e-mail to TA.

Problem Statement

The following diagram shows a 64×64 image:

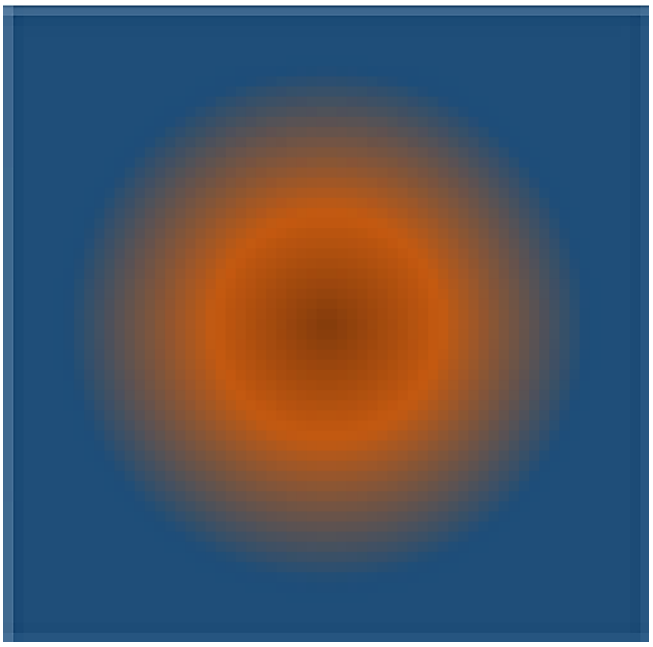


Diagram 4 – 1

Given the color of each pixel, you are going to implement a linear classifier that can classify the pixel into one of the two classes:

a) the inner circle

b) the outer circle

What is Linear Classifier?

It is one of the most fundamental models of neural networks. You can see it as a special case of multilayer perceptron (MLP). MLP is also a class of artificial neural network.

Linear classifier and MLP are very useful tools to solve image classification tasks. For example, we can train a linear classifier that can classify handwritten digits with more than 90% of accuracy.

Mathematically, a linear classifier is nothing more than some matrix operations. In this homework, what you are going to compute is:

$$z = [w_0 \quad w_1 \quad w_2] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + b \quad \dots \quad (1)$$

w_0, w_1, w_2 and b are pre-trained network parameters. Their values are as follows:

$$\begin{aligned} w_0 &= 5 \\ w_1 &= 1 \\ w_2 &= -9 \\ b &= -76 \end{aligned}$$

Block Diagram 2's complement multiplier → adder → comparator

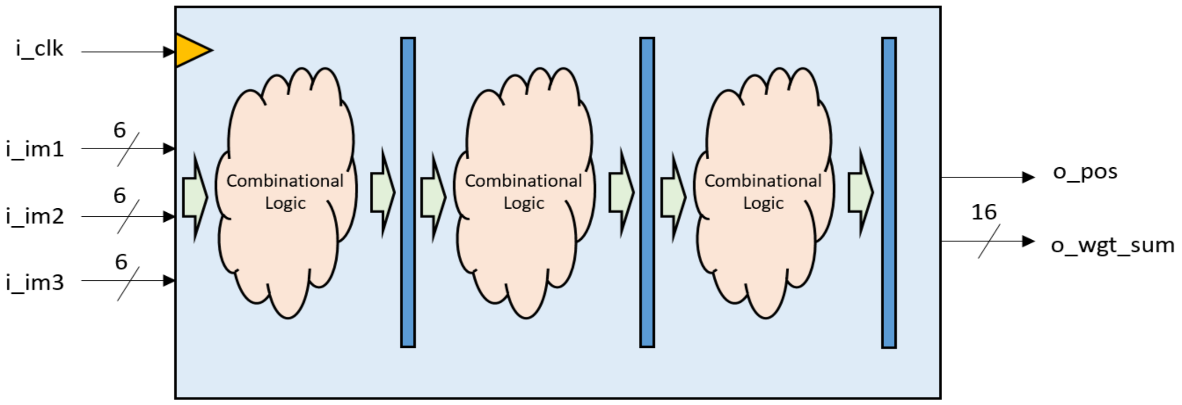
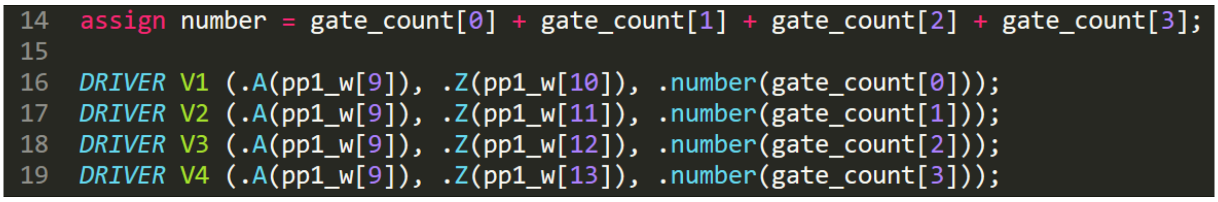


Diagram 4 – 2

The table below gives details about the I/O interfaces of the circuit:

Signals	I/O	Width	Descriptions
i_im1	I	6	• Red component of each pixel represented in 2's complement. • x_0 in equation (1).
i_im2	I	6	• Green component of each pixel represented in 2's complement. • x_1 in equation (1).
i_im3	I	6	• Blue component of each pixel represented in 2's complement. • x_2 in equation (1).
o_wgt_sum	O	16	• Weighted sum represented in 2's complement. • z in equation (1).

o_pos	O	1	• Positive flag. • HIGH if o_wgt_sum $\geq$ 0, LOW if o_wgt_sum $<$ 0
number	O	51	• This signal is used to compute the transistor count (area) of the whole circuit. • Each component in lib.v has an output port named *number*. • You should assign the sum of all the “numbers” to this port (refer to Diagram 4 – 3).

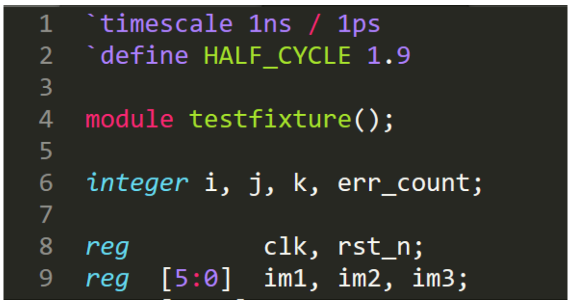


```verilog
assign number = gate_count[0] + gate_count[1] + gate_count[2] + gate_count[3];

DRIVER V1 (.A(pp1_w[9]), .Z(pp1_w[10]), .number(gate_count[0]));
DRIVER V2 (.A(pp1_w[9]), .Z(pp1_w[11]), .number(gate_count[1]));
DRIVER V3 (.A(pp1_w[9]), .Z(pp1_w[12]), .number(gate_count[2]));
DRIVER V4 (.A(pp1_w[9]), .Z(pp1_w[13]), .number(gate_count[3]));
```

Diagram 4 – 3

You can change the half cycle time at the second line of the testbench to any number you like:



```verilog
`timescale 1ns / 1ps
`define HALF_CYCLE 1.9

module testfixture();

integer i, j, k, err_count;

reg         clk, rst_n;
reg   [5:0] im1, im2, im3;
```

Diagram 4 – 4

Your half cycle time should be as short as possible; yet, long enough for your circuit to function properly.

Timing Diagram

x_0, x_1, x_2 will keep streaming in after the circuit is reset:

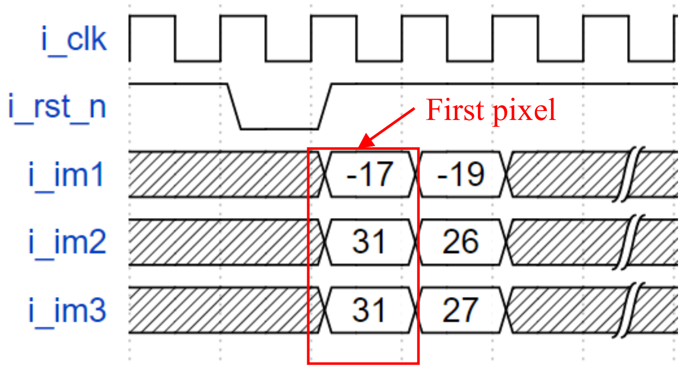


Diagram 4 – 5

The weighted sum of the corresponding inputs should come three cycles later (your circuit should have exactly 3 stages of pipelines):

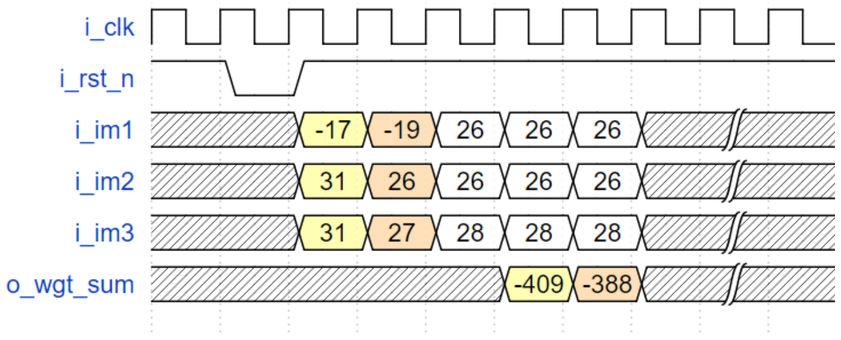


Diagram 4 – 6

The signal o_pos, should be asserted whenever the weighted sum is positive:

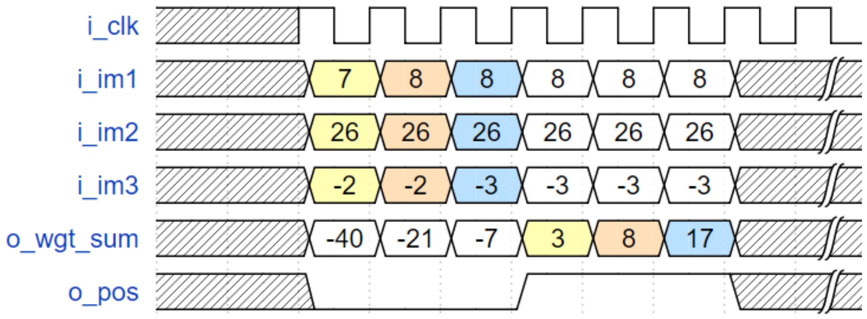
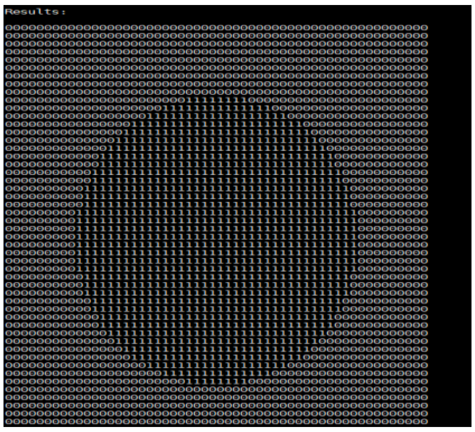


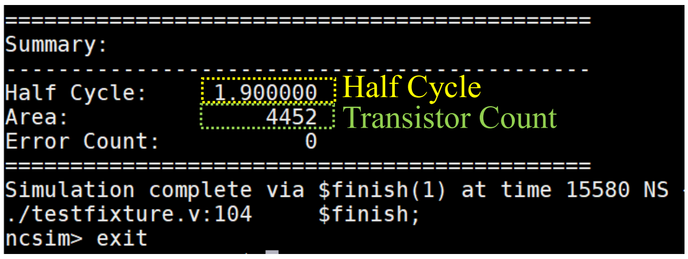
Diagram 4 – 7

Results of Simulation



```
Results:
```

Diagram 4 – 8



```
============================================
Summary:
--------------------------------------------
Half Cycle:    1.900000  Half Cycle
Area:              4452  Transistor Count
Error Count:          0
============================================
Simulation complete via $finish(1) at time 15580 NS
./testfixture.v:104     $finish;
ncsim> exit
```

Diagram 4 – 9

Grading Policy

1. Gate-level design using Verilog (70%)

Your score will depend on both the correctness and performance of your design.

(a) Correctness (50%)

We provide a test bench with 4,096 pixels which automatically grades your design. Your score in this part will be 50* (1 – error numbers / 4,096)

(b) Ranking (20%)

We will rank all students who have passed (a) in terms of the product of the half-cycle time and the number of transistors. The score will be given as follows.

Percentage of passing students	Score
If your ranking > 85 %	20
65%~84%	16
35%~64%	12
15%~34%	8
0%~14%	4
Using operands, not standard cell logic	0
Correctness failed	0
Plagiarism	0

2. Report (30%)

(a) Circuit diagram (10%)

Plot the gate-level circuit diagram of your design. You are encouraged to plot it hierarchically so that readers can understand your design easily.

(b) Simulation (10%)

(5%) Record your **minimum half-cycle time** according to your simulation and **plot critical path** on the diagram in (a).

You have to write down your minimum half-cycle time according to your simulation. This minimum half-cycle time would be verified by TAs.

(5%) Find the number of transistors in your design **by simulation.**

Compute the product of the minimum half-cycle time and the number of transistors.

The numbers of transistors in all cells are specified in the lib.v

(c) Discussion (10%)

Discuss about your design. For example, introduce your design, how do you cut your pipeline? What is the structure of your classifier? How do you improve your critical path and the number of transistors? How do you trade-off between area and speed?

Notification

- ➢ Following are the files you will need (available on the class website)
  HW4.zip includes
  - ■ **HW4_2018.pdf** : this document.
  - ■ **HW4_tutorial_2018.pdf** Tutorial in class
  - ■ **lin_class.v**:
    Dummy design file. Program the design in this file.
    The header of the top module and the declaration of the I/O ports are predefined in this file and you are not allowed to change them.
  - ■ **lib.v**: standard cells.
  - ■ **testfixture.v**:
    Test bench for your design.
  - ■ **im1.txt, im2.txt, im3.txt**:
    Input patterns for testbench.
  - ■ **golden.txt,**:
    Output patterns of correct answers for test bench.
    You should submit the **Report** to Ceiba.
- ➢ Your Verilog codes written in **only one file** should be uploaded to CEIBA by due time (Don't use e-mail).
- ➢ The following files should be compressed and uploaded to CEIBA by due time.
  - ● Report (PDF format)
  - ● lin_class.v
- ➢ File name rule : *HW4_(student id)_v#*
  Ex. HW4_b03901301_v1.zip
  Ex. HW4_b03901311_v2.zip

TA ：江子近，王鈺凱，EE2 -329

TA email: r06943159@ntu.edu.tw , r06943124@ntu.edu.tw
HW4 Office hours: 2018/12/27 19:00~21:00 @ BL214
2019/01/03 19:00~21:00 @ BL214
If you have no time at office hours, you can email TA to discuss another time for appointment.