NCTU-EE IC LAB – FALL 2019

Lab08 Exercise

Design: Image Processing (IP)

Data Preparation

1. Extract test data from TA's directory:
   % tar xvf ~iclabta01/Lab08.tar
2. The extracted LAB directory contains:
   a. EXERCISE/
   b. EXERCISE_wocg/
   c. PRACITCE/
   d. JG/

Basic Concept

This lab will give you a basic concept about digital image processing (DIP). You will learn how to enhance an image by simple image processing method.

In general, **spatial domain filtering** is commonly used to modify or enhance an image. In other words, you can emphasize or remove certain feature by doing so. Actually, filtering, smoothing, sharpening and edge enhancement are all included.

Take the following picture as example for noise reduction:

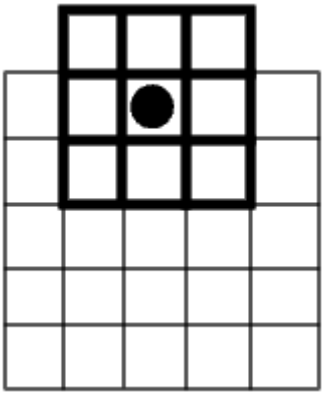
Original image with noise

Denoised image

Zero Padding of Borders

When you apply a filter to pixels on the borders of an image, some of the elements of the computational molecule may not overlap actual image pixels. For example, if the molecule is 3-by-3 and you are computing the result for a pixel on the top row of the image, some of the elements of the molecule are outside the border of

the image



In order to compute output values for the border pixels, you need to pad image matrix with zeros. In other words, the output values are computed by assuming that the input image is surrounded by additional rows and columns of zeroes.

Design Description

In this lab, there is a design called IP. The block diagram of this lab is shown below. IP will received a 8×8 grayscale image **in_image[7:0]** and then a mode **in_mode[1:0]** from the PATTERN. Each pixel in the image will be **0~255.** You are asked to output processed image 8×8 **out_number[11:0]** by doing the following two steps: **Zero padding** and **Do one of the image process function**. For next round, you may get a new image or a new mode depending on given **in_valid_1** or **in_valid_2**.

1. Zero padding:

The 8×8 image should be padded with zeros and you will get a 10x10 padded-image.

2. Do one of the image process function:

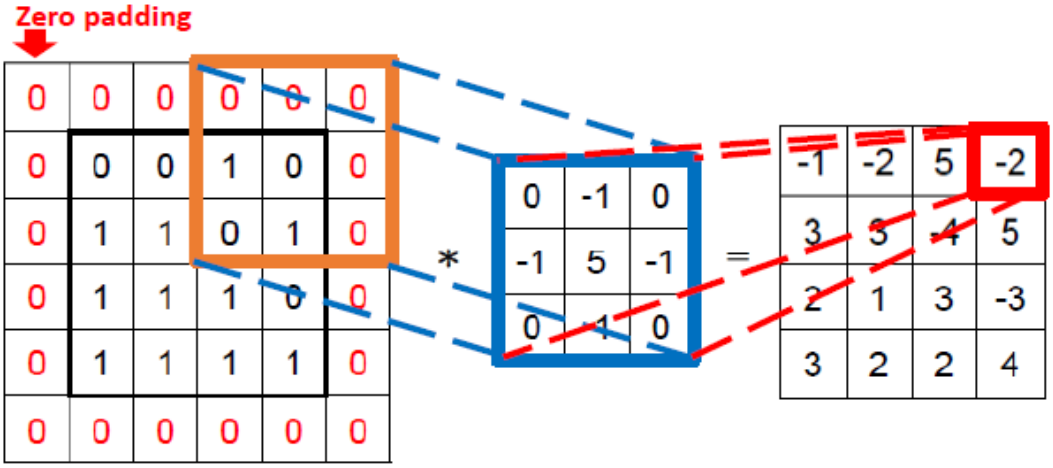
The following are three image process modes given by **in_mode[1:0]:**

(1) in_mode: 0 → **sharpen filter**

0	-1	0
-1	5	-1
0	-1	0

sharpen filter

There is a 4×4 image with zero padding as an example. Each pixel should be multiplied with the relative filter pixel. Sum all values in a window to be the center pixel. The below is an example.

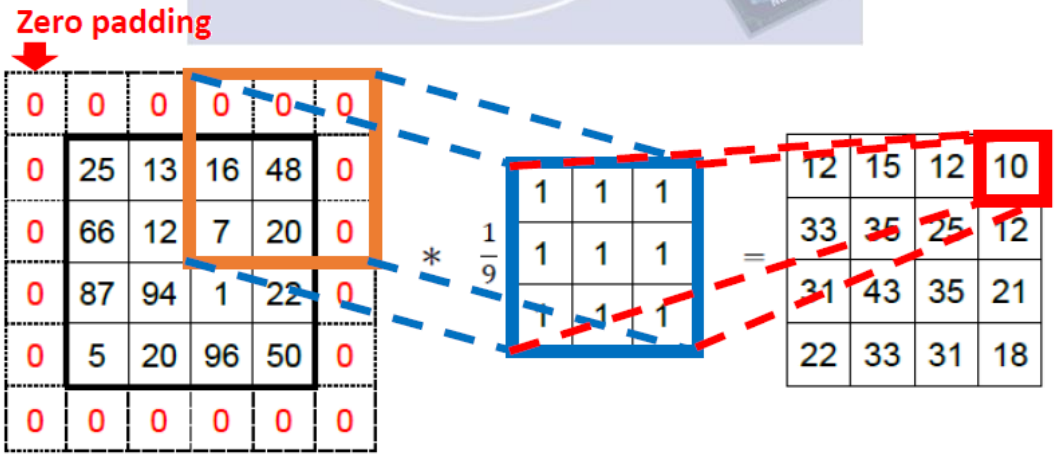


(2) in_mode: 1 → **mean filter**

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

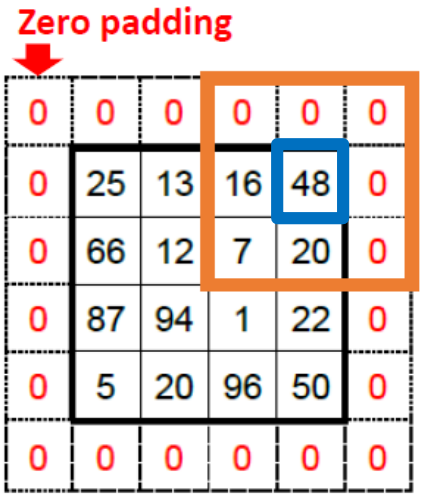
mean filter

There is a 4×4 image with zero padding as an example. Average the value of 3×3 kernel and round down to the units digit. Then, the result is the center pixel. The below is an example.



(3) in_mode: 2 → **midpoint filter**

The midpoint filter calculates the midpoint value between minimum and maximum value in a window. Midpoint filter combines both averaging and sorting. Replacing the center pixel in a window with the midpoint of max and min values. The below is an example.

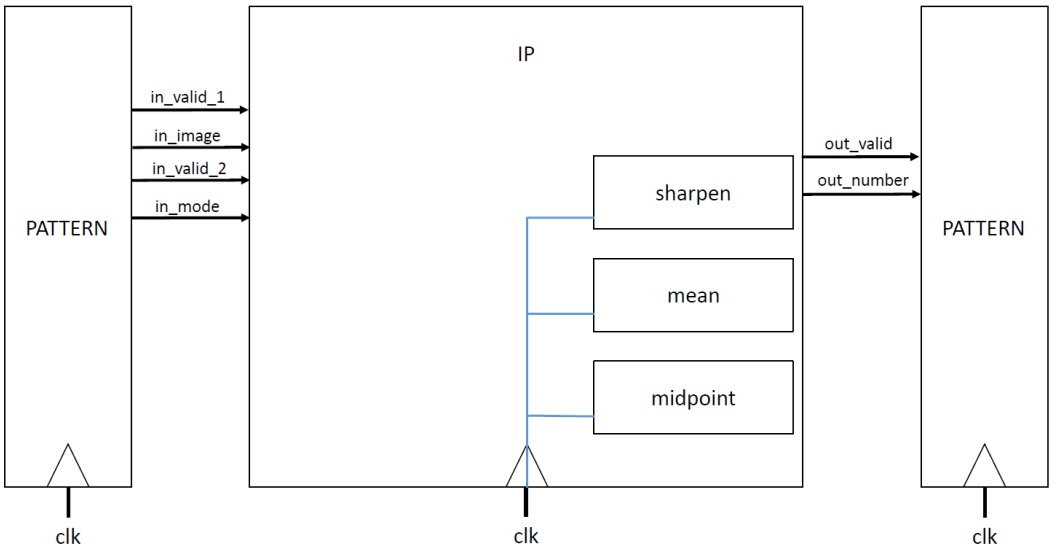


Max:48
Min:0

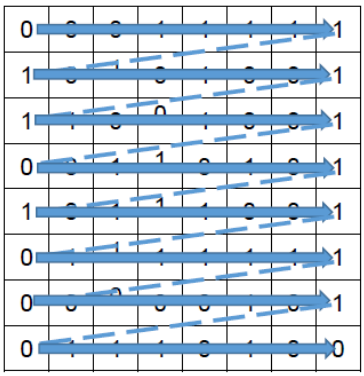
So:
48 => (48+0)/2=24

Stage 1:

At EXERCISE_wocg/ **IP_wocg.v**, you should implement an IP module as below figure.



1. **INPUT:** Receive grayscale image form PATTERN



※The in_image[7:0] will be given as the sequence above in 64 cycle.

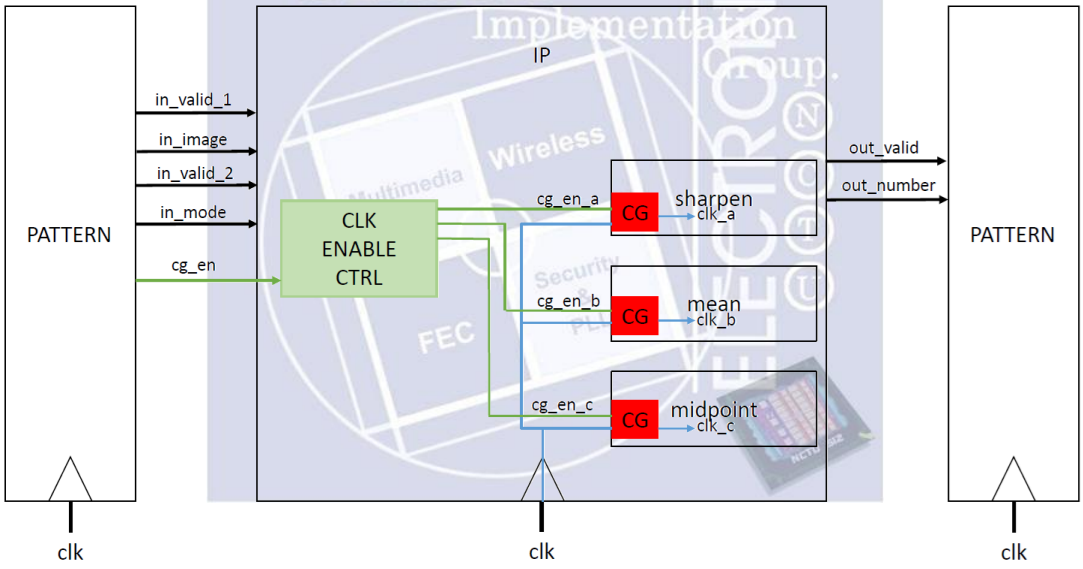
2. **IP:** Image process module

   There different image processing blocks whose functions are described as above.

3. **OUTPUT:** Output filtered image

Stage 2:

At EXERCISE /**IP.v**, you should implement an IP module with clock gating cell as below figure. Main different part is clock gating cell and cg_en input signal.



1. **INPUT:** Receive grayscale image form PATTERN
2. **IP:** Image process module

   You should add clock gating cell in the image processing block. If cg_en is high, three image processing block can perform clock gating; otherwise, if cg_en is low, three image processing block follow **clk.**

   Hint: Refer to Lab08_Practice to implement the clock gating cell.

3. **OUTPUT:** Output filtered image

Inputs

I/O	Signal name	Description
Input	**clk**	Clock
Input	**rst_n**	Asynchronous reset active low reset
Input	**cg_en**	If cg_en is high, the image processing blocks should execute clock gating. Otherwise, if cg_en is low, the image processing blocks follow **clk**. There is no this signal in IP_wocg.v.
Input	**in_valid_1**	**in_image[7:0]** valid when **in_valid_1** is high
Input	**in_image[7:0]**	**in_image[7:0]** is valid when **in_valid_1** is high Input data will be sent in **raster scan order (Figure 3)**
Input	**in_valid_2**	**in_mode[1:0]** valid when **in_valid_2** is high
Input	**in_mode[1:0]**	**in_mode[1:0]** is valid in the one cycle when **in_valid_2** is high. 0: sharpen filter, 1: mean filter, 2: midpoint filter

Outputs

I/O	Signal name	Description
output	**out_valid**	Should be set to low after reset. Should set to high when your **out_number[11:0]** is ready.
output	**out_number[11:0]**	TA's pattern will sample your result at the following negative edge of **clk** Should send output data in **raster scan order.**

Specifications

1. Top module name : IP (File name: IP.v)
2. Input pins : **clk, rst_n, cg_en, in_valid_1, in_image[7:0] , in_valid_2, in_mode[1:0]**
   Output pins : **out_valid, out_number[11:0]**
3. Use **asynchronous** reset active low architecture.
4. All your output register should be set zero after reset.
5. Changing clock period is prohibited **(fixed at 8ns)**.
6. The instance name of the clock gating cell(GATED_OR) should be GATED_XXX(e.g., GATED_sharpen).

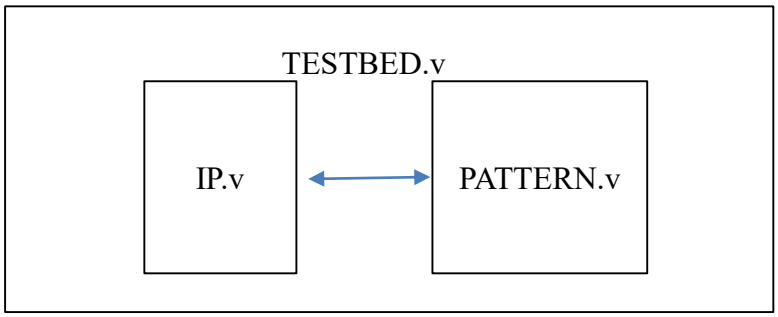
```
GATED_OR GATED_sharpen (
.CLOCK( clk ),
.SLEEP_CTRL( G_sleep_sharpen ), // gated clock
.RST_N( rst_n ),
.CLOCK_GATED( gated_sharpen )
);
```

7. After synthesis, check the “IP.area” and “IP.timing” in the folder “Report”. The area report is valid only when the slack in the end of “IP.timing” is **non-negative**.
8. The synthesis result **cannot** contain any **latch except for clocking gating latch**
9. The output loading is set to 0.05.
10. Input delay and output delay are 0.5*Clock Period.
11. You can’t have timing violation in gate-level simulation
12. **Don’t use memory in this lab.**
13. Your latency should be less than 1500 cycles.
14. Your area should be less than 1,000,000 um^2.
15. in_valid_1, in_valid_2 and out_valid should not be high at the same time.
16. Your design should have **at least 10% power reduction** from cg_en-off to cg_en-on, otherwise it will be treated as failed.
17. **Power report position: 04_PTPX/Report/CP_POWER** or **CP_CG_POWER**

Report Total Power Example:

```
  Attributes
  ----------
      i  -  Including register clock pin internal power
      u  -  User defined power group

                          Internal  Switching  Leakage    Total
Power Group               Power     Power      Power      Power   (     %)  Attrs
--------------------------------------------------------------------------------
clock_network             8.077e-06 4.576e-05 1.770e-09 5.384e-05 ( 0.65%)
register                  2.254e-03 1.129e-04 1.270e-07 2.367e-03 (28.39%)  i
combinational             4.303e-03 1.611e-03 1.054e-06 5.915e-03 (70.96%)
sequential                   0.0000    0.0000    0.0000    0.0000 ( 0.00%)
memory                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
io_pad                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box                    0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power  = 1.769e-03   (21.23%)
  Cell Internal Power  = 6.565e-03   (78.76%)
  Cell Leakage Power   = 1.183e-06   ( 0.01%)
    Intrinsic Leakage  = 1.183e-06
    Gate Leakage       =    0.0000
                         ---------
Total Power            = 8.336e-03  (100.00%)

X Transition Power     = 1.902e-04
Glitching Power        =    0.0000

Peak Power             =    0.1102
Peak Time              =     32582
```

Block Diagram



Grading Policy

Functionality:

IP_wocg: 20%

IP: 50%

JasperGold SEC check: (You can only get these points if you pass the gate level simulation. No 2nd demo chance.)

Run1: 2.5%

Run2: 2.5%

Performance: Total Power*Simulation time 25%

Note

1. Please upload the following file on NewE3 platform before **12:00 at noon** on **Nov. 25**:

   IP_wocg_iclabxx.v, IP_iclabxx.v (xx is your account no., i.e., IP_iclab99.v)

2. Template folders and reference commands:

   01_RTL/ (RTL simulation) "**./01_run**" "**./02_run_cg**"

   02_SYN/ (Synthesis) "**./01_run_dc**"

   (Check the design which contains **latch and error** or not in **syn.log**)

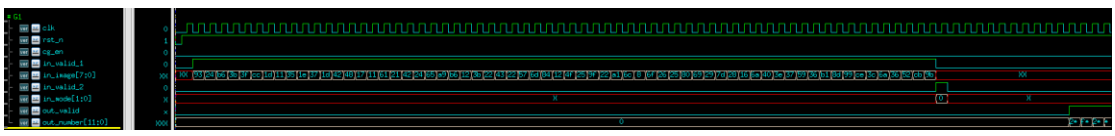
   03_GATE_SIM/ (GL simulation) "**./01_run**" "**./02_run_cg**"

   04_PTPX/run "**./01_run_ptpx**" and "**./02_run_cg_ptpx**"**to get the power of your design.**

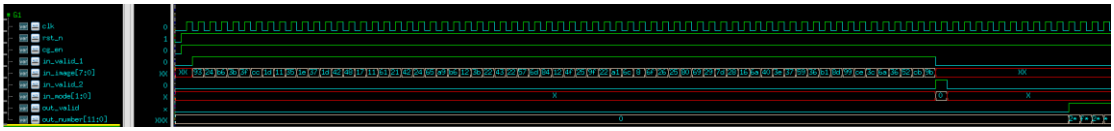
   You can key in "**./09_clean_up**" to clear all log files and dump files in each folder.

Waveform Example

For PATTERN.v, you need to give cg_en==0:

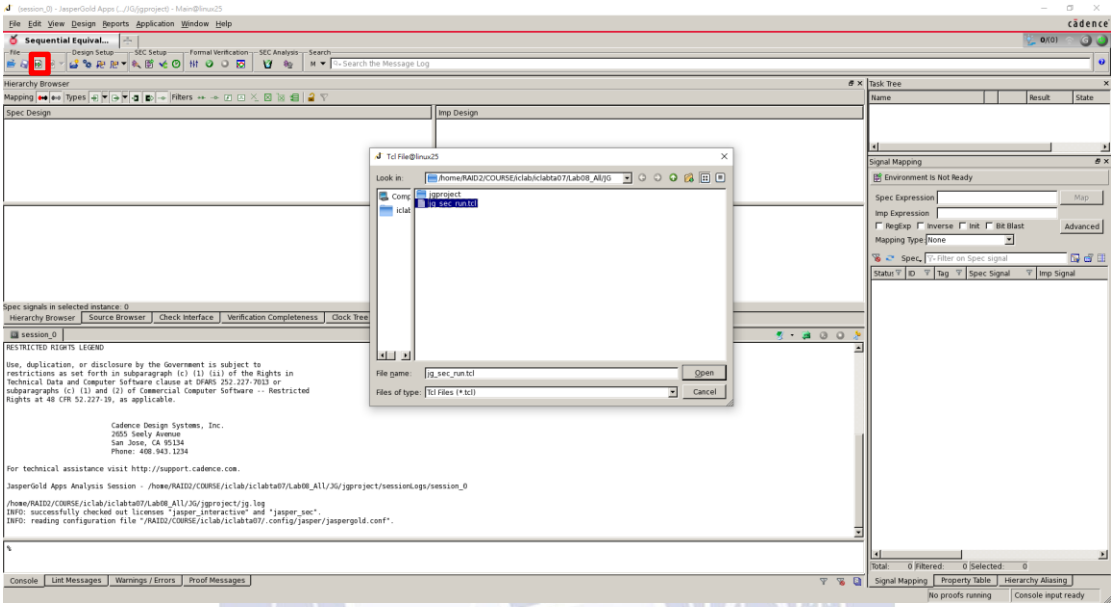


For PATTERN_CG.v, you need to give cg_en==1:

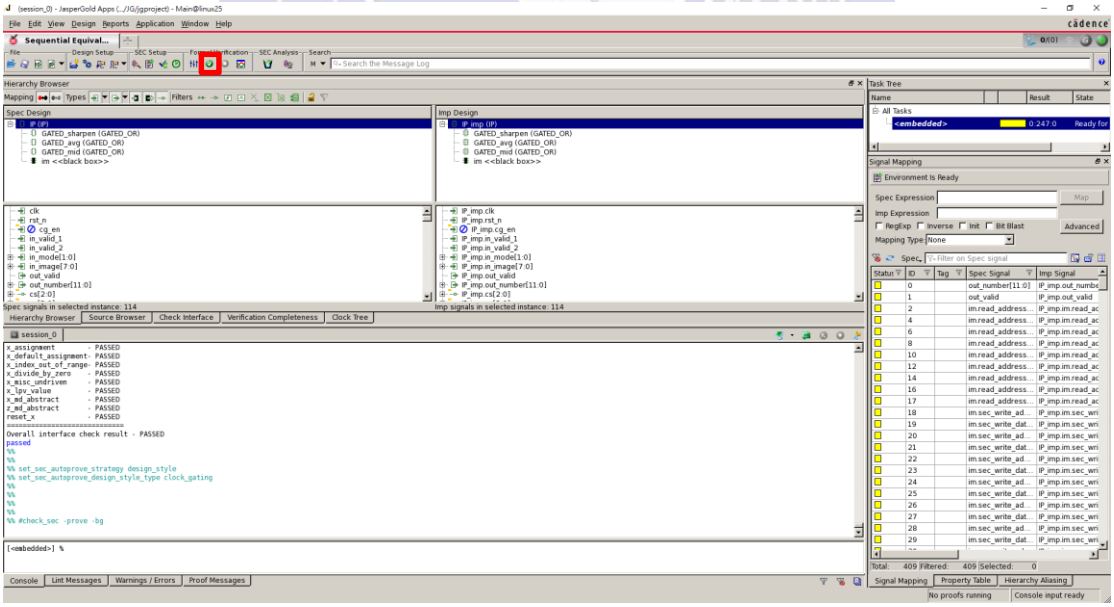


3. JasperGold SEC execution steps: In folder JG/

(1) "**./01_run**" 、"**./02_run**" or "**jg –sec &**" and click the "source" button.



(2) Then, click the "Prove the SEC Properties"



(3) Finally, check all properties pass.