2023 ICD Final Project
Image Processing Engine

Advisor : Yi-Chang Lu

TA: r10943004 黃士豪
r10943013 沈哲瑋
r11943129 林和謙
r11943131 林克帆

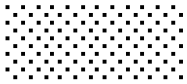
作業規劃

- **Final Project**
  - 1~3人一組
  - 請於 5/16(二) 20:00 前完成分組表單，逾時視為1人一組
  - 每組一個人填即可
  - 70% 基本分，30% 排名分(AT^2)
  - Deadline: **6/16(五) 13:00**
  - 註: 6/6(二) 期末考

Outline

- **Problem Description**
- **Specifications**
- **Function Waveform**
- **Function Description**
- **Files**
- **Simulation Scripts**
- **Simulation Results**
- **Homework Requirements**
- **Deadline**
- **Appendix**

Problem Description

- **Image Processing Engine**
  - RTL ~ APR
  - input: 16x16x1 image →  → output

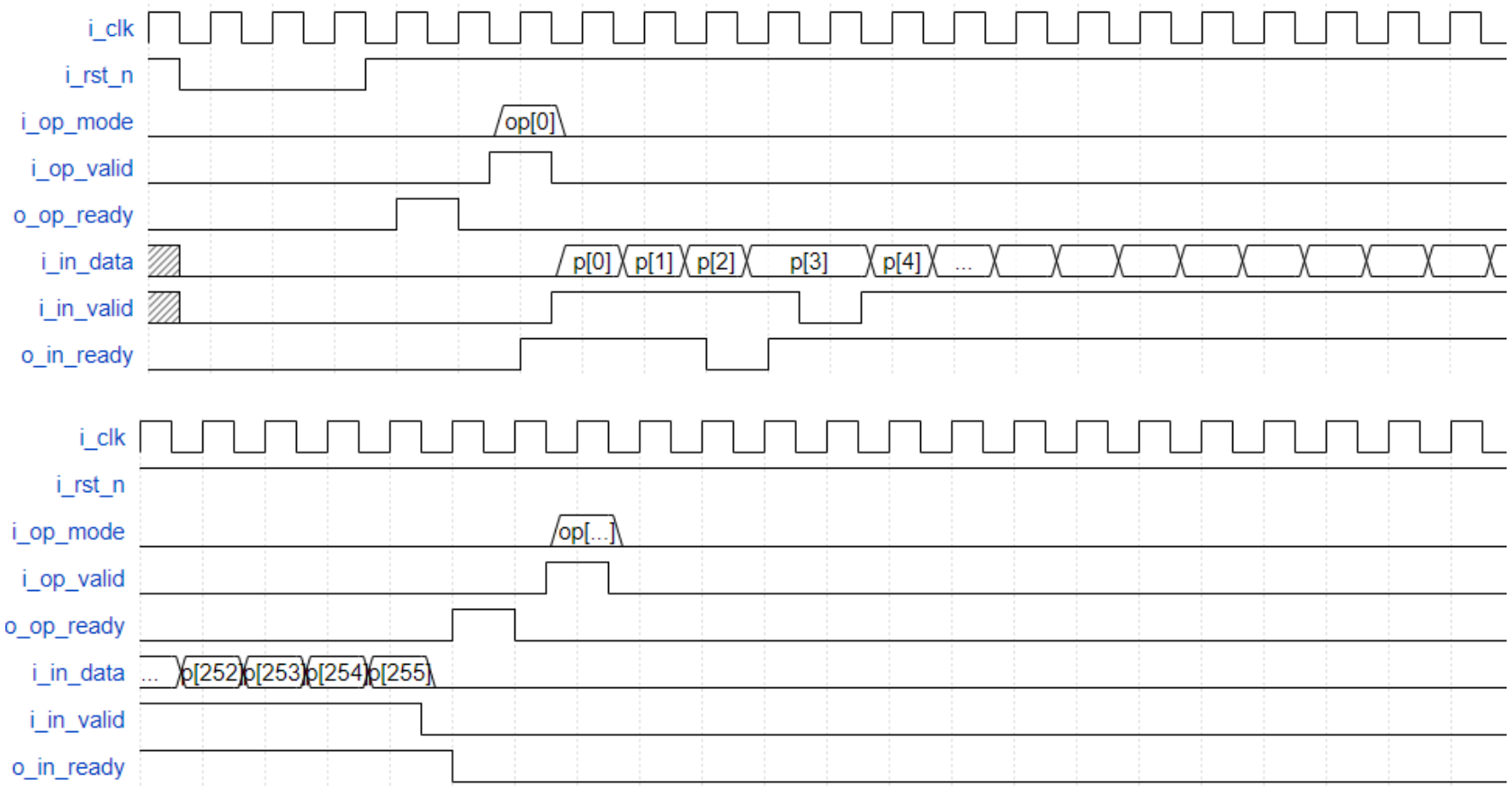
padding, image filtering, displaying triangle…

Specifications

Name	I/O	Width	Description
i_clk	Input	1	Clock signal. This design is positive-edge-triggered.
i_rst_n	Input	1	Active low asynchronous system reset signal.
i_op_valid	Input	1	If *i_op_valid* is set to high, the transferred *i_op_mode* is valid.
i_op_mode	Input	4	Operation mode for processing.
o_op_ready	Output	1	When *o_op_ready* is set to high, the testbench will transfer the next operation.
i_in_valid	Input	1	If *i_in_valid* is set to high, the transferred *i_in_data* is valid.
i_in_data	Input	8	Input pixel data, an unsigned value.
o_in_ready	Output	1	When *o_in_ready* is set to high, the testbench will transfer the next input pixel data. Only valid for *i_op_mode* = 4'b0000.
o_out_valid	Output	1	Set *o_out_valid* to high if your *o_out_data* is valid.
o_out_data	Output	8	Output pixel data, an unsigned value.

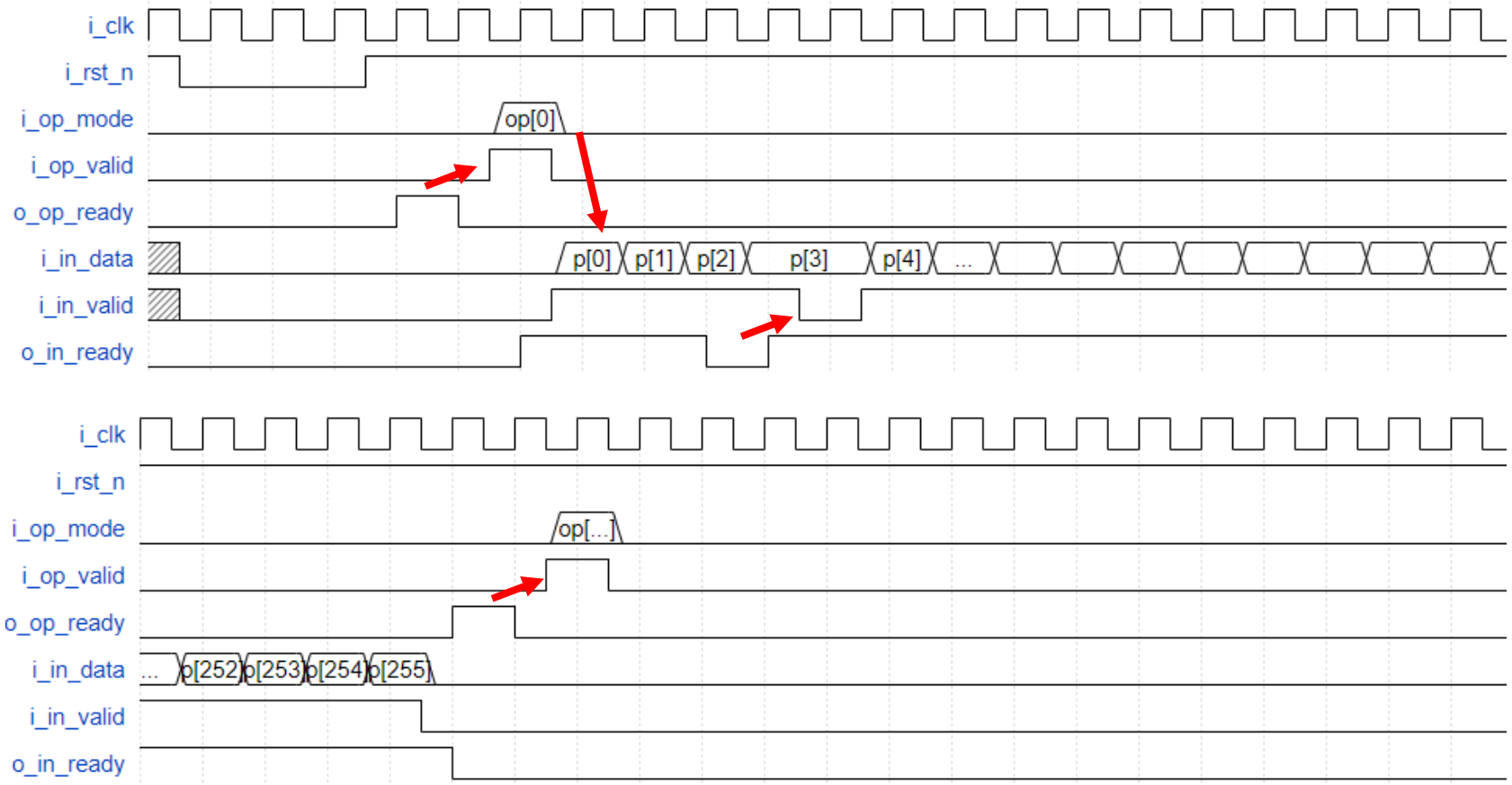
Function Waveform

- **load image (first operation after reset)**



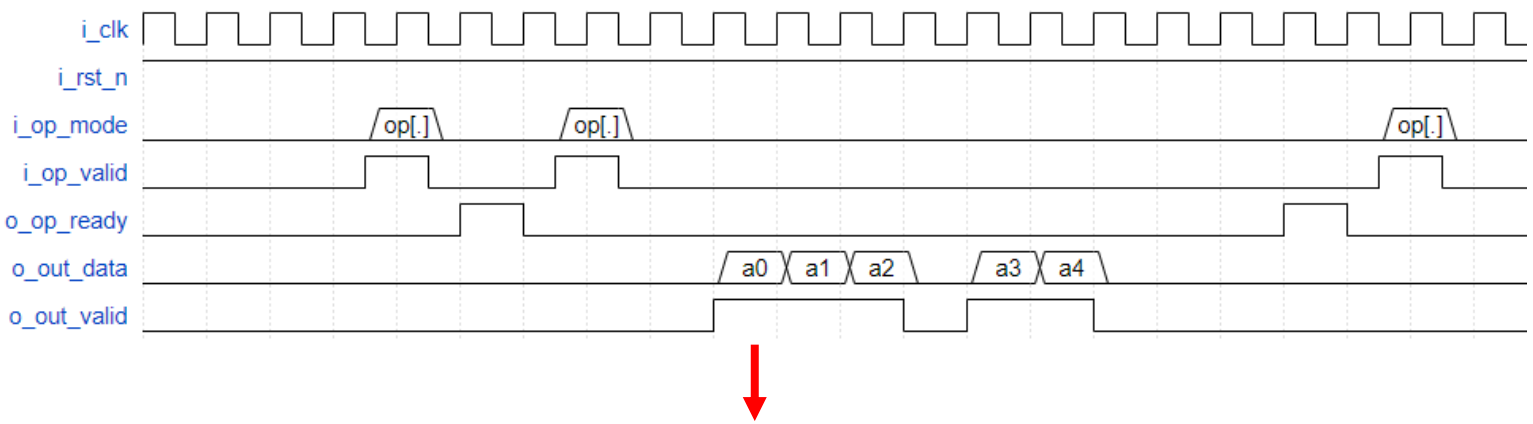
Function Waveform

- **load image (first operation after reset)**



Function Waveform

- **other operations**



testbench will check the correctness

Function Description

op_mode	Description	Need to display
4'b0000	load image	No
4'b0001	right shift	No
4'b0010	left shift	No
4'b0011	up shift	No
4'b0100	down shift	No
4'b0101	kernel size up	No
4'b0110	kernel size down	No
4'b0111	max in the kernel	Yes
4'b1000	min in the kernel	Yes
4'b1001	median in the kernel	Yes
4'b1010	blur	Yes
4'b1011	record position1	No
4'b1100	record position2	No
4'b1101	record position3	No
4'b1110	display triangle	Yes

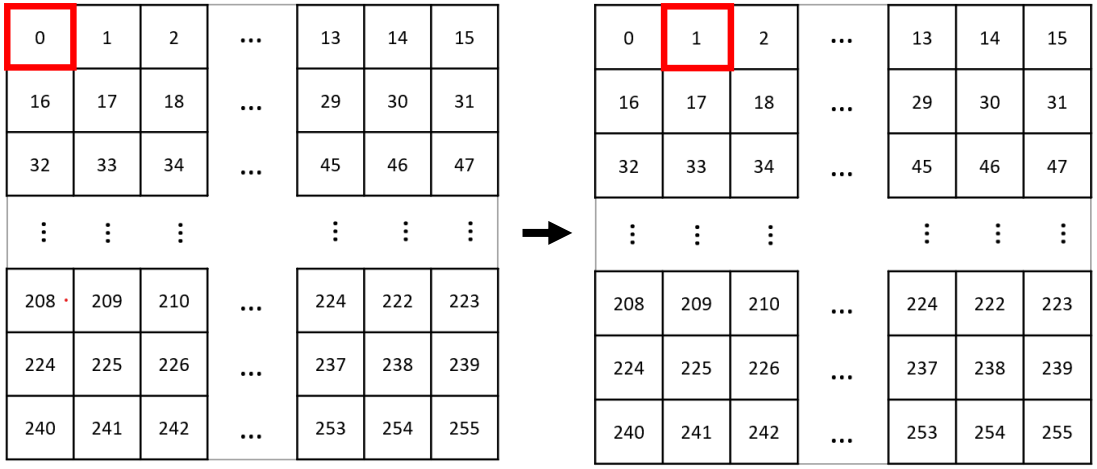
Function Description

- **load image**
  - An 16x16x1 image will be loaded for 256 cycles in raster-scan order.
  - The size of each pixel is 8 bits.

0	1	2	⋯	13	14	15
16	17	18	⋯	29	30	31
32	33	34	⋯	45	46	47
⋮	⋮	⋮		⋮	⋮	⋮
208	209	210	⋯	224	222	223
224	225	226	⋯	237	238	239
240	241	242	⋯	253	254	255

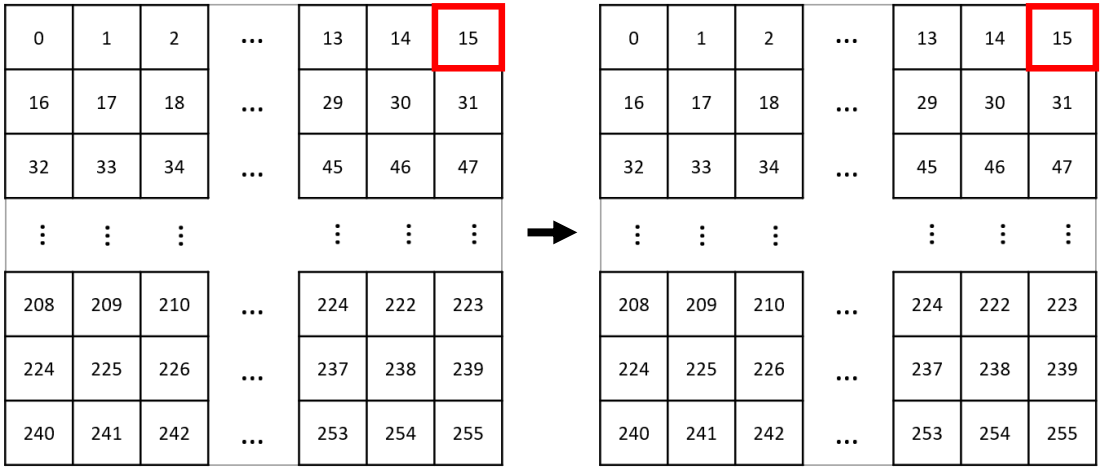
Function Description

- **origin shifting**
  - origin: start from (0, 0)
  - e.g. right shift



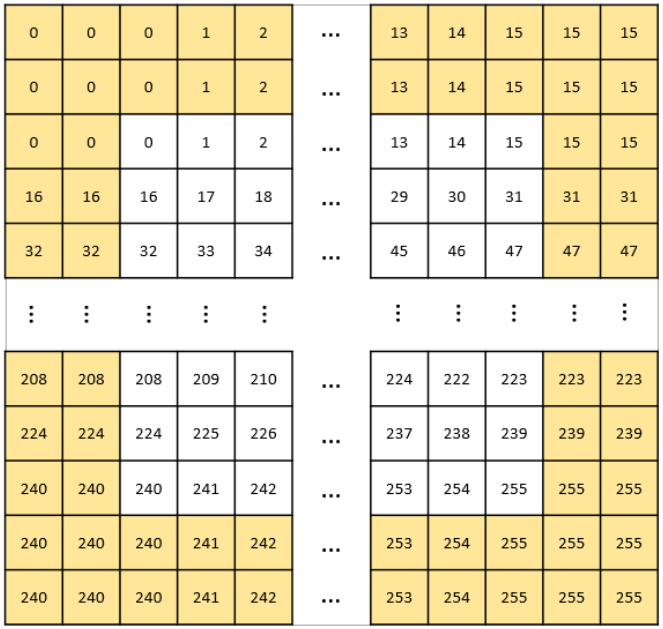
Function Description

- **origin shifting**
  - If output of display exceeds the image boundary, retain the same origin point.



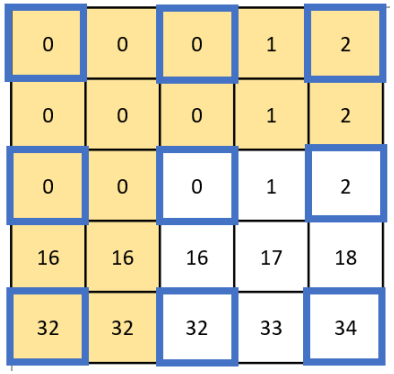
Function Description

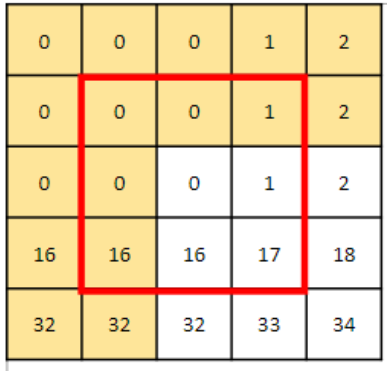
- **padding**
  - If the operation is out of range, you have to perform replicate padding.



Function Description

- **kernel size up / down**
  - kernel size
    - max/min/median
    - blur
  - sparse / dense 3x3 range





Function Description

- **blur**
  - You have to perform convolution with a 3x3 kernel.
  - ~~The weights in each channel are identical.~~
  - The accumulation results should be rounded (四捨五入) to integer.
  - The values of original pixels will not be changed.

Kernel

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

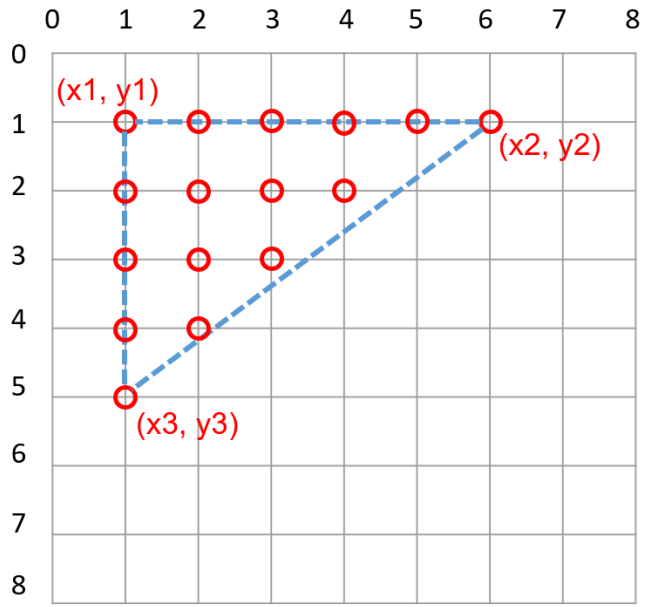
Function Description

- **max/min/median in the kernel**
  - You have to perform max/min/median filtering.
  - The values of original pixels will not be changed.

0	0	0	1	2
0	0	0	1	2
0	0	0	1	2
16	16	16	17	18
32	32	32	33	34

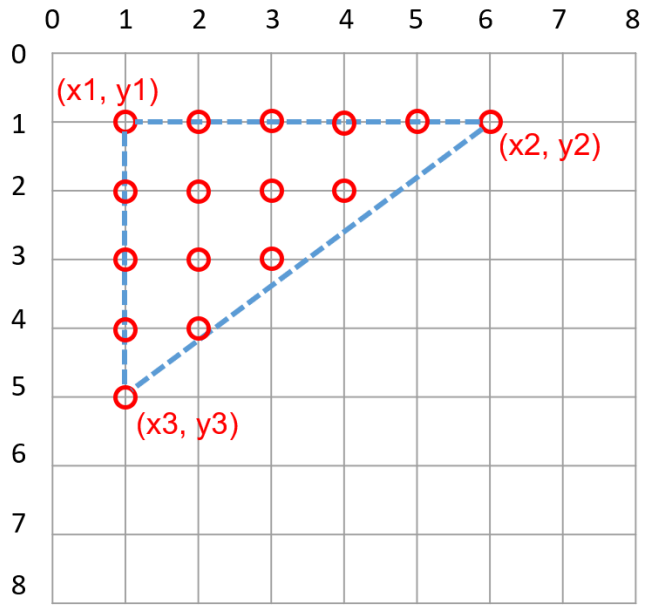
Function Description

- **record position**
  - You have to record the current position. Those positions will be considered as the vertices of a right triangle.
  - The three vertices of this triangle are input as (x1,y1), (x2,y2), and (x3,y3), and they satisfy the following relationship.
    - 0 <= x1 = x3 < x2 <= 15
    - 0 <= y1 = y2 < y3 <= 15



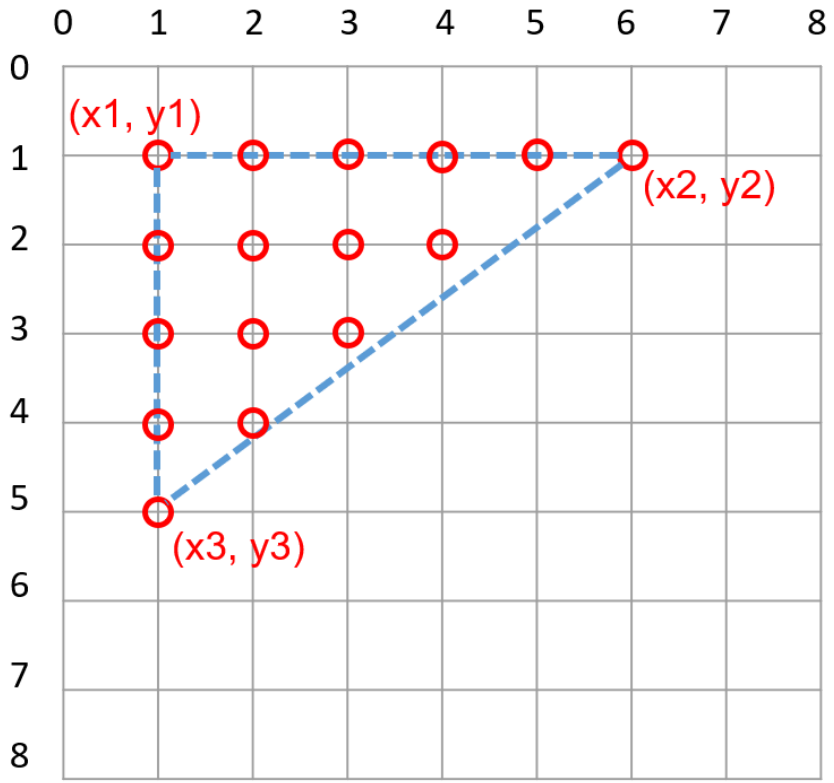
Function Description

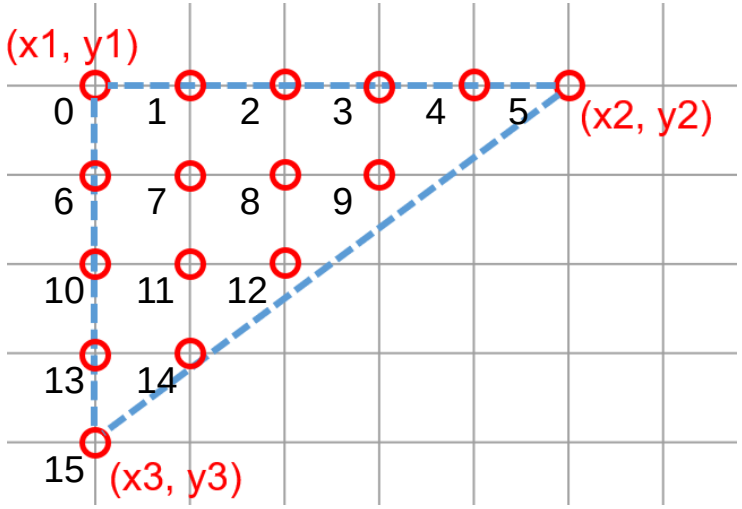
- **display triangle**
  - The following figure illustrates an example of the coordinates of a right-angled triangle. Referring to figure, the three vertex coordinates of the right-angled triangle (x1, y1), (x2, y2), and (x3, y3) are (1, 1), (6, 1), and (1, 5) respectively.
  - Please display the pixel value in raster-scan order for every pixel in the triangle.



Function Description

- **display triangle**
  - raster-scan order





Files / Simulation Scripts / Results

- **refer to ICD_Final.pdf**
  - 00_TESTBED
  - 01_RTL
  - 02_SYN
  - 03_APR
  - hierarchy 0x_xxx is not needed in the submitted files
- **4 patterns available**

Homework Requirements

a. RTL pass (30%)

b. Gate-level pass (with no latch) (20%)

c. Post layout pass (with no error) (10%)

1) With no error after verify geometry
2) Power ring: width = 2, ring number ≥ 2
3) Power stripe: width = 1, stripe number ≥ 1

d. Report (10%)

- c. fail: the score of your report will be determined based on its completeness.
- c. pass: write a simple report

Homework Requirements

e. APR performance passing testbench 3 (30%)

- Cost = Area * Time^2 (um^2 * ns^2)
  (A design with a lower cost is considered a better design.)

- Area = Total area of Core in summaryReport.rpt (see Appendix2)
  Time = Total time of testbench 3 (see Appendix2)

- The performance score is according to your ranking:
  1st place to 6th place: 30, 28, 26, 24, 22, 20 points
  7th place to the last place: 19, 18, 17, 16, … (and so on)

- If you don't pass requirement c, you will get 0 point in the performance score.

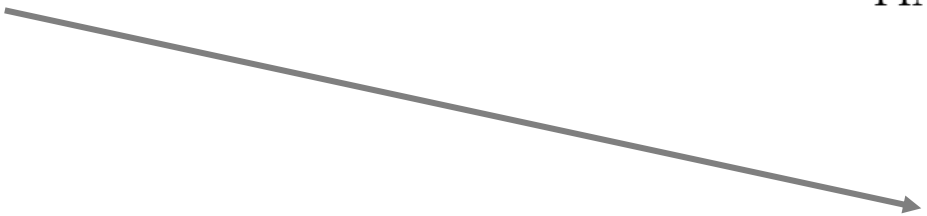
Homework Requirements

（組別之後會公布）

f. Compress the files to FINAL_G_xx.zip and submit it.

– hierarchy 0x_xxx is not needed in the submitted files

- mkdir FINAL_G_xx
- cp oooo FINAL_G_xx/



FINAL_G_xx.zip
FINAL_G_xx/
- core.v
- core_syn.v
- core_syn.sdf
- core_syn.ddc
- core_area.txt
- core_timing.txt
- core_power.txt
- core_apr.v
- core_apr.sdf
- summaryReport.rpt
- report_G_xx.pdf

- zip -r FINAL_G_xx.zip FINAL_G_xx/

Deadline

- **Deadline: 06/16 (Fri) 13:00**
- **Please submit the .zip file to NTU Cool**
  - Submitted by one person in the group.
- **If there is any problem, feel free to contact TAs by email. (Do not forget to add [積體電路設計] in your email title.)**
  - For verilog problems:
    - 黃士豪 r10943004@ntu.edu.tw
    - 沈哲瑋 r10943013@ntu.edu.tw
  - For algorithm problems:
    - 林和謙 r11943129@ntu.edu.tw
    - 林克帆 r11943131@ntu.edu.tw