Lab6:Canny Edge Detection(CED)

Instructor: Lih-Yih Chiou

Speaker: Frank

Date: 2024/03/28

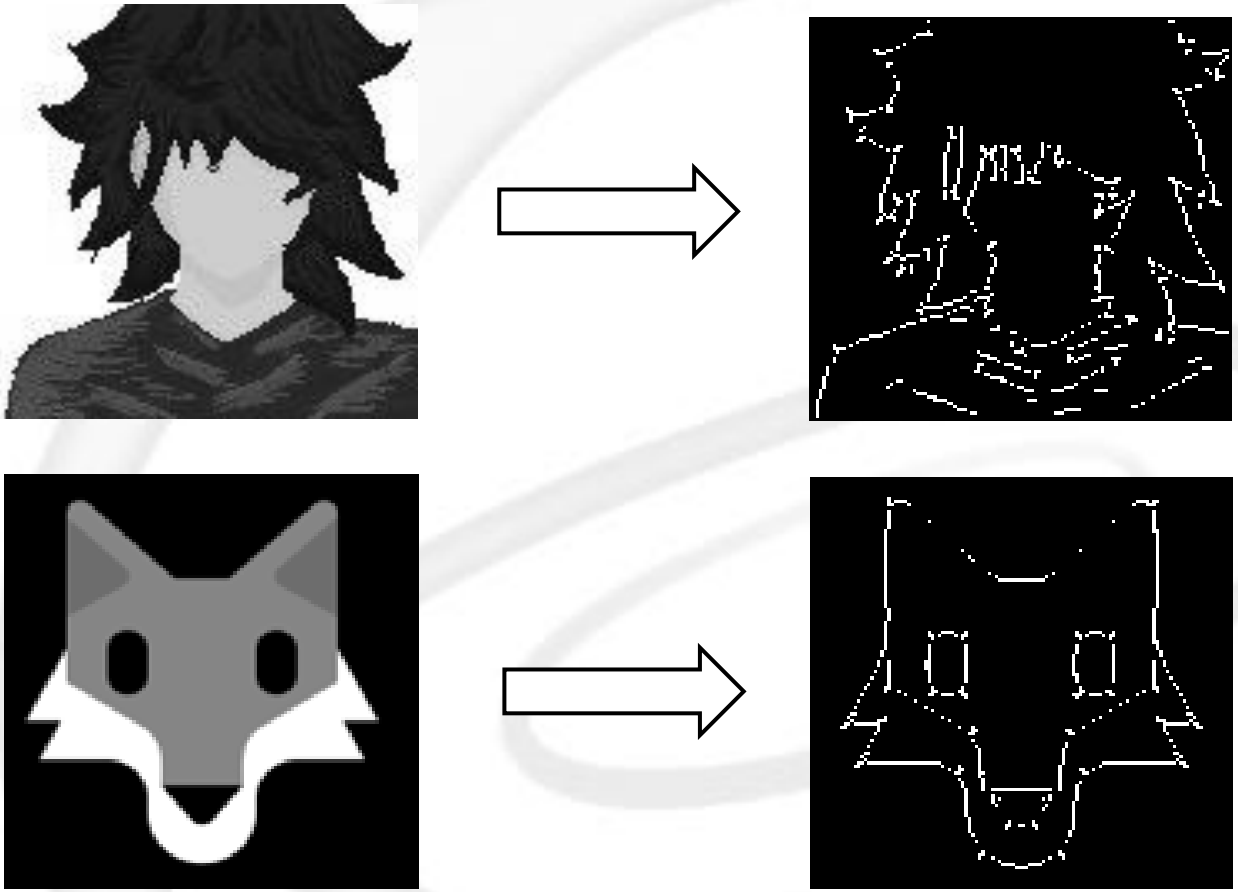
Outline

- Introduce to Canny edge detection
- Hardware description
  - Block diagram
  - I/O Information
- Lab6 Implementation
  - Finding the gradient magnitude and direction
  - Non-maximum suppression
  - Hysteresis thresholding
  - Memory read/write operation
- Criteria
  - Simulation Result
  - Grading policy
  - Requirement & file format

Introduce to Canny Edge Detection

- Canny Edge Detection is a technique used in image processing to detect edges, which are regions of rapid intensity changes in the image.
- Proposed by John Canny in 1986.
- Three main steps:
  - 1. Finding the gradient :
    - Magnitude
    - Direction
  - 2. Non-maximum suppression
  - 3. Hysteresis thresholding

Introduce to Canny Edge Detection

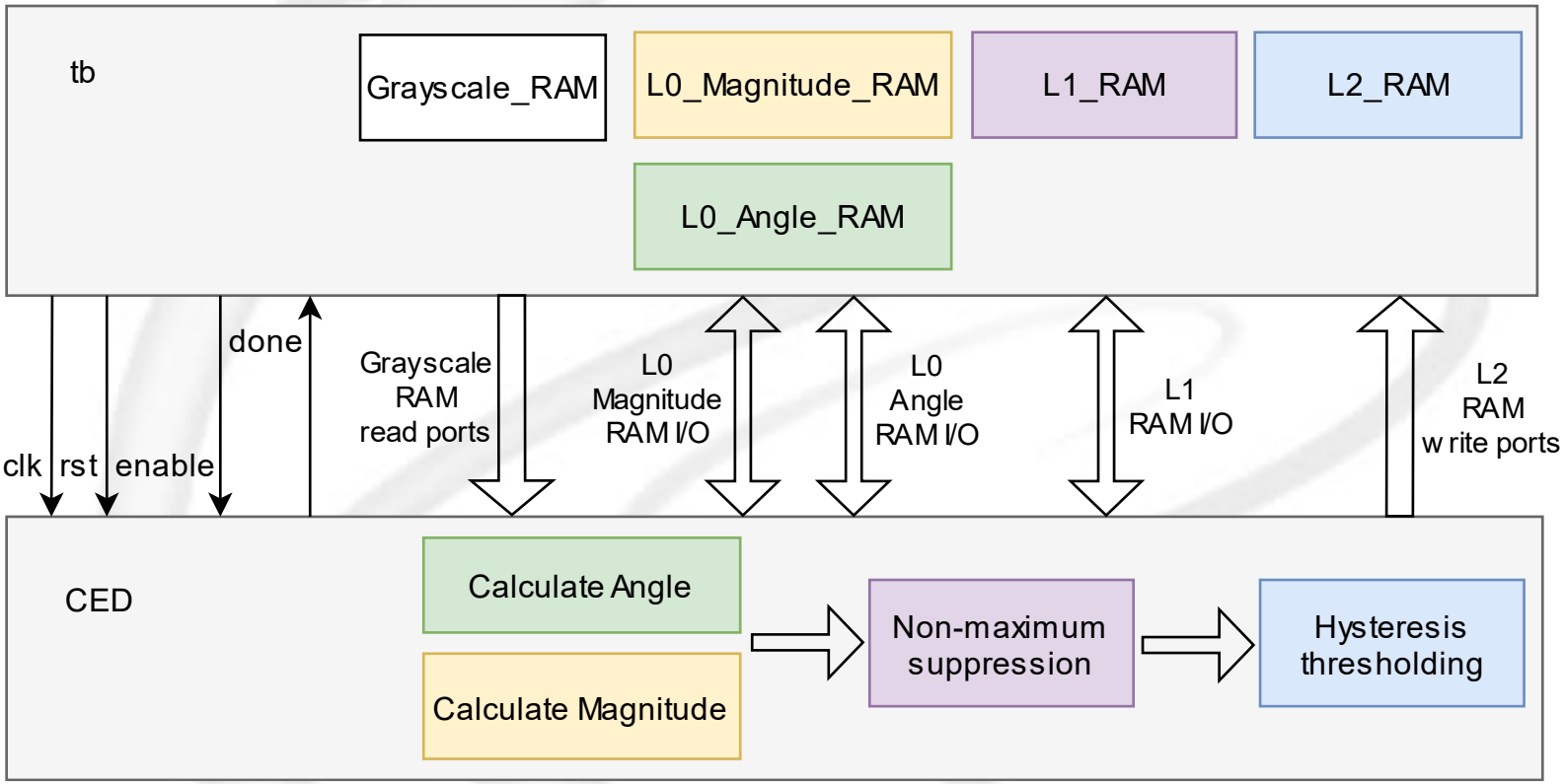


Input image

Output image

Hardware description

Block Diagram



Hardware description

I/O Information

Signal	I/O	length	Desc.
clk	I	1	positive-edged triggered
rst	I	1	asynchronous positive-edged triggered
enable	I	1	enable signal to start processing
ird	O	1	Active high read enable signal for Grayscale_RAM
iaddr	O	14	Address for Grayscale_RAM
idata	I	8	Read data from Grayscale_RAM，8-bit，unsigned
cwr_mag_0	O	1	Active high write enable signal for L0_Magnitude_RAM
cdata_mag_wr0	O	13	Write data to L0_Magnitude_RAM，13-bit，signed
crd_mag_0	O	1	Active high read enable signal for L0_Magnitude_RAM
cdata_mag_rd0	I	13	Read data from L0_Magnitude_RAM，13-bit，signed
caddr_mag_0	O	14	Address for L0_Magnitude_RAM

Hardware description

I/O Information

Signal	I/O	length	Desc.
cwr_ang_0	O	1	Active high write enable signal for L0_Angle_RAM
cdata_ang_wr0	O	13	Write data to L0_Angle_RAM，13-bit，signed
crd_ang_0	O	1	Active high read enable signal for L0_Angle_RAM
cdata_ang_rd0	I	13	Read data from L0_Angle_RAM，13-bit，signed
caddr_ang_0	O	14	Address for L0_Angle_RAM
cwr1	O	1	Active high write enable signal for L1_RAM
cdata_wr1	O	13	Write data to L1_RAM，13-bit，signed
crd1	O	1	Active high read enable signal for L1_RAM
cdata_rd1	I	13	Read data from L1_RAM，13-bit，signed
caddr_1	O	14	Address for L1_RAM

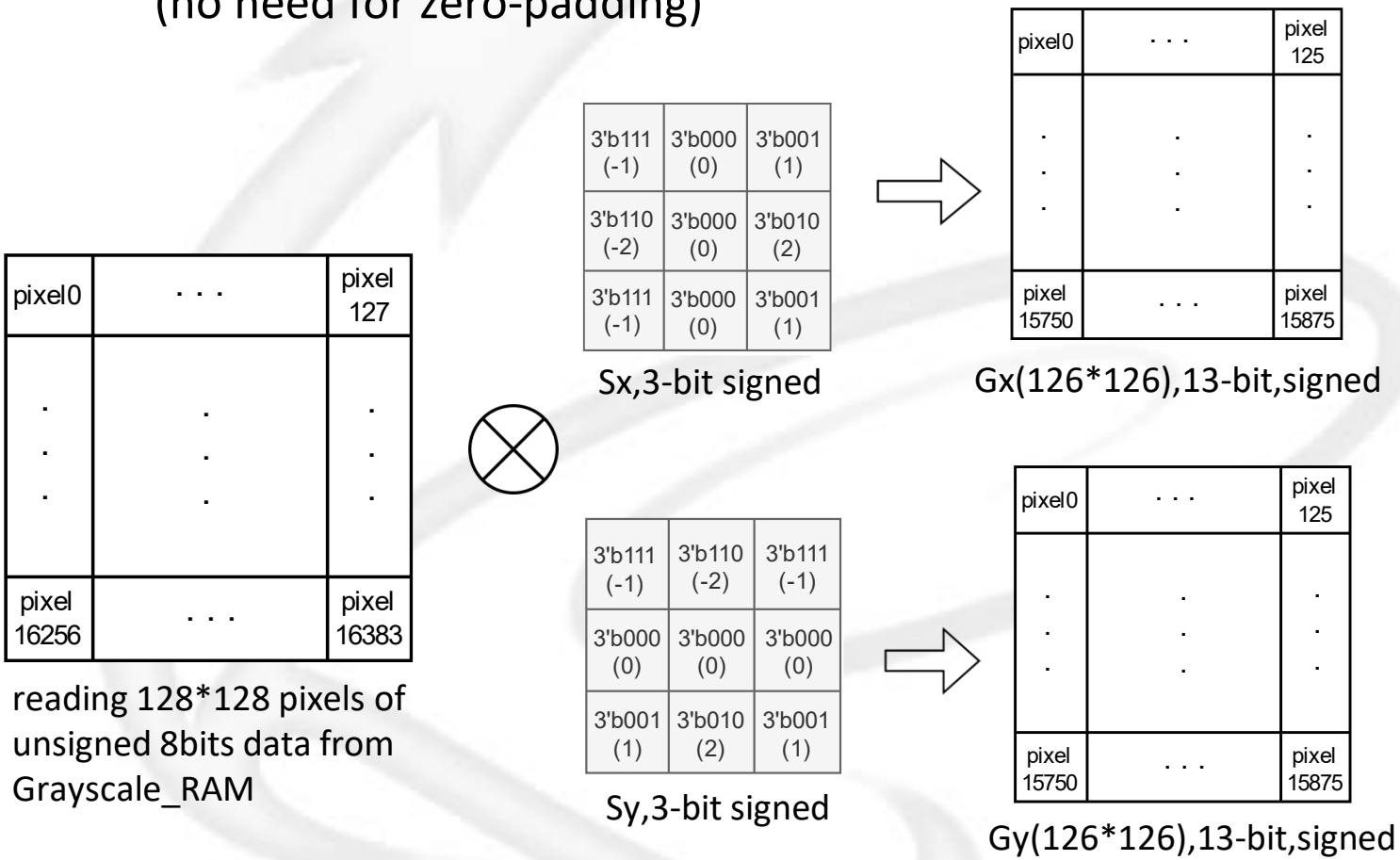
Hardware description

I/O Information

Signal	I/O	length	Desc.
cwr2	O	1	Active high write enable signal for L2_RAM
cdata_wr2	O	13	Write data to L2_RAM，13-bit，signed
caddr_2	O	14	Address for L2_RAM
done	O	1	Finish signal

Implementation

- Layer0:Finding the gradient magnitude and direction
  - Convolution with sobel operator Sx and Sy to get Gx and Gy. (no need for zero-padding)



Implementation

- Layer0:Finding the gradient magnitude and direction
  - Convolution implementation ex:

    1*(-1)+2*0+3*1+0*(-2)+1*0+2*2+3*(-1)+0*0+1*1=4

    2*(-1)+3*0+0*1+1*(-2)+2*0+3*2+0*(-1)+1*0+2*1=4

    1*(-1)+2*0+3*1+0*(-2)+1*0+0*2+0*(-1)+4*0+1*1=3
  - Data format : 8-bit unsigned * 3-bit signed = 13-bit signed.

(Grayscale data) (Sx,Sy) (Gx,Gy)

1	2	3	0	1
0	1	2	3	0
3	0	1	2	3
2	3	0	1	0
2	3	0	4	1

example

⊗

-1	0	1
-2	0	2
-1	0	1

=

4	4	-4
-4	4	2
-8	-1	3

Implementation

- Layer0:Finding the gradient magnitude and direction
  - Convolution implementation ex:

    1*(-1)+2*0+3*1+0*(-2)+1*0+2*2+3*(-1)+0*0+1*1=4

    2*(-1)+3*0+0*1+1*(-2)+2*0+3*2+0*(-1)+1*0+2*1=4

    1*(-1)+2*0+3*1+0*(-2)+1*0+0*2+0*(-1)+4*0+1*1=3
  - Data format : 8-bit unsigned * 3-bit signed = 13-bit signed.

(Grayscale data) (Sx,Sy) (Gx,Gy)

1	2	3	0	1
0	1	2	3	0
3	0	1	2	3
2	3	0	1	0
2	3	0	4	1

example

⊗

-1	0	1
-2	0	2
-1	0	1

=

4	4	-4
-4	4	2
-8	-1	3

Implementation

- Layer0:Finding the gradient magnitude and direction
  - Convolution implementation ex:

    1*(-1)+2*0+3*1+0*(-2)+1*0+2*2+3*(-1)+0*0+1*1=4

    2*(-1)+3*0+0*1+1*(-2)+2*0+3*2+0*(-1)+1*0+2*1=4

    1*(-1)+2*0+3*1+0*(-2)+1*0+0*2+0*(-1)+4*0+1*1=3
  - Data format : 8-bit unsigned * 3-bit signed = 13-bit signed.

(Grayscale data) (Sx,Sy) (Gx,Gy)

1	2	3	0	1
0	1	2	3	0
3	0	1	2	3
2	3	0	1	0
2	3	0	4	1

example

⊗

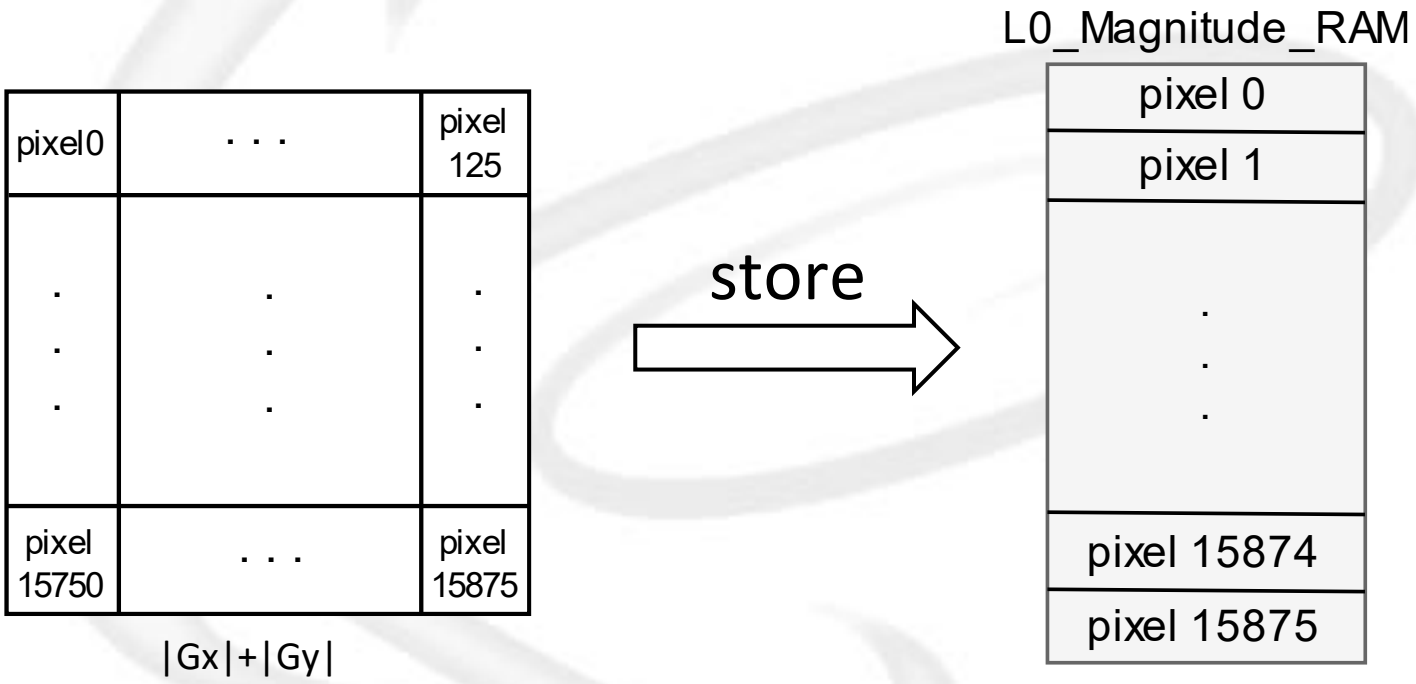
-1	0	1
-2	0	2
-1	0	1

=

4	4	-4
-4	4	2
-8	-1	3

Implementation

- Layer0:Finding the gradient magnitude
  - |Gx|+|Gy| to get magnitude.
  - Storing the result to L0_Magnitude_RAM.
  - Data format : 13-bit signed + 13-bit signed = 13-bit signed.
    (|Gx|) (|Gy|) (cdata_mag_wr0)



Implementation

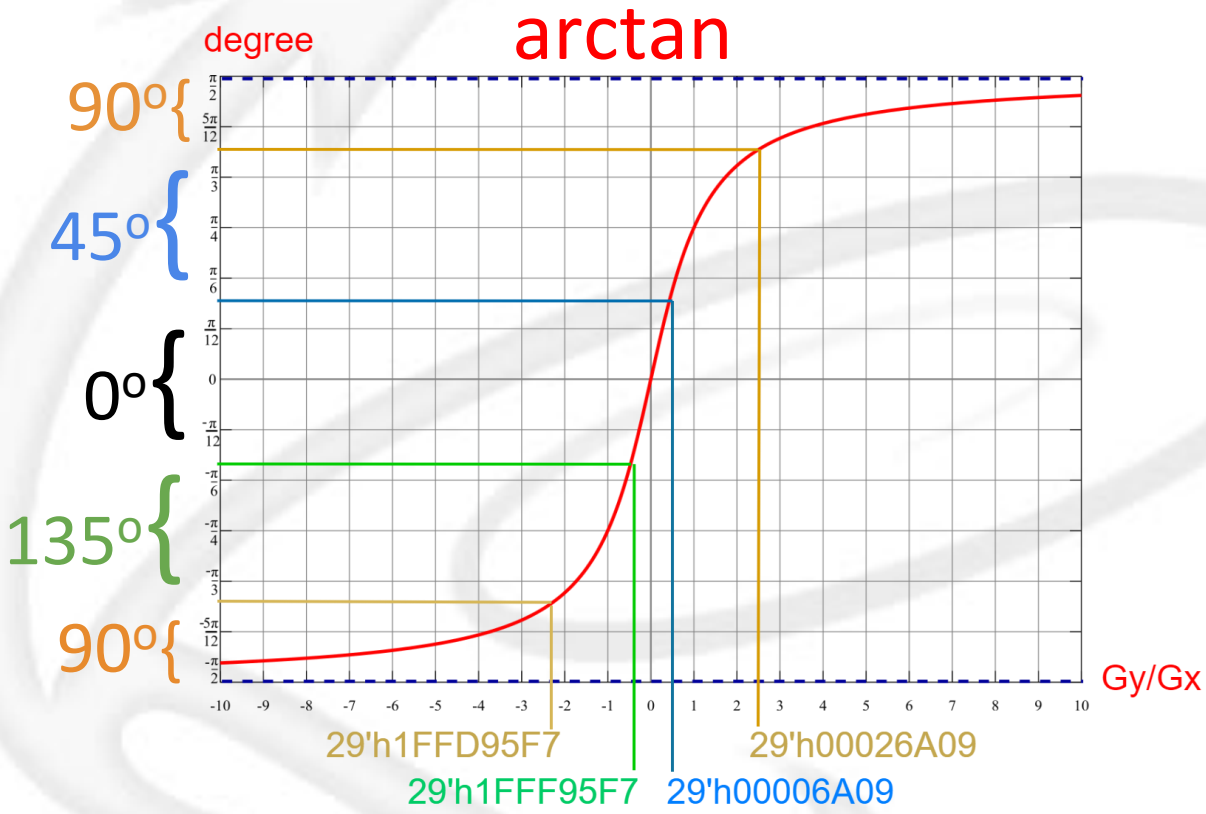
Layer0:Finding the gradient direction:arctan(Gy/Gx)

- Fixed point division : 13-bit integer + 16-bit fraction
  - Gy should left-shifted 16-bit for fractional part.
  - (Gy << 16) / Gx = temp_result.
  - temp_result is a 29-bit fixed-point number comprising a 13-bit integer part and a 16-bit fractional part.
- The angle is rounded to one of four angles : 0^o ,45^o ,90^o ,135^o :
  - temp_result<29'h00006A09 || temp_result>29'h1FFF95F7 maps to 0^o.
  - temp_result>29'h00006A09 && temp_result<29'h00026A09 maps to 45^o.
  - temp_result>29'h00026A09 || temp_result<29'h1FFD95F7 maps to 90^o.
  - temp_result<29'h1FFF95F7 && temp_result>29'h1FFD95F7 maps to 135^o.

NOTE : If Gx=0,it maps to 90^o

Implementation

- Layer0:Finding the gradient direction :arctan(Gy/Gx)
  - Storing the angle result(13'd0,13'd45,13'd90,13'd135) of each pixel to L0_Angle_RAM.



Implementation

Layer1:Non-maximum suppression

- Comparing the magnitude of the current pixel with neighboring pixels along the gradient direction, if the magnitude of the current pixel is the largest, the value will be preserved. Otherwise, the value will be set to 0.
  (Outermost pixels set to 0)

step1

pixel0	pixel1	pixel2	. . .	pixel 125
pixel 126	45°	pixel 128	. . .	pixel 251
pixel 252	pixel 253	pixel 254	. . .	pixel 377
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
pixel 15750	pixel 15751	pixel 15752	. . .	pixel 15875

Data read from L0_Angle_RAM

step2

pixel0	pixel1	120	. . .	pixel 125
pixel 126	80	pixel 128	. . .	pixel 251
50	pixel 253	pixel 254	. . .	pixel 377
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
pixel 15750	pixel 15751	pixel 15752	. . .	pixel 15875

Data read from L0_Magnitude_RAM

step3

store

L1_RAM

pixel 0=0
pixel 1=0
.
.
.
pixel 127=0
.
.
.
pixel 15874=0
pixel 15875=0

Implementation

Layer1:Non-maximum suppression

- Comparing the magnitude of the current pixel with neighboring pixels along the gradient direction, if the magnitude of the current pixel is the largest, the value will be preserved. Otherwise, the value will be set to 0.
  (Outermost pixels set to 0)

step1

pixel0	pixel1	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	135°	pixel 129	. . .	pixel 251
pixel 252	pixel 253	pixel 254	pixel 255	. . .	pixel 377
. . .	. . .	. . .	. . .	. . .	. . .
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Angle_RAM

step2

pixel0	38	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	200	pixel 129	. . .	pixel 251
pixel 252	pixel 253	pixel 254	45	. . .	pixel 377
. . .	. . .	. . .	. . .	. . .	. . .
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Magnitude_RAM

step3

store

L1_RAM

pixel 0=0
pixel 1=0
. . .
pixel 128=200
. . .
pixel 15874=0
pixel 15875=0

Implementation

Layer1:Non-maximum suppression

- Comparing the magnitude of the current pixel with neighboring pixels along the gradient direction, if the magnitude of the current pixel is the largest, the value will be preserved. Otherwise, the value will be set to 0.
  (Outermost pixels set to 0)

step1

pixel0	pixel1	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	pixel 128	pixel 129	. . .	pixel 251
pixel 252	pixel 253	0°	pixel 255	. . .	pixel 377
. . .	. . .	. . .	. . .	. . .	. . .
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Angle_RAM

step2

pixel0	pixel1	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	pixel 128	pixel 129	. . .	pixel 251
pixel 252	30	70	45	. . .	pixel 377
. . .	. . .	. . .	. . .	. . .	. . .
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Magnitude_RAM

step3

store

L1_RAM

pixel 0=0
pixel 1=0
. . .
pixel 254=70
. . .
pixel 15874=0
pixel 15875=0

Implementation

Layer1:Non-maximum suppression

- Comparing the magnitude of the current pixel with neighboring pixels along the gradient direction, if the magnitude of the current pixel is the largest, the value will be preserved. Otherwise, the value will be set to 0.
  (Outermost pixels set to 0)

step1

pixel0	pixel1	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	pixel 128	pixel 129	. . .	pixel 251
pixel 252	pixel 253	pixel 254	90°	. . .	pixel 377
pixel 378	pixel 379	pixel 380	pixel 381	. . .	pixel 503
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Angle_RAM

step2

pixel0	pixel1	pixel2	pixel3	. . .	pixel 125
pixel 126	pixel 127	pixel 128	35	. . .	pixel 251
pixel 252	pixel 253	pixel 254	45	. . .	pixel 377
pixel 378	pixel 379	pixel 380	45	. . .	pixel 503
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
pixel 15750	pixel 15751	pixel 15752	pixel 15753	. . .	pixel 15875

Data read from L0_Magnitude_RAM

step3

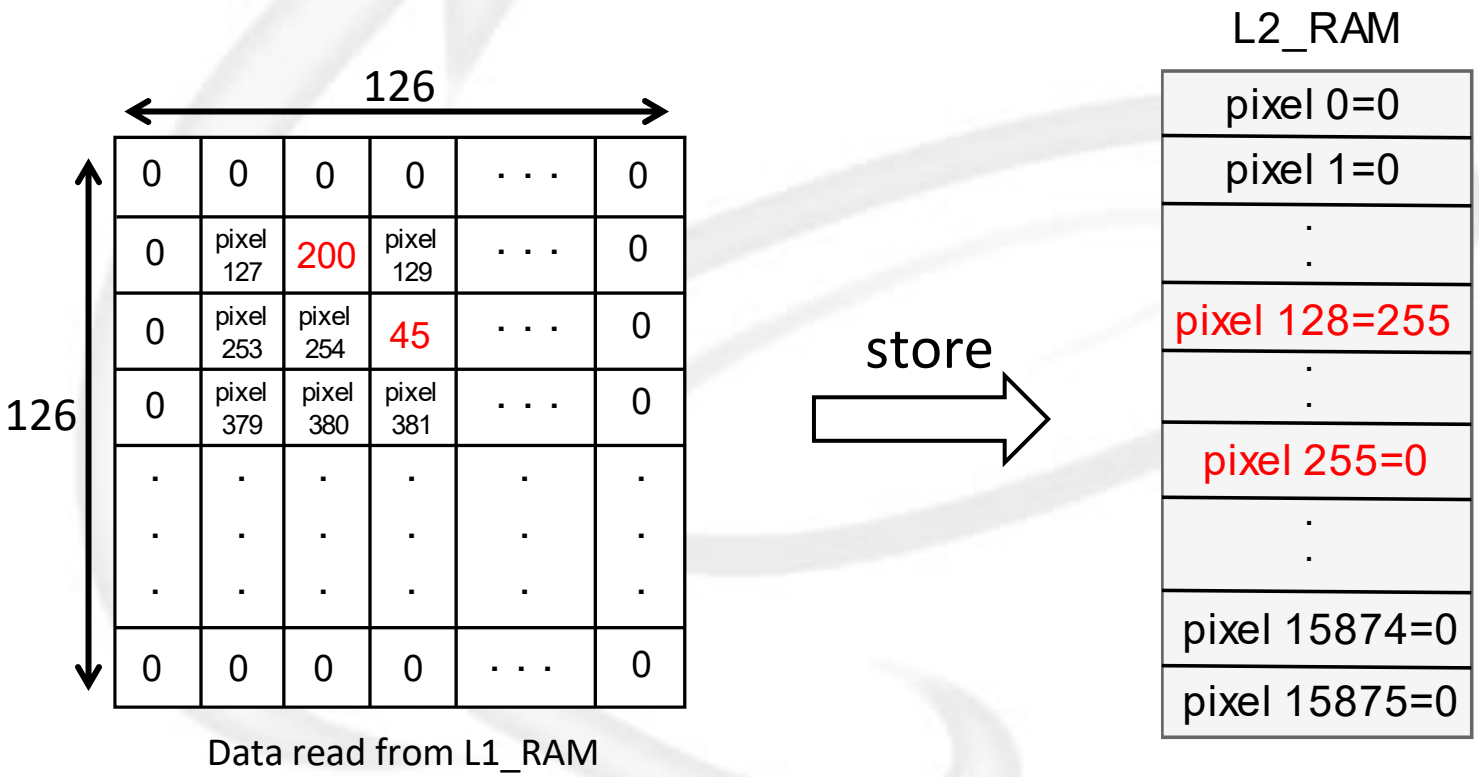
store

L1_RAM

pixel 0=0
pixel 1=0
.
.
.
pixel 255=45
.
.
.
pixel 15874=0
pixel 15875=0

Implementation

- Layer2:Hysteresis thresholding
  - If the magnitude of the current pixel >= 13'd100, it's marked as a strong edge pixel(set to 13'd255).
  - If the magnitude of the current pixel < 13'd50, it's marked as a weak edge pixel(set to 13'd0).



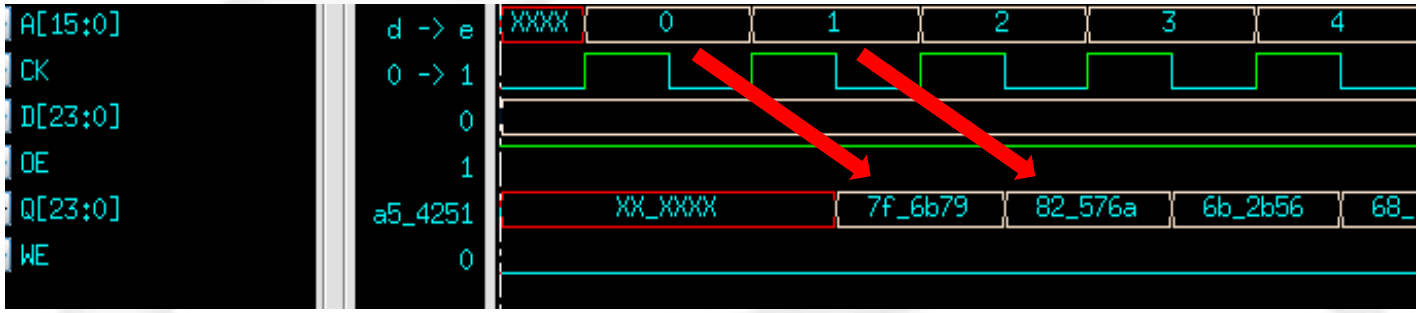
Implementation

Layer2:Hysteresis thresholding

- If the magnitude of the current pixel >=13'd50 and <13'd100, it checks whether there is a magnitude of the pixel >= 13'd100 around the current pixel. If so, the current pixel will be marked as a strong edge pixel(set to 13'd255). If not, it will be marked as a weak edge pixel(set to 13'd0).

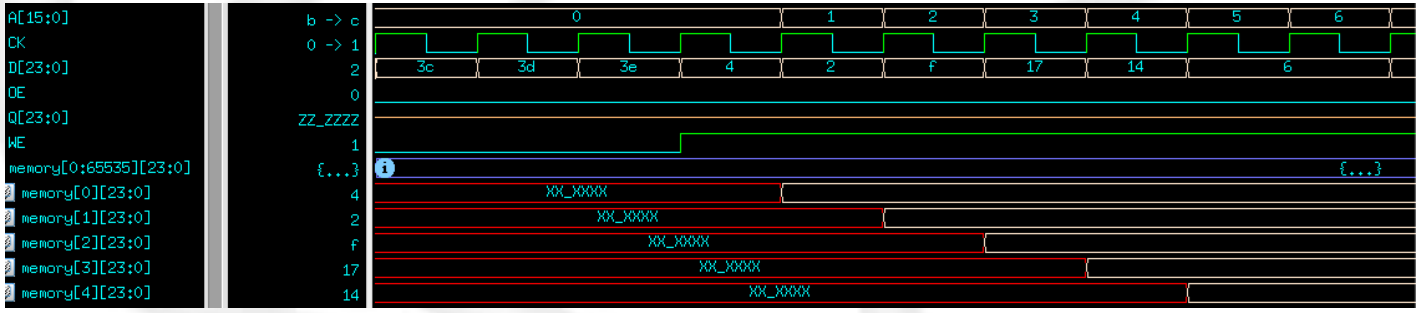
Implementation

- The timing information for Read/Write SRAM
  - Read operation(delay one cycle)



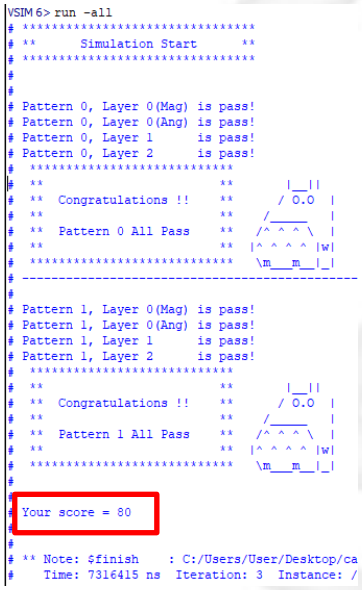
✓ The memory will output values on the negative edge, and you need to capture data on the positive edge

  - Write operation



Criteria

- Grading policy(100%)
  - Pattern1&2 Layer 0 Magnitude simulation pass (12%)
  - Pattern1&2 Layer 0 Angle simulation pass (13%)
  - Pattern1&2 Layer 1 simulation pass (25%)
  - Pattern1&2 Layer 2 simulation pass (30%)
  - Report (20%)



pattern1&2 All Pass

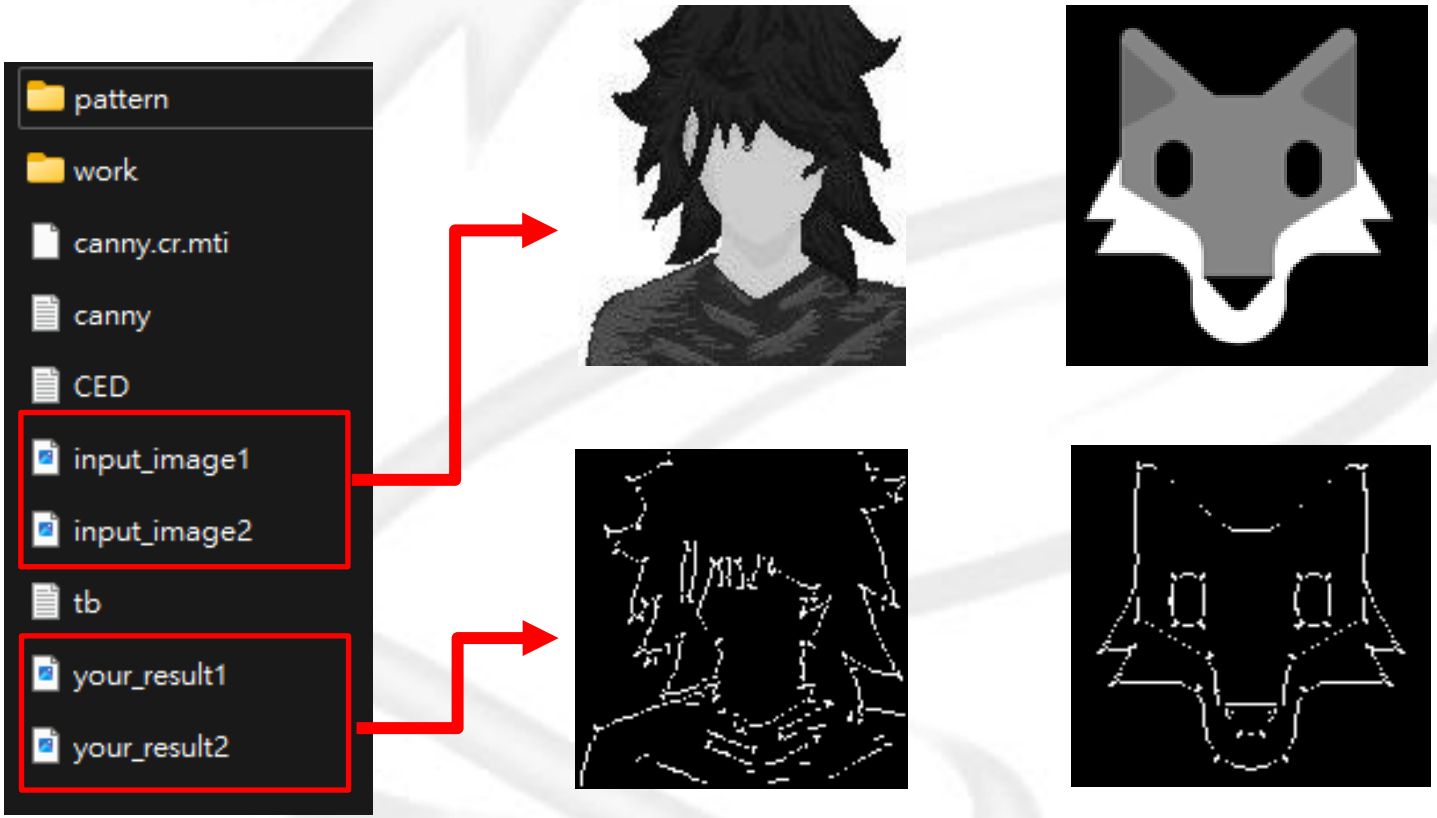
pattern1&2 Layer1,2 failed

pattern1 all pass but pattern2 failed

Criteria

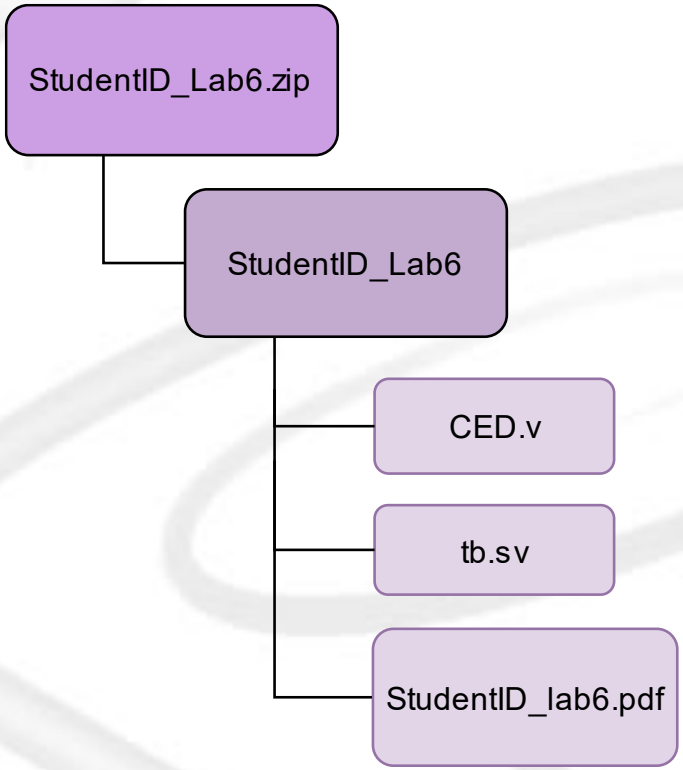
Simulation result visualization

- It will generate the input picture and your output result in a BMP file when your simulation is finished.



Requirement & file format

- File format
- Deadline:2024/04/11 8:59



Requirement & file format

Friendly reminder

- Please complete the assignment by your own, discussion with peers is recommended, but do not cheat.
- Warning! Any dishonesty found will result in zero grade.
- Warning! Any late submission will also receive zero.
- Warning! Please make sure that your code can be compiled in Modelsim, any dead body that we cannot compile will also receive zero.
- Warning! Please submit your work according to the specified file format, making sure not to include any unnecessary files. Any unnecessary file found, will lead to 10% deduction from the overall score.

LPHPLAB
VLSI Design LAB

Thanks for listening