2020 Digital IC Design

Homework 5: Image Edge Detect System

1 Introduction

Sobel operator is one of the operators in image processing, also known as Sobel-Federman operator or Sobe filter, which is often used to do edge detection in the field of image processing and computer vision. Technically, it is a discrete difference operator used to calculate the approximate value of the gradient of the image brightness function. Using this at any point in the image, the operation of the Sauber operator will produce the corresponding gradient vector or its norm. Conceptually, the Sauber operator is a small and integer filter that convolves the entire image in the horizontal and vertical directions, so it requires relatively few computing resources. On the other hand, for the frequency in the image Where the change is higher, the approximate value of the gradient obtained by it is relatively rough.

For this work, please implement an image edge detection system, use Gx and Gy to convolve the image to obtain the sobelX image and sobelY image, and then add sobelX image and sobelY image together and divide by 2 for the output image of sobelCombine.

2 Design Specifications

2.1 Block overview

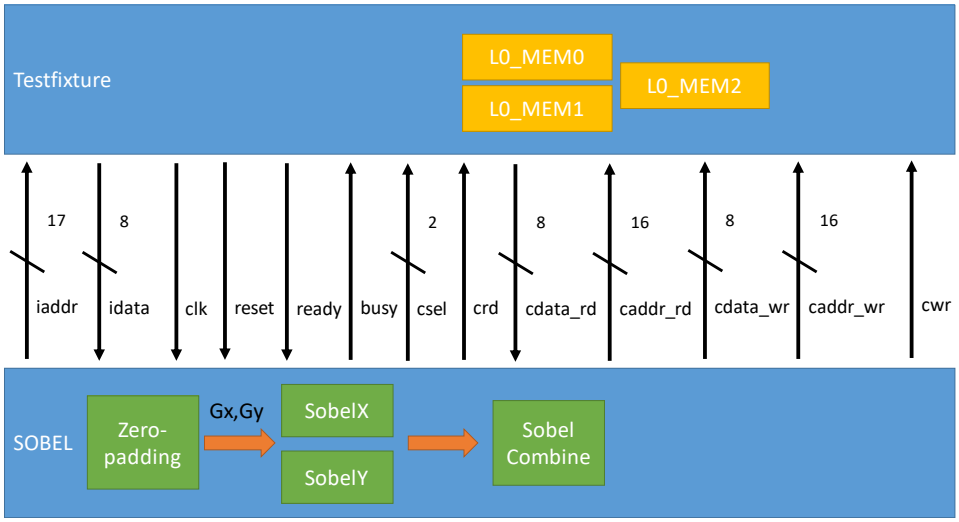


Fig. 1 – System operation flow

2.2 I/O Interface

Name	I/O	Width	Description
clk	I	1	System clock signal. This system is synchronized with the positive edge of the clock.
reset	I	1	High level NON-synchronous system reset signal
ready	I	1	The indication signal of the grayscale image is ready. When the signal is High, shows the grayscale image is prepared. At this time, SOBEL can start sending grayscale image address to testfixture to obtain the data.
busy	O	1	System busy indication signal. When SOBEL receives the ready signal as High and SOBEL is ready to start the action, the signal needs to be set to High, indicating that the module is ready to start requesting the input grayscale image data. After the calculation processing is completed and the result is written back to the testfixture, the signal is set to Low as the end of the action.
iaddr	O	17	The input grayscale image address signal. Indicate which grayscale image pixel data address you want to request.
idata	I	8	Input grayscale image pixel data signal, which is composed of 8bits unsigned integer. The testfixture sends the pixel data of the address of iaddr to SOBEL by this signal.
crd	O	1	The enable signal to read SOBEL operation output memory. When the positive edge of the clock is triggered and the signal is High, it shows the reading operation is to be performed. testfixture will read the data indicated by the caddr_rd address to cdata_rd.
cdata_rd	I	8	The CONV operation result memory read signal, which is composed of 8 bits unsigned integer (MSB). The testfixture sends the memory data to the SOBEL circuit.
caddr_rd	O	16	SOBEL operation output memory read address. The operation result of each layer of the SOBEL circuit uses this signal to indicate which address of the built-in output result memory in the testfixture will be read.
cwr	O	1	The enable signal to write SOBEL operation output

			memory. When the positive edge of the clock is triggered while this signal is High, it shows the write operation is to be performed. testfixture will write the contents of cdata_wr to the address indicated by caddr_wr.
cdata_wr	O	8	The SOBEL operation output memory write signal is composed of 8 bits unsigned integer (MSB). The operation result of the SOBEL circuit is output to testfixture using this signal.
caddr_wr	O	16	SOBEL operation output memory is written into the address. The operation result of the SOBEL circuit uses this signal to indicate to which address of the built-in output memory in the testfixture to be written.
csel	O	2	The SOBLE operation result is written to / read from the memory selection signal. This signal indicates which operation output memory of the SOBEL circuit the current write/read data is. described as follows: 2'b00: indicates that no memory is selected. 2'b01: Write/read Sobel X results. 2'b10: Write/read Sobel Y results. 2'b11: Write/read Sobel combine results.

2.3 Function Description

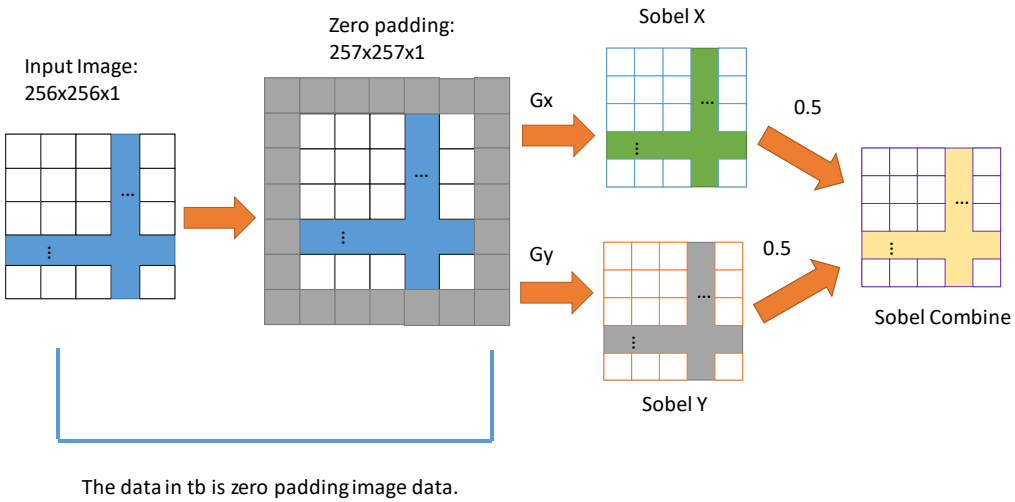


Fig. 2 - Circuit operation flow

The size of the input image in this system is 256x256, which is stored in the memory of the testfixture. The correspondence between each pixel of the grayscale image and memory arrangement is shown in Figure 4. In the operation sequence, the SOBEL circuit uses the iaddr signal to send the address of the image data to testfixture (as shown in Figure 3. t1 time point). After the negative edge of each clock, the testfixture will send the pixel data of the address into the SOBEL circuit by the idata signal (as shown in figure 3. t2 time point).

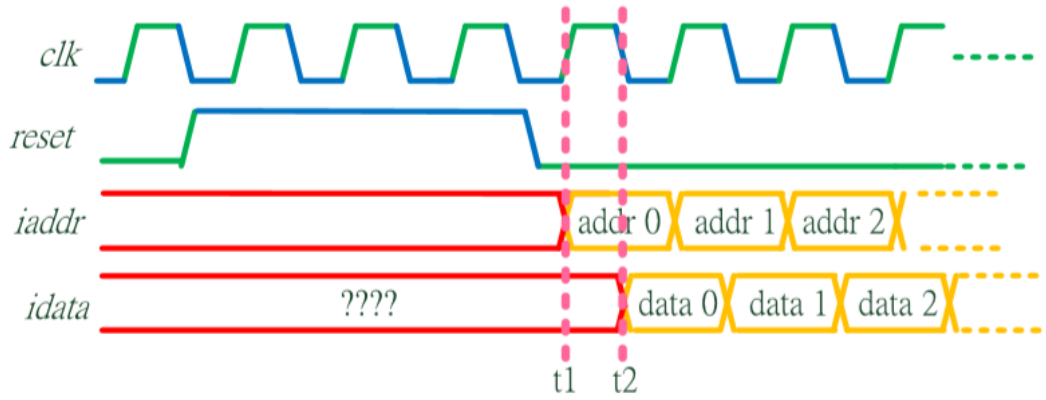


Fig. 3 – Timing diagram while reading grayscale image memory

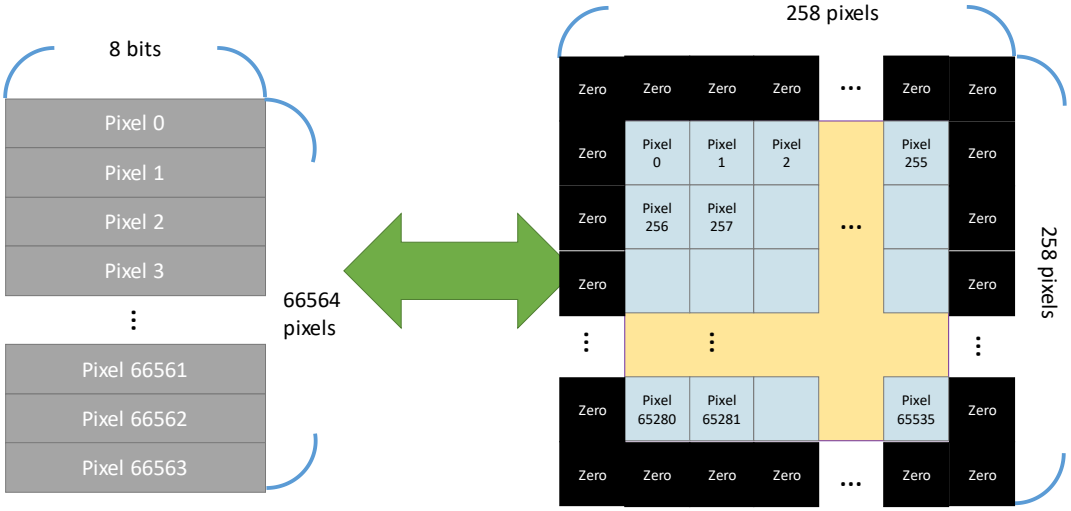


Fig. 4 – Arrangement mapping of input grayscale image and memory

The zero-padding image data is included in the memory of the system. Then convolve by Gx and Gy separately to obtain Sobel X and Sobel Y maps. In the calculation period, the value should be round into 0 to 255 if

excess. After obtaining the graphs of Sobel X and Sobel Y, use (Sobel X + Sobel Y) / 2 and then round-up for the SobelCombine. After the calculation is finished, the busy signal should set to 0, and then it will start the verification.

$$\mathbf{G_x} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

As the memory access method of this system, the output data memories L0_MEM0, L0_MEM1, and L0_MEM2 of each layer are all in RAM models and have the identical control method and timing sequence. All of them can perform writing and reading operations.

The csel signal selects the memory corresponding to each layer output. The cwr signal enables the write operation of the memory selected, and the crd as the read enable signal.

In the reading phase, the caddr_rd signal represents the memory address to read and the cdata_rd signal will provide the data. The reading operation sequence is shown in Figure 5. Note that if crd is High when the positive edge of the clock is triggered, the data at the address caddr_rd will be read to cdata_rd immediately (as time t1 in Figure 5).

When writing, the caddr_wr signal informs the memory address to write and the cdata_wr signal should provide the writing data. The writing operation sequence is shown in Figure 6. It shows that if cwr is High when the positive edge of the clock is triggered, the data of cdata_wr will be written to the address indicated by caddr_wr (as shown in Figure 6. t1 time point).

Please store SobelX data in memory L0_MEM0 (2'b01), SobelY data in memory L0_MEM1 (2'b10), and SobelCombine data in memory L0_MEM2 (2'b11).

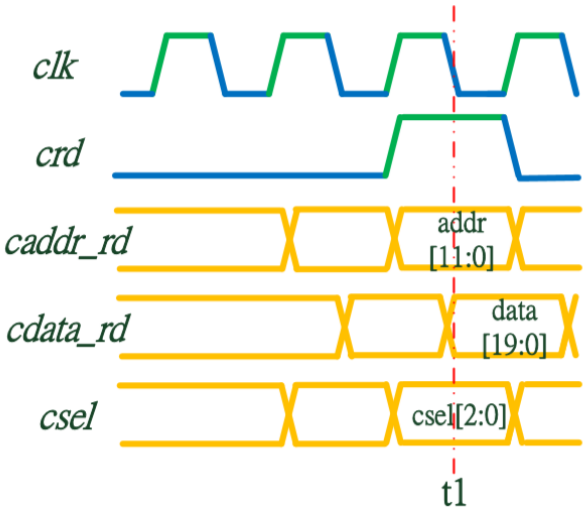


Fig. 5 – Timing diagram of reading output data memories L0_MEM0, L0_MEM1, and L0_MEM2

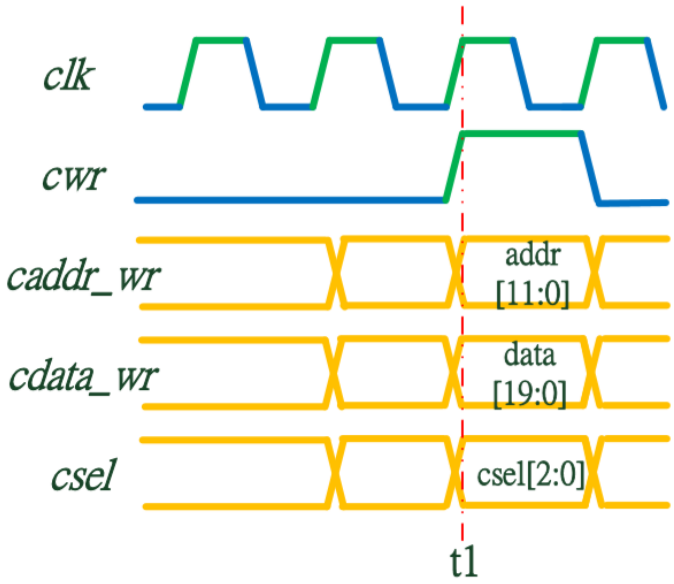


Fig. 6 – Timing diagram of writing output data memories L0_MEM0, L0_MEM1, and L0_MEM2.

3 Scoring

3.1 Functional Part

If your circuit function passes pre-sim:

[15%] Level C: The SobelX output passes the testfixture comparison.

[15%] Level B: On Level C, The SobelY output passes the testfixture comparison.

[10%] Level A: On level B, The SobelCombin output passes the testfixture

comparison.

*** Please shows the level of completion in the report ***

```
# ------------------------------------------------------
#
# START!!! Simulation Start .....
#
# ------------------------------------------------------
#
#  Sobel X  is correct !
#  Sobel Y  is correct !
#  Sobel combine is correct !
#
# ------------------------------------------------------
#
# --------------------- S U M M A R Y -----------------
#
# Congratulations! Sobel X data have been generated successfully! The result is PASS!!
#
# Congratulations! Sobel Y data have been generated successfully! The result is PASS!!
#
# Congratulations! Sobel combine data have been generated successfully! The result is PASS!!
#
# ------------------------------------------------------
#
# ** Note: $finish    : H:/DIC homework new/testfixture_fix.v(198)
#    Time: 4585020 ns  Iteration: 0  Instance: /testfixture
# 1
```

3.2 Gate Level Part

Complete Quartus synthesis (Cyclone II EP2C70F896C8) and post-sim:

[15%] Level C: The SobelX output passes the testfixture comparison.

[15%] Level B: On Level C, The SobelY output passes the testfixture comparison.

[10%] Level A: On level B, The SobelCombin output passes the testfixture comparison.

*** Please shows the level of completion in the report ***

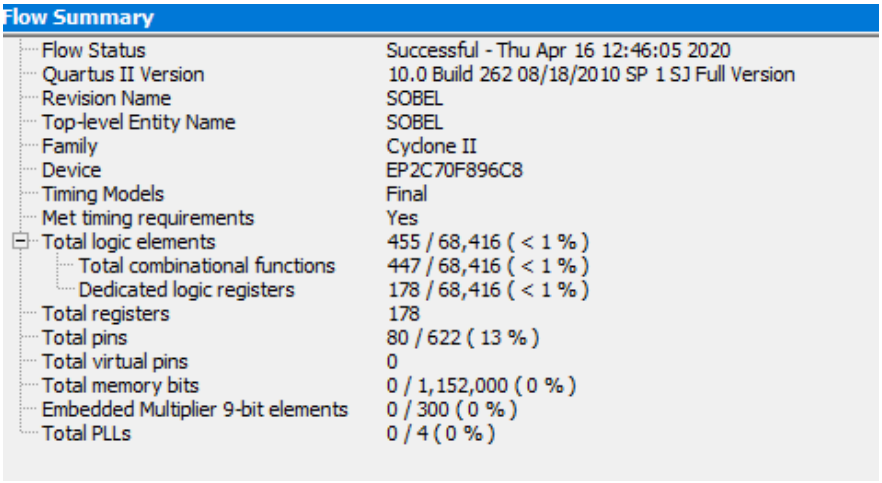
```
# ------------------------------------------------------
#
# START!!! Simulation Start .....
#
# ------------------------------------------------------
#
#  Sobel X  is correct !
#  Sobel Y  is correct !
#  Sobel combine is correct !
#
# ------------------------------------------------------
#
# --------------------- S U M M A R Y -----------------
#
# Congratulations! Sobel X data have been generated successfully! The result is PASS!!
#
# Congratulations! Sobel Y data have been generated successfully! The result is PASS!!
#
# Congratulations! Sobel combine data have been generated successfully! The result is PASS!!
#
# ------------------------------------------------------
#
# ** Note: $finish    : H:/DIC homework new/testfixture_fix.v(198)
#    Time: 4585020 ns  Iteration: 0  Instance: /testfixture
# 1
```

3.3 Performance

[20%] Complete the requirements of 3.2 and record the number of logic elements after synthesis, along with the minimum cycle time and total simulation time that can passe the simulation. Scoring by:

(Total logic elements + total memory bit + 9*embedded multiplier 9-bit element)×(longest gate-level simulation time in ns)

Less is better



Flow Summary	
Flow Status	Successful - Thu Apr 16 12:46:05 2020
Quartus II Version	10.0 Build 262 08/18/2010 SP 1 SJ Full Version
Revision Name	SOBEL
Top-level Entity Name	SOBEL
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	455 / 68,416 (< 1 %)
Total combinational functions	447 / 68,416 (< 1 %)
Dedicated logic registers	178 / 68,416 (< 1 %)
Total registers	178
Total pins	80 / 622 (13 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

4 Submission

You should classify your files into three directories and compressed to .zip format. The naming rule is HW5_studentID_name.zip.

	RTL category
*.v	All of your Verilog RTL code
	Gate-Level category
*.vo	Gate-Level netlist generated by Quartus
*.sdo	SDF timing information generated by Quartus
	Documentary category
*.pdf	The report file of your design (in pdf).

4.1 Report file

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible, and the summary results and minimum CYCLE in post-sim are necessary.

4.2 Please submit your .zip file to folder HW5 in the moodle.

Deadline: 2020-06-23 23:55

If you have any problem, please contact TAs by email:

f74044088@gs.ncku.edu.tw 倪祺婷

weiting84610@gmail.com 陳威廷