

# 2021 Digital IC Design

## Homework 3: Point Sorting Engine

### 1. Introduction:

In this homework, you are requested to design a Point Sorting Engine (PSE). The PSE circuit will take coordinate of arbitrary number of input points, and output them clockwise with first point remaining unchanged. The specification and function of PSE circuit will be described in detail in the following section.

### 2. Design Specifications:

#### 2.1 Block Overview

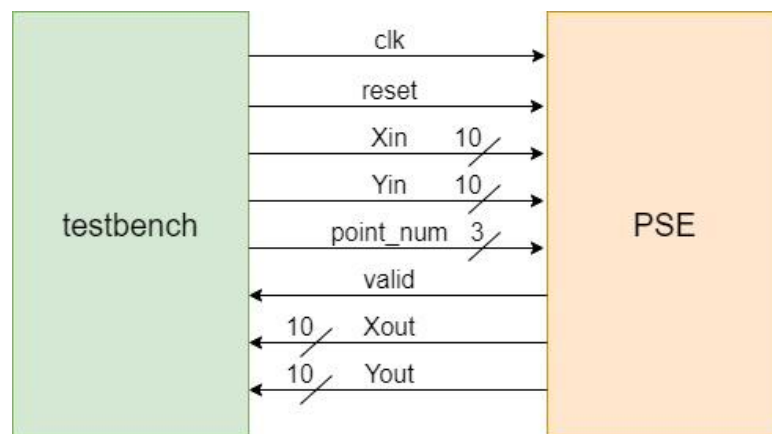


Fig. 1. The block overview.

#### 2.2 I/O Interface

Signal Name	I/O	width	Description
clk	I	1	This circuit is a synchronous design triggered at the <b>positive edge</b> of clk.
reset	I	1	Active-high <b>asynchronous</b> reset signal.
Xin	I	10	X coordinate of input point.
Yin	I	10	Y coordinate of input point.
point_num	I	3	The number of input points. The range of point_num is from 3 to 6.
valid	O	1	Output valid signal.
Xout	O	10	X coordinate of output point.
Yout	O	10	Y coordinate of output point.

## 2.3 File Description

File Name	Description
PSE.v	The top module of the design.
tb.sv	The testbench file. The content in this file is <b>not allowed</b> to be modified.
obj.data	Input data file.
golden.data	Golden data file.

## 2.4 System Description

### 2.4.1 Sorting method

Figure 2 shows an example of input points. The points are input from P1 to P6. The request of PSE circuit is to arrange the index of points in clockwise. P1 is taken as reference point, so its index will not be changed.

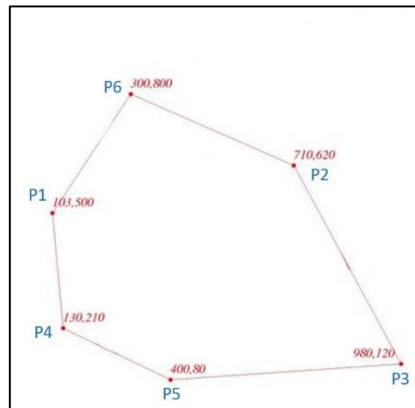


Fig. 2. Example of input points.

To sort the points clockwise, you may use cross product to compare the points pair by pair. First, connect reference point and other points to form vectors (fig.3). Then compute the cross product between vectors to judge their direction. Take the two vectors in figure 4 as example. Vector A =  $(A_x, A_y) = (x_1 - x_0, y_1 - y_0)$ . Vector B =  $(B_x, B_y) = (x_2 - x_0, y_2 - y_0)$ . The cross product of vector A and vector B is  $A_x \cdot B_y - B_x \cdot A_y$ . If the cross product is minus, B is in the clockwise direction of A and vice versa.

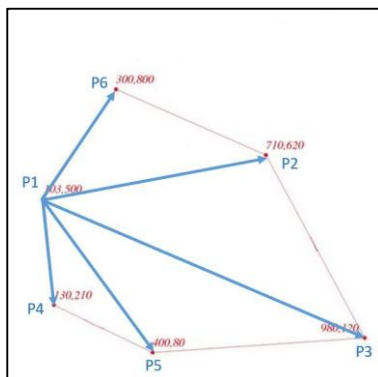


Fig. 3. Vectors.

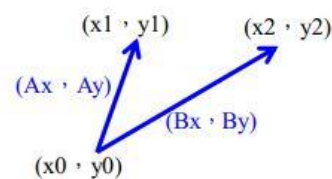


Fig. 4. Cross product calculation

After the arrangement is completed, PSE circuit has to output the coordinate of each point according to their new index. The sorted points are showed in figure 5, and they will be output from P1 to P6.

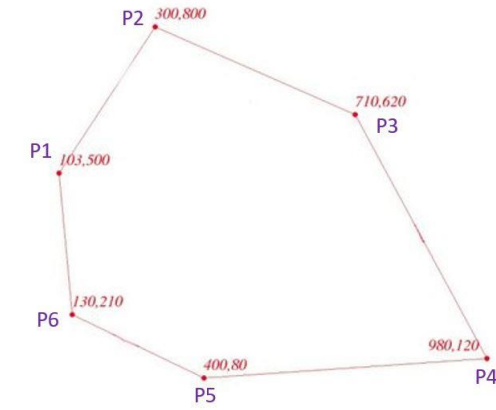


Fig. 5. Example of sorted points.

#### 2.4.2 Data input and result output

After the system reset, the number of points to sort will be input by *point\_num* port. Meanwhile, the coordinate of points will also be input by *Xin* and *Yin* ports. Testbench side will then wait for PSE circuit to complete its sorting process. PSE circuit should pull up *valid* signal when it complete calculation, and output coordinate of sorted points by *Xout* and *Yout* ports at the same time. When all sorted points are output, PSE circuit should pull down *valid* signal. Then testbench side will input next set of data.

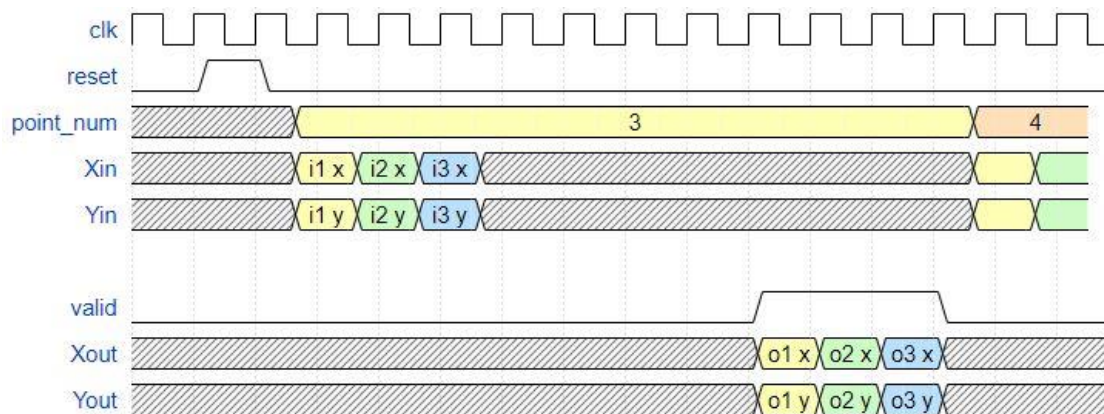


Fig. 6. Waveform of data input and result output.

### 3. Scoring:

#### 3.1 Functional Simulation [60%]

All of the result should be generated correctly, and you will get the following message in ModelSim simulation.

```
# Object43: PASS
#
# Object44: PASS
#
# Object45: PASS
#
# Object46: PASS
#
# Object47: PASS
#
# Object48: PASS
#
# Object49: PASS
#
# Object50: PASS
#
# -----
# -- Simulation finish, ALL PASS --
#
# ** Note: $finish : C:/Users/user/Desktop/lab/DIC/2021/homework/hw3/tb.sv(182)
# Time: 61250 ns Iteration: 1 Instance: /testfixture
```

Fig. 7. Functional simulation result.

#### 3.2 Gate-Level Simulation [20%]

##### 3.2.1 Synthesis

Your code should be synthesizable. After it is synthesized in Quartus, a file named *PSE.vo* will be obtained.

##### 3.2.2 Simulation

All of the result should be generated correctly using *PSE.vo*, and you will get the following message in ModelSim simulation.

```
# Object43: PASS
#
# Object44: PASS
#
# Object45: PASS
#
# Object46: PASS
#
# Object47: PASS
#
# Object48: PASS
#
# Object49: PASS
#
# Object50: PASS
#
# -----
# -- Simulation finish, ALL PASS --
#
# ** Note: $finish : C:/Users/user/Desktop/lab/DIC/2021/homework/hw3/tb.sv(182)
# Time: 61250 ns Iteration: 1 Instance: /testfixture
```

Fig. 8. Gate-level simulation result.

Device : **Cyclone II EP2C70F896C8**

### 3.3 Performance [20%]

The performance is scored by the total logic elements your design used in gate-level simulation. The score will be decided by your ranking in all received homework. (The smaller, the better)

## 4. Submission:

### 4.1 Submitted files

You should classify your files into three directories and compress them to .zip format. The naming rule is HW3\_studentID\_name.zip. **If your file is not named according to the naming rule, you will lose five points.**

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

### 4.2 Report file

Please follow the spec of report. **If your report does not meet the spec, you may lose part of score.** You are asked to describe how the circuit is designed as detailed as possible, and the flow summary result is necessary in the report. Please fill the field of total logic elements according to the flow summary of your synthesized design.

Flow Summary	
Flow Status	Successful - Thu Apr 15 16:02:37 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	PSE
Top-level Entity Name	PSE
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Total logic elements	460 / 68,416 ( < 1 % )
Total combinational functions	406 / 68,416 ( < 1 % )
Dedicated logic registers	221 / 68,416 ( < 1 % )
Total registers	221
Total pins	46 / 622 ( 7 % )
Total virtual pins	0
Total memory bits	0 / 1,152,000 ( 0 % )
Embedded Multiplier 9-bit elements	2 / 300 ( < 1 % )
Total PLLs	0 / 4 ( 0 % )

Fig. 9. The flow summary.

### 4.3 Note

This homework required your circuit to operate in given clock frequency, which is 50ns clock width. **You could not modify the defined CYCLE in testbench file.** The only part you can modify is End\_CYCLE, which decide the maximum cycles your circuit took to complete simulation.

The process of cross product calculation need multipliers, which are resource intensive units. If your design takes too much time to synthesize, you may try optimizing your design by reducing the multipliers it used.

Please submit your .zip file to folder HW3 in moodle.

**Deadline: 2021/5/17 23:55**

If you have any problem, please contact TA by email.

[lt2es.93039@gmail.com](mailto:lt2es.93039@gmail.com)