Midterm Exam

The test is divided into two parts, A and B, with a total of 110 points. If the final score exceeds 100 points, it will be recorded as 100 points.

Part A (90%):

1. Brief introduction

In this part, you are requested to design a circuit that performs the following functions: determine which of the X and Y coordinates has the relatively larger magnitude, calculate the center of mass of a triangle, identify whether the triangle is acute, obtuse, or right-angled, find the length of the triangle's longest side, and determine whether a fourth input point lies inside the triangle. You are also required to write your own testbench to test your circuit.

2. Circuit function & Grade distribution

Part	Function	Score	Formula
Compare	After receiving the X and Y coordinates, compare their magnitudes immediately. The 'Max' port should output the comparison result, with the least significant bit (LSB) sent first."	15%	See Note2
Testbench	Write your own test bench that is sufficient to debug the "Compare" part.	20%	See Note1
Center_of_mass	Find the center of mass of the triangle.	10%	Pass to get all
LongestSide	Find the length of the triangle's longest side.	15%	Pass to get all
TriangleType	Identify whether the triangle is acute, obtuse, or right-angled	15%	Pass to get all
ForthPoint	Determine where the fourth input point lies on.	15%	Pass to get all

Note1 : You may use your self-designed TestBench-1 to test your circuit functionality.

1. The code compiles, waveform is successfully generated, and the terminal displays Max Value is: XX (**8%**). *Even if the circuit function is incorrect, you will still receive 8% for this.*
2. The terminal displays an incorrect Max Value, but the waveform clearly shows that A[7:0] and B[7:0] were read in correctly (**8%**).
3. The terminal displays an incorrect Max Value, but the waveform clearly shows that the comparison result is correct (**2%**).
4. The code compiles, and the test results from your TestBench are reasonable (**2%**).

Note2 : After passing the Note1, the TA will provide you with TestBench-2, which performs more rigorous testing. Passing the first 20 test cases earns 7%, while passing all test cases earns 15%.

Note3 : You have to implement square root by yourself. You can't use design ware in this midterm exam.

3. I/O description

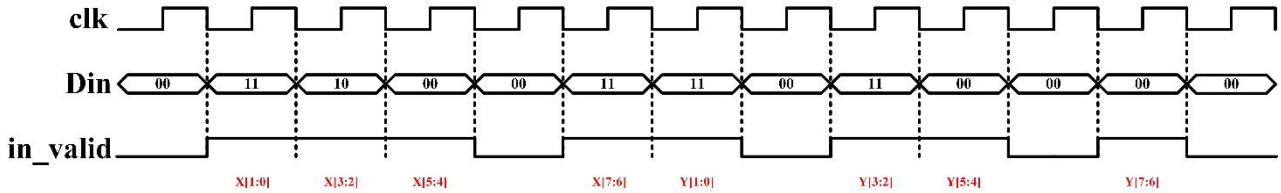
Signal	I/O	Definition
clk	Input	System clock
rst	Input	Active high reset signal
Din [1:0]	Input	Coordinate of triangle (LSB inserts first)
in_valid	Input	Din valid signal
ready	Output	Ready to receive Din signal
Max[1:0]	Output	Max of X,Y (LSB output first)
out_valid	Output	Max valid signal
out_valid_SecondStage	Output	Set 1 if Center_of_MassX, Center_of_MassY, LongestSide, TriangleType, ForthPoint is valid
Center_of_MassX [7:0]	Output	Coordinate of Center_of_MassX (round down to integer)
Center_of_MassY [7:0]	Output	Coordinate of Center_of_MassY (round down to integer)
LongestSide [8:0]	Output	Longest side length of the triangle
TriangleType [1:0]	Output	00 for right triangle 01 for acute triangle 10 for obtuse triangle
ForthPoint [1:0]	Output	00 for the forthpoint is on the line of the triangle 01 for the forthpoint is inside the triangle 10 for the forthpoint is outside the triangle

4. Circuit specification

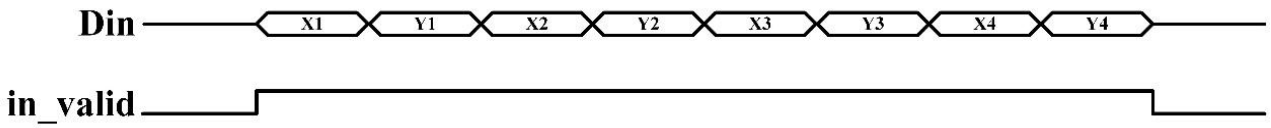
Din[1:0] input:

1. When you set the ready signal High. TestBench starts to output in_valid and Din. The value of Din is valid when in_valid is 1; otherwise, the value of Din is invalid. Your circuit needs to wait until all A[7:0] and B[7:0] are received.

When Din is valid. The values entered in sequence are:
X[1:0]、X[3:2]、X[5:4]、X[7:6]、Y[1:0]、Y[3:2]、Y[5:4]、Y[7:6]



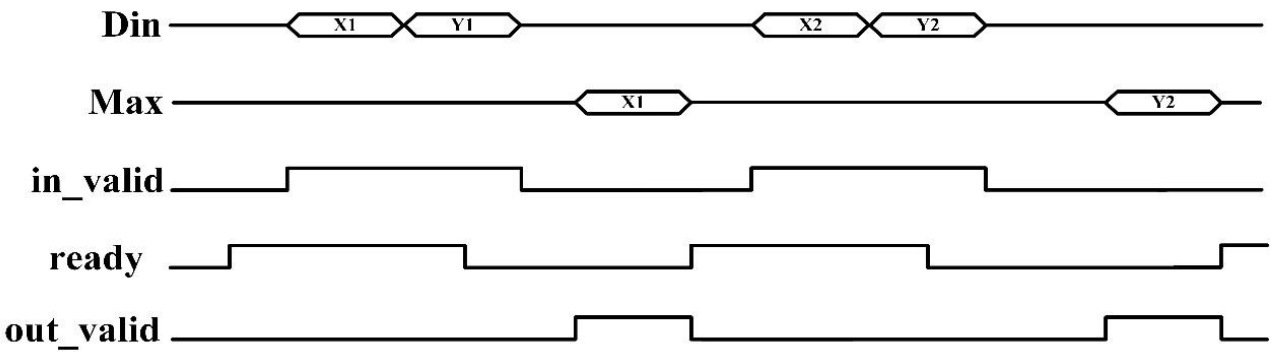
2. As shown in the figure below, the testbench input sequence starts from the X-coordinate of the first point to the Y-coordinate of the fourth point. The first three points are guaranteed to be non-collinear and will form a triangle. The fourth point is a test point used to determine whether it lies inside the triangle.



Max[1:0] output:

1. You should set “out_valid” high and set “ready” low to send your compare result out.

When out_valid set high. The values sent out in sequence are:
Max [1:0]、Max [3:2]、Max [5:4]、Max[7:6]



2. The out_valid signal should remain high for only 4 cycles. This means that once Max[7:6] has been output, all results of this comparison operation should have been fully transmitted. At this point, out_valid must be set to 0. Otherwise, if out_valid stays high for more than 4 cycles, the output of

this comparison will exceed 8 bits (since 2 bits are output per cycle, keeping out_valid high beyond 4 cycles will result in more than 8 bits being output).

TestBench -1:

Please design **TestBench -1** yourself. Below is a sample outline:

1. Generate the system clock signal (clk).
2. Generate the system reset signal (rst). The timing is up to you, but ensure reset is properly tested.
3. Based on your own ready timing, create in_valid and Din inputs. Scoring Criteria

   First set: A = 10; B = 7;

   Second set: A = 2; B = 120;

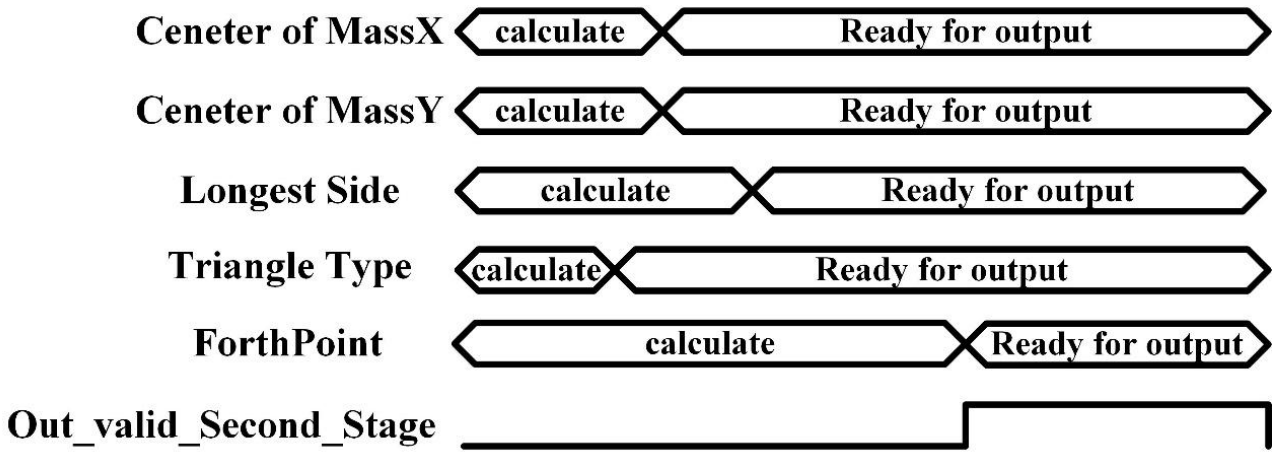
   Third set: A = 60; B = 60;

4. After reading out Max, the terminal should display:

   ```
   $display("\n\n=== Max Value is %d ===\n", Maximum);
   ```

SecondStage output:

1. out_valid_second_stage pull high if Center_of_MassX, Center_of_MassY, LongestSide, TriangleType, ForthPoint are ready to output.



Part B (20%):

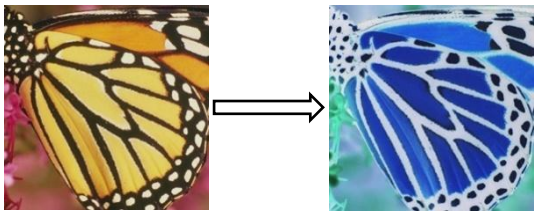
1. Brief introduction

In this part, we will provide you with a specified image and the corresponding correct output. Your task is to write a circuit that generates the negative image, along with a testbench to compare your circuit's output against the correct result. The method for generating the negative image(256*256) is as follows :

R => 255-R

G => 255-G

B => 255-B



2. Grade distribution

Part	Score	Formula
Negative image	10%	Pass to get all
Testbench	10%	Pass to get all

3. I/O description

Signal	I/O	Definition
clk	Input	System clock
rst	Input	Active high synchronous positive edge triggered reset
pixel_in [23:0]	Input	Unsigned pixel to be processed {R[7:0], G[7:0], B[7,0]}
pixel_out [23:0]	Output	Unsigned pixel after process {R[7:0], G[7:0], B[7,0]}
valid	Output	Test bench read pixel_out when valid pull high