CSIE DIC LAB 2023

Lab04b. Geofencing System

Description

This lab involves implementing a virtual geofencing system with six sensors, which is used to create a virtual fence on a two-dimensional plane, as shown in Figure 1. In this system, each sensor can accurately measure the distance between itself and the target object. Based on this critical information, the virtual geofencing system can determine whether the target object is located inside or outside of the virtual fence.

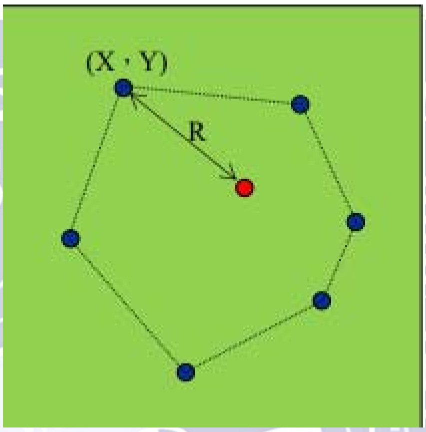

Figure 1: Geofence

The virtual geofencing system consists of six sensors, and the (X, Y) coordinates of each sensor, as well as the distance (R) between the sensor and the target object, are not input into the virtual geofencing system in any particular order. Therefore, at the beginning, it is necessary to sort the input sensor data in either clockwise or counterclockwise order.

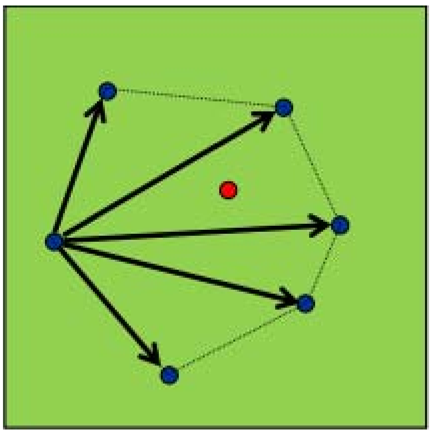

Figure 2: Use vectors to sort points

In order to facilitate subsequent processing of sensor data, the (X, Y) coordinate of one sensor can be used as a reference point for sorting. Vectors between the reference point and other sensor coordinates can be obtained, as shown in Figure 2. Next, it is necessary to determine the relationships between these vectors. For example, in Figure 3, given three coordinate points (x0, y0), (x1, y1), and (x2, y2), when (x0, y0) is taken as the reference point, two vectors, V1 and V2, can be calculated as follows:

$$\mathbf{V1 = (x1 - x0, y1 - y0)}$$
$$\mathbf{V2 = (x2 - x0, y2 - y0)}$$

Then, the cross product of these two vectors can be calculated:

$$\mathbf{cross\_product = V1 \times V2 = (x1 - x0) * (y2 - y0) - (x2 - x0) * (y1 - y0)}$$

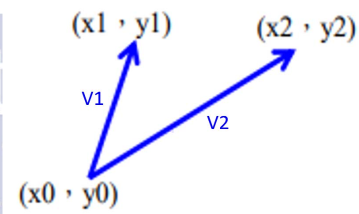

Figure 3: Vector cross product

If the cross product is greater than 0, then V1 is in the clockwise direction relative to V2; if the cross product is less than 0, then V1 is in the counterclockwise direction relative to V2. Therefore, the direction relationship between the two vectors can be determined based on the sign of the cross product, and using this information, the order relationship between these three points, shown in Figure 3, can be kept in clockwise or counterclockwise order. After several rounds of vector calculation and sorting, the data from the six sensors shown in Figure 2 can be sorted into clockwise or counterclockwise order.

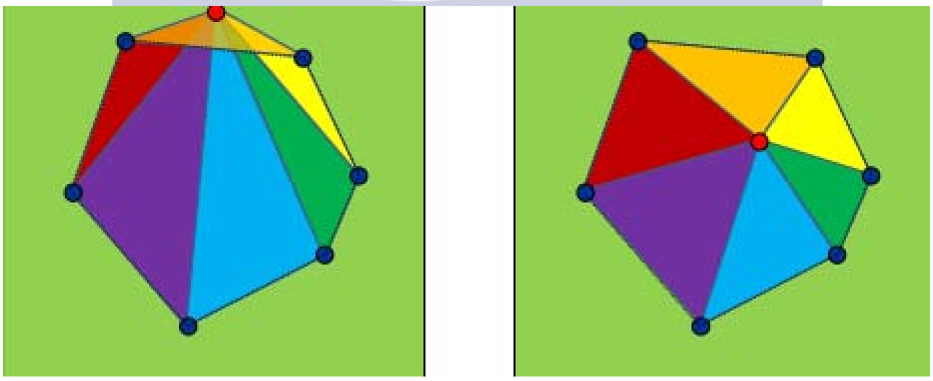

Figure 4: The relationship between geofence and object

To determine whether the target object is within a virtual fence, the following method can be used. First, the coordinates (X,Y) of the input sensors must be arranged in either a clockwise or counterclockwise order, as shown in Figure 4. Next, calculate the area of the hexagon formed by the six sensors. Then, in a clockwise or counterclockwise order, form a triangle with each consecutive pair of sensors and the target object, and calculate the area of each triangle as illustated in Figure 4.

There will be a total of six triangle areas to calculate. If the sum of these areas is greater than the area of the hexagon formed by the six sensors, then the target object is outside the virtual fence, as shown in the left part of Figure 4. Otherwise, the target object is inside the virtual fence, as shown in the right part of Figure 4.

As the distance to the target object is known by the sensors, the area of the triangle formed by the target object and the two sensors can be calculated using Heron's formula. Assuming the lengths of the three sides of the triangle are **a**, **b**, and **c**, the area of the triangle can be calculated using the following formula.

$$\mathrm{s} = (\mathrm{a} + \mathrm{b} + \mathrm{c})/2$$

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)} = \sqrt{s(s-a)} * \sqrt{(s-b)(s-c)}$$

The reason for splitting the square root into two parts is that the circuit area of directly calculating the square root is too expensive. By splitting it into two low-bit square root circuits, the area can be reduced, but accuracy is sacrificed. **<u>In this lab, you should implement square root circuits that take 21 bits inputs and output 11 bits to calculate the area of a triangle.</u>** Additionally, if the three vertices of a triangle are close to collinear, truncation errors in the calculation may result in negative values in the square root. In this case, additional judgment and processing are required.

When the coordinates (X,Y) of the input sensor have been arranged in clockwise or counterclockwise order, the area of a polygon can be calculated using the following formula. Note that when the vertex coordinates are arranged counterclockwise, the area is a positive value. Conversely, when the vertex coordinates are arranged clockwise, the area is a negative value.

$$a = \frac{1}{2}\left(\begin{vmatrix} x_0 & x_1 \\ y_0 & y_1 \end{vmatrix} + \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \cdots + \begin{vmatrix} x_{n-2} & x_{n-1} \\ y_{n-2} & y_{n-1} \end{vmatrix} + \begin{vmatrix} x_{n-1} & x_0 \\ y_{n-1} & y_0 \end{vmatrix} \right)$$

Specification

1. Top module name: geofence
2. Top module filename: geofence.v
3. Input/output definition

Signal Name	Direction	Bit Width	Description
clk	Input	1	Clock signal
reset	Input	1	Asynchronous reset signal (active high)
X	Input	10	X-coordinate of the sensor (unsigned)
Y	Input	10	Y-coordinate of the sensor (unsigned)
R	Input	11	Distance between the object and the sensor (unsigned)
is_inside	Output	1	When the object is inside the virtual fence, output is 1, otherwise is 0.
valid	Output	1	Asserted when is_inside and Area are valid
Area	Output	22	Area of the geofence

4. Input signals are unsigned number and **synchronized at the clock rising edge**.
5. The reset scheme is an **active-high asynchronous reset**.
6. The virtual fence system employs a total of six sensors, each of which stores its own coordinates (X, Y) and the distance R to the target object. Each target object is associated with six sets of related data.
7. After inputting six sets of data, the host waits for a response from the virtual geofencing system. Once the virtual geofencing system completes its calculations, the "valid" signal must be asserted to a high voltage for one cycle, and the outputs, including whether the target object is inside the virtual fence (**is_inside**) and the area of the virtual fence (**Area**) should be ready. Then, in the next cycle, the "valid" signal is pulled back to a low voltage.

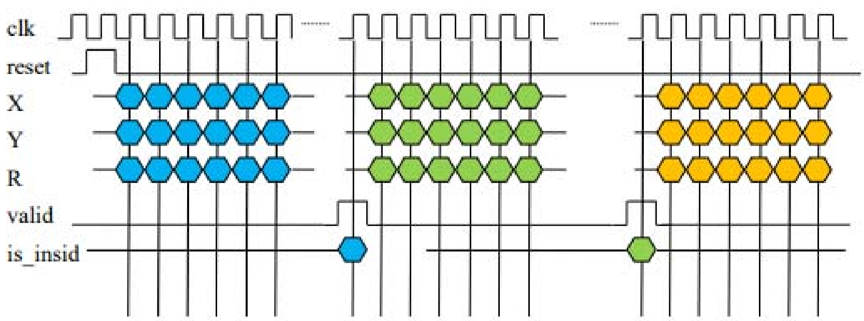

Figure 5: Input and output relationship

8. After the host pulls the "valid" signal back to a low potential, in the next cycle, it begins inputting the sensor contents of the next target object, entering six sets of

sensor data in sequence before entering a waiting state once again. The timing diagram for input and output is shown in Figure 5.

9. Each time the host inputs target object data, it is a new set of virtual fence system data unrelated to the previous input.
10. The host only checks the output value when the "valid" signal is at high voltage.
11. Try to write your design in synthesizable coding style.

Example Timing Diagram

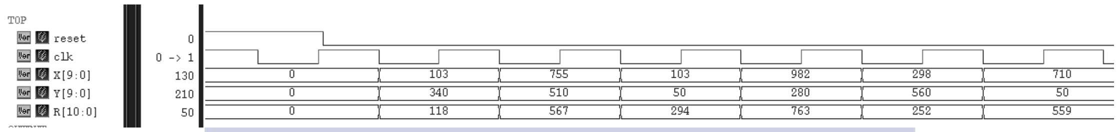

Figure 6. Input timing diagram

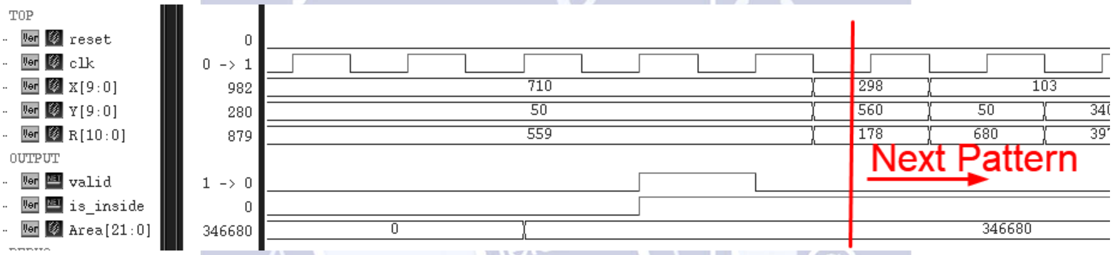

Figure 7. Output timing diagram

Lab Instructions

1. Extract LAB data from TA's directory:

   % tar xzvf ~dicta/lab04b.tar.gz

2. The extracted LAB directory (**lab04b**) contains:

   A. **tb.sv** : Design test bench
   B. **geofence.v** : Empty design module
   C. **geofence.vp** : Encrypted reference design module
   D. **grad.data** : Input data and answers
   E. **lab04b.rc** : nWave saved signals file

3. Change directory to **lab04b**

   % cd lab04b

4. Execute the Verilog simulation using the encrypted reference design (**geofence.vp**). For your guidance in unstandarding the input and output timing diagrams for this lab, protected Verilog modules are for your reference.

   % ncverilog -f run.f

5. Open simulation waveform:

   % nWave &

Choose **File → Open** from the menu, then select "**geofence.fsdb**" and press OK. To restore the recorded signal, choose **File → Restore Signal** from the menu and select "**lab04b.rc**" and press OK.

6. Write your RTL code in synthesizable coding style:

   % gedit geofence.v &
   or % gvim geofence.v &
   or % joe geofence.v
   or % vim geofence.v

7. Run RTL simulation with your design

   (**Remember to modify the include file in tb.sv from geofence.vp to geofence.v**)

   % ncverilog -f run.f

8. Meaning of signals in DEBUG group:

   objnum : object number
   obj_isin : object is inside or not
   obj_area : area of the virtual fence
   fail : accumulated error number