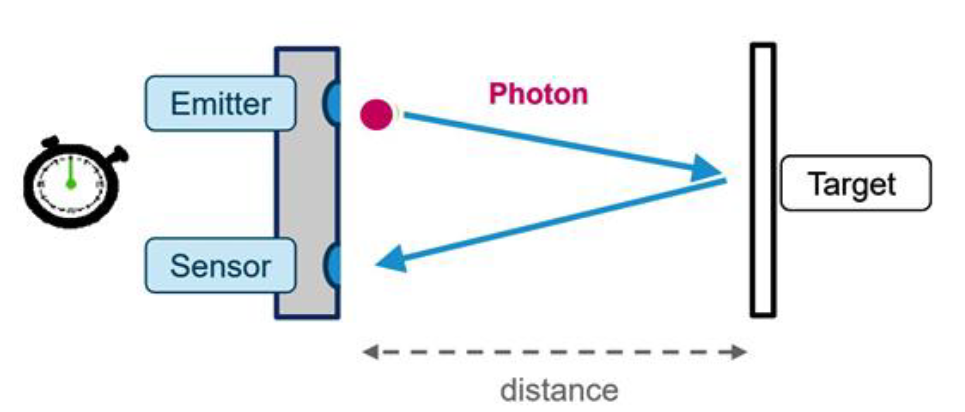
IC Lab Final Project Report

2022 Spring

**Name: 林哲宇 Student ID: A101085 Account: iclab108 y**

* Design Description
  + Time of Flight (ToF) is an important technique for distance measurement.
  + ToF measures the time taken by the light to travel back and forth between the target and the sensor.
  + A single photon detector can detect the photon and output a digital pulse when the photon hit its active region.
  + Chart, bar chart

    Description automatically generatedSuppose we have a single photon array; we can construct a 3D picture (4\*4\*255) by calculating each pixel’s distance.
* Project Description

In this project, we are going to implement an ToF chip. The goal is to calculate how many cycles have passed from the laser (start) to the sensor (stop), calculate the distance of the object, and store it in DRAM.

In real world, there will be background noise, so we cannot calculate the distance using just one start and stop.

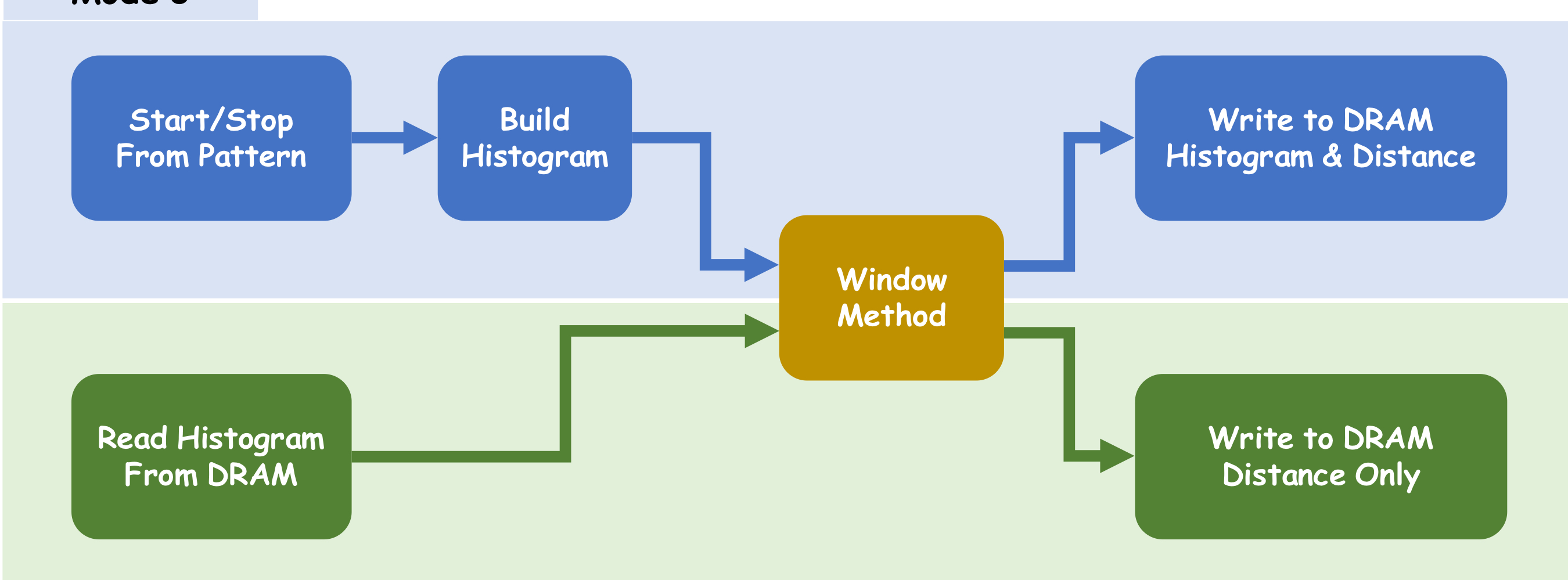
To have more accurate result, we will using multiple start and stop to calculate the actual distance. Below is an example of histogram generation. We will get the histogram and calculate the distance.



**Histogram generating example**

* System Architecture

For input type = 0, the histogram will come from DRAM via AXI-4. And for input type = 1,2,3, should construct histogram from input start and stops.



We used two more sophisticated method to enhance the measuring accuracy, temporal correlation and spatial correlation.

There are four types in this project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | #start | Temporal | Spatial | Accuracy | Score (%) | Data Source |
| Type0 | 15 | optional | no | >0.5 | 5 | DRAM |
| Type1 | 4 | optional | **required** | >0.5 | 5 | input |
| Type2 | 7 | optional | optional | Not required | 10 | input |
| Type3 | 7 | optional | optional | Not required | 15 | input |

1. Spatial Correlation

If there are spatial correlation between each pixel, we can increase the confidence of our measurement by getting consistent results between pixels.

type0: all pixels are independent (no spatial correlation)

type1: 2\*2 pixel forms a group, the distance in a group is the same

type2: convex shape with random peak position

type3: concave or convex shape with random peak position

**stops and pixel correlation Example of type1**



**Examples of convex shape**

****

**Examples of concave shape**

1. Temporal Correlation

If the pulse waveform is known, we can find the shape that match the laser pulse from the histogram.

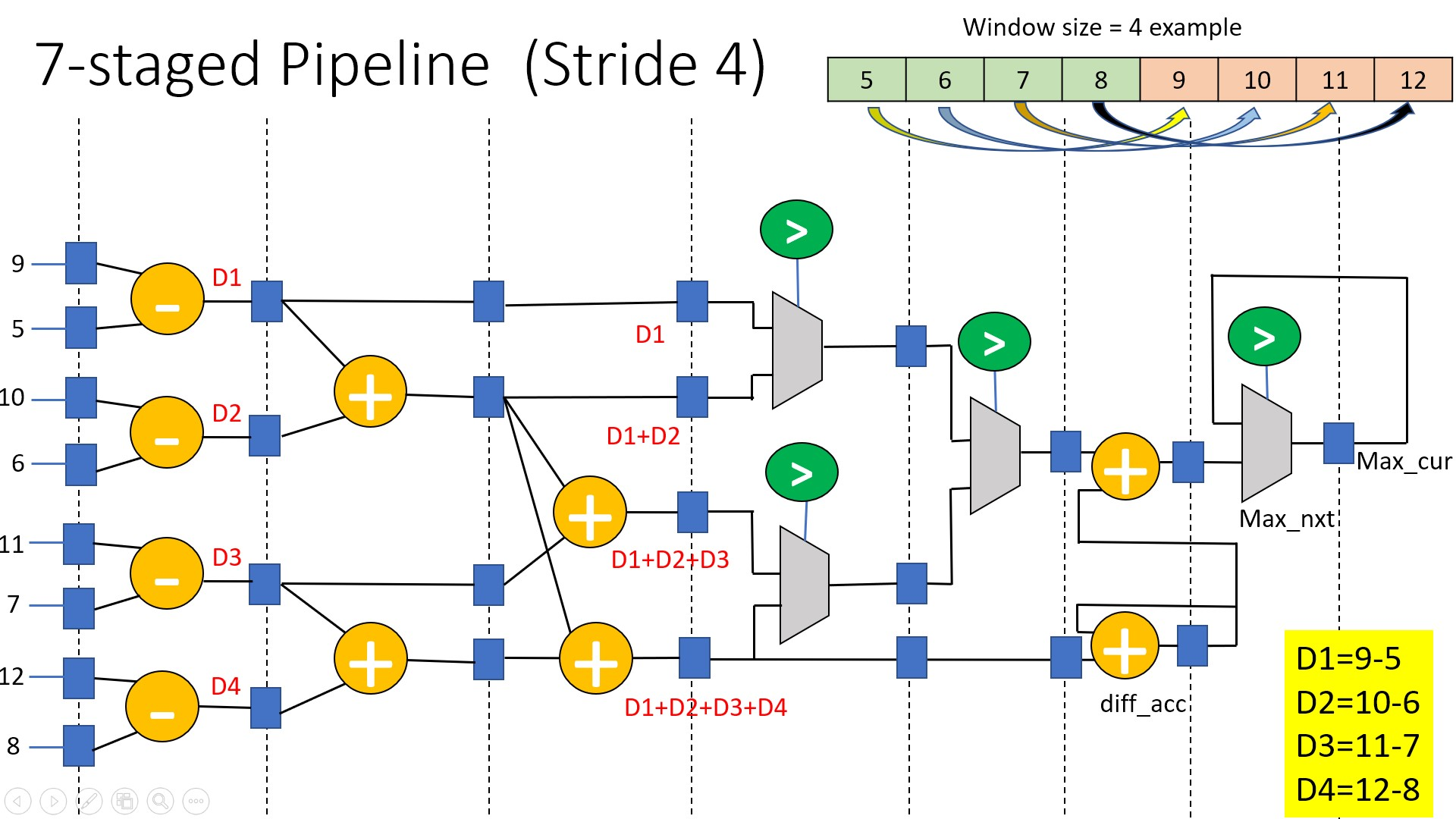
Pulse shape:

type0 & type1: {0.3, 0, 0.3, 0, 0.3}

type2 & type3: {0.1, 0.4, 0.3, 0.2, 0.1}

* Methodology

To let the circuit work more efficiency, I design pipeline architecture. Below is the architecture diagram.



* To enhance the measuring accuracy I design 4 spatial correlations in each type.
  + - 1. Type 0:

The slide window size is 5. Moreover, I use the weight set {1, 0, 1, 0, 1} to multiply the windows {p1, p2, p3, p4, p5}. Hence, I will find N adjacent bin have the maximum value with function p1+p3+p5. Then, other operation is all same as midterm project in type 1.

* + 1. Type 1:

The strategy is similar to type 0. However, I will group the pixel 0, 1, 4, 5 as one histogram, 2, 3, 6, 7 as one histogram, 8, 9, 12, 13 as one histogram and 10, 11, 14, 15 as one histogram. Just like the picture below, will sum the same color pixel as one histogram. So, the spatial correlation will be used. And other strategy is same as type 0.

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |

* + 1. Type 2:

Take pixel 0 as example, when the data come into the window, it will compare the sum of histogram, pixel 0 + pixel 4 and pixel 0 + pixel 1, then choose the larger one into the window. And the size of window is 5 with no weight set. So, the histogram in window may sometimes be pixel 0 + pixel 4, sometimes be pixel 0 + pixel 1. This operation is that if the pixel is not the peak position, then there will be at least one of the neighbor pixel have same golden answer. Hence, I try to group them just like type 1.

Then, I will also calculate the neighbor 3 pixels simultaneously. Then check the error of the distance is too large or not. If yes, then I will correct the answer to one of neighbor pixels; if no, will remain the same.

Pixel 0:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |  | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |  | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |  | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |  | 12 | 13 | 14 | 15 |

Pixel 1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |  | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |  | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |  | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |  | 12 | 13 | 14 | 15 |

Pixel 4:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |  | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |  | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |  | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |  | 12 | 13 | 14 | 15 |

Pixel 5:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |  | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |  | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |  | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |  | 12 | 13 | 14 | 15 |

And so on.

* + 1. Type 3:

The strategy is same as type 2.

* And for temporal correlation. In type 0 & type 1, I use the weight set {1, 0, 1, 0, 1} to multiply the windows {p1, p2, p3, p4, p5}. Hence, I will find N adjacent bin have the maximum value with function p1+p3+p5. In type 2 & type 3, use the weight set {1, 4, 3, 2, 1}.
* To have a better performance (Area\*Timing), I used two SRAM 64 bit 128 words to minimize the area. Moreover, to reduce the timing writing back to DRAM, I calculated the distance meanwhile control the axi4 signal to write the histogram back to the DRAM. So, I don’t need the write back state to waste more cycle.
* Results

