3D Pottery Game

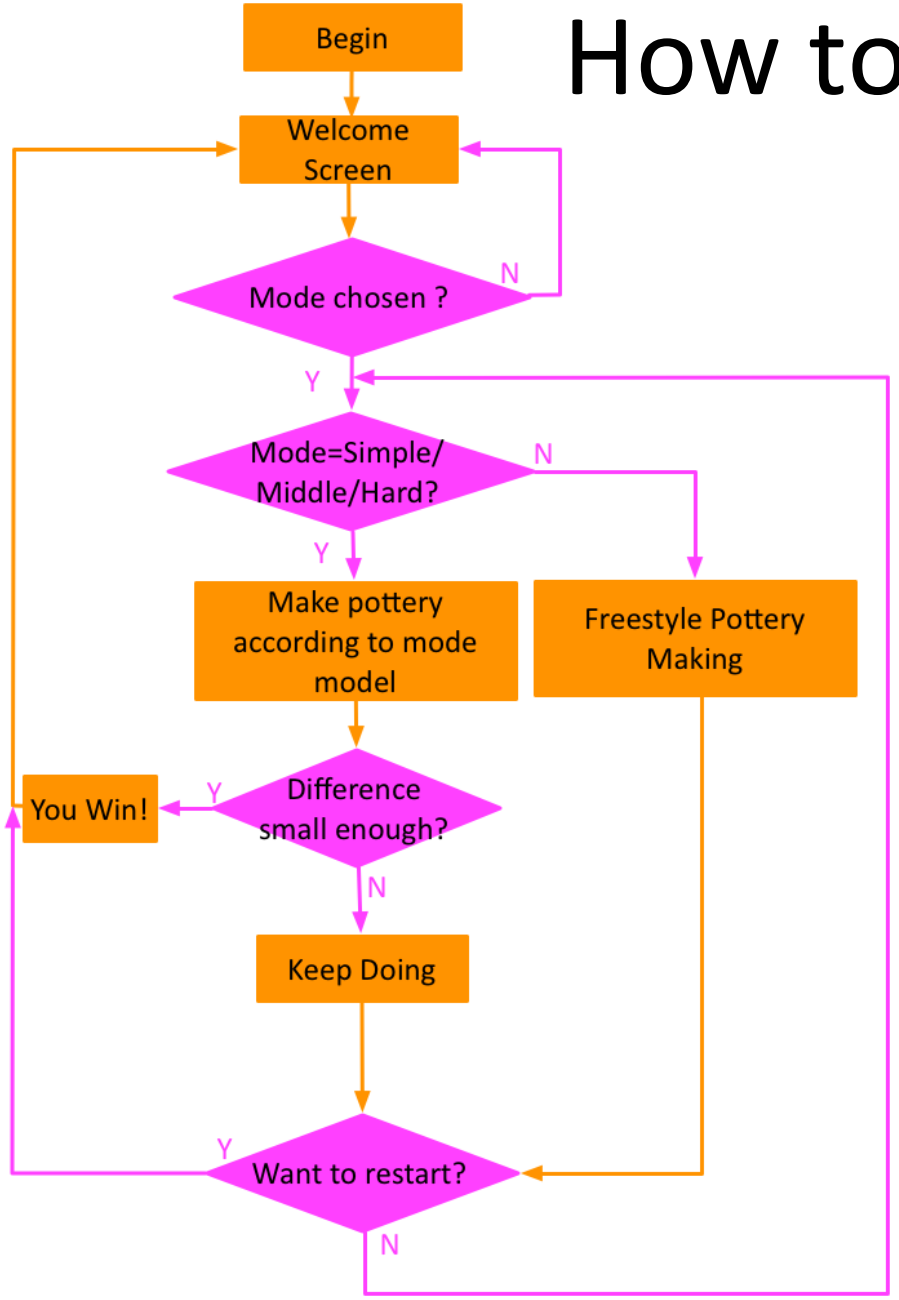
Hao Jin (hj2354) Chengxue Qian (cq2159)
Xiaowen Han (xh2204) Muqing Liu (ml3466)

May 13th, 2014

Introduction

- A little recreation game in which the user can either create their own pottery or try to make the pottery as similar to the reference ones as possible.
- User can use the mouse to select mode and change the shape.

How to Play



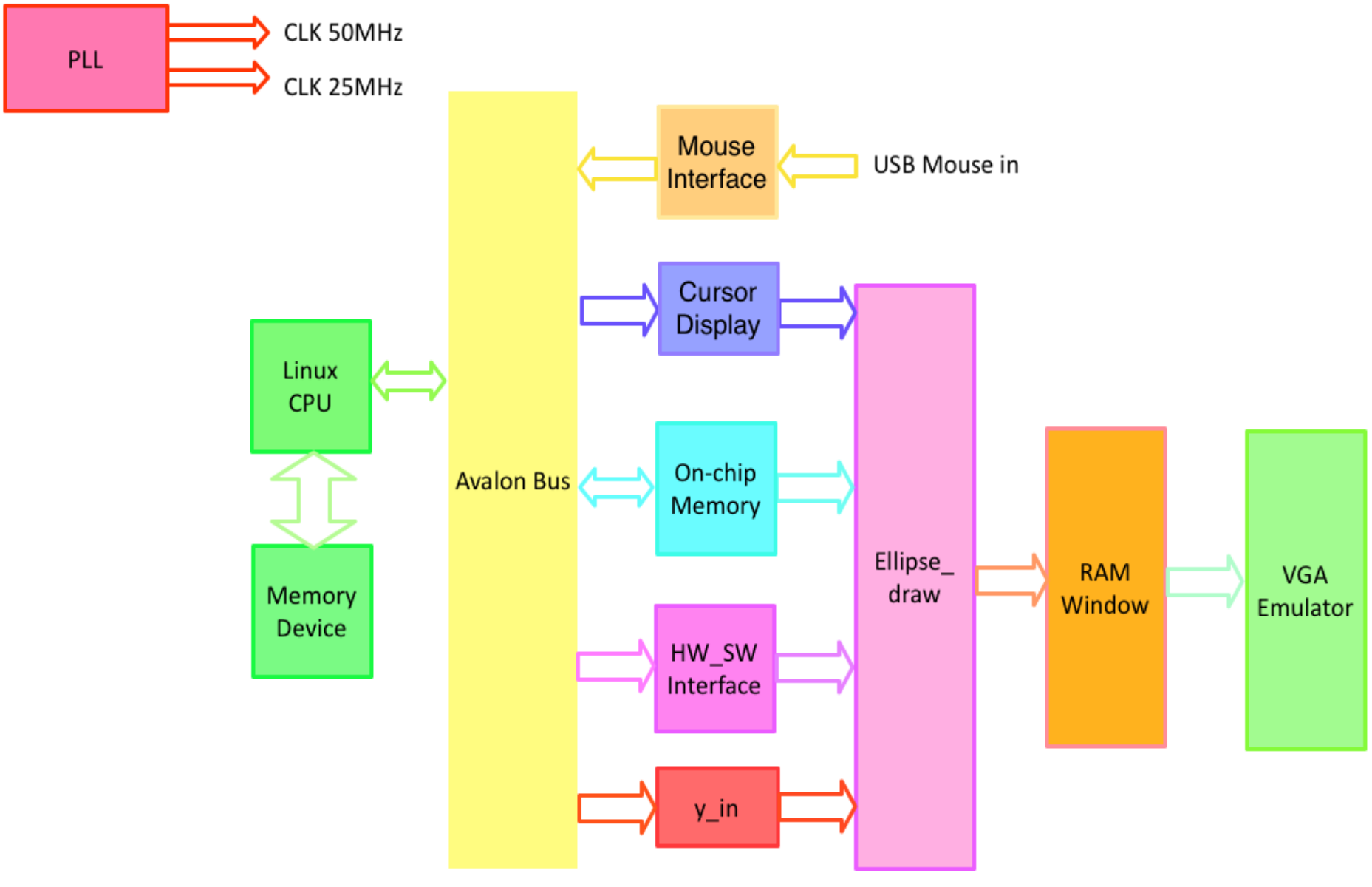
Welcome to the Pottery-Making game!

First you have to choose from three hard level models; or you can do freestyle as well.

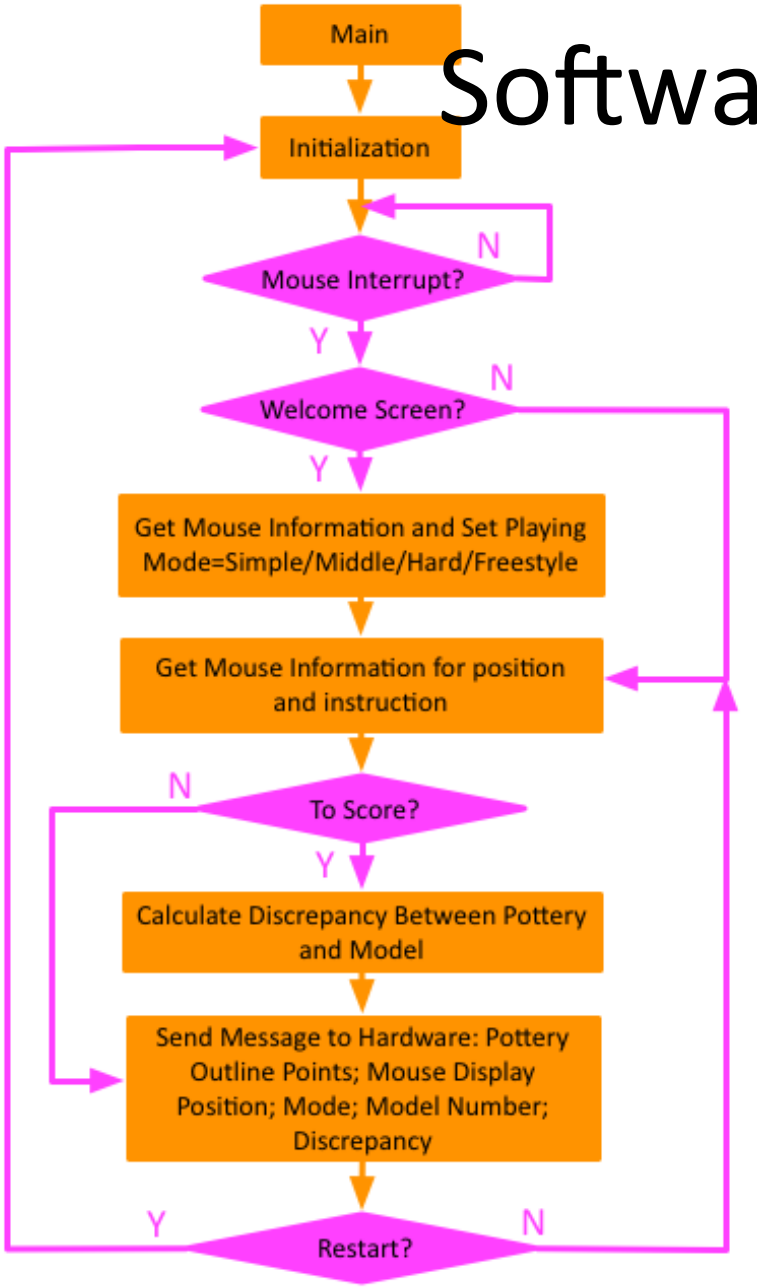
Next you can begin to play around with your original pottery.

When feel satisfied, just click to see if you win or more efforts should be made. Also, you can restart all over!

System Overview



Software Overview



In the main function, we continuously look for any mouse interrupt.

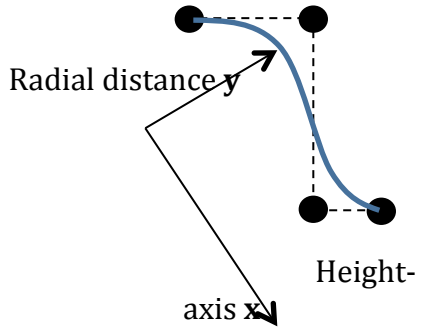
If it is in the “WELCOME” mode, we set the playing models first. Otherwise, we go straight forward to calculate a series of discrete points on the current newly fitted pottery outline curve.

If the mouse information indicates “TO SCORE”, we calculate the discrepancy between current pottery and the model.

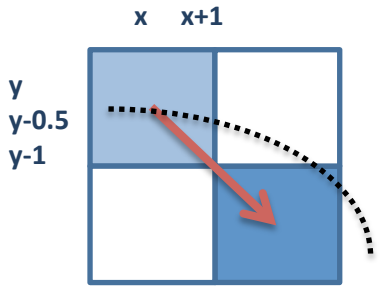
Finally, we send all message needed (pottery outline points; mouse display position; mode no.; model no.; discrepancy.) to hardware.

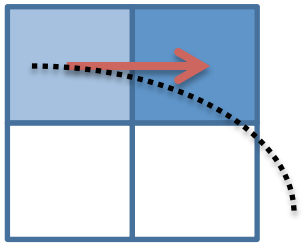
How we do curve fitting

By default, we maintain 17 control points. Using third-order uniform B-spline curve fitting, we obtain one B-spline from four adjacent points as is showed in the figure , which corresponds to a total of 17-3=14 B-splines. (B-spline, or Basis Spline, is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. It is used for curve fitting. Each third-order B-Spline will exactly fit four points.) Then, we make 9 uniform partitions between every two adjacent B-splines to split the interval into 10 parts. That is to say, we maintain 14*10=140 points to represent the curve-fitted.

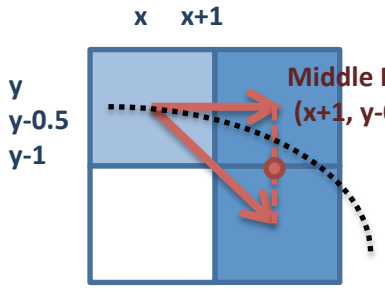


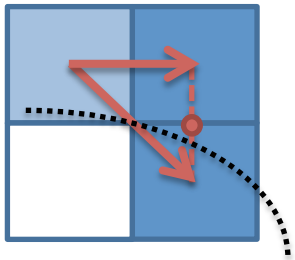
HOW TO DRAW EACH ELLIPSE





Choice 1: from (x, y) to (x+1, y-1) Choice 2: from (x, y) to (x+1, y)





Choice Decision:
If (x+1, y-0.5) is in the ellipse;
Then next step is (x+1, y).

Choice Decision:
If (x+1, y-0.5) is not in the ellipse;
Then next step is (x+1, y-1).

We use middle point method for ellipse drawing.

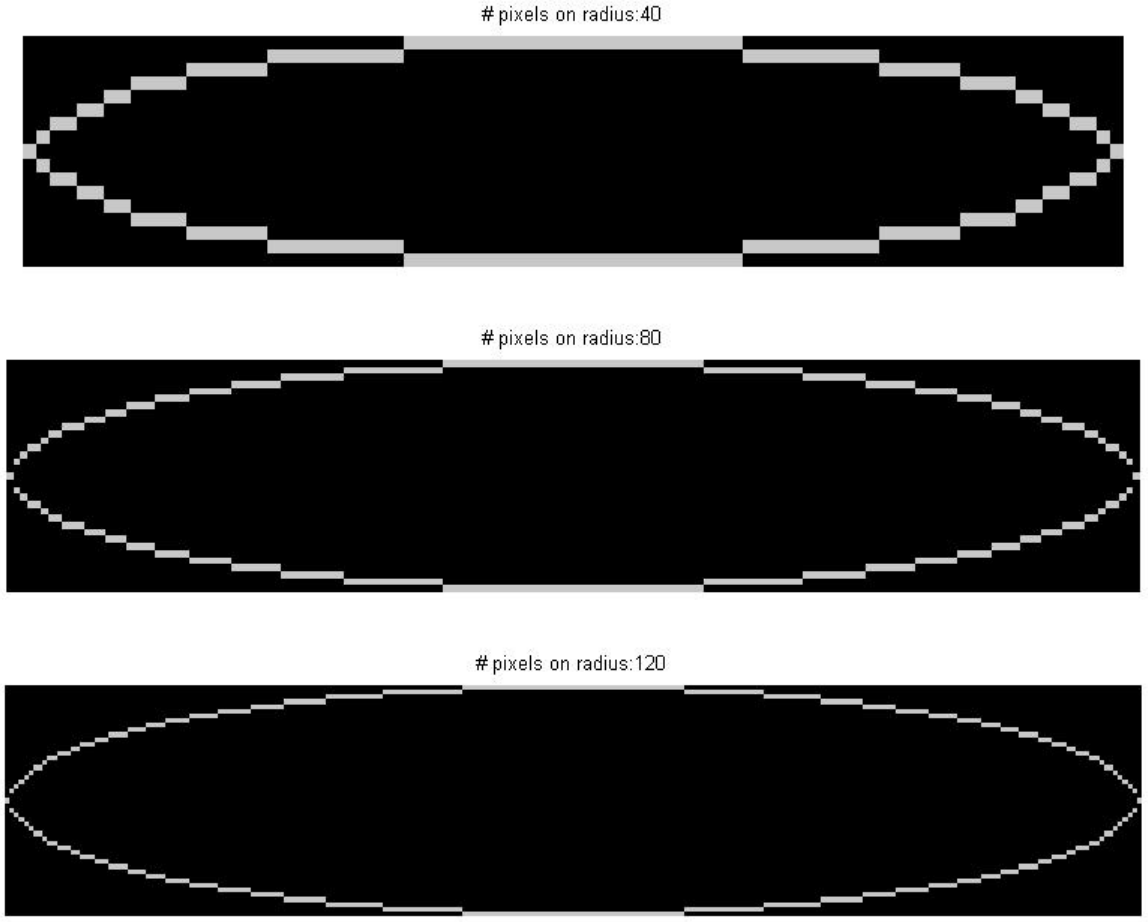
Each time we move on with one pixel leftward and decide whether to place it at same vertical position or one vertical unit downwards according to the middle point of the two choices.

If the middle point is in the ellipse, then same vertical level point is more precise to present the ellipse.

Otherwise, point with one vertical unit downwards is more precise.

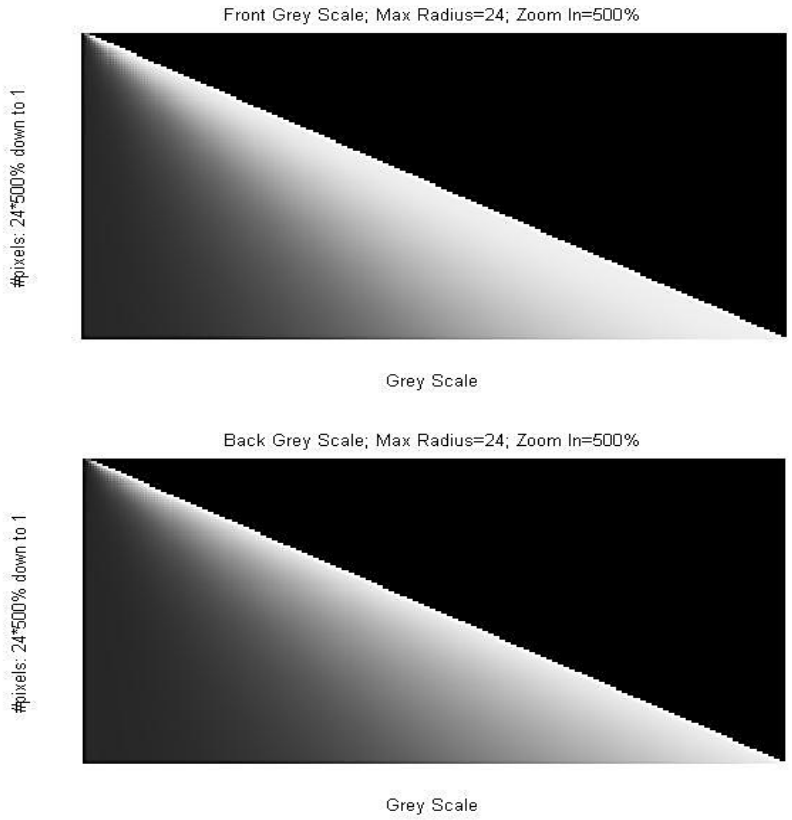
We get the up-right quarter of ellipse in this way and mirror it to get the complete ellipse.

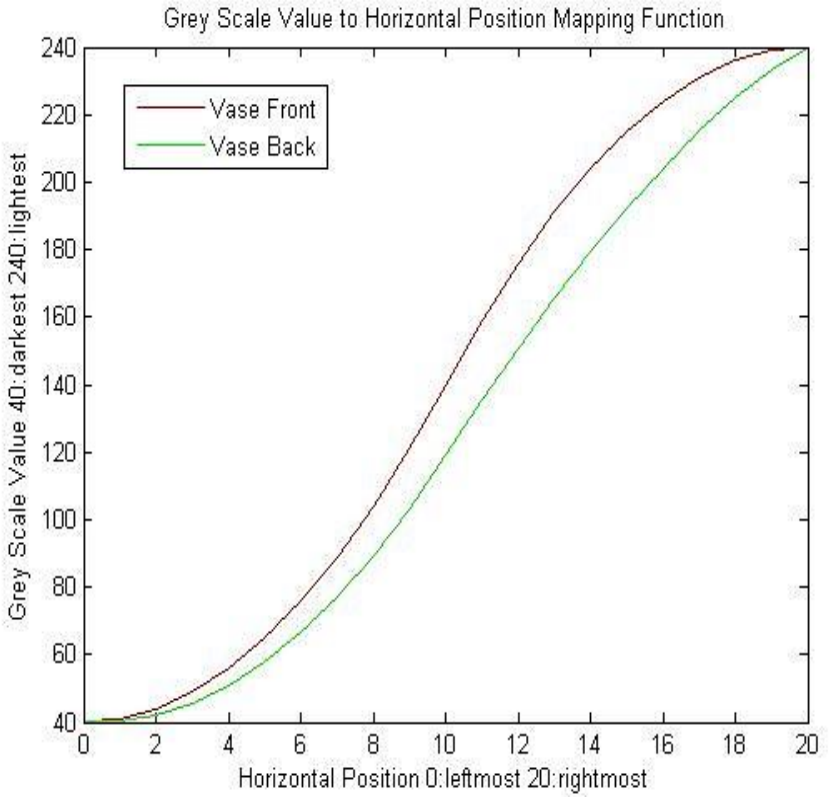
HOW TO DRAW EACH ELLIPSE



Ellipses we get by middle point method at three different precisions

How we get grey scale





Grey scale effect together with grey scale function

HOW WE GET SCORE

Care more about shape alikeness !

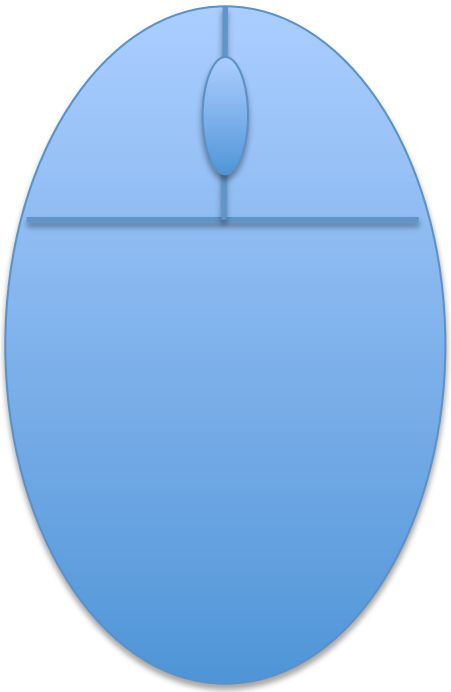
Score parameter: Discrepancy=sum(abs(ideal ratio-current ratio))

Ctrl Point no.	Hard model parameter	Ideal Ratio CP_i+1/CP_i	Middle model parameter	Ideal Ratio	Easy model parameter	Ideal Ratio
3	25		0.6		20	
4	33	1.32	0.75	1.3	20	1
5	31	0.94	0.89	1.2	20	1
6	23	0.74	0.9	1	20	1
7	50	2.17	0.89	1	20	1
8	36	0.72	0.75	0.8	20	1
9	33	0.92	0.6	0.8	20	1
10	48	1.45	0.5	0.8	20	1
11	50	1.04	0.5	1	20	1
12	33	0.66	0.5	1	20	1
13	26	0.79	0.5	1	20	1
14	25	0.96	0.5	1	20	1
15	27	1.08	0.5	1	20	1
16	31	1.15	0.5	1	20	1

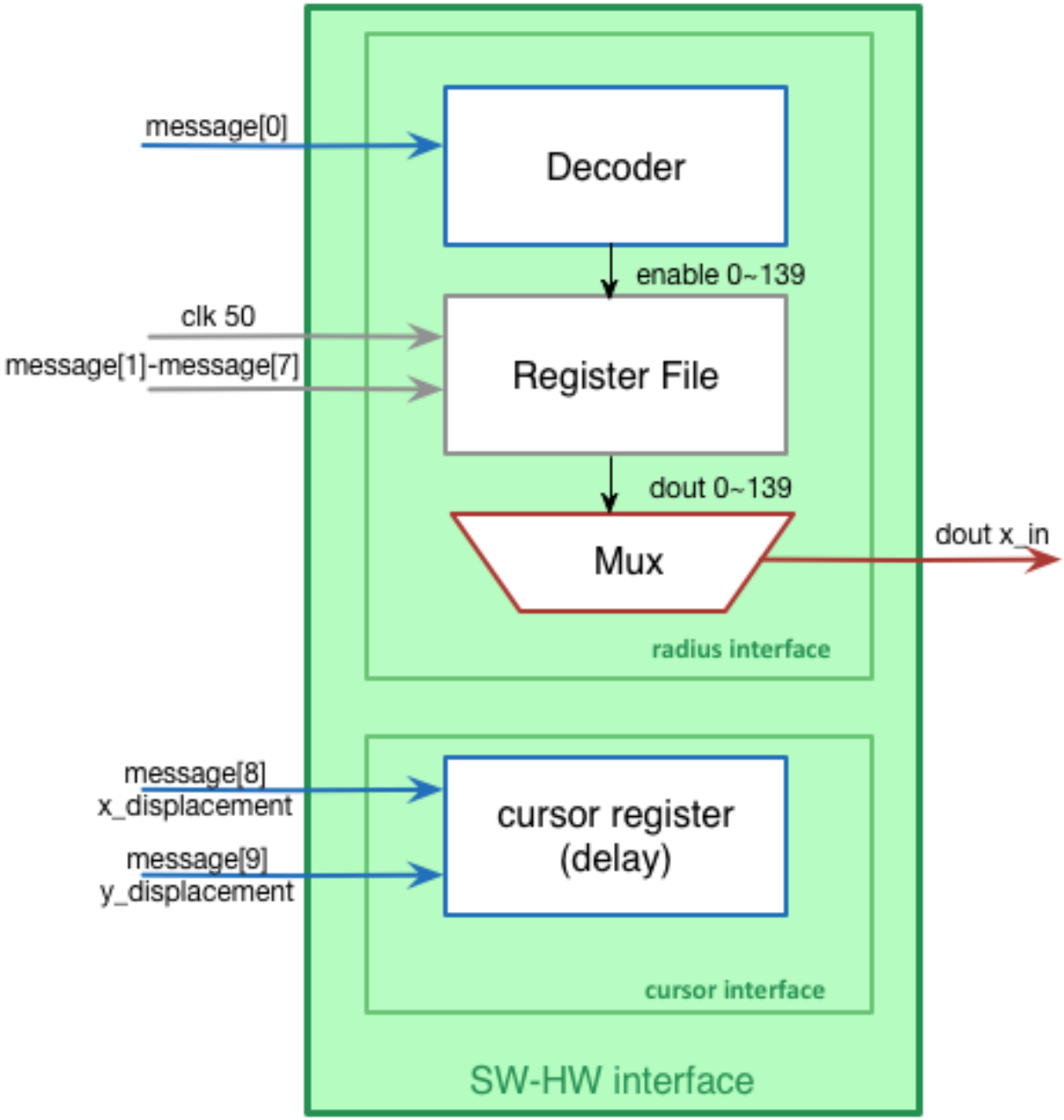
About mouse

Various_function (left/right click; mouse wheel)
Add0 (mouse wheel up/down)
X displacement
Y displacement

X/Y displacement=> current pottery position
mouse display position
Add0 =>to restart
Left click =>enlarge current position
Right click =>shrink current position
Mouse wheel click =>begin playing/to score



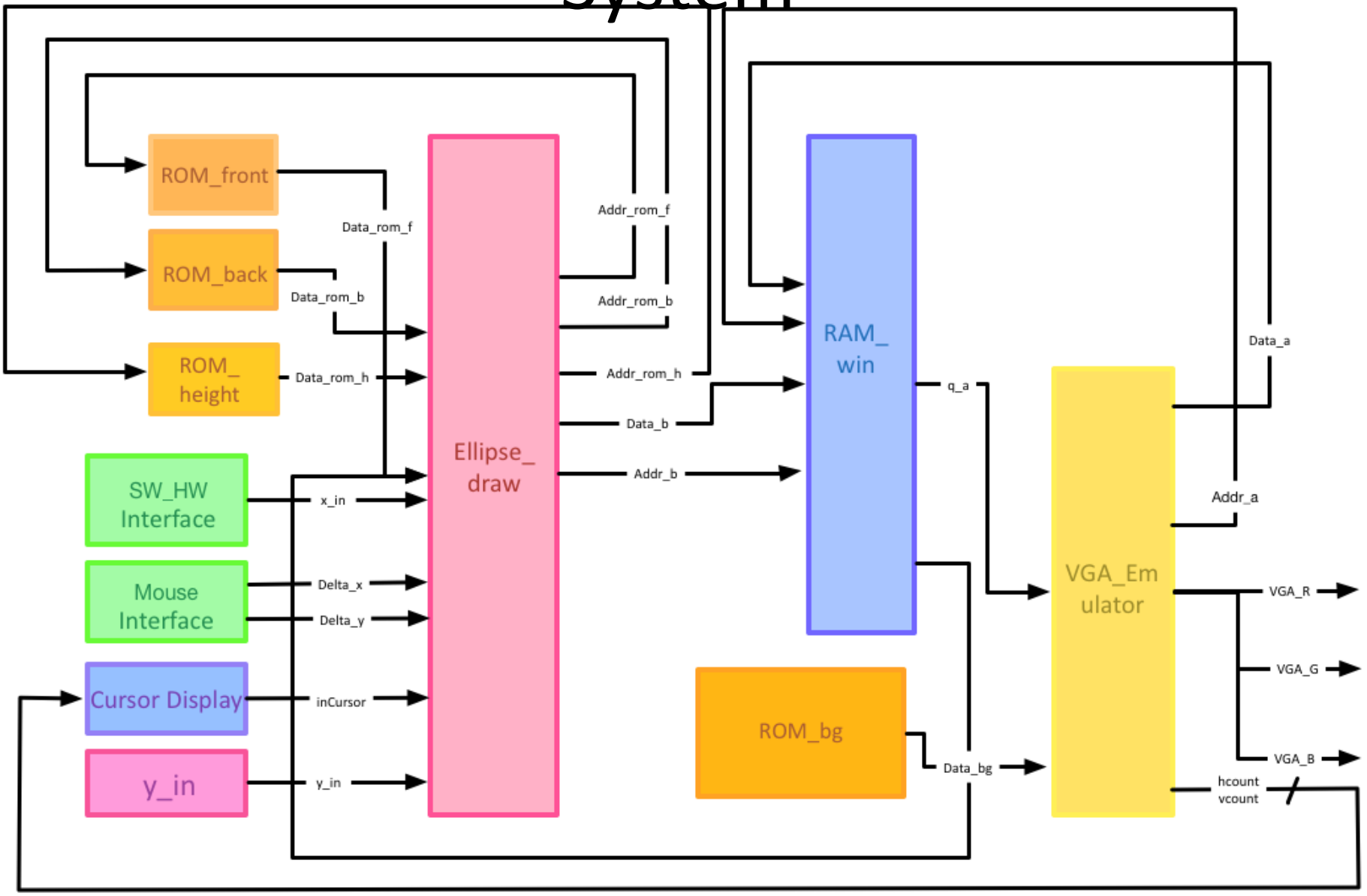
SW-HW Interface



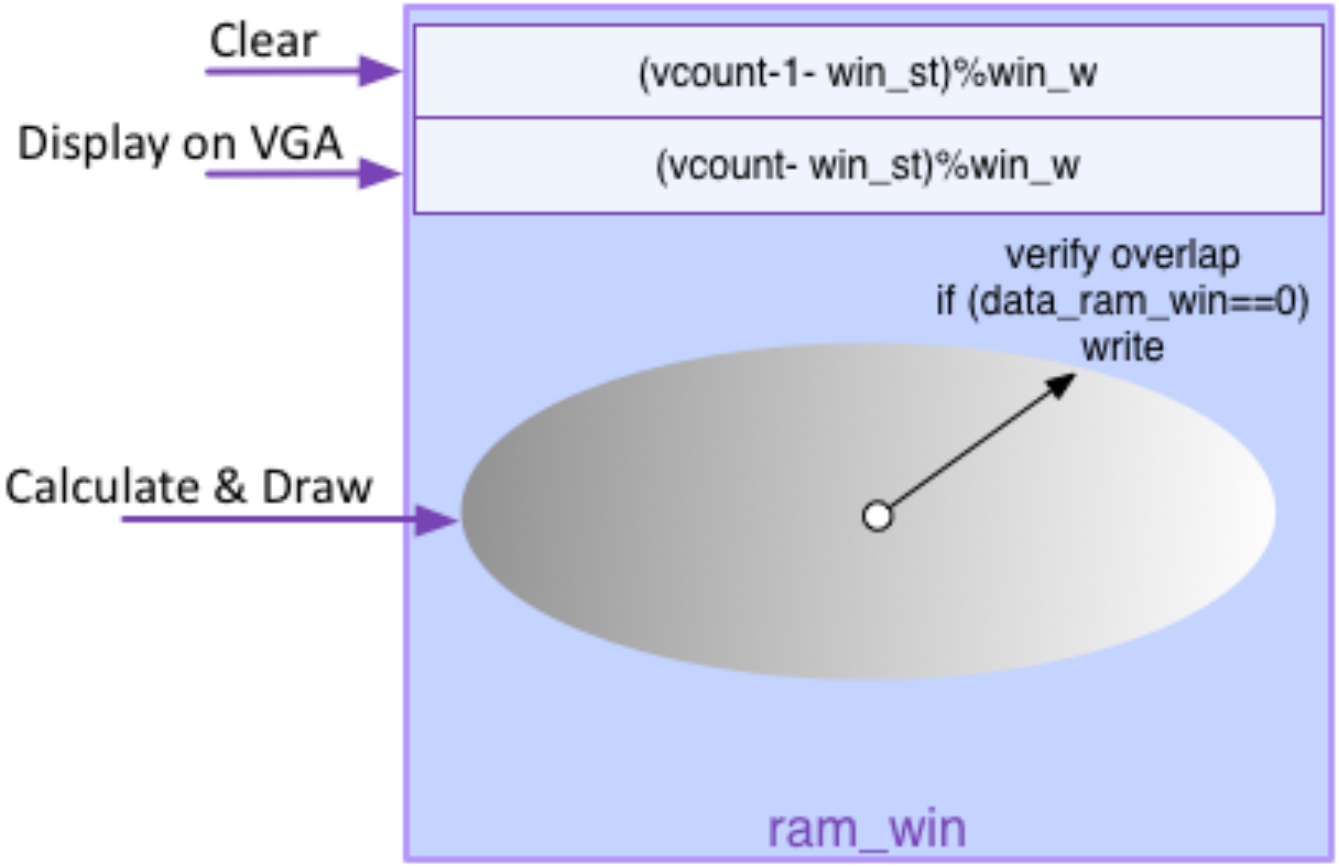
Hardware Overview

- There are 7 main blocks in hardware including **interfaces** between hardware and software (radius, cursor position, mouse), **memories** (storage roms, ram window), **ellipse drawing block** and **VGA emulator**.

Top Level View of the Hardware System



1. RAM window



Two-port RAM: Port A: Display and Clear
Port B: Write data and determine overlap

Port A: Display and Clear

```
else if (VGA_CLK==0&&enable==1&&vcount==65)begin
      rden_a<=0;
      wren_a<=1;
       address_a <= ((win_h-1)%win_h)*win_w + (hcount[10:1]-120);
      data_a<=0;
end
```

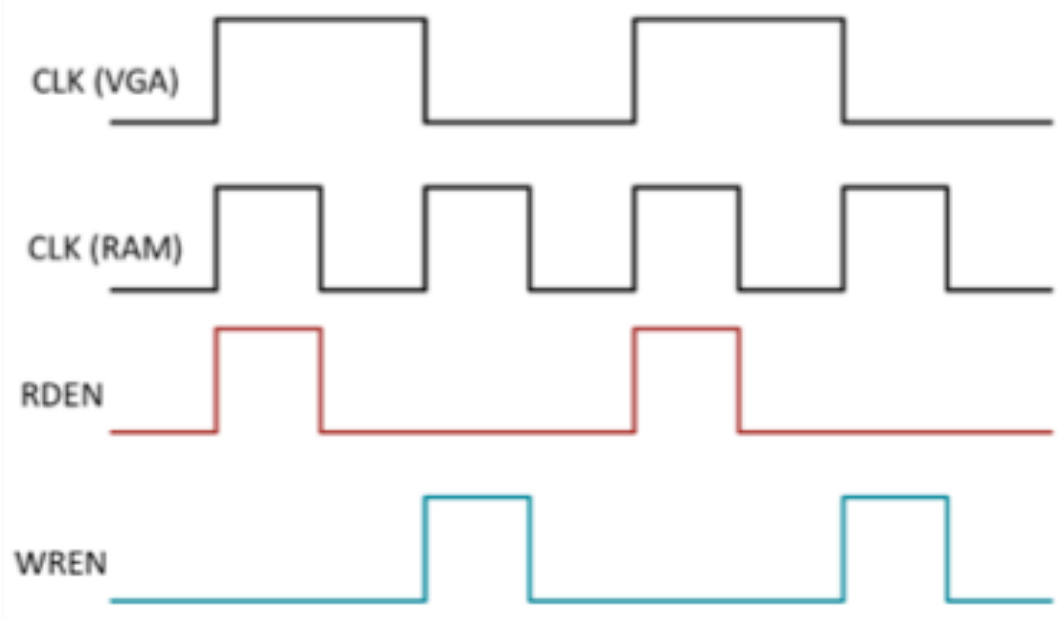
Port A is used to display the data in the ram window on VGA and clear the previous data line. When the data line A is displayed on screen, data line (A-1) is being written value ‘0’.

Port B: Write data and determine overlap

```
cent_y =(y_in-65);
if (four==2'd0) begin
    win_y = (cent_y + data_rom_h+ecc_count)% win_h;
   win_x = win_w/2 + x_count;
   addr_win = win_y*win_w + win_x;
end
```

The way of getting the position of which line is being displayed and which line is being erased is by taking the mod: (vcount-win_st)%win_w, (vcount-1-win_st)%win_w. Win_st is the start position of our window and win_w is the window width.

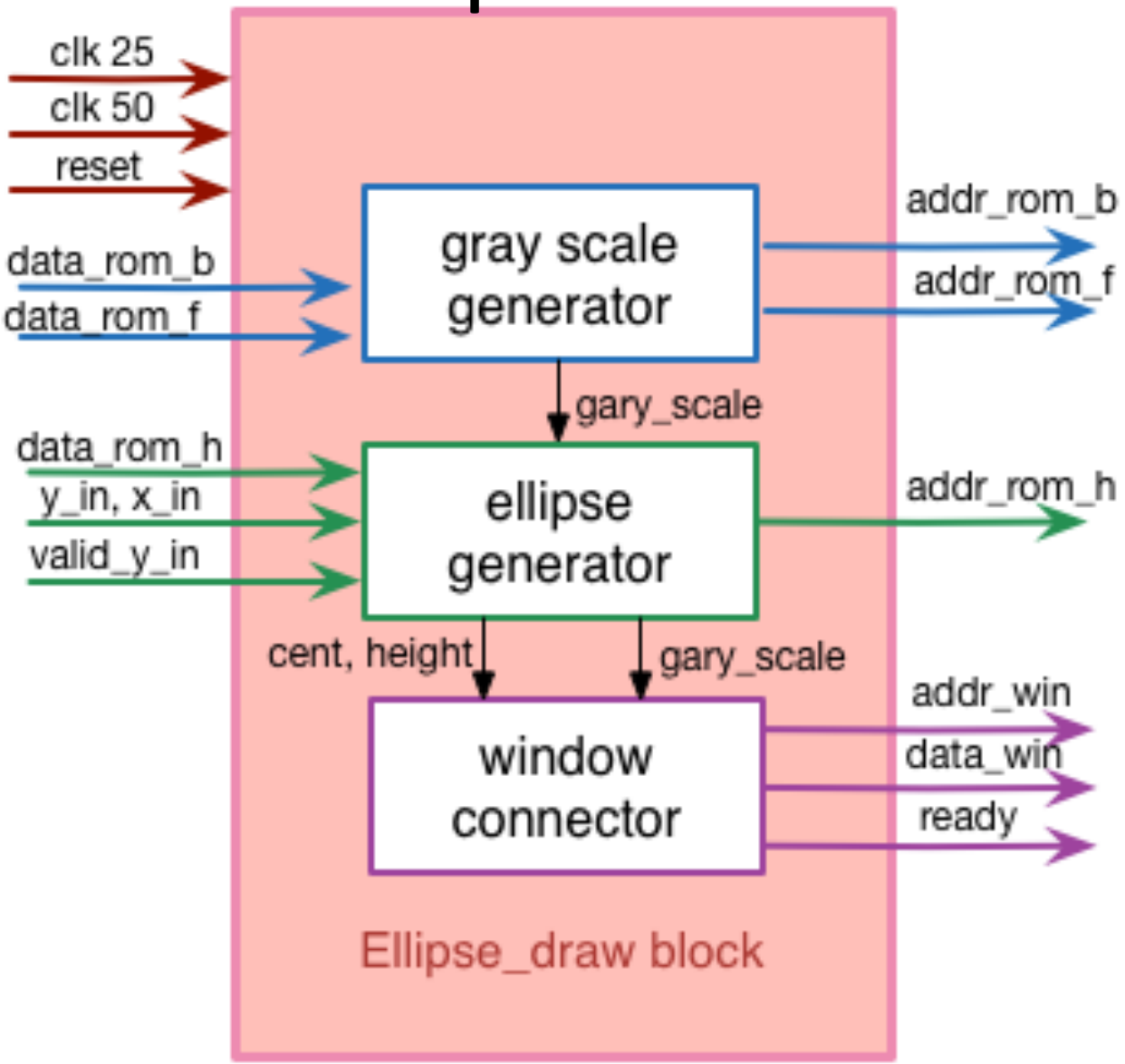
Timing Analysis



In the first cycle of the RAM clock, the RAM gets a read enable signal and port_a is going to read the data from RAM window and display it on the screen, and port_b is going to read the data from RAM window, and check the overlap information.

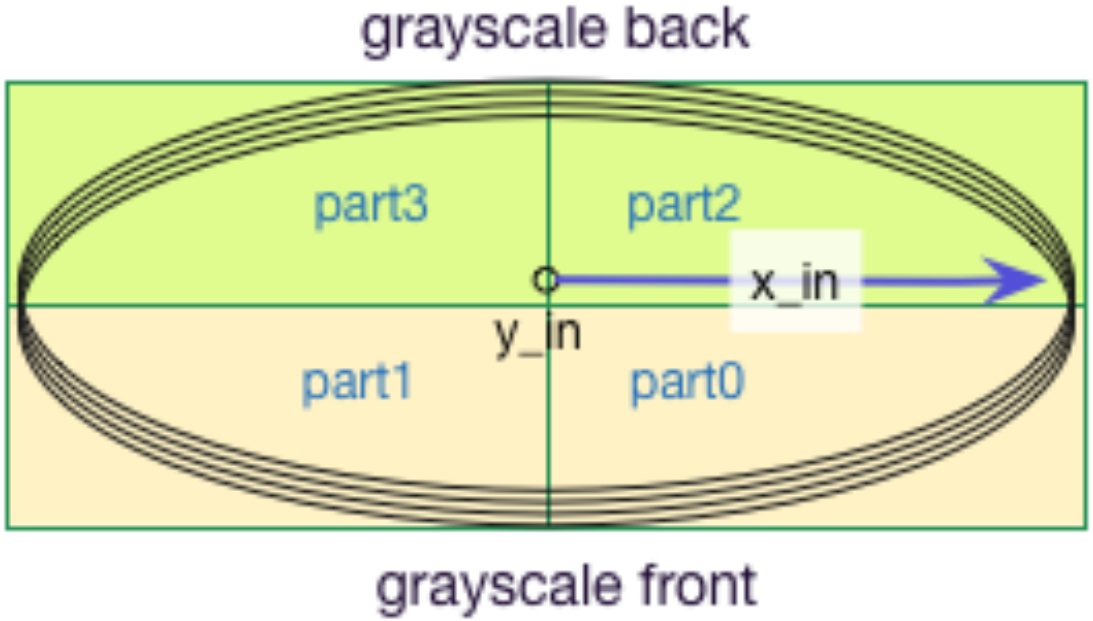
In the second cycle of the RAM clock, the RAM is write enabled and port_a will write zero to the first line of the window, which is the erasing of the first line. Port_b is going to be written and draw ellipse.

2. Ellipse Draw



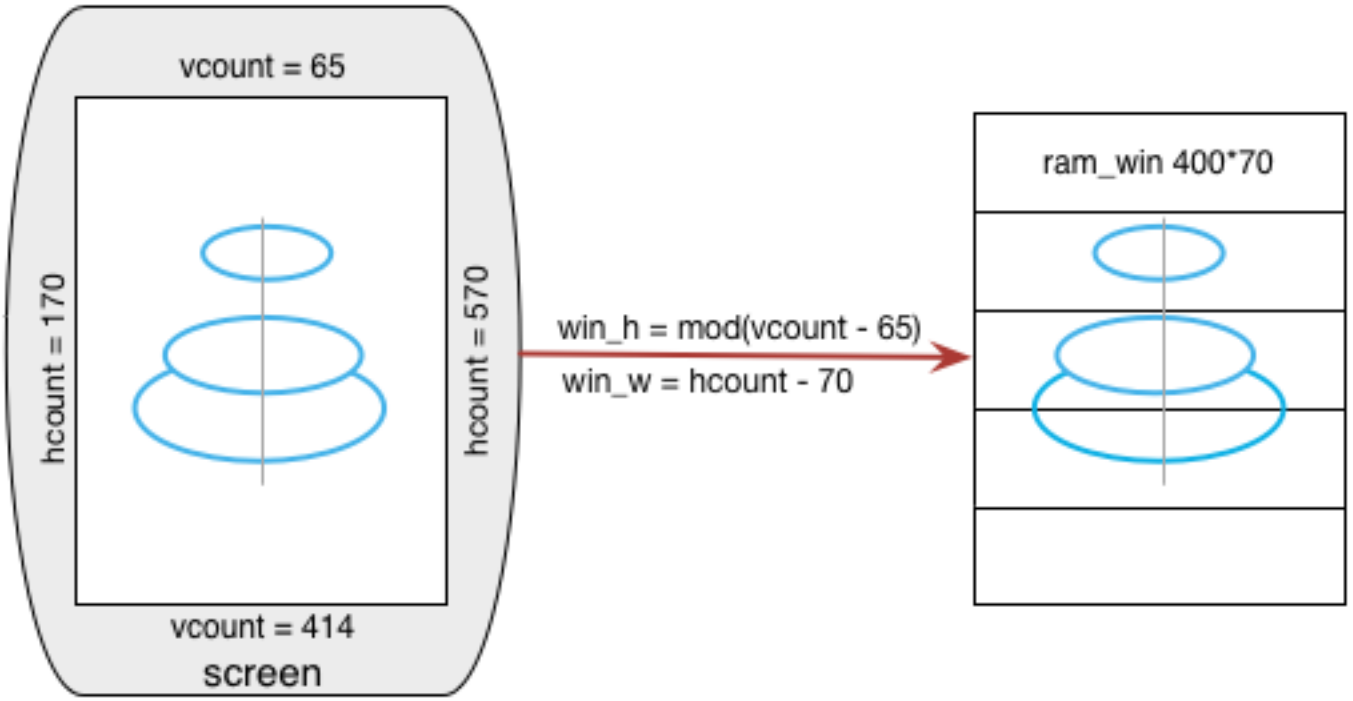
Block Overview

Grayscale and ellipse generator



In order to save the time to calculation the ellipse, we just calculate only one forth of the total ellipse and mirror the other 3 part. In this way, we could finish the calculation before VGA display that part. The order we draw the ellipse is first front and second back because the front part will overlap the back part of the other ellipse.

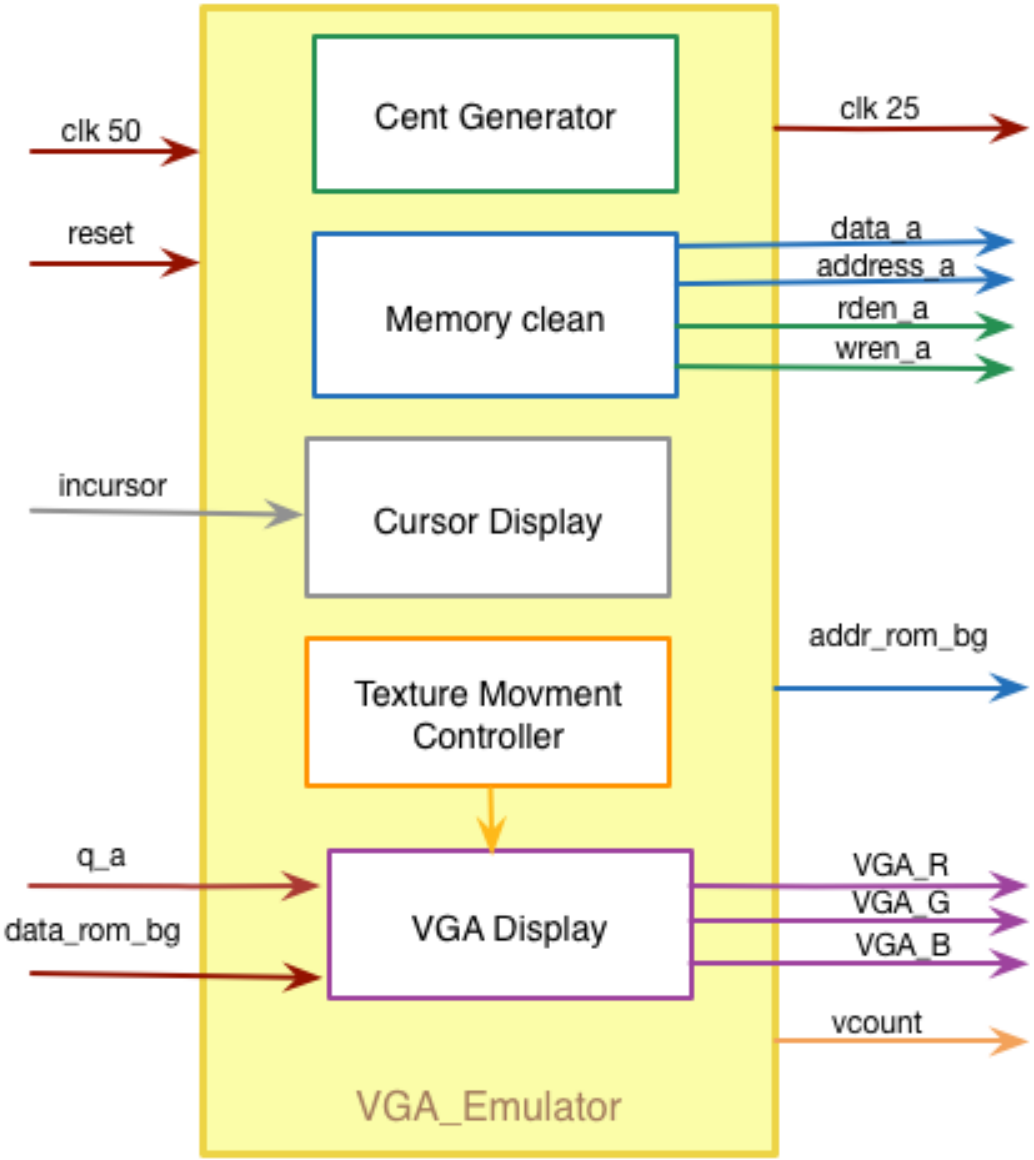
Window Connector



The window connector is a block from which we can transform the location of the ellipses on the screen to the address of the ellipses drawn in the memory window.

To determine the ellipse center position in the window, we will take the mod: (cent_y + data_rom_h+ecc_count)% win_h.

3. VGA Emulator



Block Overview

Cent generator

- We first have to determine whether hcount and vcount are in the window that we can draw ellipses.
- Calculate the address using address_a <= ((vcount-66)%win_h)*win_w + (hcount[10:1]-120).
- If it is the first line in the window, the address: address_a <= ((vcount-65)%win_h)*win_w + (hcount[10:1]-120).

Texture movement controller

- We include a counter in our scheme and compare the grayscale of the pottery with a reference value which keeps changing and the changing speed is determined by the counter_hl.

Issues and Experience

- **Data required to display the whole pottery is very huge and we don't have enough memory to store it.**
  - Implement the idea of memory window and reuse the window.
- **When we first draw the ellipses, there are bunches of ellipses continuously traveling across the screen and cannot be stable.**
  - We change the logic of drawing ellipses from combinational logic to sequential logic.

Issues and Experience

- **There are some patterns, which we don't want, displaying on the screen at every window boundaries, which seemed like noises.**
  - We assign a valid range of the positions where we can draw ellipses, and only in this range can we draw ellipses.
- **When we connect the mouse, the movement of mouse is very sensitive and hard to stabilize.**
  - We scale down the x_displacement and y_displacement of the mouse movement in software.

Lessons Learned

- Familiar with the hardware architecture of the FPGA socKit board.
- Get a deeper understanding of the design flow of a whole system.
- Improve our coding skills in System Verilog and C.
- Become more proficient in CAD tools, including Quartus, Qsys, SignalTap and so on.

Debugging Methods

- Hardware
  - RTL Viewer
  - SignalTap
  - ModelSim
- Software
  - Printf of variables