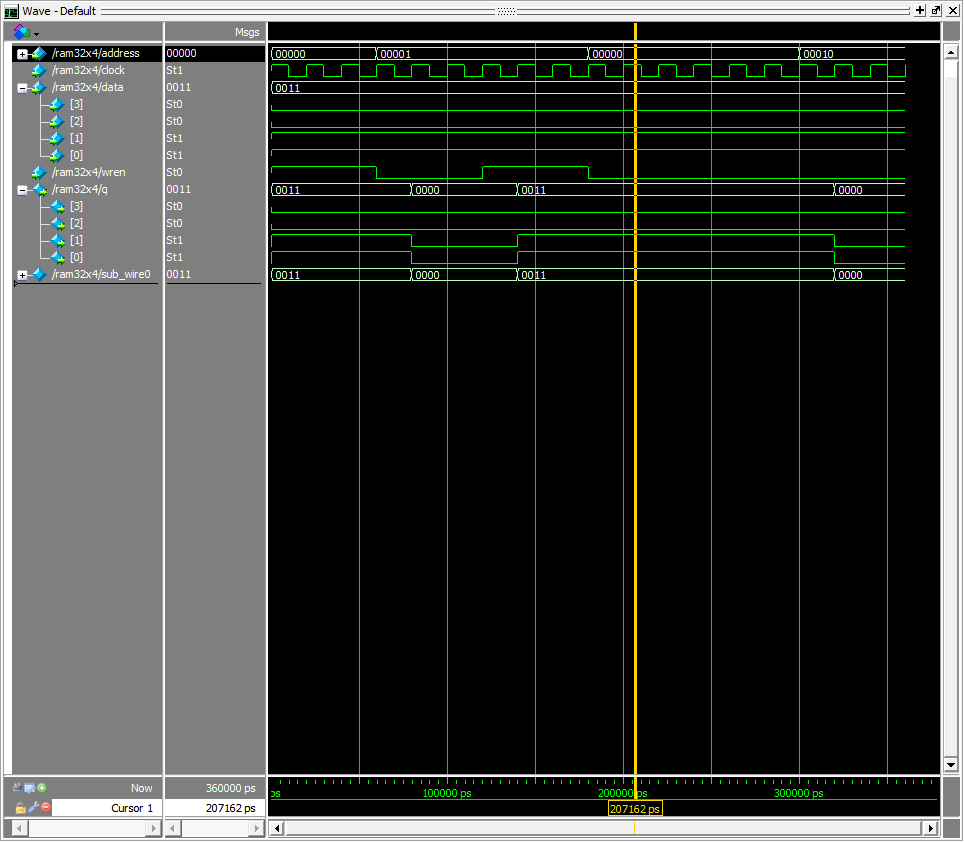
**Lab7 Pre-Lab Report**

Ziyao Chen

1002453409

**Part1**

9.



10.

module part1(SW, KEY, HEX0, HEX2, HEX4, HEX5);

input [9:0] SW;

input [3:0] KEY;

output [6:0] HEX0, HEX2, HEX4, HEX5;

wire [3:0] data\_out

ram32x4 r0(

.address(SW[8:4]),

.clock(~KEY[0]),

.data(SW[3:0]),

.wren(SW[9]),

.q(data\_out)

);

hex\_decoder h0(

.hex\_digit(data\_out),

.segments(HEX0)

)；

hex\_decoder h2(

.hex\_digit(SW[3:0]),

.segments(HEX2)

)；

hex\_decoder h4(

.hex\_digit(SW[7:4]),

.segments(HEX4)

)；

hex\_decoder h5(

.hex\_digit({000, SW[8]}),

.segments(HEX5)

)；

endmodule

module hex\_decoder(hex\_digit, segments);

input [3:0] hex\_digit;

output reg [6:0] segments;

always @(\*)

case (hex\_digit)

4'h0: segments = 7'b100\_0000;

4'h1: segments = 7'b111\_1001;

4'h2: segments = 7'b010\_0100;

4'h3: segments = 7'b011\_0000;

4'h4: segments = 7'b001\_1001;

4'h5: segments = 7'b001\_0010;

4'h6: segments = 7'b000\_0010;

4'h7: segments = 7'b111\_1000;

4'h8: segments = 7'b000\_0000;

4'h9: segments = 7'b001\_1000;

4'hA: segments = 7'b000\_1000;

4'hB: segments = 7'b000\_0011;

4'hC: segments = 7'b100\_0110;

4'hD: segments = 7'b010\_0001;

4'hE: segments = 7'b000\_0110;

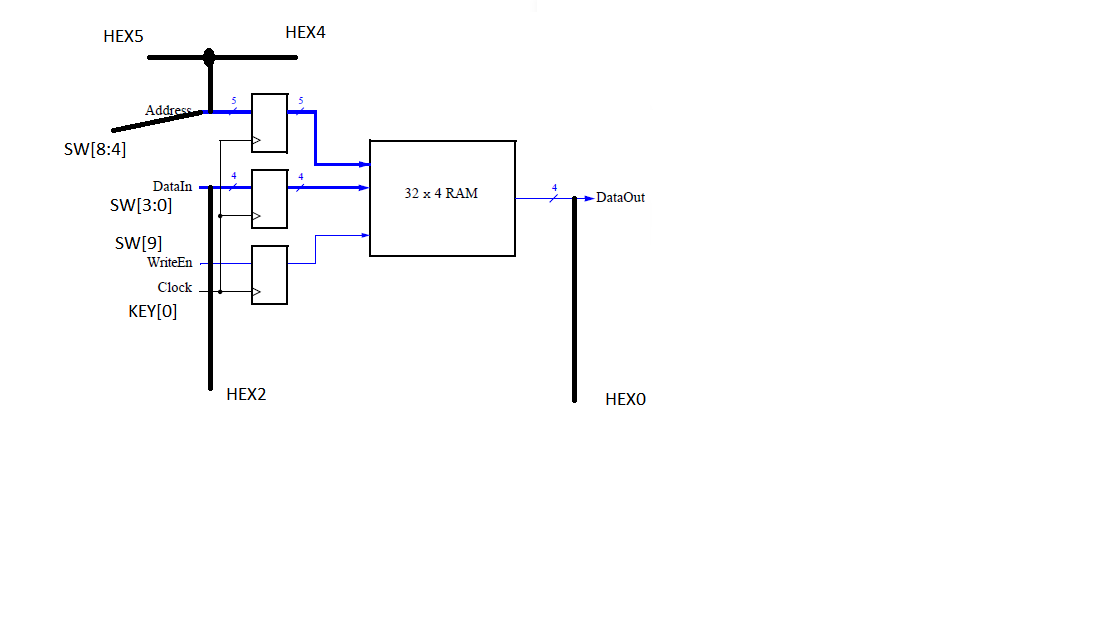
4'hF: segments = 7'b000\_1110;

default: segments = 7'h7f;

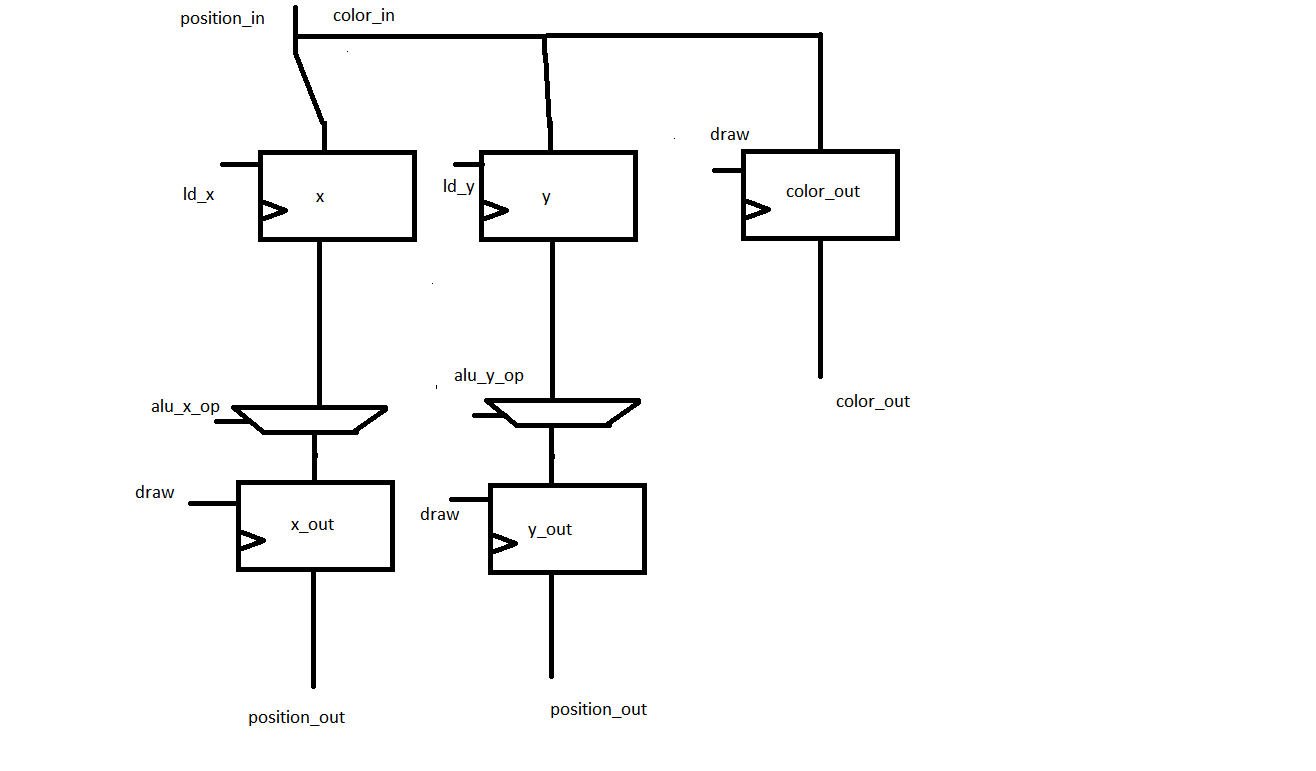
endcase

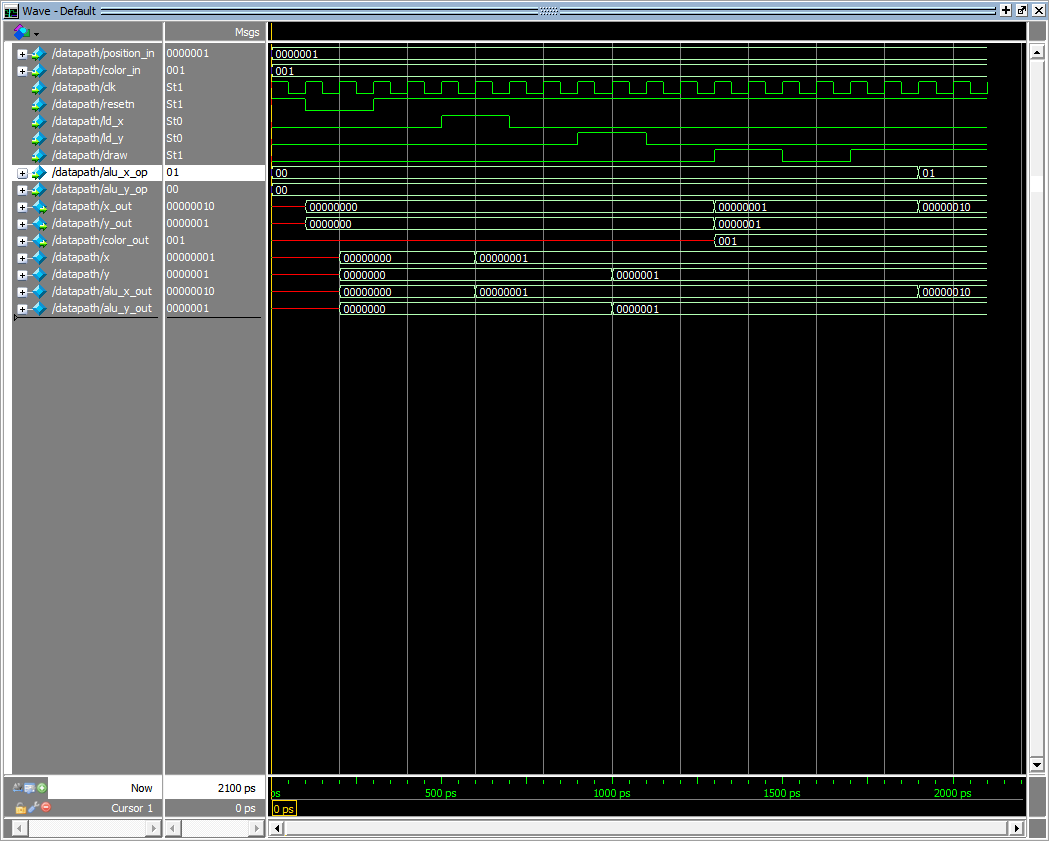
endmodule

11.

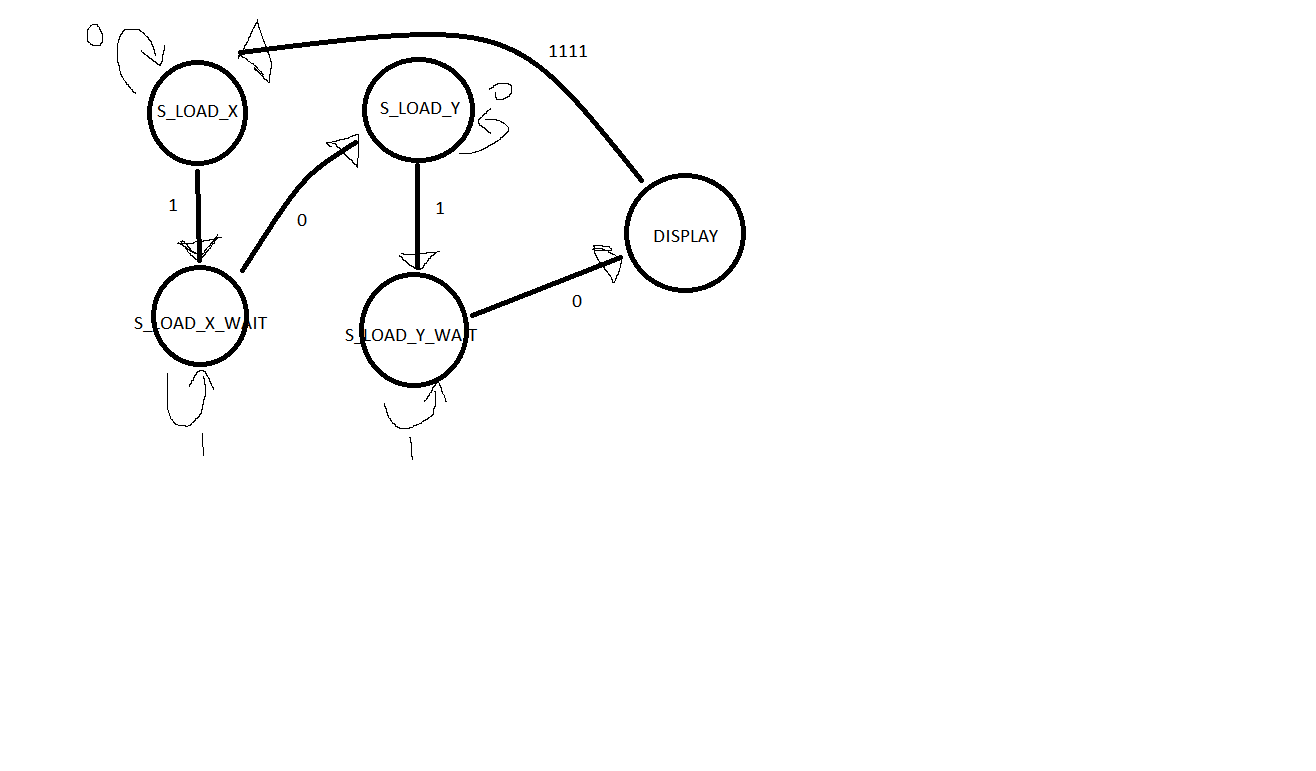
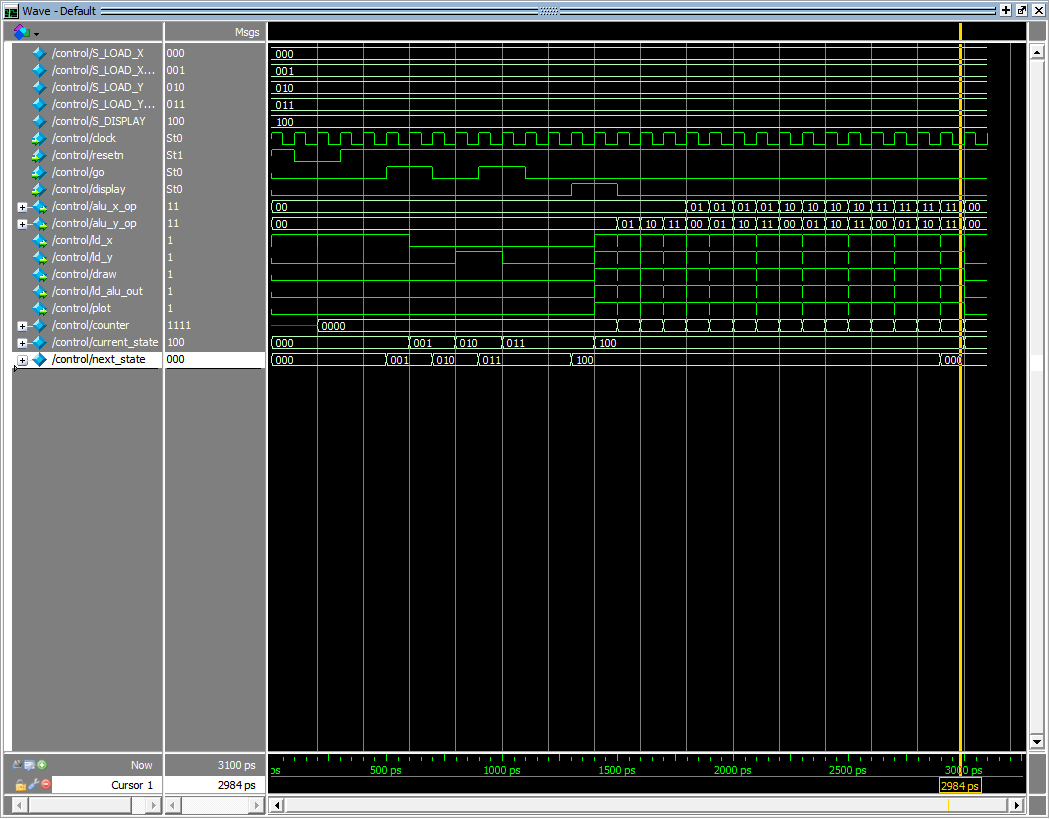


**Part2**

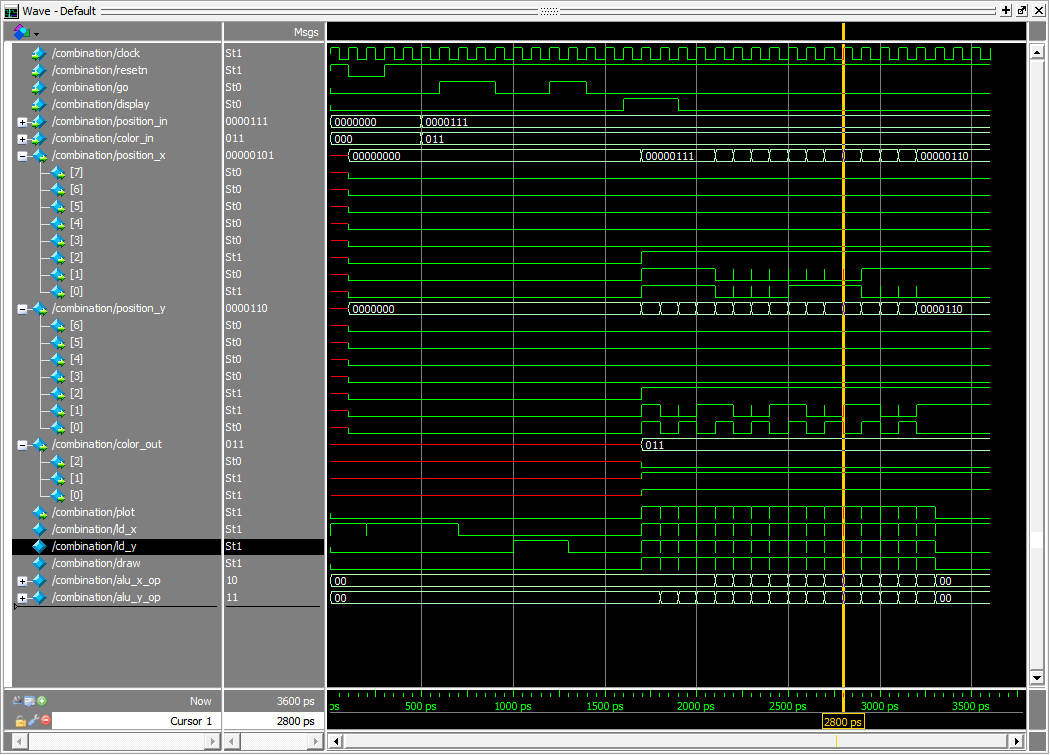
1. 





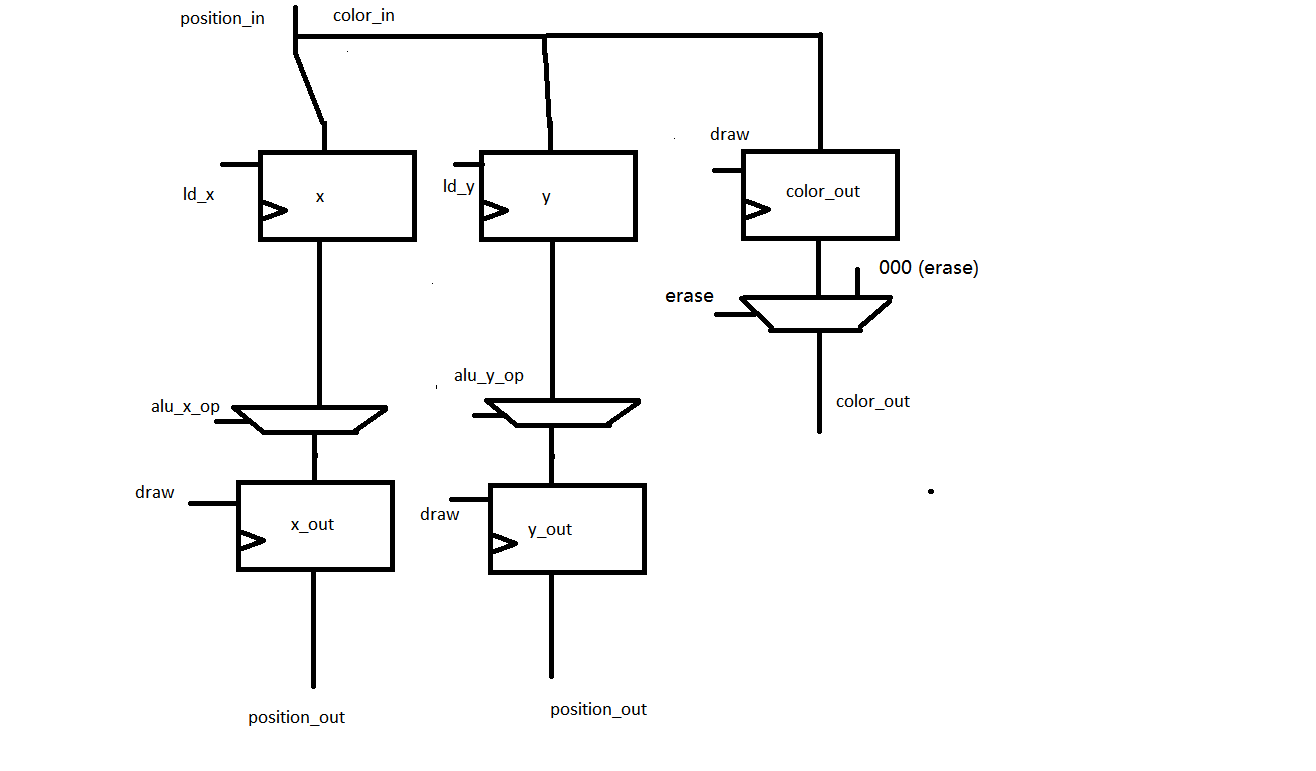


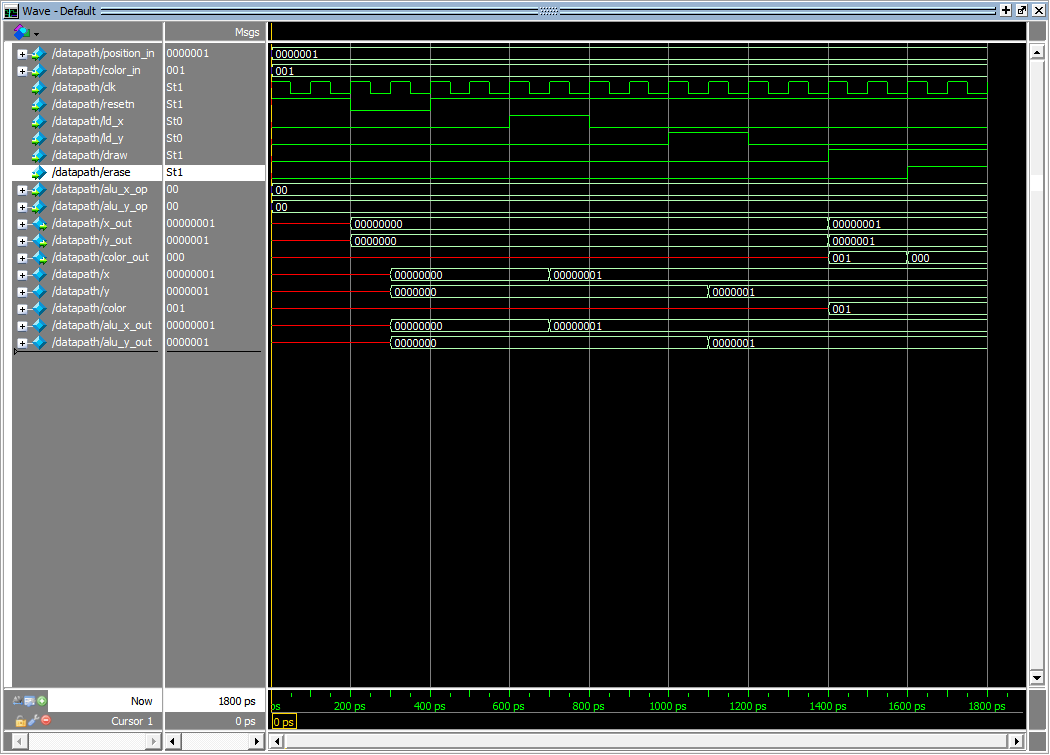




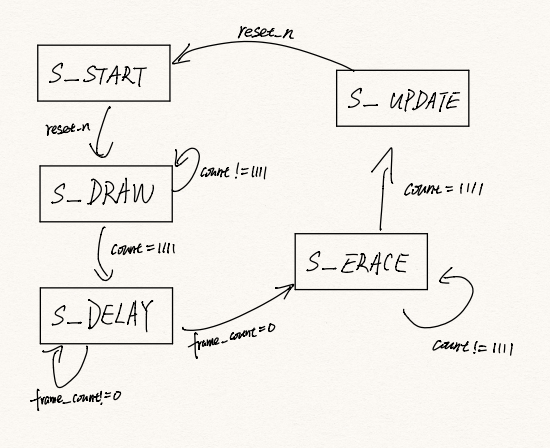
**Part3**





****

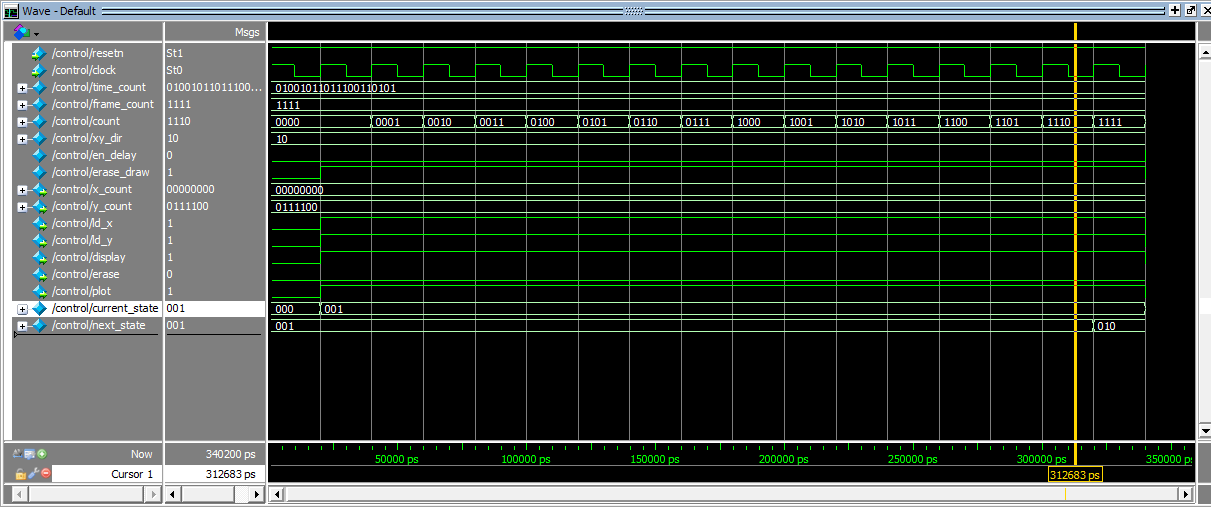




State table:

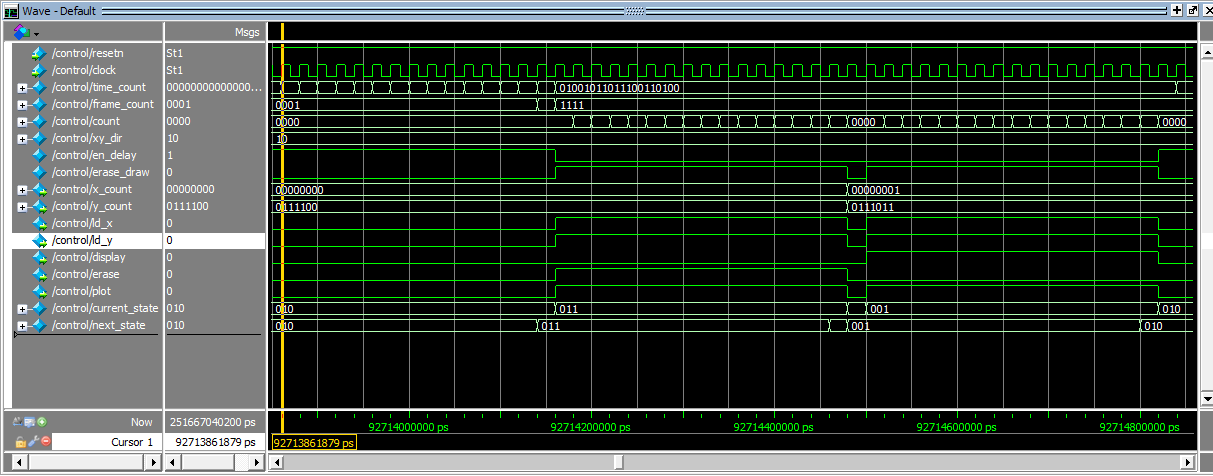
|  |  |  |  |
| --- | --- | --- | --- |
| current\_state | count == 1111 | frame == 0 | next\_state |
| S\_START | x | x | S\_DRAW |
| S\_DRAW | 0 | x | S\_DRAW |
| S\_DRAW | 1 | x | S\_DELAY |
| S\_DELAY | x | 0 | S\_DELAY |
| S\_DELAY | x | 1 | S\_ERASE |
| S\_ERASE | 0 | x | S\_ERASE |
| S\_ERASE | 1 | x | S\_UPDATE |
| S\_UPDATE | x | x | S\_START |

Simulation of control:

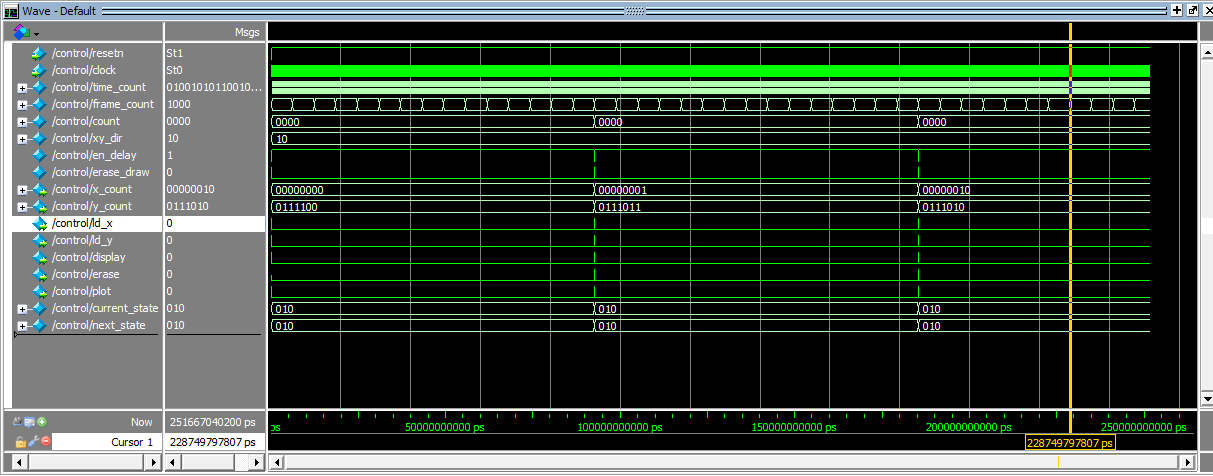


0-20 ns (the first clock cycle): current\_state = S\_START, all the counters are reseted, position set to (0,60), direction is 10(up-right).

20-340 ns (the next 16 clock cycles): current\_state = S\_DRAW, at the very end, cuurent\_state changed to S\_DELAY.



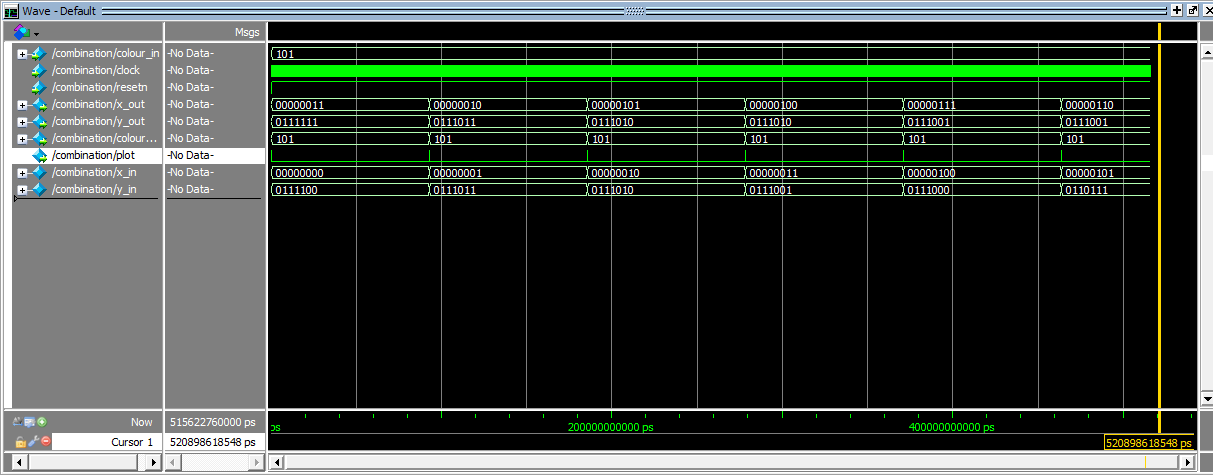
After some time of delay, the state goes to the S\_ERASE state and maintained for 16 clock cycles. Then it goes to the S\_UPDATE state for 1 clock cycle, the position have been updated to (1, 59). After updating the location, the state goes to S\_DRAW again(for 16 clock cycles), then to S\_DELAY…



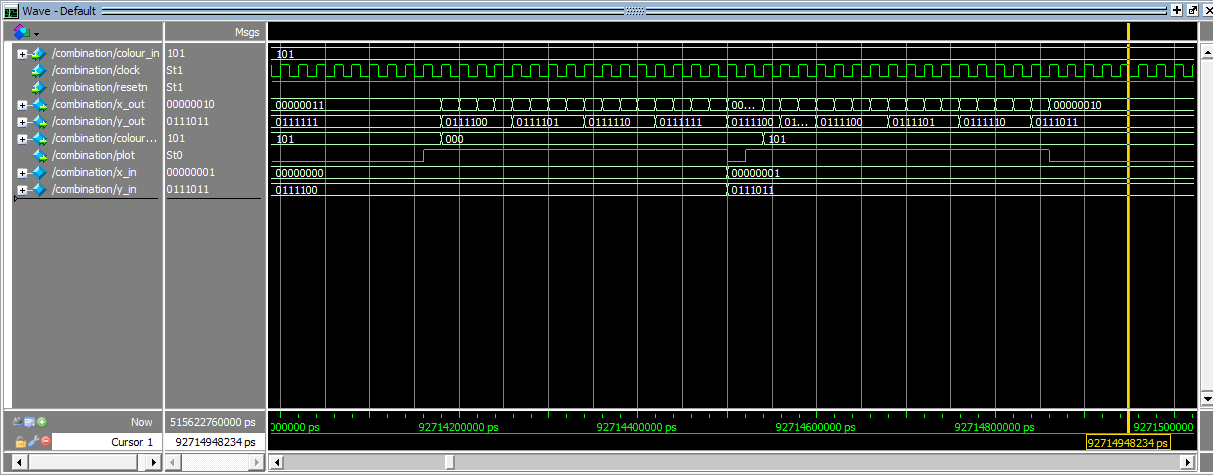
This is what the overall simulation looks like. After two states cycles, the (x,y) position have been updated to (2, 58).

3.

Simulation of combined:



After each delay cycle, the location is updated.



Expand the erase-update-draw part, we see that all 16 previous draws pixels are erased(coloured to black), the location updated and then the next 16 pixels are drawn.