HDL Digital Design (Graduate Level)
Spring 2024

HOMEWORK
REPORT

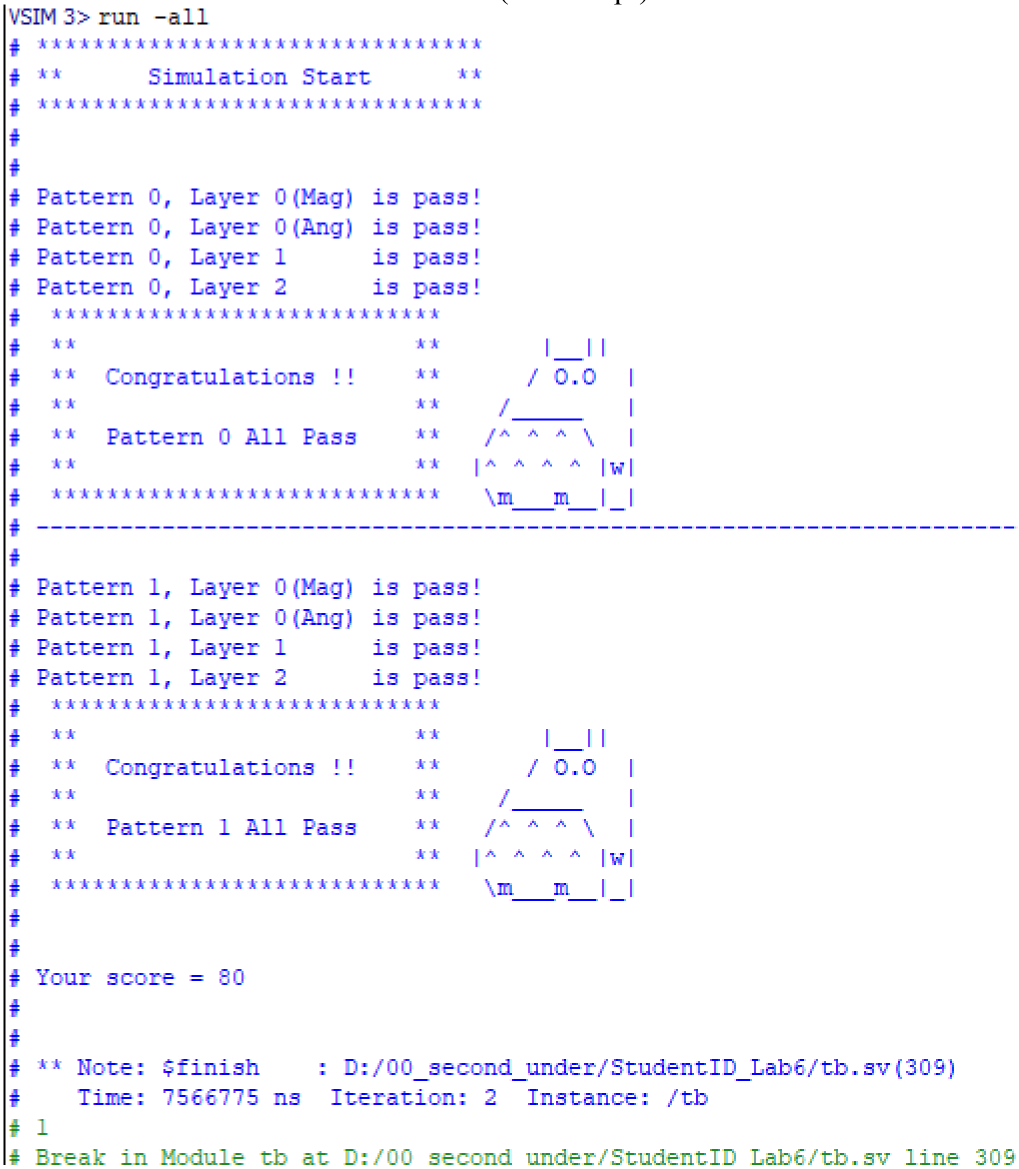
Must do self-checking before submission:

□ Compress all files described in the problem into one zip file.
□ All files can be compiled under ModelSim environment.
□ All port declarations comply with I/O port specifications.
□ Organize files according to File Hierarchy Requirement
□ No waveform files or project file in deliverables

Due Date: 2024/04/11 8:59 a.m.

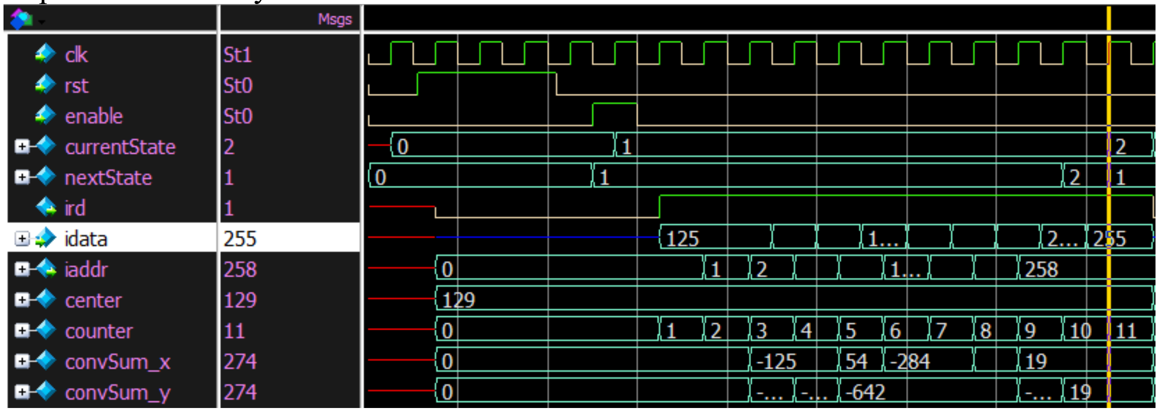
Student name: 蔡承哲
Student ID: Q36111150

1. Your simulation result on the terminal (Transcript) .



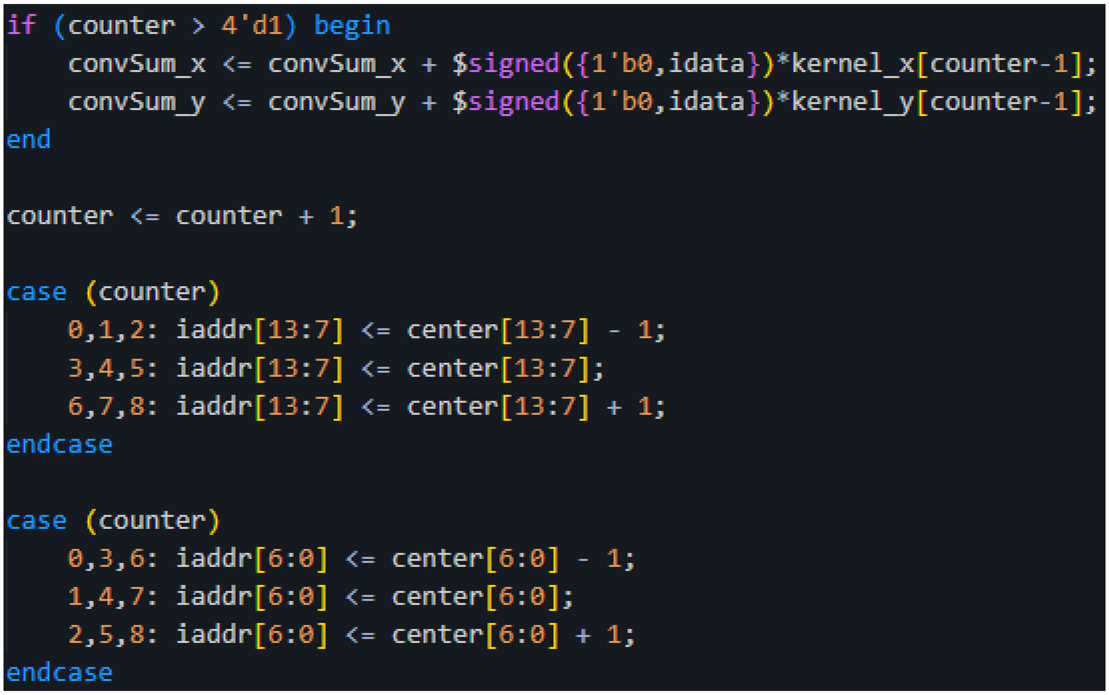
```
VSIM 3> run -all
# ********************************
# **      Simulation Start      **
# ********************************
#
#
# Pattern 0, Layer 0(Mag) is pass!
# Pattern 0, Layer 0(Ang) is pass!
# Pattern 0, Layer 1      is pass!
# Pattern 0, Layer 2      is pass!
#  ****************************
#  **                        **       |__||
#  **  Congratulations !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Pattern 0 All Pass    **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ----------------------------------------------------------------------
#
# Pattern 1, Layer 0(Mag) is pass!
# Pattern 1, Layer 0(Ang) is pass!
# Pattern 1, Layer 1      is pass!
# Pattern 1, Layer 2      is pass!
#  ****************************
#  **                        **       |__||
#  **  Congratulations !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Pattern 1 All Pass    **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
#
#
# Your score = 80
#
#
# ** Note: $finish    : D:/00_second_under/StudentID_Lab6/tb.sv(309)
#    Time: 7566775 ns  Iteration: 2  Instance: /tb
# 1
# Break in Module tb at D:/00 second under/StudentID Lab6/tb.sv line 309
```

2. Explain the result by waveform.

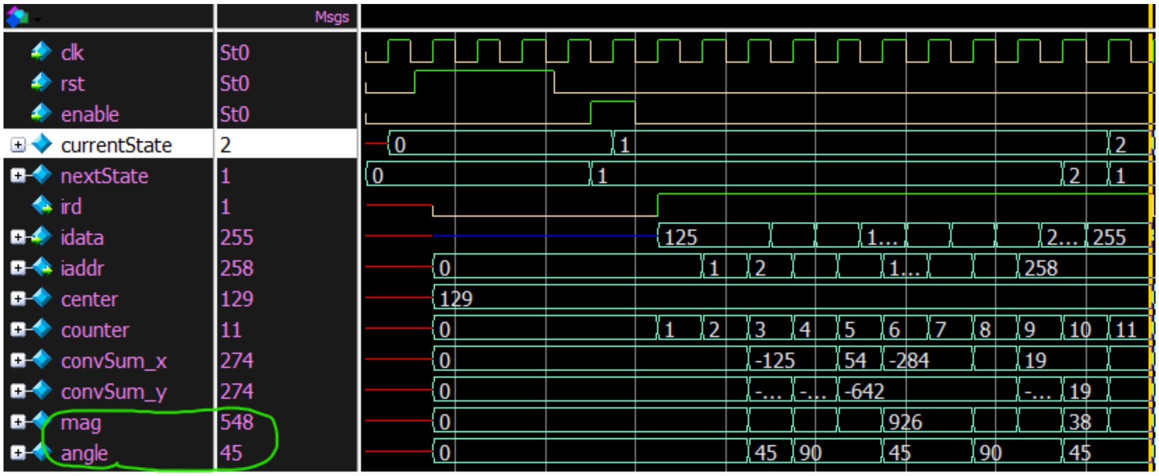


	Msgs
clk	St1
rst	St0
enable	St0
currentState	2
nextState	1
ird	1
idata	255
iaddr	258
center	129
counter	11
convSum_x	274
convSum_y	274

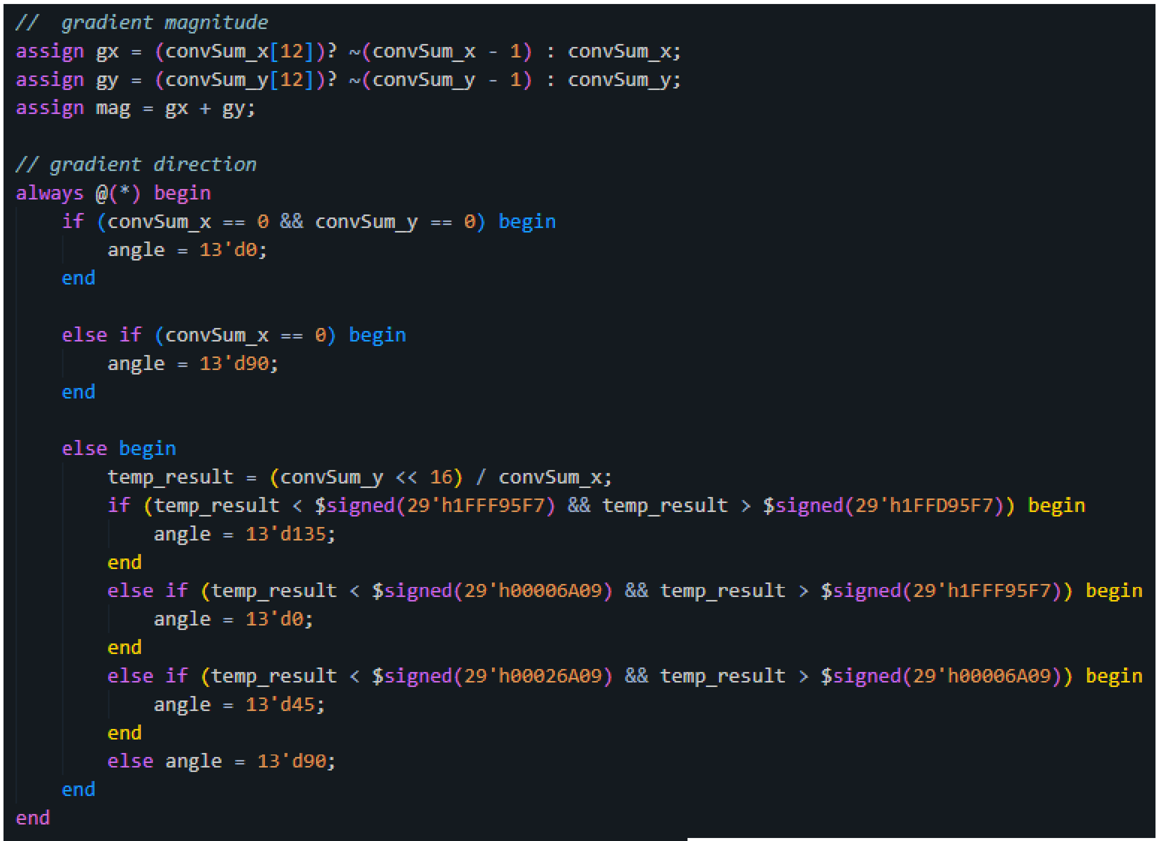
首先在 Layer0 要先做 convolution，由於這次不用做 zero padding，因此我把 center 直接設為 129，然後依序移動，直到 16254，就代表做完整張圖的 convolution。讀取對應 pixel 的方式如下，然後把 idata 乘上 kernel，並依次做累加。當作完一次 convolution，就會跳到 WB0 準備寫入 magnitude 與 angle。



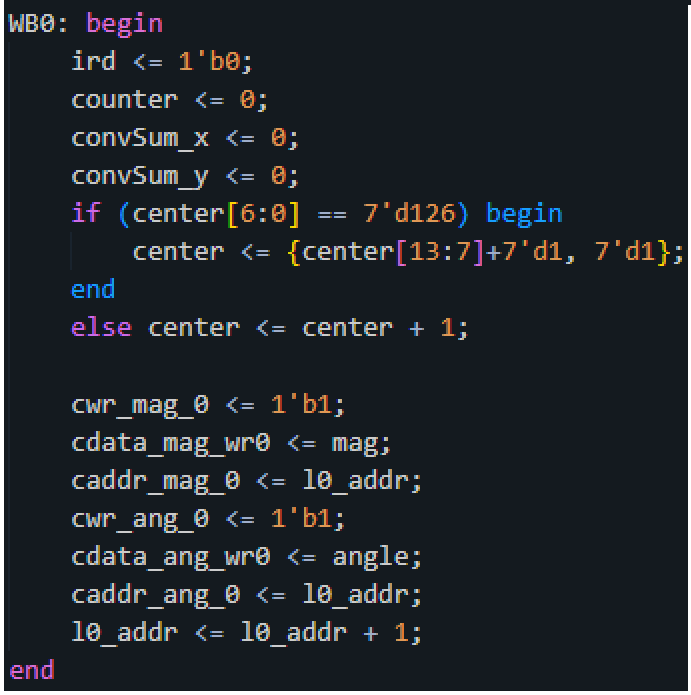
```
if (counter > 4'd1) begin
    convSum_x <= convSum_x + $signed({1'b0,idata})*kernel_x[counter-1];
    convSum_y <= convSum_y + $signed({1'b0,idata})*kernel_y[counter-1];
end

counter <= counter + 1;

case (counter)
    0,1,2: iaddr[13:7] <= center[13:7] - 1;
    3,4,5: iaddr[13:7] <= center[13:7];
    6,7,8: iaddr[13:7] <= center[13:7] + 1;
endcase

case (counter)
    0,3,6: iaddr[6:0] <= center[6:0] - 1;
    1,4,7: iaddr[6:0] <= center[6:0];
    2,5,8: iaddr[6:0] <= center[6:0] + 1;
endcase
```



	Msgs
clk	St0
rst	St0
enable	St0
currentState	2
nextState	1
ird	1
idata	255
iaddr	258
center	129
counter	11
convSum_x	274
convSum_y	274
mag	548
angle	45

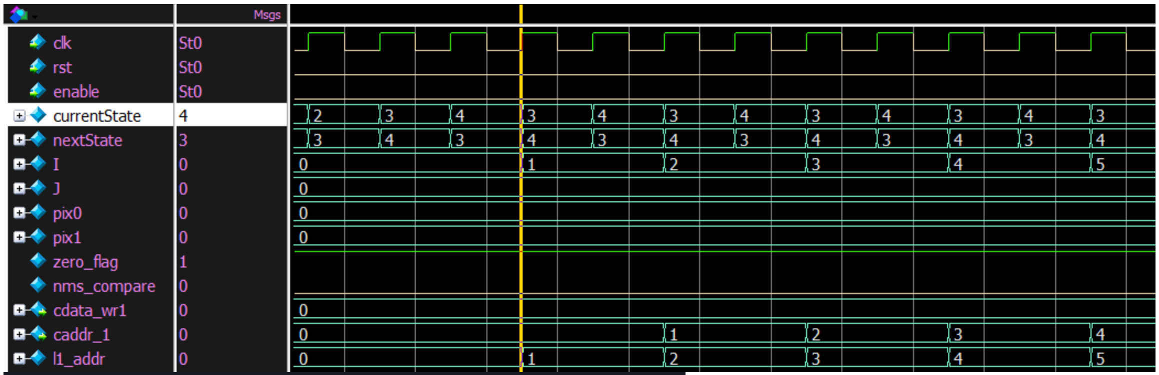


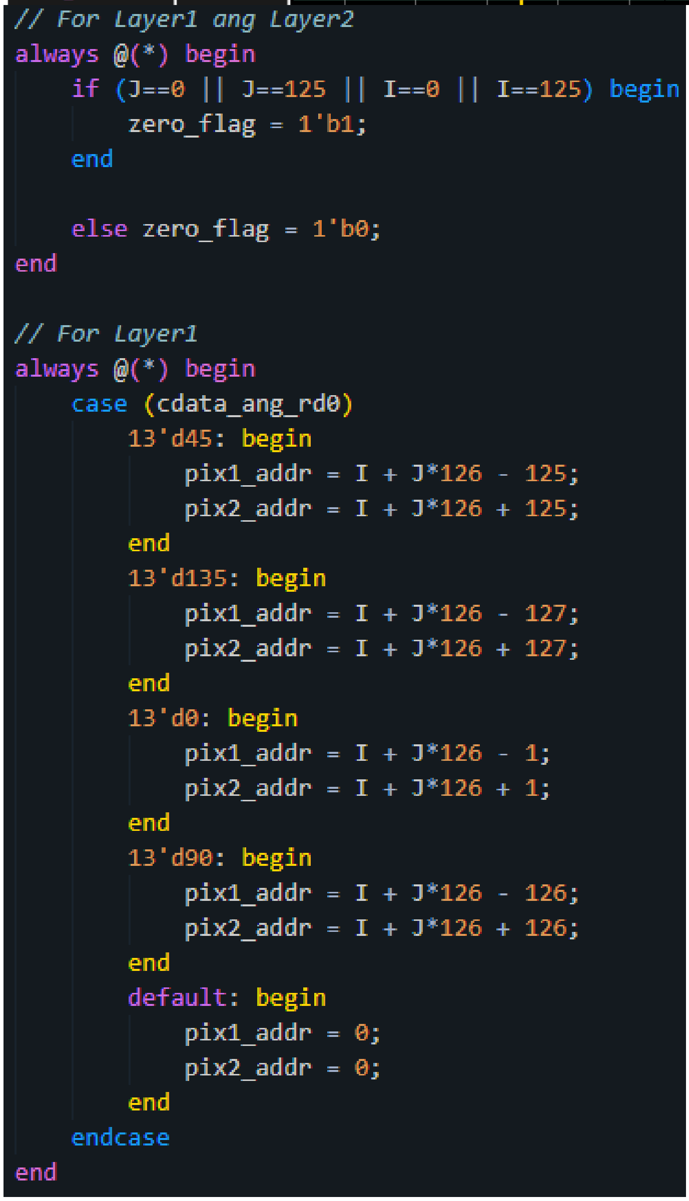
```
//  gradient magnitude
assign gx = (convSum_x[12])? ~(convSum_x - 1) : convSum_x;
assign gy = (convSum_y[12])? ~(convSum_y - 1) : convSum_y;
assign mag = gx + gy;

// gradient direction
always @(*) begin
    if (convSum_x == 0 && convSum_y == 0) begin
        angle = 13'd0;
    end

    else if (convSum_x == 0) begin
        angle = 13'd90;
    end

    else begin
        temp_result = (convSum_y << 16) / convSum_x;
        if (temp_result < $signed(29'h1FFF95F7) && temp_result > $signed(29'h1FFD95F7)) begin
            angle = 13'd135;
        end
        else if (temp_result < $signed(29'h00006A09) && temp_result > $signed(29'h1FFF95F7)) begin
            angle = 13'd0;
        end
        else if (temp_result < $signed(29'h00026A09) && temp_result > $signed(29'h00006A09)) begin
            angle = 13'd45;
        end
        else angle = 13'd90;
    end
end
```



```
WB0: begin
    ird <= 1'b0;
    counter <= 0;
    convSum_x <= 0;
    convSum_y <= 0;
    if (center[6:0] == 7'd126) begin
        center <= {center[13:7]+7'd1, 7'd1};
    end
    else center <= center + 1;

    cwr_mag_0 <= 1'b1;
    cdata_mag_wr0 <= mag;
    caddr_mag_0 <= l0_addr;
    cwr_ang_0 <= 1'b1;
    cdata_ang_wr0 <= angle;
    caddr_ang_0 <= l0_addr;
    l0_addr <= l0_addr + 1;
end
```

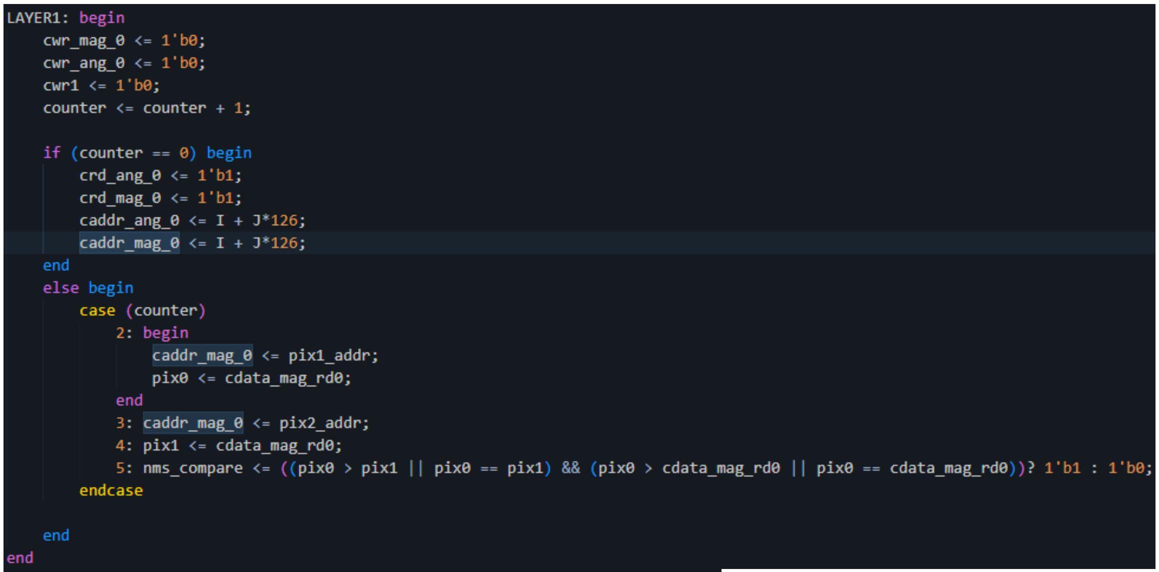
利用組合電路去計算 magnitude 與 angle，並在 WB0 做寫入的動作，同時也去移動到新的 center。



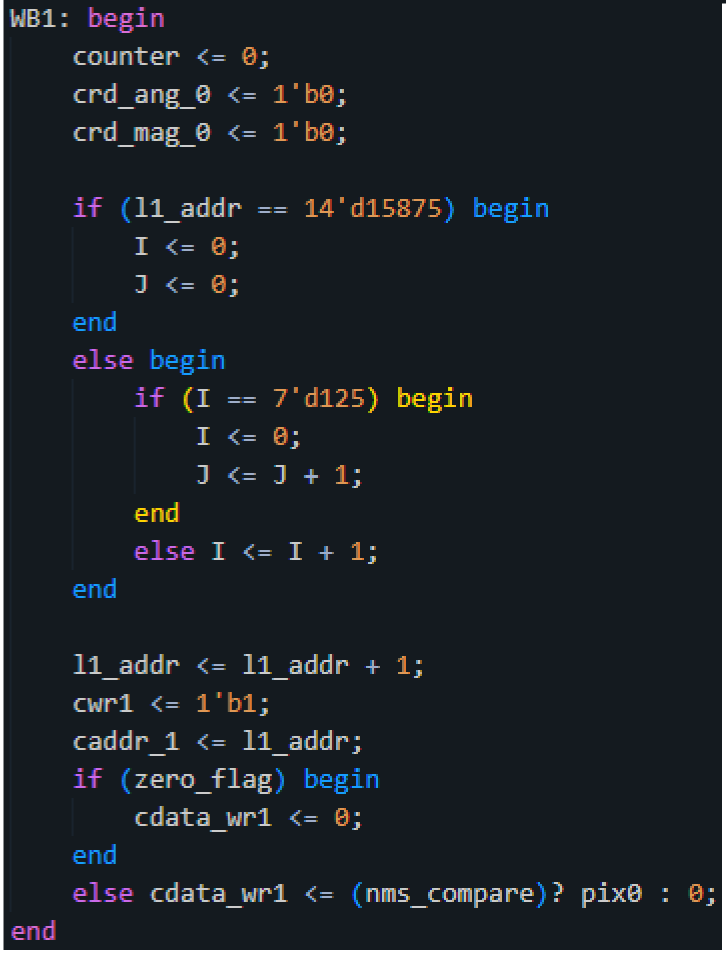


```
// For Layer1 ang Layer2
always @(*) begin
    if (J==0 || J==125 || I==0 || I==125) begin
        zero_flag = 1'b1;
    end

    else zero_flag = 1'b0;
end

// For Layer1
always @(*) begin
    case (cdata_ang_rd0)
        13'd45: begin
            pix1_addr = I + J*126 - 125;
            pix2_addr = I + J*126 + 125;
        end
        13'd135: begin
            pix1_addr = I + J*126 - 127;
            pix2_addr = I + J*126 + 127;
        end
        13'd0: begin
            pix1_addr = I + J*126 - 1;
            pix2_addr = I + J*126 + 1;
        end
        13'd90: begin
            pix1_addr = I + J*126 - 126;
            pix2_addr = I + J*126 + 126;
        end
        default: begin
            pix1_addr = 0;
            pix2_addr = 0;
        end
    endcase
end
```

在 Layer1 要做 NMS，我用 I, J 分別代表 x, y 座標，因為在邊緣的地方都是要寫入 0，因此我用 zero_flag 來判定當前 pixel 是否要寫入 0。如上面波形圖一開始一直在 State3(LAYER1)與 State4(WB1)跳動，就是因為一開始邊緣處都是要寫入 0。

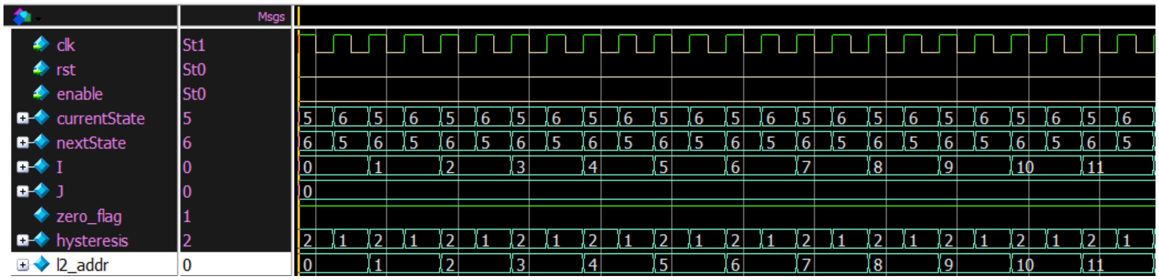


```
LAYER1: begin
    cwr_mag_0 <= 1'b0;
    cwr_ang_0 <= 1'b0;
    cwr1 <= 1'b0;
    counter <= counter + 1;

    if (counter == 0) begin
        crd_ang_0 <= 1'b1;
        crd_mag_0 <= 1'b1;
        caddr_ang_0 <= I + J*126;
        caddr_mag_0 <= I + J*126;
    end
    else begin
        case (counter)
            2: begin
                caddr_mag_0 <= pix1_addr;
                pix0 <= cdata_mag_rd0;
            end
            3: caddr_mag_0 <= pix2_addr;
            4: pix1 <= cdata_mag_rd0;
            5: nms_compare <= ((pix0 > pix1 || pix0 == pix1) && (pix0 > cdata_mag_rd0 || pix0 == cdata_mag_rd0))? 1'b1 : 1'b0;
        endcase

    end
end
```

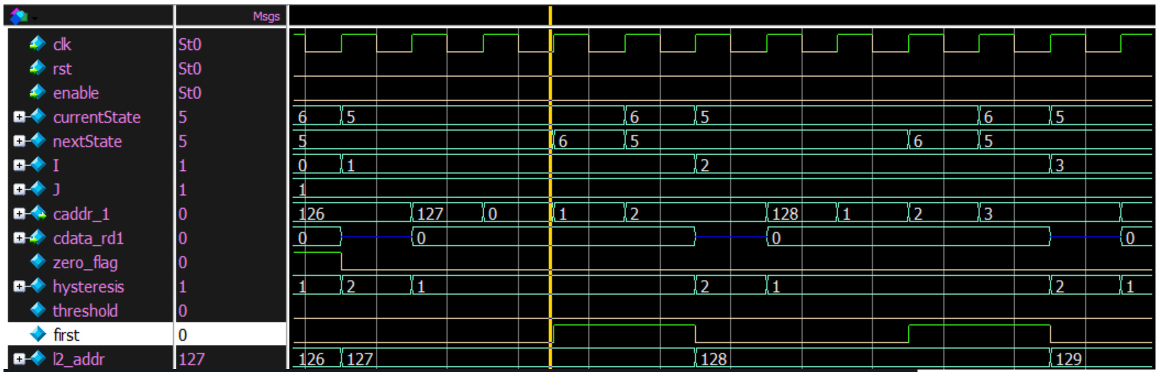


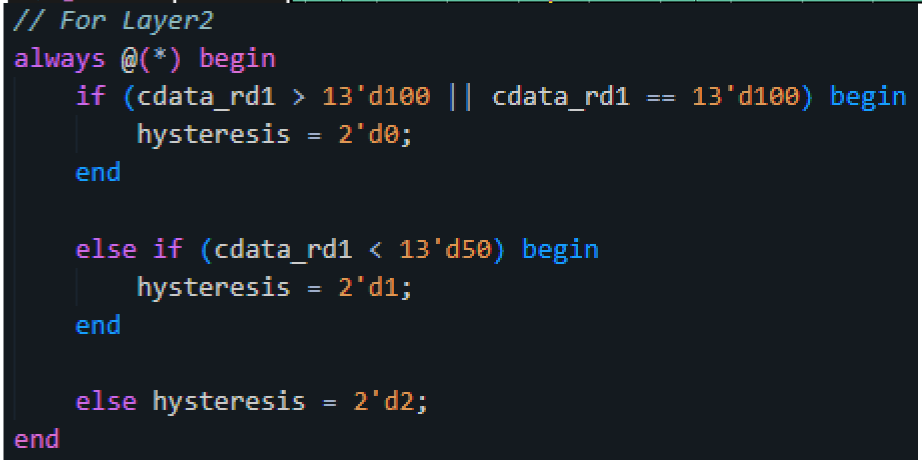
```
WB1: begin
    counter <= 0;
    crd_ang_0 <= 1'b0;
    crd_mag_0 <= 1'b0;

    if (l1_addr == 14'd15875) begin
        I <= 0;
        J <= 0;
    end
    else begin
        if (I == 7'd125) begin
            I <= 0;
            J <= J + 1;
        end
        else I <= I + 1;
    end

    l1_addr <= l1_addr + 1;
    cwr1 <= 1'b1;
    caddr_1 <= l1_addr;
    if (zero_flag) begin
        cdata_wr1 <= 0;
    end
    else cdata_wr1 <= (nms_compare)? pix0 : 0;
end
```

在不是邊緣處的 pixel 就需要根據 angle 來決定讀取對應的 pixel 並比較大小。nms_compare 為 1 時，代表當前 pixel 相較另外兩個 pixel 為最大值，所以在 WB1 時會寫入當前 pixel 值，反之則寫 0。

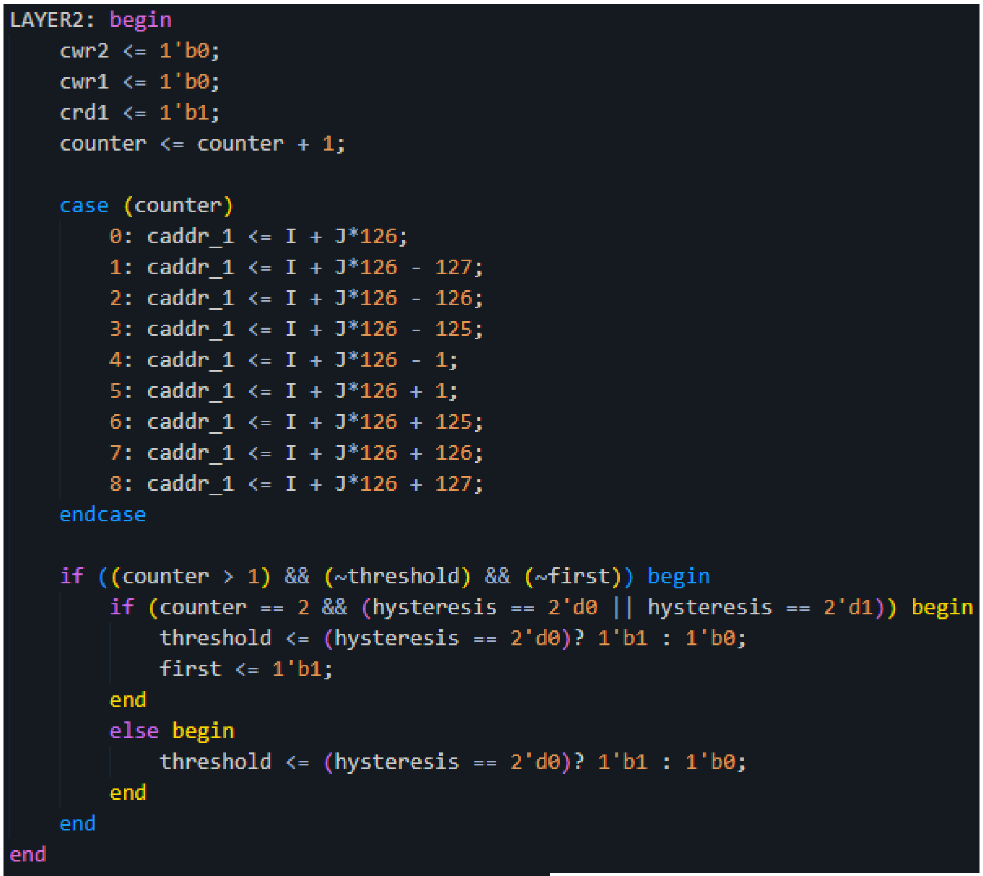


在 Layer2 要做 Hysteresis thresholding，波型圖一開一直在 State5(LAYER2)與 State6(WB2)跳動，原因跟上述一樣，也是利用 zero_flag 來判定。

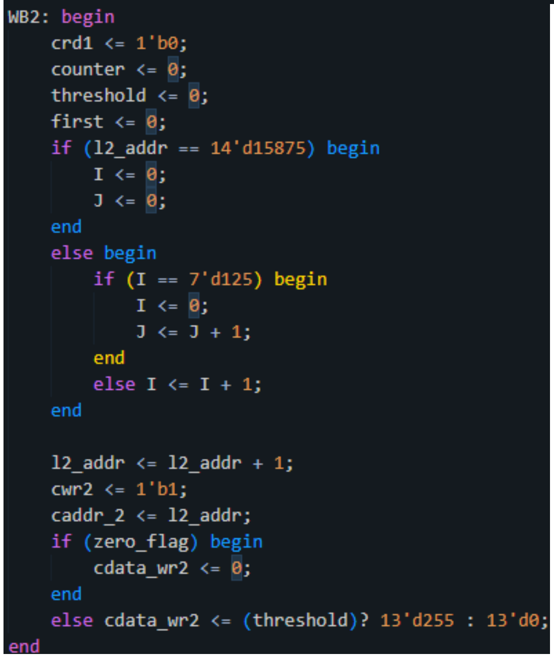




```
// For Layer2
always @(*) begin
    if (cdata_rd1 > 13'd100 || cdata_rd1 == 13'd100) begin
        hysteresis = 2'd0;
    end

    else if (cdata_rd1 < 13'd50) begin
        hysteresis = 2'd1;
    end

    else hysteresis = 2'd2;
end
```



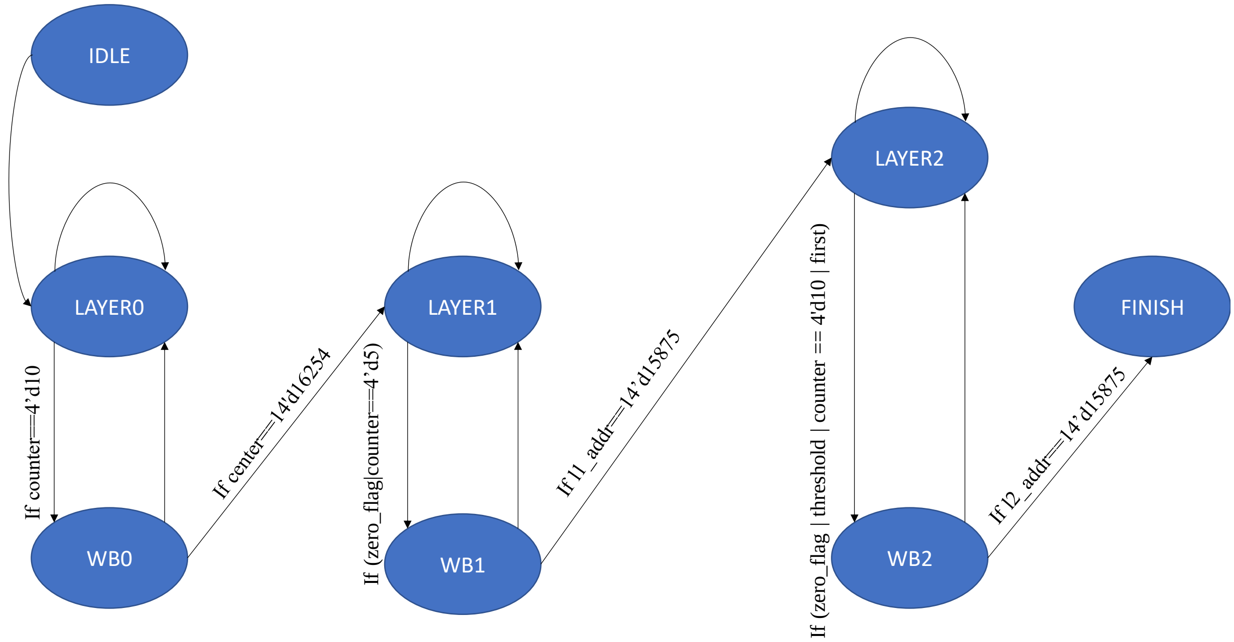
```
LAYER2: begin
    cwr2 <= 1'b0;
    cwr1 <= 1'b0;
    crd1 <= 1'b1;
    counter <= counter + 1;

    case (counter)
        0: caddr_1 <= I + J*126;
        1: caddr_1 <= I + J*126 - 127;
        2: caddr_1 <= I + J*126 - 126;
        3: caddr_1 <= I + J*126 - 125;
        4: caddr_1 <= I + J*126 - 1;
        5: caddr_1 <= I + J*126 + 1;
        6: caddr_1 <= I + J*126 + 125;
        7: caddr_1 <= I + J*126 + 126;
        8: caddr_1 <= I + J*126 + 127;
    endcase

    if ((counter > 1) && (~threshold) && (~first)) begin
        if (counter == 2 && (hysteresis == 2'd0 || hysteresis == 2'd1)) begin
            threshold <= (hysteresis == 2'd0)? 1'b1 : 1'b0;
            first <= 1'b1;
        end
        else begin
            threshold <= (hysteresis == 2'd0)? 1'b1 : 1'b0;
        end
    end
end
```



```
WB2: begin
    crd1 <= 1'b0;
    counter <= 0;
    threshold <= 0;
    first <= 0;
    if (l2_addr == 14'd15875) begin
        I <= 0;
        J <= 0;
    end
    else begin
        if (I == 7'd125) begin
            I <= 0;
            J <= J + 1;
        end
        else I <= I + 1;
    end

    l2_addr <= l2_addr + 1;
    cwr2 <= 1'b1;
    caddr_2 <= l2_addr;
    if (zero_flag) begin
        cdata_wr2 <= 0;
    end
    else cdata_wr2 <= (threshold)? 13'd255 : 13'd0;
end
```

在不是邊緣的 pixel 就會判別當前 pixel 是否大於 100 或小於 50，這裡也是用組合電路做判定，如果前述條件有一成立，就會跳到 WB2 寫入 255 或 0。若前述二條件都不成立，就會開始尋找周圍 8 個 pixel 是否有大於 100，若一找到大於 100 的 pixel 就會跳到 WB2，寫入 255，倘若周圍 8 個 pixel 都沒有大於 100，最後在 WB2 時會寫入 0。

3. Draw the flowchart for your Finite State Machine (FSM).



4. At last, please write the lesson you learned from Lab6.
   注意到 unsigned*signed 的小細節。因為 State 比之前作業多，所以這次學會更加精確控制 FSM。