

## 2025 Digital IC Design Homework 2

|  |           |          |          |          |
|--|-----------|----------|----------|----------|
| NAME   | 陳冠言       |          |          |          |
| Student ID   | P76134082 |          |          |          |
| Functional Simulation Result   |           |          |          |          |
| Pattern1   | Pattern2  | Pattern3 | Pattern4 | Pattern5 |
| Pass   | Pass      | Pass     | Pass     | Pass     |
| Pattern 1  |           |          |          |          |
| <pre>VSIM 6&gt; run -all # All data have been generated successfully! # #          /////////////////////////////////// #          /              /               __   #          / Congratulations !! /      / 0.0   #          /              /      /_____  #          / Simulation PASS !! /  / ^ ^ ^ \   #          /              /  / ^ ^ ^ ^  w  #          /// \m__m_ _  # # ** Note: \$finish      : D:/intelFPGA/DIC/HW2/testfixture.sv(162) #    Time: 1345 ns  Iteration: 0  Instance: /testfixture</pre>  |           |          |          |          |
| Pattern 2  |           |          |          |          |
| <pre>VSIM 8&gt; run -all # All data have been generated successfully! # #          /////////////////////////////////// #          /              /               __   #          / Congratulations !! /      / 0.0   #          /              /      /_____  #          / Simulation PASS !! /  / ^ ^ ^ \   #          /              /  / ^ ^ ^ ^  w  #          /// \m__m_ _  # # ** Note: \$finish      : D:/intelFPGA/DIC/HW2/testfixture.sv(162) #    Time: 1405 ns  Iteration: 0  Instance: /testfixture</pre>  |           |          |          |          |
| Pattern 3  |           |          |          |          |
| <pre>VSIM 10&gt; run -all # All data have been generated successfully! # #          /////////////////////////////////// #          /              /               __   #          / Congratulations !! /      / 0.0   #          /              /      /_____  #          / Simulation PASS !! /  / ^ ^ ^ \   #          /              /  / ^ ^ ^ ^  w  #          /// \m__m_ _  # # ** Note: \$finish      : D:/intelFPGA/DIC/HW2/testfixture.sv(162) #    Time: 1485 ns  Iteration: 0  Instance: /testfixture</pre> |           |          |          |          |

#### Pattern 4

```
VSIM 12> run -all
# All data have been generated successfully!
#
#
#      ///////////////////////////////////////////////////
#      /      Congratulations !!      /      |__||
#      /      Simulation PASS !!      /      / 0.0 |
#      /      /      ^ ^ ^ \      /      /_____|
#      /      /      ^ ^ ^ \      /      / ^ ^ ^ \ |
#      /      /      ^ ^ ^ \      /      | ^ ^ ^ |w|
#      ///////////////////////////////////////////////////      \m__m_|_|
#
#
# ** Note: $finish      : D:/intelFPGA/DIC/HW2/testfixture.sv(162)
#      Time: 1565 ns      Iteration: 0      Instance: /testfixture
```

#### Pattern 5

```
VSIM 14> run -all
# All data have been generated successfully!
#
#
#      ///////////////////////////////////////////////////
#      /      Congratulations !!      /      |__||
#      /      Simulation PASS !!      /      / 0.0 |
#      /      /      ^ ^ ^ \      /      /_____|
#      /      /      ^ ^ ^ \      /      / ^ ^ ^ \ |
#      /      /      ^ ^ ^ \      /      | ^ ^ ^ |w|
#      ///////////////////////////////////////////////////      \m__m_|_|
#
#
# ** Note: $finish      : D:/intelFPGA/DIC/HW2/testfixture.sv(162)
#      Time: 1605 ns      Iteration: 0      Instance: /testfixture
```

### LCD\_CTRL Finite-State Machine Design:

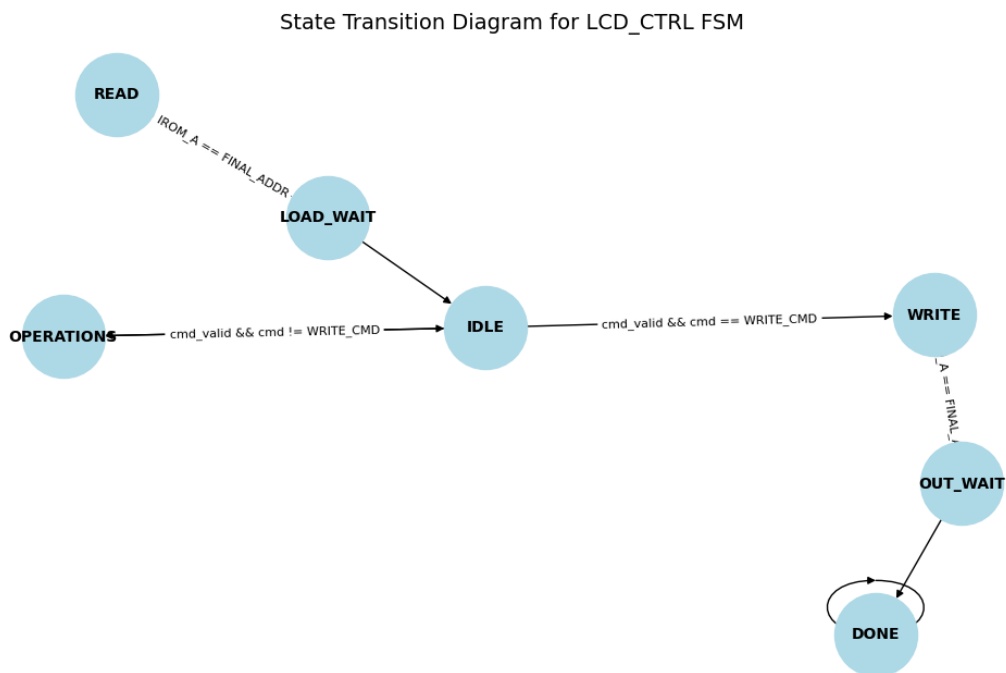
本次作業主要設計一個有限狀態機（FSM）來控制整個圖像處理流程。此狀態機會協調圖像讀取、處理和輸出等操作。以下是詳細的設計說明：

#### 狀態定義

1. READ (3'b000) :
  - 從 IROM 讀取圖像數據
  - 當 IROM\_A 達到最終地址（FINAL\_ADDR = 6'd63）時，轉移到 LOAD\_WAIT 狀態
2. LOAD\_WAIT (3'b101) :
  - 讀取完成後的等待狀態
  - 確保最後一個數據被正確讀取後再切換到 IDLE 狀態
3. IDLE (3'b001) :
  - 等待指令輸入的閒置狀態
  - 根據 cmd\_valid 和 cmd 決定下一個狀態：
    - 若 cmd\_valid 為高且 cmd 不是 WRITE\_CMD，則進入 OPERATIONS 狀態

- 若 cmd\_valid 為高且 cmd 是 WRITE\_CMD，則進入 WRITE 狀態
  - 否則保持在 IDLE 狀態
4. OPERATIONS (3'b010) :
    - 執行各種圖像處理操作（移動、MAX、MIN、AVERAGE）
    - 完成後返回 IDLE 狀態以等待新的指令
  5. WRITE (3'b011) :
    - 將處理後的圖像數據寫入 IRAM
    - 當 IRAM\_A 達到最終地址時，轉移到 OUT\_WAIT 狀態
  6. OUT\_WAIT (3'b110) :
    - 寫入完成後的等待狀態
    - 確保最後一個數據被正確寫入後再切換到 DONE 狀態
  7. DONE (3'b100) :
    - 整個處理流程完成的終止狀態
    - 保持在此狀態，直到系統重置

### 狀態轉換邏輯



### 控制訊號產生

每個狀態都對應特定的控制訊號組合，確保正確操作：

1. READ, LOAD\_WAIT 狀態 :
  - IROM\_rd = 1（啟用 IROM 讀取）
  - busy = 1（系統忙碌中）

2. **IDLE 狀態：**

- 所有控制訊號保持默認值 (IROM\_rd = 0, busy = 0, done = 0)

3. **OPERATIONS 狀態：**

- busy = 1 (系統忙碌中)

4. **WRITE, OUT\_WAIT 狀態：**

- IRAM\_ceb = 1 (啟用 IRAM 選取)
- IRAM\_web = 0 (啟用 IRAM 寫入)
- busy = 1 (系統忙碌中)

5. **DONE 狀態：**

- done = 1 (處理完成)

**數據處理與像素操作**

狀態機協調下，LCD 控制器執行以下操作：

1. **圖像讀取：**

- READ 狀態負責從 IROM 讀取圖像數據到內部緩衝區 ImageBuffer

2. **操作點移動：**

- 在 OPERATIONS 狀態下處理 UP\_CMD、DOWN\_CMD、LEFT\_CMD 和 RIGHT\_CMD
- 移動受到邊界限制 (TOP\_EDGE=2, BOTTOM\_EDGE=6, LEFT\_EDGE=2, RIGHT\_EDGE=6)

3. **像素處理：**

- 在 OPERATIONS 狀態下處理 MAX、MIN 和 AVERAGE 命令
- 計算 4x4 區域內所有像素的最大值、最小值或平均值
- 使用計算出的值更新該區域內的所有像素

4. **處理結果輸出：**

- WRITE 狀態負責將處理後的圖像數據從 ImageBuffer 寫入 IRAM