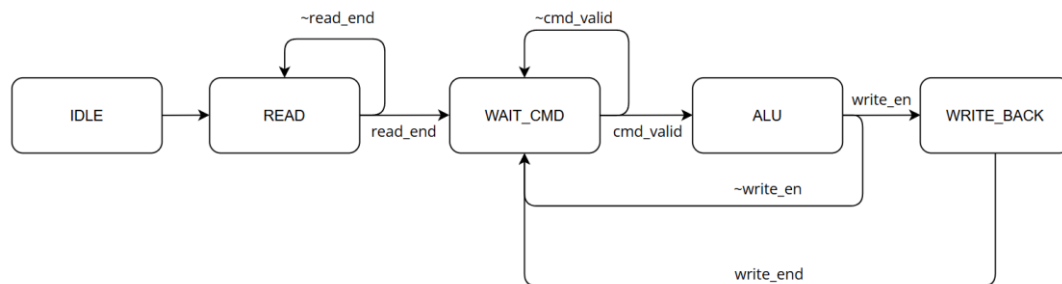


2025 Digital IC Design Homework 2

NAME	王華昀			
Student ID	N26134308			
Functional Simulation Result				
Pattern1	Pattern2	Pattern3	Pattern4	Pattern5
Pass	Pass	Pass	Pass	Pass
Pattern 1				
<pre># All data have been generated successfully! # # ////////////////////////////////// # / / __ # / Congratulations !! / / 0.0 # / / /_____ # / Simulation PASS !! / / ^ ^ ^ \ # / / ^ ^ ^ ^ w # / ////////////////////////////////// \m__m__ _ # #</pre>				
Pattern 2				
<pre># All data have been generated successfully! # # ////////////////////////////////// # / / __ # / Congratulations !! / / 0.0 # / / /_____ # / Simulation PASS !! / / ^ ^ ^ \ # / / ^ ^ ^ ^ w # / ////////////////////////////////// \m__m__ _ # #</pre>				
Pattern 3				
<pre># All data have been generated successfully! # # ////////////////////////////////// # / / __ # / Congratulations !! / / 0.0 # / / /_____ # / Simulation PASS !! / / ^ ^ ^ \ # / / ^ ^ ^ ^ w # / ////////////////////////////////// \m__m__ _ # #</pre>				
Pattern 4				
<pre># All data have been generated successfully! # # ////////////////////////////////// # / / __ # / Congratulations !! / / 0.0 # / / /_____ # / Simulation PASS !! / / ^ ^ ^ \ # / / ^ ^ ^ ^ w # / ////////////////////////////////// \m__m__ _ # #</pre>				
Pattern 5				

```
# All data have been generated successfully!
#
#
# //////////////////////////////////////
# /                                     /      |__||
# /   Congratulations !!             /      / 0.0 |
# /                                     /      /_____|
# /   Simulation PASS !!             /  / ^ ^ ^ \ |
# /                                     / | ^ ^ ^ ^ |w|
# ////////////////////////////////////// \m__m__|_|
#
#
```

LCD_CTRL Finite-State Machine Design:



這次的作業將設計分為五個 state，分別為 IDLE、READ、WAIT_CMD、ALU、WRITE BACK，以下詳細說明各個 state 的行為：

- IDLE:

當 rst 尚未拉下時，會一直卡在 IDLE 等待，在 rst 拉下的下個 cycle 會進入到 READ 狀態。

- READ:

READ 狀態主要是將 IROM 中的資料讀入內部的 buffer 中，因此 IROM_rd 在此狀態會拉高，並且 IROM_A 會每一個 clk cycle 增加 1，直到讀取完完整 IROM 的資料後(IROM_A==6'd63) read_end 會拉起，並在下一個 cycle 進入 WAIT_CMD 狀態。在 READ 狀態時，busy 會持續拉高，避免 cmd 讀入。

- WAIT CMD:

WAIT_CMD 狀態主要是等待 cmd_valid 拉起，在此狀態時 busy 會拉下使 cmd 傳入，若 cmd_valid 尚未拉起，代表 cmd 還沒有傳到，會在 WAIT_CMD 狀態等待，直到 cmd_valid 拉起才會進入 ALU 狀態。值得注意的是，由於 default 的運算做標示 (4,4)，並且若座標有進行 Shift 的動作時會在 ALU 階段就計算完畢，因此在 WAIT_CMD 狀態可以將需要進行計算的 16 個數值先準備好，確保所有運算都能在 ALU 狀態時一個 cycle 算完。

- ALU:

ALU 狀態主要是根據 cmd 的數值決定行為，主要可以分為三種行為，第一種是針對運算座標進行運算，也就是做 Shift 的動作。第二種是進行數值的運算，包括取最小值、取最大值、取平均，以上兩種行為完畢後，會回到 WAIT_CMD，等待下一條 cmd 傳入，並持續循環直到第三種行為到來。第三種也就是將資料寫回 IRAM 中，這時會將 write_en 拉起，進入到 WRITE_BACK。而上述動作皆會在一個 cycle 中完成。

- WRITE_BACK:

WRITE_BACK 狀態主要是將處理完的數值送回 IRAM 中，使用一個 counter 去進行 IRAM_A 的控制，當 $IRAM_A == 6'd63$ 即代表資料送完，將 write_end 拉起，而 busy 及 done 分別是 $\sim write_end$ 及 write_end。