

Lab1 Report

b06602037 徐子程

b06901165 郭哲聰

b07901016 朱哲廣

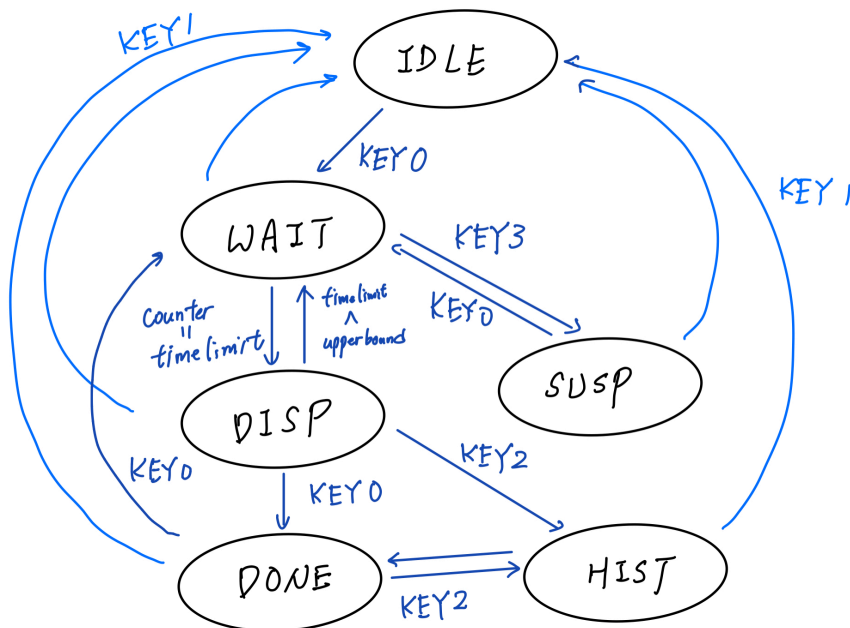
一、使用所需器材與架設方式

- DE2-115 FPGA board
- Personal computer

二、使用方式與詳細步驟

1. 按下 **KEY 0**，開始由快到慢隨機產生數字，最後會定格
 - 若在中途按下 **KEY 3**，會暫停在當下顯示的數字上
 - 再次按下 **KEY 0** 便會繼續隨機產生數字
2. 定格後若按下 **KEY 0**，將結果記錄起來，顯示 **00**，等待重新開始
3. 反之若按下 **KEY 2**，會顯示上一次的隨機產生的結果
 - 再次按下 **KEY 2**，顯示 **00**，等待重新開始
4. 任何情況按下 **KEY 1**，可強制 reset，所有數值歸 0
5. 在等待重新開始時 (顯示 **00**)，按下 **KEY 2**，會顯示剛才的隨機產生的結果
 - 再次按下 **KEY 2**，顯示 **00**，等待重新開始

三、實作設計技術細節與巧思



技術細節

states:

- **IDLE**：將所有變數初始化，等待重新開始。
- **WAIT**：產生亂數的過程都會在這個 state，當時間到達一定值，便會顯示當下的亂數 (到 DISP)
- **DISP**：顯示亂數，並在總時間未到達目標值時，跳回 WAIT 繼續產生下一次的隨機亂數。最終會定格在最後的亂數。
- **SUSP**：顯示亂數當下產生的值，但不會是最終結果
- **HIST**：顯示上一次的亂數結果

巧思

- **Random number seed 的產生**：在整個電路運作中，`cycle_time` 一直隨著 `clk` +1，並設計作為 `seed` 使用餵給亂數產生器。
- **漸慢顯示亂數實做**：使用兩個 counter 相互比較，決定何時應該要顯示下一個數字，`counter` 每個 clock cycle 都會 +1；`blink_counter` 會數到 `time_limit` 後顯示暫時的亂數。`time_limit` 則是等待的間隔，每隔一定 cycle 後左移一位 (x2)，可以使間隔時間拉長，達成漸慢效果。當 `counter` 大於 `TIME_UPPERBOUND` 則輸出最終亂數結果。
- **HIST 的邏輯**：邏輯上是製作兩個串在一起的 flip-flop，一個存最新的 random variable，另一個存上一次的 random variable，並在 `DONE` 這個 state 觸發時才寫入。

四、碰過的問題或挑戰與解決方式

- **Quartus II 環境架設**：我們希望在 Linux 下進行開發，從下載到安裝完成遇到大大小小的環境問題，例如缺少 library 及安裝程式當機，在 Google 大神的協助下，最後成功並能夠正常使用。
- **設計 History 的功能**：當初沒想好 History 的實作方式，導致連 FSM 本身邏輯都有問題，後來重新架構一次運作邏輯並修改 FSM，才得以解決。